

Special Issue Reprint

Advances in Quantum Computing

Edited by
Brian R. La Cour and Giuliano Benenti

mdpi.com/journal/entropy

Advances in Quantum Computing

Advances in Quantum Computing

Editors

Brian R. La Cour

Giuliano Benenti



Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Editors

Brian R. La Cour
Applied Research
Laboratories
The University of Texas at
Austin
Austin
USA

Giuliano Benenti
Department of Sciece and
High Technology
University of Insubria
Como
Italy

Editorial Office

MDPI
St. Alban-Anlage 66
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Entropy* (ISSN 1099-4300) (available at: www.mdpi.com/journal/entropy/special_issues/Adv_Quantum_Comput).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. <i>Journal Name</i> Year , <i>Volume Number</i> , Page Range.
--

ISBN 978-3-7258-0020-9 (Hbk)

ISBN 978-3-7258-0019-3 (PDF)

doi.org/10.3390/books978-3-7258-0019-3

© 2024 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license.

Contents

Brian La Cour

Advances in Quantum Computing

Reprinted from: *Entropy* **2023**, *25*, 1633, doi:10.3390/e25121633 1

Mingyoung Jeng, Alvir Nobel, Vinayak Jha, David Levy, Dylan Kneidel, Manu Chaudhary, et al.

Generalized Quantum Convolution for Multidimensional Data

Reprinted from: *Entropy* **2023**, *25*, 1503, doi:10.3390/e25111503 5

Brian García Sarmina, Guo-Hua Sun and Shi-Hai Dong

Principal Component Analysis and t-Distributed Stochastic Neighbor Embedding Analysis in the Study of Quantum Approximate Optimization Algorithm Entangled and Non-Entangled Mixing Operators

Reprinted from: *Entropy* **2023**, *25*, 1499, doi:10.3390/e25111499 26

Naya Nagy, Marius Nagy, Ghadeer Alazman, Zahra Hawaidi, Saja Mustafa Alsulaibikh, Layla Alabbad, et al.

Quantum Honey pots

Reprinted from: *Entropy* **2023**, *25*, 1461, doi:10.3390/e25101461 70

Chen-Fu Chiang and Paul M. Alsing

Quantum-Walk-Inspired Dynamic Adiabatic Local Search

Reprinted from: *Entropy* **2023**, *25*, 1287, doi:10.3390/e25091287 85

Saad M. Darwish, Ibrahim Abdullah Mhaimed and Adel A. Elzoghbi

A Quantum Genetic Algorithm for Building a Semantic Textual Similarity Estimation Framework for Plagiarism Detection Applications

Reprinted from: *Entropy* **2023**, *25*, 1271, doi:10.3390/e25091271 98

Wenyang Qian, Robert A. M. Basili, Mary Mehrnoosh Eshaghian-Wilner, Ashfaq Khokhar, Glenn Luecke and James P. Vary

Comparative Study of Variations in Quantum Approximate Optimization Algorithms for the Traveling Salesman Problem

Reprinted from: *Entropy* **2023**, *25*, 1238, doi:10.3390/e25081238 122

Kailu Zhang, Jingyang Liu, Huajian Ding, Xingyu Zhou, Chunhui Zhang and Qin Wang

Asymmetric Measurement-Device-Independent Quantum Key Distribution through Advantage Distillation

Reprinted from: *Entropy* **2023**, *25*, 1174, doi:10.3390/e25081174 143

Sebastian Raubitzek and Kevin Mallinger

On the Applicability of Quantum Machine Learning

Reprinted from: *Entropy* **2023**, *25*, 992, doi:10.3390/e25070992 152

Jie Gao, Yinuo Wang, Zhaoyang Song and Shumei Wang

Quantum Image Encryption Based on Quantum DNA Codec and Pixel-Level Scrambling

Reprinted from: *Entropy* **2023**, *25*, 865, doi:10.3390/e25060865 184

Manuel John, Julian Schuhmacher, Panagiotis Barkoutsos, Ivano Tavernelli and Francesco Tacchino

Optimizing Quantum Classification Algorithms on Classical Benchmark Datasets

Reprinted from: *Entropy* **2023**, *25*, 860, doi:10.3390/e25060860 200

Mark Goldsmith, Harto Saarinen, Guillermo García-Pérez, Joonas Malmi, Matteo A. C. Rossi and Sabrina Maniscalco Link Prediction with Continuous-Time Classical and Quantum Walks Reprinted from: <i>Entropy</i> 2023 , <i>25</i> , 730, doi:10.3390/e25050730	213
Gabriele Cenedese, Maria Bondani, Dario Rosa and Giuliano Benenti Generation of Pseudo-Random Quantum States on Actual Quantum Processors Reprinted from: <i>Entropy</i> 2023 , <i>25</i> , 607, doi:10.3390/e25040607	228
Corey Jason Trahan, Mark Loveland, Noah Davis and Elizabeth Ellison A Variational Quantum Linear Solver Application to Discrete Finite-Element Methods Reprinted from: <i>Entropy</i> 2023 , <i>25</i> , 580, doi:10.3390/e25040580	241
Emanuele Dri, Antonello Aita, Edoardo Giusto, Davide Ricossa, Davide Corbelleto, Bartolomeo Montrucchio and Roberto Ugoccioni A More General Quantum Credit Risk Analysis Framework Reprinted from: <i>Entropy</i> 2023 , <i>25</i> , 593, doi:10.3390/e25040593	263
Hanjing Xu, Samudra Dasgupta, Alex Pothen and Arnab Banerjee Dynamic Asset Allocation with Expected Shortfall via Quantum Annealing Reprinted from: <i>Entropy</i> 2023 , <i>25</i> , 541, doi:10.3390/e25030541	277
Krzysztof Domino, Mátyás Koniorczyk, Krzysztof Krawiec, Konrad Jałowiecki, Sebastian Deffner and Bartłomiej Gardas Quantum Annealing in the NISQ Era: Railway Conflict Management Reprinted from: <i>Entropy</i> 2023 , <i>25</i> , 191, doi:10.3390/e25020191	296
Congcong Feng, Bo Zhao, Xin Zhou, Xiaodong Ding and Zheng Shan An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance Reprinted from: <i>Entropy</i> 2023 , <i>25</i> , 127, doi:10.3390/e25010127	328
Francisco Revson F. Pereira and Stefano Mancini Entanglement-Assisted Quantum Codes from Cyclic Codes Reprinted from: <i>Entropy</i> 2023 , <i>25</i> , 37, doi:10.3390/e25010037	340
Wenlin Zhao, YINUO Wang, Yingjie Qu, Hongyang Ma and Shumei Wang Binary Classification Quantum Neural Network Model Based on Optimized Grover Algorithm Reprinted from: <i>Entropy</i> 2022 , <i>24</i> , 1783, doi:10.3390/e24121783	352
Thibault Fredon, Julien Zylberman, Pablo Arnault and Fabrice Debbsch Quantum Spatial Search with Electric Potential: Long-Time Dynamics and Robustness to Noise Reprinted from: <i>Entropy</i> 2022 , <i>24</i> , 1778, doi:10.3390/e24121778	370
Sergey Tarasov, William Shannon, Vladimir Kocharovsky and Vitaly Kocharovsky Multi-Qubit Bose–Einstein Condensate Trap for Atomic Boson Sampling Reprinted from: <i>Entropy</i> 2022 , <i>24</i> , 1771, doi:10.3390/e24121771	383
Andreas Burger, Leong Chuan Kwek and Dario Poletti Digital Quantum Simulation of the Spin-Boson Model under Markovian Open-System Dynamics Reprinted from: <i>Entropy</i> 2022 , <i>24</i> , 1766, doi:10.3390/e24121766	417
Abdirahman Alasow, Peter Jin and Marek Perkowski Quantum Algorithm for Variant Maximum Satisfiability [†] Reprinted from: <i>Entropy</i> 2022 , <i>24</i> , 1615, doi:10.3390/e24111615	437

Lihui Lv, Bao Yan, Hong Wang, Zhi Ma, Yangyang Fei, Xiangdong Meng and Qianheng Duan
Using Variational Quantum Algorithm to Solve the LWE Problem
Reprinted from: *Entropy* **2022**, *24*, 1428, doi:10.3390/e24101428 **461**

Advances in Quantum Computing

Brian La Cour

Applied Research Laboratories, The University of Texas at Austin, Austin, TX 78758, USA;
blacour@arlut.utexas.edu

Advances in quantum computing have continued to accelerate over the course of this Special Issue's publication. In the past two years, we have observed major breakthroughs in hardware and algorithm development, as well as new, deep theoretical insights. In November 2022, IBM announced their record-breaking 433-qubit quantum computer, Osprey, with plans for developing a 100,000-qubit machine in the next ten years. In June 2023, the University of Science and Technology of China (USTC) first made available to global users their 176-qubit *Zuchongzhi* quantum computer, a successor to the *Zuchongzhi 2.1*, which they claim has a record quantum computational advantage of 1.0×10^8 in sampling random circuits [1]. Shortly thereafter, in October 2023, USTC announced a breakthrough Gaussian boson sampling photonic experiment using their new *Jiuzhang 3.0* that boasts a quantum computational advantage of 1.5×10^{16} [2]. Meanwhile, in September 2023, PsiQuantum announced plans to build a one-million-qubit commercial photonic fusion-based quantum computer within the next six years.

This Special Issue has endeavored to capture some of the technical advances in this rapidly changing field. In Mingyoung Jeng et al.'s study, we find a novel method of producing depth-optimized circuits for performing multidimensional convolutions using a quantum computer (contribution 1). Brian García Sarmina and colleagues provide greater insight into the quantum approximate optimization algorithm by demonstrating that entanglement-based models possess an enhanced capacity to preserve information and maintain correlations over non-entangled models (contribution 2). Naya Nagy et al. propose a unique quantum honey pot scheme for detecting intruders using covert quantum sentinels to subtly detect when quantum information has been measured (contribution 3). Chen-Fu Chiang and Paul Alsing reconcile discrepancies between continuous-time quantum walk and adiabatic quantum computing optimization, with the latter option containing more structure than the former, using a modified catalyst Hamiltonian (contribution 4). Saad Darwish and co-workers describe a novel hybrid algorithm using semantic extraction and the quantum genetic algorithm to perform plagiarism detection, with simulations showing up to 20% improvement compared to classical genetic algorithms (contribution 5). Wenyang Qian et al. studied the use of the quantum approximate optimization algorithm (QAOA) to solve the traveling salesman problem, finding that well-balanced mixers usually outperform other QAOA mixer ansatzes (contribution 6).

From computing to communication, Kailu Zhang and colleagues propose an asymmetric measurement-device-independent quantum key distribution (MDI-QKD) protocol, the secure key rate of which is enhanced through advantage distillation (contribution 7). Sebastian Raubitsek and Kevin Mallinger investigate the applicability of quantum machine learning for classification and show that the variational quantum circuit and quantum kernel estimation methods perform better than basic machine learning algorithms (contribution 8). Jie Gao and co-workers consider enhanced quantum image encryption techniques, using a quantum DNA codec to enhance security and robustness (contribution 9). Manuel John et al. examine quantum kernel methods applied to quantum machine learning and develop several enhancements to provide significant performance improvements for several real-world classification tasks (contribution 10). Mark Goldsmith et al. study the use of quantum random walks to predict links within protein–protein interaction networks

Citation: La Cour, B. Advances in Quantum Computing. *Entropy* **2023**, *25*, 1633. <https://doi.org/10.3390/e25121633>

Received: 4 December 2023

Accepted: 7 December 2023

Published: 8 December 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

and demonstrate these walks' ability to outperform classical random walks (contribution 11). Gabriele Cenedese and colleagues propose a method to efficiently generate random quantum circuits that result in high degrees of entanglement and use this as a benchmark for real-world quantum computers (contribution 12). Corey Trahan and colleagues demonstrate the application of hybrid quantum variational solvers to discrete solutions of partial differential equations, showing that they scale polylogarithmically based on the system size (contribution 13).

For business applications, Emanuele Dri et al. develop both new and generalized variants of current credit risk analysis quantum algorithms and test them using both simulated and real-world quantum computers (contribution 14). Similarly, Hanjing Xu and co-workers formulate an investment portfolio problem as a quadratic unconstrained binary optimization problem and solve a real-world example using D-Wave quantum annealers, resulting in optimized portfolios with more than an 80% return over classically derived ones (contribution 15). In contrast, Krzysztof Domino et al. use the D-Wave quantum annealer to solve real-world Polish railway network problems, finding that neither the 2000Q nor Advantage machines performed well in terms of solving these problems (contribution 16).

Congcong Feng and co-workers propose a new polar-based similarity metric for the K-nearest neighbor (KNN) algorithm and show that its use significantly improves performance, albeit only for the quantum version of KNN (contribution 17). Francisco Pereira and Stefano Mancini provide a general procedure for producing entanglement-assisted quantum-error-correction codes, providing better error protection compared to traditional stabilizer-based codes (contribution 18). Wenlin Zhao and co-workers propose a binary quantum neural network classification model based on an optimized Grover algorithm and partial diffusion, showing better performance over existing quantum neural network classification models (contribution 19). Thibault Fredon et al. perform spatial searches using a 2D grid with a coulomb potential and apply a discrete-time quantum random walk to demonstrate a quadratic scaling of the localization solution time that is robust against noise (contribution 20).

Sergey Tarasov and colleagues consider a Bose–Einstein condensate for studying quantum statistical phenomena and, in particular, how it might be used to perform Gaussian boson sampling (contribution 21). Andreas Burger and co-workers use a seven-qubit IBM quantum computer to perform digital quantum simulations of harmonically coupled spins and demonstrate the emergence of correlations (contribution 22). Abdirahman Alasow and colleagues study the use of Grover's search algorithm to solve the maximum satisfiability problem and devise a novel quantum counter block within the standard oracle, demonstrating a significant reduction in the required number of ancilla qubits (contribution 23). Finally, Lihui Lv et al. use a variational quantum algorithm to solve the learning-with-errors problem and show a speed increase compared to classical solvers (contribution 24).

This Special Issue has shown the diversity of problems that can be solved using quantum computers, both existing noisy, intermediate-scale devices and future, fault-tolerant devices. With the recent rapid advances in quantum computing hardware, it should soon become apparent whether the promise of quantum computing anticipated in these works reflects the actual performance of future quantum computing technologies.

Conflicts of Interest: The author declares no conflict of interest.

List of Contributions

1. Jeng, M.; Nobel, A.; Jha, V.; Levy, D.; Kneidel, D.; Chaudhary, M.; Islam, I.; Rahman, M.M.; El-Araby, E. Generalized Quantum Convolution for Multidimensional Data. *Entropy* **2023**, *25*, 1503. <https://doi.org/10.3390/e25111503>.

2. Sarmina, B.G.; Sun, G.-H.; Dong, S.-H. Principal Component Analysis and t-Distributed Stochastic Neighbor Embedding Analysis in the Study of Quantum Approximate Optimization Algorithm Entangled and Non-Entangled Mixing Operators. *Entropy* **2023**, *25*, 1499. <https://doi.org/10.3390/e25111499>.
3. Nagy, N.; Nagy, M.; Alazman, G.; Hawaidi, Z.; Alsulaibikh, S.M.; Alabbad, L.; Alfaleh, S.; Aljuaid, A. Quantum Honey Pots. *Entropy* **2023**, *25*, 1461. <https://doi.org/10.3390/e25101461>.
4. Chiang, C.-F.; Alsing, P.M. Quantum-Walk-Inspired Dynamic Adiabatic Local Search. *Entropy* **2023**, *25*, 1287. <https://doi.org/10.3390/e25091287>.
5. Darwish, S.M.; Mhaimeed, I.A.; Elzoghbi, A.A. A Quantum Genetic Algorithm for Building a Semantic Textual Similarity Estimation Framework for Plagiarism Detection Applications. *Entropy* **2023**, *25*, 1271. <https://doi.org/10.3390/e25091271>.
6. Qian, W.; Basili, R.A.M.; Eshaghian-Wilner, M.M.; Khokhar, A.; Luecke, G.; Vary, J.P. Comparative Study of Variations in Quantum Approximate Optimization Algorithms for the Traveling Salesman Problem. *Entropy* **2023**, *25*, 1238. <https://doi.org/10.3390/e25081238>.
7. Zhang, K.; Liu, J.; Ding, H.; Zhou, X.; Zhang, C.; Wang, Q. Asymmetric Measurement-Device-Independent Quantum Key Distribution through Advantage Distillation. *Entropy* **2023**, *25*, 1174. <https://doi.org/10.3390/e25081174>.
8. Raubitzek, S.; Mallinger, K. On the Applicability of Quantum Machine Learning. *Entropy* **2023**, *25*, 992. <https://doi.org/10.3390/e25070992>.
9. Gao, J.; Wang, Y.; Song, Z.; Wang, S. Quantum Image Encryption Based on Quantum DNA Codec and Pixel-Level Scrambling. *Entropy* **2023**, *25*, 865. <https://doi.org/10.3390/e25060865>.
10. John, M.; Schuhmacher, J.; Barkoutsos, P.; Tavernelli, I.; Tacchino, F. Optimizing Quantum Classification Algorithms on Classical Benchmark Datasets. *Entropy* **2023**, *25*, 860. <https://doi.org/10.3390/e25060860>.
11. Goldsmith, M.; Saarinen, H.; García-Pérez, G.; Malmi, J.; Rossi, M.A.C.; Maniscalco, S. Link Prediction with Continuous-Time Classical and Quantum Walks. *Entropy* **2023**, *25*, 730. <https://doi.org/10.3390/e25050730>.
12. Cenedese, G.; Bondani, M.; Rosa, D.; Benenti, G. Generation of Pseudo-Random Quantum States on Actual Quantum Processors. *Entropy* **2023**, *25*, 607. <https://doi.org/10.3390/e25040607>.
13. Trahan, C.J.; Loveland, M.; Davis, N.; Ellison, E. A Variational Quantum Linear Solver Application to Discrete Finite-Element Methods. *Entropy* **2023**, *25*, 580. <https://doi.org/10.3390/e25040580>.
14. Dri, E.; Aita, A.; Giusto, E.; Ricossa, D.; Corbelleto, D.; Montrucchio, B.; Ugoccioni, R. A More General Quantum Credit Risk Analysis Framework. *Entropy* **2023**, *25*, 593. <https://doi.org/10.3390/e25040593>.
15. Xu, H.; Dasgupta, S.; Pothen, A.; Banerjee, A. Dynamic Asset Allocation with Expected Shortfall via Quantum Annealing. *Entropy* **2023**, *25*, 541. <https://doi.org/10.3390/e25030541>.
16. Domino, K.; Koniorczyk, M.; Krawiec, K.; Jałowiecki, K.; Deffner, S.; Gardas, B. Quantum Annealing in the NISQ Era: Railway Conflict Management. *Entropy* **2023**, *25*, 191. <https://doi.org/10.3390/e25020191>.
17. Feng, C.; Zhao, B.; Zhou, X.; Ding, X.; Shan, Z. An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance. *Entropy* **2023**, *25*, 127. <https://doi.org/10.3390/e25010127>.
18. Pereira, F.R.F.; Mancini, S. Entanglement-Assisted Quantum Codes from Cyclic Codes. *Entropy* **2023**, *25*, 37. <https://doi.org/10.3390/e25010037>.
19. Zhao, W.; Wang, Y.; Qu, Y.; Ma, H.; Wang, S. Binary Classification Quantum Neural Network Model Based on Optimized Grover Algorithm. *Entropy* **2022**, *24*, 1783. <https://doi.org/10.3390/e24121783>.

20. Fredon, T.; Zylberman, J.; Arnault, P.; Debbasch, F. Quantum Spatial Search with Electric Potential: Long-Time Dynamics and Robustness to Noise. *Entropy* **2022**, *24*, 1778. <https://doi.org/10.3390/e24121778>.
21. Tarasov, S.; Shannon, W.; Kocharovsky, V.; Kocharovsky, V. Multi-Qubit Bose–Einstein Condensate Trap for Atomic Boson Sampling. *Entropy* **2022**, *24*, 1771. <https://doi.org/10.3390/e24121771>.
22. Burger, A.; Kwek, L.C.; Poletti, D. Digital Quantum Simulation of the Spin-Boson Model under Markovian Open-System Dynamics. *Entropy* **2022**, *24*, 1766. <https://doi.org/10.3390/e24121766>.
23. Alasow, A.; Jin, P.; Perkowski, M. Quantum Algorithm for Variant Maximum Satisfiability. *Entropy* **2022**, *24*, 1615. <https://doi.org/10.3390/e24111615>.
24. Lv, L.; Yan, B.; Wang, H.; Ma, Z.; Fei, Y.; Meng, X.; Duan, Q. Using Variational Quantum Algorithm to Solve the LWE Problem. *Entropy* **2022**, *24*, 1428. <https://doi.org/10.3390/e24101428>.

References

1. Zhu, Q.; Cao, S.; Chen, F.; Chen, M.C.; Chen, X.; Chung, T.H.; Deng, H.; Du, Y.; Fan, D.; Gong, M.; et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Sci. Bull.* **2022**, *67*, 240–245. [CrossRef] [PubMed]
2. Deng, Y.H.; Gu, Y.C.; Liu, H.L.; Gong, S.Q.; Su, H.; Zhang, Z.J.; Tang, H.Y.; Jia, M.H.; Xu, J.M.; Chen, M.C.; et al. Gaussian Boson Sampling with Pseudo-Photon-Number-Resolving Detectors and Quantum Computational Advantage. *Phys. Rev. Lett.* **2023**, *131*, 150601. [CrossRef] [PubMed]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Generalized Quantum Convolution for Multidimensional Data

Mingyoung Jeng *, Alvir Nobel, Vinayak Jha, David Levy, Dylan Kneidel, Manu Chaudhary, Ishraq Islam, Muhammad Momin Rahman and Esam El-Araby

Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA; islam.alvir@ku.edu (A.N.); vinayakjha@ku.edu (V.J.); david.levy@ku.edu (D.L.); dckneidel@ku.edu (D.K.); manu.chaudhary@ku.edu (M.C.); ishraq@ku.edu (I.I.); momin.rahman@ku.edu (M.M.R.); esam@ku.edu (E.E.-A.)

* Correspondence: mingyoungjeng@ku.edu

Abstract: The convolution operation plays a vital role in a wide range of critical algorithms across various domains, such as digital image processing, convolutional neural networks, and quantum machine learning. In existing implementations, particularly in quantum neural networks, convolution operations are usually approximated by the application of filters with data strides that are equal to the filter window sizes. One challenge with these implementations is preserving the spatial and temporal localities of the input features, specifically for data with higher dimensions. In addition, the deep circuits required to perform quantum convolution with a unity stride, especially for multidimensional data, increase the risk of violating decoherence constraints. In this work, we propose depth-optimized circuits for performing generalized multidimensional quantum convolution operations with unity stride targeting applications that process data with high dimensions, such as hyperspectral imagery and remote sensing. We experimentally evaluate and demonstrate the applicability of the proposed techniques by using real-world, high-resolution, multidimensional image data on a state-of-the-art quantum simulator from IBM Quantum.

Keywords: convolution; quantum algorithms; quantum image processing; quantum computing

Citation: Jeng, M.; Nobel, A.; Jha, V.; Levy, D.; Kneidel, D.; Chaudhary, M.; Islam, I.; Rahman, M.M.; El-Araby, E. Generalized Quantum Convolution for Multidimensional Data. *Entropy* **2023**, *25*, 1503. <https://doi.org/10.3390/e25111503>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 18 September 2023

Revised: 20 October 2023

Accepted: 27 October 2023

Published: 31 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Convolution is a common operation that is leveraged in a wide variety of practical applications, such as signal processing [1], image processing [2], and most recently, convolutional neural networks [3]. However, leveraging the widespread utility of convolution operations in quantum algorithms is limited by the lack of a systematic, generalized implementation of quantum convolution. Specifically, contemporary quantum circuits for performing quantum convolution with a given filter are designed on a case-by-case basis [4–8]. In other words, implementing a novel convolution filter on a quantum computer is arduous and time consuming, requiring substantial human effort. Such a workflow is impractical for applications, such as quantum convolutional neural networks, which require a generalized, parameterized quantum circuit to iteratively test thousands of unique filters per training cycle.

In this work, we propose a generalizable algorithm for quantum convolution compatible with amplitude-encoded multidimensional data that is able to implement arbitrary multidimensional filters. Furthermore, our proposed technique implements unity stride, which is essential for capturing the totality of local features in input data. We experimentally verify our technique by applying multiple filters on high-resolution, multidimensional images and report the fidelity of the quantum results against the classically computed expectations. The quantum circuits are implemented on a state-of-the-art quantum simulator from IBM Quantum [9] in both noise-free (as a statevector) and noisy environments. Compared to classical CPU- and GPU-based implementations of convolution, we achieve an exponential improvement in time complexity with respect to data size. Additionally,

when compared to existing quantum implementations, we achieve improved circuit depth complexity when factoring in the data encoding.

The work is structured as follows. In Section 2, we cover important background information and review the related work. In Section 3, we introduce the proposed quantum convolution circuits and provide analyses of the corresponding circuit depth. In Section 4, we present the experimental setup and results, while in Section 5, we provide discussions of the results and comparisons to related work. Finally, in Section 6, we present our conclusions and potential avenues for future explorations and extensions.

2. Background

In this section, we discuss related work pertinent to quantum convolution. Quantum operations that are relevant to convolution, such as quantum data encoding, and quantum shift operation, are also presented.

2.1. Related Work

Classically, the convolution operation is implemented either directly or by leveraging *fast Fourier transform* (FFT). On CPUs, the direct implementation has a time complexity of $\mathcal{O}(N^2)$ [10], where N is the data size, while FFT-based implementation has a time complexity of $\mathcal{O}(N \log N)$ [10]. On GPUs, the FFT-based implementation has a similar $\mathcal{O}(N \log N)$ complexity [11]. It is also common to take advantage of the parallelism offered by GPUs to implement convolution using general matrix multiplications (GEMMs) with $\mathcal{O}(N_F N)$ FLOPS [12,13], where N_F is the filter size.

Techniques for performing quantum convolution have previously been reported [4–8]. However, these techniques only use fixed sizes of filter windows for specific filters, e.g., edge detection [4–8]. We will denote such methods as *fixed-filter* quantum convolution. Reportedly, these methods possess a quadratic circuit depth complexity of $\mathcal{O}(n^2)$ in terms of the number of qubits $n = \lceil \log_2 N \rceil$, where N is the size of the input data [4–8]. Because the shortest execution time of classical convolution is in the order of $\mathcal{O}(N)$ or $\mathcal{O}(2^n)$ [12,13] with respect to data size N , authors of the quantum counterparts often claim a quantum advantage [4]. However, the reported depth complexity of *fixed-filter* quantum convolution does not include the unavoidable overhead of data encoding. Furthermore, there does not exist, to the best of our knowledge, a method for performing generalized, multidimensional quantum convolution.

In reported related work [4–8], data encoding is performed with either the *flexible representation of quantum images* (FRQI) [14] or *novel enhanced quantum representation* (NEQR) [15] methods. In these encoding techniques, positional information is stored in the basis quantum states of n qubits, while color information is stored via angle encoding and basis encoding for FRQI and NEQR, respectively. FRQI and NEQR require a total of $n + 1$ and $n + q$ qubits, respectively, where q is the number of qubits used to represent color values, e.g., $q = 8$ for standard grayscale pixel representation. The reported circuit depth complexities of FRQI and NEQR are $\mathcal{O}(4^n)$ and $\mathcal{O}(qn2^n)$, respectively. When factoring in the depth complexities of either data-encoding technique, it is evident that the referenced *fixed-filter* quantum convolution techniques should be expected to perform worse than classical implementations.

In [16], the authors propose a method of edge detection based on amplitude encoding and the quantum wavelet transform (QWT), which they denote as *quantum Hadamard edge detection* (QHED). Although the work utilizes grayscale two-dimensional images, the QHED technique is highly customized for those data and does not easily scale or generalize to data of higher dimensions, such as colored and/or multispectral images. For example, the quantum discriminator operation in their technique is applied over all qubits in the circuit, without distinguishing between qubits representing each dimension, i.e., image rows or columns. Such a procedure not only forgoes parallelism and increases circuit depth but inhibits the algorithm's ability to be generalized beyond capturing one-dimensional features.

In our proposed work, we achieve an exponential improvement in time complexity compared to classical implementations of convolution with respect to data size. Additionally, when compared to existing quantum convolution implementations, we achieve improved circuit depth complexity when factoring in the data encoding. The contribution of our work is analyzed, experimentally verified, and discussed in detail in Section 5.

2.2. Classical to Quantum (C2Q)

Our method of quantum convolution leverages *amplitude encoding*, which encodes N data values directly in the complex probability amplitudes $c_i \in \mathbb{C}$ of the positional basis state $|i\rangle$ for an n -qubit state $|\psi\rangle$, where $n = \lceil \log_2 N \rceil$ and $0 \leq i < n$, see (1):

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle : c_i \in \mathbb{C}. \tag{1}$$

We use the *classical-to-quantum* (C2Q) [17] data-encoding technique to encode the amplitude encoded state $|\psi_0\rangle$ from the ground state $|0\rangle^{\otimes n}$, see Figure 1 and (2). The C2Q operation U_{C2Q} has a circuit depth complexity of $\mathcal{O}(2^n)$, a quadratic and linear improvement over FRQI and NEQR, respectively:

$$U_{C2Q}|0\rangle^{\otimes n} = |\psi_0\rangle$$

$$U_{C2Q} = [|\psi_0\rangle \quad | \times \rangle \cdots | \times \rangle], \text{ where} \tag{2}$$

$$| \times \rangle = \text{“don’t care”}$$

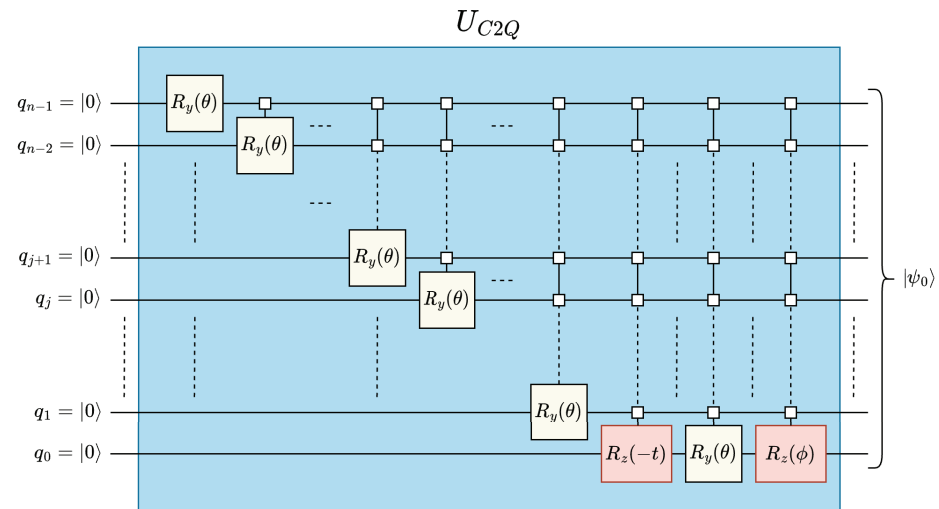


Figure 1. Quantum circuit for classical-to-quantum (C2Q) arbitrary state synthesis [17].

2.3. Quantum Shift Operation

A fundamental operation for quantum convolution is the *quantum shift operation*, denoted in this work as U_{shift}^k , which shifts the basis states of the state vector by k positions when applied to an m -qubit state $|\psi\rangle$, see (3). The quantum shift operation is critical for performing the cyclic rotations needed to prepare strided windows when performing convolution. It is also common for the operation to be described as a *quantum incrementer* when $k > 0$, see Figure 2a, and a *quantum decrementer* when $k < 0$ [16,18], see Figure 2b:

$$U_{\text{shift}}^k |\psi\rangle = \sum_{i=0}^{2^m-1} c_i |j\rangle, \text{ where } j = (i - k) \bmod 2^m \tag{3}$$

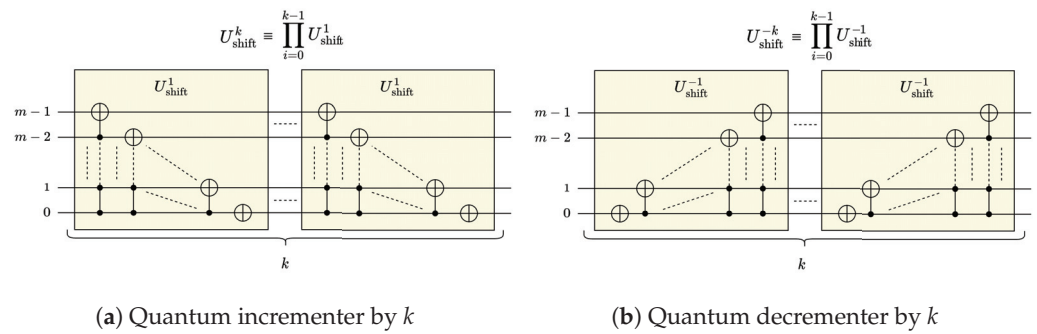


Figure 2. Quantum shift operation using quantum incrementers/decrementers.

3. Materials and Methods

In general, a convolution operation can be performed using a sequence of shift and multiply-and-accumulate operations. In our proposed methods, we implement the generalized convolution operations as follows:

1. *Shift*: Auxiliary filter qubits and controlled quantum decrementers are used to create shifted (unity-strided) replicas of input data.
2. *Multiply-and-accumulate*: Arbitrary state synthesis and classical-to-quantum (C2Q) encoding are applied to create generic multidimensional filters.
3. *Data rearrangement*: Quantum permutation operations are applied to restructure the fragmented data into one contiguous output datum.

In Section 3.1, we present our quantum convolution technique in detail for one-dimensional data. In the following sections, we illustrate optimizations to improve circuit depth and generalize our method for multidimensional data. For evaluating our proposed methods, we used real-world, high-resolution, black-and-white (B/W) and RGB images, ranging in a resolution from (8×8) pixels to (512×512) pixels and $(8 \times 8 \times 3)$ pixels to $(512 \times 512 \times 3)$ pixels, respectively. We also performed experiments on 1-D real-world audio data and 3-D real-world hyperspectral data to demonstrate our method’s applicability to data and filters of any dimensionality. Further details about our experimental setup and dataset can be found in Section 4.

3.1. Quantum Convolution for One-Dimensional Data

The proposed structure of quantum convolution for one-dimensional (1-D) data is shown in Figure 3. The following sections show the details of the five steps of the convolution operation procedure to transform the initial encoded data $|\psi_0\rangle$ to the final state $|\psi_5\rangle$, see Figure 3.

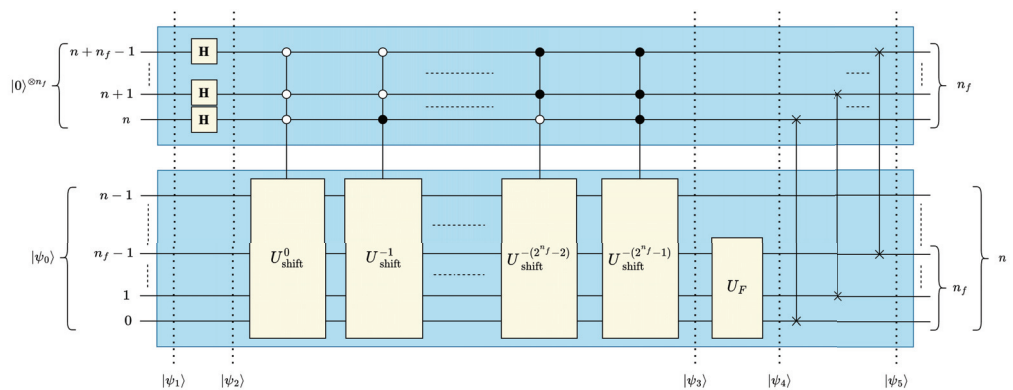


Figure 3. The 1-D quantum convolution circuit.

3.1.1. Shift Operation

To perform convolution with unity stride with a filter of size N_f terms, N_f replicas of the input data must be made, strided for $0 \leq k < N_f$. To store these replicas, we add $n_f = \lceil \log_2 N_f \rceil$ auxiliary qubits, which we denote as “filter qubits”, to the most significant positions of the initial quantum state $|\psi_0\rangle$, see (4) and Figure 3:

$$|\psi_1\rangle = |0\rangle^{\otimes n_f} \otimes |\psi_0\rangle = \begin{bmatrix} |\psi_0\rangle \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \updownarrow 2^n \\ \\ \\ \downarrow 2^{n+n_f} \end{matrix} \quad (4)$$

Placing the filter qubits in superposition using Hadamard (H) gates creates 2^{n_f} identical replicas of the initial data $|\psi_0\rangle$, as shown in (5):

$$|\psi_2\rangle = (H^{\otimes n_f} \otimes I^{\otimes n})|\psi_1\rangle = \frac{1}{\sqrt{2^{n_f}}} \begin{bmatrix} |\psi_0\rangle \\ \vdots \\ |\psi_0\rangle \end{bmatrix} \begin{matrix} \updownarrow 2^n \\ \vdots \\ \updownarrow 2^n \end{matrix} \begin{matrix} \updownarrow 2^n \\ \\ \\ \downarrow 2^{n+n_f} \end{matrix} \quad (5)$$

Finally, multiplexed quantum shift operations can be used to generate the strided replicas, see (6):

$$|\psi_3\rangle = U_{\text{mux}}|\psi_2\rangle = \frac{1}{\sqrt{2^{n_f}}} \begin{bmatrix} U_{\text{shift}}^0 |\psi_0\rangle \\ \vdots \\ U_{\text{shift}}^{-(2^{n_f}-1)} |\psi_0\rangle \end{bmatrix} \begin{matrix} \updownarrow 2^n \\ \vdots \\ \updownarrow 2^n \end{matrix} \begin{matrix} \updownarrow 2^n \\ \\ \\ \downarrow 2^{n+n_f} \end{matrix}, \text{ where} \quad (6)$$

$$U_{\text{mux}} = \begin{bmatrix} U_{\text{shift}}^0 & & \\ & \ddots & \\ & & U_{\text{shift}}^{-(2^{n_f}-1)} \end{bmatrix}$$

3.1.2. Multiply-and-Accumulate Operation

For the traditional convolution operation, applying a filter $F \in \mathbb{R}^{N_f}$ to an array of data $W \in \mathbb{R}^{N_f}$ produces a scalar output $x \in \mathbb{R}$, which can be expressed as $x = F^T W$. In the quantum domain, we can represent an array of data as the partial state $|\phi\rangle$ and the normalized filter as $|F\rangle$. Accordingly, the output state can be expressed as shown in (7):

$$|\psi_{\text{out}}\rangle = \sum_{i=0}^{2^n-1} \langle F|\phi_i\rangle \cdot |i\rangle, \text{ where} \quad (7)$$

$$|\phi_i\rangle = \sum_{j=0}^{2^{n_f}-1} \langle k'|\psi_3\rangle \cdot |k'\rangle, \text{ and } k' = (2^{n_f} \cdot i) + j$$

To calculate $|\psi_4\rangle$ from $|\psi_3\rangle$, it is necessary to embed $\langle F|$ into a unitary operation U_F as shown in (8). Since $\langle F|$ is a normalized row vector, we can define U_F as a matrix such that its first row is $\langle F|$ and the remaining rows are arbitrarily determined to preserve the unitariness of U_F such that $U_F^\dagger U_F = U_F U_F^\dagger = I^{\otimes n_f}$. From (2), we can realize U_F using an inverse C2Q operation, see (8):

$$\begin{aligned}
 |\psi_4\rangle &= (I^{\otimes n} \otimes U_F)|\psi_3\rangle = \begin{bmatrix} U_F|\phi_0\rangle \\ \vdots \\ U_F|\phi_{2^n-1}\rangle \end{bmatrix} \begin{matrix} \uparrow 2^{n_f} \\ \vdots \\ \downarrow 2^{n_f} \end{matrix} \begin{matrix} \uparrow \\ \vdots \\ \downarrow \end{matrix} 2^{n+n_f}, \text{ where} \\
 U_F &= \begin{bmatrix} \langle F| \\ \langle \times| \\ \vdots \\ \langle \times| \end{bmatrix} \begin{matrix} \uparrow \\ \vdots \\ \downarrow \end{matrix} 2^{n_f} = U_{C2Q}^\dagger
 \end{aligned} \tag{8}$$

3.1.3. Data Rearrangement

As of $|\psi_4\rangle$, the desired values of $|\psi_{out}\rangle$ are fragmented among undesired/extraneous values, which we denote using the symbol “ \times ”. We apply SWAP permutations to rearrange and coalesce our desired values to be contiguous in the final statevector $|\psi_5\rangle$, see (9) and Figure 3:

$$\begin{aligned}
 |\psi_5\rangle &= U_{SWAP}^{1-D}|\psi_4\rangle \\
 &= \begin{bmatrix} \langle F|\phi_0\rangle \\ \vdots \\ \langle F|\phi_{2^n-1}\rangle \\ \times \\ \vdots \\ \times \end{bmatrix} \begin{matrix} \uparrow \\ \vdots \\ \downarrow \end{matrix} \begin{matrix} \uparrow \\ \vdots \\ \downarrow \end{matrix} \begin{matrix} 2^n \\ \vdots \\ 2^{n+n_f} \end{matrix} \\
 &= \begin{bmatrix} |\psi_{out}\rangle \\ \times \\ \vdots \\ \times \end{bmatrix} \begin{matrix} \uparrow 2^n \\ \vdots \\ \downarrow \end{matrix} 2^{n+n_f}, \text{ where} \\
 U_{SWAP}^{1-D} &= \prod_{j=n_f-1}^0 (I^{\otimes(n_f-1-j)} \otimes SWAP(j, j+n) \otimes I^{\otimes j})
 \end{aligned} \tag{9}$$

3.1.4. Circuit Depth Analysis of 1-D Quantum Convolution

When considering the circuit depth complexity of the proposed 1-D quantum convolution technique, it is evident from Figure 3 that the operations described by (5) and (9) are performed using parallel Hadamard and SWAP operations, respectively, and thus are of constant depth complexity, i.e., $\mathcal{O}(1)$. In contrast, the U_{mux} and U_F operations incur the largest circuit depth, as they are both serial operations that scale with the data size and/or filter size, see Figure 3.

For the implementation of U_{mux} in Figure 3, there are a total of 2^{n_f} controlled quantum shift operations, where the i -th shift operation is a quantum decremter $U_{shift}^{-i} = (U_{shift}^{-1})^i$. Since all the U_{shift}^{-1} operations are performed in series, the circuit depth of U_{mux} depends on the total number of unity quantum decremters, $N_{U_{shift}^{-1}}$, see (10):

$$N_{U_{shift}^{-1}}(n_f) = \sum_{i=0}^{2^{n_f}-1} i = \frac{2^{n_f}(2^{n_f}-1)}{2} = \frac{4^{n_f}}{2} - 2^{n_f-1} \tag{10}$$

As shown in Figure 2b, each quantum decremter U_{shift}^{-1} acting on m qubits can be realized using m multi-controlled CNOT (MCX) gates, where the i -th MCX gate is controlled by i qubits and $0 \leq i < m$. Accordingly, for each quantum decremter U_{shift}^{-1} that is controlled by c qubits, its i -th MCX gate is controlled by a total of $i + c$ qubits.

Therefore, the depth of the quantum decremter circuits can be expressed in terms of the MCX gate count as shown in (11):

$$\mathcal{D}_{U_{\text{shift}}^{-1}}(m, c) = \sum_{i=c}^{m+c-1} \mathcal{D}_{\text{MCX}}(i) \quad (11)$$

The depth of an MCX gate with a total of m qubits can be expressed with a linear function in terms of fundamental single-qubit rotation gates and CNOT gates [19] as shown in (12), where α represents the first-order coefficient and β represents the constant bias term. Thus, the depth complexity of U_{shift}^{-1} can be expressed as shown in (13):

$$\mathcal{D}_{\text{MCX}}(m) = \alpha m + \beta : \alpha, \beta \in \mathbb{R} \quad (12)$$

$$\begin{aligned} \mathcal{D}_{U_{\text{shift}}^{-1}}(m, c) &= \sum_{i=0}^{m-1} \alpha(i+c) + \beta \\ &= \frac{\alpha}{2}m^2 + \left(\alpha\left(c - \frac{1}{2}\right) + \beta\right)m \\ &= \mathcal{O}(m^2) \end{aligned} \quad (13)$$

To derive the circuit depth complexity of U_{mux} , we leverage the definitions of $N_{U_{\text{shift}}^{-1}}(n_f)$ and $\mathcal{D}_{U_{\text{shift}}^{-1}}(m, c)$ as shown in (14), where $m = n$ and $c = n_f$:

$$\begin{aligned} \mathcal{D}_{U_{\text{mux}}}(n, n_f) &= N_{U_{\text{shift}}^{-1}}(n_f) \cdot \mathcal{D}_{U_{\text{shift}}^{-1}}(n, n_f) \\ &= \left(\frac{4^{n_f}}{2} - 2^{n_f-1}\right) \cdot \left(\frac{\alpha}{2}n^2 + \left(\alpha\left(n_f - \frac{1}{2}\right) + \beta\right)n\right) \\ &= \left(4^{n_f-1} - 2^{n_f-2}\right) \cdot \left(\alpha n^2 + 2\alpha n_f n - (\alpha - 2\beta)n\right) \\ &= \mathcal{O}\left(4^{n_f}n^2 + 4^{n_f}n_f n\right) \end{aligned} \quad (14)$$

As discussed in Section 3.1.2, we implement the filter operation U_F by leveraging the C2Q arbitrary synthesis operation [17]. Although C2Q incurs a circuit depth of exponential complexity in terms of fundamental quantum gates, as shown in (15), U_F is only applied to n_f qubits, a small subset of the total number of qubits, which somewhat mitigates the circuit depth. Furthermore, in most practical scenarios, the dimensions of the filter are typically much smaller than the dimensions of the input data, i.e., $n_f \ll n$. As a result, U_F should not incur overly large circuit depth relative to other circuit components, e.g., U_{mux} . Altogether, the overall circuit depth complexity of the 1-D quantum convolution operation can be expressed according to (16):

$$\mathcal{D}_{U_F}(n_f) = \mathcal{O}(2^{n_f}) \quad (15)$$

$$\mathcal{D}_{1\text{-D conv}}(n, n_f) = \mathcal{O}\left(4^{n_f}n^2 + 4^{n_f}n_f n + 2^{n_f}\right), \text{ where } n \gg n_f \quad (16)$$

3.2. Depth-Optimized 1-D Quantum Convolution

In Figure 4, we present an optimized implementation of U_{mux} that greatly reduces the circuit depth.

In Section 3.1, we implemented U_{mux} with 2^{n_f} controlled quantum decremterers U_{shift}^{-k} , where $0 \leq k < 2^{n_f}$. We can represent each k in binary notation, as shown in (17), to express U_{shift}^{-k} as a sequence of controlled shift operations by powers of 2. As shown in (18), we can

denote such operations with the notation $U_{\text{shift}}^{-b_j 2^j}(n)$, where $0 \leq j < n_f$, and (n) reflects that the shift operation is applied to an n -qubit state.

$$k = \left(b_{n_f-1} b_{n_f-2} \cdots b_j \cdots b_1 b_0 \right)_2 = \sum_{j=0}^{n_f-1} b_j 2^j : b_j \in \{0, 1\} \tag{17}$$

$$U_{\text{shift}}^{-k}(n) = U_{\text{shift}}^{-\sum_{j=0}^{n_f-1} b_j 2^j}(n) = \prod_{j=0}^{n_f-1} U_{\text{shift}}^{-b_j 2^j}(n) \tag{18}$$

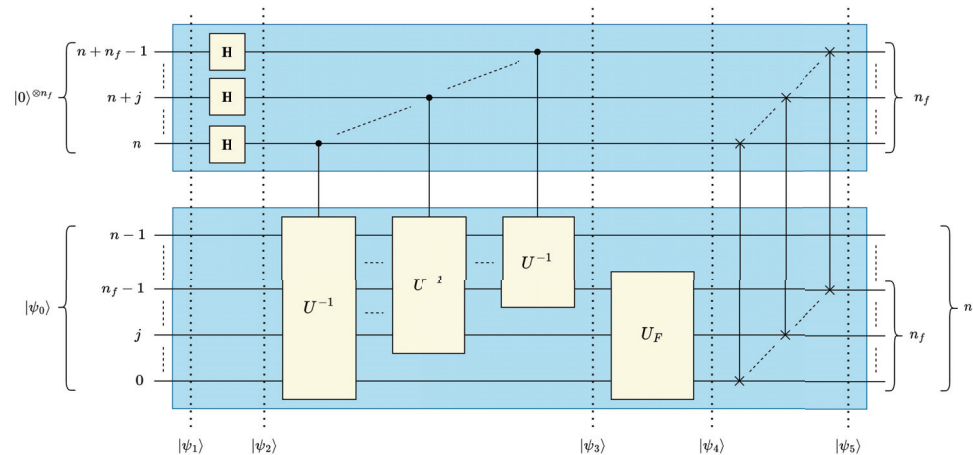


Figure 4. Depth-optimized 1-D quantum convolution circuit.

The binary decomposition of the uniformly controlled U_{shift}^{-k} operations is conducive to several circuit depth optimizations. As shown in (19), the value of b_j is dependent only on the state of the j -th filter qubit q_{n+j} . In other words, each $U_{\text{shift}}^{-2^j}(n)$ can only be controlled by one qubit q_{n+j} , independently from the other control qubits. Accordingly, it is possible to coalesce the multiplexed $U_{\text{shift}}^{-2^j}(n)$ operations across the k control indices. The resultant implementation of U_{mux} , therefore, becomes a sequence of 2^{n_f} single-controlled $U_{\text{shift}}^{-2^j}(n)$ operations, where $0 \leq j < n_f$, which comparatively has a smaller circuit depth by a factor of 2^{n_f} . Furthermore, each $U_{\text{shift}}^{-2^j}(n)$ operation can be implemented using a single $U_{\text{shift}}^{-1}(n-j)$ operation in lieu of sequential $U_{\text{shift}}^{-1}(n)$ operations, see (20) and Figure 4, further reducing the depth by a factor of 2^j per operation:

$$b_j = \begin{cases} 1, & |q_{n+j}\rangle = |1\rangle \\ 0, & \text{otherwise} \end{cases}, \quad \forall k \in [0, 2^{n_f}] \tag{19}$$

$$U_{\text{shift}}^{-2^j}(n) \equiv \prod_{2^j} U_{\text{shift}}^{-1}(n) = U_{\text{shift}}^{-1}(n-j) \otimes I^{\otimes j} \tag{20}$$

Circuit Depth Analysis of Optimized 1-D Quantum Convolution

With the aforementioned optimizations, the depth of the updated U_{mux} operation can be expressed in terms of $\mathcal{D}_{U_{\text{shift}}^{-1}}(m, c)$ as described by (21), where $m = n - j$ and $c = 1$ for all $0 \leq j < n_f$. In comparison with the depth complexity of the unoptimized U_{mux} , see (14), the dominant term remains quadratic, i.e., n^2 , in terms of the data qubits n . However, its coefficient is improved exponentially, from 4^{n_f} to n_f , see (14) and (21). Note that a cubic term n_f^3 in terms of the number of filter qubits is introduced in the optimized U_{mux} implementation, see (21). However, when considering the total depth for the optimized 1-D quantum convolution circuit $\mathcal{D}_{1\text{-D conv}}^{\text{opt}}$, the n_f^3 term becomes negligible because of U_F ,

whose complexity $\mathcal{O}(2^{n_f})$ is exponential in terms of the number of filter qubits, see (15), (21), and (22):

$$\begin{aligned} \mathcal{D}_{U_{\text{mux}}}^{\text{opt}}(n, n_f) &= \sum_{j=0}^{n_f-1} \mathcal{D}_{U_{\text{shift}}^{-1}}(n-j, 1) = \sum_{j=0}^{n_f-1} \left[\frac{\alpha}{2}(n-j)^2 + \left(\frac{\alpha}{2} + \beta\right)(n-j) \right] \\ &= \frac{\alpha}{2}n_f n^2 - \frac{\alpha}{2}n_f^2 n + (\alpha + \beta)n_f n + \frac{\alpha}{6}n_f^3 - \frac{\alpha + \beta}{2}n_f^2 + \frac{2\alpha + 3\beta}{6}n_f \\ &= \mathcal{O}\left(n_f n^2 - n_f^2 n + n_f^3\right), \text{ where } n \gg n_f \end{aligned} \tag{21}$$

$$\begin{aligned} \mathcal{D}_{1\text{-D conv}}^{\text{opt}}(n, n_f) &= \mathcal{O}\left(n_f n^2 - n_f^2 n + n_f^3 + 2^{n_f}\right) \\ &= \mathcal{O}\left(n_f n^2 - n_f^2 n + 2^{n_f}\right), \text{ where } n \gg n_f \end{aligned} \tag{22}$$

3.3. Generalized Quantum Convolution for Multidimensional Data and Filters

In this section, we present the quantum circuit of our proposed quantum convolution technique generalized for multidimensional data and filters. Although quantum statevectors are one-dimensional, it is possible to map multidimensional data to a 1-D vector in either *row-* or *column-*major order. In this work, we represent multidimensional input data and convolutional filters in a quantum circuit in *column-major order*. In other words, for d -dimensional data of size $(N_0 \times \dots \times N_i \times \dots \times N_{d-1})$ data values, the positional information of the i -th dimension is encoded in the $\sum_{j=0}^{i-1} n_j$ to $(\sum_{j=0}^i n_j) - 1$ qubits, where $n_i = \lceil \log_2 N_i \rceil$. Using this representation, the optimized 1-D quantum convolution circuit shown in Figure 4 can be generalized for d dimensions by “stacking” d 1-D circuits as shown in Figure 5.

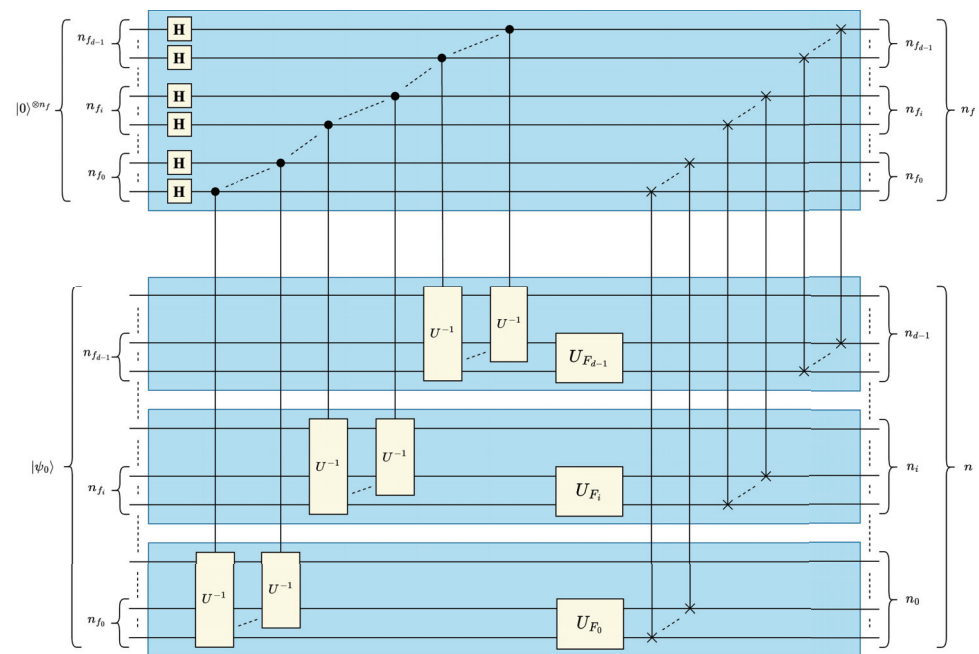


Figure 5. Multidimensional quantum convolution circuit with distributed/stacked 1-D filters.

The “stacked” quantum circuit in Figure 5 is based on the assumption that the overall (lumped) d -dimensional filter operator U_F is separable and decomposable into d one-dimensional filters U_{F_i} for $0 \leq i < d$. However, it would be more practically useful to generalize our multidimensional quantum convolution technique independently from the separability/decomposability of U_F . For this purpose, the identity in (23), which could be easily derived from either Figure 3 or Figure 4 for 1-D convolution, can be leveraged and

generalized for multidimensional convolution circuits, see (24). The identity in (24) allows us to reverse the order of multiply-and-accumulate and data rearrangement steps and, therefore, generate one generic lumped U_F that acts on the contiguous $n_f = \sum_{i=0}^{d-1} n_{f_i}$ filter qubits, where n_{f_i} is the number of qubits representing the filter dimension i for $0 \leq i < d$, see Figure 6. U_F can be derived based on the given arbitrary multidimensional filter F using the method discussed in Section 3.1.2 when F is represented as a normalized 1-D vector $|F\rangle$ in a column major ordering:

$$U_{\text{SWAP}}^{1-D} \cdot (I^{\otimes n} \otimes U_F) = (U_F \otimes I^{\otimes n}) \cdot U_{\text{SWAP}}^{1-D} \tag{23}$$

$$U_{\text{SWAP}}^{d-D} \cdot \left(I^{\otimes n_f} \otimes \left(\bigotimes_{i=d-1}^0 [I^{\otimes(n_i-n_{f_i})} \otimes U_{F_i}] \right) \right) = \left(\left(\bigotimes_{i=d-1}^0 [U_{F_i}] \right) \otimes I^{\otimes n} \right) \cdot U_{\text{SWAP}}^{d-D} \\ = (U_F \otimes I^{\otimes n}) \cdot U_{\text{SWAP}}^{d-D}, \text{ where}$$

$$U_F = \bigotimes_{i=d-1}^0 [U_{F_i}] \equiv U_{F_{d-1}} \otimes U_{F_{d-2}} \otimes \dots \otimes U_{F_1} \otimes U_{F_0}, \tag{24}$$

$$U_{\text{SWAP}}^{d-D} = \prod_{i=d-1}^0 \prod_{j=n_{f_i}-1}^0 \left(I^{\otimes(n_f-1-j-q_{f_i})} \otimes \text{SWAP}(j+q_i, j+n+q_{f_i}) \otimes I^{\otimes(j+q_i)} \right),$$

$$q_{f_i} = \sum_{k=0}^{i-1} n_{f_k}, \quad q_i = \sum_{k=0}^{i-1} n_k, \quad n_f = \sum_{k=0}^{d-1} n_{f_k}, \quad \text{and} \quad n = \sum_{k=0}^{d-1} n_k$$

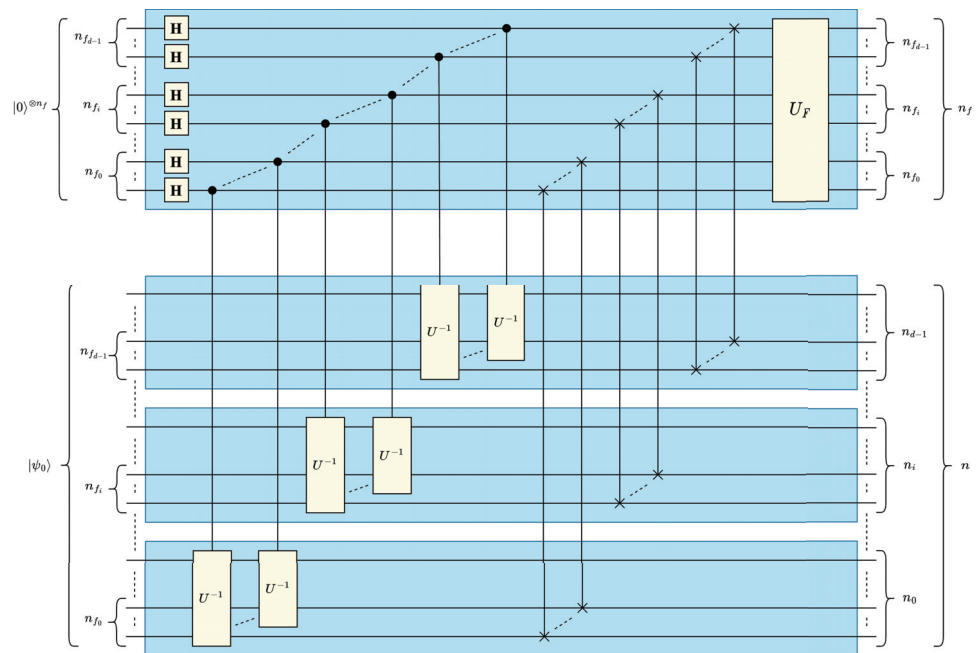


Figure 6. Generalized multidimensional quantum convolution circuit.

Circuit Depth Analysis of Generalized Multidimensional Quantum Convolution

As a result of the “stacked” structure, the data of all d dimensions could be concurrently processed in parallel. Consequently, the circuit depth of the multidimensional quantum circuit becomes dependent on the largest data dimension $N_{\text{max}} = 2^{n_{\text{max}}}$, where $n_{\text{max}} = \max_{i=0}^{d-1} (n_i)$, in lieu of the total data size N . The circuit component of the optimized 1-D circuit with the greatest depth contribution U_{mux} is performed in parallel on each dimension in the generic d -D circuit. Specifically, U_{mux} scales with the number of qubits used to represent the largest data dimension $n_{\text{max}} = \lceil \log_2 N_{\text{max}} \rceil$. Note that the

parallelization across dimensions applies to the Hadamard and SWAP operations from (5) and (9), see Figure 6, and therefore these operations are of constant depth complexity, i.e., $\mathcal{O}(1)$. The depth complexity of the multidimensional U_F operation is determined by the total number of elements in the filter N_F , and therefore the C2Q-based implementation of U_F does not benefit from multidimensional stacking. Accordingly, the circuit depth of the generalized multidimensional quantum convolution operation could be derived from (22) and expressed in (25), where $n_{f_{\max}} = \max_{i=0}^{d-1}(n_{f_i})$ is the number of qubits representing the maximum filter dimension $N_{f_{\max}} = 2^{n_{f_{\max}}}$. It is worth mentioning that the generic multidimensional formula in (25) reduces to the 1-D formula in (22) when $n = n_{\max}$ and $n_f = n_{f_{\max}}$:

$$\mathcal{D}_{d\text{-D conv}}^{\text{opt}}(n, n_f) = \mathcal{O}\left(n_{f_{\max}} n_{\max}^2 - n_{f_{\max}}^2 n_{\max} + 2^{n_f}\right), \text{ where} \tag{25}$$

$$n_{\max} = \max_{i=0}^{d-1}(n_i), n_{f_{\max}} = \max_{i=0}^{d-1}(n_{f_i}), \text{ and } n_{\max} \gg n_{f_{\max}}$$

4. Experimental Setup and Results

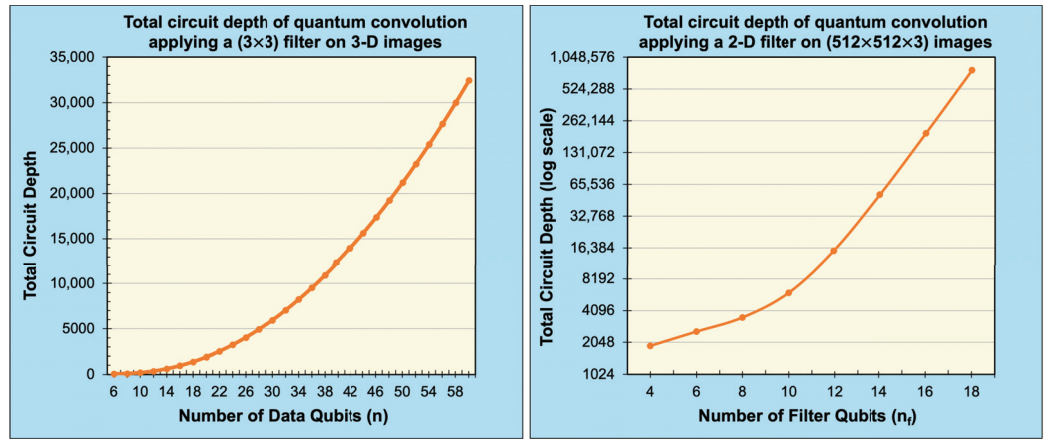
We experimentally demonstrate our proposed technique for generalized, multidimensional quantum convolution with unity stride on real-world, high-resolution, multi-dimensional image data, see Figure 7. By leveraging the Qiskit SDK (v0.39.4) from IBM Quantum [9], we simulate our quantum circuits in the following formats: (1) classically (to present the ideal/theoretical expectation), (2) noise-free (using statevector simulation), and (3) noisy (using 1,000,000 “shots” or circuit samples). Moreover, we present a quantitative comparison of the obtained results using fidelity [20] as a similarity metric between compared results ρ and σ , see (26). Experiments were performed on a 16-core AMD EPYC 7302P CPU with frequencies up to 3.3 GHz, 128 MB of L3 cache, and access to 256 GB of DDR4 RAM. In our analysis, we evaluated the correctness of the proposed techniques by comparing classical results with noise-free results. We also evaluated the scalability of the proposed techniques for higher-resolution images by comparing the classical results with both the noise-free and noisy results. Finally, we plotted the circuit depth of our techniques with respect to the data size and filter size as shown in Figure 8.



(a) Black-and-white (B/W) image.

(b) Color (RGB) Image

Figure 7. Real-world, high-resolution, multidimensional images used in experimental trials.



(a) Depth with respect to data size.

(b) Depth with respect to filter size.

Figure 8. Circuit depth of quantum convolution with respect to data and filter qubits.

$$\text{Fidelity}(\rho, \sigma) = \text{tr} \sqrt{\sqrt{\rho} \cdot \sigma \cdot \sqrt{\rho}} \tag{26}$$

In our experiments, we evaluated our techniques using well-known (3×3) and (5×5) filters, i.e., Averaging F_{avg} , Sobel edge-detection F_{S_x}/F_{S_y} , Gaussian blur F_{blur} , and Laplacian of Gaussian blur (Laplacian) $F_{\mathcal{L}}$, see (27)–(30). We applied zero padding to maintain the size of the filter dimensions at a power of two for quantum implementation. In addition, we used wrapping to resolve the boundary conditions, and we restricted the magnitude of the output between $[0, 255]$ to mitigate quantization errors in the classical domain:

$$F_{\text{avg}}^{3 \times 3} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad F_{\text{avg}}^{5 \times 5} = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{27}$$

$$F_{\text{blur}}^{3 \times 3} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad F_{\text{blur}}^{5 \times 5} = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \tag{28}$$

$$F_{S_x} = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad F_{S_y} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{29}$$

$$F_{\mathcal{L}}^{3 \times 3} = \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad F_{\mathcal{L}}^{5 \times 5} = \frac{1}{20} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -24 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{30}$$

We applied 2-D convolution filters to black-and-white (B/W) and RGB images, see Figure 7, ranging in resolution from (8×8) pixels to (512×512) pixels and $(8 \times 8 \times 3)$ pixels to $(512 \times 512 \times 3)$ pixels, respectively. The number of filter qubits can be obtained by the size of filter dimensions, i.e., $n_f = \lceil \log_2 3 \rceil + \lceil \log_2 3 \rceil = 4$ qubits for (3×3) filters and $n_f = \lceil \log_2 5 \rceil + \lceil \log_2 5 \rceil = 6$ qubits for (5×5) filters. Therefore, our simulated quantum circuits ranged in size $(n + n_f)$ from 10 qubits to 26 qubits. Figures 9 and 10 present the reconstructed output images for classical, noise-free, and noisy experiments using (128×128) and $(128 \times 128 \times 3)$ -pixel input images, respectively.

To demonstrate our method's applicability to data and filters of any dimensionality, we also performed experiments applying the 1-D and 3-D averaging filter to 1-D real-world audio data and 3-D real-world hyperspectral data, respectively. The audio files were sourced from the publicly available sound quality assessment material published by the European Broadcasting Union and modified to be single channel with data sizes ranging from 2^8 values to 2^{20} values when sampled at 44.1 kHz [21]. Figures 11 and 12 present the reconstructed output images and fidelity, respectively, from applying (3) and (5) averaging filters. The hyperspectral images were sourced from the Kennedy Space Center (KSC) dataset [22] and resized to range from $(8 \times 8 \times 8)$ pixels to $(128 \times 128 \times 128)$ pixels. Figures 13 and 14 present the reconstructed output images and fidelity, respectively, from applying $(3 \times 3 \times 3)$ and $(5 \times 5 \times 5)$ averaging filters.

Comparison of the noise-free quantum results against the ideal classical results demonstrates a 100% fidelity across all trials. Thus, in a noise-free (statevector) environment, our proposed quantum convolution technique correctly performs an identical operation to classical convolution given the same input parameters and boundary conditions.

When considering the behavior of noisy (sampled) environments, Figures 12, 14 and 15 plot the fidelity of the noisy quantum results against the ideal classical results. We observe a monotonic decrease in fidelity as the data size (image resolution) increases, consistent with previously reported behavior [23]. Such behavior derives from how the number of shots required to properly characterize a quantum state increases with the corresponding number of qubits in order to reduce the effects of statistical noise. Notably, the fidelity varies dramatically depending on the filter category and size, where the largest discrepancy occurs between the (5×5) Averaging and (5×5) Laplacian filters for a data size of 65,536 values. Specifically, for a B/W image of (256×256) pixels, the Averaging filter had a fidelity of 94.84%, while the Laplacian filter had a fidelity of 8.82%—a difference of 86.02%, see Figure 15a. In general, we observed that the average and blur filters perform better than the edge detection methods (Sobel/Laplacian). Since the output data are reconstructed from only a portion of the final state $|\psi_5\rangle$, it is likely that sparse filters, represented in Figures 9 and 10 as being mostly black, are significantly less likely to be recorded during sampled measurement, resulting in reduced fidelity. For practical applications, dimension reduction techniques, such as pooling, can be used to mitigate information loss [23].


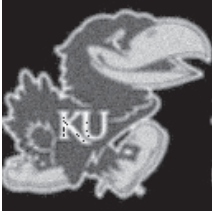
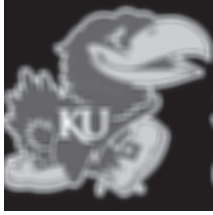
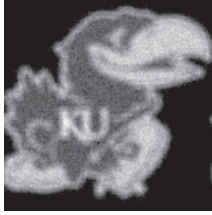

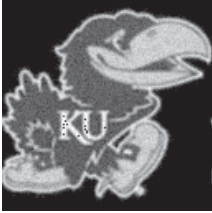

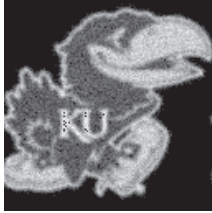








Filter	(3 × 3) filter Classical/ Noise-Free	(3 × 3) filter Noisy (10 ⁶ shots)	(5 × 5) filter Classical/ Noise-Free	(5 × 5) filter Noisy (10 ⁶ shots)
Average				
Gaussian				
Sobel-X			Not Applicable	Not Applicable
Sobel-Y			Not Applicable	Not Applicable
Laplacian				

Figure 9. The 2-D convolution filters applied to a (128 × 128) B/W image.




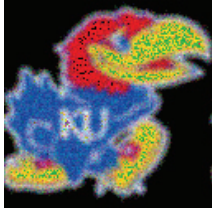



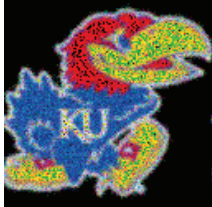

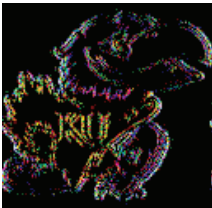
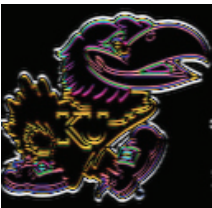
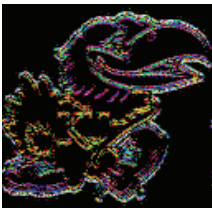

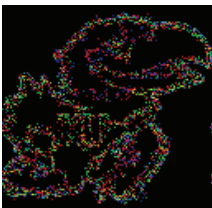

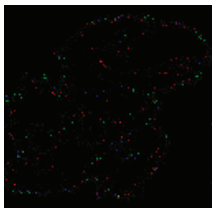
Filter	(3 × 3) filter Classical/ Noise-Free	(3 × 3) filter Noisy (10 ⁶ shots)	(5 × 5) filter Classical/ Noise-Free	(5 × 5) filter Noisy (10 ⁶ shots)
Average				
Gaussian				
Sobel-X			Not Applicable	Not Applicable
Sobel-Y			Not Applicable	Not Applicable
Laplacian				

Figure 10. The 2-D convolution filters applied to a (128 × 128 × 3) RGB image.

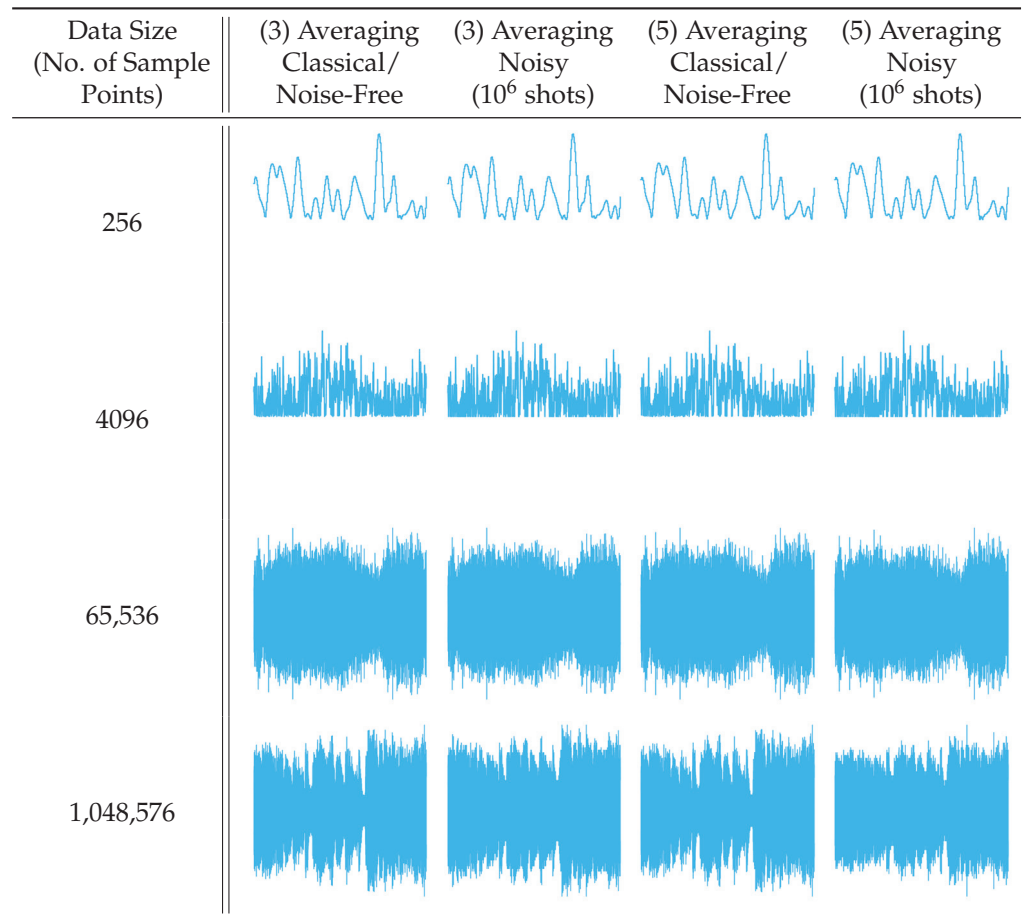


Figure 11. The 1-D convolution (averaging) filters applied to 1-D audio samples.

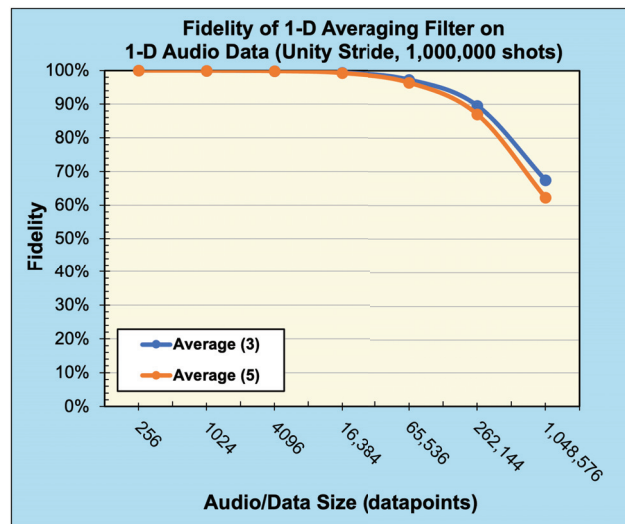


Figure 12. Fidelity of 1-D convolution (averaging) filters with unity stride on 1-D audio data (sampled with 1,000,000 shots).

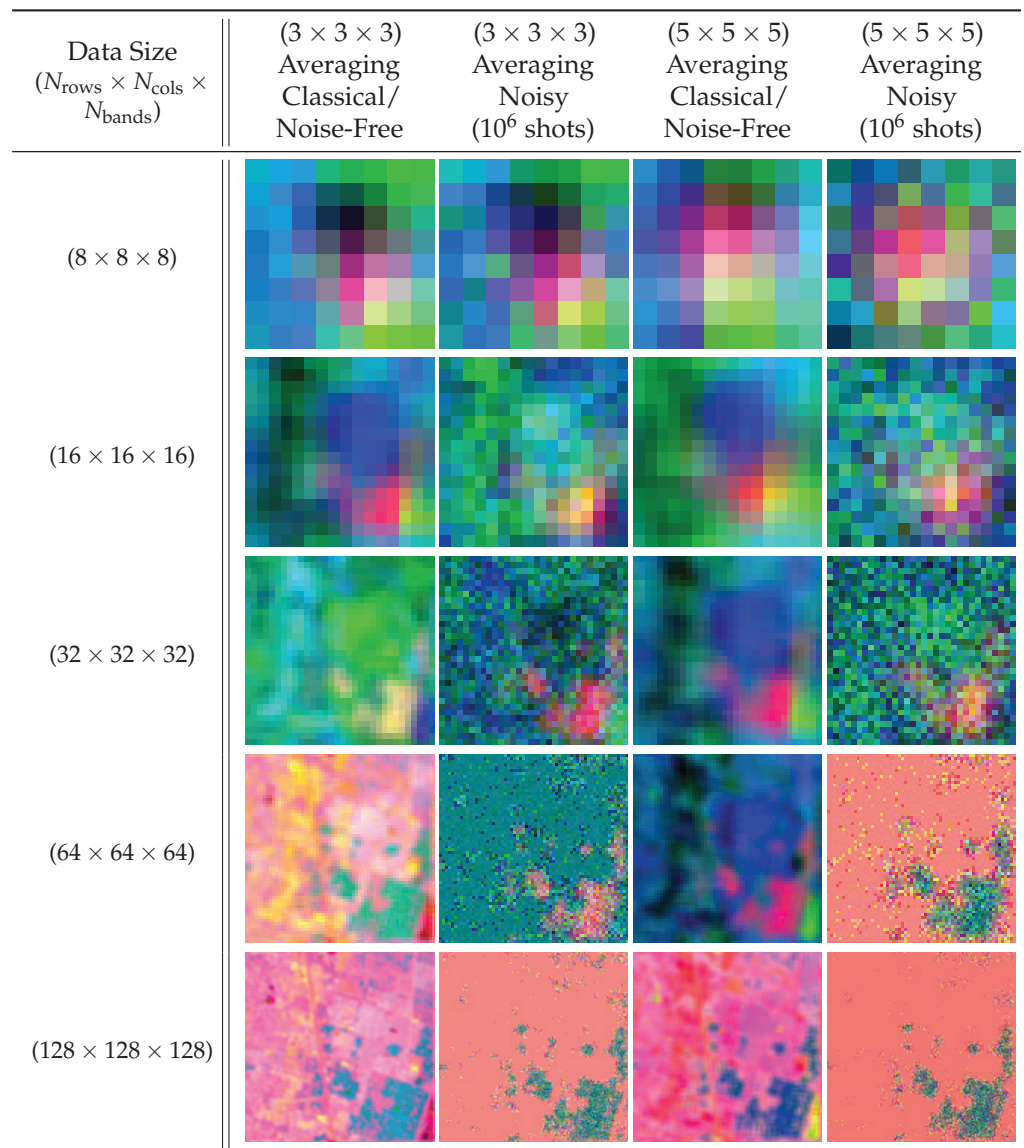


Figure 13. The 3-D convolution (averaging) filters applied to 3-D hyperspectral images (bands 0, 1, and 2).

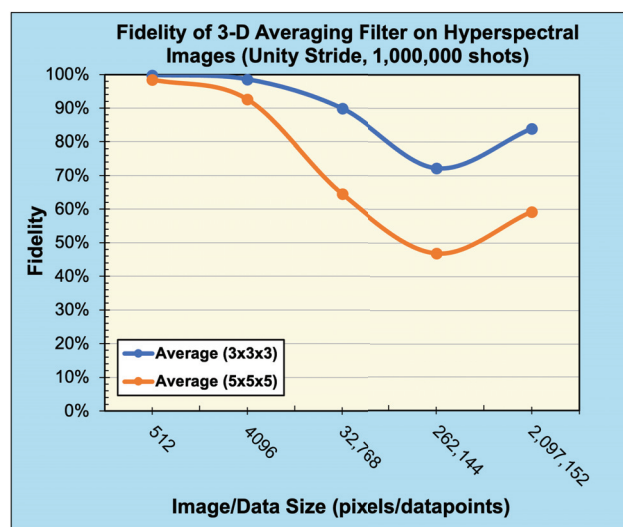
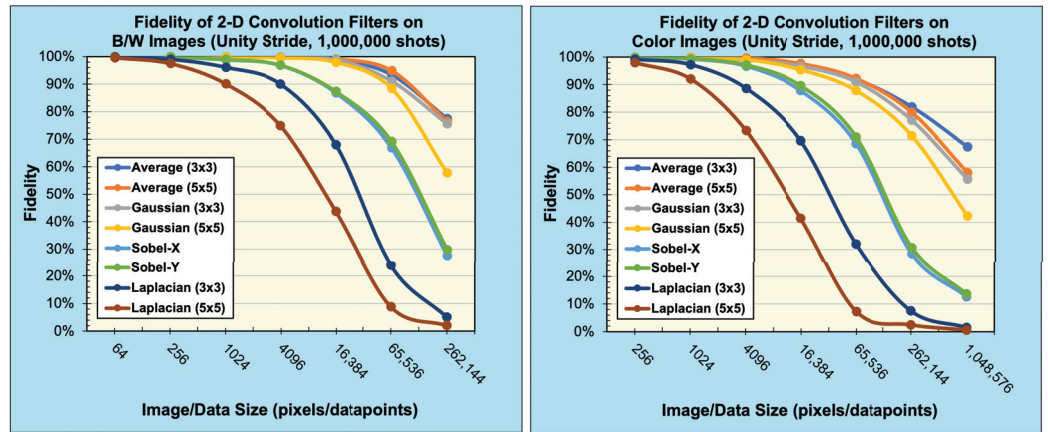


Figure 14. Fidelity of 3-D convolution (averaging) filters with unity stride on 3-D hyperspectral data (sampled with 1,000,000 shots).



(a) Black-and-white (B/W) images

(b) Color (RGB) images

Figure 15. Fidelity of 2-D convolution with unity stride (sampled with 1,000,000 shots).

5. Discussion

In the following section, we compare our proposed method of quantum convolution to the related work discussed in Section 2.1 in terms of filter generalization and circuit depth.

5.1. Arbitrary Multidimensional Filtering

Our generalizable and parameterized technique of quantum convolution with unity stride offers distinct workflow advantages over existing *fixed-filter* quantum convolution techniques in variational applications, such as quantum machine learning. Our technique does not require extensive development for each new filter design. For instance, current quantum convolutional filters are primarily two dimensional, only focusing on image processing. However, the development of even similar filters targeting audio and video processing, for example, would require extensive development and redesign. Our method offers a systematic and straightforward approach for generating practical quantum circuits given fundamental input variables.

5.2. Circuit Depth

Figure 8a,b show the circuit depth for our proposed technique of generalized quantum convolution with respect to the total number of data qubits $n = \sum_{i=0}^{d-1} (n_i)$ and the total number of filter qubits $n_f = \sum_{i=0}^{d-1} (n_{f_i})$, respectively. The results were gathered using the `QuantumCircuit.depth()` method built into Qiskit for a `QuantumCircuit` transpiled to fundamental single-qubit and CNOT quantum gates. Figure 8a illustrates quadratic circuit depth complexity with respect to the data qubits n for a fixed filter size $N_F = 2^{n_f}$, aligning with our theoretical expectation in (25). Note that $n = \sum_{i=0}^{d-1} n_i$ and $n_{\max} = \max_{i=0}^{d-1} (n_i)$ for d -dimensional data. Similarly, Figure 8b (plotted on a log-scale) illustrates exponential circuit depth complexity with respect to n_f for a fixed data size $N = 2^n$, which also aligns with our theoretical expectation in (25).

The time complexity comparison of our proposed quantum convolution technique against related work is shown in Table 1. Compared to classical direct implementations on CPUs, our proposed technique for generalized quantum convolution demonstrates an exponential improvement with respect to data size $N = 2^n$, i.e., $\mathcal{O}(n^2)$ vs. $\mathcal{O}(N^2)$, see (25) and Table 1c. As discussed in Section 2.1, the fastest classical GEMM implementation of convolution on GPUs (excluding data I/O overhead) [12,13] has a complexity of $\mathcal{O}(N_F N)$, see Table 1c. Even when including quantum data encoding, which is equivalent to data I/O overhead, our method remains to demonstrate a linear improvement with respect to data size N by a factor of the filter size N_F , see (31), over classical GEMM GPUs:

$$\begin{aligned} \mathcal{D}_{\text{proposed}}(n) &= \mathcal{D}_{\text{C2Q}}(n) + \mathcal{D}_{d\text{-D conv}}^{\text{opt}}(n) \\ &= \mathcal{O}\left(2^n + n_{\text{max}}^2\right) = \mathcal{O}(2^n) = \mathcal{O}(N), \text{ for fixed } n_f \end{aligned} \tag{31}$$

Compared to *fixed-filter* quantum convolution techniques [4–8], our proposed arbitrary filter quantum convolution technique (for unity stride) demonstrates improved circuit depth complexity with respect to data size when factoring in the circuit depth contribution from data encoding. For a fixed filter, i.e., n_f is constant, the depth of the proposed method scales quadratically with the largest data dimension n_{max} , see (25) and (31). As described in Section 2.1, *fixed-filter* quantum convolution techniques similarly show quadratic depth scaling with respect to the number of qubits n , see Table 1b. For data encoding, the *fixed-filter* techniques use either the FRQI [14] or NEQR [15] algorithms, which have circuit depth complexities of $\mathcal{O}(4^n)$ or $\mathcal{O}(qn2^n)$, respectively. In contrast, our proposed technique uses C2Q data encoding [17], which has a depth complexity of $\mathcal{O}(2^n)$ —a quadratic and linear improvement over FRQI and NEQR, respectively, see Table 1a.

Table 1. Comparison of depth/time complexity of proposed generalized quantum convolution technique against related work.

a Depth complexity of quantum data encoding (I/O) techniques			
FRQI [14]	NEQR [15]	C2Q [17]	
$\mathcal{O}_{\text{I/O}}(4^n)$	$\mathcal{O}_{\text{I/O}}(qn2^n)$	$\mathcal{O}_{\text{I/O}}(2^n)$	
b Depth complexity of quantum convolution algorithms for a fixed filter			
Proposed	Related Work [4–8]		
$\mathcal{O}_{\text{alg}}(n_{\text{max}}^2)$	$\mathcal{O}_{\text{alg}}(n^2)$		
c Complexity of proposed technique compared to classical convolution			
Proposed	Direct (CPU) [10]	FFT (CPU/GPU) [10,11]	GEMM (GPU) [12,13]
$\mathcal{O}_{\text{alg}} \left(\begin{matrix} n_{f_{\text{max}}} n_{\text{max}}^2 - n_{f_{\text{max}}}^2 n_{\text{max}} \\ + 2^{n_f} \end{matrix} \right)$	$\mathcal{O}_{\text{alg}}(N^2) \equiv \mathcal{O}_{\text{alg}}(4^n)$	$\mathcal{O}_{\text{alg}}(N \log N) \equiv \mathcal{O}_{\text{alg}}(n2^n)$	$\mathcal{O}_{\text{alg}}(N_F N) \equiv \mathcal{O}_{\text{alg}}(2^{(n+n_f)})$
$\mathcal{O}_{\text{alg} + \text{I/O}} \left(\begin{matrix} n_{f_{\text{max}}} n_{\text{max}}^2 - n_{f_{\text{max}}}^2 n_{\text{max}} \\ + 2^n + 2^{n_f} \end{matrix} \right)$			

6. Conclusions

In this work, we proposed and evaluated a method for generalizing the convolution operation with arbitrary filtering and unity stride for multidimensional data in the quantum domain. We presented the corresponding circuits and their performance analyses along with experimental results that were collected using the IBM Qiskit development environment. In our experimental work, we validated the correctness of our method by comparing classical results to noise-free quantum results. We also demonstrated the practicality of our method for various convolution filters by evaluating the noisy quantum results. Furthermore, we presented experimentally verified analyses that highlight our technique’s advantages in terms of time complexity and/or circuit depth complexity compared to existing classical and quantum methods, respectively. Future work will focus on adapting our proposed technique for arbitrary strides. In addition, we will investigate multidimensional quantum machine learning as a potential application of our proposed technique.

Author Contributions: Conceptualization: M.J., V.J., D.K. and E.E.-A.; Methodology: M.J., V.J., D.K. and E.E.-A.; Software: M.J., D.L., D.K. and E.E.-A.; Validation: M.J., V.J., D.L., D.K. and E.E.-A.; Formal analysis: M.J., V.J., D.L., D.K. and E.E.-A.; Investigation: M.J., A.N., V.J., D.L., D.K., M.C., I.I., M.M.R. and E.E.-A.; Resources: M.J., A.N., V.J., D.L., D.K., M.C., I.I., M.M.R. and E.E.-A.; Data curation: M.J., A.N., V.J., D.L., D.K., M.C., I.I., M.M.R. and E.E.-A.; Writing—original draft preparation: M.J., A.N., V.J., D.L., D.K., M.C., I.I. and E.E.-A.; Writing—review and editing: M.J., A.N., V.J., D.L., D.K., M.C., I.I., M.M.R. and E.E.-A.; Visualization: M.J., A.N., V.J., D.L., D.K., M.C.,

I.I., M.M.R. and E.E.-A.; Supervision: E.E.-A.; Project administration: E.E.-A.; Funding acquisition: E.E.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The audio samples used in this work are publicly available from the European Broadcasting Union at <https://tech.ebu.ch/publications/sqamcd> (accessed on 19 October 2023) as file 64.f1ac [21]. The hyperspectral data used in this work are publicly available from the Grupo de Inteligencia Computacional (GIC) at [https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Kennedy_Space_Center_\(KSC\)](https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Kennedy_Space_Center_(KSC)) (accessed on 19 October 2023) under the heading Kennedy Space Center (KSC) [22].

Acknowledgments: This research used resources of the Oak Ridge Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

1-D	One Dimensional
C2Q	Classical to Quantum
FFT	Fast Fourier Transform
FRQI	Flexible Representation of Quantum Images
GEMM	General Matrix Multiplication
NEQR	Novel Enhanced Quantum Representation
QHED	Quantum Hadamard Edge Detection
QWT	Quantum Wavelet Transform

References

- Rhodes, W. Acousto-optic signal processing: Convolution and correlation. *Proc. IEEE* **1981**, *69*, 65–79. [CrossRef]
- Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160. [CrossRef]
- LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional networks and applications in vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; pp. 253–256. [CrossRef]
- Fan, P.; Zhou, R.G.; Hu, W.; Jing, N. Quantum image edge extraction based on classical Sobel operator for NEQR. *Quantum Inf. Process.* **2018**, *18*, 24. [CrossRef]
- Ma, Y.; Ma, H.; Chu, P. Demonstration of Quantum Image Edge Extraction Enhancement Through Improved Sobel Operator. *IEEE Access* **2020**, *8*, 210277–210285. [CrossRef]
- Zhang, Y.; Lu, K.; Gao, Y. QSobel: A novel quantum image edge extraction algorithm. *Sci. China Inf. Sci.* **2015**, *58*, 1–13. [CrossRef]
- Zhou, R.G.; Yu, H.; Cheng, Y.; Li, F.X. Quantum image edge extraction based on improved Prewitt operator. *Quantum Inf. Process.* **2019**, *18*, 261. [CrossRef]
- Li, P. Quantum implementation of the classical Canny edge detector. *Multimed. Tools Appl.* **2022**, *81*, 11665–11694. [CrossRef]
- IBM Quantum. Qiskit: An Open-source Framework for Quantum Computing. 2021. Available online: <https://zenodo.org/records/7416349> (accessed on 19 October 2023).
- Burrus, C.S.; Parks, T.W. *DFT/FFT and Convolution Algorithms: Theory and Implementation*, 1st ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1991.
- Podlozhnyuk, V. FFT-based 2D convolution. *NVIDIA* **2007**, *32*. Available online: https://developer.download.nvidia.com/compute/cuda/1.1-Beta/x86_64_website/projects/convolutionFFT2D/doc/convolutionFFT2D.pdf (accessed on 19 October 2023).
- NVIDIA. Convolution Algorithms. 2023. Available online: <https://docs.nvidia.com/deeplearning/performance/dl-performance-convolutional/index.html#conv-algo> (accessed on 19 October 2023).
- NVIDIA. CUTLASS Convolution. 2023. Available online: https://github.com/NVIDIA/cutlass/blob/main/media/docs/implicit_gemm_convolution.md (accessed on 19 October 2023).
- Le, P.Q.; Dong, F.; Hirota, K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process.* **2011**, *10*, 63–84. [CrossRef]
- Zhang, Y.; Lu, K.; Gao, Y.; Wang, M. NEQR: A novel enhanced quantum representation of digital images. *Quantum Inf. Process.* **2013**, *12*, 2833–2860. [CrossRef]
- Yao, X.W.; Wang, H.; Liao, Z.; Chen, M.C.; Pan, J.; Li, J.; Zhang, K.; Lin, X.; Wang, Z.; Luo, Z.; et al. Quantum Image Processing and Its Application to Edge Detection: Theory and Experiment. *Phys. Rev. X* **2017**, *7*, 031041. [CrossRef]

17. El-Araby, E.; Mahmud, N.; Jeng, M.J.; MacGillivray, A.; Chaudhary, M.; Nobel, M.A.I.; Islam, S.I.U.; Levy, D.; Kneidel, D.; Watson, M.R.; et al. Towards Complete and Scalable Emulation of Quantum Algorithms on High-Performance Reconfigurable Computers. *IEEE Trans. Comput.* **2023**, *72*, 2350–2364. [CrossRef]
18. Li, X.; Yang, G.; Torres, C.M.; Zheng, D.; Wang, K.L. A Class of Efficient Quantum Incrementer Gates for Quantum Circuit Synthesis. *Int. J. Mod. Phys. B* **2014**, *28*, 1350191. [CrossRef]
19. Balauca, S.; Arusoae, A. Efficient Constructions for Simulating Multi Controlled Quantum Gates. In Proceedings of the Computational Science—ICCS 2022, London, UK, 21–23 June 2022; Groen, D., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloat, P.M.A., Eds.; Springer: Cham, Switzerland, 2022; pp. 179–194.
20. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*; Cambridge University Press: Cambridge, UK, 2010; p. 409. [CrossRef]
21. Geneva, S. Sound Quality Assessment Material: Recordings for Subjective Tests. 1988. Available online: <https://tech.ebu.ch/publications/sqamcd> (accessed on 19 October 2023).
22. Graña, M.; Veganzons, M.A.; Ayerdi, B. Hyperspectral Remote Sensing Scenes. Available online: [https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Kennedy_Space_Center_\(KSC\)](https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Kennedy_Space_Center_(KSC)) (accessed on 19 October 2023).
23. Jeng, M.; Islam, S.I.U.; Levy, D.; Riachi, A.; Chaudhary, M.; Nobel, M.A.I.; Kneidel, D.; Jha, V.; Bauer, J.; Maurya, A.; et al. Improving quantum-to-classical data decoding using optimized quantum wavelet transform. *J. Supercomput.* **2023**, *79*, 20532–20561. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Principal Component Analysis and t-Distributed Stochastic Neighbor Embedding Analysis in the Study of Quantum Approximate Optimization Algorithm Entangled and Non-Entangled Mixing Operators

Brian García Sarmina ^{1,*}, Guo-Hua Sun ¹ and Shi-Hai Dong ^{1,2,*}

¹ Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City 07738, Mexico; gsun@cic.ipn.mx

² Research Center for Quantum Physics, Huzhou University, Huzhou 310003, China

* Correspondence: brian.garsar.6@gmail.com (B.G.S.); dongsh2@yahoo.com (S.-H.D.); Tel.: +52-5521775512 (B.G.S.)

Abstract: In this paper, we employ PCA and t-SNE analyses to gain deeper insights into the behavior of entangled and non-entangled mixing operators within the Quantum Approximate Optimization Algorithm (QAOA) at various depths. We utilize a dataset containing optimized parameters generated for max-cut problems with cyclic and complete configurations. This dataset encompasses the resulting RZ, RX, and RY parameters for QAOA models at different depths (1L, 2L, and 3L) with or without an entanglement stage within the mixing operator. Our findings reveal distinct behaviors when processing the different parameters with PCA and t-SNE. Specifically, most of the entangled QAOA models demonstrate an enhanced capacity to preserve information in the mapping, along with a greater level of correlated information detectable by PCA and t-SNE. Analyzing the overall mapping results, a clear differentiation emerges between entangled and non-entangled models. This distinction is quantified numerically through explained variance in PCA and Kullback–Leibler divergence (post-optimization) in t-SNE. These disparities are also visually evident in the mapping data produced by both methods, with certain entangled QAOA models displaying clustering effects in both visualization techniques.

Keywords: QAOA; mixing operator; entangled operator; non-entangled operator

Citation: Sarmina, B.G.; Sun, G.-H.; Dong, S.-H. Principal Component Analysis and t-Distributed Stochastic Neighbor Embedding Analysis in the Study of Quantum Approximate Optimization Algorithm Entangled and Non-Entangled Mixing Operators. *Entropy* **2023**, *25*, 1499. <https://doi.org/10.3390/e25111499>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 13 September 2023

Revised: 16 October 2023

Accepted: 17 October 2023

Published: 30 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The analysis of how Variational Quantum Algorithms (VQAs) work has been extensively studied in recent years [1–5]. In particular, the Quantum Approximate Optimization Algorithm (QAOA) has garnered much attention in the research community [6,7]. One of the most exciting aspects of analyzing these algorithms is understanding how they explore the problem space, what relationships exist between the rotation gates used in the circuit [6,8], and how these gates impact the overall performance of the algorithm [2,5,9]. Previous studies have attempted to shed light on these aspects [10,11] and also study the relationship between the Hamiltonian structure of a certain problem and the landscape (search space) generated [12,13]. However, as quantum hardware becomes more complex and quantum circuits become deeper, the need for a better understanding of these algorithms becomes even more critical [14,15].

In general, the analysis of QAOAs can be influenced by factors such as circuit depth and optimization strategies, which may impact the accuracy of problem results. Studies in this area often focus on the problem landscape representation, which is a critical aspect to consider when studying QAOAs from a problem resolution perspective [2,4,16,17]. However, it is important to note that there are other crucial aspects to consider when studying QAOAs, such as the extraction of information about the underlying models and

their potential limitations or strengths, prior to their application to specific problems; e.g., max-cut, max-cut, Ising model, etcetera [18–21].

In this paper, our primary objective is to contribute to the existing body of knowledge by conducting an in-depth analysis of entangled and non-entangled mixing operators within the context of QAOAs. We leverage Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) techniques to scrutinize the parameters generated within the RZ , RX , and RY gates across various QAOA models at different depths ($1L$, $2L$, and $3L$).

Our overarching goal is to discern unique patterns of behavior that can offer valuable insights into how QAOA gate parameters behave under different scenarios; specifically, whether there are discernible differences in parameter distribution when an entanglement stage is present or absent in the mixing operator. Furthermore, we aim to present a clear and insightful visualization of these behaviors, both numerically and graphically, to enhance our understanding of the underlying dynamics of the mixing operator.

Some notable examples of visualization studies in VQAs are the works by Moussa et al. (2022) [22] and Rudolph et al. (2021) [13]. In Moussa et al. (2022), t-SNE was utilized as a preprocessing step to reveal clustering tendencies in QUBO problems and assist in determining the parameters for the QAOA. They also explored the use of supervised techniques when clusters did not adequately represent the corresponding points, leading to a more effective prediction of QAOA parameters. Rudolph et al. (2021) conducted an analysis of various visualization techniques, including PCA, applied to different VQAs. Their focus was on generating a mapping of the optimization landscape for specific problems, as well as studying aspects of parameter concentration in QAOAs, and other phenomena.

In contrast, our study focuses specifically on the representation, visualization, and information extraction from the resulting QAOA parameters, which are the RZ , RX , and RY gate values, acquired from the max-cut problem dataset. We evaluate the effectiveness of PCA and t-SNE strategies in providing comprehensive insights into the models. This evaluation encompasses both graphical representations and internal metrics derived from both methods.

2. Motivation and Methodology

The motivation for this work is rooted in the concepts discussed by D. Koch et al. (2020) [6], particularly in Lesson 10 referred to the QAOA. In their work, the authors raise the notion that the conventional mixing operator, which includes RX and RY gates with individual gate rotations, may prove inadequate in exploring all the feasible states within the associated Hilbert space of the system. This limitation becomes particularly prominent in scenarios involving high-dimensional spaces, where individual rotations often lead to separable states, thereby impacting the overall effectiveness of the QAOA algorithm.

In response to the challenge of limited state reachability, discussed by D. Koch et al. (2020) [6], the proposed solution involves the incorporation of an entanglement stage within the mixing operator. This modification enables access to entangled states, which constitute the majority of possible states in composite quantum systems of two or more qubits. The specific structure or properties of the entanglement stage were not detailed in their work. While there have been various studies exploring the structure of entanglement stages in quantum circuits, this remains a topic with many unanswered questions and areas for further research [23–25].

2.1. Motivation for Studying Entangled and Non-Entangled Mixing Operator

Considering the previous ideas, several questions that motivate this work arise: Is there an observable difference in the distribution of parameter values generated by the mixing operator when we introduce an entanglement stage? Can visualization techniques like PCA or t-SNE reveal visual and/or numerical disparities in the distribution of parameter values between mixing operators with and without an entanglement stage?

In our quest to address these questions, we conducted a comprehensive review of the state of the art. Our objective was to explore whether any existing research had analyzed the distribution of parameter values across a set of solutions for a specific problem, with a particular emphasis on the visualization aspect. However, our findings indicated a gap in the literature. Most research in this domain tends to analyze each individual experiment separately and often employs techniques such as heatmaps to examine the landscape of the solution space.

While we encountered some works that touched upon related aspects, such as Moussa et al. (2022) [22] and Rudolph et al. (2021) [13], these studies primarily focused on different facets of VQAs and optimization. Their specific emphases did not align with the questions that we were trying to solve in our research. Consequently, our study represents a unique contribution to the field, shedding light on the distribution of parameter values in the context of entangled and non-entangled mixing operators within the QAOA.

In the following sections, we explain the results obtained to answer these questions, where indeed the results show differences in the mapping data (both numerically and visually) using PCA and t-SNE as our visualization techniques when we encounter an entanglement stage in the mixing operator.

2.2. Methodology of QAOA Dataset Usage

In this paper, we utilized a dataset containing the optimized parameters acquired for the phase (RZ gates) and mixing operators (RX and RY gates). These parameters were obtained by applying the Quantum Approximate Optimization Algorithm (QAOA) in conjunction with the Stochastic Hill Climbing with Random Restarts (SHC-RR) optimization method to a series of max-cut problems. SHC-RR does not exhibit a specific tendency in its exploration of the search space, making it a more unbiased strategy suitable for data generation.

The dataset of optimized max-cut problems was created for an upcoming study, which also involves an analysis of QAOAs. In this work, we do not delve into the methodology of solving the max-cut problems using QAOAs or assess the quality of the optimized solutions provided in the dataset. Our sole focus is on utilizing the generated parameters associated with the optimized solutions for a set of max-cut problems. For comprehensive results, including the optimized parameters, please refer to reference [26].

Additionally, it is important to note that our analysis in this study does not consider the expected energy value or evaluated cost obtained from the solution of a particular experiment in a max-cut problem.

The dataset comprises the parameters obtained with QAOAs using SHC-RR to solve max-cut problems with cyclic and complete configurations, involving different numbers of nodes: 4 nodes ($4n$), 10 nodes ($10n$), and 15 nodes ($15n$). Each problem was simulated 100 times, where different QAOA depths were tested, including $1L$, $2L$, and $3L$. Additionally, each QAOA model was evaluated both with and without an entanglement stage (for every depth) in the mixing operators.

The dataset generated for a particular model and problem contains 100 simulations, each representing different solution scenarios due to the inherent variability of SHC-RR in parameter distributions. Therefore, we do not consider the quality of the solution, as each simulation could yield a better or worse-optimized result. The primary focus of this research is to extract the properties of the parameters without factoring in the quality of the solution. This approach is valid because the only difference between the compared models is the presence or absence of the entanglement stage in the mixing operator. In this comparison, the depth ($1L$, $2L$, or $3L$), problem type (configuration and number of nodes), and optimization method (SHC-RR) are held constant across all compared models.

PCA and t-SNE are employed in two distinct ways to analyze the QAOA models. In individual analysis, each method is applied to a specific QAOA model depth either with or without an entanglement stage in the mixing operator, for a particular max-cut problem

dataset. This yields numeric and graphical results for each method, allowing us to compare the individual outcomes.

In paired analysis, we directly compare the entangled and non-entangled QAOA models (with the same depth) using a single PCA or t-SNE model. In other words, one PCA or t-SNE model is applied to the values from both datasets of QAOA parameters, representing the entangled and non-entangled versions of the QAOA model at a specific depth.

These two approaches applied to the QAOA datasets provide us with a comprehensive understanding of how the entangled and non-entangled mixing operators behave under different conditions.

3. Problems to Analyze

To provide a comprehensive foundation for understanding the differences between entangled and non-entangled mixing operators in the context of the Hamiltonian and circuit structure, this section has been developed.

We commence by offering a general overview of the max-cut problem, which serves as the basis for the dataset's content. Subsequently, we explore the fundamental structure of the phase operator, which plays a crucial role in comprehending the two distinct types of max-cut problems found in the dataset. Finally, we arrive at the crux of our investigation: the representation of both entangled and non-entangled mixing operators. This representation is of paramount importance, as it sets the stage for subsequent visualization and numerical analyses. It also plays a pivotal role in differentiating the entangled and non-entangled QAOA models, alongside the depth factor.

3.1. Max-Cut Problem

A maximum cut (max-cut) problem is a combinatorial optimization problem that is often used in the field of both quantum and classical computer science, and operations research. In this problem, one is given an undirected graph where each edge is associated with a weight, and the goal is to partition the vertices of the graph into two sets, called A and B, in such a way that the sum of the weights of the edges that cross between these two sets is maximized.

In Figure 1, we illustrate the process of solving a max-cut problem using a simple example. The graph in question consists of four nodes labeled from 1 to 4. To find the optimal solution, we divide the graph into two distinct groups: Group A, comprising nodes 1 and 3, and Group B, comprising nodes 2 and 4. It is important to note that, in this example and in the parameters dataset, the weighted connections between nodes are considered to be unitary, each with a value of 1. Given this assumption, the solution comes from determining the number of connections that cross between the two groups, which transition from the green-colored nodes (Group A) to the red-colored nodes (Group B). In this particular example, we observe four such connections.

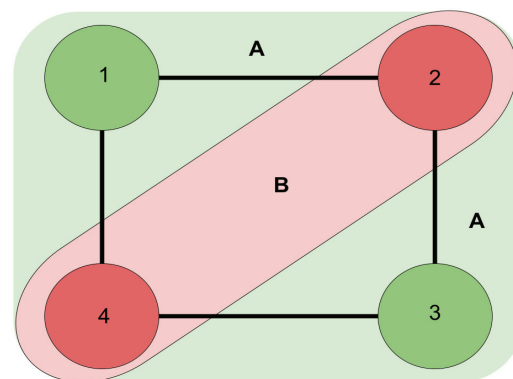


Figure 1. Max-cut example.

When applying QAOA to address max-cut problems, the solution entails a set of values assigned to the gates of the phase and mixing operators, effectively determining the state with the highest probability of representing the optimal solution. Referring back to our previous example, this solution could manifest as either $|0101\rangle$ or $|1010\rangle$ state, each with an associated probability. In this representation, qubits in state 0 correspond to one group, while those in state 1 correspond to the other group. The aim is to maximize the probability of obtaining the correct solution state, ideally approaching a probability close to 100%, contingent on the quality of the QAOA model, which encompasses factors such as gate parameter precision and the number of parameters employed.

3.2. Hamiltonian and Circuit Description

Transitioning to the discussion of the Hamiltonian and circuit description, we begin by elucidating the distinctions between the non-entangled and entangled mixing operators, as depicted in the figure below.

Within the dataset, each of the QAOA models' depth has a different number of parameters, namely, the 1L model has one set of RZ , RZ , and RY parameter values, the 2L model has two sets of RZ , RX , and RY parameter values, and the 3L model has three sets of RZ , RX , and RY parameter values. Each QAOA model is evaluated both with and without the inclusion of an entanglement stage in the mixing operator (see Figure 2).

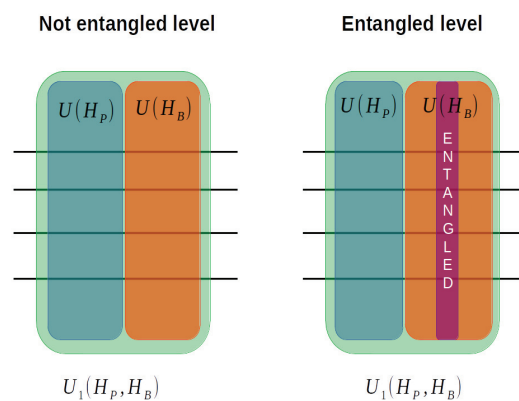


Figure 2. Non-entangled and entangled mixing operators.

In Figure 3, we provide a visual representation of the various levels of QAOA quantum circuit depths. Each of the 1L, 2L, and 3L depths corresponds to the number of pairs of phase and mixing operator applications in the QAOA model dataset, 1L being a pair of operators, 2L two pairs of operators, and 3L three pairs of operators. Detailed explanations of these operators will be presented in the following sections.

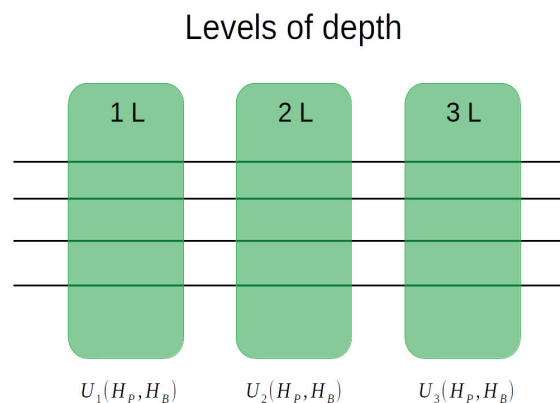


Figure 3. Levels of depth, with one pair of phase and mixing operator 1L, two pairs 2L, and three pairs 3L.

As a side note, each of the 10 and 15-node problem datasets contain experiments using $1L$, $2L$, and $3L$ depths. However, for the four-node problem datasets, we have only depths of $1L$ and $2L$ due to their relatively simpler nature.

The Hamiltonian configurations for dataset problems can be viewed in the following phase operators for the cyclic and complete configuration.

$$U(H_{cyc}, \gamma) = e^{-i\gamma H_{cyc}} = \prod_{\langle j,k \rangle} e^{-i\gamma Z_j Z_k} \tag{1}$$

In the max-cut problem with the cyclic configuration, the representation of the phase operator can be expressed using Equation (1), where $\langle j, k \rangle$ denotes the notation for neighboring nodes. For instance, in the case of the four-node problem, this equation can be interpreted as connections between nodes 1 to 2, nodes 2 to 3, nodes 3 to 4, and nodes 4 to 1, where the final connection completes the cycle.

$$U(H_{com}, \gamma) = e^{-i\gamma H_{com}} = \prod_{\{j,k|j \neq k\}} e^{-i\gamma Z_j Z_k} \tag{2}$$

For the complete configuration, the representation for the phase operator follows Equation (2). In this case, there is a connection between every pair of nodes in the graph, excluding self-connections $\{j, k \mid j \neq k\}$. Also, connections between nodes are not repeated, meaning that a connection from node j to node k is considered the same as a connection from node k to node j . This is due to the absence of directionality in the max-cut problem.

$$U(H_B, \beta_1, \beta_2) = e^{i\beta_1 \beta_2 H_B} = \prod_j e^{i\beta_1 X_j} e^{i\beta_2 Y_j}, \tag{3}$$

The mixing operator without entanglement for both max-cut configurations is represented by Equation (3). This equation includes RX and RY rotations in the mixing operator with the associated parameters for each gate.

The entangled mixing operator includes an additional term compared to the non-entangled case. This term is generated by applying $CNOT$ gates between each qubit (node) in the system, similar to the complete configuration. The equation representing the $CNOT$ action is as follows:

$$e^{i\frac{\pi}{4}(I-Z) \otimes (I-X)} = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{4}$$

to represent the entangled mixing operator in our notation, we use the expression $e^{iI_j X_k}$ to indicate that qubit j is controlling qubit k as the target. By adding the term $e^{iI_j X_k}$ to the previous mixing operator without entanglement, we obtain the following expression:

$$\begin{aligned} U(H_B, \beta_1, \beta_2) &= e^{i\beta_1 \beta_2 H_B} \\ &= \prod_j e^{i\beta_1 X_j} \prod_{\{j,k|j \neq k\}} e^{iI_j X_k} \prod_j e^{i\beta_2 Y_j}, \end{aligned} \tag{5}$$

which represents the mixing operator with an entanglement stage between the RX and RY rotations. The entanglement stage generates interactions between each pair of qubits in the system, ensuring that there are no repeated interactions and no self-interactions.

The quantum circuit representation of the two types of tested mixing operators can be seen in Figures 4 and 5. Figure 4 corresponds to the non-entangled mixing operator, while Figure 5 corresponds to the entangled mixing operator.

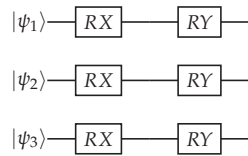


Figure 4. Individual rotations in mixing operators.

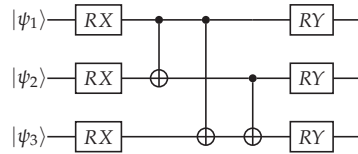


Figure 5. Entangled rotations in mixing operators.

Both methods (PCA and t-SNE) are applied to each model, where the models are 1L considering one phase operator with one associated parameter γ and one mixing operator with two associated parameters β_1 and β_2 ; a 2L model that has two phase operators with γ_1 and γ_2 and two mixing operators (connected between each phase operator, one per operator) with β_{1-1} , β_{1-2} , and β_{2-1} and β_{2-2} ; and a 3L model that was tested for some problems (not all) with γ_1 , γ_2 , and γ_3 with the corresponding six parameters for the three mixing operators (two for each operator, as in the previous cases).

4. PCA and t-SNE Description

In this section, we explain how the PCA and t-SNE approach is used to analyze the properties of the QAOA dataset.

4.1. PCA

Principal component analysis is a method (with statistical or geometric interpretation) that aims to reduce the dimensionality of a dataset, retaining as much of the original information as possible [27,28].

PCA operates by employing specific structures known as principal components, which are designed to capture the maximum variance in the directions they are projected. Utilizing these principal components, we have the ability to transform the original data into a new coordinate system. Typically, the first two principal components are used for this transformation, enabling the data to be visualized in a more interpretable and meaningful manner [27,28].

In our study, our PCA process commenced with the calculation of the covariance between a pair of the resulting parameter values of RZ, RX, and RY gates obtained from a particular experiment within the dataset. It is important to note that there could be multiple sets of values depending on the model under consideration.

$$cov(i, j) = \frac{1}{n - 1} \sum_{k=1}^n x_{k,i} x_{k,j} \tag{6}$$

We use Equation (6) to calculate the covariance between parameters of the RZ and RX, RZ and RY, and RX and RY gates, where n takes a value of 100, which is the number of experiments performed for each QAOA model in a particular max-cut problem.

$$\Sigma = \sum_i \sum_j cov(i, j) \tag{7}$$

Then, we calculate the eigenvalues and eigenvectors of the resulting covariance matrix (Equation (7)) for the parameters of a certain model in the dataset.

Let $i = 1, 2, \dots, n$, where $n < N$ is the number of principal components and N is the dimension of the original data. Let $\theta = [\theta_1, \theta_2, \dots, \theta_N]$ be the original data vector, P_i^T be

the transpose of the eigenvector matrix (obtained using Σ), and $\phi = [\phi_1, \phi_2, \dots, \phi_n]$ be the resulting transformed vector in the principal component space. The projection using PCA can be described as follows:

$$\phi_i = P_i^T \theta_i, \tag{8}$$

where Equation (8) represents the projection (via the dot product between P_i^T and θ_i) onto a principal component i represented by ϕ_i . This process is then repeated to obtain all the principal components, where each new component is orthogonal to the previous ones.

$$\text{Var}(\phi_1) \geq \text{Var}(\phi_2) \geq \dots \geq \text{Var}(\phi_i) > 0 \tag{9}$$

The variance of the principal components follows the relationship described by Equation (9). This equation indicates that the variance of the principal components generally decreases as the index increases. Consequently, higher index values correspond to a reduced amount of variance information contained in the data.

4.2. t-SNE

The t-distributed stochastic neighbor embedding method is similar to PCA in the sense that it is used as an algorithm for data visualization and dimensionality reduction. The main difference (besides the methodology) with t-SNE is its capability to represent non-linear relationships in the data and its ability to preserve the high-dimensional structure of the original data into a lower-dimensional space [29,30].

The t-SNE algorithm creates pairwise similarities using a Gaussian kernel by measuring the distance between the points in the original dataset. Then, the algorithm generates probability distributions over pairs of points, where the probability of being similar is related to the pairwise similarity. The resulting selected objects get mapped to a similar probability distribution in a lower-dimensional space. The algorithm minimizes the difference between the two selected distributions with the objective of finding a lower-dimensional representation that preserves the original data structure [29,30].

Similar to the PCA method, our approach involves utilizing the parameters of the RZ, RX, and RY gates for algorithm development. In this instance, we compute pairwise similarities by measuring the distance between each pair of parameter gate values within the dataset.

$$p_{ij} = \frac{p_{ji} + p_{ij}}{2n} \tag{10}$$

In the equations, p_{ji} and p_{ij} represent the conditional probabilities of a point j given point i and vice versa, with n denoting the total number of points in the dataset. Notably, for t-SNE, p_{ii} and p_{jj} are both zero, and p_{ij} is equivalent to p_{ji} . Equation (10) is responsible for computing pairwise similarities within the original space. For instance, it calculates the similarity between a specific parameter of an RZ gate and the parameter of an RX gate. This calculation is repeated for each pair of parameter gate values in the dataset, specific to a given QAOA model. Then,

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq i} \left(1 + \|y_k - y_i\|^2\right)^{-1}}, \tag{11}$$

is applied to create the map candidates $y = y_1, y_2, \dots, y_n$ in the lower dimensional space. These candidates are initially set randomly, commonly using a Gaussian distribution with a small variance centered at the origin. In order to find the best mapping relations, t-SNE minimizes the Kullback–Leibler divergence, given by:

$$KL(P||Q) = \sum_{ij} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right) \tag{12}$$

5. Experiments and Results

The PCA method attempts to find correlations between data by analyzing the variance between points in the dataset and then reducing the original attributes into a new basis with fewer dimensions. For the experiments performed in this study, we also obtained the explained variance to identify how much variance or information is presented in each component. To compare the entangled and non-entangled models, we performed a PCA model individually and for each pair of models (with compatible dimensions) that differed only in the entanglement stage of the mixing operator.

In t-SNE, we follow a similar procedure; however, t-SNE provides us with information about the relationships between the data using a different method. Specifically, t-SNE generates dimensionality reduction by modeling the data as a pairwise probability distribution, where each distribution represents the likelihood of each data point being related to other data points. During the process of representing the data in a lower-dimensional space, the algorithm reduces the Kullback–Leibler divergence, which is based on the relative Shannon entropy. The new represented data hold as much information as possible from the original data points. In our experiments, we also obtained the Kullback–Leibler divergence (KL-Divergence) after optimization, which represents the amount of information loss in the final embedding. A low divergence value is generally considered as better, as it indicates that the low-dimensional embedding is a good representation of the high-dimensional data, while a high divergence value indicates a significant loss of information in the final embedding. In simpler terms, a lower KL-Divergence value signifies better information preservation. This means that there are more detectable correlations between the data when utilizing t-SNE.

5.1. PCA Applied to QAOA Dataset

In our application of PCA to each dataset, we initially performed an individual PCA analysis using the first three PCA components and recorded the corresponding explained variance. Subsequently, we compared the individual PCA projections by pairing models that had the same number of parameter gate values (same dimensions). This comparison involved using the first three PCA components of both models and examining how the combined PCA projection differed from the original individual maps. This allowed us to assess the variations between the projections.

It is important to note that the PCA individual and paired approaches were applied to all three different levels of depth to establish a fair basis for comparison between models. For the $1L$ depth level, which corresponds to the $3p$ model (three parameters), PCA is not necessary for dimensionality reduction since the number of parameters is equal to the number of PCA components that we are seeking. However, employing PCA in this case allows us to identify correlations between the parameters, indicating the relative importance of certain gates within the QAOA. For the $2L$ and $3L$ depth levels, corresponding to the $6p$ (six parameters) and $9p$ (nine parameters) models, respectively, the first three PCA components provide information about parameter correlations within the QAOA as well as dimensionality reduction.

Table 1 presents the explained variances for the individual PCA projections for the $4n$ cyclic configuration max-cut problem. In the case of the first $1L$ model, which comprises three parameters, we observed that the second row ($3p$ entangled model) exhibited a decrease in correlation in the first PCA component compared to the non-entangled model. This decrease in correlation suggests a reduced significance of this component in terms of representation importance.

However, the second and third values increased in the $3p$ entangled model, indicating an increased contribution to the variance for the remaining PCA components.

For the $6p$ parameter models, the entangled model demonstrated higher explained variance values when comparing the first three components. Consequently, the total amount of variance contained in the components increased.

Table 1. Individual PCA projections explained variance (4n cyclic) for the first 3 PCA components.

Parameters	PCA 1	PCA 2	PCA 3
3 p	0.44317179	0.29359184	0.26323637
3 p ent	0.4189022	0.30327991	0.2778179
6 p	0.23918661	0.20909359	0.16835319
6 p ent	0.2949565	0.22615379	0.1985824

In Figure 6, we present individual PCA graphs for the 4n cyclic max-cut problem. The first two graphs of the red model (non-entangled 3p) exhibit particular line patterns for PCA 1 vs. PCA 2 and PCA 1 vs. PCA 3.

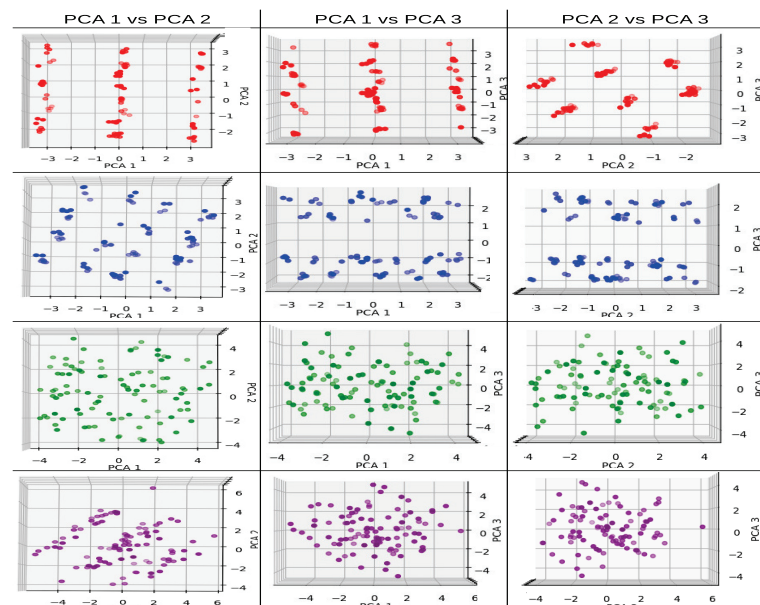


Figure 6. PCA individual graphs for 4n cyclic configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

For the blue (entangled 3p) model, a similar line behavior is observed in the first graph (PCA 1 vs. PCA 2) with one more line compared to the non-entangled model. For the last two graphs, PCA 1 vs. PCA 3 and PCA 2 vs. PCA 3, a separation pattern with two groups is visible.

In the 6p parameter (2L depth) models, the green model (non-entangled) does not exhibit any recognizable pattern or cluster in the graphs. For the purple (entangled) model, PCA 1 vs. PCA 2 has three distinct cluster lines, but no recognizable patterns are observed in the rest of the planes.

In the case of using a pair PCA model in the 4n cyclic max-cut problem for the 3p and 6p, the results for the PCA components are shown in Table 2. The PCA explained values for the 3p pair model follow an intermediate trend between the entangled and non-entangled models of the separate PCA model.

Table 2. Pair PCA projections explained variance for the first 3 PCA components for the 4n cyclic max-cut problem.

Parameters	PCA 1	PCA 2	PCA 3
3 p	0.42701248	0.29829486	0.27469266
6 p	0.22597391	0.19226576	0.18187536

For the $6p$ pair model, the variance of the PCA components seems to be closer to the $6p$ non-entangled model from the separate PCA models. Additionally, the sum of the first three components for the $6p$ accounts for only 60% of the information variance, which indicates that, for that type of model, it is harder to find a specific trend due to the low original information maintained in the new map.

In Figure 7, we present the pair PCA model for the $3p$ and $6p$ models. In the $3p$ models (red non-entangled and blue entangled), we observe that the behavior from the individual graphs is preserved. However, when we have both entangled and non-entangled models, we can see how the data of the models are projected in different areas while still following the same patterns as in the previous graphs.

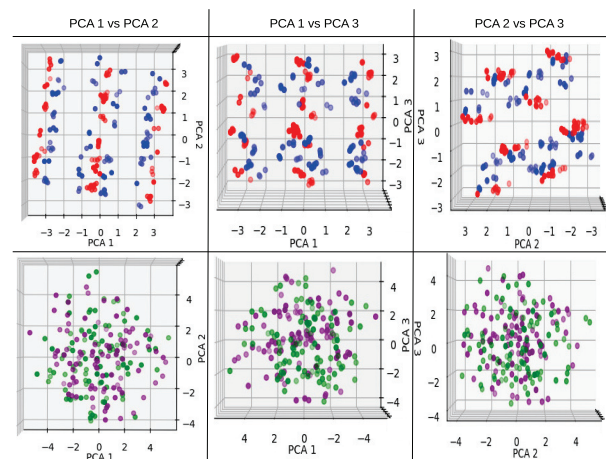


Figure 7. PCA pair graphs for $4n$ cyclic configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the $3p$ parameter $1L$ non-entangled, blue $3p$ parameter $1L$ entangled, green $6p$ parameter $2L$ non-entangled, and purple $6p$ parameter $2L$ entangled model.

In the $6p$ models, the previous patterns do not hold, and the distribution of projected points seems to be random in the majority of the graphs. Only the PCA 1 vs. PCA 3 graph shows some pattern with small centered line clusters for the entangled model (purple), while the non-entangled model (green) is more scattered compared to the purple data.

5.2. t-SNE Applied to QAOA Dataset

t-SNE was used as an additional method to identify patterns in the QAOA dataset. We aim to have multiple tools to extract information about the entanglement stage and investigate how these stages affect the overall relationships between the data.

We used different perplexity values for t-SNE analysis, which is an important parameter that determines the number of nearest neighbors used in the lower dimensional representation. We tested three different values of perplexity—3, 30 (the default value in the Sklearn package), and 99 (an extra value of 199 only for pair models)—with the goal of identifying different data behaviors at different perplexity levels. We used PCA for initialization embedding (inside t-SNE) as it provides a more globally stable solution compared to random initialization, which allows for a more precise comparison between models.

Also, as in the case of PCA, we created individual and paired t-SNE graphs for each dataset problem and for each model using $3p$ parameters, $6p$ parameters, and $9p$ parameters (for the $10n$ and $15n$ problems).

For the first problem dataset, at a perplexity of 3, the non-entangled models had higher (worse) KL-D values after the mapping compared to the entangled models as shown in Table 3. At a perplexity of 30, the $3p$ non-entangled model had a better (lower) KL-D value compared to the entangled model, and for the $6p$ models, the entangled model still had a better KL-D value compared to the non-entangled one.

Table 3. Individual KL-Divergence for 4n cyclic max-cut problem with different numbers of perplexity, considering the 3p non-entangled, 3p entangled, 6p non-entangled, and 6p entangled models.

Parameters	KL-Divergence		
	KL-D	KL-D	KL-D
3 p	0.12658161	0.18968964	0.00003131
3 p ent	0.08551478	0.20780504	0.00004092
6 p	0.55863631	0.4951154	0.00003326
6 p ent	0.35603735	0.39002356	0.00004086

Finally, at 99 perplexity, all models showed good KL-D values, indicating a better mapping for all the perplexity values tested, where the non-entangled and entangled models had a slight difference and the non-entangled models performed slightly better at this perplexity; however, this can be considered as negligible.

The individual t-SNE graphs for the 4n cyclic max-cut problem are shown in Figure 8. For the 3p non-entangled model (red), the most significant pattern can be observed at the 99 perplexity level, which has a linear pattern similar to the one obtained in the PCA graph for that particular model and problem dataset. For the 3p entangled model (blue), the 30 perplexity level shows a two-cluster pattern, and the 99 perplexity level shows a circular pattern with no data points in the center of the plane.

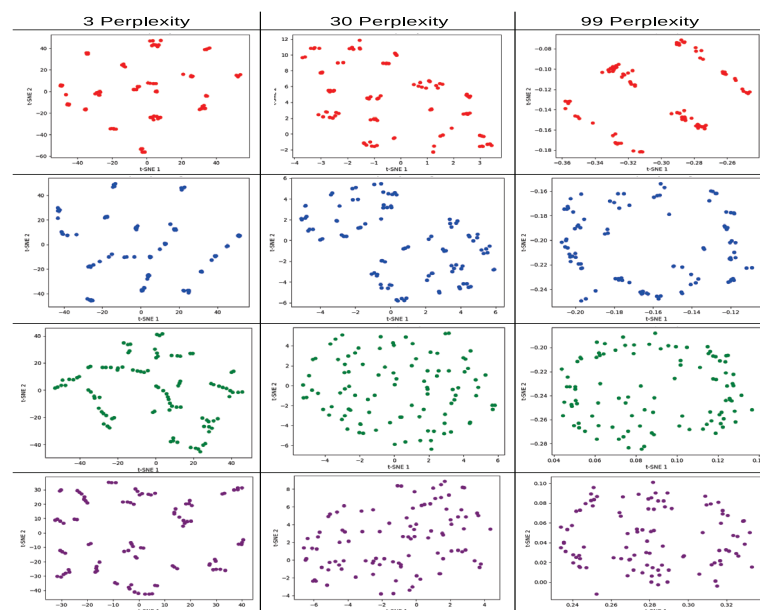


Figure 8. t-SNE individual graphs for 4n cyclic configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

For the 6p non-entangled model (green), the 30 perplexity level has a similar distribution to the one obtained in the PCA individual graph for the same model, with a random distribution pattern. And, for the 6p entangled model (purple), the most significant pattern can be observed at the 99 perplexity level, which has an external circle with a middle line pattern.

The KL-D values for the pair models in the 4n cyclic max-cut problem are presented in Table 4. Interestingly, all the best KL-D values were obtained by the 3p models. Consistent with the individual t-SNE analysis, the best KL-D values were obtained with the highest perplexity value.

In the pair t-SNE models graphs (Figure 9) for the 4n cyclic max-cut problem dataset, we start by focusing on the 3p models non-entangled and entangled (red and blue, respectively) at 199 perplexity. The line patterns of the red model are maintained, but the blue model shows a completely different distribution, where it has a similar pattern to the red model. The entangled model data contain the red points at the center, but, at the extremes, the red model seems to contain the blue data.

For the 6p models non-entangled and entangled (green and purple, respectively), interesting results are seen at the 99 and 199 perplexity values. The purple model tends to be grouped in certain areas of the plane at 99 perplexity, while the green model has a random distribution in the plane with no particular pattern. For 199 perplexity, the patterns seen in the individual graphs are maintained, with an elliptical behavior, and, in particular, the purple model shows a line pattern at the center.

Table 4. Pair KL-Divergence for 4n cyclic max-cut problem with different numbers of perplexity, considering the 3p parameters (non-entangled and entangled) and 6p parameters (non-entangled and entangled) models.

Parameter	KL-Divergence	
	3 per	30 per
3 p	0.11827804	0.24608216
6 p	0.58829921	0.73377264
	99 per	199 per
3 p	0.17905515	0.00003183
6 p	0.33970055	0.00004709

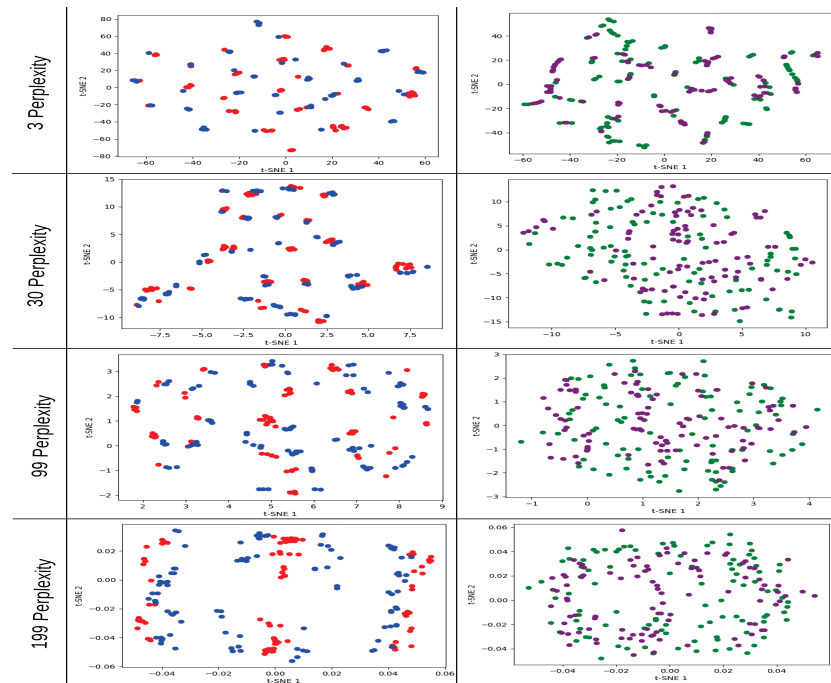


Figure 9. t-SNE pair graphs for 4n cyclic configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, 99, and 199. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

5.3. Results Analysis

In this subsection, we provide a comprehensive overview of our findings. We observed distinct parameter distribution patterns between entangled and non-entangled models across all datasets, whether in individual or paired analysis. Furthermore, numerical disparities were evident, as seen in the explained variance for PCA and KL-Divergence in t-SNE, highlighting the differences between the two mixing operator variants.

In the results obtained from the PCA method, the $3p$ models (corresponding to the $1L$ depth) for both cyclic and complete max-cut problems exhibit the best values for explained variance in the first three components. This is due to the fact that these models have the same number of parameters (dimensions) as the number of PCA components, resulting in no dimensionality reduction and no loss of information. This characteristic sets these models apart, and it is interesting to observe that there are differences between the components, indicating correlations between certain gate parameters within the QAOA. However, further studies are needed to determine the specific interactions between gates that are more significant, which are perceived as greater variances for certain PCA components.

From a graphical perspective, the $3p$ models tend to exhibit linear pattern behaviors, where the type of problem where the parameter values come from contributes to a certain consistency in the observed patterns.

In the $6p$ models (corresponding to $2L$ depth) analyzed using PCA, we observe more interesting behaviors due to the increased complexity of these models. The dimensionality reduction provided by the first three PCA components allows for a more effective application of the PCA strategy overall.

Examining the results of explained variances in the individual PCA graphs for the cyclic configuration problem datasets of $4n$, $10n$, and $15n$ (Tables 1, A3 and A7), we find that the entangled models consistently yield to better (higher) values in the PCA components. This behavior is interesting as it indicates a discernible difference between the entangled and non-entangled models. It suggests that the presence of the entanglement stage in the mixing operator leads to a greater amount of information (variance) contained in the QAOA parameters, which can be detected and maintained by the PCA method.

From a graphical standpoint, focusing on the PCA 1 vs. PCA 2 plane, which contains the most relevant or informative data, as seen in Figures 6, A3 and A9, there are noticeable differences between the non-entangled (green) and entangled (purple) models. In the case of the non-entangled models, the distribution of mapped data appears to be random, which can be attributed to the individual rotations (gates) of the model. On the other hand, the entangled models exhibit clustering behaviors, leading to distinct visual differences in the graphs. Despite the PCA method being unaware of the fact that the processed data originate from an entangled circuit, it is capable of detecting and representing the differences in data distribution.

Also, in the case of the $6p$ models for complete configuration problems, the $4n$, $10n$, and $15n$ problem datasets (Tables A1, A5 and A9), we observe a similar behavior as seen in the cyclic problem datasets. The entangled models consistently exhibit higher values in the PCA components compared to the non-entangled models, which could be seen in the individual variance for each PCA component (most cases) and the total amount of variance contained by the PCA model (all cases).

Examining the graphical representations (Figures A1, A6 and A12), most of the entangled models exhibit clustering behaviors, while the non-entangled models do not show a clear pattern or distribution. These observations further highlight the distinguishing characteristics between entangled and non-entangled models in terms of their PCA representations.

For the $9p$ models ($3L$ depth), both in the cyclic and complete configurations, we once again observe higher PCA values for the entangled models regardless of the type of problem. However, it is important to note that the total amount of variance in the $9p$ models is relatively low. Consequently, when examining the graphical representations (Figures A4, A7, A10 and A13), we should not draw definitive conclusions. The observed

behaviors or patterns in the graphs tend to vary from one problem to another. Therefore, further analysis and investigation are needed to fully understand the implications of the PCA results for $9p$ (or more complex) models.

In the pair PCA models, we observed a decreasing trend in variances as the number of parameters increased, namely for the $3p$, $6p$, and $9p$ models. It is important to note that the $3p$ models should not be compared in the same manner as the $6p$ and $9p$ models due to the number of PCA components generated.

The purpose of the paired graphs (Figures 7, A2, A5, A8, A11 and A14) was to determine if the individual behaviors could be captured within a pair PCA. This would suggest that differences between parameters in QAOA models could be detected within the same PCA. In most cases, the individual behaviors were indeed maintained in the pair graphs, supporting the notion that distinct parameter characteristics could be identified using the pair PCA approach.

For the t-SNE analysis, the results presented in Tables 3, A11, A13, A15, A17 and A19 show a clear tendency of generating better (lower) KL-Divergence values for the entangled models, with the difference being more pronounced depending on the perplexity value.

In the individual t-SNE analysis, the best results were generally generated by the $3p$ models. This can be attributed to the fact that these models have only three parameters (only one set of RZ , RX , and RY gate parameters), and the t-SNE projection to the plane does not lose a significant amount of information in the process. The best KL values were reported at 99 perplexity, where all the models generated good values that were closer to zero.

For the paired t-SNE models presented in Tables 4, A12, A14, A16, A18 and A20, the best KL values were also reported for the $3p$ models. In this case, the quality of the reported values decreased as the number of parameters increased for the majority of perplexity values tested.

The worst KL values were obtained at 30 perplexity. At 199 perplexity, we obtained the best KL values, similar to the individual case at 99 perplexity. Most paired models generated good KL-Divergence values, indicating a better representation in the low-dimensional space when using the t-SNE method.

In the graphical results for the individual t-SNE models presented in Figures 8, A15, A17, A18, A20, A21, A23, A24, A26 and A27, we selected the graphs generated at 99 perplexity as the best representation due to the value of the KL-Divergence value.

In the $3p$ models, we observed similar behaviors as those observed in the PCA graphs. The non-entangled $3p$ model (red) consistently exhibited a three-line cluster pattern across different problems and depths of the QAOA model. On the other hand, the entangled $3p$ model (blue) showed varying patterns depending on the problem type and QAOA depth, often generating line clustering patterns, although they were not as well-defined as those of the non-entangled model.

For the $6p$ models, we observed similar patterns between the entangled (purple) and non-entangled (green) models. At 99 perplexity, most of the graphs displayed an elliptical pattern, with the entangled models sometimes exhibiting more pronounced grouping behavior in certain areas of the plane. The same behavior observed in the $6p$ models was also reported in the more complex $9p$ models, where, at the highest perplexity, both non-entangled (orange) and entangled (brown) models generated elliptical patterns, with the non-entangled models exhibiting a more evenly distributed pattern around the ellipse.

In the paired t-SNE graphical results presented in Figures 9, A16, A19, A22, A25 and A28, we observed distinct patterns and behaviors. In the $3p$ models, at certain perplexity values, particularly higher perplexity values like 99 or 199, clear differences were observed between the non-entangled (red) and entangled (blue) models. This indicates that the paired t-SNE is capable of differentiating between different types of data within the model.

In some cases, the distributions observed in the individual graphs were maintained in the paired t-SNE plot, while, in other cases, similarities with the PCA graphs were observed. For the $6p$ models, significant differences were observed between the non-entangled (green)

and entangled (purple) models. The non-entangled model tended to exhibit a more random distribution in the t-SNE plane across different perplexity values, while the entangled model showed a tendency to be more concentrated in certain areas. At 199 perplexity, both models recreated the elliptical behavior observed in the individual graphs, where the entangled model exhibited differences compared to the individual graphs.

Specifically, the entangled model showed more clusters around a certain distribution, while the non-entangled model continued to exhibit a more evenly distributed pattern.

For the $9p$ models, similar behaviors were observed as in the $6p$ models. The non-entangled model (orange) tended to be more evenly distributed in the t-SNE plane across different perplexity values, and a clear elliptical pattern was generated at 199 perplexity. On the other hand, the entangled model (brown) displayed a tendency to be more grouped in certain areas of the plane at different perplexity values. At 199 perplexity, the entangled models followed the elliptical pattern but appeared more compressed in certain areas of the distribution.

6. Conclusions

PCA and t-SNE show graphical and numerical differences between the parameter distributions of the QAOA models: the entangled models achieved greater correlations while the non-entangled models showed lower levels of correlations between parameters in the different QAOA models datasets.

The PCA method reveals differences in the amount of variance contained in the PCA components depending on the type of model dataset processed. The entangled models consistently exhibit higher variance values, either in each PCA component or in the total amount of variance.

However, the PCA method is not suitable for achieving good mapping in a low-dimensional space for the datasets investigated in this work. We observed a significant reduction in the amount of information captured by the PCA components as the number of parameters increased. The most complex model tested ($9p$ parameters, $3L$ layers of depth) usually contained less than 50% of the original variance in the first three PCA components.

In some cases, the paired PCA graphs were able to retain the patterns observed in the individual PCA graphs, which is important for visually distinguishing between entangled and non-entangled models.

In general, t-SNE, whether applied to individual or paired models, outperformed the PCA method. This can be observed from the KL-Divergence values obtained at different perplexities, indicating a better representation in the low-dimensional space.

In the individual t-SNE models, we also noticed variations in the KL values between non-entangled and entangled models. Broadly speaking, the entangled models displayed superior (lower) KL values, which we attribute to the presence of the entanglement stage in the mixing operator. This stage enhances the capacity to preserve correlation information arising from the relationships between parameter gate values, something that the t-SNE models can effectively capture.

Lastly, in the paired t-SNE models, clear differences were observed between non-entangled and entangled models at various perplexity values. For $3p$ models, more linear pattern distributions were observed, while for $6p$ and $9p$ models, non-entangled models exhibited more random and elliptical distributions, whereas entangled models displayed a tendency to cluster while following a certain pattern depending on the dataset. These findings highlight the ability of t-SNE to visually distinguish the differences in data relationships between non-entangled and entangled models.

7. Future Work

For future research, it is important to conduct additional investigation into the interpretation of the observed distributions. At present, it is premature to conclude whether these specific patterns occur universally in all models with an entanglement stage, regardless of the problem. It is also unclear whether different patterns may emerge in other

types of problems, indicating the presence or absence of an entanglement stage in QAOAs. Further exploration and analysis are necessary to gain a comprehensive understanding of these phenomena.

Additionally, it would be valuable to explore alternative optimization methods for QAOA dataset generation in order to compare the obtained results. This analysis would help to identify which behaviors persist across different optimization methods and which ones are influenced by the specific method employed to solve the presented problems.

Author Contributions: Conceptualization, B.G.S.; Methodology, B.G.S. and S.-H.D.; Software, B.G.S.; Validation, B.G.S., G.-H.S. and S.-H.D.; Formal analysis, B.G.S., G.-H.S. and S.-H.D.; Investigation, B.G.S., G.-H.S. and S.-H.D.; Resources, B.G.S.; Data curation, B.G.S.; Writing—original draft, B.G.S.; Writing—review & editing, G.-H.S. and S.-H.D.; Visualization, B.G.S.; Supervision, G.-H.S. and S.-H.D.; Project administration, G.-H.S. and S.-H.D.; Funding acquisition, G.-H.S. and S.-H.D. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the support of the projects 20230316-SIP-IPN, 20231289-SIP-IPN, Mexico.

Data Availability Statement: All the experiment data results can be found in https://github.com/BrianSarmina/QAOA_SHC-RR (accessed on 12 September 2023).

Acknowledgments: We would like to thank the referees for making invaluable suggestions and criticisms that have improved the manuscript greatly. The author, B. Sarmina, would like to express sincere gratitude and acknowledge the support of CONACYT (National Council for Science and Technology) of Mexico for providing a scholarship during his Ph.D. program in computer science at the Center for Computing Research of Instituto Politécnico Nacional. S. H. Dong started this work on the leave of IPN due to permission of research stay in China.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

QAOA	Quantum Approximate Optimization Algorithm
VQA	Variational Quantum Algorithm
NISQ	Noisy Intermediate-Scale Quantum
PCA	Principal Component Analysis
t-SNE	t-distributed Stochastic Neighbor Embedding
KL-Divergence	Kullback–Leibler Divergence
SHC-RR	Stochastic Hill Climbing with Random Restarts

Appendix A. PCA Variances and Graphs

In this appendix, we present the complementary results for the experiments developed in order to have a bigger set of results to generate better insights and conclusions about the PCA analysis (and the variances obtained) applied in the max-cut problems solved using QAOAs.

The PCA explained variance for the $4n$ complete max-cut problem is presented in Table A1. For the $3p$ models, the entangled model exhibits more variance in PCA 1 and PCA 2. However, an interesting aspect to note is that the variance in PCA 3 for the entangled model is almost zero, which is a completely different phenomenon compared to the previous explained variances of the $4n$ cyclic problem. The variance for the $6p$ models is closer to the one observed in the previous problem. In both entangled models, the variances in PCA 1 increase compared to the same number of parameter models without entanglement.

Table A1. Individual PCA projections explained variance (4n complete) for the first 3 PCA components.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.50298884	0.311534	0.18547716
3 parameters ent	0.57473804	0.42450533	0.00075663
6 parameters	0.2283193	0.21189936	0.18558982
6 parameters ent	0.23935453	0.22027888	0.18451389

Upon analyzing Figure A1, we observe that, for the $3p$ models, the behavior of the non-entangled model (red) is similar to the previous problem, where the major change is presented in the PCA 1 vs. PCA 2 graph, which has the same type of linear distribution but with a different orientation. In the entangled model (blue), there is a two-line cluster pattern that differs from the previous result. It is interesting to note that these two clusters are presented in the PCA 1 vs. PCA 2 and PCA 2 vs. PCA 3 graphs, only changing the perspective. In the case of the $6p$ models, the green model (non-entangled) has a similar distribution as before, with a random dispersion of points in the different plane perspectives, with no distinguishable clusters or patterns. However, for the entangled model (purple), the three-cluster behavior from the cyclic problem can be observed in the PCA 1 vs. PCA 2 graph again. This phenomenon could represent an increase in correlations between the data when the entanglement stage is implemented.

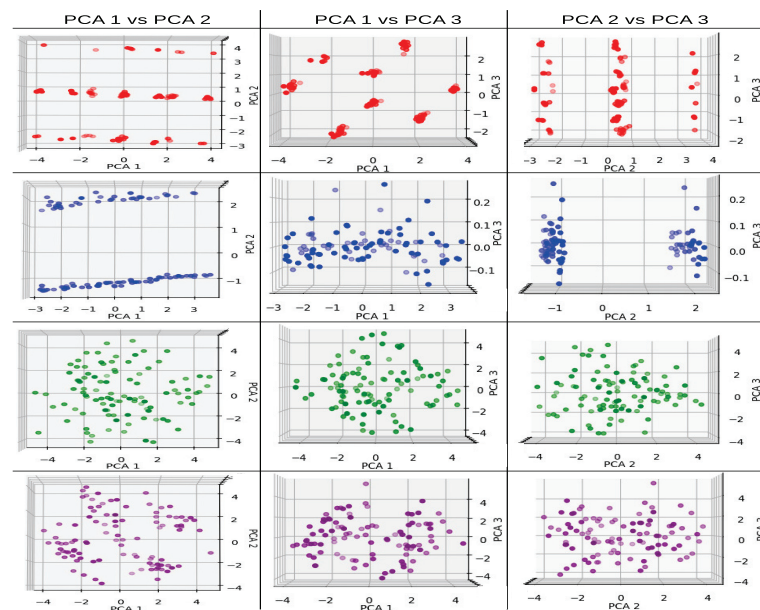


Figure A1. PCA individual graphs for 4n complete configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the $3p$ parameter 1L non-entangled, blue $3p$ parameter 1L entangled, green $6p$ parameter 2L non-entangled, and purple $6p$ parameter 2L entangled model.

The explained variance for pair PCA models for $3p$ and $6p$ (corresponding to 1L and 2L of depth, respectively) presented in Table A2 corresponds to the 4n complete max-cut problem. The variances obtained are similar to those in the cyclic problem, with the only noticeable difference being the variance in the PCA 3 component for the $3p$ pair PCA, which is considerably lower than in the previous problem. For the $6p$ pair PCA, the values are pretty close to one another, differing at most by 0.3 in the first three PCA components.

Table A2. Pair PCA projections explained variance for the first 3 PCA components for the 4n complete max-cut problem.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.4655341	0.34885069	0.18561521
6 parameters	0.21851466	0.18343424	0.17652118

The pair PCA graphs for the 4n max-cut complete configuration problem are shown in Figure A2. For the 3p models (red and blue), the PCA 1 vs. PCA 2 graph shows patterns similar to those observed in the individual PCA graphs. In the PCA 2 vs. PCA 3 graph, the pattern appears to be preserved for the non-entangled model, while the entangled model shows a distribution of points that is closer together. It is worth noting that the blue points in the PCA 1 vs. PCA 3 graph are contained in a particular line pattern that shows a clear difference between the projection of the non-entangled and entangled models. Moving on to the 6p models (green and purple), the first PCA 1 vs. PCA 2 graph shows a scatter distribution of points for the non-entangled model (green), while the entangled model (purple) has two lightly clustered areas that are barely distinguishable. However, in the PCA 1 vs. PCA 3 graph, there is a clear pattern of three elliptic clusters for the entangled model. Adding the PCA 2 and PCA 3 graph with two clusters presented in the entangled model, these clusters can be interpreted as the pair PCA model being capable of detecting particular correlations between the non-entangled and entangled data due to the distribution of values from the different models. Overall, the pair PCA graphs suggest that the entanglement stage is capable of revealing additional information about the correlations between the different models.

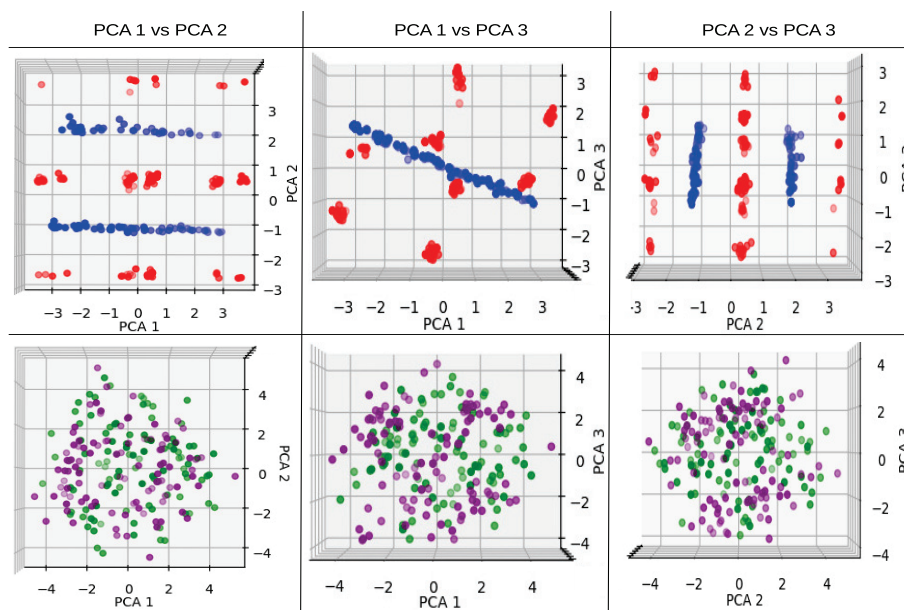


Figure A2. PCA pair graphs for 4n complete configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

The explained variances for the 10n cyclic max-cut problem are presented in Table A3. For this problem, we compiled results for three different levels of depth: 1L (3p parameters), 2L (6p parameters), and 3L (9p parameters). The first two models (1L and 2L) have similar results compared with the previous cyclic problem for 4n. However, it is important to mention that the entangled models with 6p and 9p parameters increase the amount of variance in the PCA 1 component compared with the non-entangled models. Additionally, the PCA 2 and PCA 3 components have more variance in general compared with the

non-entangled models (including the $3p$ model). This increase in variance around the components is due to the entanglement stage, which increases the amount of covariances between the elements.

Table A3. Individual PCA projections explained variance (10n cyclic) for the first 3 PCA components.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.4824018	0.32759113	0.19000707
3 parameters ent	0.42582802	0.29568193	0.27849004
6 parameters	0.22421832	0.19958267	0.19614923
6 parameters ent	0.27777425	0.20329209	0.18833349
9 parameters	0.1681149	0.14530348	0.13463924
9 parameters ent	0.17743445	0.1615457	0.14796169

The graphs for the $3p$ and $6p$ parameter models in the $10n$ cyclic max-cut problem are presented in Figure A3. For the first $3p$ non-entangled model (red), the distribution is similar to that of the first cyclic problem, while the entangled model (blue) has a quite different data projection with no recognizable pattern. In the $6p$ parameter models, the behavior has some similarities with the previous cyclic problem. The non-entangled model (green) has a random distribution of points with no distinguishable cluster or pattern, but in the entangled model (purple) in the PCA 1 vs. PCA 2 graph, there is one major cluster in the center with two smaller ones at the sides, which is similar to the distribution in the previous problem, where the rest of the PCA 1 vs. PCA 3 and PCA 2 vs. PCA 3 graphs (in the purple model) also seem to be conglomerating the points at the center of each graph.

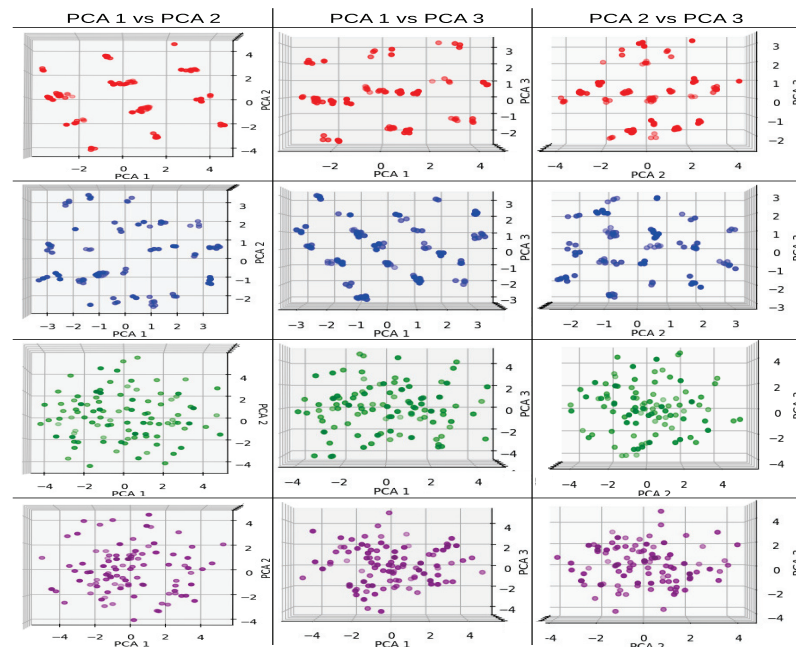


Figure A3. PCA individual graphs for $10n$ cyclic configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the $3p$ parameter $1L$ non-entangled, blue $3p$ parameter $1L$ entangled, green $6p$ parameter $2L$ non-entangled, and purple $6p$ parameter $2L$ entangled model.

In the last $3L$ depth model with $9p$ parameters (Figure A4, the distribution for the non-entangled model (orange) is very similar to the $6p$ non-entangled model, with a random distribution of points and no distinguishable clusters. For the entangled model, only the

PCA 1 vs. PCA 2 graph seems to have a pattern, with two light clusters divided by a central line.

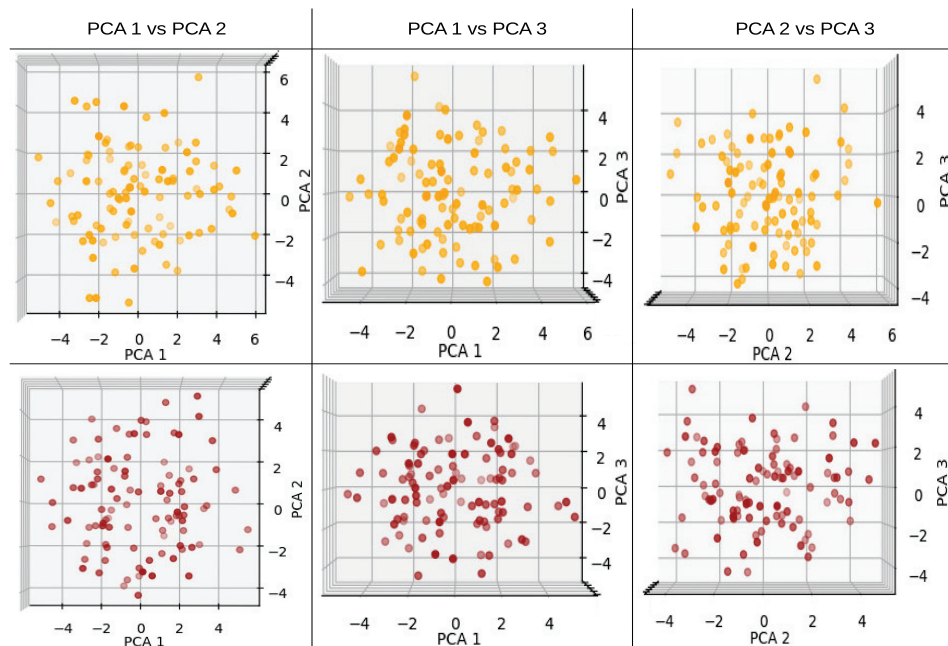


Figure A4. PCA individual graphs for 10n cyclic configuration max-cut problem solved using QAOA, first 3 components. Orange 9p parameter 3L non-entangled and purple 9p parameter 3L entangled model.

The pair PCA explained variances for the 10n cyclic max-cut problem are presented in Table A3. The variances for the 3p and 6p models were very similar to the previous results, with a considerable decrease in the amount of variance represented in each PCA component as the number of parameters increased. This trend persists with the 9p pair PCA model values.

Table A4. Pair PCA projections explained variance for the first 3 PCA components for the 10n cyclic max-cut problem.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.45436643	0.31238244	0.23325113
6 parameters	0.21821814	0.1962479	0.17828477
9 parameters	0.14672728	0.13850982	0.12026834

The pair PCA graphs are presented in Figure A5. The 3p and 6p models have similar behaviors to the previous cyclic problem. The 3p model preserves almost the same distribution of points in the projection as the individual graphs, while, for the 6p models, the green (non-entangled) model has a random distribution of points similar to the individual graph. However, the purple (6p entangled) model exhibits a clear cluster pattern behavior for the PCA 1 vs. PCA 2 and PCA 1 vs. PCA 3 graphs. In the more complex models with 9p, there is no clear behavior of the projection distribution, and due to the low variance for each PCA component presented in the graph, we cannot establish a precise interpretation of the results because the PCA mapping has lost a large amount of original information.

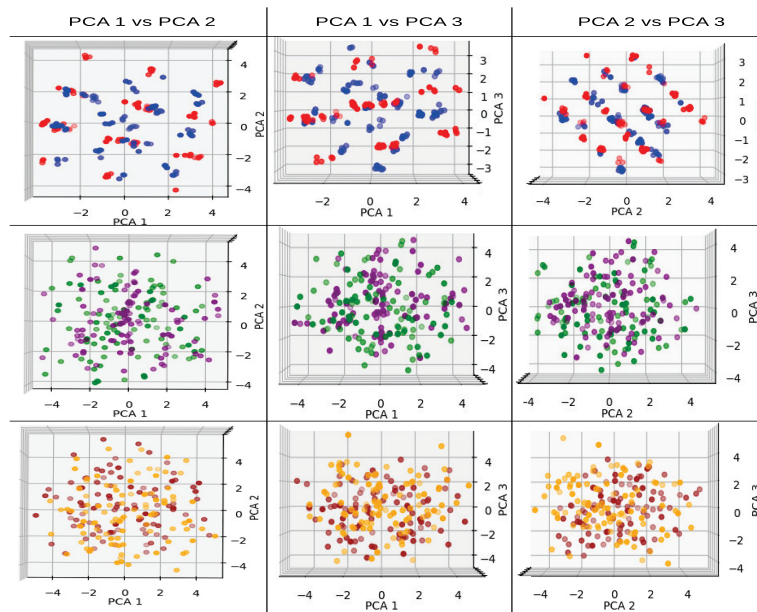


Figure A5. PCA pair graphs for 10n cyclic configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, purple 6p parameter 2L entangled model, orange 9p parameter 3L non-entangled, and brown 9p parameter 3L entangled model.

The explained variances for the first three PCA components in the individual models for the 10n complete problem are presented in Table A5. Comparing the table with the individual variances for the cyclic problem, we observe some interesting results. Starting with the 3p model, the entangled version shows an increase in the amount of variance associated with PCA 1 and PCA 2 compared to the non-entangled model, which is the opposite of what was observed in the cyclic problem. For the 6p and 9p models, the entangled versions show a decrease in the amount of variance contained in PCA 1 and PCA 2 components with respect to the non-entangled models, while PCA 3 has a greater value in general. Analyzing these results with the cyclic problem, we can observe how the problem’s structure modifies how the entanglement stage in the mixing operator can affect the variance distribution along the PCA components. However, the difference between the components in the entangled models seems to be lower compared to the non-entangled ones. Another important observation is that, in the 6p and 9p models, the total amount of variance captured by the first three PCA components is slightly higher in the entangled models compared to the non-entangled ones.

Table A5. Individual PCA projections explained variance (10n complete) for the first 3 PCA components.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.49767238	0.33234628	0.16998134
3 parameters ent	0.588512290	0.411087639	0.0004000709
6 parameters	0.24720666	0.19582826	0.16085243
6 parameters ent	0.23681691	0.19549636	0.17376718
9 parameters	0.17181209	0.14208608	0.12933972
9 parameters ent	0.15950657	0.14702026	0.14211113

The individual graphs presented in Figure A6 exhibit a behavior similar to that of the 4n complete max-cut problem. The 3p non-entangled (red) and entangled (blue) models have almost the same type of distribution for the PCA 1 vs. PCA 2, albeit with different

orientations. In the case of the $6p$ non-entangled model (green), it has a similar random distribution as in the previous problems (not only the complete problems). Meanwhile, the $6p$ entangled model (purple) has a major cluster on the left and two small clusters on the right in the PCA 1 vs. PCA 2 graph, and, the PCA 2 vs. PCA 3 graph has a two-cluster distribution.

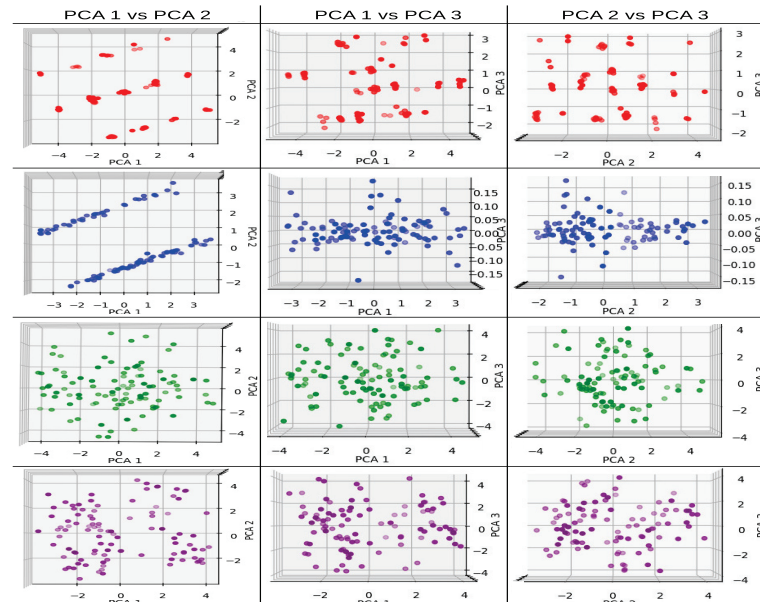


Figure A6. PCA individual graphs for $10n$ complete configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the $3p$ parameter $1L$ non-entangled, blue $3p$ parameter $1L$ entangled, green $6p$ parameter $2L$ non-entangled, and purple $6p$ parameter $2L$ entangled model.

Completing the individual graphs in the $10n$ complete max-cut problems, we have the $9p$ parameter models ($3L$ depth) in Figure A7. In this case, there is no clear behavior in the non-entangled and entangled models. This is not surprising because the amount of variance that each perspective has is very low, and it cannot give a correct representation of the original information.

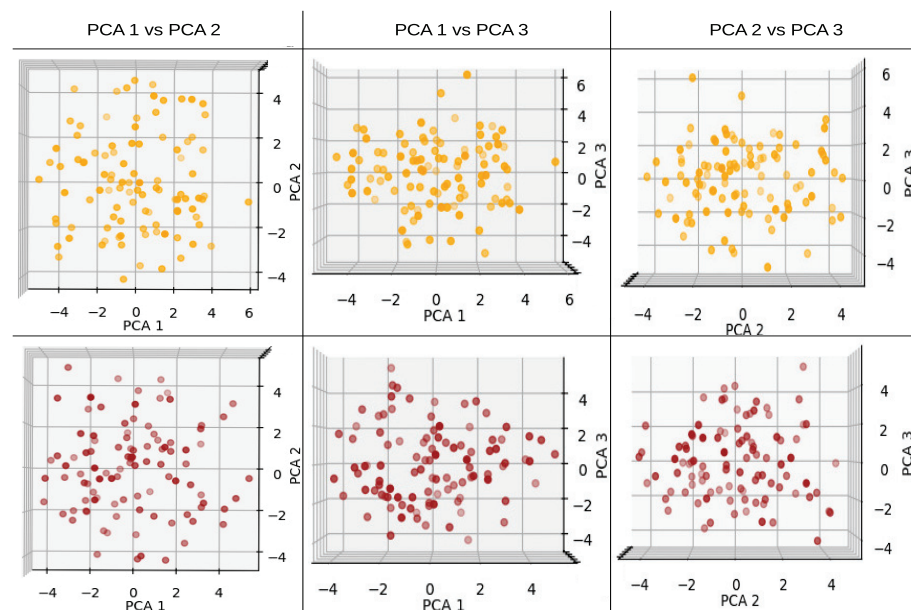


Figure A7. PCA individual graphs for $10n$ complete configuration max-cut problem solved using QAOA, first 3 components. Orange $9p$ parameter $3L$ non-entangled and brown $9p$ parameter $3L$ entangled model.

The explained variances for the 10n complete max-cut problem are presented in Table A5. The values of variance are very similar to the ones obtained in the cyclic problem, showing a decreasing trend in importance (due to the decrease in variance) with an increase in the number of parameters. In the case of the 9p models, the amount of variance in the first three components is less than 40%.

Table A6. Pair PCA projections explained variance for the first 3 PCA components for the 10n complete max-cut problem.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.46514143	0.34410486	0.19075371
6 parameters	0.20148807	0.19106115	0.17569887
9 parameters	0.1569094	0.137659	0.12379254

The pair PCA model for the 10n complete max-cut problem is presented in Figure A8. In the case of the 3p models, the distribution is very similar to the individual graphs, with changes observed in the PCA 1 vs. PCA 3 and PCA 2 vs. PCA 3 distributions for the entangled model (blue). For the 6p parameter models, the distribution shows no clear pattern or clusters, with only two light clusters and some outliers in the PCA 1 vs. PCA 2 plot. However, due to the low variance of the PCA components, these results cannot be considered conclusive. Finally, for the 9p models, the distribution appears to be random, with no clear patterns observed. Again, due to the low variance, these results are to be expected.

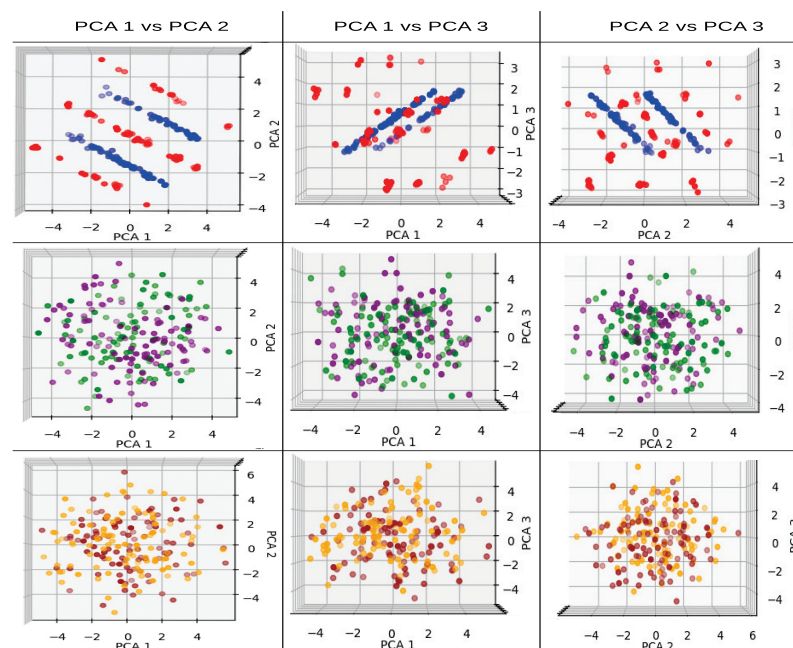


Figure A8. PCA pair graphs for 10n complete configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, purple 6p parameter 2L entangled model, orange 9p parameter 3L non-entangled, and brown 9p parameter 3L entangled model.

The results shown in Table A7 exhibit similar behavior to those observed in the 10n cyclic problem. For the 3p models, the non-entangled model demonstrates greater values for PCA 1 and PCA 2, whereas the entangled model produces a more evenly distributed variance in PCA 2 and PCA 3. For the 6p models, the entangled model displays higher variance values for PCA 1 and PCA 2, as well as for the first three components, which is

consistent with the earlier findings. In the case of the $9p$ models, the entangled model has higher values for all PCA components, although the difference is not substantial. Overall, these results suggest that the entangled models generally perform better in terms of the amount of variance information that the model is able to detect and project in the new PCA space.

Table A7. Individual PCA projections explained variance (15n cyclic) for the first 3 PCA components.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.4647624	0.32860527	0.20663233
3 parameters ent	0.38826281	0.34828084	0.26345635
6 parameters	0.22713273	0.19519285	0.18337327
6 parameters ent	0.27183177	0.23569836	0.19037519
9 parameters	0.1592166	0.1491276	0.14043665
9 parameters ent	0.17716564	0.15707367	0.13968417

For the individual PCA graphs of the 15n cyclic max-cut problem, refer to Figure A9. In the $3p$ models, both non-entangled (red) and entangled (blue), we observe a behavior similar to previous experiments. Particularly, interesting patterns can be observed in the PCA 1 vs. PCA 2 and PCA 2 vs. PCA 3 planes. Shifting our focus to the $6p$ models, the non-entangled model (green) exhibits patterns consistent with previous observations, with no clear discernible behavior or pattern across different PCA planes. However, for the entangled model (purple), the presence of the three-line clustering behavior, previously observed in the PCA 1 vs. PCA 2 plane for the $4n$ and $10n$ cyclic problems, reappears.

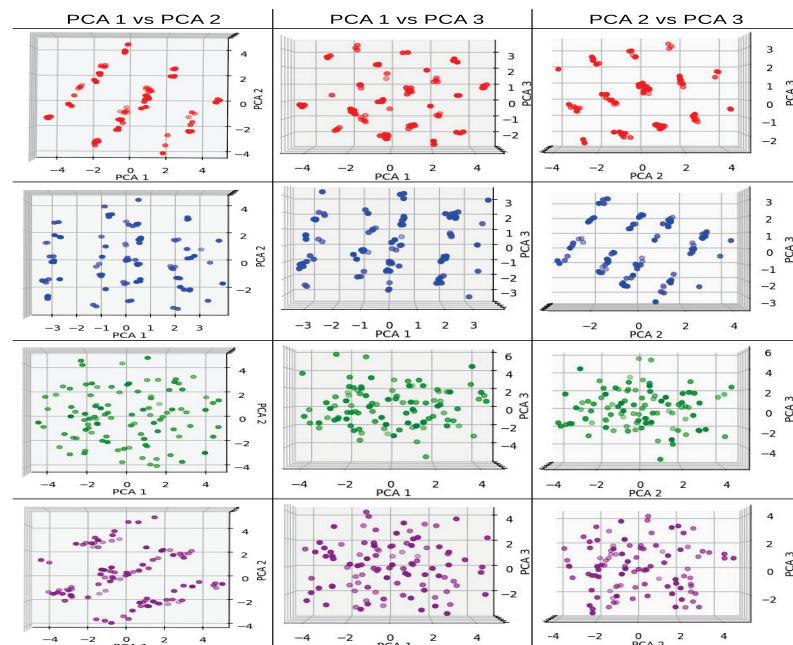


Figure A9. PCA individual graphs for 15n cyclic configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the $3p$ parameter $1L$ non-entangled, blue $3p$ parameter $1L$ entangled, green $6p$ parameter $2L$ non-entangled, and purple $6p$ parameter $2L$ entangled model.

In the $9p$ parameters models (Figure A10), no clear patterns can be distinguished in both the non-entangled (yellow) and entangled (brown) models. This lack of clear patterns is not surprising considering the low variance associated with the first three

PCA components. As previously mentioned, when the variance is low, it becomes more challenging to achieve a meaningful mapping in the low-dimensional space using PCA.

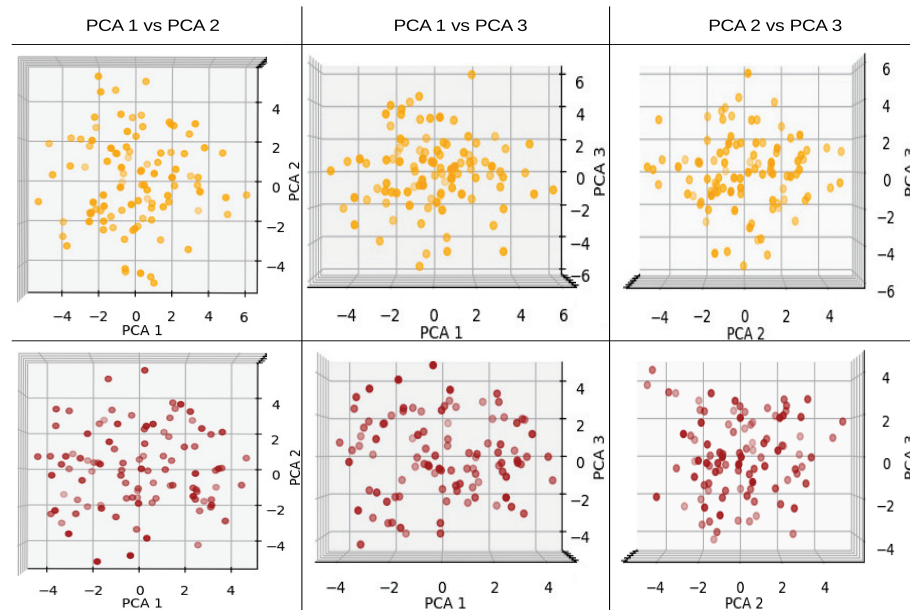


Figure A10. PCA individual graphs for 15n cyclic configuration max-cut problem solved using QAOA, first 3 components. Orange $9p$ parameter $3L$ non-entangled and brown $9p$ parameter $3L$ entangled model.

Now, regarding the pair PCA models in the 15n cyclic problem, we observe similar behavior as in the previous cyclic problem. The $3p$ models present the best PCA values, which is not surprising since this model has the same dimension as the PCA components. The $6p$ models accumulate approximately 60% of the variance in the original data for the first three PCA components, making them the second-best performing models. Finally, the $9p$ models have lower PCA values, with less than 40% of the variance of the data in the first three components.

Table A8. Pair PCA projections explained variance for the first 3 PCA components for the 15n cyclic max-cut problem.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.41927731	0.34498266	0.23574004
6 parameters	0.23265555	0.18633541	0.18280923
9 parameters	0.15291139	0.14312175	0.12984339

The results presented in Figure A11 exhibit similar patterns to those observed in previous problems for the $3p$ and $6p$ models. In particular, for the $6p$ models, both the entangled (purple) and non-entangled (green) models continue to exhibit their respective distribution behaviors. The non-entangled model displays a scattered distribution in the PCA 1 vs. PCA 2 plane, while the entangled model demonstrates clustering behavior in the PCA 1 vs. PCA 3 plane. However, for the $9p$ models, neither the entangled (brown) nor the non-entangled (yellow) models exhibit clear patterns. The only noticeable difference is that the data points in the entangled model tend to be closer together, although this distinction is difficult to discern.

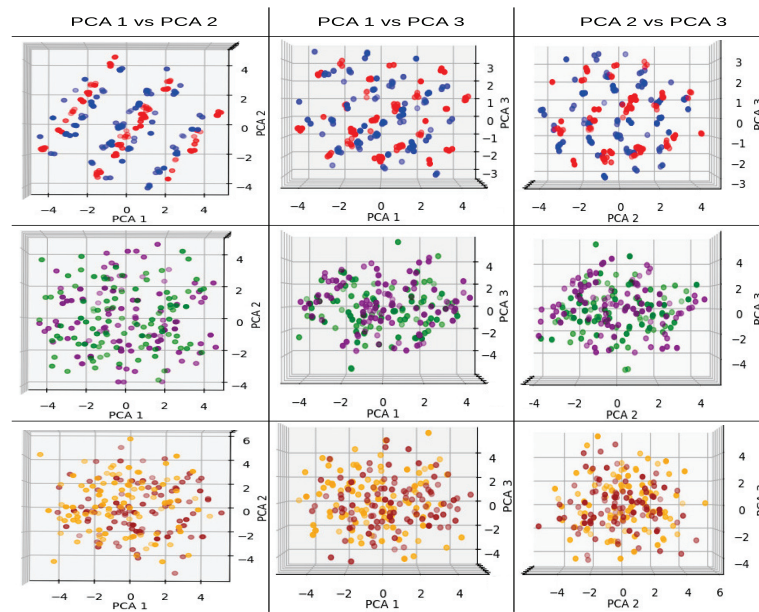


Figure A11. PCA pair graphs for 15n cyclic configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, purple 6p parameter 2L entangled model, orange 9p parameter 3L non-entangled, and brown 9p parameter 3L entangled model.

The final problem examined using PCA is the 15n complete configuration max-cut problem. The individual PCA variances are presented in Table A9. In the 3p models (both entangled and non-entangled), the behavior aligns with the previous findings. However, in the 6p models, the distribution of explained variance differs from the 10n complete configuration problem. Here, the entangled model demonstrates a noticeable increase in variance due to the presence of the entanglement stage, resembling the behavior observed in the cyclic problems. Similarly, in the 9p models, the entangled QAOA exhibits a higher total amount of variance in the first 3 PCA components, mirroring the results observed in the 6p models. Additionally, consistent with the 10n problem, the total amount of variance is higher in the entangled models for both the 6p and 9p cases.

Table A9. Individual PCA projections explained variance (15n complete) for the first 3 PCA components.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.41775729	0.35261959	0.22962311
3 parameters ent	0.38723863	0.36655932	0.24620205
6 parameters	0.22555648	0.19689321	0.18005628
6 parameters ent	0.28402724	0.21446195	0.18652736
9 parameters	0.16446156	0.15145779	0.14506611
9 parameters ent	0.18999579	0.16075822	0.13010846

The individual graphs using PCA for the 15n complete max-cut problem are presented in Figure A12. In the 3p models, both the entangled (blue) and non-entangled (red) models exhibit patterns similar to those observed in the previous 4n and 10n problems. However, there are some differences in the entangled model, particularly in the PCA 1 vs. PCA 2 and PCA 1 vs. PCA 3 planes, where more line patterns are observed compared to the one or two line patterns seen in the previous problems. Moving on to the 6p models, the non-entangled model (green) continues the trend observed in previous problems, showing no clear tendency or discernible behavior in the data distribution. In contrast, the entangled

model (purple) exhibits no clear distribution in the PCA 1 vs. PCA 2 plane, which is different from the patterns observed in the 4n and 10n problems. The PCA 1 vs. PCA 3 plane shows some noisy cluster distribution, but it is not well-defined.

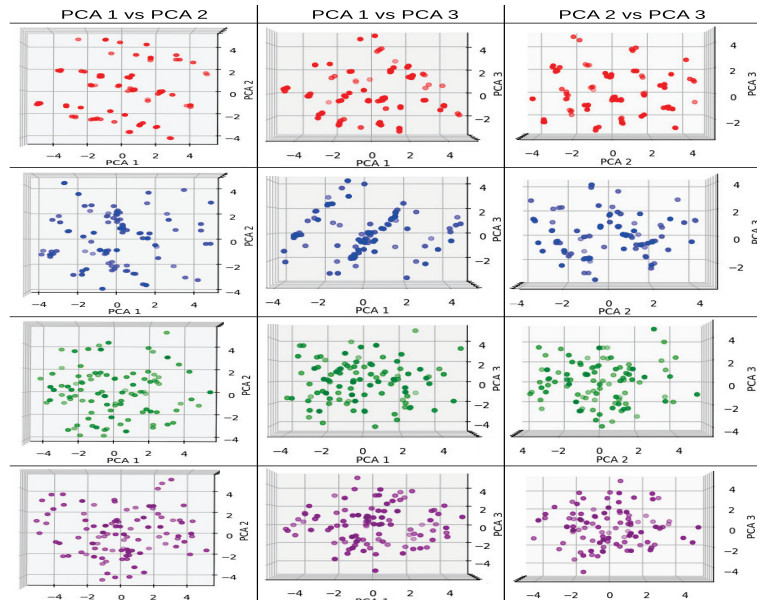


Figure A12. PCA individual graphs for 15n complete configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

In the 9p models depicted in Figure A13, no clear or distinguishable patterns can be observed in both the non-entangled (yellow) and entangled (brown) models. This behavior is consistent with the patterns observed in the 10n complete max-cut problem.

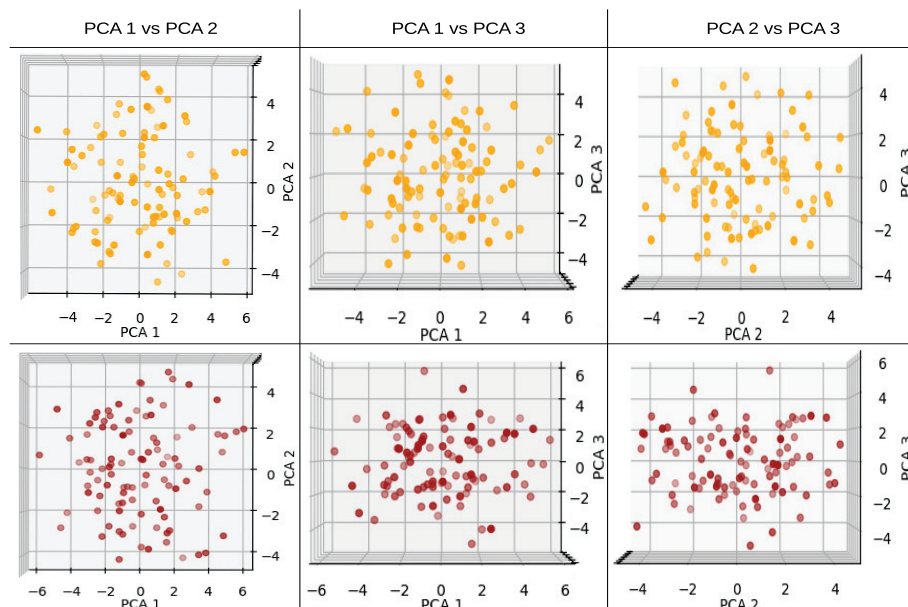


Figure A13. PCA individual graphs for 15n complete configuration max-cut problem solved using QAOA, first 3 components. Orange 9p parameter 3L non-entangled and brown 9p parameter 3L entangled model.

The explained variance for the 15n complete max-cut problem is presented in Table A10. The distribution of PCA variance per model follows a similar trend to that observed in the

previous problems. The $3p$ models exhibit the highest variance values, which is expected as the number of parameters matches the number of PCA components. As the number of parameters increases, the quality of the components decreases, resulting in lower variance values. Notably, the $6p$ models show a slight increase in the total amount of variance compared to the $10n$ problem, bringing them closer to the values obtained in the cyclic problems. The variance values for the $9p$ models are similar to those observed in the $10n$ problems, both for cyclic and complete configurations, representing the lowest values among the tested models.

Table A10. Pair PCA projections explained variance for the first 3 PCA components for the $15n$ complete max-cut problem.

Parameters	PCA 1	PCA 2	PCA 3
3 parameters	0.36035505	0.35400924	0.28563571
6 parameters	0.24550126	0.19657842	0.16664964
9 parameters	0.1512382	0.13967618	0.12791876

The pair PCA graphs for the $15n$ complete max-cut problem are presented in Figure A14. In the $3p$ models, both non-entangled (red) and entangled (blue), the essence of the individual graphs is preserved, similar to the previous pair graphs. However, the $6p$ models do not exhibit a clear behavior or pattern in any of the planes. This behavior is consistent with the $15n$ cyclic problem but differs from the distribution observed in the $4n$ and $10n$ complete problems, where some clustering patterns were observed. Lastly, in the $9p$ models, the non-entangled model (yellow) displays a random distribution pattern across all planes, while the entangled model (brown) shows a slightly more concentrated patterns in certain areas, as seen in the PCA 1 vs. PCA 2 and PCA 1 vs. PCA 3 planes.

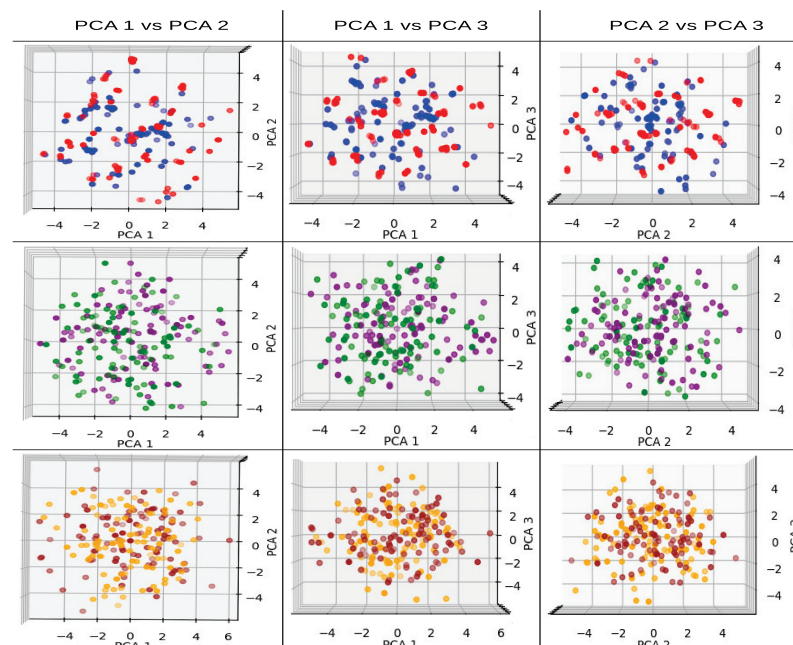


Figure A14. PCA pair graphs for $15n$ complete configuration max-cut problem solved using QAOA, first 3 components. Red corresponds to the $3p$ parameter $1L$ non-entangled, blue $3p$ parameter $1L$ entangled, green $6p$ parameter $2L$ non-entangled, purple $6p$ parameter $2L$ entangled model, orange $9p$ parameter $3L$ non-entangled, and brown $9p$ parameter $3L$ entangled model.

Appendix B. t-SNE Graphs and KL Divergence Values

In this appendix, we present the complementary results for the experiments developed using t-SNE analysis (and KL-D values obtained) applied in the max-cut problems solved using the QAOA.

The results for KL-D for individual t-SNE in the $4n$ complete max-cut problem are presented in Table A11. In general, the values for 3 and 30 perplexity have values that are closer compared to the cyclic $4n$ problem, with the entangled models having a better perplexity value (lower) compared to the non-entangled models. The best KL-D values were obtained with the 99 perplexity, which is interpreted as the best model that represents the original properties of the data.

Table A11. Individual KL-Divergence for $4n$ complete max-cut problem with different numbers of perplexity, considering the $3p$ non-entangled, $3p$ entangled, $6p$ non-entangled, and $6p$ entangled models.

Parameters	KL-D (3 per)	KL-D (30 per)	KL-D (99 per)
3 parameters	0.15324634	0.17391348	0.00002262
3 parameters ent	0.14360289	0.05689293	0.00003242
6 parameters	0.55960602	0.52287281	0.00004825
6 parameters ent	0.37428555	0.38009176	0.00002795

The individual graphs for the $4n$ complete max-cut problem (Figure A15) show that the $3p$ non-entangled (red) model has a distribution similar to the previous problem, and, specifically for the 99 perplexity, the three-line pattern is similar to the one obtained before. The pattern in this perplexity value is also similar to some perspectives obtained in the PCA graphs. The $3p$ entangled (blue) model has very different patterns than the ones obtained in the cyclic problem. The most interesting results are the similarities of the two-line clusterization obtained at the 30 and 99 perplexity, which replicate some patterns from PCA graphs obtained in the same problem. Moving to the $6p$ models, the non-entangled (green) model has a random distribution behavior observed in the cyclic problem and the PCA graphs at the 30 perplexity. At the 99 perplexity, the elliptical pattern of the cyclic problem is observed again, but with a wider edge compared to the cyclic t-SNE graph. Last, for the $6p$ entangled (purple) model, the 99 perplexity shows a particular pattern with two small elongated clusters at the extremes of the graph and two small clusters at the center of the plane with some outlier points trying to connect both small clusters.

The KL-D results for the pair t-SNE models in the $4n$ complete max-cut problem are presented in Table A12. The results from 3 to 99 perplexity are quite similar to those obtained in the cyclic problem, where the $3p$ model shows better KL-D results, leading to a better representation of the data in the final plane. However, in the case of the 199 perplexity, the $6p$ parameter models exhibit better KL-D values, which is a different result compared to the cyclic problem.

Table A12. Pair KL-Divergence for $4n$ complete max-cut problem with different numbers of perplexity, considering the $3p$ parameters (non-entangled and entangled) and $6p$ parameters (non-entangled and entangled) models.

Parameters	KL-Divergence			
	3 per	30 per	99 per	199 per
3 parameters	0.17788552	0.22446597	0.11483472	0.00005328
6 parameters	0.58645886	0.77581	0.36295095	0.00005059

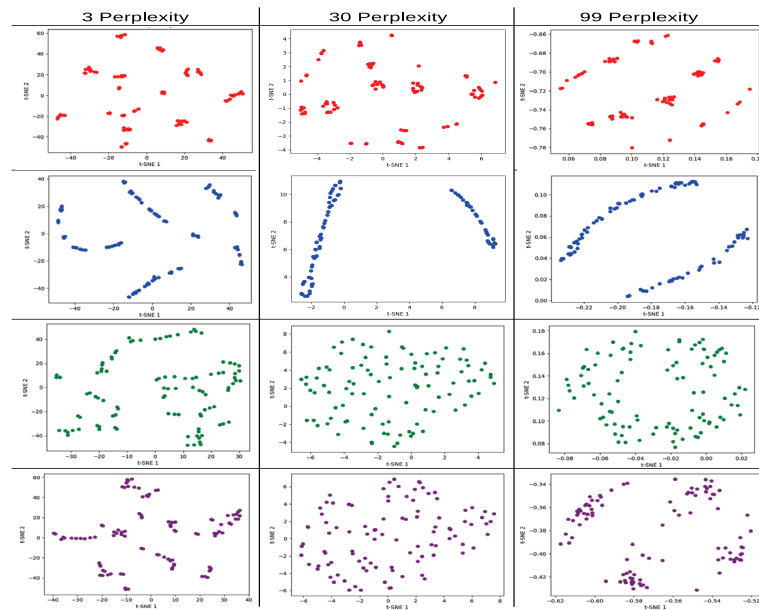


Figure A15. t-SNE individual graphs for 4n complete configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Red corresponds to the $3p$ parameter 1L non-entangled, blue $3p$ parameter 1L entangled, green $6p$ parameter 2L non-entangled, and purple $6p$ parameter 2L entangled model.

In the pair graphs for the 4n complete configuration (Figure A16), we can observe interesting behavior patterns starting with the $3p$ models. The entangled model (blue) shows a similar pattern in all perplexity values, which can be observed more clearly from the 30 to 199 perplexity range. The blue model distributes itself over particular areas on the t-SNE mapped plane, but with a smooth distribution of mapped points. For the $6p$ models, the most interesting distribution is observed at a 199 perplexity value. Here, the mapped data distribution is very similar to the one obtained in the cyclic problem. However, in this case, there are only a few points that cross the middle of the elliptic pattern.

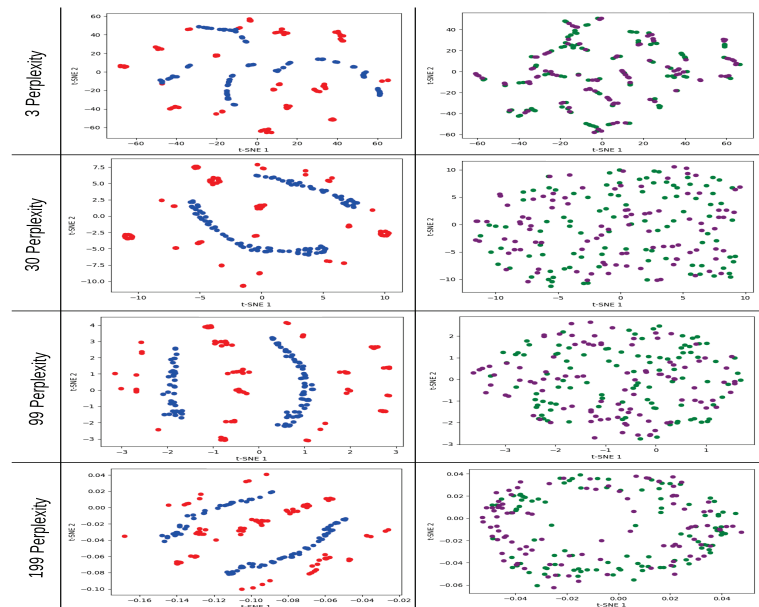


Figure A16. t-SNE pair graphs for 4n pair complete configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, 99, and 199. Red corresponds to the $3p$ parameter 1L non-entangled, blue $3p$ parameter 1L entangled, green $6p$ parameter 2L non-entangled, and purple $6p$ parameter 2L entangled model.

Table A13. Individual KL-Divergence for 10n cyclic max-cut problem with different perplexity values, considering the 3p non-entangled, 3p entangled, 6p non-entangled, 6p entangled, 9p non-entangled, and 9p entangled models.

Parameters	KL-D (3 per)	KL-D (30 per)	KL-D (99 per)
3 parameters	0.11449474	0.18797217	0.00002232
3 parameters ent	0.08714075	0.17103997	0.00004818
6 parameters	0.61964202	0.53861362	0.00004456
6 parameters ent	0.33782312	0.40035829	0.0000446
9 parameters	0.6599322	0.60457009	0.00003925
9 parameters ent	0.690759	0.54887885	0.00004668

When analyzing the 10n max-cut problem with cyclic configuration using t-SNE, we observed that the 3p entangled model performed the best in terms of KL-D value for the 3 perplexity. However, as the number of parameters increased, the quality of the projected model decreased, but, on average, entangled models produced better results than non-entangled ones. For the 30 perplexity, the 3p entangled model remained the best, and all the entangled models had better KL-D results. At the 99 perplexity, the 3p non-entangled model had the best KL-D value, but every model at this perplexity level showed a good KL-D value, which enabled a good representation of the data in the t-SNE plane.

The graphs for the 10n cyclic max-cut problem are presented in Figure A17. The patterns observed in the 3p models, both entangled (blue) and non-entangled (red), are pretty similar to the ones observed in the 4n problem. For the 6p non-entangled model (green), the distribution of data is similar to the one obtained in the 4n problem. However, for the entangled model (purple) at 99 perplexity, the elliptic behavior is no longer distinguishable. In this case, the green and purple models at 99 perplexity have a similar distribution of points.

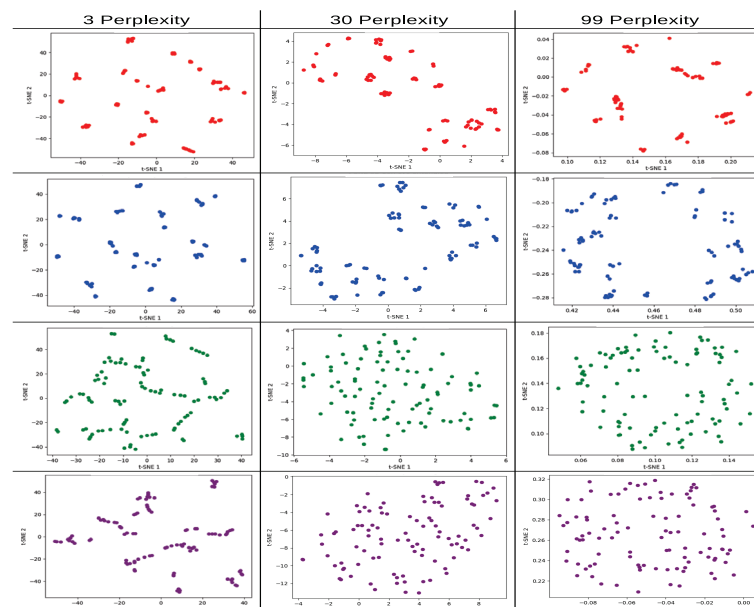


Figure A17. t-SNE individual graphs for 10n cyclic configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

In Figure A18, at 99 perplexity, the $9p$ non-entangled model (orange) exhibits a similar elliptic pattern as the $6p$ non-entangled model, while the entangled model (brown) displays a more defined elliptic pattern.

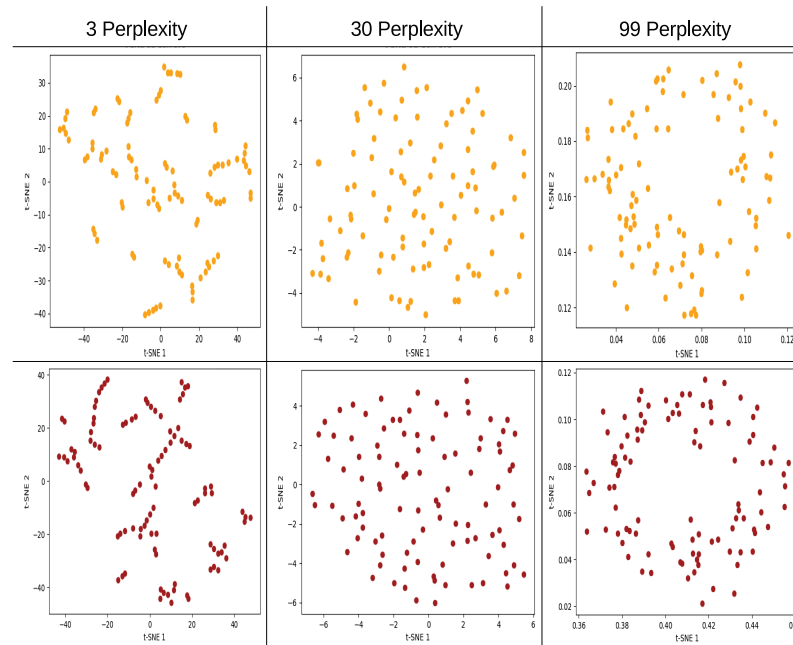


Figure A18. t-SNE individual graphs for 10n cyclic configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Orange corresponds to the $9p$ parameter $3L$ non-entangled and brown $9p$ parameter $3L$ entangled.

Table A14 presents the pair KL-D divergences for different depth QAOA models. For the first three perplexity values (3, 30, and 99), the best KL-D values were obtained by the $3p$ models. At 30 perplexity, it is interesting to see a value greater than 1 obtained by the $9p$ models, which is the highest (lower quality) value obtained so far. Finally, at 199 perplexity, the best KL-D values for each t-SNE model were obtained, with the best KL-D value corresponding to the $9p$ models.

Table A14. Pair KL-Divergence for 10n cyclic max-cut problem with different perplexity values, considering the $3p$ parameters (non-entangled and entangled), $6p$ parameters (non-entangled and entangled), and $9p$ parameters (non-entangled and entangled) models.

KL-Divergence				
Parameters	3 per	30 per	99 per	199 per
3 parameters	0.12718032	0.22481607	0.1436608	0.00006457
6 parameters	0.61863911	0.73845208	0.37629709	0.0000484
9 parameters	0.80385178	1.04350173	0.423794	0.00004115

The pair t-SNE model graphs for the 10n cyclic max-cut problem can be seen in Figure A19. At the 3 perplexity, the non-entangled models (red, green, and orange) seem to be distributed in specific patterns in the plane, while the entangled models (blue, purple, and brown) match in certain areas of the t-SNE plane. Moving on to the 30 perplexity, the $3p$ models (red non-entangled and blue entangled) distribute in very particular patterns that cannot be interpreted as a specific structure. In the $6p$ models, the green (non-entangled) model seems to have a smooth random distribution, while the purple (entangled) model is concentrated in certain areas of the plane. The $9p$ model follows a similar behavior as the $6p$ graph, where the orange (non-entangled) model is almost randomly distributed, and

the brown (entangled) model is more concentrated. For the 99 perplexity, the $3p$ graph has a similar pattern to the one observed in the $4n$ cyclic problem, where the red and blue models have fewer matches compared to the previous perplexity values. In the $6p$ graph, the behavior is similar to the one observed at the 30 perplexity, where the green (non-entangled) model is scattered in the t-SNE plane and the purple (entangled) model is more concentrated in certain areas. For the $9p$ graph, there is a difference between the orange (non-entangled) and brown (entangled) models, where the orange model maintains the scattered distribution and the brown model has three areas where most of the points are plotted. Finally, at the 199 perplexity, the $3p$ graph has a distribution that forms a rotated square with no additional specific behavior. The $6p$ graph has a completely different distribution from the ones observed in previous graphs, even in different problems. The scale of the graph is very small, generating the presence of outliers and a particular cluster containing both entangled (purple) and non-entangled (green) models. In the $9p$ graph, both orange (non-entangled) and brown (entangled) models have an elliptic pattern, where the orange model is more scattered compared to the brown model, which preserves the elliptic pattern better.

The KL-Divergence values presented in Table A15 show similar results to those observed in the $10n$ cyclic problem, where most of the entangled models present a better KL-Divergence value after optimization, resulting in a better mapping of points in the t-SNE plane.

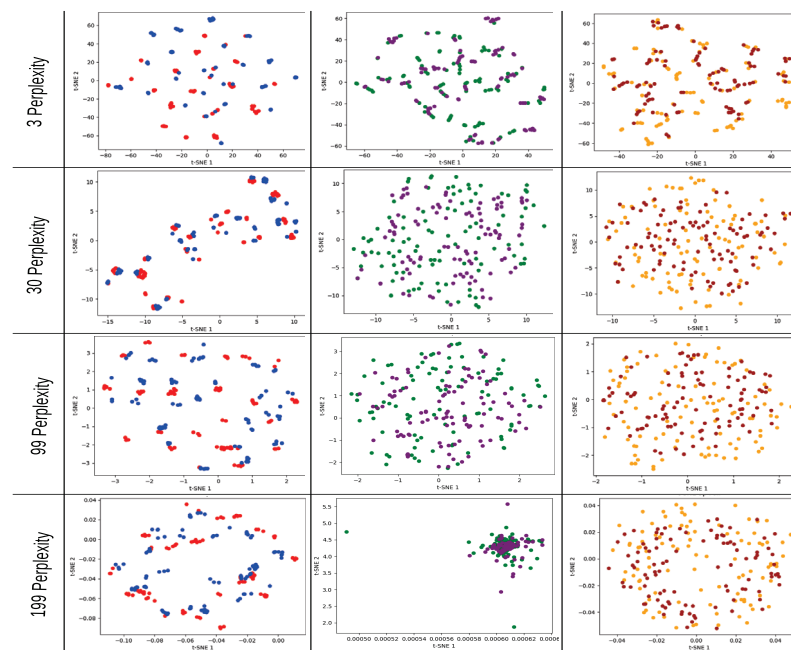


Figure A19. t-SNE pair graphs for $10n$ cyclic configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, 99, and 199. Red corresponds to the $3p$ parameter $1L$ non-entangled, blue $3p$ parameter $1L$ entangled, green $6p$ parameter $2L$ non-entangled, purple $6p$ parameter $2L$ entangled, orange $9p$ parameter $3L$ non-entangled, and brown $9p$ parameter $3L$ entangled model.

Figure A20 displays the individual t-SNE graphs for the $10n$ complete configuration max-cut problem. For the $3p$ models, the red (non-entangled) and blue (entangled) models at 3 perplexity do not exhibit a clear pattern, consistent with previous results. At 30 perplexity, the entangled model generates a line with an empty space in the middle, and the non-entangled model continues without a clear pattern. At 99 perplexity, the non-entangled (red) model presents a pattern similar to the one seen in the $4n$ problem with a complete configuration, as well as a similar pattern to the one obtained in the individual PCA graphs (PCA 1 vs. PCA 2) for the $4n$ and $10n$ problems with a similar configuration. The entangled

model (blue) at 99 perplexity presents a two-line pattern, similar to the one obtained in the previous 4n problem and the individual PCA graphs (PCA 1 vs. PCA 2) for the 4n and 10n complete configuration problems. For the 6p models, at 3 and 30 perplexity, there is no clear pattern, consistent with previous results. However, at 99 perplexity, the non-entangled (green) model appears to be distributed in an elliptical pattern at the sides of the t-SNE plane, and the entangled (purple) model creates four clusters distributed at the sides of the plane. This last result shares some similarities with the 4n complete configuration problem.

Table A15. Individual KL-Divergence for 10n complete max-cut problem with different perplexity values, considering the 3p non-entangled, 3p entangled, 6p non-entangled, 6p entangled, 9p non-entangled, and 9p entangled models.

Parameters	KL-D (3 per)	KL-D (30 per)	KL-D (99 per)
3 parameters	0.1108679	0.15153457	0.00001909
3 parameters ent	0.13275136	0.05032415	0.00004749
6 parameters	0.48524341	0.51412958	0.00005262
6 parameters ent	0.41168147	0.42243937	0.00003066
9 parameters	0.73269081	0.6897254	0.00004309
9 parameters ent	0.87109852	0.63736594	0.00004298

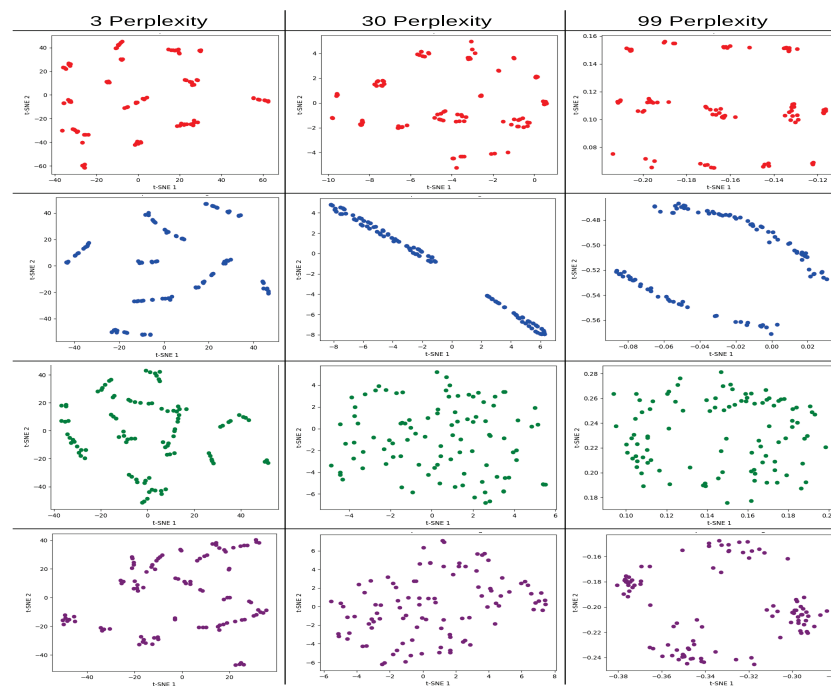


Figure A20. t-SNE individual graphs for 10n complete configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

The individual t-SNE graphs for the 10n complete configuration max-cut problem are shown in Figure A7. At both 3 and 30 perplexity, there is no clear pattern observed, with the distribution appearing random with no apparent clusters. At 99 perplexity, there is also no distinguishable pattern observed, which is different from the elliptical behavior observed in the 10n cyclic problem but consistent with the individual PCA graphs obtained for the same problem.

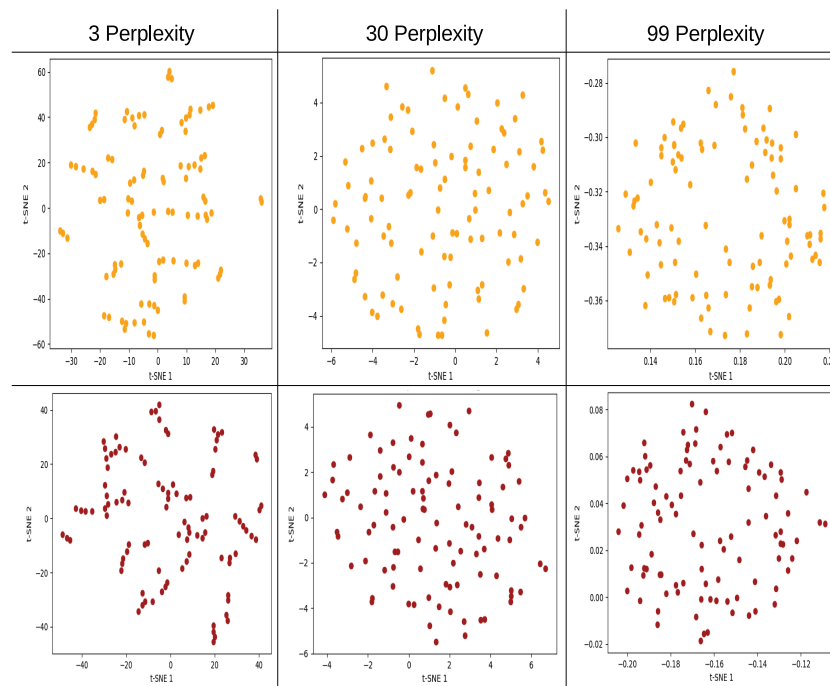


Figure A21. t-SNE individual graphs for 10n complete configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Orange corresponds to the 9p parameter 3L non-entangled and brown 9p parameter 3L entangled.

The KL-Divergence values for the pair-wise t-SNE models are presented in Table A16. The values are similar to those observed in the cyclic problem with 10n, where the worst KL-D values were obtained at 30 perplexity, particularly in the 9p parameter models, and the best KL-D values were obtained at 199 perplexity. The overall best performance was seen in the 9p models.

Table A16. Pair KL-Divergence for 10n complete max-cut problem with different perplexity values, considering the 3p parameters (non-entangled and entangled), 6p parameters (non-entangled and entangled), and 9p parameters (non-entangled and entangled) models.

Parameters	KL-Divergence			
	3 per	30 per	99 per	199 per
3 parameters	0.15265435	0.19996087	0.11178039	0.00005902
6 parameters	0.56295419	0.78792441	0.38925377	0.00004848
9 parameters	0.90329468	1.03177929	0.43718094	0.00004415

The pair graphs obtained for the 10n complete configuration problem can be seen in Figure A8. For the 3p non-entangled (red) and entangled (blue) models, similar patterns are observed as in the individual t-SNE graphs, where the entangled model preserves a two-line clusterization and the non-entangled model generates different types of lines that can be observed from 99 to 199 perplexity. It is also important to mention that the distribution for the 3p models is very similar to the ones observed in the 4n complete configuration problem in PCA 1 vs. PCA 2 and the paired t-SNE graphs. For the 6p models, the most interesting behavior is presented at 199 perplexity, where the non-entangled (green) model has an elliptical pattern with some points at the center, and the entangled (purple) model has two definite areas where the points are plotted, which are two parts of the elliptical pattern. This pattern has many similarities with the 4n nodes problem at the same perplexity. For the 9p models, in general, the non-entangled model (orange) seems to be randomly distributed at different perplexities, while the entangled (brown) model tends

to be more concentrated in certain areas of the t-SNE plane. At 199 perplexity, both models tend to generate an elliptical behavior, where the non-entangled model is better distributed around the ellipse, and the entangled model is more scattered; this pattern is similar to the one observed in the $9p$ models for the $10n$ cyclic problem.

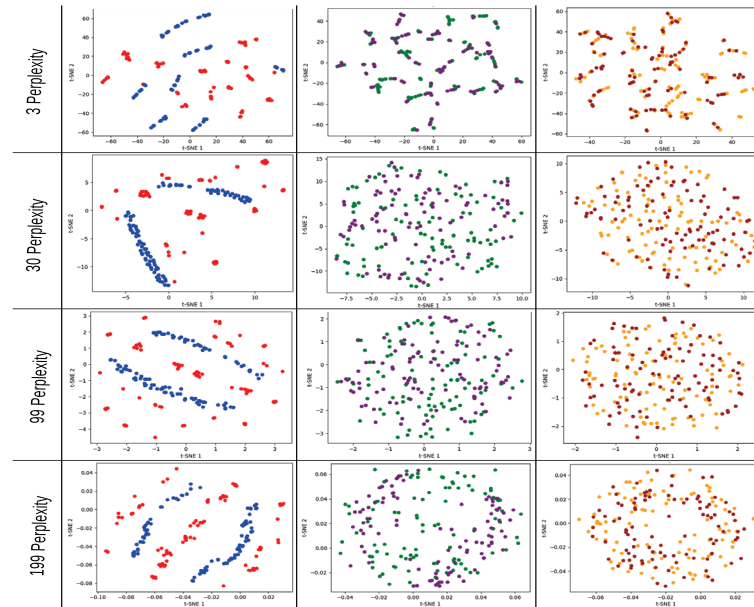


Figure A22. t-SNE pair graphs for $10n$ complete configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, 99, and 199. Red corresponds to the $3p$ parameter $1L$ non-entangled, blue $3p$ parameter $1L$ entangled, green $6p$ parameter $2L$ non-entangled, purple $6p$ parameter $2L$ entangled, orange $9p$ parameter $3L$ non-entangled, and brown $9p$ parameter $3L$ entangled model.

The KL-Divergence values presented in Table A17 correspond to the $15n$ cyclic problem. At a perplexity of 3, the entangled approaches consistently produced better KL values across all models, with the best KL value obtained in the $3p$ entangled model. At a perplexity of 30, the trend of entangled models performing better in terms of KL values continues for the more complex models with $6p$ and $9p$ ($2L$ and $3L$ depths, respectively). At a perplexity of 99, all models exhibit good KL values, which are closer to zero. When comparing these results with those reported in the $10n$ cyclic problem, we observe a consistent trend where entangled models generally yield better KL-Divergence values for different perplexities. Additionally, the best KL values for mapping are obtained at a perplexity of 99.

Table A17. Individual KL-Divergence for $15n$ cyclic max-cut problem with different perplexity values, considering the $3p$ non-entangled, $3p$ entangled, $6p$ non-entangled, $6p$ entangled, $9p$ non-entangled, and $9p$ entangled models.

Parameters	KL-D (3 per)	KL-D (30 per)	KL-D (99 per)
3 parameters	0.12295903	0.19203556	0.0000241
3 parameters ent	0.10832428	0.24514797	0.0000398
6 parameters	0.63913888	0.5105567	0.00004167
6 parameters ent	0.3930757	0.45114037	0.00003364
9 parameters	0.71460283	0.66517001	0.00004309
9 parameters ent	0.64783859	0.55502474	0.00004505

In the t-SNE individual graphs for the 15n cyclic max-cut problem (Figure A23), we observe similar behaviors as in the previous 4n and 10n cyclic problems. For the 3p models, both the non-entangled (red) and entangled (blue) models exhibit different patterns at different perplexities, and at a perplexity of 99, the non-entangled model generates the line pattern observed in previous t-SNE and PCA graphs. In the case of the 6p models, both the non-entangled (green) and entangled (purple) models show distributions that are consistent with previous problems. The non-entangled model generates an elliptic pattern with some points in the middle, while the entangled model exhibits a similar external pattern but with a more pronounced line in the middle.

The patterns observed in the 9p models at 3 and 30 perplexity (Figure A24) closely resemble those observed in the 10n cyclic and complete configuration problems. At 99 perplexity, the distribution of the non-entangled model (orange) is consistent with the previous problems, while the distribution of the entangled model (brown) follows a similar trend but with additional noise. The general pattern can still be perceived, but it is not as clear as in the previous cyclic problem.

For the paired t-SNE models of the 15n cyclic max-cut problem presented in Table A18, the observed values are similar to those of the 10n cyclic problem. At 3 perplexity, the best KL value corresponds to the paired 3p model, and as the number of parameters increases, the quality of KL-Divergence values decreases. At 30 perplexity, the best value is again obtained by the 3p models, but overall, this perplexity level yields the worst KL values. The trend of the KL quality decreasing with an increasing number of parameters persists. At 99 perplexity, the values for the 6p and 9p models are improved compared to the previous perplexities, but the 3p models remain the best performers. Finally, at 199 perplexity, the overall best KL values are reported, with all models exhibiting good KL-Divergence values, indicating a well-mapped low-dimensional space where the 6p models yield the best KL value in this case.

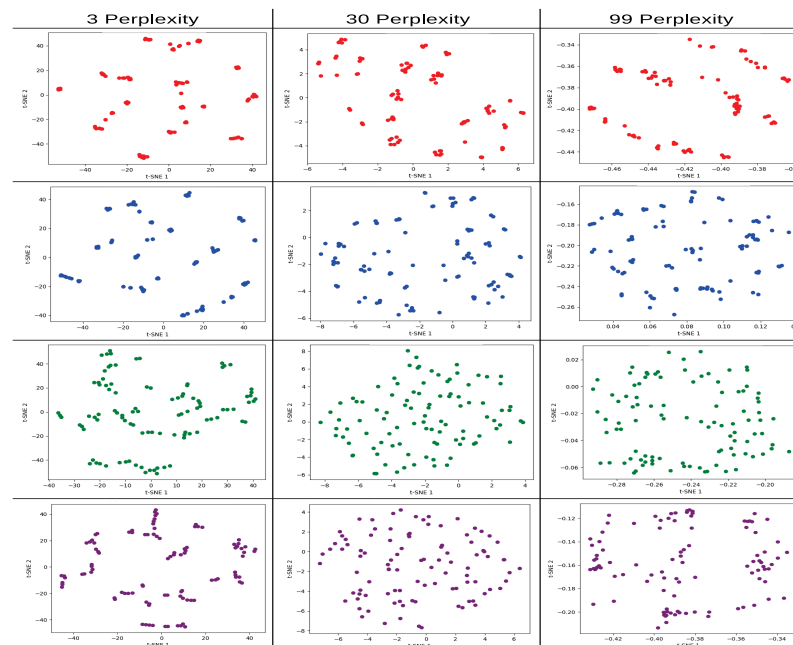


Figure A23. t-SNE individual graphs for 15n cyclic configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

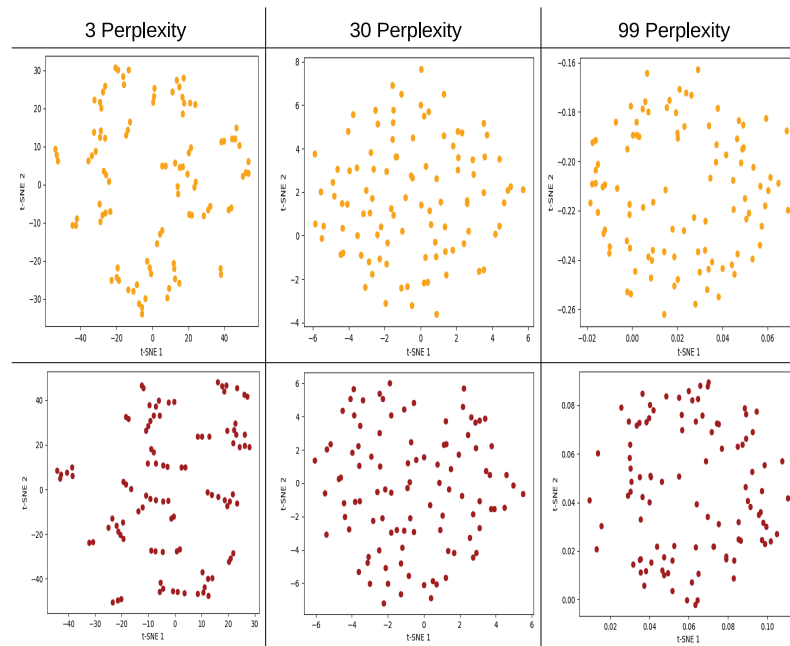


Figure A24. t-SNE individual graphs for 15n cyclic configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Orange corresponds to the $9p$ parameter $3L$ non-entangled and brown $9p$ parameter $3L$ entangled.

The graphical representation of the paired t-SNE models is presented in Figure A25. Starting with the $3p$ models, both non-entangled (red) and entangled (blue) present similar behaviors as in the $10n$ cyclic case. From 3 to 99 perplexity values, the patterns of both models appear relatively similar, with each model tending to group more in certain areas. At 199 perplexity, the difference between models becomes more pronounced, where the non-entangled model exhibits a pattern with three lines, while the entangled model simulates a containment pattern of the non-entangled model. For the $6p$ models, the observed behaviors are also similar to those reported in the $10n$ cyclic problem. The non-entangled model (green) appears more scattered in the plane from 3 to 99 perplexity, while the entangled model (purple) tends to be more concentrated in certain areas. At 199 perplexity, the non-entangled and entangled models share a closer distribution, but the entangled model stands out due to the presence of three soft clusters. Finally, for the $9p$ models, the distributions are similar to the $6p$ models from 3 to 99 perplexity, where the non-entangled model (orange) shows a random distribution across most of the t-SNE plane, while the entangled model (brown) exhibits a higher concentration in certain areas. At 199 perplexity, both models generate an elliptical pattern, with the entangled model being more grouped in certain parts of the elliptical pattern.

Table A18. Pair KL-Divergence for 15n cyclic max-cut problem with different perplexity values, considering the $3p$ parameters (non-entangled and entangled), $6p$ parameters (non-entangled and entangled), and $9p$ parameters (non-entangled and entangled) models.

Parameters	KL-Divergence			
	3 per	30 per	99 per	199 per
3 parameters	0.13307488	0.26993424	0.17919816	0.00004499
6 parameters	0.64315343	0.82122898	0.34502453	0.00004286
9 parameters	0.8600843	1.03565741	0.42468697	0.00004554

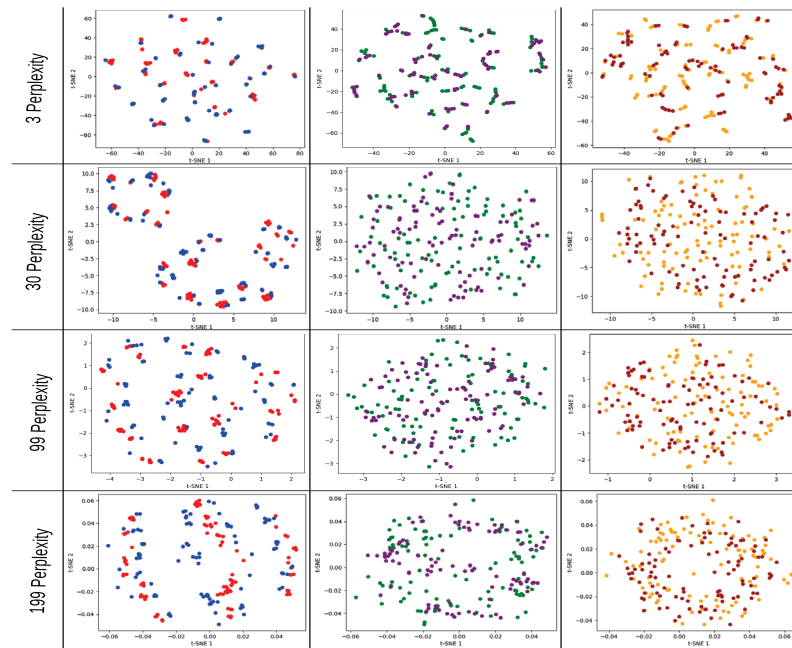


Figure A25. t-SNE pair graphs for 15n cyclic configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, 99, and 199. Red corresponds to the $3p$ parameter 1L non-entangled, blue $3p$ parameter 1L entangled, green $6p$ parameter 2L non-entangled, purple $6p$ parameter 2L entangled, orange $9p$ parameter 3L non-entangled, and brown $9p$ parameter 3L entangled model.

The last individual t-SNE KL-Divergence values correspond to the 15n complete max-cut problem, and they are presented in Table A19. At 3 perplexity, the entangled models for $6p$ and $9p$ present better KL values. However, for the $3p$ models, the non-entangled model has the best KL value overall, which differs from the values observed in the 10n complete problem where only the $6p$ entangled model was better than the non-entangled model. At 30 perplexity, all the entangled models show better values compared to their corresponding non-entangled models. This behavior is similar to what was observed in the 10n complete problem at the same perplexity. Finally, at 99 perplexity, all the models exhibit good KL-Divergence values, with the best value obtained by the $3p$ entangled model. Overall, the KL values for this problem demonstrate better results for the entangled models. They also share more similarities with the values observed in the 15n cyclic problem and, at certain perplexities, with the 10n complete problem.

Table A19. Individual KL-Divergence for 15n complete max-cut problem with different perplexity values, considering the $3p$ non-entangled, $3p$ entangled, $6p$ non-entangled, $6p$ entangled, $9p$ non-entangled, and $9p$ entangled models.

Parameters	KL-D (3 per)	KL-D (30 per)	KL-D (99 per)
3 parameters	0.09598967	0.21283571	0.00005414
3 parameters ent	0.21084341	0.16879296	0.00003727
6 parameters	0.62666488	0.56735194	0.00003864
6 parameters ent	0.34622833	0.42480648	0.00004639
9 parameters	0.82630664	0.66207534	0.00004564
9 parameters ent	0.66798007	0.57983494	0.00004709

The graphs for the 15n complete max-cut problem can be viewed in Figure A26. For the 3p models, non-entangled (red) and entangled (blue), at 3 perplexity, we observe similar patterns to those observed in previous problems. At 30 perplexity, the distribution is different from what was observed in the 10n complete problem, resembling the pattern observed in the 15n cyclic problem. At 99 perplexity, the non-entangled model exhibits a similar three-line pattern as in previous problems, but the entangled model shows a distribution with two separate areas from the middle, forming line patterns. For the 6p models, non-entangled (green) and entangled (purple), the behavior at 3 and 30 perplexity is similar to what was reported in the 10n complete and 15n cyclic problems. At 99 perplexity, the non-entangled model displays an elliptic pattern with some random points around it, while the entangled model generates a deformed elliptic pattern, resembling a butterfly-like distribution.

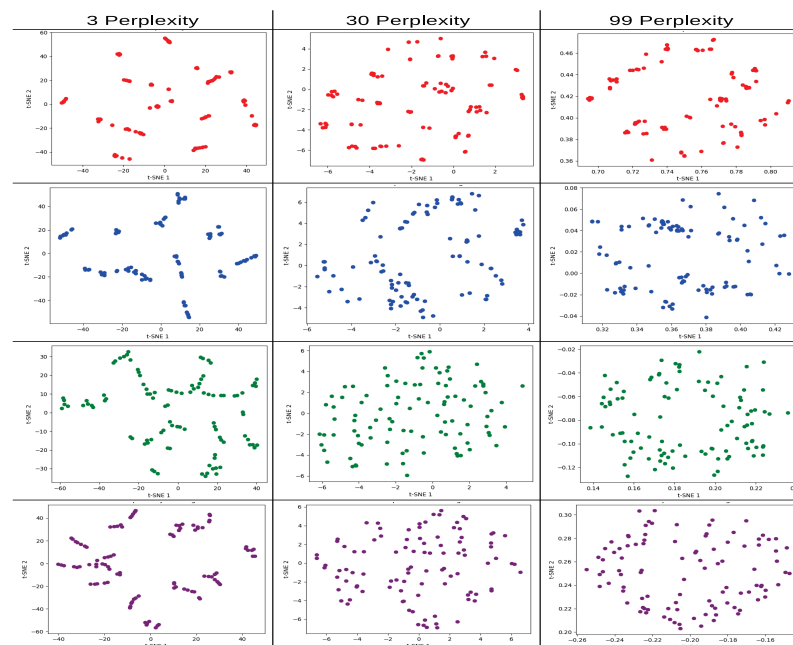


Figure A26. t-SNE individual graphs for 15n complete configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, and purple 6p parameter 2L entangled model.

Finally, for the 9p models, the graphical results are presented in Figure A27. At 3 and 30 perplexity, the patterns observed for the non-entangled (orange) and entangled (brown) models are similar to the ones observed in the 10n complete and 15n cyclic problems. At 99 perplexity, both the non-entangled and entangled models exhibit a tendency to concentrate more toward the sides of the t-SNE plane, creating a somewhat elliptical pattern that is not very distinct.

The KL-Divergence values for the paired t-SNE models in the 15n complete max-cut problem are presented in Table A20. The values at 3, 30, and 99 perplexity exhibit similar behaviors to the 15n cyclic problem, where the best KL value was generated by the 3p models and the worst values were obtained at 30 perplexity for the 9p models specifically. Furthermore, at 199 perplexity, the best KL values were reported, with all the models generating good values and the best among them being the 9p models.

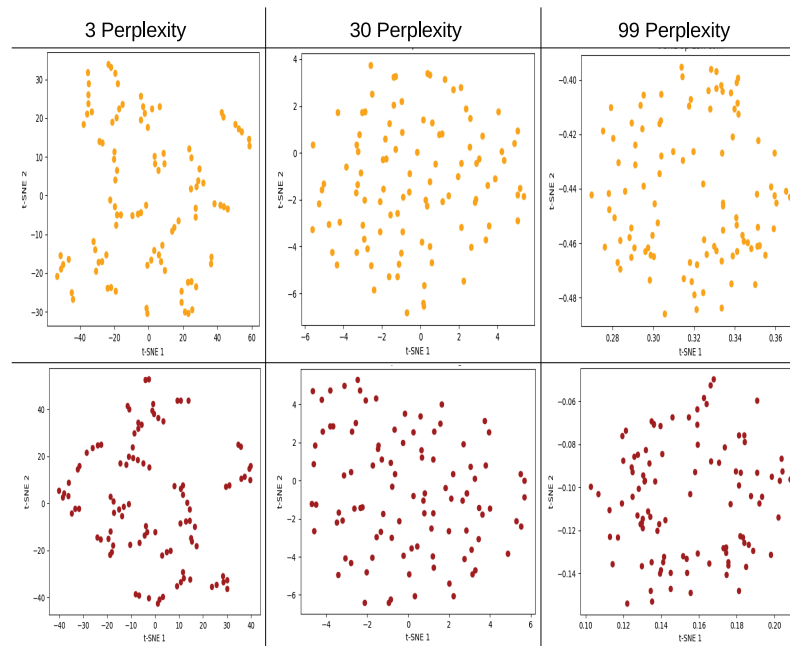


Figure A27. t-SNE individual graphs for 15n complete configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, and 99. Orange corresponds to the $9p$ parameter $3L$ non-entangled and brown $9p$ parameter $3L$ entangled.

Table A20. Pair KL-Divergence for 15n complete max-cut problem with different perplexity values, considering the $3p$ parameters (non-entangled and entangled), $6p$ parameters (non-entangled and entangled), and $9p$ parameters (non-entangled and entangled) models.

KL-Divergence				
Parameters	3 per	30 per	99 per	199 per
3 parameters	0.15160248	0.26059961	0.165535	0.00004839
6 parameters	0.6180442	0.75135112	0.34869462	0.0000503
9 parameters	0.943533	1.02061999	0.43895942	0.00003921

The paired t-SNE models for the 15n complete max-cut problem are presented in Figure A28. In the $3p$ models, non-entangled (red) and entangled (blue), the behavior observed at different perplexities is very similar between them, with no clear distribution even at 199 perplexity. This result differs from the patterns observed in the 10n complete problem and the 15n cyclic problem. Moving on to the $6p$ models, the patterns observed in the non-entangled (green) and entangled (purple) models are consistent with the previous graphs. The non-entangled model tends to be randomly scattered across the plane, while the entangled model shows more grouping behavior at 3, 30, and 99 perplexities. Only at 199 perplexity do the models distribute themselves at the sides of the plane, with the entangled model being more concentrated in certain areas of the distribution. Finally, for the $9p$ models, at 3, 30, and 99 perplexities, the non-entangled (orange) and entangled (brown) models exhibit similar distributions to the $6p$ models. The non-entangled model is more scattered, while the entangled model generates small group patterns in certain areas of the plane. At 199 perplexity, both models exhibit some sort of elliptical pattern previously observed in other problems, with the non-entangled model showing a more pronounced elliptic shape and the entangled model following the pattern but with less smoothness.

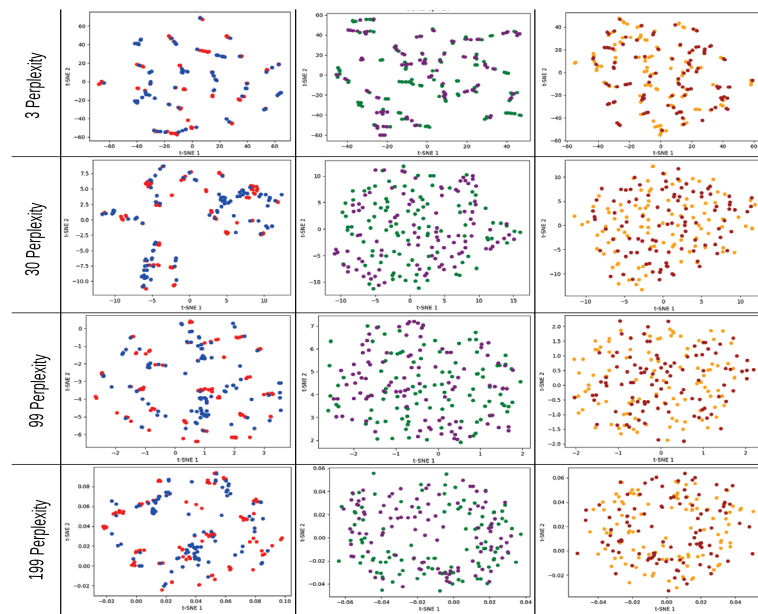


Figure A28. t-SNE pair graphs for 15n complete configuration max-cut problem solved using QAOA, with different perplexity values 3, 30, 99, and 199. Red corresponds to the 3p parameter 1L non-entangled, blue 3p parameter 1L entangled, green 6p parameter 2L non-entangled, purple 6p parameter 2L entangled, orange 9p parameter 3L non-entangled, and brown 9p parameter 3L entangled model.

References

- Farhi, E.; Goldstone, J.; Gutmann, S. A quantum approximate optimization algorithm. *arXiv* **2014**, arXiv:1411.4028.
- Sack, S.H.; Serbyn, M. Quantum annealing initialization of the quantum approximate optimization algorithm. *Quantum* **2021**, *5*, 491. [CrossRef]
- Alam, M.; Ash-Saki, A.; Ghosh, S. Accelerating quantum approximate optimization algorithm using machine learning. In Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 686–689.
- Shaydulin, R.; Safro, I.; Larson, J. Multistart methods for quantum approximate optimization. In Proceedings of the 2019 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 24–26 September 2019; pp. 1–8.
- Sud, J.; Hadfield, S.; Rieffel, E.; Tubman, N.; Hogg, T. A Parameter Setting Heuristic for the Quantum Alternating Operator Ansatz. *Bull. Am. Phys. Soc.* **2023**. [CrossRef]
- Koch, D.; Patel, S.; Wessing, L.; Alsing, P.M. Fundamentals in quantum algorithms: A tutorial series using Qiskit continued. *arXiv* **2020**, arXiv:2008.10647.
- Guerreschi, G.G.; Matsuura, A.Y. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Sci. Rep.* **2019**, *9*, 6903. [CrossRef] [PubMed]
- Marshall, J.; Wudarski, F.; Hadfield, S.; Hogg, T. Characterizing local noise in QAOA circuits. *IOP SciNotes* **2020**, *1*, 025208. [CrossRef]
- Bärttschi, A.; Eidenbenz, S. Grover mixers for QAOA: Shifting complexity from mixer design to state preparation. In Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, 12–16 October 2020; pp. 72–82.
- Galda, A.; Liu, X.; Lykov, D.; Alexeev, Y.; Safro, I. Transferability of optimal QAOA parameters between random graphs. In Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA, 17–22 October 2021; pp. 171–180.
- Akshay, V.; Rabinovich, D.; Campos, E.; Biamonte, J. Parameter concentrations in quantum approximate optimization. *Phys. Rev. A* **2021**, *104*, L010401. [CrossRef]
- Stechly, M.; Gao, L.; Yogendran, B.; Fontana, E.; Rudolph, M. Connecting the Hamiltonian structure to the QAOA energy and Fourier landscape structure. *arXiv* **2023**, arXiv:2305.13594.
- Rudolph, M.S.; Sim, S.; Raza, A.; Stechly, M.; McClean, J.R.; Anschuetz, E.R.; Serrano, L.; Perdomo-Ortiz, A. ORQVIZ: Visualizing High-Dimensional Landscapes in Variational Quantum Algorithms. *arXiv* **2021**, arXiv:2111.04695.
- Xue, C.; Chen, Z.Y.; Wu, Y.C.; Guo, G.P. Effects of quantum noise on quantum approximate optimization algorithm. *Chin. Phys. Lett.* **2021**, *38*, 030302. [CrossRef]
- Boulebnane, S.; Montanaro, A. Predicting parameters for the Quantum Approximate Optimization Algorithm for MAX-CUT from the infinite-size limit. *arXiv* **2021**, arXiv:2110.10685.

16. Wang, Z.; Hadfield, S.; Jiang, Z.; Rieffel, E.G. Quantum approximate optimization algorithm for MaxCut: A fermionic view. *Phys. Rev. A* **2018**, *97*, 022304. [CrossRef]
17. Willsch, M.; Willsch, D.; Jin, F.; De Raedt, H.; Michielsen, K. Benchmarking the quantum approximate optimization algorithm. *Quantum Inf. Process.* **2020**, *19*, 1–24. [CrossRef]
18. Yu, Y.; Cao, C.; Wang, X.B.; Shannon, N.; Joynt, R. Solution of SAT problems with the adaptive-bias quantum approximate optimization algorithm. *Phys. Rev. Res.* **2023**, *5*, 023147. [CrossRef]
19. Herrman, R.; Treffert, L.; Ostrowski, J.; Lotshaw, P.C.; Humble, T.S.; Siopsis, G. Impact of graph structures for QAOA on MaxCut. *Quantum Inf. Process.* **2021**, *20*, 289. [CrossRef]
20. Pagano, G.; Bapat, A.; Becker, P.; Collins, K.S.; De, A.; Hess, P.W.; Kaplan, H.B.; Kyprianidis, A.; Tan, W.L.; Baldwin, C.; et al. Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 25396–25401. [CrossRef] [PubMed]
21. Pelofske, E.; Baertschi, A.; Eidenbenz, S. *Short-Depth QAOA Circuits and Quantum Annealing on Higher-Order Ising Models*; Technical Report; Los Alamos National Laboratory (LANL): Los Alamos, NM, USA, 2023.
22. Moussa, C.; Wang, H.; Bäck, T.; Dunjko, V. Unsupervised strategies for identifying optimal parameters in Quantum Approximate Optimization Algorithm. *EPJ Quantum Technol.* **2022**, *9*, 11. [CrossRef]
23. Chen, Y.; Zhu, L.; Mayhall, N.J.; Barnes, E.; Economou, S.E. How much entanglement do quantum optimization algorithms require? In Proceedings of the Quantum 2.0. Optica Publishing Group, Boston, MA, USA, 13–16 June 2022; p. QM4A-2.
24. Sreedhar, R.; Vikstål, P.; Svensson, M.; Ask, A.; Johansson, G.; García-Álvarez, L. The Quantum Approximate Optimization Algorithm performance with low entanglement and high circuit depth. *arXiv* **2022**, arXiv:2207.03404.
25. Dupont, M.; Didier, N.; Hodson, M.J.; Moore, J.E.; Reagor, M.J. Entanglement perspective on the quantum approximate optimization algorithm. *Phys. Rev. A* **2022**, *106*, 022423. [CrossRef]
26. Sarmina, B. QAOA_SHC-RR. 2023. Available online: https://github.com/BrianSarmina/QAOA_SHC-RR (accessed on 12 September 2023).
27. Vidal, R.; Ma, Y.; Sastry, S.S.; Vidal, R.; Ma, Y.; Sastry, S.S. *Principal Component Analysis*; Springer: Berlin/Heidelberg, Germany, 2016.
28. Shlens, J. A tutorial on principal component analysis. *arXiv* **2014**, arXiv:1404.1100.
29. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
30. Van Der Maaten, L. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221–3245.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Quantum Honey Pots

Naya Nagy ^{1,*}, Marius Nagy ², Ghadeer Alazman ¹, Zahra Hawaidi ¹, Saja Mustafa Alsulaibikh ¹, Layla Alabbad ¹, Sadeem Alfaleh ¹ and Areej Aljuaid ¹

¹ Department of Networks and Communication, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia; gralazman@iau.edu.sa (G.A.); zahra.hawaidi@outlook.com (Z.H.); sj.mustafa.ah@gmail.com (S.M.A.); layla997alabbad@gmail.com (L.A.); sadeem.faleh@hotmail.com (S.A.); areejaljuaid@outlook.com (A.A.)

² College of Computer Engineering and Science, Prince Mohammad Bin Fahd University, Al-Khobar 31952, Saudi Arabia; mnagy@pmu.edu.sa

* Correspondence: nmnagy@iau.edu.sa

Abstract: Quantum computation offers unique properties that cannot be paralleled by conventional computers. In particular, *reading* qubits may change their state and thus signal the presence of an intruder. This paper develops a proof-of-concept for a quantum honeypot that allows the detection of intruders on reading. The idea is to place quantum sentinels within all resources offered within the honeypot. Additional to classical honeypots, honeypots with quantum sentinels can trace the reading activity of the intruder within any resource. Sentinels can be set to be either visible and accessible to the intruder or hidden and unknown to intruders. Catching the intruder using quantum sentinels has a low theoretical probability per sentinel, but the probability can be increased arbitrarily higher by adding more sentinels. The main contributions of this paper are that the monitoring of the intruder can be carried out at the level of the information unit, such as the bit, and quantum monitoring activity is fully hidden from the intruder. Practical experiments, as performed in this research, show that the error rate of quantum computers has to be considerably reduced before implementations of this concept are feasible.

Keywords: honeypot; post-quantum security; quantum security; quantum networks

Citation: Nagy, N.; Nagy, M.; Alazman, G.; Hawaidi Z.; Alsulaibikh, S.M.; Alabbad, L.; Alfaleh, S.; Aljuaid, A. Quantum Honey Pots. *Entropy* **2023**, *25*, 1461. <https://doi.org/10.3390/e25101461>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 14 August 2023

Revised: 10 October 2023

Accepted: 12 October 2023

Published: 18 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Honey pots are software-based security devices that operate within the general effort of protecting computing systems: servers, databases, networks, or, more generally, organizations. A honeypot is [1] intentionally constructed to be attacked, explored, and compromised. It is frequently used for detecting and dispersing unauthorized activities. Furthermore, its primary functionality is to investigate the conduct of attackers and to experience and pinpoint specific unknown attacks. The definition may vary between authors, a close to unifying definition places the value of a honeypot in its characteristics of being open to be inspected, attacked, and ultimately compromised [2]. The concept of honeypots moves the defence strategy from a passive paradigm to a proactive paradigm. Rather than building a strong defence for the sensitive system and waiting for an attacker to try out various attacks, the honeypot approach walks a totally different path by creating an alternate/fake environment that is offered to attackers. Attackers unknowingly try to exploit the fake environment and are misled to fake resources. Thus, honeypots are dedicated to attracting hackers by presenting services and open ports that are potentially vulnerable. The purpose is to monitor and analyze the activities of hackers and intruders in a way in which they do not know that they are being observed. Further, current attack methods and trends can be classified and studied in order to find the appropriate protection.

Quantum computation already has a mature theoretical background [3] with a fast emerging practical technology. In terms of theoretical results, quantum computational

results have been achieved by algorithms with asymptotic speed-ups over classical counterparts: Grover's search algorithm [4], Shor's factorization algorithm [5], and others. Nevertheless, arguably the most successful branch of quantum computation is quantum cryptography, with the promise of unconditionally secure communication protocols [6]. Quantum security protocols have been designed for key distribution [7], delayed secure decisions, and zero knowledge protocols. Note that, in all these security primitives, the quantum protocols plays a passive role in the protection of a system. Our paper, by contrast, is the first to propose an *active* quantum protocol of the type of a honeypot. It will be shown that, by employing quantum networks and quantum communication systems, the honeypot can be enhanced with additional monitoring capabilities, while fully hiding its presence.

The idea in this paper is to use quantum computation techniques to better hide the fact that the services offered to the attackers are fake, as well as to monitor the intruder without being detected. This is achieved through quantum sentinels. Quantum sentinels is a new concept proposed for the first time in this paper. We define two types of sentinels: *positional sentinels* with recognizable positions and *hidden sentinels*, which are not detectable from the outside. In the first case, the intruder is able to directly read and otherwise act on the *positional* sentinel. The position of the sentinel, within the array of qubits, may be secret or public, yet by direct reading, its existence is still visible to the outside world. In the second case, the *hidden* sentinel cannot be seen or read from outside and is thus accessible only to the honeypot system. The hidden sentinel relies on the quantum Fourier transform to connect to the luring datum qubit. The quantum sentinels flag the presence of an intruder when the intruder *reads* some information. The information may be part of a file, an address, the content of memory or a hard disk sector.

In terms of technological readiness for the commercial implementation of a quantum honeypot scheme, consider that a mere decade ago, quantum computers were technologically questionable [8], whereas the situation now shows that quantum supremacy has been affirmed to be achieved by several academic and commercial sources [9,10].

A honeypot may present itself as an entire system, such as a node in the internet, that looks as if it contains useful information and data, but, in reality, is meant only to lure unlawful activities. The interface to the outside world is there, but in the background there is no useful application. Any honeypot consists of two essential elements: decoy and captor. The decoy lures the attacker by offering information system resources, whether physical or virtual. The captor is the part of the system that actually inspects and records the activity of the intruder. It acts in detecting the intruder, responding to the requests, and profiling the attacker. Our proposed quantum sentinels are in the category of the Captor. Additionally to the classical captor, these sentinels can be fully hidden.

There are three categories of honeypots depending on the information system resources that are provided to the attacker: low-interaction honeypot, medium-interaction honeypot, and high-interaction honeypot [11]. As the names suggest, the three levels allow for increasing penetration capability. High-interaction honeypots define an entire operating system for tampering. Quantum sentinels can be added to all elements of a functional operating system, making them suitable for high-interaction honeypots. Thus, with quantum sentinels, the entire activity of the attacker can be monitored to a more detailed level. In fact any reading of information can be monitored both in terms of the actual reading action as well as the time of the action.

This paper builds on the idea that *any* activity within the honeypot is categorized as malicious [11]. With quantum sentinels added to honeypots, the extent of malicious activity can be detailed to the next level, where hackers may be caught on any particular bit they access for reading. Additionally, the honeypot exhibits a better hiding effect. Note that the purpose of the design in this paper is the recognition of the intruder's behavior and that it does not deal directly with allowing the intruder to compromise the system.

The rest of the paper is organized as follows. Section 2 describes the quantum properties used in the honeypot algorithms as well as the quantum network setting. Section 3 presents the honeypot quantum algorithms that capture the activity of the intruder on the

simple reading of any storage medium. The difference between *positional sentinels* and *hidden sentinels* is also described here. Section 4 shows the behavior of the algorithms as implemented on a real quantum computer, IBMQX, as a proof of concept of how quantum sentinels work. Section 5 concludes the paper.

2. Quantum Properties

Discussions about quantum computation revolve not only around quantum supremacy [12], but also around quantum network communication. Quantum network communication algorithms are studied for various problems. García-Cobo [13] defines a quantum algorithm for key distribution within a large quantum network. The network experiments have been done with simulations over a known geographical territory in Castilla using quantum repeaters to propagate quantum signals.

In our setting, we suppose to have a network with quantum connections and the devices connected to the network are also able to do quantum computations on qubits, at least in part of the memory.

The quantum honeypot connects to the outside worlds through quantum connection. Users, such as fake users and hackers, communicate with the honeypot via quantum channels. Quantum channels allow the bidirectional transmission of messages. In order to put no limitation on the amount of communication, we consider these messages to be arbitrarily large. Note that this assumption is common in quantum algorithms, such as quantum key distribution [7,14], where the analysis of the algorithm allows the size of the message transmitted from one partner to the other to be arbitrarily long.

2.1. Measurement and State Collapse

A qubit in Dirac's notation [3] is a superposition of the base vectors $|0\rangle$ and $|1\rangle$.


$$q = \alpha|0\rangle + \beta|1\rangle,$$

where the coefficients α, β and the amplitudes are complex numbers and the vector is of a unitary norm, i.e., $\sqrt{|\alpha|^2 + |\beta|^2} = 1$. There are two specific balanced superpositions, namely $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. They will play a role in the algorithms below.

When qubits are measured, they are measured on some basis. The simplest measurement base is the computational base, with the base vectors $|0\rangle$ and $|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. This base is not unique. Another common measurement base is the Hadamard base, with base vectors $|+\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = H|1\rangle$. Both these bases are orthonormal bases. In any case, when an arbitrary qubit is measured, the state of the qubit collapses to one of the base vectors. Thus, an arbitrary qubit $q = \alpha|0\rangle + \beta|1\rangle$ can be measured in multiple measurement bases. When q is measured in the computational basis, it collapses either to $|0\rangle$ with probability α^2 , or to $|1\rangle$ with probability β^2 . Again, when q is measured in the Hadamard basis, it collapses either to $|+\rangle$ or to $|-\rangle$. The probabilities of collapse can be seen from rewriting the qubit as $q = \alpha|0\rangle + \beta|1\rangle = \frac{\alpha+\beta}{\sqrt{2}} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{\alpha-\beta}{\sqrt{2}} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. q is measured as $|+\rangle$ with probability $\frac{(\alpha+\beta)^2}{2}$ and as $|-\rangle$ with probability $\frac{(\alpha-\beta)^2}{2}$.

2.2. Qubits and Quantum Gates

Quantum gates can apply on one or more qubits. The condition on quantum gates is that they be reversible. Therefore, all quantum gates have the same number of inputs and outputs. Here is a list of the gates used in our algorithms.

- A The **Hadamard gate**  rotates the states $|0\rangle$ and $|1\rangle$ into $|+\rangle$ and $|-\rangle$, respectively. It contributes towards a universal gate set on a quantum computer as the single quantum gate needed in addition to a universal gate set for classical computation. The Hadamard gate is useful for creating balanced superpositions. The reverse is also

true, namely that a Hadamard gate applied to a balanced superposition brings the qubit to the respective base state. The Hadamard gate is its own inverse.

- B The **NOT gate** \oplus is also known as the Pauli-X gate and, in this form, is also applied on a single qubit. The $|0\rangle$ state flips to $|1\rangle$ and vice versa. As shown below, it is represented by the Pauli matrix:

$$X = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- C The **Controlled Phase Shift gate** is a two qubit gate, built from a simple phase shift gate.

The simple phase shift gate (R_z) operates on a single qubit. It rotates the qubit around the z axis of the Bloch sphere [3]. Thus, the gate changes the phase and the angle of the $|0\rangle$ and $|1\rangle$, but not the respective percentages of the two within the superposition.

The controlled phase shift gate, CR_z , has an additional control qubit R_z . This gate is a two qubit gate. The control modifier determines whether the shift is applied or not on the original data-qubit. If the control is $|1\rangle$, the gate is active, and it is inactive for a control equal to $|0\rangle$.

In the following algorithms, the phase shift gate always has the rotation angle of either $\frac{\pi}{2}$, or its opposite $-\frac{\pi}{2}$. Consider a few applications of the transformation, as they appear in this study. The rotation of $\pm\frac{\pi}{2}$ on the computational base vectors has no effect, $R_z^{\pm\frac{\pi}{2}}|0\rangle = |0\rangle$ and $R_z^{\pm\frac{\pi}{2}}|1\rangle = |1\rangle$. On the Hadamard base vectors, the $\frac{\pi}{2}$ rotation changes the phase, $R_z^{\frac{\pi}{2}}|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$ and $R_z^{\frac{\pi}{2}}|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$. A double application of $R_z^{\frac{\pi}{2}}$ moves from one Hadamard base vector to the other, $R_z^{\frac{\pi}{2}}R_z^{\frac{\pi}{2}}|+\rangle = |-\rangle$ and $R_z^{\frac{\pi}{2}}R_z^{\frac{\pi}{2}}|-\rangle = |+\rangle$. In the case of a rotation with $-\frac{\pi}{2}$, the sign before the imaginary term i is reversed.

3. Quantum Sentinels

Sentinels in computer science are entities, such as variables, that block the access of a program to a certain area or that flag a state of error/emergency within a program. In the case of a quantum sentinel, as described here, we use the latter situation. Quantum sentinels are quantum entities, primarily qubits, that flag a state of emergency. Within a honeypot, a quantum sentinel marks the presence of an intruder.

The physical principle on which quantum sentinels function is the collapse of superposition at measurement. A collapse of superposition can then potentially be detected by a legitimate checking system.

We define two types of quantum sentinels:

1. **Positional sentinels**, which are visible to the intruder, though their quantum state is unknown.
2. **Hidden sentinels**, which are hidden from the user. In this case, both the quantum state and the operation of the sentinel remain unknown to the intruder.

Quantum sentinels capture the action of reading. As such, they can be placed in any part of a computer system where the reading of information is possible. This can be viewed as practical devices that carry information: hard disks, random access memory, network card information, video card memory, external memory-carrying devices, etc. Alternatively, quantum sentinels can be viewed as logically positioned in places holding key information: operating system settings, boot sectors of hard disks, meta data storage files, log and important history files, configuration files, environment variable settings, keyword and password locations, sensitive information locations, or crucial data paths. The two types of quantum sentinels presented here have different behaviors and, therefore, can be employed with somehow different purposes. The positional sentinel marks an important information carrier, such as a sensitive parameter setting for an application or an operating system. In this case, the sentinel may be visible, and the intruder may

know of the presence of sentinels, while being unable to avoid reading them. The hidden sentinel has a more insidious capability of remaining unnoticed for the entire activity of the intruder. In this case, a longer observation of the intruder is possible; it is possible to trace the intruder in terms of actions, using their target and timeline to get the entire action plan of the intrusion. Hidden sentinels are more costly, as they involve a small circuit for each quantum sentinel, they should be employed more sparingly according to needs, whereas positional sentinels are less involved.

The idea of quantum sentinels comes from the fact that an unknown quantum state, when read, collapses along the measurement basis. Thus, using two measurement bases, a qubit read by an intruder in the wrong basis changes its original state and this state change can be detected by the honeypot server, with a certain probability. This property has been used in public key distribution [7]. Thus, consider the two measurement bases:

1. The Computational Basis, with the base vectors $|0\rangle$ and $|1\rangle$
2. The Hadamard Basis, with the base vectors $|+\rangle$ and $|-\rangle$

Suppose that the only allowed states of a qubit are the four base vectors above: $|0\rangle$, $|1\rangle$, $H|0\rangle$, and $H|1\rangle$.

The sentinel capacity comes from the qubit’s unknown state to the intruder. The interplay between computational and Hadamard bases makes the qubit’s state vulnerable to changes when accessed by an unknowing intruder.

3.1. Positional Sentinels

Positional sentinels refer to sentinels that are controlled by the server and that the client can see.

A positional quantum sentinel is a qubit in one of the four states: $|0\rangle$, $|1\rangle$, $|+\rangle$, or $|-\rangle$. When read, a sentinel qubit may keep its exact state or may change the state to align with the reading basis, computational or Hadamard.

Depending on the value of the qubit, reading the qubit may or may not collapse the qubit to another value and thus change, or not change, its state. A qubit of state $|0\rangle$ or $|1\rangle$ does not change when measured in the computational basis, but when measured in the Hadamard basis, it collapses to $|+\rangle$ or $|-\rangle$. The exact reverse happens for a qubit in state $|+\rangle$ or $|-\rangle$; when measured in the Hadamard basis, the quantum state remains unchanged, and when measured in the computational basis, the qubit collapses to $|0\rangle$ or $|1\rangle$. Thus, a sentinel has to be measured in the correct basis in order to remain unchanged. Consider that a valid user is knowledgeable about the inherent basis of the sentinel qubit.

This means that an intruder, not knowing the basis of the sentinel, risks changing its state by inadvertent reading. Table 1 synthesizes all the possibilities of a sentinel value versus the reading options of the legitimate user and the intruder. The conclusion is that a sentinel catches an intruder with a probability of $\frac{1}{4}$. To improve the probability overall, a simple addition of sentinels in desired positions increases the probability to any desired value. One positional sentinel remains undetected with probability $1 - \frac{1}{4} = \frac{3}{4}$. Thus, the detection rate for m positional sentinels is

$$p_{\text{positional}}^m = 1 - \left(\frac{3}{4}\right)^m.$$

To exemplify this growth, eight sentinel qubits catch an intruder with probability $1 - \left(\frac{75}{100}\right)^8 = 89.9\%$ and the probability grows exponentially with the number of sentinels.

The server can check the state of a sentinel by reading it in the correct basis. The drawback of positional sentinels is that they are exposed to the intruder. The secrecy of the position of the sentinels may be part of the honeypot concept, but the sentinel itself is part of the information read by the intruder. Thus, as the value of the sentinel depends on the reading basis, its value may affect the meaning of the information read by the intruder and, thus, reveal the presence of the sentinel itself. Nevertheless, this is not all the capability of quantum sentinels, as sentinels themselves can remain fully hidden from the intruder.

Table 1. Positional sentinels and their behavior at reading.

Sentinel Value	Correct Reading Basis	User		Server	Probability to Catch
		Reading Type	Outcome	Measurement	
0⟩	Computational	honest, Computational	0⟩	0⟩	Not applicable
		intruder, Computational	0⟩	0⟩	
		intruder, Hadamard	+⟩ or −⟩	0⟩ or 1⟩	25%
1⟩	Computational	honest, Computational	1⟩	1⟩	Not applicable
		intruder, Computational	1⟩	1⟩	
		intruder, Hadamard	−⟩ or +⟩	1⟩ or 0⟩	25%
+⟩	Hadamard	honest, Hadamard	+⟩	+⟩	Not applicable
		intruder, Hadamard	+⟩	+⟩	
		intruder, Computation	0⟩ or 1⟩	+⟩ or −⟩	25%
−⟩	Hadamard	honest, Hadamard	−⟩	−⟩	Not applicable
		intruder, Hadamard	−⟩	−⟩	
		intruder, Computation	1⟩ or 0⟩	−⟩ or +⟩	25%

3.2. Hidden Sentinels

Hidden sentinels are sentinels controlled by the server that the intruder cannot see. They are physically not addressable by the user or intruder. A hidden quantum sentinel is a qubit that is never exposed to the user, but is connected through entanglement to a datum-qubit that is available to the intruder to read. The main idea is that, when an intruder reads the available datum-qubit in a wrong basis, the sentinel’s state changes to a value consistent with the measurement of the datum-qubit and this change is detectable by the honeypot server.

The circuit involved for every hidden sentinel is simple, using two gates: a Hadamard gate and a phase shift gate. It is a portion of the quantum Fourier transform.

The Quantum Fourier Transform [15] allows the phase of a qubit to be changed, that is, the qubit is rotated around the Oz axis. The value of the rotation is given by a series of control qubits in such a way that the impact of each successive control qubit is half the angle rotation effect of the previous one. In our case, we are interested in only one control qubit, see Figure 1. The top qubit is the datum-qubit, which acts as the control qubit to the gates. This is also the qubit that the intruder acts on. The phase shift gate, in purple, acts on the second qubit. The second qubit is the sentinel. If the datum-qubit is one, then it has an effect on the sentinel qubit, as it injects a rotation of the phase with $\frac{\pi}{2}$. The sentinel undergoes two such gates in reverse. The first gate prepares the sentinel before the honeypot is offered to attack. The second gate is used for checking the state of the sentinel. Note that each time a sentinel is checked, the sentinel is destroyed. This means that, to re-activate a hidden sentinel, the sentinel has to undergo preparation again.

The behavior of the hidden sentinel is also based on the intruder not knowing the correct reading basis of the data qubit. If lucky, the intruder will not be caught. If unlucky, the intruder will be caught with a chance of $\frac{1}{4}$. As the intruder may or may not be lucky with equal probability, the overall theoretical probability to catch an intruder with a hidden sentinel is $\frac{1}{8}$. This is half of the probability of a positional sentinel. Thus, there is a drawback of using hidden sentinels, in that the probability is lower.

The following formulas describe the situations in detail. The ensemble of two qubits is always written with the datum first and the sentinel second, $|D\rangle|S\rangle = q_1q_0$. The preparation of the sentinel, as shown in Figure 1, can be described by the transformation $R_z^{\frac{\pi}{2}}(I_2 \otimes H)$.

The server’s checking of the sentinel is similar, namely $(I_2 \otimes H)R_z^{-\frac{\pi}{2}}$. Additionally, if the intruder is unlucky, the data qubit may undergo a Hadamard transformation, as imposed by the intruder; this is the transformation $H \otimes I_2$.

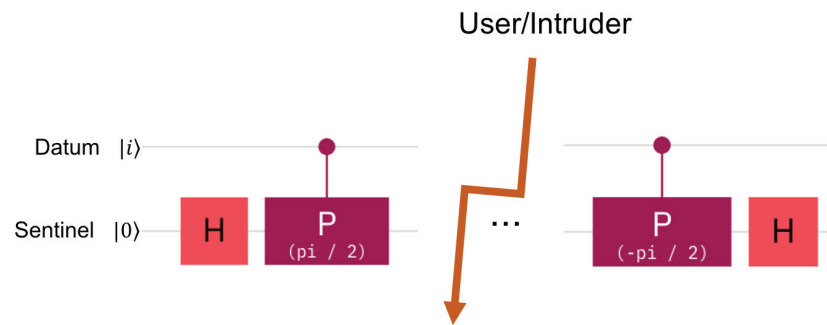


Figure 1. The hidden sentinel is the second qubit in the figure. It is acted on by the datum sentinel via the control of phase shift gates. The middle of the figure shows the area and time when the datum-qubit is exposed to the user.

Consider **the data qubit to be $|0\rangle$** . In this case, the initial state of the system is $|D\rangle|S\rangle = |0\rangle|0\rangle$. If the intruder is lucky and uses the computational basis himself, the following transformation happens to the system.

$$\begin{aligned}
 \text{result-lucky} &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (I_2 \otimes I_2) R_z^{\frac{\pi}{2}} (I_2 \otimes H) |0\rangle|0\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (I_2 \otimes I_2) R_z^{\frac{\pi}{2}} |0\rangle|+\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} |0\rangle|+\rangle \\
 &= (I_2 \otimes H) |0\rangle|+\rangle = |0\rangle|0\rangle
 \end{aligned} \tag{1}$$

The operation in red represents the action of the intruder, which, in this case, is the identity transformation that is no action at all. This is because this measurement is aligned with the state of the qubit and does not change the state of the system. The sentinel is found in its original value $|0\rangle$. The server checker concludes that no intrusion happened.

Now consider that the intruder makes the mistake and reads in the Hadamard basis.

$$\begin{aligned}
 \text{result-unlucky} &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (H \otimes I_2) R_z^{\frac{\pi}{2}} (I_2 \otimes H) |0\rangle|0\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (H \otimes I_2) |0\rangle|+\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} |+\rangle|+\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} \left(\frac{1}{\sqrt{2}}|0\rangle|+\rangle + \frac{1}{\sqrt{2}}|1\rangle|+\rangle \right) \\
 &= (I_2 \otimes H) \left(\frac{1}{\sqrt{2}}|+\rangle|0\rangle + \frac{1}{2}|1\rangle(|0\rangle + i|1\rangle) \right)
 \end{aligned} \tag{2}$$

In this case, when the sentinel is measured by the checker in the computational basis, the theoretical probability to measure $|0\rangle$ is $\frac{3}{4}$ and the probability to measure a $|1\rangle$ is $\frac{1}{4}$. If the checker sees a 1, this value indicates the presence of the intruder.

Overall, the probability of the intruder being caught by the checker, in this case, is $p = \frac{1}{2} * \frac{1}{4} = \frac{1}{8}$. Though the calculations vary from case to case, the overall result is shown to be the same. When **the data qubit is $|1\rangle$** , the formulas are very similar with some difference in the signs.

Consider **the data qubit to be $|+\rangle$** . In this case, the initial state of the system is $|D\rangle|S\rangle = |+\rangle|0\rangle$. If the intruder is lucky and uses the Hadamard basis of measurement, then the measurement is aligned with the actual qubit value and its state is undisturbed.

This means that no change is applied to the data qubit, and the measurement is formally described by $I_2 \otimes I_2$. The system is transformed as follows.

$$\begin{aligned}
 \text{result-lucky} &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (I_2 \otimes I_2) R_z^{\frac{\pi}{2}} (I_2 \otimes H) |+\rangle|0\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (I_2 \otimes I_2) R_z^{\frac{\pi}{2}} |+\rangle|+\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (I_2 \otimes I_2) R_z^{\frac{\pi}{2}} \left(\frac{1}{\sqrt{2}}|0\rangle|+\rangle + \frac{1}{\sqrt{2}}|1\rangle|+\rangle \right) \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (I_2 \otimes I_2) \frac{1}{\sqrt{2}} \left(|0\rangle|+\rangle + |1\rangle \frac{|0\rangle + i|1\rangle}{\sqrt{2}} \right) \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} \frac{1}{\sqrt{2}} \left(|0\rangle|+\rangle + |1\rangle \frac{|0\rangle + i|1\rangle}{\sqrt{2}} \right) \\
 &= (I_2 \otimes H) \frac{1}{\sqrt{2}} \left(|0\rangle|+\rangle + |1\rangle|+\rangle \right) = |+\rangle|0\rangle \tag{3}
 \end{aligned}$$

The sentinel preserves its original value $|0\rangle$ and no intruder can be detected.

Nevertheless, when the intruder mistakenly measures in the computational basis, the sentinel is changed. Note that, in this case, the intruder actually affects the $|D\rangle$ qubit and its state is collapsed along the computational basis, which is equivalent to applying a Hadamard gate in the middle.

$$\begin{aligned}
 \text{result-unlucky} &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (H \otimes I_2) R_z^{\frac{\pi}{2}} (I_2 \otimes H) |+\rangle|0\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (H \otimes I_2) R_z^{\frac{\pi}{2}} |+\rangle|+\rangle \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (H \otimes I_2) R_z^{\frac{\pi}{2}} \left(\frac{1}{\sqrt{2}}|0\rangle|+\rangle + \frac{1}{\sqrt{2}}|1\rangle|+\rangle \right) \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} (H \otimes I_2) \frac{1}{\sqrt{2}} \left(|0\rangle|+\rangle + |1\rangle \frac{|0\rangle + i|1\rangle}{\sqrt{2}} \right) \\
 &= (I_2 \otimes H)R_z^{-\frac{\pi}{2}} \frac{1}{\sqrt{2}} \left(|+\rangle|+\rangle + |-\rangle \frac{|0\rangle + i|1\rangle}{\sqrt{2}} \right) \\
 &= (I_2 \otimes H) \frac{1}{2} \left(|0\rangle|+\rangle + |1\rangle \frac{|0\rangle + i|1\rangle}{\sqrt{2}} + |0\rangle \frac{|0\rangle + i|1\rangle}{\sqrt{2}} + |1\rangle|+\rangle \right) \\
 &= \frac{1}{2} \left(|0\rangle|0\rangle + |1\rangle \frac{|0\rangle - i|1\rangle}{\sqrt{2}} + |0\rangle \frac{|0\rangle - i|1\rangle}{\sqrt{2}} + |1\rangle|0\rangle \right) \tag{4}
 \end{aligned}$$

From the formula, it can be seen that the probability to get 1 when the checker measures is, again, $\frac{1}{4}$.

Considering that the intruder is lucky or unlucky with equal probability, the checker catches the intruder with a probability of $\frac{1}{8}$. A similar result can be obtained for a **data qubit of $|-\rangle$** .

The overall conclusion is that a hidden sentinel can catch the intruder with a probability of $\frac{1}{8}$. This probability is lower than in the case of positional sentinels, but, again, can be increased arbitrarily higher by adding more sentinels in the area of interest. To evaluate the detection rate of m hidden sentinels, note one hidden sentinel remains undetected with probability $1 - \frac{1}{8} = \frac{7}{8}$. m hidden sentinels remain undetected with probability $(\frac{7}{8})^m$. Thus, the detection rate for m hidden sentinels can be computed using the formula

$$p_{hidden}^m = 1 - \left(\frac{7}{8}\right)^m.$$

To get a perception of how the detection rate changes exponentially with the value m , consider that for only $m = 8$ sentinels, the detection rate is already $p_{hidden}^8 = 1 - (\frac{7}{8})^8 = 66\%$.

There is an additional justification on using controlled phase shift gates for setting up hidden sentinels. Note that the data qubit has to explicitly act on the hidden sentinel,

whenever touched by a user or intruder. This *must* be carried out by an explicit two-qubit gate. A simple Bell state type entanglement of the data and the sentinel cannot work here. The reason is that entanglement cannot be used as an information carrier. Otherwise, information could be transmitted instantaneously, rather than, at most, the speed of light, which is the tenet of today’s physics.

It is now the time to check the experiments and how they fit the theoretical calculations.

4. Quantum Implementation and Experiments

Consider that the quantum server opens a honeypot that includes quantum sentinels. Thus, the regular services offered by the honeypot are peppered with quantum sentinels, within non-volatile, volatile memory locations or any other bit/qubit arrays. We have defined two types of sentinels, positional and hidden, and they have different characteristics in terms of visibility to the user and efficiency in catching the intruder.

All experiments are implemented using IBM Quantum Experience [16] with a real quantum processor, and both types of sentinels were implemented. The results for each quantum circuit are collected from 1024 runs. The results show a general alignment to the theoretical expectations, except that they are unreliable to a certain degree and sometimes give spurious results.

Experiments were performed for both legal users and intruders. A legal user knows the state of the sentinels and acts accordingly. The intruder takes the best guess, which, in the case of quantum sentinels, is simple random guessing.

4.1. Experiments with Positional Sentinels

In the case of positional sentinels, each qubit can play the role of a sentinel. The availability of qubits gives the size of the experiment, namely, to four sentinels. In each experiment, we have two players, the server and the client. The server prepares the honeypot sentinels and the client exploits them. In all figures, the red rectangle pertains to the server and the green rectangle pertains to the client. Measurement is also part of the server activity.

The first type of experiment is that the server generates all possible types of positional sentinels. Recall that there are four types of sentinels according to the four base vectors of the computational and Hadamard measurement bases, $|0\rangle$, $|1\rangle$, $|+\rangle$, and $|-\rangle$, in some arbitrary order. Thus, the server side is fixed. It remains for us to define the behavior of the client. We show two directions. The first direction defines a circuit with a legal user, and the second direction contains two experiments where the user is an intruder.

4.1.1. Positional Sentinels with a Legal User

A legal user knows the setting of the sentinels and reads them correctly, see Figure 2. Note that the sentinels have been prepared as $q_0 = |-\rangle$, $q_1 = |0\rangle$, $q_2 = |+\rangle$, and $q_3 = |1\rangle$. The legal users measures correctly, namely q_0, q_2 in the Hadamard Basis, and q_1, q_3 in the computational basis.

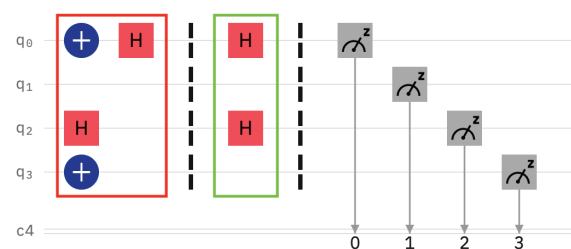


Figure 2. The legal user measures the sentinels in the correct bases.

The results obtained after running the circuit (see Figure 3) show that, indeed, the legal user is correctly classified as such, 93.5% times. Nevertheless, the false negatives are not negligible at 6.5%.

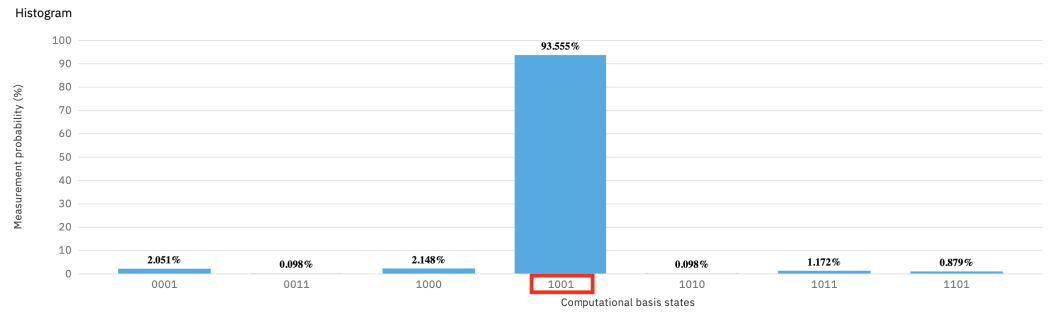


Figure 3. The measurement probability of Figure 2, which has four sentinels, and the user is legal.

4.1.2. Positional Sentinels with an Intruder as User

The intruder does not know the settings of the sentinels and, therefore, has no other options than to randomly choose the reading bases. The next two experiments show the intruder with two different choices. The randomization options of the client actions were carried out on a randomization tool.

For the first “intruder” experiment, the server has prepared the four sentinels as $q_0 = |0\rangle$, $q_1 = |+\rangle$, $q_2 = |1\rangle$, and $q_3 = |-\rangle$ (see Figure 4). It can be seen in the same figure that the client happened to read the sentinels in the following bases: computational, Hadamard, Hadamard, and Hadamard. Thus, the only sentinel that is wrongly read is qubit q_2 . Here, the server prepared the qubit in the computational basis, but the client used the Hadamard basis for reading instead. We expect the intruder client to be caught with a probability of $\frac{1}{2}$.

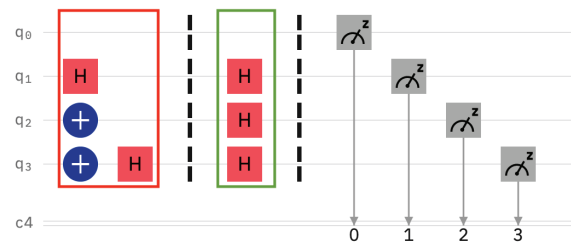


Figure 4. Experiment with four positional sentinels. The intruder’s behavior is random, as the state of the sentinels is not known to the user. In this particular case, the intruder makes a mistake on q_2 and, therefore, the detection probability is theoretically $\frac{1}{2}$.

The measurement that the server expects from a legal user is 1100. Theoretically, when the server measures 1000, this signals the presence of the intruder. Figure 5 shows the actual measured probabilities. It can be seen that the intruder catching measurement for 1000 is 44.727%, which is significantly different from 50%. The problem is that the quantum computer produces spurious results as well. This is the case for all the values different from 1100 and 1000. Because the server expects exactly 1100 from the legal user, it means that all spurious results contribute to catching the user. Thus, the measured percentage of positively signaling the user becomes $100 - 50.879\% = 49.121\%$, which is very close indeed to the theoretical expectation.

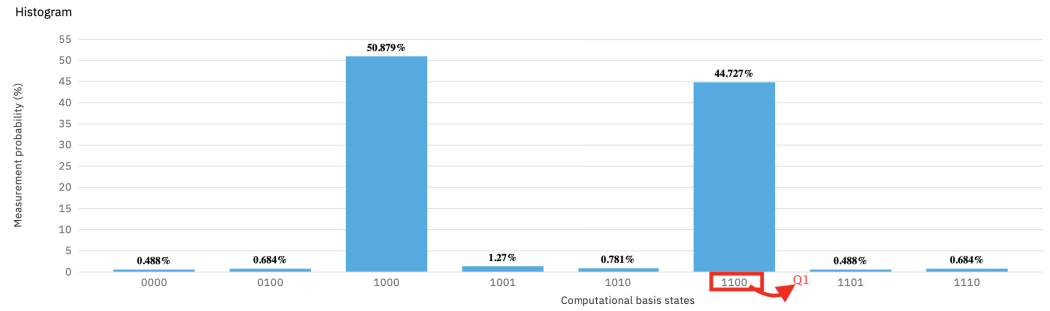


Figure 5. The measurement probability of Figure 4, which contains four positional sentinels, and the intruder misses one.

For the second “intruder” experiment (see Figure 6), the intruder makes three mistakes, on q_0 , q_1 , and q_3 , respectively. The probabilities of the server to measure the expected 1100 is theoretically $\frac{1}{2}^3 = 12.5\%$. All other binary measurements reveal the presence of the intruder, which, again, would theoretically be $100\% - 12.5\% = 87.5\%$.



Figure 6. Four positional sentinels are set to all possible values. The experiment shows the option where the intruder is lucky on only one qubit, namely q_2 .

The practical measurement results, as shown in Figure 7, detect the intruder with $1 - 11.426\% = 88.574\%$ probability. Again, we see a slight deviation from the theoretical expectation, but within workable limits.

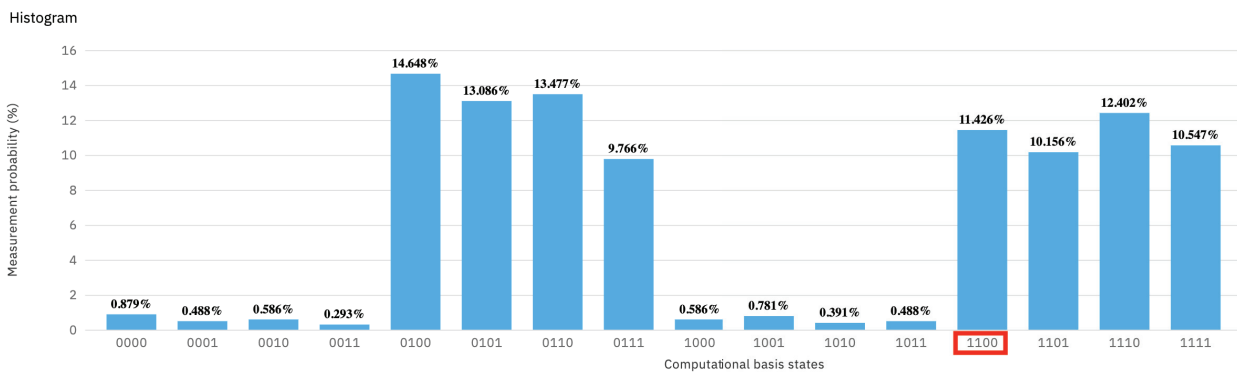


Figure 7. The measurement probability of Figure 6 with four positional sentinels and the intruder missing three of the sentinels. The result in red refers to the probability of the intruder to escape detection.

The conclusion may be that the implementation of positional sentinels seems to be close to feasibility in practical cases. Some false positives and some false negatives have to be contended with.

4.2. Hidden Sentinels

In the case of hidden sentinels, the data qubit is considered to be in one of the four basic states, $|0\rangle$, $|1\rangle$, $|+\rangle$, or $|-\rangle$. Note that the data qubit does not have an arbitrary value, but has to follow one of these choices. Nevertheless, the user can read it freely, oblivious to

the presence or absence of the sentinel, as the sentinel itself is *another* qubit. The intervention of an intruder can be tested by applying the quantum Fourier transform twice: directly and in reverse. The datum qubit serves as the control to the phase shift rotation gates, CR_z with rotation $\frac{\pi}{2}$ and $-\frac{\pi}{2}$ and the sentinel qubit is the qubit that the gates act on. The circuit that implements this transform is shown in Figure 8. The datum, q_0 , is set to an initial state and then at the end reset to $|0\rangle$. This state is not fixed as described above. The middle of the circuit shows the action of the user, encircled in a golden rectangle. This action will also be variable, depending on the intruder’s choice. The meaning that this circuit offers is that, if the sentinel, q_1 , is measured to the value 0, then the conclusion is that the user is legal and if the value is 1 then the sentinel signals an intruder. As hidden sentinel experiments need two qubits for each experiment, the circuit represents one such sentinel setting.

Figure 8 shows the circuit for a legal user or a lucky intruder, whereas Figure 9 shows the circuit with an unlucky intruder. When run, the circuit with a legal user shows an approximate 10% of false positives, whereas the circuit with the illegal user has a similar deviation from the theoretical expectation. The question remains: how does the error of the quantum computer scale in the case of several sentinels?

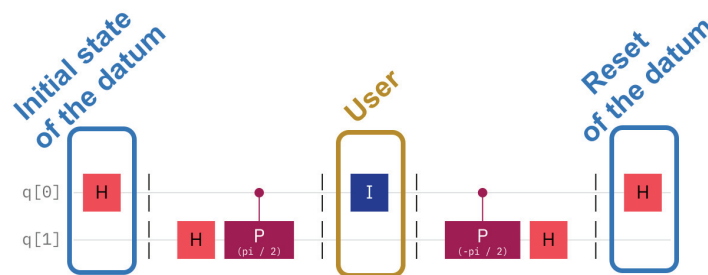


Figure 8. Legal user reading a datum qubit with a hidden sentinel. The same circuit applies to a lucky intruder. The user does not disturb the state of the datum qubit.

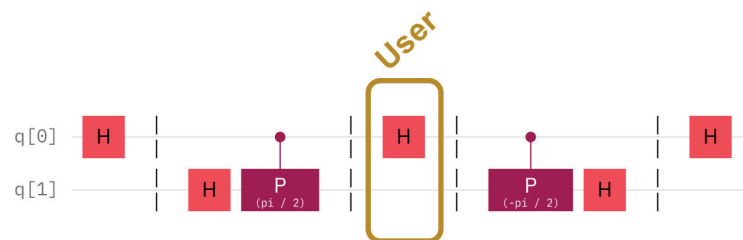


Figure 9. Unlucky intruder reading a quantum qubit with a hidden sentinel. In the case of an unlucky intruder, an extra Hadamard gate on the datum qubit disturbs the hidden sentinel.

4.2.1. Errors on Hidden Sentinels

The problem with available quantum computers today is that their error rate is still prohibitively high. In a real honeypot with quantum sentinels, the number of sentinels should be peppered over all resources, bringing them, in number terms, to a fraction of the entire address space. Nevertheless, the erroneous quantum measurements make this scenario unrealistic, as the following example will show.

Consider a sample circuit with two hidden sentinels (see Figure 10), such that the intruder has been unlucky on both data qubits, that is, the intruder has been consistently unlucky. Qubits q_0 and q_1 form the first hidden sentinel circuit, such that q_0 is the datum qubit and q_1 is the sentinel. It can be seen that the datum qubit, q_0 , is set to $H|0\rangle$. The setting of the datum qubit is not important for the success of the circuit. The important characteristic is that the user did not read the datum in the correct, namely the Hadamard, basis. This can be seen by the presence of an extra Hadamard gate on q_0 in the very middle of the Figure 10. This has the same meaning as in Figure 9. In Figure 10, there is the additional pair q_2 and q_3 , with q_2 being the datum and q_3 the sentinel. A slight difference is that the datum has another initial setting, namely $q_2 = |1\rangle$. As before, the initial value of $q_2 = |1\rangle$

does not affect the capability of catching the intruder. As the intruder can be seen to erroneously read both q_1 and q_3 , the chance to signal the presence of the intruder is increased. Theoretically, the signalling chance of the intruder in this case is $p = 1 - (\frac{3}{4})^2 \approx 0.44$. The measurements on the real computer *Belem* deviate from the expected theoretical result, as shown in Figure 11. On the positive side, the practical test shows that the intruder is caught with probability $p_{pr} = 49.42$. Nevertheless, the problem is the unreliability; this number should not deviate that much from the theoretical expectation. In the cases where the server does not catch the intruder, namely, outcomes 0000, 0001, 0100, 0101, the measured values deviate by 11.82%. In the cases where the server catches the intruder, which are all the others, the difference is even worse, namely 21.98%. The worrying situation shows at the very base state 0000, where the theoretical percentage should be 6.13%, but is actually 20.2%. It seems that the state of a qubit easily and spontaneously reverts back to the base state $|0\rangle$.

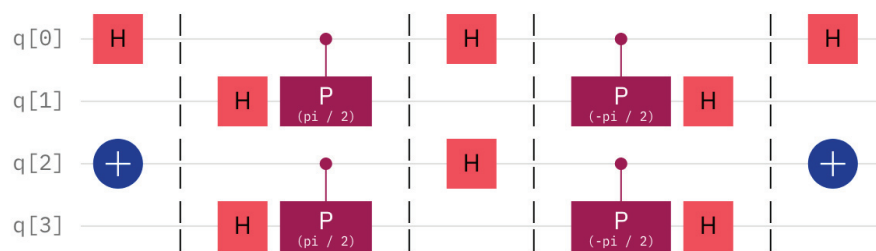


Figure 10. The circuit with two active hidden sentinels shows an intruder that has wrongly measured two datum qubits, q_0 and q_2 , that act on two hidden sentinels, q_1 and q_3 .

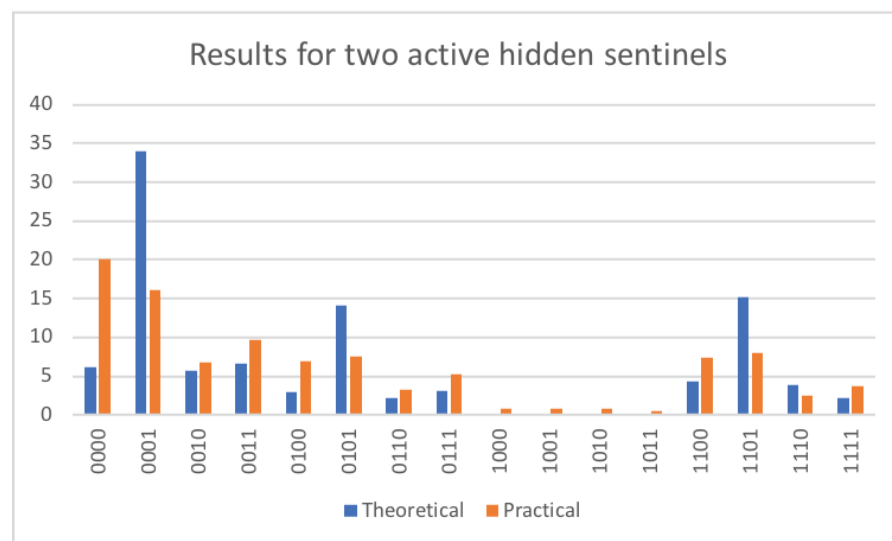


Figure 11. The left panel shows the practical results obtained on running a circuit with two hidden sentinels. The panel on the right shows the theoretical expectation.

It remains to be seen that these ideas can be implemented once error free quantum computers are available.

4.2.2. Sentinel Complexity Comparison

Positional sentinels versus hidden sentinels exhibit differences in terms of behavior, scope, implementation complexity, and cost.

The main difference is that positional sentinels are part of the data that are exposed to the intruder, whereas hidden sentinels are simply acted on by exposed data, but are not accessible to the user at all. As such, positional sentinels incur a simple cost of one qubit, while hidden sentinels need two qubits. Setting up a positional sentinel means simply

setting the value of the qubit to a meaningful value. This means zero, one or two gates, depending on the value to be set. For hidden sentinels, additionally to setting the value of the datum, which is identical to the positional sentinel, the circuit requires two more Hadamard gates as well as two controlled phase shift gates. As controlled phase shift gates are two qubit gates, they are more complex and more prone to error. Thus, the hidden sentinels are more costly, both in terms of number of qubits as well as circuitry. The setting up of a hidden sentinel may also be considered more time consuming, though direct time evaluations of this action are hard to carry out for our limited experimental capacity. The below table offers a brief comparison. The table considers N sentinels.

Type of Sentinel	Number of qubits for N sentinels	Number of single qubit gates	Number of two qubit gates
Positional sentinel	N	from 0 to $2N$	0
Hidden sentinel	$2N$	from $2N$ to $4N$	$2N$

The resource comparison between positional and hidden sentinels is done for N sentinels.

Thus, hidden sentinels are more costly, but also more insidious.

The practicality of the methods presented here depends on the availability of quantum technology. The size of quantum computers today includes some tens of qubits. IBM quantum computers were based on 53 qubits in 2019. Some very rapid growth is expected in the near future. Quantum annealers have been reported as having 5000 qubits. Quantum networks, which allow quantum communication to happen, have stepped into the size of 700 optical fibers built in 2021 [17]. As this field is now growing on several fronts, the values given here may already be obsolete by the time the ink dries. The necessity of quantum honeypots may have to wait for a few more break-throughs in quantum technology.

5. Conclusions

This paper shows that a quantum network setting can contribute to the power of a honeypot system. This is because qubits can be checked for *reading* by adding sentinels to them. Sentinels can be added to as many qubits as wished for by the honeypot administrator. Quantum sentinels check whether a qubit, field, memory or disk location has been accessed for reading (only). There is no need for actual writing to detect the presence of illegal activity. Detecting the reading activity relies on quantum properties, such as the collapse of superposition and controlled quantum gates. Therefore, there is no possibility of mimicking the same capability by classical computational means.

Two types of quantum sentinels have been defined: positional-visible sentinels and hidden-invisible sentinels. The meaning captured in their name is that they are visible or invisible to the user of the honeypot system. In the case of hidden quantum sentinels, the illegal user is entirely unaware of the presence of the sentinel. The probability of catching an intruder on any sentinel is low and varies with the sentinel type. In the case of positional sentinels, the probability is $\frac{1}{4}$, and, in the case of hidden sentinels, the probability is $\frac{1}{8}$. Though hidden sentinels have the advantage of remaining hidden from the intruder, their drawback is a more complex quantum circuit with an extra qubit and a lower detection rate. Nevertheless, in both cases, the probability of catching an intruder can be increased to any arbitrary value by adding more sentinels.

Thus, quantum sentinels add the following properties to honeypots:

1. The monitoring of malicious activity can be detailed to the level of bit, that is the information unit.
2. The presence of the monitoring system can be fully hidden via hidden quantum sentinels.

Finally, today's quantum computers do not offer the accuracy necessary to practically implement such quantum honeypots, as our experiments show. The errors of both signaling a legal user or ignoring an intruder deviate from the theoretical expectations by 10 to 20%.

These values have been measured for two sentinels only. At this point, a larger experiment with more accurate error rates is not necessary, as even this value condemns the system as not being feasible as of yet. How soon this problem will be remedied remains an open question.

Author Contributions: Conceptualization, N.N. and M.N.; methodology, G.A.; software, L.A., S.A., Z.H., S.M.A. and A.A.; validation, G.A. and N.N.; formal analysis, N.N. and M.N.; writing—original draft preparation, N.N., G.A., L.A., S.A., Z.H., S.M.A. and A.A.; writing—review and editing, N.N. and G.A.; supervision, N.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fan, W.; Du, Z.; Fernández, D.; Villagra, V. Enabling an Anatomic View to Investigate Honey-pot Systems: A Survey. *IEEE Syst. J.* **2017**, *12*, 3906–3919. [CrossRef]
2. Spitzner, L. *Honeypots: Tracking Hackers*; Addison-Wesley: Boston, MA, USA, 2002.
3. Nielsen, M.; Chuang, I. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2000.
4. Grover, L. A fast quantum mechanical algorithm for database search. In Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–220.
5. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *Spec. Issue Quantum Comput. SIAM J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
6. Nagy, M.; Akl, S.G. Quantum computation and quantum information. *Int. J. Parallel Emergent Distrib. Syst.* **2006**, *21*, 1–59.
7. Bennett, C.H.; Brassard, G. Quantum cryptography: Public key distribution and coin tossing. Theoretical Aspects of Quantum Cryptography—Celebrating 30 years of BB84. *Theor. Comput. Sci.* **2014**, *560*, 7–11. [CrossRef]
8. Ladd, T.D.; Jelezko, F.; Laflamme, R.; Nakamura, Y.; Monroe, C.; O’Brien, J.L. Quantum computers. *Nature* **2010**, *464*, 45–53. [CrossRef] [PubMed]
9. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Br, ao, F.G.S.L.; Buell, D.A. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef] [PubMed]
10. Asproni, L.; Caputo, D.; Silva, B.; Fazzi, G.; Magagnini, M. Accuracy and minor embedding in subqubo decomposition with fully connected large problems: A case study about the number partitioning problem. *Quantum Mach. Intell.* **2020**, *2*, 4. [CrossRef]
11. Ng, C.K.; Pan, L.; Xiang, Y. *Honeypot Frameworks and Their Applications: A New Framework*; SpringerBriefs on Cyber Security Systems and Networks; Springer: Singapore, 2018. [CrossRef]
12. Zhong, H.S.; Wang, H.; Deng, Y.H.; Chen, M.C.; Peng, L.C.; Luo, Y.H.; Qin, J.; Wu, D.; Ding, X.; Hu, Y.; et al. Quantum computational advantage using photons. *Science* **2020**, *370*, 1460–1463. [CrossRef] [PubMed]
13. García-Cobo, I.; Menéndez, H.D. Designing large quantum key distribution networks via medoid-based algorithms. *Future Gener. Comput. Syst.* **2021**, *115*, 814–824. [CrossRef]
14. Mehic, M.; Niemiec, M.; Rass, S.; Ma, J.; Peev, M.; Aguado, A.; Martin, V.; Schauer, S.; Poppe, A.; Pacher, C.; et al. Quantum Key Distribution: A Networking Perspective. *ACM Comput. Surv.* **2020**, *53*, 96:1–96:41. [CrossRef]
15. Nagy, M.; Akl, S.G. Coping with Decoherence: Parallelizing the Quantum Fourier Transform. *Parallel Process. Lett.* **2010**, *20*, 213–226. [CrossRef]
16. IBM-QX. IBM Quantum Experience. 2020. Available online: <https://quantum-computing.ibm.com/> (accessed on 1 July 2020).
17. Chen, Y.A.; Zhang, Q.; Chen, T.Y.; Cai, W.Q.; Liao, S.K.; Zhang, J.; Chen, K.; Yin, J.; Ren, J.G.; Chen, Z.; et al. An integrated space-to-ground quantum communication network over 4600 kilometres. *Nature* **2021**, *589*, 214–219. [CrossRef] [PubMed]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Quantum-Walk-Inspired Dynamic Adiabatic Local Search

Chen-Fu Chiang ^{1,*} and Paul M. Alsing ²¹ Department of Computer Science, State University of New York Polytechnic Institute, Utica, NY 13203, USA² Information Directorate, Air Force Research Laboratory, Rome, NY 13441, USA; paul.alsing@us.af.mil

* Correspondence: chiangc@sunypoly.edu

Abstract: We investigate the irreconcilability issue that arises when translating the search algorithm from the Continuous Time Quantum Walk (CTQW) framework to the Adiabatic Quantum Computing (AQC) framework. For the AQC formulation to evolve along the same path as the CTQW, it requires a constant energy gap in the Hamiltonian throughout the AQC schedule. To resolve the constant gap issue, we modify the CTQW-inspired AQC catalyst Hamiltonian from an XZ operator to a Z oracle operator. Through simulation, we demonstrate that the total running time for the proposed approach for AQC with the modified catalyst Hamiltonian remains optimal as CTQW. Inspired by this solution, we further investigate adaptive scheduling for the catalyst Hamiltonian and its coefficient function in the adiabatic path of Grover-inspired AQC to improve the adiabatic local search.

Keywords: quantum walk; adiabatic quantum computing; adiabatic path scheduling; catalyst Hamiltonian

1. Introduction

Quantum technologies have advanced dramatically in the past decade, both theoretically and experimentally. From the view of theoretical computational complexity, Shor's factoring algorithm [1] and Grover's search algorithm [2] are well-known for their improvements over the best possible classical algorithms designed for the same purpose. From a perspective of universal computational models, Quantum Walks (QWs) have become a prominent model of quantum computation due to their direct relationship to the physics of the quantum system [3,4]. It has been shown that the QW computational framework is universal for quantum computation [5,6], and many algorithms now are presented directly in the quantum walk formulation rather than through a circuit model or other abstracted method [3,7]. Besides being search algorithms, CTQWs have been applied in fields such as quantum transport [8–11], state transfer [12,13], link prediction in complex networks [14] and the creation of Bell pairs in a random network [15]. Some other well known universal models include the quantum circuit model [16–18], topological quantum computation [19], adiabatic quantum computation (AQC) [20], resonant transition-based quantum computation [21] and measurement-based quantum computation [22–25]. Investigating relationships among the frameworks helps to identify violations when mapping frameworks and potential solutions. By studying the mapping, one can extend the techniques from one framework to another for some potential improvement in terms of speed [26].

In this work, we investigate the irreconcilability issue that arises when translating the search algorithm from the Continuous Time Quantum Walk (CTQW) framework to the Adiabatic Quantum Computing (AQC) framework as first pointed out by Wong and Meyer [27]. This irreconcilability issue can be described as follows. One first notes that the CTQW is the unique continuous time quantum walk formulation of Grover's discrete search algorithm. While the CTQW search evolves the initial unbiased (equal amplitude) state to the unknown (marked) state on the order of time $T \sim \mathcal{O}(\sqrt{N})$ (where N is the size of the search space), it does not follow the same evolution path (on the Bloch sphere) as that of Grover's algorithm. The uniqueness of the CTQW formulation stems from the fact

Citation: Chiang, C.-F.; Alsing, P.M. Quantum-Walk-Inspired Dynamic Adiabatic Local Search. *Entropy* **2023**, *25*, 1287. <https://doi.org/10.3390/e25091287>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 10 August 2023

Accepted: 27 August 2023

Published: 31 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

that the unknown marked state only acquires a (time-dependent) phase from the oracle operation. Most importantly the marked states do not undergo evolution, and thus the CTQW effectively employs a dichotomous “Yes/No” oracle, for which the discrete Grover’s algorithm has been proven to be optimal.

The AQC formulation of the search algorithm with a non-uniform adiabatic evolution schedule [28] also finds the marked state in time $T \sim \mathcal{O}(\sqrt{N})$ while following the same path as Grover’s algorithm. Thus, if one investigates what adiabatic Hamiltonian gives rise to the same evolution path as the CTQW formulation, one finds [27] that the AQC formulation introduces an extra “catalyst” Hamiltonian which introduces a structure beyond the standard “Yes/No” oracle employed in the CTQW or discrete (Grover’s) search algorithm. A scaled version of the AQC Hamiltonian leads to a constant energy gap that implies that the marked state can be found in time $T \sim \mathcal{O}(1)$. This discrepancy between the formulations of the two versions of a continuous time search algorithm was termed the “irreconcilability (difference) issue” between CTQW and AQC by Wong and Meyer [27].

In this work, we address the CTQW/AQC search algorithm irreconcilability issue by modifying the constant energy gap Hamiltonian of the AQC formulation. Our contribution is twofold. We first adapt the result from the mapping of CTQW to AQC by selecting the regular oracle Z operator as the catalyst Hamiltonian and explore an alternative for the coefficient function for the catalyst Hamiltonian in order to attempt to avoid the irreconcilability issue. Through the simulation, the modified model provides optimal results in terms of the time required for the search.

The second improvement is on the Grover-Search-inspired adiabatic local search, we add an additional sluggish parameter δ which delineates the width of the adiabatic run time schedule over which the catalyst Hamiltonian effectively acts (i.e., the “slowdown” region in the vicinity of the system’s smallest energy gap Δ). The sluggish parameter tracks the increase of running time $t = t(s)$ with respect to schedule parameter $0 \leq s \leq 1$ where $\delta = |d^2t/ds^2|$. The catalyst is employed when $\delta \geq \delta_0$ to facilitate the process; we have found that the threshold value of $\delta_0 = 64$ provides good results. When simulated, this modification reduces the running time of the original adiabatic local search by certain constant factors.

The outline of this work is as follows. The background information regarding CTQW and AQC is given in Section 2 where the translation of CTQW to AQC is described in Section 3. The irreconcilability issue that occurs during the translation is explained in Section 3.1 and our proposed solution is provided in Section 3.2. The mapping of Grover search to AQC as an adiabatic local search is summarized in Section 4. We propose and describe the catalyst Hamiltonian mechanism in Section 4.1.2 and determine the sluggish interval where it is employed. We further explore three coefficient functions of the catalyst Hamiltonian in Section 4.1.3. The simulation results for the proposed modifications are discussed in Section 5. Finally, our conclusions are given in Section 6.

2. Background

2.1. Continuous Time Quantum Walk

Given a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, the CTQW on G is defined as follows. Let A be the adjacency matrix of G , the $|V| \times |V|$ matrix is defined component-wise as

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $i, j \in V$. A CTQW starts with a uniform superposition state $|\psi_0\rangle$ in the space, spanned by nodes in V , and evolves according to the Schrödinger equation with Hamiltonian A . After time t , the output state is thus

$$|\psi_t\rangle = e^{-iAt}|\psi_0\rangle. \quad (2)$$

The probability that the walker is in the state $|\tau\rangle$ at time t is given by $|\langle\tau|e^{-iAt}|\psi_0\rangle|^2$. To find the marked node $|\omega\rangle$ starting from an initial state $|\psi_0\rangle$ via a CTQW, one has to maximize the success probability

$$|\langle\omega|e^{-iAt}|\psi_0\rangle|^2 \tag{3}$$

while minimizing the time t . For instance, initially at time $t = 0$, the success probability is

$$|\langle\omega|e^{-iA0}|\psi_0\rangle|^2 = O\left(\frac{1}{|V|}\right). \tag{4}$$

The success probability is extremely small when the search space $|V| = N$ is large and $|\psi_0\rangle$ is a uniform superposition state.

When applied to spatial search, the purpose of a CTQW is to find a marker basis state $|\omega\rangle$ [29,30]. For this purpose, the CTQW starts with the initial state $|\psi_0\rangle = \sum_{i=1}^N \frac{1}{\sqrt{N}}|i\rangle$, and evolves according to the Hamiltonian [31]

$$H = -\gamma A - |\omega\rangle\langle\omega| \tag{5}$$

where γ is the coupling factor between connected nodes. The value of γ has to be determined based on the graph structure such that the quadratic speedup of CTQW can be preserved. Interested readers can refer to [29,31] for more details.

2.2. Adiabatic Quantum Computing

In the AQC model, H_0 is the initial Hamiltonian, H_f is the final Hamiltonian. The evolution path for the time-dependent Hamiltonian is

$$H(s) = (1 - s)H_0 + sH_f \tag{6}$$

where $0 \leq s \leq 1$ is a schedule function of time t . For convenience, we denote s as $s(t)$ and use them interchangeably. The variable s increases slowly enough that the initial ground state evolves and remains as the instantaneous ground state of the system. More specifically,

$$H(s(t))|\lambda_{k,t}\rangle = \lambda_{k,t}|\lambda_{k,t}\rangle \tag{7}$$

where $\lambda_{k,t}$ is the corresponding eigenvalue the eigenstate $|\lambda_{k,t}\rangle$ at time t and k labels for the k_{th} excited eigenstate. The minimal eigenvalue gap is defined as

$$g_{min} = \min_{0 \leq t \leq T_a} (\lambda_{1,t} - \lambda_{0,t}) \tag{8}$$

where T_a is the total evolution time of the AQC. Let $|\psi(T_a)\rangle$ be the state of the system at time T_a evolving under the Hamiltonian $H(s(t))$ from the ground state $|\lambda_{0,0}\rangle$ at time $t = 0$. The Adiabatic theorem [32,33] states that the final state $|\psi(T_a)\rangle$ is ϵ -close to the real ground state $|\lambda_{0,T_a}\rangle$ as

$$|\langle\lambda_{0,T_a}|\psi(T_a)\rangle|^2 \leq 1 - \epsilon^2, \tag{9}$$

provided that

$$\frac{|\langle\lambda_{1,t}|\frac{dH}{dt}|\lambda_{0,t}\rangle|}{g_{min}^2} \leq \epsilon. \tag{10}$$

There are several variations of AQC to improve the performance. The variations are based on modifying the initial Hamiltonian and the final Hamiltonian [34,35] or adding a catalyst Hamiltonian H_e [34], which is turned on/off at the beginning/end of the adiabatic

evolution. In this work, we are interested in the catalyst approach. A conventional catalyst Hamiltonian-assisted AQC path is expressed as

$$H(s) = (1 - s)H_0 + s(1 - s)H_e + sH_f. \tag{11}$$

3. Continuous Time Quantum Walk to Adiabatic Search Mapping

One can construct a time-dependent AQC Hamiltonian $H(s)$ as shown in [27] where the adiabatic search follows the CTQW search on a complete graph with N vertices. Let us define the following variables. The coupling factor γ is set to $1/N$ and $|\psi_0\rangle$ is the uniform superposition of all states in the search space. State $|r\rangle$ is the uniform superposition of non-solution states, state $|\omega\rangle$ is the solution state. Treating the state evolving in the CTQW system as the time-dependent ground state of $H(s)$, one constructs $H(s)$ in the $\{|\omega\rangle, |r\rangle\}$ basis as [27]

$$H(s) = \sqrt[4]{\frac{s(1-s)}{4\epsilon^2 N}} [(1-s)H_0 + \sqrt{s(1-s)}H_e + sH_f] \tag{12}$$

where $s(t) = \sin^2(\frac{t}{\sqrt{N}})$ with

$$\begin{aligned} H_0 &= |\psi_0^\perp\rangle\langle\psi_0^\perp| - |\psi_0\rangle\langle\psi_0|, & H_f &= |\gamma\rangle\langle\gamma| - |\omega\rangle\langle\omega|, \\ H_e &= 2i\sqrt{\frac{N-1}{N}}(|r\rangle\langle\omega| - |\omega\rangle\langle r|), \end{aligned} \tag{13}$$

or explicitly in the $\{|w\rangle, |r\rangle\}$ basis as

$$\begin{aligned} H_0 &= \begin{pmatrix} \frac{N-2}{N} & -2\frac{\sqrt{N-1}}{N} \\ -2\frac{\sqrt{N-1}}{N} & -\frac{N-2}{N} \end{pmatrix}, \\ H_e &= \begin{pmatrix} 0 & -2i\sqrt{\frac{N-1}{N}} \\ 2i\sqrt{\frac{N-1}{N}} & 0 \end{pmatrix}, & H_f &= \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned} \tag{14}$$

3.1. The Irreconcilability Issue: Constant Gap Catalyst Hamiltonian and Small Norm

The main concerns that are raised from Equation (12) are twofold. The first issue is the factor $\sqrt[4]{\frac{s(1-s)}{4\epsilon^2 N}}$ of $H(s)$. The adiabatic theorem [36] states that the system achieves a fidelity of $1 - \epsilon$ to the target state, provided that

$$\frac{|\langle \frac{dH}{dt} \rangle_{0,1}|}{g_{min}^2} \leq \epsilon, \text{ where } g_{min} = \min_{0 \leq t \leq T} E_1(t) - E_0(t). \tag{15}$$

Here, $\langle \frac{dH}{dt} \rangle_{0,1}$ are the matrix elements of dH/dt between the two corresponding eigenstates. $E_0(t)$ and $E_1(t)$ are the ground energy and the first excited energy of the system at time t . Given the $H(s)$ in Equation (12), one might conclude that a factor of $O(\sqrt[4]{1/N})$ significantly reduces the time required to achieve $1 - \epsilon$ precision. This might be misleading as the g_{min} of $H(s)$ also carries the same factor. The second issue is that the catalyst H_e provides power greater than a typical Yes/No oracle as it maps non-solution states to a solution state and a solution state to non-solution states. Provided that we initially start with a superposition state with an amplitude of $\sqrt{\frac{N-1}{N}}$ for a non-solution, it takes a time of $O(1)$ for this catalyst to drive the initial (unbiased, equal amplitude) state to the solution state. In the following, we will relax this constraint by using a normal oracle. For the rest of the paper, let us simply treat $\epsilon \ll 1$ as a small negligible constant.

3.2. Modified CTQW-Inspired Adiabatic Search

In Equation (12), the following parameters were computed during the mapping [27]:

- the scaling factor $\sqrt[4]{\frac{s(1-s)}{4\epsilon^2N}}$ of Hamiltonian H_0 ,
- $H_e = 2i\sqrt{\frac{N-1}{N}}(|r\rangle\langle\omega| - |\omega\rangle\langle r|)$, catalyst Hamiltonian
- the coefficient function of H_e as $\sqrt{s(1-s)}$.

In [37], the cost of the adiabatic algorithm was defined to be the dimensionless quantity (using $\hbar = 1$)

$$cost = t_f \max_s ||H(s)||, \tag{16}$$

where t_f is the running time. To prevent the cost from being manipulated to be arbitrarily small by changing the time units or distorting the scaling of the algorithm by multiplying the Hamiltonians by some size-dependent factor as shown in the irreconcilability concern [27], the norm of $H(s)$ should be fixed to some constant, such as 1.

To address the irreconcilability issue, the scaling factor is dropped and the catalyst Hamiltonian H_e is modified. Since $H_e = 2\sqrt{\frac{N-1}{N}}iXZ$ in the $\{|\omega\rangle, |r\rangle\}$ basis provides more power than a standard oracle, for our modification we remove the imaginary number i and the X operator. The operator Z alone behaves as a conventional “Yes /No” oracle in the $\{|\omega\rangle, |r\rangle\}$ basis. Let $M = 2\sqrt{\frac{N-1}{N}}$ and choose the modified adiabatic path $H_m(s)$ as

$$H_m(s) = (1-s)H_0 + f_z(s)MZ + sH_f, \tag{17}$$

where $f_z(s)$ is our chosen s -dependent coefficient for catalyst Z . In addition to $f_z(s) = \sqrt{s(1-s)}$ that was used in [27], functions that reach their maximum when $s = 1/2$ are good candidates for $f_z(s)$, such as $f_z(s) = \frac{\sin(s\pi)}{2}$. The use of the factor 1/2 on the sine function is to offset the magnitude M to bound the norm of H_e as described in Equation (16).

4. Grover Search to Adiabatic Local Search Mapping

In this section we consider the mapping of Grover’s algorithm to an adiabatic search. Given the initial driving Hamiltonian H_0 and the final Hamiltonian H_f as

$$H_0 = I - |\psi_0\rangle\langle\psi_0|, \quad H_f = I - |\omega\rangle\langle\omega|, \tag{18}$$

where

$$H_0 = \begin{pmatrix} \frac{N-1}{N} & -\frac{\sqrt{N-1}}{N} \\ -\frac{\sqrt{N-1}}{N} & \frac{1}{N} \end{pmatrix}, H_f = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \tag{19}$$

in the $\{|\omega\rangle, |r\rangle\}$ basis. The adiabatic path [27,28] in the $\{|\omega\rangle, |r\rangle\}$ basis is given by

$$H(s) = (1-s)H_0 + sH_f \tag{20}$$

$$= \begin{pmatrix} (1-s)\frac{N-1}{N} & -(1-s)\frac{\sqrt{N-1}}{N} \\ -(1-s)\frac{\sqrt{N-1}}{N} & 1 - (1-s)\frac{N-1}{N} \end{pmatrix}. \tag{21}$$

Instead of employing a linear evolution of $s(t)$, Equation (20) adapts the evolution ds/dt to the local adiabaticity condition [28] such that

$$|\frac{ds}{dt}| = \epsilon g^2(t) \tag{22}$$

where $g(t)$ is the energy gap of the system at time t . The running time t is then a function of schedule s such that

$$t(s) = \frac{N}{2\epsilon\sqrt{N-1}} \left\{ \arctan\left(\sqrt{N-1}(2s-1)\right) \right. \tag{23}$$

$$\left. + \arctan\left(\sqrt{N-1}\right) \right\}. \tag{24}$$

The relationship between the schedule s and the running time t is shown in Figure 2 in [28]. It is a tailored schedule that goes fast in the outer regions and slows down near the gap. It is clear that the system evolves quickly when the gap is large (s away from $1/2$) and slowly when the gap is small ($s \simeq 1/2$) [28]. In this example, the sluggish period $s \in [0.4, 0.6]$. For completeness, we provide the formal proof of the close form of the squared gap function $g^2(t)$ (second order in s) with respect to the schedule s in Appendix A.

4.1. Adaptive Scheduling

For a fixed schedule of an adiabatic path, the schedule s moves fast when the eigen-energy gap is large, and slowly when the gap is small. We desire to employ the catalyst Hamiltonians H_e to amplify the eigen-energy gap during the “slow down” period such that the total time to pass through the sluggish period is reduced ($s \in [0.4, 0.6]$ in Figure 2 in [28]).

4.1.1. Schedule-Dependent Gap Function

In this section, we consider employing gap-dependent scheduling functions. Let H_f be an arbitrary 2 by 2 Hermitian Hamiltonian. Let the time-dependent Hamiltonian $H(s)$ be

$$H(s) = (1-s)H_0 + f_x(s)\sigma_x + f_z(s)\sigma_z + sH_f. \tag{25}$$

Operators σ_x and σ_z are chosen as catalyst Hamiltonians. Let $H_0 = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$, $H_f = \begin{bmatrix} p & r \\ r & q \end{bmatrix}$ where a, b, c, p, q, r are some given constants. The matrix form of the time-dependent Hamiltonian is given by

$$H(s) = \begin{bmatrix} (1-s)a + sp + f_z(s) & (1-s)c + sr + f_x(s) \\ (1-s)c + sr + f_x(s) & (1-s)b + sq - f_z(s) \end{bmatrix} \tag{26}$$

and the schedule-dependent gap can be analytically computed to yield

$$g^2(s) = ((1-s)(a-b) + s(p-q) + 2f_z(s))^2 + 4((1-s)c + sr + f_x(s))^2, \tag{27}$$

(see Appendix B for a derivation). By using Equation (22), the total running time T_{strt}^{stp} from $s = s_{strt}$ to $s = s_{stp}$ is thus

$$T_{strt}^{stp} = \int_{s_{strt}}^{s_{stp}} \frac{ds}{\epsilon g^2(s)} \tag{28}$$

where $0 \leq s_{strt} \leq s_{stp} \leq 1$. In brief, the time spent during a certain period of a schedule can be obtained by use of a gap function. The gap function can be expressed via the entries of H_0 , H_e , H_f , schedule s and the coefficient functions of the catalyst Hamiltonians.

4.1.2. Determining the Sluggish Interval for the Catalyst Hamiltonian

By using the condition $f'(s) = dt/ds = \frac{1}{\epsilon g^2(s)}$ (see Appendix A), the region where the gap quickly significantly decreases or increases is during the sluggish period of s . That is the portion of the schedule s where a catalyst should be employed. The region where $|df^2(s)/ds^2| \geq \delta_0$ is the sluggish period. The threshold value $\delta_0 = 64$ was chosen because

if we choose a threshold proportional to N , as N increases exponentially, the quantity d^2t/ds^2 might never reach the N -dependent threshold within the adiabatic evolution schedule $0 \leq s \leq 1$. By using this threshold, the starting point s_{str}^{slug} and the stopping point s_{stp}^{slug} used to mark the sluggish period can be identified. Using the example in [28], we can re-plot and get t as a function of s as $t = f(s)$ and $f'(s) = dt/ds$ in Figures 1 and 2 with $N = 64$.

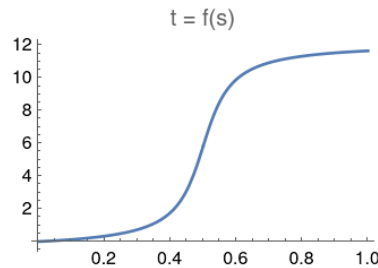


Figure 1. Time t as a function of schedule s for adiabatic local search with $N = 64$.

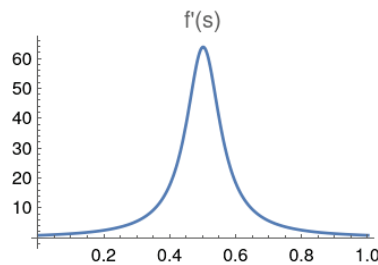


Figure 2. dt/ds for adiabatic local search with $N = 64$.

4.1.3. Catalyst Coefficient Functions

As discussed in Section 3.2, we are interested in the $H_e = Z$ case in Equation (17) and its coefficient function $f_z(s)$. Three coefficient functions of the catalyst Hamiltonian Z are proposed as the following

$$\begin{aligned}
 f_z^{sine}(s) &= \sin(((s - s_{str}^{slug}) * \pi) / (s_{stp}^{slug} - s_{str}^{slug})), \\
 f_z^{ss}(s) &= (s - s_{str}^{slug})(s_{stp}^{slug} - s), \\
 f_z^{grid}(s) &= a * f_z^{sine}(s) + b * (f_z^{sine}(s))^2
 \end{aligned}
 \tag{29}$$

where $0 \leq a, b \leq 1$ under the constraint that $a^2 + b^2 = 1$. In the grid search a increased from 0 to 1 by 0.1 in each iteration. From the 10 pairs of (a, b) , we find the values of a, b that give the shortest sluggish time interval.

5. Experiment and Result

For our simulations we used (Wolfram) Mathematica (version 12.3 run on a Linux Ubuntu 20.04 LTS laptop). The code is available upon request. The running time is based on Equation (28). The size N (number of nodes) ranges from $2^5, 2^6, \dots$ to 2^{25} . We observe the corresponding running time and sluggish time for each of the proposed models. The result of the original adiabatic local search serves as the baseline for comparison, which used $N = 64$ [28]. In this work, we generalize the setting for any arbitrary size N .

Given an arbitrary complete graph of size N with coupling factor $1/N$, one can compute the entries in the reduced Hamiltonian for H_0 and H_f in the $\{|\omega\rangle, |r\rangle\}$ basis. The values of variables a, b, c, p, q and r as discussed in Section 4.1.1 can be obtained from Equation (14) for the CTQW case and from Equation (19) for the adiabatic local search. It is worth noticing that the ground state energy is -1 in the CTQW case, but is 0 in the adiabatic local search case. Based on the adiabatic path Equation (25) and the gap function

in Equation (27) with given schedule s , coefficient function $f_z(s)$ for σ_z , we perform the simulation with the running time computed from Equation (28).

5.1. Modified CTQW-Inspired Adiabatic Search Simulation

This experiment aimed to demonstrate that the modified adiabatic paths addressing the irreconcilable issues remain optimal. The three proposed modifications we explored are as follows:

- $H_{org}(s)$ takes Equation (12) and drops the scaling factor as explained in Section 3.2. The adiabatic path is $H_{org}(s) = (1 - s)H_0 + \sqrt{s(1 - s)}H_e + sH_f$
- $H_{m1}(s)$ replaces the computed catalyst Hamiltonian H_e with an ordinary Z oracle operator and keeps the magnitude M . This was used to address the constant gap H_e irreconcilability issue. We have $H_{m1}(s) = (1 - s)H_0 + \sqrt{s(1 - s)}MZ + sH_f$
- $H_{m2}(s)$ uses $\frac{\sin(s\pi)}{2}$ as the coefficient function for the catalyst Hamiltonian Z . The adiabatic path is $H_{m2}(s) = (1 - s)H_0 + \frac{\sin(s\pi)}{2}MZ + sH_f$

For the above three models, simulations were run on a Hamiltonian of size $N \in [2^5, 2^6, \dots, 2^{25}]$. In the following figures, the abscissa is $\log_2 N$ while the ordinate is the required total running time T . The time is computed based on Equation (28). As the dimension of the Hamiltonian increases, the difference in running times for the three models considered are magnified.

The simulation results are shown in Figure 3. It is clear to see that H_{org} is a constant time scheme as it does not scale as the size N increases. This indicates that the original catalyst Hamiltonian $H_e = MXZ$ in $H_{org}(s)$ is indeed a constant gap Hamiltonian. This also shows the irreconcilability issue as suggested in [27]. From the simulations we can conclude that both $H_{m1}(s), H_{m2}(s)$ perform optimally with respect to running time, namely $T \sim \mathcal{O}(\sqrt{N})$, similar to that of the original adiabatic local search but with a minor constant factor which can be ignored in the Big O notation. As the simulation suggests, both modified CTQW-inspired approaches outperform the original adiabatic local search. When the $N \leq 2^{21}$, the $H_{m2}(s)$ outperforms $H_{m1}(s)$. When problem size N is larger then 2^{21} , $H_{m1}(s)$ is a better choice over $H_{m2}(s)$.

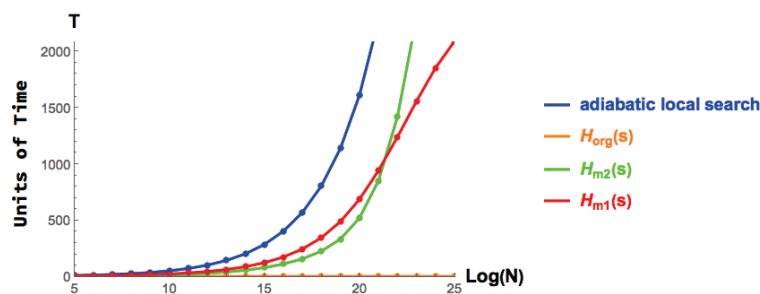


Figure 3. Case when $N \in [2^5, 2^{25}]$ and the running times of $H_{org}(s)$ (orange), $H_{m1}(s)$ (red) and $H_{m2}(s)$ (green) with the original adiabatic local search (blue) serving as the baseline.

5.2. Adaptive Adiabatic Local Search Simulation with Various Coefficient Functions

In the previous Section 5.1, the proposed modifications are optimal, in the sense that $T \sim \mathcal{O}(\sqrt{N})$ up to a minor constant factor. For further improvement, the adaptive scheduling scheme is applied. The adiabatic path to be explored is therefore

$$H_{adapt}(s) = (1 - s)H_0 + f(s)Z + sH_f \tag{30}$$

where $f(s) \in [f_z^{sine}, f_z^{ss}, f_z^{grid}]$, as seen in Equation (29). The catalyst Hamiltonian Z operator is only employed during the sluggish period and hence $f(s) = 0$ when $s \notin [s_{stri}^{slug}, s_{stp}^{slug}]$. The

H_0 and H_f are based on Equation (19). As the catalyst is only employed within the sluggish period, to compare the performance of each proposed modification, one only needs to compute the running time within this period.

In Figure 4, f_z^{ss} provides the minimal reduced sluggish time while f_z^{sine} and f_z^{grid} provide significant improvements. The difference in the runtimes becomes significant for $N \geq 2^{15}$.

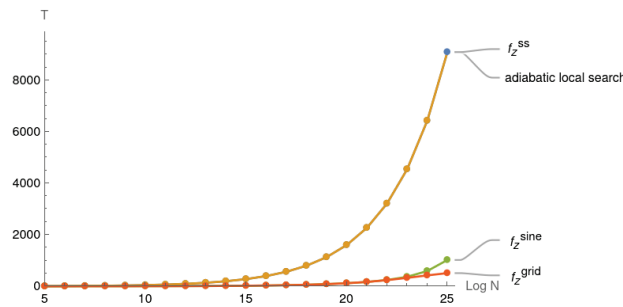


Figure 4. Case when $N \in [2^5, 2^{25}]$ and time spent in during the sluggish period for adiabatic paths with $(f_z^{ss}, f_z^{sine}, f_z^{grid})$ coefficient functions where the original adiabatic local search serves as the baseline.

In Figure 5, both f_z^{sine} and f_z^{grid} have a more than 75% reduced sluggish time in comparison to the original adiabatic local search when N reaches 2^{25} . f_z^{sine} gradually outperforms the original adiabatic local search after $N = 2^{10}$ and remains almost as good as f_z^{grid} until $N = 2^{23}$. When $N = 2^{25}$, the sluggish time of f_z^{sine} is only twice that of f_z^{grid} . In general, the grid search is a costly procedure as we have to run 10 pairs of (a, b) for slightly different $H(s)$ for each value of $N = 2^n$. If the time reduction of the sluggish period is not greater than 90% of the original, it might be a better choice to use f_z^{sine} . For the near term it might be more beneficial to use the f_z^{sine} model, instead of the grid search model f_z^{grid} .

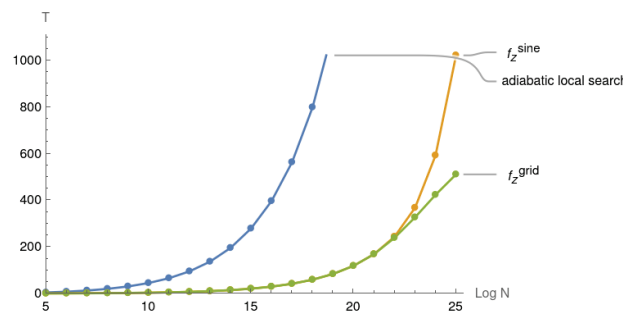


Figure 5. Case when $N \in [2^5, 2^{25}]$ and time spent during the sluggish period for adiabatic paths with (f_z^{sine}, f_z^{grid}) coefficient functions where the original adiabatic local search serves as the baseline.

6. Conclusions

In this work, we investigated different Hamiltonians for resolving the irreconcilability issue [27] when mapping the CTQW search algorithm to AQC. We modified the time-dependent Hamiltonian by (1) removing the original scaling CTQW factor $\sqrt[4]{\frac{s(1-s)}{4c^2N}}$ and (2) replacing $iXZ \rightarrow Z$ in the original catalyst H_e Hamiltonian obtained from mapping CTQW to AQC. These modifications were made in order to resolve the irreconcilability issue. We further optimized the schedule s of the CTQW-inspired adiabatic path by an adaptive scheduling procedure.

The modified CTQW-inspired adiabatic search simulation experiment demonstrates that indeed the H_e without any modification leads to a constant time in the total running time, regardless of the search space size N . This result echoes the irreconcilability issue stated in [27]. On the other hand, the modified CTQW-inspired adiabatic path with catalyst Hamiltonian coefficient $\frac{\sin(s\pi)}{2}$ behaves similarly to the behavior of the optimal adiabatic

local search. Furthermore, the modifications are optimal and outperform the original adiabatic local search.

Lastly, in the adaptive adiabatic local search simulation with various coefficient functions experiment, we further investigated how to reduce the time wasted in the sluggish period of an adiabatic local search path. As our numerical experiments show, the function $f_z^{sine}(s)$ and $f_z^{grid}(s)$ provide significant improvement and both outperform the original adiabatic local search. Even though the grid search $f_z^{grid}(s)$ approach could have further reduced the length of the sluggish (“slow down”) interval, the benefit was offset by the additional cost incurred from its implementation over that of the other two methods.

Author Contributions: Conceptualization, C.-F.C.; Methodology, C.-F.C. and P.M.A.; Validation, P.M.A.; Formal analysis, C.-F.C.; Investigation, C.-F.C. and P.M.A.; Writing—original draft, C.-F.C.; Writing—review & editing, P.M.A.; Supervision, P.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by U.S. Air Force Research Lab Summer Faculty Fellowship Program (SFFP).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code for the simulation is available at <https://github.com/omnibox/Quantum-Walk-Inspired-Dynamic-Adiabatic-Local-Search>.

Acknowledgments: C.-F.C. gratefully acknowledges the support from the support from the Air Force Research Laboratory Summer Faculty Fellowship Program (SFFP). P.M.A. would like to acknowledge the support of this work from the Air Force Office of Scientific Research (AFOSR). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Air Force Research Laboratory. The appearance of external hyperlinks does not constitute endorsement by the United States Department of Defense (DoD) of the linked websites, or the information, products or services contained therein. The DoD does not exercise any editorial, security or other control over the information you may find at these locations.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Time Integration of Adiabatic Local Search

Given a spectral gap polynomial of the second order, that is

$$g^2(s) = A(s^2 + bs + c) \tag{A1}$$

where s is the adiabatic schedule and (this is the same as $g^2(t)$ as for each t there is only one corresponding s) $\frac{ds}{dt} = \epsilon g^2(s)$, by integration on t one obtains

$$T = \int dt = \int_0^1 \frac{ds}{\epsilon g^2(s)} = \frac{1}{\epsilon A} \int_0^1 \frac{ds}{(s^2 + bs + c)}. \tag{A2}$$

(I) Case $b^2 - 4c > 0$: Let $r_{\pm} = \frac{-b \pm \sqrt{b^2 - 4c}}{2}$.

$$\int_0^1 \frac{ds}{(s^2 + bs + c)} = \frac{1}{r_+ - r_-} \int_0^1 \left(\frac{1}{s - r_+} - \frac{1}{s - r_-} \right) ds \tag{A3}$$

since $\int \frac{1}{s-a} ds = \ln |s - a|$. Thus, we have

$$T = \frac{1}{\epsilon A(r_+ - r_-)} \ln \left| \frac{s - r_+}{s - r_-} \right|_0^1 \tag{A4}$$

$$t = \frac{1}{\epsilon A(r_+ - r_-)} \left(\ln \left| \frac{s - r_+}{s - r_-} \right| - \ln \left| \frac{r_+}{r_-} \right| \right). \tag{A5}$$

(II) Case $b^2 - 4c = 0$:

$$\int_0^1 \frac{ds}{(s^2 + bs + c)} = \int_0^1 \frac{1}{(s + b/2)^2} ds \tag{A6}$$

since $\int (s - a)^{-2} ds = -(s - a)^{-1}$, hence

$$T = \frac{-1}{\epsilon A} \frac{1}{(s + (b/2))} \Big|_0^1 \tag{A7}$$

$$t = \frac{1}{\epsilon A} \left(\frac{s}{(b/2)(s + (b/2))} \right) \tag{A8}$$

(III) Case $b^2 - 4c < 0$:

$$\int_0^1 \frac{ds}{(s^2 + bs + c)} = \int_0^1 \frac{1}{(s + b/2)^2 + \frac{4c-b^2}{4}} ds \tag{A9}$$

$$= \int_{b/2}^{1+(b/2)} \frac{1}{x^2 + (\sqrt{\frac{4c-b^2}{4}})^2} dx \tag{A10}$$

since $\int \frac{1}{a^2+x^2} dx = \frac{1}{a} \arctan \frac{x}{a}$. With $a = \sqrt{\frac{4c-b^2}{4}}$, we obtain

$$T = \frac{1}{\epsilon A} \left(\frac{1}{a} \right) \left(\arctan \frac{x}{a} \right) \Big|_{b/2}^{1+(b/2)} \tag{A11}$$

$$t = \frac{1}{\epsilon A} \left(\frac{1}{a} \right) \left(\arctan \frac{s + (b/2)}{a} - \arctan \frac{(b/2)}{a} \right) \tag{A12}$$

Appendix B. Energy Gap

Given an arbitrary 2 by 2 non-negative-entry Hermitian matrix H as

$$H = \begin{bmatrix} \alpha & \gamma \\ \gamma & \beta \end{bmatrix}, \tag{A13}$$

via computing the determinant and eigenvalues, the energy gap ΔE is

$$\Delta E = |\lambda_+ - \lambda_-| = \sqrt{(\alpha - \beta)^2 + 4\gamma^2}. \tag{A14}$$

Simply from the view of energy gap, as long as $|\gamma|$ increases and the gap, $|\alpha - \beta|$, between the diagonal entries increases, the energy gap would increase. The increase of $|\gamma|$ can be adapted by σ_x while $|\alpha - \beta|$ can be increased by σ_z . They should be good candidates for the catalyst perturbation in the AQC path. Similarly, if the Hamiltonian has an imaginary part in the off-diagonal entries,

$$H = \begin{bmatrix} \alpha & \gamma - di \\ \gamma + di & \beta \end{bmatrix} \tag{A15}$$

$$\Delta E = |\lambda_+ - \lambda_-| = \sqrt{(\alpha - \beta)^2 + 4(\gamma^2 + d^2)}. \tag{A16}$$

The Hamiltonian H (with no imaginary entries) can be expressed in terms of Pauli matrices as

$$H = \frac{\alpha + \beta}{2} \mathbb{I} + \frac{\Delta E}{2} \left(\left(\frac{2\gamma}{\Delta E} \right) \sigma_x + \left(\frac{\alpha - \beta}{2} \right) \left(\frac{2}{\Delta E} \right) \sigma_z \right) \quad (\text{A17})$$

$$= \frac{\alpha + \beta}{2} \mathbb{I} + \frac{\Delta E}{2} A \quad (\text{A18})$$

such that, by use of the power of Pauli matrices,

$$e^{-iHt} = \cos\left(\frac{\Delta Et}{2}\right) \mathbb{I} - i \sin\left(\frac{\Delta Et}{2}\right) A. \quad (\text{A19})$$

References

- Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
- Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
- Farhi, E.; Gutmann, S. Quantum computation and decision trees. *Phys. Rev. A* **1998**, *58*, 915. [CrossRef]
- Kempe, J. Quantum random walks: An introductory overview. *Contemp. Phys.* **2003**, *44*, 307–327. [CrossRef]
- Childs, A.M. Universal computation by quantum walk. *Phys. Rev. Lett.* **2009**, *102*, 180501. [CrossRef] [PubMed]
- Lovett, N.B.; Cooper, S.; Everitt, M.; Trevers, M.; Kendon, V. Universal quantum computation using the discrete-time quantum walk. *Phys. Rev. A* **2010**, *81*, 042330. [CrossRef]
- Qiang, X.; Loke, T.; Montanaro, A.; Aungskunsiri, K.; Zhou, X.; O'Brien, J.L.; Wang, J.B.; Matthews, J.C. Efficient quantum walk on a quantum processor. *Nat. Commun.* **2016**, *7*, 1–6. [CrossRef] [PubMed]
- Caruso, F.; Chin, A.W.; Datta, A.; Huelga, S.F.; Plenio, M.B. Highly efficient energy excitation transfer in light-harvesting complexes: The fundamental role of noise-assisted transport. *J. Chem. Phys.* **2009**, *131*, 09B612. [CrossRef]
- Mohseni, M.; Rebentrost, P.; Lloyd, S.; Aspuru-Guzik, A. Environment-assisted quantum walks in photosynthetic energy transfer. *J. Chem. Phys.* **2008**, *129*, 11B603. [CrossRef] [PubMed]
- Rebentrost, P.; Mohseni, M.; Kassal, I.; Lloyd, S.; Aspuru-Guzik, A. Environment-assisted quantum transport. *New J. Phys.* **2009**, *11*, 033003. [CrossRef]
- Plenio, M.B.; Huelga, S.F. Dephasing-assisted transport: Quantum networks and biomolecules. *New J. Phys.* **2008**, *10*, 113019. [CrossRef]
- Bose, S. Quantum communication through an unmodulated spin chain. *Phys. Rev. Lett.* **2003**, *91*, 207901. [CrossRef]
- Kay, A. Perfect, efficient, state transfer and its application as a constructive tool. *Int. J. Quantum Inf.* **2010**, *8*, 641–676. [CrossRef]
- Omar, Y.; Moutinho, J.; Melo, A.; Coutinho, B.; Kovacs, I.; Barabasi, A. Quantum Link Prediction in Complex Networks. *APS* **2019**, *2019*, R28-003.
- Chakraborty, S.; Novo, L.; Ambainis, A.; Omar, Y. Spatial search by quantum walk is optimal for almost all graphs. *Phys. Rev. Lett.* **2016**, *116*, 100501. [CrossRef] [PubMed]
- Shor, P.W. Quantum computing. *Doc. Math.* **1998**, *1*, 1.
- Yao, A.C.C. Quantum circuit complexity. In Proceedings of the 1993 IEEE 34th Annual Foundations of Computer Science, Palo Alto, CA, USA, 3–5 November 1993; pp. 352–361.
- Jordan, S.P.; Lee, K.S.; Preskill, J. Quantum algorithms for quantum field theories. *Science* **2012**, *336*, 1130–1133. [CrossRef] [PubMed]
- Nayak, C.; Simon, S.H.; Stern, A.; Freedman, M.; Sarma, S.D. Non-Abelian anyons and topological quantum computation. *Rev. Mod. Phys.* **2008**, *80*, 1083. [CrossRef]
- Mizel, A.; Lidar, D.A.; Mitchell, M. Simple proof of equivalence between adiabatic quantum computation and the circuit model. *Phys. Rev. Lett.* **2007**, *99*, 070502. [CrossRef] [PubMed]
- Chiang, C.F.; Hsieh, C.Y. Resonant transition-based quantum computation. *Quantum Inf. Process.* **2017**, *16*, 120. [CrossRef]
- Morimae, T.; Fujii, K. Blind topological measurement-based quantum computation. *Nat. Commun.* **2012**, *3*, 1036. [CrossRef]
- Gross, D.; Eisert, J. Novel schemes for measurement-based quantum computation. *Phys. Rev. Lett.* **2007**, *98*, 220503. [CrossRef]
- Briegel, H.J.; Browne, D.E.; Dür, W.; Raussendorf, R.; Van den Nest, M. Measurement-based quantum computation. *Nat. Phys.* **2009**, *5*, 19–26. [CrossRef]
- Raussendorf, R.; Browne, D.E.; Briegel, H.J. Measurement-based quantum computation on cluster states. *Phys. Rev. A* **2003**, *68*, 022312. [CrossRef]
- Cutugno, M.; Giani, A.; Alsing, P.M.; Wessing, L.; Schnore, S. Quantum Computing Approaches for Mission Covering Optimization. *Algorithms* **2022**, *15*, 224. [CrossRef]
- Wong, T.G.; Meyer, D.A. Irreconcilable difference between quantum walks and adiabatic quantum computing. *Phys. Rev. A* **2016**, *93*, 062313. [CrossRef]

28. Roland, J.; Cerf, N.J. Quantum search by local adiabatic evolution. *Phys. Rev. A* **2002**, *65*, 042308. [CrossRef]
29. Childs, A.M.; Goldstone, J. Spatial search by quantum walk. *Phys. Rev. A* **2004**, *70*, 022314. [CrossRef]
30. Childs, A.M.; Cleve, R.; Deotto, E.; Farhi, E.; Gutmann, S.; Spielman, D.A. Exponential algorithmic speedup by a quantum walk. In Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, San Diego, CA, USA, 9–11 June 2003; pp. 59–68.
31. Novo, L.; Chakraborty, S.; Mohseni, M.; Neven, H.; Omar, Y. Systematic dimensionality reduction for quantum walks: Optimal spatial search and transport on non-regular graphs. *Sci. Rep.* **2015**, *5*, 13304. [CrossRef]
32. Farhi, E.; Goldstone, J.; Gutmann, S.; Sipser, M. Quantum computation by adiabatic evolution. *arXiv* **2000**, arXiv:quant-ph/0001106.
33. Albash, T.; Lidar, D.A. Adiabatic quantum computation. *Rev. Mod. Phys.* **2018**, *90*, 015002. [CrossRef]
34. Farhi, E.; Goldstone, J.; Gosset, D.; Gutmann, S.; Meyer, H.B.; Shor, P. Quantum Adiabatic Algorithms, Small Gaps, and Different Paths. *Quantum Info. Comput.* **2011**, *11*, 181–214. [CrossRef]
35. Perdomo-Ortiz, A.; Venegas-Andraca, S.E.; Aspuru-Guzik, A. A study of heuristic guesses for adiabatic quantum computation. *Quantum Inf. Process.* **2011**, *10*, 33–52. [CrossRef]
36. Griffiths, D.J.; Schroeter, D.F. *Introduction to Quantum Mechanics*; Cambridge University Press: Cambridge, UK, 2018.
37. Aharonov, D.; Van Dam, W.; Kempe, J.; Landau, Z.; Lloyd, S.; Regev, O. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Rev.* **2008**, *50*, 755–787. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

A Quantum Genetic Algorithm for Building a Semantic Textual Similarity Estimation Framework for Plagiarism Detection Applications

Saad M. Darwish ^{1,*}, Ibrahim Abdullah Mhaimed ² and Adel A. Elzoghbi ¹

¹ Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, 163 Horreya Avenue, El Shatby, P.O. Box 832, Alexandria 21526, Egypt; adel.elzoghby@alexu.edu.eg

² Department of Computer Science, University of Technology, Baghdad 10082, Iraq; ibrahim.hjwel99@gmail.com

* Correspondence: saad.darwish@alexu.edu.eg; Tel.: +20-1222632369

Abstract: The majority of the recent research on text similarity has been focused on machine learning strategies to combat the problem in the educational environment. When the originality of an idea is copied, it increases the difficulty of using a plagiarism detection system in practice, and the system fails. In cases like active-to-passive conversion, phrase structure changes, synonym substitution, and sentence reordering, the present approaches may not be adequate for plagiarism detection. In this article, semantic extraction and the quantum genetic algorithm (QGA) are integrated in a unified framework to identify idea plagiarism with the aim of enhancing the performance of existing methods in terms of detection accuracy and computational time. Semantic similarity measures, which use the WordNet database to extract semantic information, are used to capture a document's idea. In addition, the QGA is adapted to identify the interconnected, cohesive sentences that effectively convey the source document's main idea. QGAs are formulated using the quantum computing paradigm based on qubits and the superposition of states. By using the qubit chromosome as a representation rather than the more traditional binary, numeric, or symbolic representations, the QGA is able to express a linear superposition of solutions with the aim of increasing gene diversity. Due to its fast convergence and strong global search capacity, the QGA is well suited for a parallel structure. The proposed model has been assessed using a PAN 13-14 dataset, and the result indicates the model's ability to achieve significant detection improvement over some of the compared models. The recommended PD model achieves an approximately 20%, 15%, and 10% increase for TPR, PPV, and F-Score compared to GA and hierarchical GA (HGA)-based PD methods, respectively. Furthermore, the accuracy rate rises by approximately 10–15% for each increase in the number of samples in the dataset.

Keywords: plagiarism detection; semantic analysis; optimization; quantum evolutionary algorithms

Citation: Darwish, S.M.; Mhaimed, I.A.; Elzoghbi, A.A. A Quantum Genetic Algorithm for Building a Semantic Textual Similarity Estimation Framework for Plagiarism Detection Applications. *Entropy* **2023**, *25*, 1271. <https://doi.org/10.3390/e25091271>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 13 July 2023

Revised: 16 August 2023

Accepted: 24 August 2023

Published: 29 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the last few decades, forensic linguistics has developed and used a type of language analysis that has helped put in place reliable ways to find plagiarism. Forensic linguistics research, which looks at how language affects the law, has shown that it is possible to figure out how likely it is that two or more texts were written independently. So, this analysis can be used as both a way to find out more and as proof, not just in legal situations but also in ethical ones [1–5]. Today, more and more cases of plagiarism are being reported. This could be because of one or more of the following: easy access to information; intense pressure to publish in academia for career advancement; lack of confidence and writing skills; or writing manuscripts quickly or under stress to meet a deadline. Also, because authors do not know what plagiarism is, they do not know that it is wrong to copy and paste word-for-word, even if they give a reference to the original text. Plagiarism detection (PD) methods look for text that is similar or the same between two or more

documents [6]. As most plagiarists reuse the text from other source papers to disguise plagiarism by changing terms with synonyms or paraphrasing, and maybe by rearranging the sentences, detecting plagiarism can be a very difficult process. On the other hand, it has inspired the creation of automated detection methods. Publishing houses have recently shown an eagerness to combat plagiarism [7].

Current PD approaches might have some shortcomings that reduce their effectiveness in detecting plagiarized texts. Here are the issues [8]: (1) Most algorithms can only identify word-for-word plagiarism, while others can detect random alterations. Online PDs fail or lose efficiency at greater degrees of complexity [9]. (2) Plagiarists have it easier with automatic translators, summarizers, and other tools. (3) Idea plagiarism detection tools are ineffective [10]. (4) Most PD methods may not detect structural alterations [11]. (5) Passage-level detections may lack linguistic, semantic, and soft computing tools. Syntactic, semantic, structural, and linguistic features must be evaluated to reveal hidden obfuscations. (6) Finally, there are not enough benchmark data to evaluate plagiarism techniques [12]. Plagiarism can take place in two ways: (1) Literal plagiarism, in which the plagiarist uses all or part of another person's work in their own. (2) Semantic plagiarism (intelligent) is when someone steals the content of another person's work but uses different words to describe it.

Plagiarism can be as simple as copying and pasting or as complicated as changing the words around. See [8] for more information. Textual documents can be divided into two basic types based on how similar their languages are or how different they are. These are monolingual and cross-lingual (CL) [13,14]. There are not many ways to find CL plagiarism because it is hard to find closeness between two text segments in different languages [14]. Unlike its multilingual counterpart, monolingual plagiarism detection focuses on pairs of languages that are mutually exclusive, such as English and English. This kind of detection approach constitutes the vast majority [14]. Detection may be further subdivided into the intrinsic type and the extrinsic type based on whether or not external references are used. Intrinsic detection is a document analysis technique that identifies potentially harmful files based only on linguistic features such as authorial style, paragraph structure, and section formulations [8]. In extrinsic detection, the suspect document is compared to a database or collection of source documents.

Optimization is an interesting area of research. In general, there are two types of optimization solution methods: deterministic and stochastic methods. Every method has its own pros and cons [15]. In deterministic methods, the initial values of the parameters and the conditions completely determine the model's output. Some randomness is built into stochastic methods [16]. Although various random approaches have been developed, such as swarm intelligence, genetic algorithms are becoming more popular for solving complex, large-scale optimization issues [17]. The quantum genetic algorithm (QGA) is an innovative evolutionary algorithm that combines quantum computing with conventional genetic algorithm technology. The approach can solve the same types of problems as the traditional genetic algorithm, but it does it far more quickly because of quantum parallelization and the entanglement of the quantum state, which speeds up the evolutionary process. A global search for a solution may be performed with quick convergence and a small population size by combining the probabilistic mechanism of quantum computing with the evolutionary algorithm. These methods have proven effective in a broad range of combinatorial and functional optimization problems [18–20].

1.1. Problem Statement

Even if that is true, putting plagiarism in a legal context is hard because you have to find strong proof that a suspicious text has been copied. When the text is copied and pasted word-for-word, it is usually enough to compare the suspect text to the possible source text to find the overlap. Most cases, though, are much more complicated. New ways to find plagiarism lead to new ways to avoid being caught, which in turn require new ways to find plagiarism. Plagiarism is when someone passes off someone else's work as

their own without giving credit. Plagiarism covers a wide range of things, from copying someone else's words to copying someone else's ideas. Recently, there have been many PD approaches based on semantic similarity and sentence-based concept extraction that may facilitate the discovery of paraphrases. To detect instances of plagiarism, several algorithms delve into the document's semantic concept by analyzing factors like the author's writing style, the structure of the paragraphs, the arrangement of the sections, etc. Obfuscated plagiarism cannot be prevented using these techniques, however.

1.2. Contribution and Methodology

In this paper, a modified PD algorithm is utilized to detect plagiarism using the semantic concept and the QGA. Adopting the QGA inside the PD model can facilitate the optimization of a similarity search. Furthermore, the QGA is employed to find sentences that briefly show the concept of the source document. On the other hand, semantic-level concepts are captured by applying semantic similarity metrics, which depend on the WordNet database for extracting semantic information. How successfully individuals are mapped to fitness metrics is what gives the QGA its usefulness in our context. Since all quantum individuals are reduced to a single solution during the measurement of the fitness function, the benefits disappear if the mapping is one-to-one. More individual-to-fitness mappings mean a higher potential diversity benefit for the QGA.

The remainder of this paper consists of the following sections: Some background on quantum genetic algorithms is briefly discussed in Section 2. The third section provides a literature review of relevant publications for the PD framework. The suggested approach is presented in Section 4. The assessment of the suggested technique, including results and discussion, is presented in Section 5. The study is concluded, and possible future directions are discussed in Section 6.

2. Preliminaries

In this section, we will go through the fundamental concepts of quantum genetic algorithms that will be used in the proposed framework. Primarily, evolutionary algorithms (EAs) are stochastic searches and optimization techniques inspired by the concepts of natural biological evolution. EAs have many advantages over more conventional optimization techniques, including their scalability, versatility, and independence from domain-specific heuristics. However, it is challenging to incorporate the characteristics of population diversity and selection pressure concurrently into EAs like the genetic algorithm (GA). In the face of rising selection pressure, the search narrows in on the best individuals in the population, but the resulting exploitation reduces genetic variety. The reason for this is that deterministic values are used in the definition of representations of EAs [20,21].

QGAs are a hybrid of conventional GAs and quantum algorithms. The superposition of quantum mechanical states, or "qubits", is the primary foundation for these. Here, instead of being represented as a binary string, for example, chromosomes are vectors of qubits (quantum registers). This means that a chromosome may stand in for a superposition of all possible states. The QGA is distinguished by its simultaneous capacity for quick convergence and global search. Quantum computing concepts and principles like qubits and a linear superposition of states form the basis of the QGA [22,23]. One way to express the status of a qubit is as follows:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

$$|\alpha|^2 + |\beta|^2 = 1 \quad (2)$$

The probabilities of the qubit being in the '0' and '1' states are specified by the expressions $|\alpha|^2$ and $|\beta|^2$, respectively, where α and β are complex numbers describing the probability amplitudes of the two states. Information on the states of a system may be stored in a system

of m -qubits. However, a quantum state collapses to a classical one upon observation [24]. For m -qubits, the representation is:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{bmatrix}, |\alpha_i|^2 + |\beta_i|^2 = 1, i = 1, 2, \dots, m \tag{3}$$

Consider a three-qubits system with three pairs of amplitudes:

$$\left[\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \middle| \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \middle| \begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{pmatrix} \right] \tag{4}$$

The current system status may be represented by:

$$\frac{1}{4}|000\rangle + \frac{\sqrt{3}}{4}|001\rangle - \frac{1}{4}|010\rangle - \frac{\sqrt{3}}{4}|011\rangle + \frac{1}{4}|100\rangle + \frac{\sqrt{3}}{4}|101\rangle - \frac{1}{4}|110\rangle - \frac{\sqrt{3}}{4}|111\rangle \tag{5}$$

This allows for eight possible states of information storage inside the three-qubit machine. Evolutionary computing with a qubit representation offers a more diverse feature than conventional approaches since it may express the superposition of states. While in classical representation at least eight chromosomes are needed to represent a state, just one qubit chromosome is needed to represent eight states. Convergence may also be attained using the qubit format. The qubit chromosome converges to a single state and loses its distinctive feature of diversity when either $|\alpha_i|^2$ or $|\beta_i|^2$ approaches 1 or 0. Therefore, it is possible for the qubit representation to have both exploratory and exploitation properties [24]. The structure of the QGA is described in Algorithm 1 [21,24].

Algorithm 1: QGA Procedure

```

Begin
  t = 0 Initialize Q(t)
  Make P(t) by observing Q(t) states
  Evaluate P(t)
  Save the best solution among P(t)
  While (not termination-condition) do
    Begin
      t = t + 1
      Make P(t) by observing Q(t - 1) states
      Evaluate P(t)
      Update Q(t) using quantum gates U(t)
      Store the best solution among P(t)
    End
  End
End

```

The QGA maintains a population of qubit chromosomes, $Q(t) = \{q_1^t, q_2^t, q_3^t, \dots, q_n^t\}$ at generation t , where n is the population size, m denotes the total number of qubits and indicates the string length of the qubit chromosome, and q_j^t is the definition of a qubit chromosome:

$$q_j^t = \left[\begin{bmatrix} \alpha_1^t & \alpha_2^t & \dots & \alpha_m^t \\ \beta_1^t & \beta_2^t & \dots & \beta_m^t \end{bmatrix} \right], j = 1, 2, \dots, n \tag{6}$$

$$|\Psi_{q_j^t}\rangle = \sum_{k=1}^{2^m} \frac{1}{\sqrt{2^m}} |S_k\rangle \tag{7}$$

$$U(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \tag{8}$$

where S_k is the k -th state represented by the binary string $(x_1x_2 \dots x_n)$, $x_i, i = 1, 2, \dots, m$, is either 0 or 1, and θ is the rotation angle. The effectiveness (fitness) of each solution is ranked. Then, among the available binary options, the $P(t)$ is chosen as the best possible starting point and saved. $Q(t)$ uses the binary solutions and the best-stored solution to construct an updated solution, which is then processed via the relevant quantum gates $U(t)$. To solve real-world issues, we may tailor the design of quantum gates to meet specific needs.

3. The State of the Art

Plagiarism often falls into one of three categories: (1) If the original texts are available, the study centers on comparing the suspect text(s) to the potential originals to uncover linguistic evidence to infer that the suspect text is truly a derivative or original; (2) if the source texts are unknown but plagiarism is suspected, the analysis focuses on determining whether the material in question is plagiarized or not based on its inherent stylistic evidence; or (3) if two or more texts are suspected of joint rather than individual composition, the linguistic study will center on determining whether any probable overlap between the texts is coincidental or the consequence of collaboration. Therefore, linguistic studies seek to determine whether instances of textual overlap across various papers are suggestive of plagiarism and if such overlap constitutes fraudulent behavior [1–5].

To aid in the building of the suggested model, this section discusses a few related PD models and plagiarism prevention efforts from the cited literature. Figure 1 shows the taxonomy of the existing PD models. In Ref. [25], the authors developed an approach based on Semantic Role Labeling (SRL) to determine semantic similarity between texts. All of WordNet’s ideas were combined into one node called the “topic signature node,” which instantly captures suspicious elements from documents. This method identifies copy–paste and semantic plagiarism, synonym substitution, phrase restructuring, and passive-to-active voice changes. Hence, since not all arguments impact the PD process, the fuzzy inference system should be used to increase the similarity score that argument weighting improves.

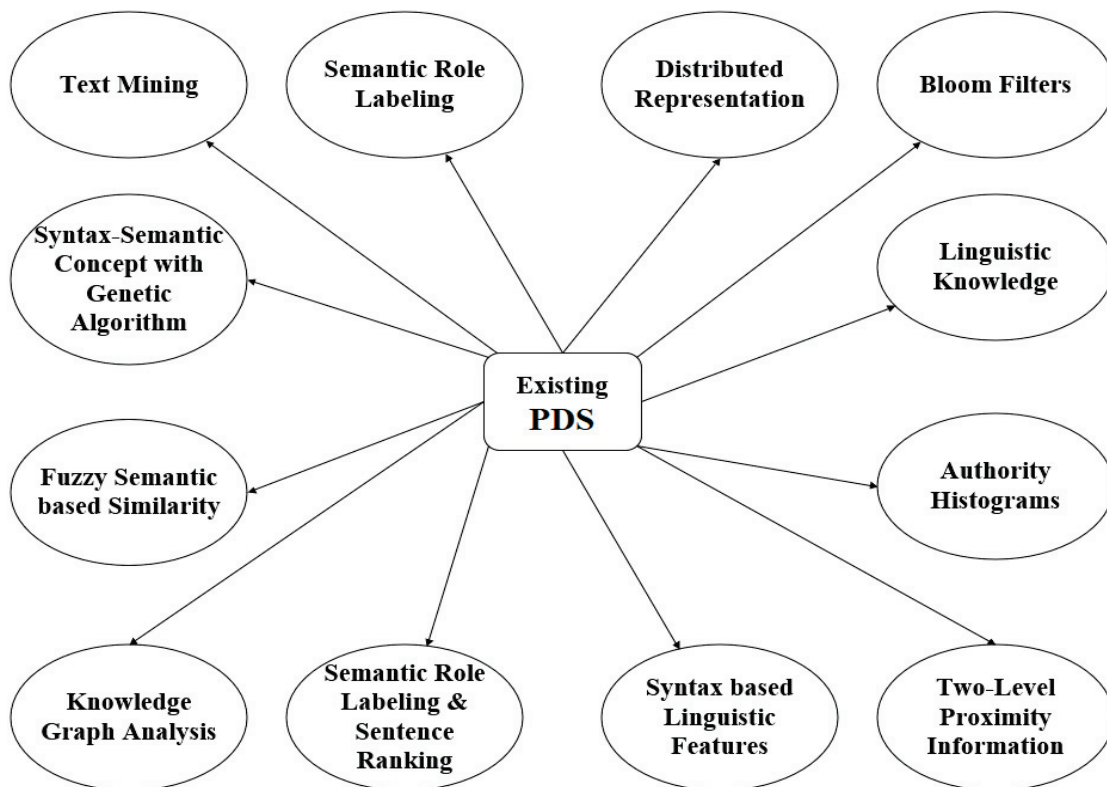


Figure 1. Taxonomy of existing PD models.

In Ref. [7], the authors studied sentence ranking for PD and SRL. Vectorizing the material generates suspicious and original sentence pairings. Pre-processing, candidate retrieval, sentence rating, SRL, and similarity detection are the five stages of the approach. The proposed technique leverages SRL to determine the semantic functions of each sentence word based on its verb. This depends on the word's semantic meaning. The algorithm recognizes copy-paste, close copy, synonym substitution, phrase reordering, and active/passive voice conversion faster and more accurately. It is unknown what degree of syntax is required to provide a thorough study of semantic roles and how the state of the art constrains SRL tagging and parsing performance.

In Ref. [26], the semantic and syntactic relationships between words are integrated. This strategy improves PD because it avoids picking source text sentences with high similarity to suspect text sentences with dissimilar meanings. It can identify copied text, paraphrases, sentence translations, and word structure changes. This approach cannot discriminate between active and passive sentences, however. In Ref. [27], the authors suggested a fuzzy semantic-based similarity approach for detecting obfuscated plagiarism. After feature extraction, the text characteristics are entered into a fuzzy inference system, which models semantic similarity as a membership function. Once the rules have been evaluated, the results are averaged to obtain a single score that indicates how similar two texts are. The technology detected literal and disguised plagiarism. The system cannot generalize and is not resilient to topological changes. Such modifications need rule-based adjustments and an expert to develop inference rules.

Another approach was suggested in [28] which treated document-level-text PD as a binary classification issue. The original source of a document was identified and that information was used to determine whether or not the document in question contained plagiarized content. The main parts are feature extraction, feature selection, and classification using machine learning. After pre-processing and filtering, part-of-speech (POS) tags and chunks removed extraneous data. The method investigated the influence of plagiarism categories and complexity on attributes and behavioral variances. The lack of a large database of manual plagiarism instances is a concern; thus, creating one is necessary for testing detection techniques.

The work in [8] presented another effort to identify plagiarism. The described study explores GA syntax-semantic concept extractions to detect idea plagiarism. Pre-processing, GA source sentence extraction, document level, and passage level are the four major components. Natural language processing (NLP) approaches are utilized for word-level extraction within documents. Sentence-based comparisons employing integrated semantic similarity metrics are employed in the passage-level identification step. Using passage boundary conditions, the passage level is detected. In the offered technique, the concept of plagiarism enforced via summarizations is emphasized. The results demonstrated substantial performance in catching plagiarized texts. Plagiarism may also occur via elaboration and paraphrase, etc., which the system cannot detect.

In order to find instances of plagiarism, the study in [29] constructed a cutting-edge system that relies on semantic properties. For each possible suspect and source phrase combination, the system generates a relation matrix that uses semantic characteristics to calculate the level of similarity. This study presents two weighted inverse distance and gloss Dice algorithms that illustrate different text qualities (e.g., synonyms) and develops a novel similarity metric for plagiarism detection, which overcomes the limits of the current features. In addition, this study examines the efficacy of individual characteristics in identifying copied works, combining the most effective ones by giving varying weights to their individual contributions to further improve the system's performance. The inverse weighted distance functions have a drawback in that the function must have a maximum or minimum at the data points (or on a boundary of the study region).

The study given in [30] outlines a three-stage process that, together, provides a hybrid model for intelligent plagiarism detection: initially, we cluster the data; then, we create vectors inside each cluster according to semantic roles, normalize the data, and compute

a similarity index; and lastly, we use an encoder–decoder to provide a summary. For the purpose of choosing the words to be used in the production of vectors, K-means clustering, which is calculated using the synonym set, has been proposed as a method. Only if the last stage’s estimated value is greater than a threshold value is the following semantic argument evaluated. A brief description is generated for plagiarized documents if their similarity score is high enough. The experimental results demonstrated the effectiveness of the strategy in identifying not only literal but also connotative and concealing forms of concept copying. However, long sequences take a long time to process because of the slowness of the neural network’s processing and the difficulty of training it if activation functions are used. Finally, it has problems like gradient vanishing and explosions.

In Ref. [31], the authors introduced an efficient method for determining the structural and semantic similarity between two publications by only analyzing a subset of the material of each document instead of the whole thing. To improve plagiarism detection regardless of word order changes, a collection of remarkable keywords and different combinations are used to compute similarity. The importance of a word varies depending on where in the article it appears. As a final step, a weighted similarity is determined using an AHP (Analytical Hierarchy Process) model. It was shown that the suggested method outperformed its competitors in terms of runtime and accuracy for detecting semantic academic plagiarism. One potential drawback of the AHP is the high number of pairwise comparisons it requires. This is due to the fact that comparing each criterion and then each option with regard to a given criterion is required.

In Ref. [32], the authors offered an approach to detecting two common forms of paraphrased text: those that involve the use of synonyms and those that use the reordering of words in plagiarized sentence pairs. They introduced a three-stage technique that makes use of context matching and pertained word embedding to detect instances of synonymous replacement and word reordering. Their experiments revealed that the Smith–Waterman method for plagiarism detection combined with ConceptNet batch-pertained word embedding yields the highest scores. Methods to determine paraphrase styles for plagiarism detection may be used from this study to supplement similarity reports from existing plagiarism detection systems. Even though it is the most sensitive technique for detecting sequence similarity, the Smith–Waterman approach does not come without its price. Time is a major restriction, as conducting a Smith–Waterman search requires a lot of processing power and time.

Two methods for identifying external plagiarism are provided in [33]. Both methods use a bag-of-words strategy-based two-stage filtering procedure, first at the document level and then at the sentence level, to reduce the search area; only the outputs of both filters are then evaluated for plagiarism. One uses the WordNet ontology and the term frequency–inverse document frequency (TF-IDF) weighting technique to create two structural and semantic matrices; the other uses a pre-trained network technique of words embedding fast text and TF-IDF weighting to create the same outcome. After forming the aforementioned matrices, the structural similarity of the weighted composition and the Dice similarity are used to determine the degree of similarity between the pairs of matrices representing each phrase. The similarity between the suspect text and the minimum criterion is used to classify documents as plagiarism or non-plagiarism. Using the PAN-PC-11 database, the authors conducted experiments to determine whether or not a word embedding network, as opposed to the WordNet ontology, would be more successful in detecting instances of extrinsic plagiarism. However, TF-IDF weighting does have certain restrictions. It may be time-consuming for large vocabularies since it calculates document similarity directly in the word-count space. It assumes that evidence for similarity may be found in the counts of various terms. One potential problem with the adaptable layout described above is that WordNet’s meaning and scope might quickly diverge from one another. We cannot be sure that we will be encoding the same relationships or that we will be covering the same conceptual ground [34,35].

In Ref. [36], the authors created a new database that contains all the characteristics that indicate various linguistic similarities. As a solution to textual plagiarism issues, the developed database is offered for use in intelligent learning. The produced database is then used to propose a deep-learning-based plagiarism detection system. During development, many deep learning techniques, including convolutional and recurrent neural network topologies, were taken into account. To assess the efficacy of the presented intelligent system, comparison research was conducted using the PAN 2013 and PAN 2014 benchmark datasets. In comparison to state-of-the-art ranking systems, the test findings demonstrated that the suggested system based on long short-term memory (LSTM) ranked first. However, LSTMs are easy to overfit and are sensitive to different random weight initializations.

Using the fuzzy MCDM (multi-criteria decision-making) technique, the research in [37] compared and contrasted many academic plagiarism detection strategies and offered guidelines for creating effective plagiarism detection tools. They described a framework for ranking evaluations and analyzed the cutting-edge methods for detecting plagiarism that may be able to overcome the limitations of the state-of-the-art software currently available. In this way, the research might be seen as a “blueprint” for developing improved plagiarism detection systems. An innovative and cutting-edge technique known as compressive sensing-based Rabin Karp is offered for use in the system presented in [38]. This technique calculates both syntactic and semantic similarities between documents using a sampling module to shrink the dataset and a cost function to identify document repetition. Yet, simply applying the hash function based on the generated table may result in cases where the hash codes for the pattern and text are the same, yet the pattern’s characters do not match those in the text. For current surveys that include the most up-to-date research in the plagiarism detection area, please refer to [39,40].

A novel plagiarism detection approach is presented in [41] to extract the most useful sentence similarity features and build a hyperplane equation of the chosen features to accurately identify similarity scenarios. The first phase, which contains three steps, is used to pre-process the papers. The second phase is dependent on two different strategies: the first strategy relies on the standard paragraph-level comparison, while the second strategy relies on the calculated hyperplane equation utilizing Support Vector Machine (SVM) and Chi-square methods. The best plagiarized segment is taken out in the third step. On the whole test corpus of the PAN 2013 and PAN 2014 datasets, the recommended approach attained the best values of 89.12% and 92.91% of the Plagdet scores and 89.34% and 92.95% of the F-measure scores, respectively.

The present plagiarism detection solutions now on the market compare plagiarism only when the input document includes text, despite the fact that there are a number of tools available that address the issue of plagiarism using various methodologies and features. However, when the input document is an image, the techniques currently in use do not check for plagiarism. The authors in [42] suggested a tool that searches both the text and text hidden in images using an exhaustive searching approach. The project’s suggested tool compares the input document’s content to that of websites and returns findings on how similar they are. The source and suspect papers are in two different languages, making it difficult to identify cross-lingual plagiarism (CLP). In this context, a number of solutions to the issue of CPD in text documents were proposed. To obtain comparability metrics, the authors in [43] employed the one-gram and tri-gram of the pre-processed text. The models are constructed using five ML classifiers: KNN, Naive Bayes, SVM, Decision Tree, and Random Forest. The trial demonstrates that KNN, RF, and other models offer superior outcomes versus other models.

Commercial plagiarism detection tools are accessible online for purchase or subscription. EVE2, Plag Aware, Write Check, Turnitin, and Ithenticate are some of the most well known [44]. Turnitin is an online similarity detection service that compares submitted papers to various databases using a proprietary algorithm to check for possibly plagiarized material. In addition to scanning its own databases, it has licensing arrangements with significant academic private databases. Turnitin does not deal with the causes of

academic integrity problems, and so it does not fix them. Instead, it might give students the impression that they are being held accountable for cheating from the very first day of class or that their work is being used against them and others without their permission. iThenticate is a plagiarism prevention tool that assesses written material (such as journal article manuscripts, proposals, research reports, theses, and dissertations, among other things) against millions of published works that are accessible online and via paid databases. The following are some benefits of iThenticate: The finest tool for detecting plagiarism in academic writing is iThenticate, which employs cutting-edge algorithms to evaluate submitted text against a huge library of scholarly publications.

Despite decades of study, PD might be strengthened to better prevent intellectual property theft. Still, PD should account for things like running time and computational complexity. The available PD approaches are not all suitable to be employed in all applications. To address these issues and outperform competing methods, a model combining semantic idea extraction and the QGA for optimizing similarity search has been proposed. The QGA is structurally similar to classical genetic algorithms, with the exception that quantum gates and quantum superposition are used to construct the initial and updated populations, with consideration given to the adaptation of such operators to meet GA-based PD issues. One clear benefit of a QGA is that its population tends to be more diverse than that of a non-QGA. To put it another way, a quantum population may be exponentially greater than its “size” in the classical world. Only one possible solution may be represented by each individual in a classical population. Each “individual” in a quantum population is a superposition of many different possible solutions. In this sense, the population of a quantum system is far greater than that of a classical system.

4. The Proposed QGA-Based Plagiarism Detection Model

This section presents the suggested model for QGA-based idea (semantic) extraction for plagiarism detection. PD exploits document notions at several structural levels for document-level (DL) and passage-level (PL) detection. QGA-based sentence scoring is examined for sentence-level extraction. The DLD stage captures nouns and verbs using natural language processing (NLP) methods. In the PLD phase, phrase-based assessments utilizing a joint similarity measure with WorldNet detect plagiarized sentence pairings. We decided to use a quantum-inspired evolutionary algorithm to solve the PD problem because of the many benefits of quantum-inspired metaheuristics. (1) With quantum gates and quantum parallelism, it is possible to compute all possible values of a given variable simultaneously, which not only enhances the quality of the result but also drastically shortens the search time. (2) The use of quantum superposition and quantum gates to represent individuals in a population results in (a) more diversity, (b) enhanced search capacity, (c) faster and more accurate convergence, and (d) efficient escape from local optima. Due to the limited number of individuals, the method may quickly and efficiently probe the search space for a global solution, even if it only contains a single element. (3) There is a balance between diversification/intensification and exploration/exploitation [18,21,22]. Figure 2 shows the suggested framework, and each module is discussed in the following subsections.

4.1. Pre-Processing and Document Representation

The database, which includes both source and suspect documents, is pre-processed in the first module. The steps included in this section are as follows.

4.1.1. Sentence Segmentation and Tokenization

First, suspicious (X_{susp}) and source (X_{src}) documents are sentence-segmented. Text segmentation is a pre-processing procedure that divides text into meaningful components like sentences or words. The document is split into sentences. Then, source and suspect phrases are tokenized. Punctuation and capitalization are eliminated [7,8,27].

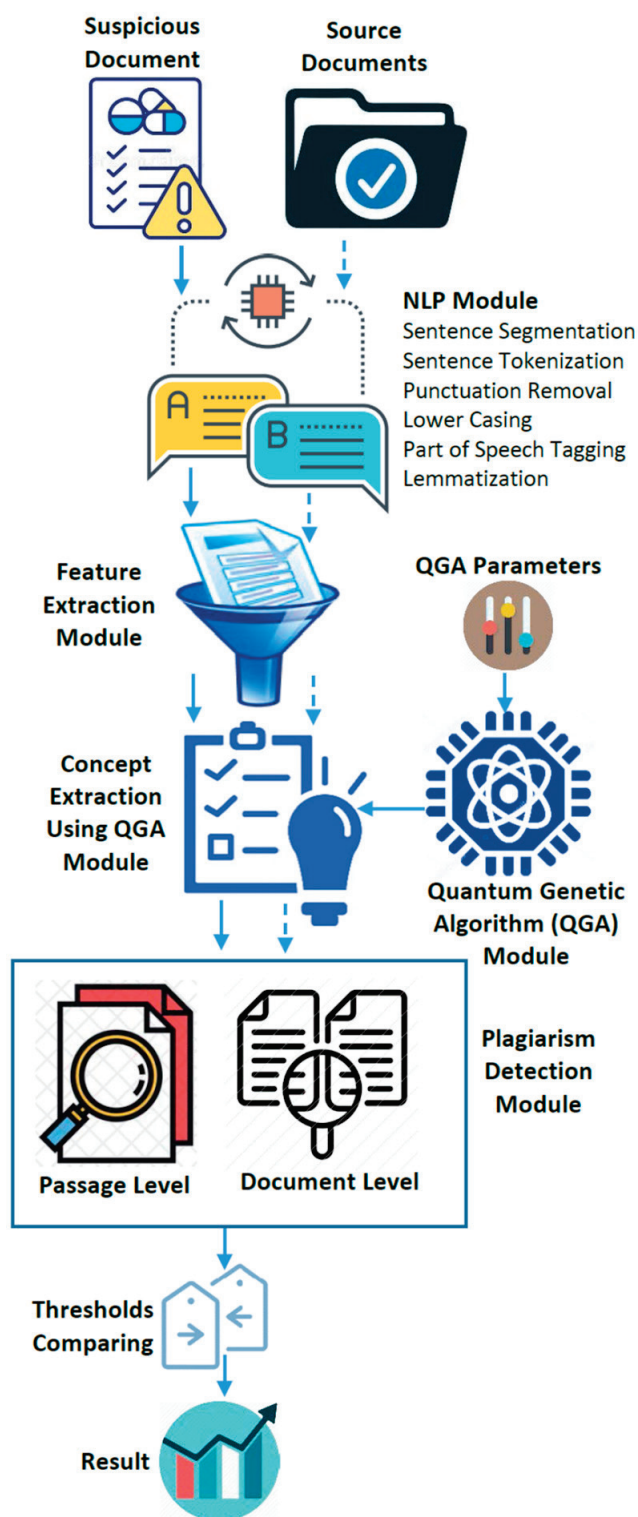


Figure 2. The proposed plagiarism detection framework using the QGA.

4.1.2. Part-of-Speech Tagging and Lemmatization

After the pre-processing step, tokenized words are employed for the part-of-speech (POS) tagging of suspect and source tokenized phrases. Each word is labelled as a noun, verb, adjective, preposition, etc. Noun, verb, adjective, and adverb tags are the only semantic tags that are kept. Conjunctions, prepositions, articles, pronouns, and determinants were taken out of the sentence, along with anything else that did not add meaning. By conserving

memory and speeding up processing, removing such words improves accuracy and time. Lemmatization reduces words to their dictionary base forms and allows for comparisons. The Stanford Log-Linear Speech Tagger and WordNet Lemmatizer were employed for POS tagging [8,45]. The pre-processed suspicious sentence in X_{susp} is S_{susp} , while the source sentence in X_{src} is S_{src} . Each pre-processed source and suspect tokenized sentence includes lemmatized and POS-tagged words available for feature extraction [7,8,27,45].

4.1.3. Feature Extraction

The pre-processed source and suspect documents are a collection of tokenized sentences, and the Vector Space Model (VSM) with term-frequency-inverse sentence frequency ($tf - isf$) weighting reflects the vocabulary of the lemmatized and POS-tagged words contained in these documents [8]. ($tf - isf$) is a metric developed for use in information retrieval (IR) that attempts to quantify a word’s significance within the context of a phrase [28,45–48]. The $w(t, S)$ weight is calculated using:

$$tf(t, S) = f(t, S) \tag{9}$$

$$isf(t, X) = \log \frac{|X|}{|\{S \in X; t \in S\}|} \tag{10}$$

$$w(t, S) = tf(t, S) * isf(t, X) \tag{11}$$

The number of times a term t appears in any generic sentence S is denoted by term frequency $tf(t, S)$. The term-inverse sentence frequency (isf) is used to highlight the fact that the computation is performed over individual sentences as opposed to whole documents, where X is the collection of all sentences found in the provided documents. Sentence vectors for the source and suspect sentences are denoted by \vec{s}_{src} and \vec{s}_{susp} , respectively.

4.2. The Quantum Genetic Algorithm for Extracting Sentence Concepts

Concept extraction using the QGA is feasible when the documents have been pre-processed and expressed in $tf - isf$ weight form. The documents’ syntactic concepts are derived from their respective structural levels. Paragraphs, phrases, sentences, and keywords are all ways in which these ideas may be found across a document [49]. The suggested approach starts by using sentence scoring methods with the QGA to extract sentence-based ideas from the original documents. In order to simplify the content of a lengthy text into a few carefully chosen sentences, the QGA is used.

4.2.1. Population Initialization

Pre-processed source sentences, each of which will be given a fixed score, are the QGA’s input. Static scores, together with relevance and theme scores, may be calculated. Sentence weights are assigned to each S_{src} in X_{src} by extracting features from X_{src} based on $w(t, S)$. Both the relevance score and the thematic score may achieve this [47,48].

- Relevance Score

The relevance score expresses S_{src} using $tf - isf$ weights, which is the source sentence’s pre-processed word count:

$$Rel(S_{src}) = \frac{\sum_{i=1}^{|S_{src}|} w(t_i, S_{src})}{|S_{src}|}; Rel(S_{src}) \in [0, 1] \tag{12}$$

where $w(t, S_{src})$ denotes the sum of the $tf - isf$ weights of each word t in S_{src} and $|S_{src}|$ is the source sentence length.

- Thematic Score

The thematic score is calculated by retrieving and sorting the words from the pre-processed X_{src} . The top L words are then saved in the X_{src} keyword set $kw(X_{src})$:

$$Thm(S_{src}) = \frac{|kw(S_{src})|}{L};$$

$$Kw(S_{src}) = \{t|t \in S_{src} \wedge t \in kw(X_{src})\}; Thm(S_{src}) \in [0, 1] \tag{13}$$

where $|kw(S_{src})|$ is the number of words between $kw(X_{src})$ and S_{src} in X_{src} and $kw(X_{src})$ has L words. After calculating the relevance and thematic scores, $Stat(S_{src})$ is calculated.

$$Stat(S_{src}) = Rel(S_{src}) + Thm(S_{src}); Stat(S_{src}) \in [0, 2] \tag{14}$$

The S_{src} with the associated $Stat(S_{src})$ will be employed for building the QGA population. A population with N chromosomes is randomly chosen. A chromosome is conceptually equivalent to a quantum register made up of a string of m -qubits. A quantum chromosome's structure can be seen in Figure 3. All qubit amplitudes may be conveniently set to the value $1/\sqrt{2}$ [22] to generate the starting population. This implies that each of the possible quantum superposition states is equally represented in a chromosome. To begin, we create N quantum registers and give them the labels $Reg1_0$ through $Reg1_{N-1}$, where N is the total number of individuals in the population. Then, each of these registers is layered on top of one another to create a superposition of all potential individuals. This means that each register is capable of storing all potential individuals. The next step is to apply the fitness function to each of the N quantum registers, and then store the results in a second set of N quantum registers, which are designated by the labels $Reg2_0$ through $Reg2_{N-1}$. The application of the fitness function will result in an entanglement being produced between the first set of registers and the second set of registers.

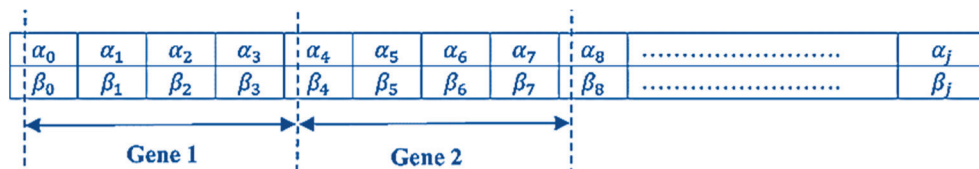


Figure 3. Quantum chromosome structure.

4.2.2. Fitness Function Computation

The quality of each quantum chromosome in the population is quantified at this stage in order to facilitate reproduction. A superposition of all the individuals who may have been there is included in each of the initial registers. Because of this, the data stored in each of the second registers is a superposition of all of the feasible fitnesses. Even if every individual was examined, which led to the generation of every fitness, there was still only one instance of the fitness function that needed to be applied to each register. The parallelism of quantum mechanics may be shown here [22–24]. The optimal solution would be to evaluate the highest fitness in register $Reg2_i$, which would then cause register $Reg1_i$ to collapse into a superposition of perfect individuals. The outcome of a measurement is completely unpredictable, and the probabilities are based on the amplitudes of the probabilities. Therefore, the likelihood of achieving a maximum level of fitness ($Fit(C)$) is precisely the same as the probability of accidentally producing an ideal individual. In our case, the fitness function ($Fit(C)$) is calculated as follows:

$$Fit(C) = \sum_{i=1}^{|C|} Tot(S_{src}) \tag{15}$$

in which, a dynamic cohesiveness factor is generated for each phrase in C and supplemented using $Stat(S_{src})$. The cohesiveness factor determines sentence relatedness [50]. Cosine

similarity measures lexical cohesiveness [51]. Cosine similarity between the source sentence vectors is calculated first.

$$\text{Cos}\left(\vec{S}_{srci}, \vec{S}_{srcj}\right) = \frac{\vec{S}_{srci} \cdot \vec{S}_{srcj}}{\left\| \vec{S}_{srci} \right\| \left\| \vec{S}_{srcj} \right\|}; \forall i, j \vec{S}_{srci}, \vec{S}_{srcj} \in C \quad (16)$$

$\text{Cos}\left(\vec{S}_{srci}, \vec{S}_{srcj}\right)$ denotes the cosine similarity between an S_{src} vector pair $\left(\vec{S}_{srci}, \vec{S}_{srcj}\right)$ such that each sentence is a chromosomal C element. Cosine similarities are calculated and stored in a symmetric matrix with diagonal entry 1. The S_{srci} sentence cohesion factor is then calculated.

$$\text{Coh}(S_{srci}) = \frac{\sum_{j=1, j \neq i}^{|C|} \text{cos}(S_{srci}, S_{srcj})}{\max\{(S_{srci}, S_{srcj})\}}, \forall ij = \{1, 2, \dots, |C|\}, i \neq j \quad (17)$$

To avoid self-similarity, $i \neq j$ is used; otherwise, the denominator is 1. After computing the sentence cohesiveness factor, the total score for each source sentence $Tot(S_{src})$ is determined.

$$Tot(S_{src}) = Stat(S_{src}) + Coh(S_{src}) \quad (18)$$

Using quantum selection and crossover, the fitness value C is used to build the next generation.

4.2.3. Quantum Selection and Crossover

Our initial population will be represented by a set of N paired registers, with half of the registers carrying fitness values and the other half having the superposition of individuals based on those fitness values. Normal procedures are followed upon crossover. The information included in the register $Reg1_i$ is combined with the information found in the register $Reg1_j$. Since both registers already contain a superposition of individuals, we obtain two additional superpositions as a result. In particular, if $Reg1_i$ contains all individuals with fitness values $Fit(C_i)$ and $Reg1_j$ contains all individuals with fitness values $Fit(C_j)$, then the superposition of all individuals that may be generated by crossing at the given location is achieved. The N registers ($Reg1_0$ through $Reg1_{N-1}$) will then be subjected to the fitness function. The second set of registers is used to store the results and is entangled with the first set of registers in the same way that the initial population was. The next step is to take a measurement. This reduces the number of individuals from $Reg1_0$ through $Reg1_{N-1}$ to only those with the measured fitness, and it also collapses the superimposed fitness values to a single value. The generation ends when a selection is performed based on the calculated fitness values. Any desired mutations may be included [49,52,53].

Obtaining a result is the last action when the termination condition is met. The final product will be N pairs of registers, where each pair's first register has a set of superimposed individuals with the same fitness value, and is entangled with the second register of the pair, which has the measured fitness value. A measurement of the first register will be able to identify one of the individuals as having the specified fitness level. This provides the effect that was sought, which is a single individual of the fitness level that was specified. It is necessary to conduct an observation on each qubit if we are to successfully utilize the superposed states of qubits (measuring chromosomes). Because of this, we are able to obtain a traditional chromosome, as illustrated in Figure 4. The purpose of this is to make it possible to evaluate each quantum chromosome. A final set of best C is generated, where the highest $Fit(C)$ is picked, representing the best source sentence set S_{src_sel} [8].

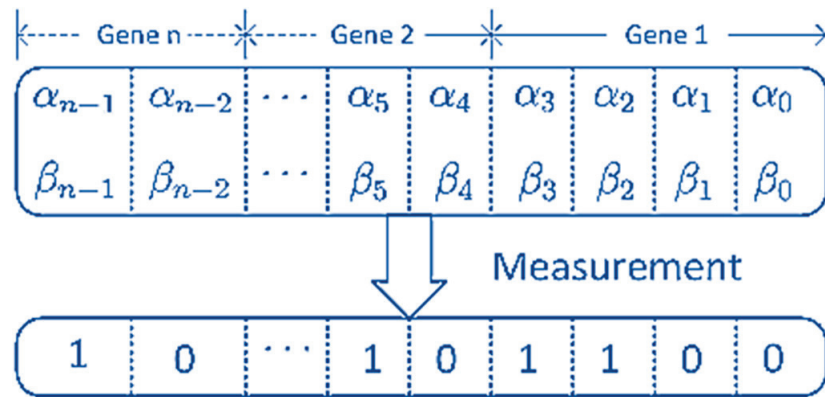


Figure 4. Measured chromosome.

The interference operation allows for the modification of specific amplitudes in order to optimize performance. It mostly entails shifting the state of each qubit in the direction of the optimal solution’s value. This is important for narrowing down the search for the best option. The amplitudes (α_i, β_i) and the value of the corresponding bit in the reference solution determine the angle of the rotation that may be carried out using a unit transformation. Early convergence may be prevented by appropriately setting the rotation angle $\delta\theta$. The direction of the change is determined by the values of α_i, β_i , and the qubit inserted at location i in the individual (chromosome) being altered, all of which are typically estimated experimentally. The population $Q(t)$ is revised when the qubits making up individuals are rotated using quantum gates. Equation [22] explains the rotation method that is employed:

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\delta\theta_i) & -\sin(\delta\theta_i) \\ \sin(\delta\theta_i) & \cos(\delta\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix} \tag{19}$$

where $\delta\theta_i$ is the rotation angle of each quantum chromosome’s qubit quantum gate i , as illustrated in Figure 5 [53]. As stated in [22], it is frequently taken via a lookup table to guarantee convergence; see Table 1.

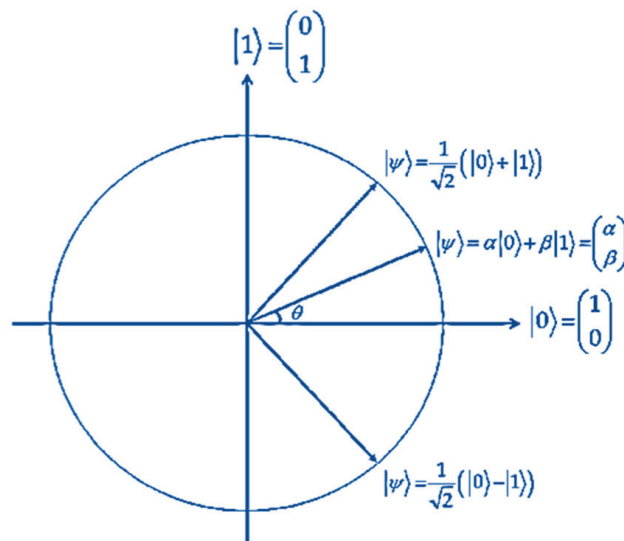


Figure 5. Qubit transformations with Hadamard gate.

Table 1. Lookup table for quantum gate rotation [22].

$x_i \ b_i$	$f(x) > f(b)$	$\delta\theta_i$	$S(a_i, b_i)$			
			$a_i \cdot b_i > 0$	$a_i \cdot b_i < 0$	$a_i = 0$	$b_i = 0$
0 0	0	0.001π	-	+	\pm	\pm
0 0	1	0.001π	-	+	\pm	\pm
0 1	0	0.08π	-	+	\pm	\pm
0 1	1	0.001π	-	+	\pm	\pm
1 0	0	0.08π	+	-	\pm	\pm
1 0	1	0.001π	+	-	\pm	\pm
1 1	0	0.001π	+	-	\pm	\pm
1 1	1	0.001π	+	-	\pm	\pm

The i -th bits of x and b (the optimal solution) are denoted by x_i and b_i , respectively. The rotation angle θ_i has a sign that may be written as $S(a_i, b_i)$, and f is the fitness function. Using the lookup table, we can see that this method increases the amplitudes of poor qubits according to angle $\delta\theta_1 = 0.08\pi$, while decreasing the amplitudes of good qubits according to angle $\delta\theta_2 = 0.001\pi$. Quantum bit amplitudes are adjusted in accordance with the signs of the amplitudes, the optimal solution, and the solution extracted with the respective container. Because reducing amplitudes only helps to correct stochastic mistakes, preventing genetic drift and guaranteeing genetic diversity, it stands to reason that $\delta\theta_1 > \delta\theta_2$ [22].

4.3. The Document-Level Plagiarism Detection Phase

After selecting the important sentence-level ideas, the word-level concepts are retrieved. As most document ideas are transmitted using nouns and verbs, S_{src_sel} picks out nouns and verbs [8]. S_{susp} collects nouns and verbs from each X_{susp} . The number of common source and suspect word ideas is utilized to detect document-level plagiarism in the DLD phase. If the count value is more than the threshold ϵ , the document is deemed to be plagiarized. After DLD, suspicious source document pairings that are determined as plagiarized proceed to the PLD phase.

4.4. The Passage-Level Plagiarism Detection Phase

Semantic concept extractions are used for passage-level comparisons to calculate semantic similarity. Plagiarized suspicious source pre-processed document pairings are given to PLD. In this step, suspicious sentences are compared to S_{src_sel} . The source sentences result from QGA’s sentence-level idea extraction. Since sentences are pre-processed, unnecessary words are deleted and each word is tagged (POS tag). For sentence comparisons, WordNet extracts semantic-based word synsets. Synsets are groups of semantically similar data elements [28]. POS information is compared to determine if a suspicious source sentence pair (S_{susp}, S_{src_sel}) is plagiarized. That implies only comparing nouns and verbs, etc. Comparing word classes seems meaningless.

For each suspicious source word pair (w_q, w_k) , WordNet is used to derive the synset lists W_{q_syn} of w_q and W_{k_syn} of w_k of each word. Only synsets in the same POS class as the word are retrieved for these lists. Common words between suspicious source sentence pairs (S_{susp}, S_{src_sel}) are calculated and kept in the list *Count*. Suspicious word w_q is checked for in S_{src_sel} . The synonyms of w_q ’s are taken from WordNet if it is not in S_{src_sel} . $Syns(w_q)$ represents a suspicious word’s synonym list. Common words between $Syns(w_q)$ and S_{src_sel} are calculated and added to *Count*, which includes the number of frequent terms or synonyms between suspicious and source sentence pairs. Using threshold τ , a suspicious source sentence combination is found to be plagiarized or not. If WordNet’s similarity score is higher than the set value, the phrases are plagiarized [54]. Algorithm 2 outlines the main steps of the suggested technique.

Algorithm 2: QGA for Plagiarism Detection

Input: Dataset X_{src} ; Suspicious Document X_{susp} ; QGA Parameters, WordNet

```

1- while  $n < \text{size of documents}$  do
2-    $S \leftarrow \text{Sentence Segmentation } (X_{src})$ 
3-    $y \leftarrow 0$ 
4-   While  $y < S \neq \text{NULL}$  do
5-      $T \leftarrow \text{Tokenization } (S)$ 
6-      $z \leftarrow 0$ 
7-     while  $z < \text{size of } T$  do
8-        $M \leftarrow \text{POS Tagging } (T)$ 
9-        $N \leftarrow \text{Lemmatization } (M)$ 
10-       $z++$ 
11-    end
12-     $\text{tf-isf } (N)$ 
13-     $y++$ 
14-  end
15-   $n++$ 
16- end
17-  $t \leftarrow 0$ 
18- while termination condition not satisfied do
19-    $t \leftarrow t+1$ 
20-   Call Algorithm 1 // QGA Procedure
21-   Return  $\text{Best\_Pop} \leftarrow \text{New\_Pop}$  // Store the best solution among  $P(t)$ 
22- end
23-  $\text{sim}_1 \leftarrow \text{sum of words in } X_{susp}$  // the number of common word-level concepts in  $X_{susp}$ 
   that collects nouns and verbs
24-  $\text{sim}_2 \leftarrow \text{sum of words in } X_{src}$  // the number of common word-level concepts in  $X_{src}$ 
25- If  $\text{sim}_1 - \text{sim}_2 > \epsilon$ 
26-   Doc. Status = =Plagiarized
27- end
28- For each suspicious-source word pair  $(w_q, w_k)$  // To compute the semantic similarity
29-   - WordNet is used to derive the synset lists  $W_{q\_syn}$  of  $w_q$  and  $W_{k\_syn}$  of  $w_k$ 
30-   of each word.
31-   - Only synsets in the same POS class as the word are retrieved for these lists
32- end
33-  $\text{Count} \leftarrow \text{The common words between the compared suspicious-source sentence}$ 
   pair  $(S_{susp}, S_{src\_sel})$  //  $S_{src\_sel}$  is the best set of selected source sentences
   extracted from QGA' procedure
34- If  $\text{count} > \tau$ 
35-   Doc. Status = = Plagiarized
36- end
37- Else
38-   Doc. Status = = not plagiarized
39- end
40- Output = Doc. Status

```

5. Experimental Results

The effectiveness and reliability of the suggested model were evaluated using MATLAB implementation and QuTiP package Release 4.7.1 [55] for building quantum genetic algorithm modules. The prototype verification method was developed in a modular form and tested on a Dell™ Inspiron™ N5110 Laptop, Dell computer Corporation, Texas, which included the following specifications: 64-bit Windows 7 Home Premium, 4.00 GB RAM, Intel(R) Core(TM) i5-2410M CPU, 2.30 GHz. Benchmark datasets [56] provided the source for these data. Table 2 displays the proportion of plagiarized to original papers in each group of suspects. The Summary Obfuscation (SO) training and test datasets provided by the PAN13-14 text alignment task were utilized to evaluate the plagiarism detection

(PD) model in Figure 6. Different performance metrics, as shown in Table 3, were employed to assess the performance of the suggested model [57]. All test data examples may be predicted by a binary classifier as positive or negative. Table 4 displays the current QGA setup settings.

Table 2. Data statistics of PAN 13-14.

PAN 13-14 Dataset	Files					
	Training Data		Testing Data-1		Testing Data-2	
	Source	Suspicious	Source	Suspicious	Source	Suspicious
Non-plagiarized (NP)	947	-	97	-	949	-
Plagiarized (P)	238	-	24	-	236	-
Total	1185	237	121	102	1185	237

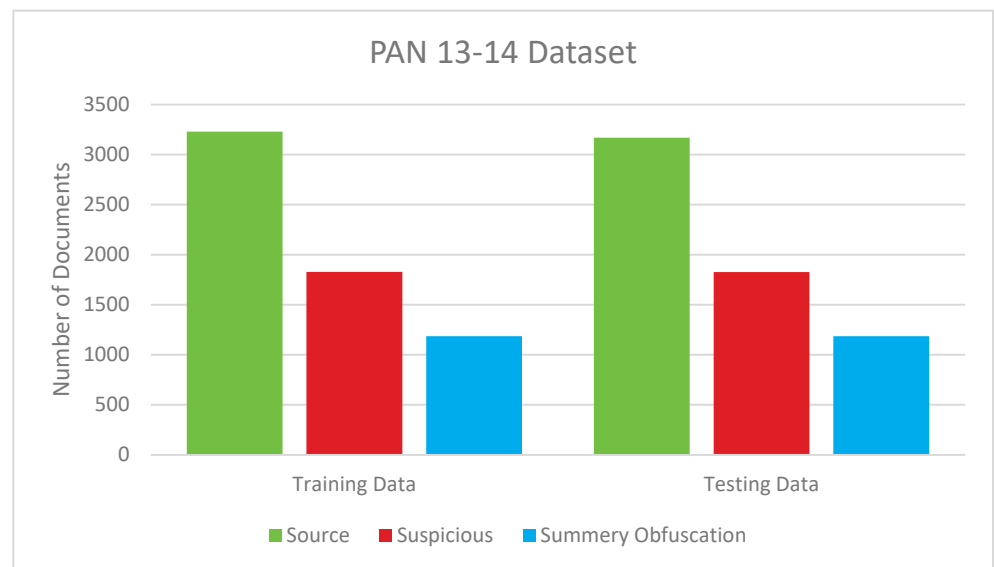


Figure 6. Size of training and testing datasets.

Table 3. Performance metrics [50].

Define	Metric
Accuracy	$ACC = \frac{TP+TN}{TP+TN+FN+FP} = \frac{TP+TN}{P+N}$
Sensitivity or Recall or Hit Rate or True Positive Rate	$TPR = \frac{TP}{TP+FN} = \frac{TP}{P}$
Precision or Positive Predictive Value	$PPV = \frac{TP}{TP+FP}$
F-Score	$F-Score = \frac{2*PPV*TPR}{PPV+TPR} = \frac{2TP}{2TP+FP+FN}$

Table 4. QGA parameters settings.

Parameter	Value
Population Size (N)	50
Max. No. of Generations (Max_Gen)	10
Selection	Highest Fitness
Probability of Crossover	0.7
Probability of Mutation	0.3
Termination Condition	Max_Gen

5.1. Experiment 1: A Comparative Study of the Different Types of GA

To validate the benefits of implementing the proposed model (semantic concept extractions with the QGA) for PD, this experiment compares the suggested model with

related PD models that include syntax–semantic concept extractions with the GA [8] and syntax–semantic concept extractions with the hierarchical GA (HGA) [58]. The experiment was reported for datasets PAN13-14 in terms of TPR, PPV, and F-Score for all the used datasets. It is observable that the results of the QGA-based PD model are better than those that depend on both the HGA and the traditional GA. Table 5 reveals the superiority of the suggested model for document detection in terms of TPR, PPV, and F-Score. The recommended PD model achieves an approximately 20%, 15%, and 10% increase for TPR, PPV, and F-Score compared to the GA and HGA, respectively. These results might be explained by the fact that the proposed methodology uses semantic idea extraction to identify instances of plagiarism. Additionally, using the QGA aids in efficiently removing the non-plagiarized documents. It also lowers the number of PLD-phase sentence comparisons.

Table 5. Comparison between semantic concept extractions with the QGA, HGA, and GA methods (P: plagiarized, NP: non-plagiarized) for the PAN13-14 dataset. (Average of testing data-1 and testing data-2)

PD Methods	TPR Value		PPV Value		F-Score	
	P	NP	P	NP	P	NP
Semantic concept extractions with the QGA (proposed model)	1	0.99	0.99	0.98	0.99	0.98
Semantic concept extractions with the HGA [58]	0.98	0.97	0.98	0.95	0.98	0.96
Semantic concept extractions with the GA [8]	0.97	0.95	0.96	0.93	0.97	0.94

5.2. Experiment 2: QGA-Based PD Model Validation

The purpose of these tests was to verify the QGA’s usefulness in the features selection module by measuring its effect on accuracy. In this investigation, the adaptive feature selection technique is used to focus in on the most relevant details for enhancing the PD model. It compares the GA-based PD and the proposed QGA-based PD model for different datasets and provides a confusion matrix for all used datasets. The definitions regarding the confusion matrix are summarized in Table 6 [8]. Tables 7 and 8 reveal that the QGA for PD achieves better results with the confusion matrix compared to the GA procedure. The QGA produces an approximate increase (of about 5%, on average) in plagiarism detection compared to the GA. The way the QGA works is that it facilitates the capturing of the non-plagiarized documents efficiently. Moreover, the QGA decreases sentence comparison numbers in the PLD. Utilizing the space’s desirable features is a discriminatory way to highlight individual differences. The feature selection issue is often multi-modal since there are often numerous optimal solutions. That is why, in this case, a typical evolutionary process might lead to convergence, freeing up time for further exploration of the space issue.

Table 6. Confusion matrix.

		Predicted Class	
		Condition Positive (P)	Condition Negative (N)
Actual Class	Condition Positive (P) The number of real positive cases in the data	True Positive (TP) Correct positive prediction	False Positive (FP) Incorrect positive prediction, Type I error.
	Condition Negative (N) The number of real negative cases in the data	False Negative (FN) Incorrect negative prediction, Type II error	True Negative (TN) Correct negative prediction.

Table 7. GA-based PD confusion matrix (average).

		Predicted Class	
		Positive (P)	Negative (N)
Actual Class	Positive (P)	93%	7%
	Negative (N)	7%	93%

Table 8. QGA-based PD confusion matrix (average).

		Predicted Class	
		Positive (P)	Negative (N)
Actual Class	Positive (P)	98%	2%
	Negative (N)	2%	98%

5.3. Experiment 3: A Self-Assessment with Different Values of ϵ and τ

The objective of the third set of experiments is to test the TPR, PPV, and F-Score of the model with different values of ϵ and τ for the PAN 13-14 dataset. As shown in Tables 9 and 10, the proposed model achieves better results as compared with the GA version in terms of TPR, PPV, and F-Score, which shows a general trend for documents as θ and β increase, TPR decreases, and PPV increases. At $\epsilon = 8$ and $\tau = 30$, the best F-score is obtained for the documents. That means that changing ϵ and τ will affect the value of the TPR, PPV, and F-Score. The superiority of the θ comes from the fact that it helps to minimize false detection. By adjusting the ϵ parameter, we may reduce the number of document-level comparisons performed during the passing stage and hence the number of plagiarized documents. How much of a sentence from a questionable source is plagiarized is determined by the threshold τ .

Table 9. Model performance with different ϵ values.

ϵ Values	TPR Value		PPV Value		F-Score	
	QGA	GA	QGA	GA	QGA	GA
1	1	0.98	0.23	0.20	0.39	0.38
2	1	0.98	0.25	0.23	0.41	0.39
3	1	0.98	0.26	0.26	0.44	0.41
4	1	0.98	0.34	0.32	0.53	0.52
5	1	0.97	0.44	0.43	0.62	0.61
6	0.98	0.97	0.73	0.72	0.84	0.81
7	0.98	0.97	0.85	0.84	0.94	0.93
8	0.97	0.97	0.98	0.96	0.97	0.96
9	0.97	0.96	1	0.98	0.95	0.95
10	0.93	0.92	1	1	0.94	0.94

Table 10. Model performance with different τ values.

τ Values	TPR Value		PPV Value		F-Score	
	QGA	GA	QGA	GA	QGA	GA
10	1	0.97	0.46	0.43	0.64	0.62
15	1	0.97	0.47	0.45	0.66	0.63
20	1	0.97	0.60	0.51	0.74	0.69
25	0.99	0.95	0.62	0.62	0.76	0.75
30	0.95	0.89	0.80	0.82	0.84	0.82
35	0.86	0.65	0.83	0.84	0.77	0.74
40	0.71	0.60	0.85	0.86	0.73	0.71
45	0.58	0.46	0.96	0.96	0.65	0.62
50	0.36	0.27	0.99	0.98	0.44	0.41

5.4. Experiment 4: Performance Accuracy with Different Training Samples

The objective of the fourth set of experiments is to test the accuracy of the model with different values of document datasets. As the model has more enrolled samples, the chance of a correct hit increases. The accuracy of the proposed model achieves better results with an increasing number of training documents. For all values of the documents' number, accuracy increases by approximately 10% on average. This means that if the model is trained with more samples, it will be better at finding plagiarized documents. As shown in Table 11, as expected, the identification rate increases as the number of samples grows. The accuracy rate rises by approximately 10–15% for each increase in the number of samples in the dataset. The accuracy may reach 98% when all samples are used to train the proposed model, owing to the role played by the QGA in determining which characteristics to use. In order to achieve this improvement, the time needed to train the model increases. When compared to the time invested in testing, however, this delay is small. The optimum feature selection module is the most time-consuming part of the training procedure.

Table 11. Accuracy of the model with different numbers of samples.

No of Samples	5	10	15	30	50	100	500	750	1000	1535
Accuracy (%)	15	35	40	50	55	60	70	75	80	98

5.5. Experiment 5: A Comparative Study with Recent Related Work

The fifth set of experiments was also conducted to evaluate the proposed system compared with the recent models. Models from [41,59–63] were selected to compare the proposed model to other well-known methods for text similarity detection. In the study described in [59], plagiarism is only evaluated after two levels of filtering have been applied using the bag-of-words approach, one at the document level and the other at the sentence level. In Ref. [60], a three-stage method based on the Smith–Waterman algorithm for plagiarism detection employs context matching and pre-trained word embeddings to detect instances of synonym substitution and word reordering. By combining linguistic features such as path similarity and depth estimation measures to compute the resemblance between the pair of words and assigning different weights to each feature, the work presented in [61] uses semantic knowledge to detect the plagiarized part of the text.

In Ref. [62], text embedding vectors are used to compare document similarity for plagiarism detection; these vectors include both semantic and syntactic information about the text, and they provide effective text alignment between the suspect and original documents. Sentences with the greatest resemblance are regarded as candidates or seeds of plagiarism cases by comparing their appearances in the source and suspect documents. Syntactic similarities between source and suspect phrases may be revealed using part-of-speech tag n-grams, as shown in [63]. Word2Vec, a word embedding method, is employed to quantify the semantic relatedness between words, while the longest common subsequence approach is used to quantify the semantic similarity between the source and suspect sentences. Table 12 shows the performance results of the proposed system compared to other related systems in terms of precision and F-measure.

The performance results for the PAN 13-14 corpus demonstrate that the proposed system outperforms the state-of-the-art systems on all documents. It can be seen that the majority of the previous systems acquired varying ranks in the various datasets. This variation is due to the structure of the dataset and the kinds of plagiarism that were included in it. However, the suggested method maintained its position as the best across all of the datasets. The suggested approach thus achieves effectiveness and reliability in detecting the various types of textual plagiarism based on these results. They also indicate the ability of the QGA to find the hyperplane equation of the selected features to detect the different types of text similarities. Utilizing the GQA helps to identify the interconnected, cohesive sentences that effectively convey the source document's main idea with more accuracy. See [64] for a more comparative study of different PD methods. Regarding the

running time, we find that there are no major variations between any of the approaches and that the average variance between them is just 4 s. The total time largely depends on the size of the corpus (1535 documents in our case). The suggested approach requires more time, but the results are more precise.

Table 12. Comparison results of the proposed text similarity system and other relevant systems in the PAN 13-14 dataset. (Average for testing data-1 and testing data-2.)

PD Methods	Precision (%)	F-Measure (%)	Run Time (Sec)
Arabi, H., Akbari, M [59]	90.08	86.65	56
Alvi, F. et al. [60]	92.52	86.84	55
Ahuja, L. et al. [61]	85.60	88.65	49
Gharavi, E. et al. [62]	89.75	90.15	53
Yalcin, K. et al. [63]	92.76	90.18	54
El-Rashidy M. et al. [41]	92.61	89.43	51
Proposed System	97.91	94.68	58

5.6. Experiment 6: Run Time and Complexity Analysis

The last set of experiments is meant to prove that the suggested QGA-based PD model converges quickly compared to the traditional GA-based model for PAN 13–14 datasets with different population sizes. The results shown in Table 13 confirm this fact with an average 1% reduction. As discussed earlier, the total running time largely depends on the size of the corpus.

Table 13. Running time (average) with different population sizes for both the QGA and the traditional GA-based PD model for PAN 13-14.

Population Size	5	10	15	30	50	100
QGA-based PD Model	49	51	52	53	55	58
GA-based PD model	54	56	57	59	60	65

It is usually true that quantum algorithms may reduce the complexity of their classical counterparts. We can roughly estimate the complexity decrease by comparing the global complexity of the QGA to that of the GA. The global complexity for the QGA is $O(N)$, where N is the total population size (Evaluation + Interference). The global complexity of an ordinary GA is often in the order of $O(N^2)$ (Evaluation + Selection + Crossover + Mutation). Indeed, one can foresee what would occur if we were to study a very large population of chromosomes; the QGA instead of the GA would be extremely beneficial. Our experimental results show that the QGA can be a very promising tool for exploring large search spaces while preserving the relation efficiency/performance. See [22] for more details.

6. Conclusions

From the standpoint of a forensic linguist, it is critical to determine with absolute certainty whether a text is an original or the consequence of plagiarism. Expert evidence from a forensic linguist is often required in court cases, but this field is not only concerned with law; forensic linguists also study public-facing topics. Therefore, incorrect judgments must be avoided at all costs to avoid miscarriages of justice, whether in the classroom or the courtroom. In this paper, a new approach based on the semantic similarity concept and the QGA for PD is proposed. The proposed model includes four main steps: the pre-processing and document representation module, sentence-level concept extraction using the QGA, the document-level detection phase, and the passage-level detection phase.

The semantic similarity concept, which depends on intelligent techniques, is employed for extracting the concepts from documents in an effective way to enhance the model's

performance. The QGA is employed to find relatedness between sentences that show the concept of the source document briefly, enhancing the model's processing time. The solution based on PDS has the advantage of detecting plagiarized ideas in documents presented via summarization.

The proposed model was evaluated by using samples of benchmarked datasets. Based on the obtained results, the proposed model for the detection of plagiarism shows an excellent performance in terms of accuracy. It has been compared with the HGA and the GA-based PD model, and it has come up with better results against them. The QGA has been proven to provide better results in terms of accuracy without adding any complications to the model. The solution's shortcomings, such as WordNet's inability to measure all possible semantic relationships between words, reduce its efficiency. Despite the method's general effectiveness, there are other methods to implement the idea, such as paraphrasing and expanding upon concepts. A possible future study includes making use of a different database to determine how closely related terms are semantically. Furthermore, future work will focus on comparing different QGA strategies to study the effect of choosing rotation gate angles. Another perspective of this work is to study parallel QGAs because QGAs are highly parallelizable.

Author Contributions: Conceptualization, S.M.D. and I.A.M.; methodology, S.M.D. and A.A.E.; software, I.A.M.; validation, S.M.D. and A.A.E.; formal analysis, S.M.D. and A.A.E.; investigation, S.M.D., I.A.M., and A.A.E.; resources, I.A.M.; data curation, S.M.D. and I.A.M.; writing—original draft preparation, S.M.D. and I.A.M.; writing—review and editing, A.A.E. and I.A.M.; visualization, I.A.M.; supervision, S.M.D.; project administration, S.M.D.; funding acquisition, I.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets for this research are available via <https://pan.webis.de/data.html> accessed on 1 January 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ali, J. Forensic Linguistics: A Study in Criminal Speech Acts. *Beni-Suef Univ. Int. J. Humanit. Soc. Sci.* **2020**, *2*, 39–65.
2. Umiyati, M. A Literature Review of Forensic Linguistics. *Int. J. Forensic Linguist.* **2020**, *15*, 23–29.
3. Supriadi, N.; Gunawan, W.; Muniroh, R. Bullies' Attitudes on Twitter: A Forensic Linguistics Analysis of Cyberbullying (Systemic Functional Linguistics Approach). *Passage* **2020**, *8*, 111–124.
4. Woolls, D. Computational Forensic Linguistics: Computer-Assisted Document Comparison. In *Routledge Handbook of Forensic Linguistics*; Routledge: Abingdon, UK, 2020; pp. 593–607.
5. Moura, R.; Sousa-Silva, R.; Cardoso, H.L. Automated Fake News Detection Using Computational Forensic Linguistics. In *EPIA Conference on Artificial Intelligence, Proceedings of the 20th EPIA Conference on Artificial Intelligence, EPIA 2021, Virtual, 7–9 September 2021*; Springer: Cham, Switzerland, 2021; pp. 788–800.
6. Sánchez-Vega, F.; Villatoro-Tello, E.; Montes-y-Gomez, M.; Villaseñor-Pineda, L.; Rosso, P. Determining and Characterizing the Reused Text for Plagiarism Detection. *Expert Syst. Appl.* **2013**, *40*, 1804–1813.
7. Paula, M.; Jamal, S. An Improved SRL based Plagiarism Detection Technique using Sentence Ranking. *Procedia Comput. Sci.* **2015**, *46*, 223–230.
8. Vani, K.; Gupta, D. Detection of Idea Plagiarism using Syntax–Semantic Concept Extractions with Genetic Algorithm. *Expert Syst. Appl.* **2017**, *73*, 11–26.
9. Maurer, H.; Kappe, F.; Zaka, B. Plagiarism—A Survey. *J. Univers. Comput. Sci.* **2006**, *12*, 1050–1084.
10. Alzahrani, S.; Salim, N.; Abraham, A. Understanding Plagiarism Linguistic Patterns, Textual Features, & Detection Methods. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 133–149.
11. Vani, K.; Gupta, D. Study on Extrinsic Text Plagiarism Detection Techniques and Tools. *J. Eng. Sci. Technol. Rev.* **2016**, *9*, 150–164. [CrossRef]
12. Wang, Y.; Huang, J.; Dong, W.; Yan, J.; Tian, C.; Li, M.; Mo, W. Two-Stage based Ensemble Optimization Framework for Large-Scale Global Optimization. *Eur. J. Oper. Res.* **2013**, *228*, 308–320.

13. Geravand, S.; Ahmadi, M. An Efficient and Scalable Plagiarism Checking System using Bloom Filters. *Comput. Electr. Eng.* **2014**, *40*, 1789–1800.
14. Mohammed, R.; Shaaban, O.; Mahran, D.; Attellawy, H.; Makhlof, A.; Albasri, A. Plagiarism in Medical Scientific Research. *J. Taibah Univ. Med. Sci.* **2015**, *10*, 6–11.
15. Dao, S.; Abhary, K.; Marian, R. An Improved Structure of Genetic Algorithms for Global Optimization. *Prog. Artif. Intell.* **2016**, *5*, 155–163.
16. Floudas, C. *Deterministic Global Optimization: Theory, Methods and Applications*; Springer: Berlin/Heidelberg, Germany, 2000.
17. Shahlaei, M.; Sobhani, A.; Saghaie, L.; Fassihi, A. Application of an Expert System based on Genetic Algorithm-Adaptive Neuro-Fuzzy Inference System (GA-ANFIS) in QSAR of Cathepsin K Inhibitors. *Expert Syst. Appl.* **2012**, *39*, 6182–6191.
18. Amal, R.; Ivan, J. A Quantum Genetic Algorithm for Optimization Problems on the Bloch Sphere. *Quantum Inf. Process.* **2022**, *21*, 43.
19. Wu, N.; Sun, J. Fatigue Detection of Air Traffic Controllers Based on Radiotelephony Communications and Self-Adaption Quantum Genetic Algorithm Optimization Ensemble Learning. *Appl. Sci.* **2022**, *12*, 10252.
20. Ling, Z.; Hao, Z. Intrusion Detection Using Normalized Mutual Information Feature Selection and Parallel Quantum Genetic Algorithm. *Int. J. Semant. Web Inf. Syst.* **2022**, *18*, 1–24.
21. Martino, F.; Sessa, S. A Novel Quantum Inspired Genetic Algorithm to Initialize Cluster Centers in Fuzzy C-Means. *Expert Syst. Appl.* **2022**, *191*, 116340.
22. Laboudi, Z.; Chikhi, S. Comparison of Genetic Algorithm and Quantum Genetic Algorithm. *Int. Arab J. Inf. Technol.* **2012**, *9*, 243–249.
23. Acampora, G.; Schiattarella, R.; Vitiello, A. Using Quantum Amplitude Amplification in Genetic Algorithms. *Expert Syst. Appl.* **2022**, *209*, 118203.
24. Man, Z.; Li, J.; Di, X.; Mu, Y. Application of Quantum Genetic Algorithm in High Noise Laser Image Security. *Optoelectron. Lett.* **2022**, *18*, 59–64.
25. Osman, A.; Salim, N.; Binwahlan, M.; Alteeb, R.; Abuobieda, A. An Improved Plagiarism Detection Scheme based on Semantic Role Labelling. *Appl. Soft Comput.* **2012**, *12*, 1493–1502.
26. Abdi, A.; Idris, N.; Alguliyev, R.; Aliguliyev, R. PDLK: Plagiarism Detection using Linguistic Knowledge. *Expert Syst. Appl.* **2015**, *42*, 8936–8946.
27. Alzahrani, S.; Salim, N.; Palade, V. Uncovering Highly Obfuscated Plagiarism Cases using Fuzzy Semantic-based Similarity Model. *J. King Saud Univ.-Comput. Inf. Sci.* **2015**, *27*, 248–268.
28. Vani, K.; Gupta, D. Text Plagiarism Classification using Syntax based Linguistic Features. *Expert Syst. Appl.* **2017**, *88*, 448–464.
29. Kaur, M.; Gupta, V.; Kaur, R. Semantic-based Integrated Plagiarism Detection Approach for English Documents. *IETE J. Res.* **2021**, *21*, 1–7. [CrossRef]
30. Nazir, A.; Mir, R.; Qureshi, S. Idea Plagiarism Detection with Recurrent Neural Networks and Vector Space Model. *Int. J. Intell. Comput. Cybern.* **2021**, *14*, 321–332.
31. JavadiMoghaddam, S.; Roosta, F.; Noroozi, A. Weighted Semantic Plagiarism Detection Approach Based on AHP Decision Model. *Account. Res.* **2022**, *29*, 203–223.
32. Alvi, F.; Stevenson, M.; Clough, P. Paraphrase type identification for plagiarism detection using contexts and word embeddings. *Int. J. Educ. Technol. High. Educ.* **2021**, *18*, 42.
33. Arabi, H.; Akbari, M. Improving Plagiarism Detection in Text Document Using Hybrid Weighted Similarity. *Expert Syst. Appl.* **2022**, *207*, 118034.
34. Zouhir, A.; El Ayachi, R.; Biniz, M. A comparative Plagiarism Detection System methods between sentences. *J. Phys. Conf. Ser.* **2021**, *1743*, 012041. [CrossRef]
35. Kaur, M.; Gupta, V.; Kaur, R. Review of Recent Plagiarism Detection Techniques and Their Performance Comparison. In *Proceedings of International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*; Springer: Singapore, 2021; pp. 157–170.
36. El-Rashidy, M.; Mohamed, R.; El-Fishawy, N.; Shouman, M. Reliable Plagiarism Detection System Based on Deep Learning Approaches. *Neural Comput. Appl.* **2022**, *34*, 18837–18858. [CrossRef]
37. Jambi, K.; Khan, I.; Siddiqui, M. Evaluation of Different Plagiarism Detection Methods: A Fuzzy MCDM Perspective. *Appl. Sci.* **2022**, *12*, 4580. [CrossRef]
38. Kumar, V.; Bhatt, C.; Namdeo, V. A Framework for Document Plagiarism Detection Using Rabin Karp Method. *Int. J. Innov. Res. Technol. Manag.* **2021**, *5*, 7–30.
39. Ali, A.; Taqa, A. Analytical Study of Traditional and Intelligent Textual Plagiarism Detection Approaches. *J. Educ. Sci.* **2022**, *31*, 8–25. [CrossRef]
40. Abdelhamid, M.; Azouaou, F.; Batata, S. A Survey of Plagiarism Detection Systems: Case of Use with English, French and Arabic Languages. *arXiv* **2022**, arXiv:2201.03423.
41. El-Rashidy, M.; Mohamed, R.; El-Fishawy, N.; Shouman, M. An effective text plagiarism detection system based on feature selection and SVM techniques. *Multimed. Tools Appl.* **2023**, *82*, 1–38. [CrossRef]
42. Kulkarni, R.; Ganesh, C.; BK, D.; Harshitha, B.; Reddy, A. Novel Approach to Detect Plagiarism in the Document. In *Proceedings of the 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics*, Ballar, India, 29–30 April 2023; pp. 1–6.

43. Zahid, M.; Abid, K.; Rehman, A.; Fuzail, M.; Aslam, N. An Efficient Machine Learning Approach for Plagiarism Detection in Text Documents. *J. Comput. Biomed. Inform.* **2023**, *4*, 241–248.
44. Pokharana, A.; Garg, U. A Review on diverse algorithms used in the context of Plagiarism Detection. In Proceedings of the 2023 International Conference on Advancement in Computation & Computer Technologies, Gharuan, India, 5–6 May 2023; pp. 1–6.
45. Kholodna, N.; Vysotska, V.; Markiv, O.; Chyrun, S. Machine Learning Model for Paraphrases Detection Based on Text Content Pair Binary Classification. In *CEUR Workshop Proc., Proceedings of the 4th International Workshop on Modern Machine Learning Technologies and Data Science, Leiden, The Netherlands, November 25–26, 2022*; CEUR-WS: Aachen, Germany, 2022; Volume 3312, pp. 283–306.
46. Chae, D.; Ha, J.; Kim, S.; Kang, B.; Im, E.; Park, S. Credible, Resilient, and Scalable Detection of Software Plagiarism using Authority Histograms. *Knowl.-Based Syst.* **2016**, *95*, 114–124. [CrossRef]
47. Geeganage, D.; Xu, Y.; Li, Y. Semantic-based topic representation using frequent semantic patterns. *Knowl.-Based Syst.* **2021**, *216*, 106808. [CrossRef]
48. Chang, C.Y.; Lee, S.J.; Wu, C.H.; Liu, C.F.; Liu, C.K. Using word semantic concepts for plagiarism detection in text documents. *Inf. Retr. J.* **2021**, *24*, 298–321. [CrossRef]
49. SaiToh, A.; Rahimi, R.; Nakahara, M. A Quantum Genetic Algorithm with Quantum Crossover and Mutation Operations. *Quantum Inf. Process.* **2014**, *13*, 737–755. [CrossRef]
50. Halliday, M.; Hasan, R. *Cohesion in English*; Taylor & Francis Group: Abingdon, UK, 2014.
51. Nandhini, K.; Balasundaram, S. Use of Genetic Algorithm for Cohesive Summary Extraction to Assist Reading Difficulties. *Appl. Comput. Intell. Soft Comput.* **2013**, *2013*, 945623. [CrossRef]
52. Liu, H.; Zhao, B.; Huang, L. A novel quantum image encryption algorithm based on crossover operation and mutation operation. *Multimed. Tools Appl.* **2019**, *78*, 20465–20483. [CrossRef]
53. Lahoz-Beltra, R. Quantum genetic algorithms for computer scientists. *Computers* **2016**, *5*, 24. [CrossRef]
54. Shehata, S.; Karray, F.; Kamel, M. An Efficient Model for Enhancing Text Categorization using Sentence Semantics. *Comput. Intell.* **2010**, *26*, 215–231. [CrossRef]
55. Nation, P.; Johansson, J.; Pitchford, A.; Granade, C. QuTiP: Quantum Toolbox in Python. 2022. Available online: <https://qutip.org/> (accessed on 1 January 2023).
56. PAN. Data. Available online: <https://pan.webis.de/data.html> (accessed on 1 January 2023).
57. Potthast, M.; Gollub, T.; Hagen, M.; Tippmann, M.; Kiesel, J.; Rosso, P.; Stamatatos, E.; Stein, B. Overview of the 5th International Competition on Plagiarism Detection. In Proceedings of the Conference and Labs of the Evaluation Forum, Valencia, Spain, 23–26 September 2013; pp. 1–3.
58. Darwish, S.; Moawad, M. An Adaptive Plagiarism Detection System Based on Semantic Concept and Hierarchical Genetic Algorithm. In Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 26–28 October 2019; Springer: Cham, Switzerland, 2019; pp. 739–749.
59. Muangprathub, J.; Kajornkasirat, S.; Wanichsombat, A. Document plagiarism detection using a new concept similarity in formal concept analysis. *J. Appl. Math.* **2021**, *2021*, 1–10. [CrossRef]
60. Asghari, H.; Fatemi, O.; Mohtaj, S.; Faili, H.; Rosso, P. On the use of word embedding for cross language plagiarism detection. *Intell. Data Anal.* **2019**, *23*, 661–680. [CrossRef]
61. Ahuja, L.; Gupta, V.; Kumar, R. A new hybrid technique for detection of plagiarism from text documents. *Arab. J. Sci. Eng.* **2020**, *45*, 9939–9952. [CrossRef]
62. Gharavi, E.; Veisi, H.; Rosso, P. Scalable and language-independent embedding-based approach for plagiarism detection considering obfuscation type: No training phase. *Neural Comput. Appl.* **2020**, *32*, 10593–10607. [CrossRef]
63. Yalcin, K.; Cicekli, I.; Ercan, G. An external plagiarism detection system based on part-of-speech (POS) tag n-grams and word embedding. *Expert Syst. Appl.* **2022**, *197*, 116677. [CrossRef]
64. Mansoor, M.; Al-Tamimi, M. Computer-based plagiarism detection techniques: A comparative study. *Int. J. Nonlinear Anal. Appl.* **2022**, *13*, 3599–3611.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Comparative Study of Variations in Quantum Approximate Optimization Algorithms for the Traveling Salesman Problem

Wenyang Qian ^{1,2,*}, Robert A. M. Basili ³, Mary Mehrnoosh Eshaghian-Wilner ⁴, Ashfaq Khokhar ³, Glenn Luecke ⁴ and James P. Vary ²

¹ Instituto Galego de Física de Altas Enerxias (IGFAE), Universidade de Santiago de Compostela, E-15782 Santiago de Compostela, Spain

² Department of Physics and Astronomy, Iowa State University, Ames, IA 50011, USA; jvary@iastate.edu

³ Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA; basiliro@iastate.edu (R.A.M.B.); ashfaq@iastate.edu (A.K.)

⁴ Department of Mathematics, Iowa State University, Ames, IA 50011, USA; mmew@iastate.edu (M.M.E.-W.); grl@iastate.edu (G.L.)

* Correspondence: qian.wenyang@usc.es or wqian@iastate.edu

Abstract: The traveling salesman problem (TSP) is one of the most often-used NP-hard problems in computer science to study the effectiveness of computing models and hardware platforms. In this regard, it is also heavily used as a vehicle to study the feasibility of the quantum computing paradigm for this class of problems. In this paper, we tackle the TSP using the quantum approximate optimization algorithm (QAOA) approach by formulating it as an optimization problem. By adopting an improved qubit encoding strategy and a layer-wise learning optimization protocol, we present numerical results obtained from the gate-based digital quantum simulator, specifically targeting TSP instances with 3, 4, and 5 cities. We focus on the evaluations of three distinctive QAOA mixer designs, considering their performances in terms of numerical accuracy and optimization cost. Notably, we find that a well-balanced QAOA mixer design exhibits more promising potential for gate-based simulators and realistic quantum devices in the long run, an observation further supported by our noise model simulations. Furthermore, we investigate the sensitivity of the simulations to the TSP graph. Overall, our simulation results show that the digital quantum simulation of problem-inspired ansatz is a successful candidate for finding optimal TSP solutions.

Keywords: quantum computing; quantum simulation; quantum approximate optimization algorithm; traveling salesman problem; noisy simulation

Citation: Qian, W.; Basili, R.A.M.; Eshaghian-Wilner, M.M.; Khokhar, A.; Luecke, G.; Vary, J.P. Comparative Study of Variations in Quantum Approximate Optimization Algorithms for the Traveling Salesman Problem. *Entropy* **2023**, *25*, 1238. <https://doi.org/10.3390/e25081238>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 17 July 2023

Revised: 18 August 2023

Accepted: 19 August 2023

Published: 21 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

For over a century, the traveling salesman problem (TSP) [1] has inspired hundreds of works and dozens of algorithms, of both exact and heuristic approaches. Today, the TSP has become so quintessential in modern computing that it is commonly considered the prototypical NP-hard combinatorial optimization problem, possessing far-reaching impact on countless applications in science, industry, and society. Consequently, the TSP is frequently taken as an ideal candidate for new computational models and non-standard algorithmic approaches, including approximate approaches like simulated annealing [2] and self-organizing maps [3], which have been widely employed to tackle the TSP.

Recent advancements in quantum technologies have paved the way for various quantum computing approaches to tackle the traveling salesman problem (TSP). These approaches include the quantum Held–Karp algorithm [4], quantum annealing (QA) [5–9], and the more general variational quantum algorithm [10,11] (VQA). VQA approaches have found extensive applications in diverse fields such as chemistry [11], physics [12], and finance [13], among others. Although complete demonstrations of quantum advantage over classical algorithms are currently limited due to the noisy intermediate-scale quantum

(NISQ) era [14], exploring these quantum algorithms remains crucial, as experimentation on prototype quantum hardware continues to rapidly approach what can be classically simulated by even the world's largest supercomputers. Notably, the quantum approximate optimization algorithm (QAOA) [10,15], a subclass of the general VQA, has been successfully applied to a number of optimization problems [16], including the max-cut problem [17,18], vehicle routing [19], DNA sequencing [20], protein folding [21], as well as the TSP [22]. In comparison to the popular hardware-efficient VQA, the QAOA takes advantage of the domain knowledge of the specific problem at hand to produce a variational ansatz with fewer parameters and a shallower depth. Furthermore, an extension of the original QAOA called the quantum alternating operator ansatz [23,24] offers a generalized approach that specializes in solving problems with hard constraints.

In the NISQ era, the QAOA approach can be particularly advantageous for addressing the challenges of the traveling salesman problem (TSP), owing to the QAOA's hybrid feature, hardware-friendly structure, and controlled optimization. Being a hybrid approach, the QAOA exhibits robust tolerance to systematic errors by leveraging classical computer optimizers. Its layered ansatz structure inspired by the problem Hamiltonian allows for high flexibility in the circuit depth and qubit coherence time, incorporating the capabilities offered by the quantum backends. Compared with the QA [25,26], the QAOA also enables fine control of optimization through its finite layers, which is particularly beneficial in the current NISQ era. However, the numerical simulation of the QAOA on the TSP, especially in the multiple-layer region, is not well understood, since the non-adiabatic mechanism of the QAOA differs significantly from that of the QA [27]. Therefore, it becomes imperative to explore various implementations of the QAOA to determine the optimal path for simulation. Conducting investigations into these problems on digital quantum computers or simulators is essential, as they have the potential to unveil new quantum simulation strategies for traditional optimization tasks. We distinguish the present work from the previous studies by constructing our QAOA using different ansatzes and comparing their performances in both numerical accuracy and resource cost, which addresses a crucial aspect that is often neglected in conventional studies.

In this work, we study the effectiveness of three distinct designs of the QAOA in solving the TSP by adopting a layer-wise learning optimization protocol [28] on digital quantum simulators via Qiskit [29]. We organize this paper as follows: In Section 2, we introduce the TSP and its mathematical formulation as a binary constraint optimization problem. In Section 3, we outline the QAOA methods, with particular focus on the initialization, mixer ansatz, and measurement protocol employed in this work. In Section 4, we present and compare the numerical results of the QAOA simulation on TSP instances with 3, 4, and 5 cities, utilizing different ansatz designs. We discuss the impact of the device noise and TSP variations on the simulation results. In Section 5, we summarize the results and discuss plans for the future.

2. Traveling Salesman Problem

In this section, we first define the TSP as an optimization problem and then improve its formulation by taking advantage of the symmetry in the solution.

2.1. TSP Formulation as an Optimization Problem

The traveling salesman problem asks for the shortest path that visits each city exactly once and returns to the starting city. In the symmetric case where the distance between any two cities is the same regardless of the traveling direction, the TSP can be reformulated as an undirected graph problem where its vertices represent cities and the edge weights represent traveling distances. Mathematically, given an undirected graph G with vertices V and edges E , i.e., $G = (V, E)$, we aim to find a Hamiltonian cycle that goes through all $|V|$ nodes exactly once with the smallest total weights of the connecting edges on the path.

In this graph formulation of the TSP, any valid cycle, be it minimum or not, can be represented by a visiting order or a permutation of integers, such as $\{0, 1, \dots, n - 1\}$,

where the integers are the city indices starting at 0 for a total of n cities. Alternatively, the visiting order on a TSP graph can be conveniently described by a sequence of binary decision variables $x_{i,t}$, indicating whether city- i is visited at time t [30]. If $x_{i,t} = 1$, then city- i is visited at t ; otherwise, the city is not visited by the traveling salesman. Naively, to fully describe the solution to an n -city TSP, a total of n^2 binary variables is needed in this representation.

Alternatively, this “one-hot” representation of binary decision variables can be written collectively in either a matrix or flattened array format for numerical implementation. For instance, a valid Hamiltonian cycle of permutation $x = (0, 1, 2, 3)$ is translated into binary decision variables x as:

$$x = (0, 1, 2, 3) \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \equiv 1000010000100001, \tag{1}$$

where the matrix row index represents each city index, and the column index represents each time instance. City- i is visited at time t if and only if $x_{i,t} = 1$. In this work, all three notations (permutation, matrix, and bit string array) are used interchangeably. Any Hamiltonian cycle in the TSP has a unique sequence of binary decision variables or “bit string”. However, the reverse is not true, since a large portion of the possible bit strings may not correspond to any meaningful permutation. Specifically, we classify any bit string x into three categories or states:

$$x = \begin{cases} \text{true,} & x \text{ is a permutation and gives the shortest path,} \\ \text{false,} & x \text{ is a permutation but does not give the shortest path,} \\ \text{invalid,} & x \text{ is not a permutation,} \end{cases} \tag{2}$$

where the true and false bit strings are also called valid bit strings. Any bit string can be translated to a Hamiltonian cycle if and only if it is a permutation. Clearly, invalid solutions are disallowed traveling orders to the TSP.

With binary decision variables x , a true solution to an n -city TSP can be found by finding an x that minimizes the following cost function [30]:

$$C_{\text{dist}}(x) = \sum_{0 \leq i, j < n} \omega_{ij} \sum_{t=0}^{n-1} x_{i,t} x_{j,t+1}, \tag{3}$$

where ω_{ij} is the distance (or edge weight in the undirected graph) between city- i and city- j (in the symmetric TSP, $\omega_{ij} = \omega_{ji}$ and $\omega_{ii} = 0$). Here, $C_{\text{dist}}(x)$ also gives the shortest TSP distance when x is a true solution. Since the cost function itself does not forbid invalid solutions in general, additional constraint conditions must be satisfied for a valid Hamiltonian cycle, such as:

$$\sum_{i=0}^{n-1} x_{i,t} = 1 \quad \text{for } t = 0, 1, \dots, n - 1 \tag{4}$$

$$\sum_{t=0}^{n-1} x_{i,t} = 1 \quad \text{for } i = 0, 1, \dots, n - 1, \tag{5}$$

where Equation (4) forbids multiple cities visited by the traveler at the same time, and Equation (5) forbids revisiting the same city. Alternatively, in the matrix format, these constraints are easily implemented by requiring that any row or column sum to exactly one. These two hard constraints are the necessary conditions for any valid solution, though not necessarily a true solution to a TSP. To formulate the TSP as a minimum-optimization

problem, these constraint conditions are conveniently incorporated as the penalty terms, such that the combined cost function, $C(x)$ becomes:

$$C(x) = C_{\text{dist}}(x) + \lambda C_{\text{penalty}}(x) \tag{6}$$

$$= \sum_{0 \leq i, j < n} \omega_{ij} \sum_{t=0}^{n-1} x_{i,t} x_{j,t+1} + \lambda \left\{ \sum_{t=0}^{n-1} \left(1 - \sum_{i=0}^{n-1} x_{i,t} \right)^2 + \sum_{i=0}^{n-1} \left(1 - \sum_{t=0}^{n-1} x_{i,t} \right)^2 \right\}, \tag{7}$$

where λ is the weight factor of the penalty term, serving as the Lagrange multiplier, which should be positive and sufficiently large. It is easy to see that bit string x gives the minimum of $C(x)$ if and only if x is a true solution to the given TSP. Finding a Hamiltonian cycle to the TSP is now equivalent to finding an x^* that minimizes $C(x)$ in Equation (6), i.e., $x^* = \arg \min C(x)$.

2.2. Improved TSP by Eliminating Rotational Symmetry

Symmetry plays a vital role in many graph optimization problems, and exploiting them can help reduce the problem’s complexity. In the previously introduced TSP optimization, one uses n^2 decision variables for n cities. However, solutions obtained after the optimization display “rotational” symmetry; they are physically identical up to some rotation. For example, a visiting order of permutation $(0, 1, 2)$ is equivalent to $(1, 2, 0)$ and $(2, 0, 1)$ for a three-city TSP. They form a natural equivalence class for the solution sets. To reduce the size of the search space (and the number of qubits to encode), a simple but significant improvement can be made by fixing the starting city [30].

Without loss of generality, we fix city-0 as our starting point. The traveling salesman will return to city-0 after visiting all other cities exactly once. Then, the improved cost functions $C'_{\text{dist}}(x)$ and $C'(x)$ become:

$$C'_{\text{dist}}(x) = \sum_{1 \leq i, j < n} \omega_{ij} \sum_{t=1}^{n-2} x_{i,t} x_{j,t+1} + \sum_{i=1}^{n-1} \omega_{0i} (x_{i,1} + x_{i,n-1}), \tag{8}$$

$$C'(x) = C'_{\text{dist}}(x) + \lambda C'_{\text{penalty}}(x) \tag{9}$$

$$= \sum_{1 \leq i, j < n} \omega_{ij} \sum_{t=1}^{n-2} x_{i,t} x_{j,t+1} + \sum_{i=1}^{n-1} \omega_{0i} (x_{i,1} + x_{i,n-1}) + \lambda \left\{ \sum_{t=1}^{n-1} \left(1 - \sum_{i=1}^{n-1} x_{i,t} \right)^2 + \sum_{i=1}^{n-1} \left(1 - \sum_{t=1}^{n-1} x_{i,t} \right)^2 \right\}. \tag{10}$$

In this new cost function, decision variables $x_{i,t}$ only take value $i = \{1, 2, \dots, n\}$ and $t = \{1, 2, \dots, n\}$, and thus we only need effectively $(n - 1)^2$ decision variables for an n -city TSP after fixing the initial city. The reduction in the length of the bit string is especially advantageous because it is ultimately equivalent to reducing the number of qubits for encoding the problem on a quantum circuit. Additionally, it is important to point out that this TSP optimization formulation works for a generally symmetric TSP, not relying on a flat surface, which can be generalized to many real-world applications where non-planar relations are ubiquitous, such as social networks, stock markets, materials science, and so forth. Asymmetric TSP ($\omega_{ij} \neq \omega_{ji}$) can also be formulated similarly in principle but is not considered within the scope of this work.

There are many other ways to formulate n -city TSP as an optimization problem [31,32], usually requiring more than n^2 variables. Recent work [33–35] explores unique features of the TSP as an optimization problem and leads to even fewer qubits and computational resources. Within the n^2 -variable formulation, an alternative approach to formulating the TSP expresses the cost function in terms of the adjacency matrix:

$$C_{\text{adj}}(x) = \sum_{0 \leq i, j < n} \omega_{ij} x_{ij}^{\text{adj}}, \tag{11}$$

where x^{adj} is the adjacency/connectivity representation of a permutation. The adjacency matrix representation can be particularly useful in symmetric TSP because time degrees of freedom are automatically factored out. Penalty terms for the cost function can be conveniently included by means of the symmetry about the main diagonal. However, unlike our adopted construction, it is not straightforward to reduce the number of decision variables in Equation (11), and therefore we leave it for a future study. In the subsequent section, we introduce the quantum approximate optimization algorithm based on the improved TSP optimization formulation according to Equation (10).

3. Quantum Approximate Optimization Algorithm (QAOA)

The quantum approximate optimization algorithm (QAOA) [10,23] is a general quantum heuristic approach for solving optimization problems. In this section, we introduce the QAOA workflow in detail and its application to the TSP formulation introduced in Section 2.2.

3.1. QAOA Workflow

The QAOA is deeply connected with the adiabatic quantum computation (AQC) [36], which is based on the adiabatic theorem. In AQC, the whole simulation process can be viewed as a time-dependent Hamiltonian evolution represented by $H(t)$, where:

$$H(t) = (1 - \frac{t}{T})H_M + \frac{t}{T}H_P. \tag{12}$$

Here, H_M represents a known ansatz, and H_P is the target Hamiltonian that one aims at to find a ground state. According to the adiabatic theorem, by gradually introducing perturbation, an initial eigenstate of $H(t = 0) = H_M$ will evolve into the ground state of $H(t = T) = H_P$. However, in practice, simulating this process can be extremely time-consuming, and accurately estimating a suitable duration poses its own challenges. The fundamental idea behind the QAOA is to approximate this adiabatic process by parameterizing the infinitely-long time evolution into finite time steps, addressing practical considerations. In both the original QAOA [10] and the extended QAOA [23], the hybrid quantum approach consists of three essential parts:

1. **State initialization** with initial state $|s\rangle$.
2. **Parameterized unitary ansatz** $U_p(\vec{\beta}, \vec{\gamma})$, a variational ansatz of p layers for the TSP, based on two alternating Hamiltonians, H_P and H_M , using respective parameters $\vec{\beta}$ and $\vec{\gamma}$.
3. **Measurement and optimization** of the cost expectation $\langle \vec{\beta}, \vec{\gamma} | C(x) | \vec{\beta}, \vec{\gamma} \rangle$ for the final state $|\vec{\beta}, \vec{\gamma}\rangle$, where an optimizer on a classical computer is used for the minimization.

Putting the three parts together, we construct the complete QAOA circuit, where the final state after the evolution is:

$$|\vec{\beta}, \vec{\gamma}\rangle = U_p(\vec{\beta}, \vec{\gamma}) |s\rangle = \left(\prod_{i=1}^p U_M(\beta_i) U_P(\gamma_i) \right) |s\rangle \tag{13}$$

$$= U_M(\beta_p) U_P(\gamma_p) \dots U_M(\beta_1) U_P(\gamma_1) |s\rangle, \tag{14}$$

where p is referred to as the depth (or layer number) of the QAOA. Specifically, the two alternating unitary ansatzes in each layer are:

$$U_P(\gamma_i) = e^{-i\gamma_i H_P}, U_M(\beta_i) = e^{-i\beta_i H_M}, \tag{15}$$

where H_P is the problem Hamiltonian derived from the cost function, and H_M is the mixer Hamiltonian that explores the feasible subspace. In this work, we refer to the QAOA ansatz with p layers as p -QAOA. Note that $\vec{\gamma}$ and $\vec{\beta}$ are parameter vectors of length p to be optimized, and there is only one single parameter γ_i (β_i) for the associated

unitary ansatz $U_P (U_M)$ per layer. This means there are only two parameters per layer for the QAOA, independent of the number of qubits (i.e., problem size), which makes the approach highly scalable. These parameters or angles can also be regarded as mimicking the Trotterization time steps in the QAOA to approximate the adiabatic evolution in Equation (12); nonetheless, the behavior in the finite layer limit can be drastically different.

In the last few years, many variants of the QAOA approach have emerged [37]. One such variant is the multi-angle QAOA (*ma*-QAOA) [38], which uses a unique angle for each element of the Hamiltonian. This approach could potentially reduce circuit depth required for solving the TSP. Another variant, the digitized-counterdiabatic QAOA (DC-QAOA) [39,40] introduces an additional problem-dependent counterdiabatic driving term in each layer to enhance the convergence rate of the optimization process. Additionally, the adaptive-QAOA (ADAPT-QAOA) [41], inspired by the adaptive VQE, systematically selects the mixer ansatz based on the optimization, potentially improving the simulation outcome. Since these more advanced QAOAs generally require more than two parameters per layer and additional simulation time, we opted not to incorporate them in this initial work; however, we have plans to include these variants in a subsequent study, allowing for a more comprehensive analysis of the QAOA to the TSP.

3.2. From Binary Decision Variables to Qubits

To carry out the optimization on quantum computers, an efficient qubit encoding scheme is necessary to map the binary decision variable in the TSP formulation to quantum computers. Here, we use the standard boolean binary variable mapping strategy [42]. For an n -city TSP, we simply map:

$$x_{i,t} \mapsto (I_{(i,t)} - Z_{(i,t)})/2, \tag{16}$$

where $Z_{(i,t)}$ is the Pauli-Z matrix (see Appendix A) at qubit location (i, t) on a two-dimensional lattice. To identify the qubit on the lattice with its realistic index in a quantum device, one may use the ideal mapping (ignoring the device connectivity) that takes $(i, t) \rightarrow ni + t$ for the original TSP formulation in Equation (6). For the improved TSP formulation according to Equation (10), since both sets of the $i = 0$ and $t = 0$ qubits are never used, we economically map:

$$(i, t) \mapsto (n - 1)(i - 1) + (t - 1), \tag{17}$$

such that only a total of $(n - 1)^2$ qubits is needed, from index 0 to $(n - 1)^2 - 1$, for n cities. Reducing qubit number is crucial in the practical quantum simulation, and therefore we adopt the mapping strategy in Equation (17) for the improved TSP formulation throughout this work.

3.3. State Initialization

The initial states are one of the key components in the QAOA approach. In the original QAOA [10], the initial states are always set to be $|+\rangle^{\otimes N}$, where N is the total number of qubits. For an n -city TSP, with the original $n^2 = N$ case for simplicity, it means the initial state becomes:

$$|s_H^n\rangle = H^{\otimes n^2} |0\rangle = |+\rangle^{\otimes n^2} = \frac{1}{\sqrt{2^{n^2}}} \sum_{x=0}^{2^{n^2}-1} |x\rangle. \tag{18}$$

In this way, the initial quantum state $|s_H^n\rangle$ is a superposition of all possible basis states for the problem. While this strategy is easy to implement on a quantum device using Hadamard gates H , the magnitude of each basis state in the initial state shrinks exponentially as the number of cities increases because the dimension of the search space grows as $\mathcal{O}(2^N)$.

Recently, additional initialization strategies of a restricted quantum search space following their corresponding mixing ansatzes have been considered in the QAOA. In particular, the so-called W_N states [43] can be especially useful as they represent one-hot encoding on the quantum circuit suitable for binary decision variables. For example, a W_3 state on three qubits is written as:

$$W_3 = \frac{1}{\sqrt{3}} \left(|100\rangle + |010\rangle + |001\rangle \right), \tag{19}$$

where each bit string always sums to one. With the property of the W state, we can construct an improved initial state to satisfy the temporal or spatial constraints of the TSP automatically, i.e., Equation (4) or Equation (5):

$$|s_W^n\rangle = \left(w_n |0\rangle \right)^{\otimes n} = \left(\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |2^i\rangle \right)^{\otimes n}, \tag{20}$$

where the temporal constraint is satisfied by putting together multiple W states in parallel (technically, there are many ways to build the $w_n |0\rangle$ state; we followed the method in Ref. [44]).

With a sufficiently powerful ansatz, one may also consider a permutation initial state, ignoring all superpositions:

$$|s_P^n\rangle = |(0, 1, \dots, n - 1)\rangle, \tag{21}$$

where its construction is simplest, using a few Pauli-X gates. We also considered an equal superposition of all permutation states, representing the minimal Hilbert space containing all the valid solutions; however, we found it to be the most challenging to initialize on the circuit.

These choices of initial states provide dramatically different initial search spaces, with dimensions ranging from $\mathcal{O}(2^{n^2})$, $\mathcal{O}(n^n)$, to $\mathcal{O}(1)$, respectively, along with their set relation $\{|s_P\rangle\} \subset \{|s_W\rangle\} \subset \{|s_H\rangle\}$. Notably, both the $|s_H\rangle$ and $|s_W\rangle$ are a superposition of solution states, but $|s_P\rangle$ is not. The selection of initial states plays a vital role in the QAOA, as it can reduce the number of potential candidates in the quantum evolution, albeit at the expense of an increased number of quantum gates. Lastly, these initial states will be used together with their respective mixer Hamiltonians of the QAOA, which are introduced in the next section.

3.4. Variational Ansatzes

Variational ansatzes are essential for optimizing the quantum state to represent the true solution. The variational ansatz U_p introduced in Equation (13) consists of the following two parts.

3.4.1. Problem Hamiltonian

The problem Hamiltonian is the qubitized cost function encoding the specific TSP instance to be solved in the QAOA approach. Specifically, these problem Hamiltonians are obtained by mapping the cost functions (Equations (6) and (10)) onto the quantum circuit according to the encoding strategy, Equation (16):

$$C_{\text{dist}}(x), C'_{\text{dist}}(x) \rightarrow H_{\text{dist}}, \tag{22}$$

$$C_{\text{penalty}}(x), C'_{\text{penalty}}(x) \rightarrow H_{\text{penalty}}, \tag{23}$$

where the obtained operators are a sum of the Pauli-Z and Pauli-ZZ operators, known as the Ising Hamiltonian [45]. Combining them, we obtain H_P , the problem Hamiltonian of the TSP instance:

$$H_P = H_{\text{dist}} + \lambda H_{\text{penalty}} = \sum_i c_i Z_i + \sum_{ij} c_{ij} Z_i Z_j. \tag{24}$$

As a consequence of qubit encoding, a ground state of H_P is guaranteed to be a true solution state that minimizes the respective TSP cost function. The Ising representation of the Hamiltonian is easily translated into a quantum circuit using a sequence of quantum gates.

3.4.2. Mixer Hamiltonian

The mixer Hamiltonian defines how the state space is to be explored and impacts how the quantum state evolves significantly with each iteration. Based on the Trotter product formula, the mixer Hamiltonian must not commute with the problem Hamiltonian, $[H_M, H_P] \neq 0$, to simulate a Trottered optimization like the QAOA. Many mixer Hamiltonians have been proposed [24,46,47] for different problems solved via QAOA. For different mixers, appropriate initial states as the eigenstates of the mixer Hamiltonian must be used in accordance with the adiabatic theorem. In evaluating the numerical performance of QAOA for TSP, we consider three types of mixers: X mixer, XY mixer, and row-swap mixer (RS mixer), with details explained below.

(a) The **X mixer** is the original mixer proposed in the QAOA that works together with a number of problems such as the max-cut problem [10]. It takes s_H for its state initialization. In the n -city TSP, the X mixer is:

$$H_{M_X} = \sum_{i=0}^{n-1} \sum_{t=0}^{n-1} X_{i,t}. \tag{25}$$

The X mixer strategy proves most useful for quantum annealing applications, especially on practical D-Wave systems [8]. It is easy to implement on most quantum backends, only requiring $\mathcal{O}(n^2)$ single-qubit X gates per layer in the QAOA.

(b) The **XY mixer** is another natural candidate for the mixing Hamiltonian, preserving the Hamming distance among the acted qubits [48], which is especially suited to the one-hot encoding realized by the initial states s_W . Here, we construct the XY mixer for the n -city TSP as:

$$H_{M_{XY}} = \sum_{i=0}^{n-1} \sum_{t=0}^{n-1} XY_{(i,t),(i,t+1)}, \tag{26}$$

where the XY gate is implemented via the Pauli-XX and Pauli-YY gates on the circuit. The block-wise construction allows for the conservation of probability for each city in the TSP, reinforcing the satisfaction of the temporal constraint, as in Equation (4). A generic XY gate across any two points (i, t) and (j, s) on the 2D lattice is:

$$XY_{(i,t),(j,s)} = X_{i,t} X_{j,s} + Y_{i,t} Y_{j,s}, \tag{27}$$

where X (Y) is the Pauli-X (Pauli-Y) matrix. Here, one should understand Equation (26) as a cyclic iteration of the XY gate. For example, $X_{n-1,n} \equiv X_{n-1,0}$ in the n -city case; other variants, such as non-cyclic and fully-connected XY gates, can also be used. The XY gate is often interchangeably referred to as the swap gate, as they both redistribute the amplitudes between two qubits while preserving the total amplitude of the quantum state. Alternatively, one could use the SWAP gate [49] instead of the XY gate to implement the XY mixer via:

$$\text{SWAP}_{u=(i,t),v=(i,t+1)} = \frac{1}{2} \left(X_u X_v + Y_u Y_v + Z_u Z_v + I_u I_v \right), \tag{28}$$

where a similar performance is produced. Therefore, we choose to use the simpler XY gate to implement the XY mixer throughout this work. Compared with the X mixer, the XY mixer is more expensive to implement by having $\mathcal{O}(n^2)$ XY gates per layer.

(c) The **row-swap (RS) mixer** has recently been proposed in the QAOA as a means of embedding hard constraints directly into the mixer Hamiltonian [23,24]. Although the RS mixer also uses the XY gate, it simultaneously swaps all non-overlapping rows of qubits (corresponding to different cities) as a whole. The RS mixer can be represented as:

$$H_{M_{RS}} = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \prod_{t=0}^{n-1} XY_{(i,t),(j,t)}, \tag{29}$$

where the first two sums represent all possible swapping between city- i and city- j , and the last product denotes the simultaneous swap of all corresponding entries in the associated cities. In this way, the RS mixer is capable of exploring the entire space of valid solutions when initialized on any single valid state, i.e., s_p . However, it should be noted that the RS mixer incurs a significant computational cost during the simulation due to the involvement of many tensor products of the Pauli-XX or Pauli-YY matrices. One can mitigate this expense by relying on a set of creation and annihilation operators constructed from four-qubit gates [24]. Nevertheless, the $H_{M_{RS}}$ ansatz remains computationally expensive, requiring $\mathcal{O}[(n-1)(n-2)/2]$ four-qubit gates per layer with each four-qubit gate itself being expensive to construct.

3.5. Measurement and Optimization Protocol

Based on the unitary ansatz and its appropriate initial state, the cost expectation of the QAOA is evaluated by measurements performed on quantum devices and subsequently optimized using gradient-free optimizers such as COBYLA [50–52] and SPSA [53,54]. The optimization process continues until convergence or until the maximum iteration threshold is reached. The resulting solution to the TSP is then determined by identifying the most dominant quantum state (or binary decision variable encoded in a bit string). To account for statistical fluctuations in measurements, we run each quantum simulation multiple times (typically 5–10) with different random seeds and report the result with the lowest converged expectation value. Considering that the expectation values are TSP-specific, we use the standard evaluation metric called the approximation ratio (AR) to evaluate the performance by normalizing against the ideal cost in different TSPs. The AR is calculated as:

$$AR = \frac{\text{simulation cost}}{\text{ideal cost}} = \frac{\langle \vec{\beta}, \vec{\gamma} | C(x) | \vec{\beta}, \vec{\gamma} \rangle}{C_{\text{ideal}}} \geq 1, \tag{30}$$

where a lower AR corresponds to a lower expectation cost, indicating a closer approximation to the exact solution. Classical optimizers play a vital role in the optimization, and their advantages can be further utilized in the QAOA. The expectation values of individual bit strings are cached and retrieved on the classical optimizer to enable fast computation of the final cost expectation during each iteration. The option to use constraint bounds of $[0, 2\pi)$ for the ansatz parameters in the case of COBYLA can also accelerate the convergence, which is the main reason we primarily focused on simulations using the COBYLA optimizer in our study, although a comprehensive analysis with other available optimizers can be explored in future research.

To optimize the QAOA, we employed the **layer-wise learning (LL)** protocol introduced in Ref. [28]. In comparison to complete depth learning (CDL), LL proved to be advantageous in reducing the optimization cost, particularly as the number of qubits and circuit depth increased. It also helps mitigate the likelihood of barren plateaus (BP) [28]. In short, the LL is a two-part optimization protocol, as illustrated in Figure 1.

(A) **Progressive pre-training:** In the first part (Figure 1a), we construct the QAOA ansatz by gradually adding layers. Initially, we train and optimize over the leading few

layers (typically two layers). Then, for a p -layer QAOA simulation, we freeze the parameters in the first $(p - 1)$ -th layers, obtained from previous simulations, and exclusively optimize the parameters in the p -th layer. Optimal parameters of the current layer that yield the lowest cost expectation are selected. Note the initial values for the parameters of the p -th layer are zero. If no lower cost is found at the p -th layer compared with previous costs, we use zeros for the parameters of that layer. In this way, the cost is always non-increasing over the entire simulation. This progressive optimization protocol proves to be efficient and leads to an increasingly optimized solution as the number of layers increases. It also reduces the computational cost in parameter searching for very thick layers. We denote this protocol with the letter A and an integer to indicate the depth being optimized.

- (B) **Randomized retraining:** In the second part (Figure 1b), we take the pre-trained QAOA ansatz from part (A) and randomly select a larger portion of the parameters to be trained at a time. Typically, we free 50% of the parameters in each iteration of retraining. Although it is more computationally expensive, this retraining is still less costly than the CDL, which allows us to train the QAOA ansatz as a whole. This mitigates the risk of becoming trapped in local minima, which could occur when using the protocol of part (A) exclusively. We use the protocol of part (B) with a number to indicate which iteration of retraining is being conducted.

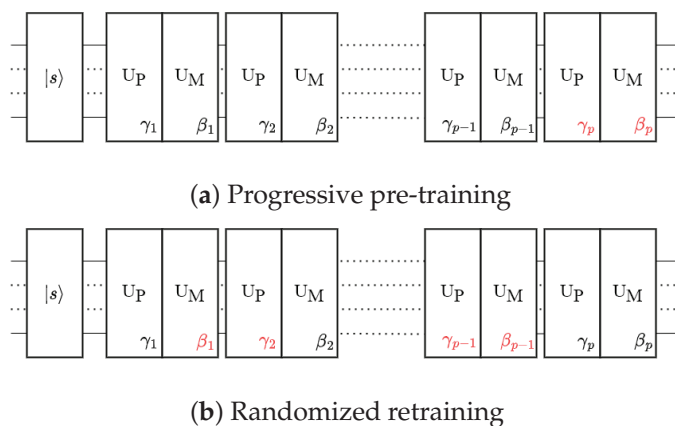


Figure 1. Two-part layer-wise learning protocol of the QAOA. Horizontal lines represent the qubits; rectangular boxes are the unitary operators. Fixed parameters are in black; free parameters are in red.

It should be mentioned that there are other variations to the LL, such as sequential block-wise learning used in Ref. [55], where one block/layer is optimized at a time while fixing all other blocks. Layer-wise learning may also be prone to systematic layer saturations [56] that require special treatments, which we leave for future study. Here, our goal is not to prove the superiority of the LL over classical algorithms or any other quantum variational protocols. Instead, we take the LL as a common optimization method to compare the practical performance of various QAOA mixer ansatzes in solving the TSP, which is the highlight of this work. For the numerical results presented in this work, we always use the LL optimization protocol, as its computational cost and solution accuracy consistently outweigh those of the CDL. In Figure 2, we show an example of layer-wise learning applied to the QAOA with the X mixer, showing the optimization in both one protocol step and the full LL procedure; similar performances are also found for other mixers. It is important to point out that the mean square error of the ARs calculated from different batches is negligible, especially toward the end of the optimizations. By contrast, the uncertainty of the ARs calculated from different TSP graphs is significant. The same observation is found for other measurable quantities as well. Therefore, throughout this paper, we exclusively refer to the uncertainty due to various TSP instances.

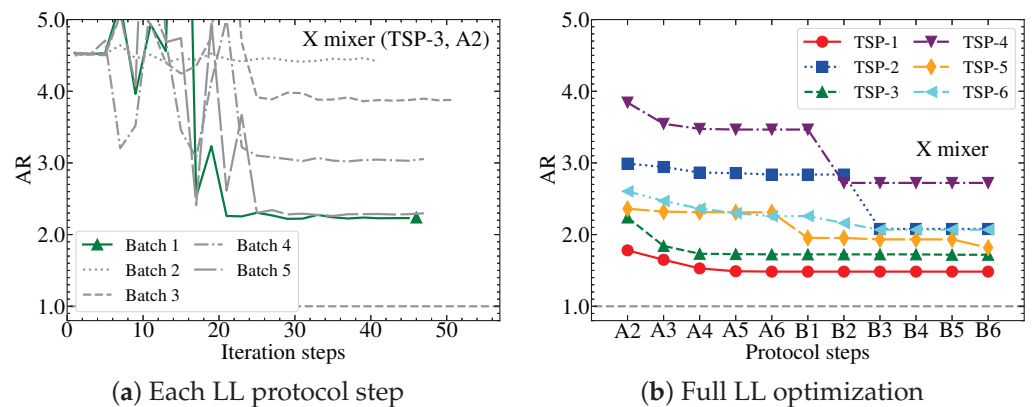


Figure 2. Example of the layer-wise learning protocol applied to the QAOA X mixer simulation. The left panel (a) shows the simulation for a selected LL protocol step A2 of a specific TSP instance, TSP-3. The right panel (b) shows the overall LL optimization for 6 different TSP instances. Here, a TSP instance is a random TSP graph with 3 nodes and a maximum edge weight of 20.

4. Numerical Results

With both the TSP optimization and QAOA method introduced, we perform numerical quantum simulation on the IBM Quantum QASM simulator using `aer.QasmSimulator`. The problem and mixer Hamiltonian operators are constructed using the `qiskit.opflow` library. For the circuit implementations of the three mixers, we use Pauli-Z and Pauli-ZZ gates for the X mixer and the XY mixer, and we use the `PauliEvolutionGate` library for the XY mixer. We focus on quantum simulations using the layer-wise learning protocol for 3-, 4-, and 5-city TSPs on a sufficiently powerful local Ubuntu machine (Ubuntu 22.04.2 LTS machine using one CPU core with 32.0 GB memory and Intel i9 processor of 3.50 GHz), and we compare their performances in terms of numerical accuracy and resource costs. To obtain converged results, we always use a sufficient number of TSP instances, varying from 7 to 10 graphs, depending on the number of cities, mixer, and simulated noise, each with 5–10 repeated runs of quantum simulation.

4.1. Simulation Accuracy

We follow the LL optimization protocol introduced in Section 3.5 and use $(n - 1)^2$ qubits based on the improved TSP formulation (Equation (10)) for each quantum simulation with n cities. In Figure 3, we present the QAOA simulation results when solving various instances of 4-city and 5-city TSPs using the X, XY, and XY mixers. We temporarily leave out the 3-city TSPs in the figure, since all the mixers performed reasonably close to the ideal case. The XY mixer is able to reach an AR of 1.00 given any TSP graph, while the X mixer reaches a value of 1.28 by comparison. The RS mixer is not discussed because any valid solution would be a true solution in the 3-city case for the RS.

The performance is evaluated with three criteria: (a) approximation ratio (AR), (b) percentage of the true solution, and (c) rank of the true solution. The two-part LL optimization is indicated by letters A and B, followed by the specific depth and iteration numbers, respectively. We use a sufficient number of layers in the QAOA simulation (4 layers for 3-city cases and 6 for 4-/5-city cases) to ensure convergence. The uncertainty bars depicted in Figure 3 represent the standard deviations of the respective results calculated for various TSP graph instances. A comprehensive comparison of all the results can be found in Table 1, which includes the results for the 3-city TSP simulations as well.

(a) **Approximation ratio (AR):** Expectation cost, or equivalently AR, is the primary observable that is measured during the quantum simulation. It directly influences the classical optimizer’s ability to find the optimal parameters. Figure 3a,b demonstrate that both pre-training and retraining parts of the LL are necessary to optimize the AR for various TSP instances. Among the three types of QAOA mixers, the RS mixer achieves the lowest AR, reaching values as low as 1.01 ± 0.01 (in the 4-city case) and 1.18 ± 0.14 (in the 5-city

case). On the other hand, the X mixer performs the poorest, particularly as the problem size increases, partially due to the limitations of the ansatz's expressibility. It is worth noting that even the heuristic VQE ansatz outperforms the X mixer in the 4-city case, with a lower AR around 2.19 ± 0.37 , compared to 2.33 ± 0.83 (see Table 1). Considering the temporal constraints during construction, the XY mixer exhibits intermediate performance, with AR values of around 1.44 ± 0.23 and 1.89 ± 0.66 for 4- and 5-city TSPs, respectively.

Lastly, we can also see that all the mixers are only able to reach sub-optimal solutions in the 5-city TSP case. It would be interesting to fully investigate the expressibility of the QAOA ansatzes in obtaining the optimal solution and distinguish that from the optimization itself. We will leave this for an extensive study in the future that involves more TSP cities.

(b) **True percentage:** The percentage of the true solution is also known as the **overlap** between the quantum state and the expected true solution. While the true percentage is determined only after the simulation, it is desirable to have it as large as possible for the accurate extraction of the optimal solution. In Figure 3c,d, we present the true percentages for the three mixers as the TSP problem size increases. Undeniably, RS is the dominating mixer, reaching around $96.3 \pm 3.8\%$ and $41.1 \pm 29.5\%$. However, the large uncertainty suggests a highly unstable pattern in the obtained solution; see Appendix B for an explanation. On the other hand, the X mixer gives the lowest percentages, reflecting a poor performance in accurately identifying the true solution. Lastly, the XY mixer is again holding a middle ground, with true percentages of approximately at $36.6 \pm 5.7\%$ and $7.4 \pm 0.6\%$, respectively.

(c) **Rank:** The rank of the true solution specifies how many other states possess a higher probability than the state corresponding to the true solution, which is a crucial indicator of the simulation's accuracy. Achieving a rank of 1 for the true solution signifies consistent identification of the correct solution, as it means the quantum state with the highest probability is always the true solution's state (so we want to have a rank as low as possible). The results are presented in Figure 3e,f. In the case of the X mixer, it exhibits a significantly high rank, indicating a low likelihood of picking the correct solution among the top quantum states. On the other hand, the ranks of the XY and RS mixers are comparable, both reaching around rank-1 for 4-city TSPs and around rank-2 for 5-city TSPs. Notably, for the 5-city case, we observe lower ranks of the XY mixer in the early stages compared to the final stages, showcasing the effectiveness of the XY mixer in even shallower QAOA for certain TSP instances.

Based on the observations in AR, true percentage, and rank, several conclusions can be made. First, we can see that X mixers consistently under-perform in all three criteria, compared to the other two mixers. This behavior is expected because the Hadamard initialization produces a uniform superposition of all possible states, i.e., 2^{16} states in the 5-city case, without any constraints on the solution. As a result, it becomes challenging for the classical optimizer to filter out the invalid and false solutions based solely on the problem Hamiltonian. In particular, when the problem size increases, the X mixer alone is not suitable for the QAOA simulation of the TSP. Secondly, we observe that the RS mixer stands out as the dominating mixer in terms of AR and true percentage, which makes it a reliable candidate for QAOA. In terms of rank, the performances of the XY and RS mixers are quite similar. The strategies employed by the two mixers are very different: the RS mixer relies heavily on the expressibility of the mixer itself, while the XY mixer combines the initialization and the mixing Hamiltonian to achieve its results. By utilizing a single-bit string as the initial state, RS may potentially overlook the benefits of having superposition states in a quantum simulation. In a sense, the XY mixer takes a more balanced approach, whereas the RS mixer takes a more assertive approach; this distinction between the two mixers can have implications for the resource cost, which is discussed in the following section.

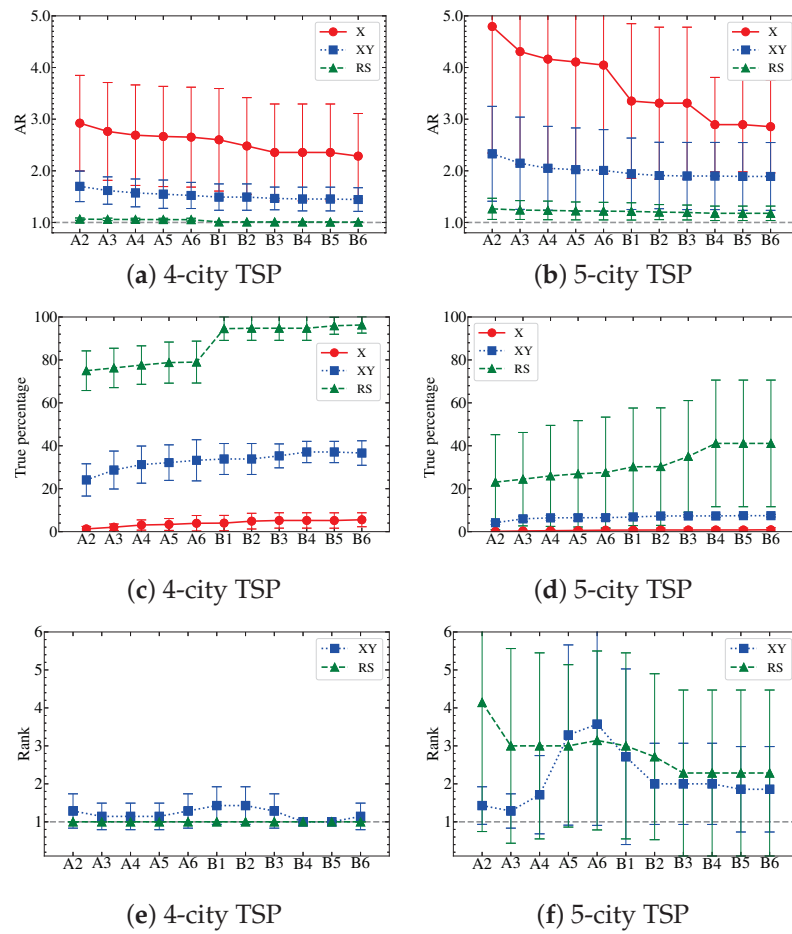


Figure 3. Performance comparison of the 3 QAOA mixers for samples of the 4-city TSP (left column) and 5-city TSP (right column). In both cases, we compare the AR in panels (a,b), the percentage of the true solution in panels (c,d), and the rank of the true solution in panels (e,f). The uncertainty bars are standard deviations obtained from simulations of different TSPs. Notably, we leave out the ranks for the X mixers due to their significantly higher values, which further complicates the presentation.

Table 1. Comprehensive comparison of the numerical accuracy for the QAOA mixers and heuristic ansatzes used to solve the TSP. The standard deviations of the quantities obtained from variation in TSP graphs are provided in the parentheses. Problem-specific VQE to the TSP such as in Ref. [57] may produce a significantly better result. Notably, the RS mixer is excluded for the 3-city TSP because any starting bit string is a true solution for the RS, which makes it trivial to simulate.

# City	Mixers	After Pre-Training			After Retraining		
		AR	True %	Rank	AR	True %	Rank
3	VQE	1.02 (0.02)	87.6 (28.6)	1.1 (0.3)	1.01 (0.02)	90.9 (20.2)	1.1 (0.4)
	X	1.34 (0.14)	38.5 (24.8)	2.3 (2.2)	1.28 (0.10)	44.2 (28.2)	1.9 (1.4)
	XY	1.00 (0)	100.0 (0)	1.0 (0)	1.00 (0)	100.0 (0)	1.0 (0)
4	VQE	2.23 (0.29)	12.8 (13.3)	35.6 (44.1)	2.19 (0.37)	11.8 (13.9)	39.1 (42.2)
	X	2.65 (0.97)	3.9 (3.6)	67.7 (140.8)	2.33 (0.83)	5.5 (3.2)	4.7 (5.3)
	XY	1.52 (0.25)	33.2 (9.6)	1.3 (0.5)	1.44 (0.23)	36.6 (5.7)	1.1 (0.4)
	RS	1.05 (0.03)	79.0 (9.8)	1.0 (0)	1.01 (0.01)	96.3 (3.8)	1.0 (0)
5	X	4.05 (2.10)	0.7 (0.8)	1814.1 (4348.8)	2.85 (0.90)	0.9 (1.0)	94.3 (105.0)
	XY	2.01 (0.79)	6.5 (1.3)	3.6 (2.7)	1.89 (0.66)	7.4 (0.6)	1.9 (1.1)
	RS	1.22 (0.17)	27.6 (25.8)	3.1 (2.4)	1.18 (0.14)	41.1 (29.5)	2.3 (2.2)

4.2. Resource Evaluations

Besides numerical accuracy, resource cost estimation is another crucial factor to consider in quantum simulation, as any computational resource is always finite. On the quantum computer and simulator, many factors will contribute to the performance of the simulation, including attributes of the transpiled quantum circuits, such as the number of qubits, the number of single-qubit (double-qubit) gates, and the quantum circuit depth. In a practical calculation, properties of the quantum device, such as qubit connectivity, coherent error, and incoherent noise, will also come into play. For this section, we focus on the quantum circuits of the three QAOA mixers and compare their resource costs on ideal devices; practical calculation is discussed in the subsequent section using noisy simulation.

In Table 2, we compare the properties of the quantum circuits of the three mixers after transpilation for both finite and generic TSP cases. As expected, the complexity of the circuit, measured in terms of quantum gates and circuit depth, generally increases with the number of cities, resulting in a longer simulation time. Notably, the RS mixer incurs a significantly higher resource cost compared to the X and XY mixers, as reflected in the simulation time in practice. As discussed earlier, this increased cost is primarily attributed to the utilization of four-qubit gates in the RS mixer, leading to a quadratic scaling, i.e., $\mathcal{O}(n^4)$, of single-qubit and double-qubit gates. The abundance of double-qubit gates is anticipated to pose serious challenges in executing the simulation on a real quantum device or when employing a noise model [58]. Interestingly, despite requiring fewer resources, the X mixer actually takes a longer time to run in practice compared to the XY mixer, particularly as the number of qubits increases. This observation is likely due to the computational burden of the optimizer when evaluating the expected cost for a dense superposition of bit strings. On the other hand, the XY mixer requires relatively low computational resources, scaling linearly with the circuit depth and quadratically with the number of quantum gates, which is a more economical choice for running QAOA simulations. Considering both optimization accuracy and computational cost, the XY mixer emerges as a more balanced choice for the QAOA. Nonetheless, a resource cost of $\mathcal{O}(n^2)$ gates and qubits for the XY mixer is still quite expensive as n increases. Notably, building the XY mixer at the pulse level [59] has the potential to further enhance its numerical performance. Lastly, it should be acknowledged that the resource costs of all mixers would be even higher when simulating on current NISQ or future fault-tolerant quantum computers. In the interest of addressing this aspect, we present noise-model simulations in the subsequent section.

Table 2. Quantum resource estimation per QAOA layer of various mixers considered in the 3-, 4-, and 5-city TSPs. The circuit depth, the count of single-qubit gates, and the count of the double-qubit gates are evaluated after transpilation (light transpilation, no approximation with `qiskit.compiler.transpile`) to the standard basis gate sets {CX, I, RZ, SX, X} used by the IBM Quantum. Exact numbers for the circuit depths and quantum gates are obtained whenever available; otherwise, asymptotic scalings are provided. Estimating the circuit depth exactly is difficult for the RS mixer. In comparison, it appears to increase linearly with a large slope of 1181 for small city numbers.

# City	# Qubits	Mixers	Circuit Depth	Single-Qubit Gates	Double-Qubit Gates
3	4	X	5	20	0
		XY	26	64	16
4	9	X	5	45	0
		XY	37	144	36
		RS	668	477	432
5	16	X	5	80	0
		XY	48	256	64
		RS	1553	1808	1728
n	$(n - 1)^2$	X	5	$5n^2$	0
		XY	$26 + 11(n - 3)$	$\mathcal{O}(n^2)$	$4n^2$
		RS	$\mathcal{O}(n)^1$	$\mathcal{O}(n^2(n - 1)^2)$	$\mathcal{O}(n^2(n - 1)^2)$

4.3. Robustness Against Noise

Estimating the performance of the QAOA simulation in the presence of noise is crucial to implementations on NISQ and fault-tolerant devices in the future. In this section, we employ the NoiseModel class from Qiskit to study the sensitivity of the simulation on different noise levels. In particular, we focus on noisy QAOA simulations with XY and RS mixers for the same set of 4-city TSP problems. In Figure 4, we compare the performance of various noise simulations in terms of AR, true percentage, and rank. We consider noise models with different degrees of single-qubit errors: 0.005%, 0.01%, 0.05%, and 0.1%. Besides single-qubit errors, we set the double-qubit errors to be 10 times their respective single-qubit errors, which is a reasonable approximation for realistic two-qubit gates such as the CX gate. For the current study, we have omitted other potential errors for simplicity, such as the qubit connectivity and thermal relaxation time, which can also be implemented with the noise model.

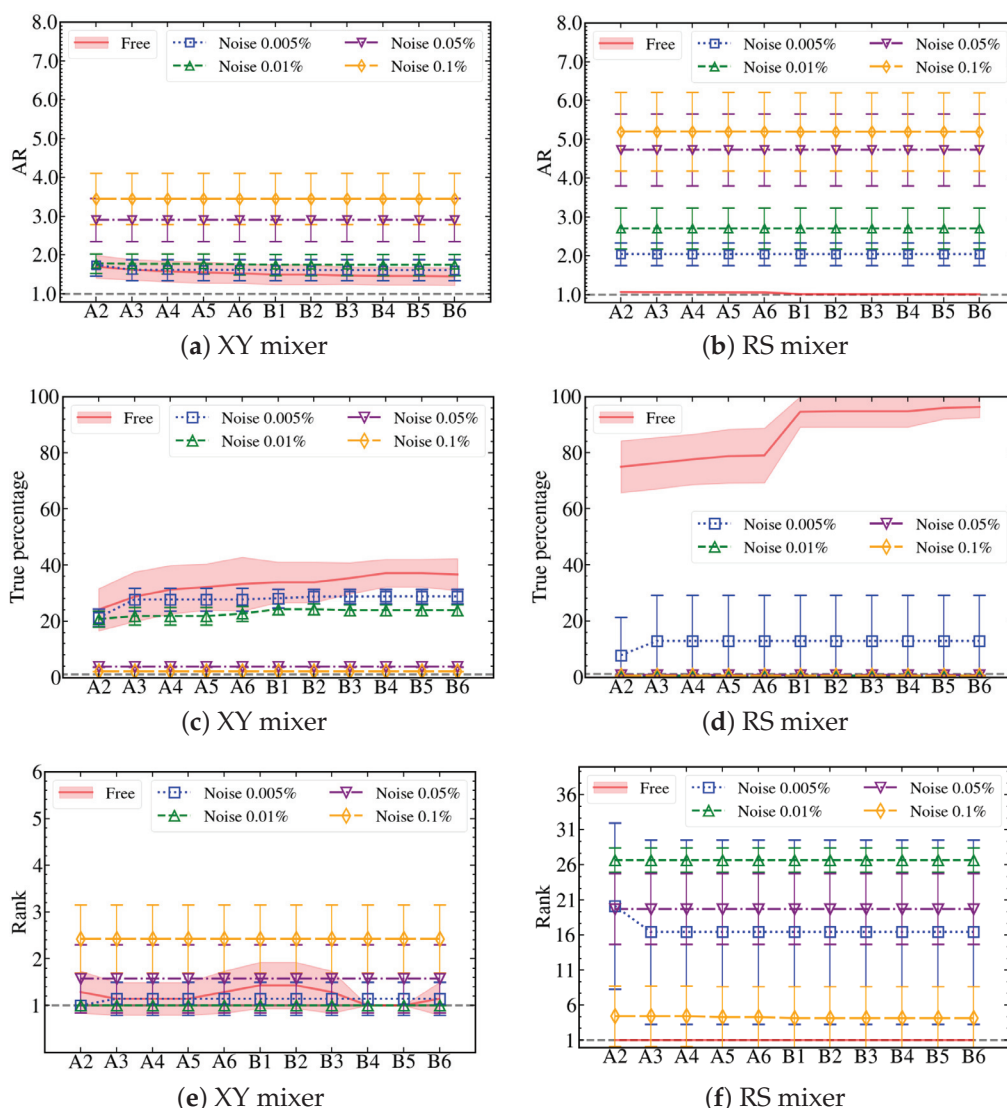


Figure 4. Noisy QAOA simulation results of the XY and RS mixers compared with the noise-free simulation of the 4-city TSP graph. In the legend, we show the single-qubit error used for each noisy simulation. The uncertainty bars/bands are standard deviations obtained from simulations of different TSPs. The same scale is used for XY and RS, except for the plot of their ranks.

From the results presented in Figure 4, it is evident that the gate errors in the noise model directly impact the quality of the simulation. As expected, QAOA simulations with larger errors perform poorly compared to smaller ones. Interestingly, there seems to be a noise threshold in the simulation results: noisy simulations with error less than or equal to 0.01% exhibit qualitatively different behavior compared to those with higher errors, as shown in Figure 4a–c. Additional details of the noisy simulation are provided in Table 3, where we can clearly see that the LL protocol fails to optimize the QAOA simulation at 0.1% and 0.05% noise levels. Comparing the two ansatzes, the XY mixer outperforms the RS mixer in all indicators for all noisy simulations. Surprisingly, the XY mixer achieves performance similar to the ideal simulation with errors less than or equal to 0.01%, indicating its potential resilience against simulation noise. Our result suggests that the XY mixer is a more suitable choice among the three mixers when considering noise effects in the QAOA simulation.

Table 3. Details of noisy simulation for the 4-city TSP case. Noise percentage refers to single-qubit errors used in the noise model simulation. The standard deviation of the quantities obtained from variation in TSP graphs is provided in parentheses.

Noise %	Protocol	XY Mixer			RS Mixer		
		AR	True %	Rank	AR	True %	Rank
0.1	A2	3.44 (0.67)	2.09 (0.03)	2.43 (0.73)	5.20 (1.02)	0.4 (0.1)	4.4 (4.3)
	A6	3.44 (0.67)	2.09 (0.03)	2.43 (0.73)	5.20 (1.02)	0.4 (0.1)	4.3 (4.4)
	B6	3.44 (0.67)	2.09 (0.03)	2.43 (0.73)	5.19 (1.00)	0.4 (0.1)	4.1 (4.5)
0.05	A2	2.90 (0.56)	3.7 (0.1)	1.6 (0.7)	4.73 (0.92)	0.6 (0.1)	19.7 (5.1)
	A6	2.90 (0.56)	3.7 (0.1)	1.6 (0.7)	4.73 (0.92)	0.6 (0.1)	19.7 (5.1)
	B6	2.90 (0.56)	3.7 (0.1)	1.6 (0.7)	4.73 (0.92)	0.6 (0.1)	19.7 (5.1)
0.01	A2	1.78 (0.25)	20.8 (2.9)	1.0 (0)	2.70 (0.53)	0.6 (0)	26.7 (1.8)
	A6	1.76 (0.27)	22.7 (2.7)	1.0 (0)	2.70 (0.53)	0.6 (0)	26.7 (1.8)
	B6	1.74 (0.27)	23.9 (2.0)	1.0 (0)	2.70 (0.53)	0.6 (0)	26.7 (1.8)
0.005	A2	1.74 (0.29)	21.3 (3.1)	1.0 (0)	2.04 (0.29)	7.7 (13.5)	20.1 (11.8)
	A6	1.62 (0.27)	27.8 (4.0)	1.1 (0.4)	2.04 (0.29)	12.9 (16.3)	16.4 (13.1)
	B6	1.61 (0.28)	28.8 (2.5)	1.1 (0.4)	2.04 (0.29)	12.9 (16.3)	16.4 (13.1)

4.4. Problem Dependence

It is important to investigate the problem dependence of the QAOA simulation of the TSP in preparation for the full-fledged quantum simulation. Here, we study several TSP problem dependencies, such as the topology of the TSP graphs and the penalty weight. The topology of the TSP graph could potentially have a significant impact on the performance of the quantum simulation algorithm. One characteristic we consider is the “skewness” of the TSP graphs, which represents the level of asymmetry. To measure the skewness, we analyze the distribution of the edge weights ω_{ij} in the graph using Fisher–Pearson’s moment coefficient [60,61]. Specifically, we calculate the skewness parameter g by:

$$g = \frac{m_3}{m_2^{3/2}}, \quad m_k = \sum_{0 \leq i < j < n} \frac{(\omega_{ij} - \bar{\omega})^k}{|\omega|}, \quad (31)$$

where $\bar{\omega}$ is the mean of the edge weights, and $|\omega|$ is total number of edges in the graph. Here, m_3 is the third moment of the edges, and m_2 is the variance, the square of the standard deviation. Intuitively, the skewness can also be computed as the average value of the cubed z-scores. For instance, a skewness value of 0 indicates a symmetric/normal distribution of the edge, and skewness values of greater than 1 or less than -1 typically indicate highly-skewed distributions. Negative (positive) skewness indicates a left-skewed/right-leaning (right-skewed/left-leaning) distribution.

In Figure 5, we present the quantum simulation using X, XY, and RS mixers on various 4-city TSP graphs with varying skewnesses. Here, we focus on the approximation ratio in the final step of layer-wise learning to assess the dependence on the TSP graph's skewness. Taking every bit string solution into account, AR represents the overall effectiveness of the simulation, which is suitable to analyze the skewness. We observe that the simulation tends to perform less effectively with right-skewed edge distributions, possibly due to the presence of more low-weight edges in positively-skewed graphs. Further investigations that include sampling uncertainties are necessary to fully study the consequences of varying graph topology for TSPs with more cities.

Additionally, the penalty weight λ in the TSP cost equation (Equation (10)) is essential to examine, for it directly controls the gaps between valid and invalid solutions. By a similar analysis for the skewness, we find that the simulation performs optimally when λ is in the range of $[1.0E_{G,\max}, 4.5E_{G,\max}]$, where $E_{G,\max}$ represents the maximum TSP edge weight. This analysis further supports the choice of the penalty weight used in this study.

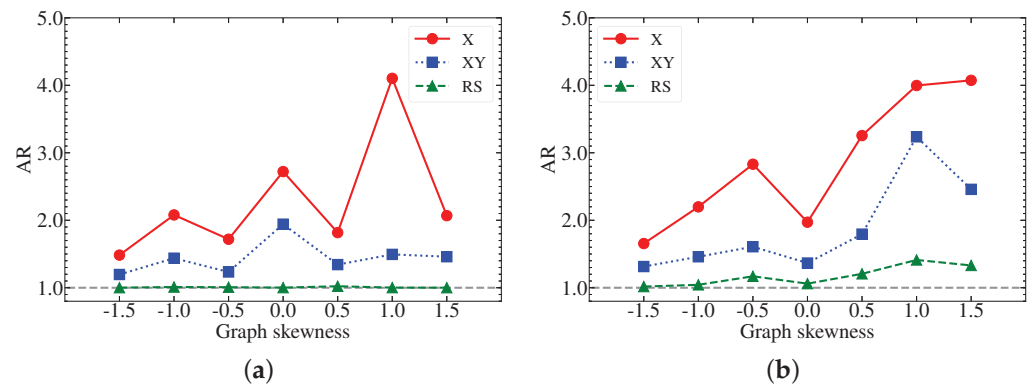


Figure 5. Results of the approximation ratio (AR) of the three QAOA mixers for the 4-city TSP (panel a) and the 5-city TSP (panel b) of distinct graph skewnesses.

5. Summary and Discussions

In this paper, we solved the symmetric TSP (traveling salesman problem) as an optimization problem by using three distinct ansatzes to the QAOA (quantum approximate optimization algorithm) approach. By adopting a layered learning optimization protocol, we performed numerical quantum simulations on gate-based quantum simulators for various 3-, 4-, and 5-city TSPs. In particular, we presented and compared the performance of the three types of mixer ansatzes for the QAOA: the X mixer, the XY mixer, and the RS mixer. For the few-city TSPs studied in this work, we demonstrated that a well-balanced quantum simulation, such as using the XY mixer, is potentially more suitable in terms of both numerical accuracy and computational cost. These findings are further validated through the noise model simulations. Additionally, we highlighted other factors that may play a role in the quantum simulation, such as the TSP graph skewness and cost function penalty.

Our research is a significant step towards finding a successful strategy for the TSP optimization problem using the gate-based QAOA approach, which is particularly interesting for the current NISQ paradigm. The QAOA simulation complements traditional quantum annealing methods in the infinite time region, where efficient qubit reduction techniques, improved optimization protocols, and resource-efficient mixer ansatzes investigated in this work are expected to be valuable for realistic quantum device simulations. Moving forward, we plan to extend our investigations to larger-city TSPs, employing deeper QAOA circuits on noisy quantum backends. By utilizing an adaptive shot-frugal optimizer [62] and implementing digitized-counterdiabatic quantum approximate optimization methods [39,40], we aim to further enhance the accuracy and efficiency of our TSP simulations.

Author Contributions: Conceptualization, W.Q.; methodology, W.Q.; software, W.Q.; validation, W.Q. and R.A.M.B.; formal analysis, W.Q., R.A.M.B., M.M.E.-W. and J.P.V.; investigation, W.Q. and R.A.M.B.; data curation, W.Q. and R.A.M.B.; writing—original draft preparation, W.Q.; writing—review and editing, W.Q., R.A.M.B., M.M.E.-W., A.K., G.L. and J.P.V.; funding acquisition, W.Q., A.K. and J.P.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the U.S. Department of Energy (DOE), Office of Science, under Grant Nos. DE-FG02-87ER40371, DE-SC0023692, and DE-SC0023707 (Quantum Horizons/NuHaQ). This research was also supported in part by the Palmer Department Chair Endowment at Iowa State University. W.Q. was supported by Xunta de Galicia (Centro singular de investigación de Galicia accreditation 2019–2022), European Union ERDF, the “Maria de Maeztu” Units of Excellence program under project CEX2020-001035-M, the Spanish Research State Agency under project PID2020-119632GB-I00, the European Research Council under project ERC-2018-ADG-835105 YoctoLHC, and the European Union’s MSCA Postdoctoral Fellowships HORIZON-MSCA-2022-PF-01 under Grant Agreement No. 101109293.

Data Availability Statement: Not applicable.

Acknowledgments: We wish to thank Meijian Li, I-Chi Chen, Michael Kreshchuk, and Soham Pal for valuable discussions. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors and do not reflect the official policy or position of IBM or the IBM Quantum team.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TSP	Traveling Salesman Problem
QA	Quantum Annealing
QAOA	Quantum Approximate Optimization Algorithm
VQA	Variational Quantum Algorithm
NISQ	Noisy Intermediate-Scale Quantum
AQC	Adiabatic Quantum Computation
VQE	Variational Quantum Eigensolver
DC-QAOA	Digitized-Counterdiabatic QAOA
RS mixer	Row-Swap Mixer
LL	Layer-wise Learning
BP	Barren Plateaus
CDL	Complete Depth Learning
AR	Approximation Ratio

Appendix A. Pauli Gates

The Pauli gates are defined as:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (\text{A1})$$

with their subscripts indicating the gate’s qubit location when included.

Appendix B. Comparison of the Three Mixers on a Single TSP Instance

We compare the performance of the X, XY, and RS mixers using the “total percentage” plot on the same TSP graph in Figure A1. As explained in the main text, the total percentage of solutions to the TSP consists of true, false, and invalid solutions obtained after the simulation. The respective rank of the true solution is also presented at the top of each step in the layer-wise learning procedure. We see that the ranks are extraordinarily high for the X mixer. For the RS mixer, it can be situational whether we obtain the true solution as the most dominant one; therefore, we show the results obtained from two different simulations with the RS mixers in panels (c) and (d), which explain the high standard deviations shown

in Figure 3d,f. The XY result, by contrast, is less sensitive to the simulation. More examples of the total percentage plots for the three mixers can be found in Ref. [63].

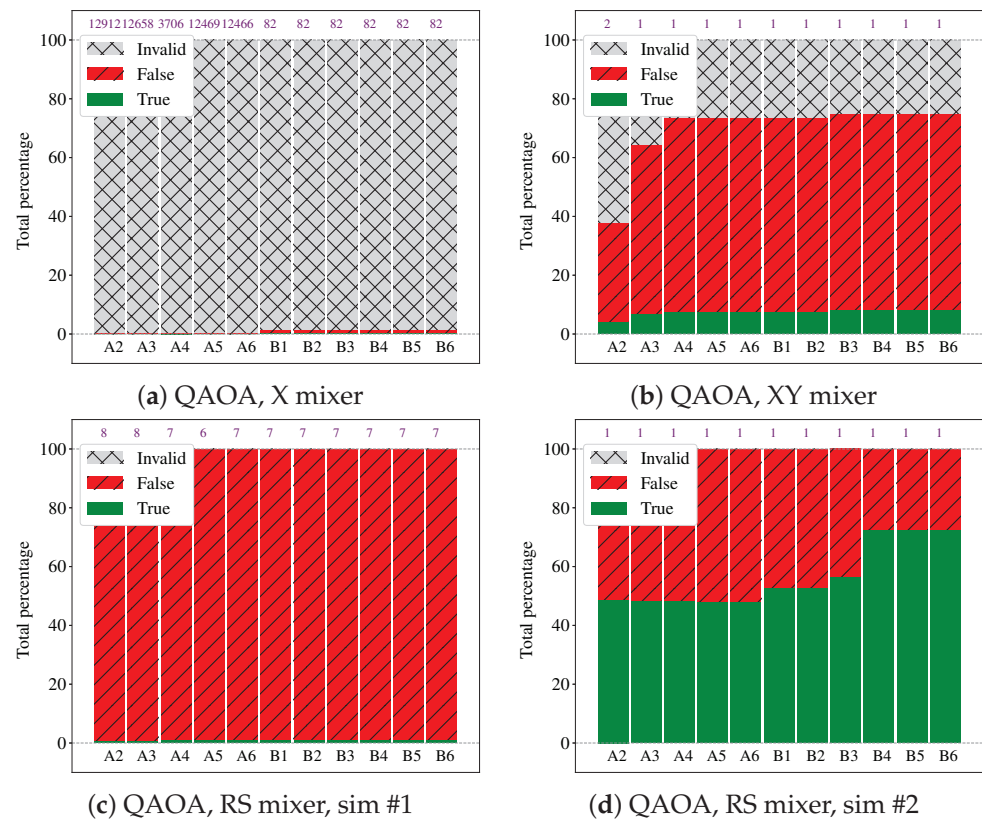


Figure A1. Performance comparison among the three mixers on a single TSP graph. The respective rank of the true solution in each optimization step is included at the top of the percentage.

References

1. Biggs, N.; Lloyd, E.K.; Wilson, R.J. *Graph Theory, 1736–1936*; Clarendon Press: Cary, NC, USA, 1986.
2. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
3. Kohonen, T. The self-organizing map. *Neurocomputing* **1998**, *21*, 1–6. [CrossRef]
4. Ambainis, A.; Balodis, K.; Iraids, J.; Kokainis, M.; Prūsis, K.; Vihrovs, J. Quantum Speedups for Exponential-Time Dynamic Programming Algorithms. In Proceedings of the SODA '19: Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, San Diego, CA, USA, 6–9 January 2019; pp. 1783–1793.
5. Finnila, A.; Gomez, M.; Sebenik, C.; Stenson, C.; Doll, J. Quantum annealing: A new method for minimizing multidimensional functions. *Chem. Phys. Lett.* **1994**, *219*, 343–348. [CrossRef]
6. Kadowaki, T.; Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **1998**, *58*, 5355–5363. [CrossRef]
7. Warren, R.H. Solving combinatorial problems by two D-Wave hybrid solvers: A case study of traveling salesman problems in the TSP Library. *arXiv* **2021**, arXiv:2106.05948. <https://doi.org/10.48550/ARXIV.2106.05948>.
8. Jain, S. Solving the Traveling Salesman Problem on the D-Wave Quantum Computer. *Front. Phys.* **2021**, *9*, 760783. [CrossRef]
9. Villar-Rodriguez, E.; Osaba, E.; Oregi, I. Analyzing the behaviour of D'WAVE quantum annealer: Fine-tuning parameterization and tests with restrictive Hamiltonian formulations. In Proceedings of the 2022 IEEE Symposium Series on Computational Intelligence (SSCI), Singapore, 4–7 December 2022. [CrossRef]
10. Farhi, E.; Goldstone, A.; Gutmann, S. A quantum approximate optimization algorithm. *arXiv* **2014**, arXiv:1411.4028.
11. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.; Gambetta, J. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242–246. [CrossRef]
12. Qian, W.; Basili, R.; Pal, S.; Luecke, G.; Vary, J.P. Solving hadron structures using the basis light-front quantization approach on quantum computers. *Phys. Rev. Res.* **2022**, *4*, 043193. [CrossRef]
13. Egger, D.J.; Gambella, C.; Marecek, J.; McFaddin, S.; Mevissen, M.; Raymond, R.; Simonetto, A.; Woerner, S.; Yndurain, E. Quantum Computing for Finance: State-of-the-Art and Future Prospects. *IEEE Trans. Quantum Eng.* **2020**, *1*, 3101724. [CrossRef]
14. Preskill, J. Quantum Computing in the NISQ era and beyond. *arXiv* **2018**, arXiv:1801.00862.
15. Zhou, L.; Wang, S.T.; Choi, S.; Pichler, H.; Lukin, M.D. Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. *Phys. Rev. X* **2020**, *10*, 021067. [CrossRef]

16. Mesman, K.; Al-Ars, Z.; Möller, M. QPack: Quantum Approximate Optimization Algorithms as universal benchmark for quantum computers. *arXiv* **2021**, arXiv:2103.17193.
17. Harrigan, M.P.; Sung, K.J.; Neeley, M.; Satzinger, K.J.; Arute, F.; Arya, K.; Atalaya, J.; Bardin, J.C.; Barends, R.; Boixo, S.; et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **2021**, *17*, 332–336. [CrossRef]
18. Cook, J.; Eidenbenz, S.; Bärttschi, A. The Quantum Alternating Operator Ansatz on Maximum k-Vertex Cover. In Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, 12–16 October 2020; pp. 83–92. [CrossRef]
19. Azad, U.; Behera, B.K.; Ahmed, E.A.; Panigrahi, P.K.; Farouk, A. Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 7564–7573. [CrossRef]
20. Sarkar, A.; Al-Ars, Z.; Bertels, K. QuASer: Quantum Accelerated de novo DNA sequence reconstruction. *PLoS ONE* **2021**, *16*, e0249850. [CrossRef] [PubMed]
21. Fingerhuth, M.; Babej, T.; Ing, C. A quantum alternating operator ansatz with hard and soft constraints for lattice protein folding. *arXiv* **2018**, arXiv:1810.13411.
22. Khumalo, M.T.; Chieza, H.A.; Prag, K.; Woolway, M. An investigation of IBM Quantum Computing device performance on Combinatorial Optimisation Problems. *Neural Comput. Appl.* **2022**, 1–16. [CrossRef]
23. Hadfield, S.; Wang, Z.; O’Gorman, B.; Rieffel, E.G.; Venturelli, D.; Biswas, R. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms* **2019**, *12*, 34. [CrossRef]
24. Hadfield, S.; Wang, Z.; Rieffel, E.G.; O’Gorman, B.; Venturelli, D.; Biswas, R. Quantum Approximate Optimization with Hard and Soft Constraints. In Proceedings of the Second International Workshop on Post Moores Era Supercomputing, Denver, CO, USA, 12–17 November 2017.
25. Streif, M.; Leib, M. Comparison of QAOA with quantum and simulated annealing. *arXiv* **2019**, arXiv:1901.01903.
26. Martoňák, R.; Santoro, G.E.; Tosatti, E. Quantum annealing of the traveling-salesman problem. *Phys. Rev. E* **2004**, *70*, 057701. [CrossRef] [PubMed]
27. Jiang, Z.; Rieffel, E.G.; Wang, Z. Near-optimal quantum circuit for Grover’s unstructured search using a transverse field. *Phys. Rev. A* **2017**, *95*, 062317. [CrossRef]
28. Skolik, A.; McClean, J.R.; Mohseni, M.; van der Smagt, P.; Leib, M. Layerwise learning for quantum neural networks. *Quantum Mach. Intell.* **2021**, *3*, 5. [CrossRef]
29. ANIS, M.S.; Abraham, H.; AduOffei; Agarwal, R.; Agliardi, G.; Aharoni, M.; Akhalwaya, I.Y.; Aleksandrowicz, G.; Alexander, T.; Amy, M.; et al. Qiskit: An Open-source Framework for Quantum Computing, 2021. Available online: <https://zenodo.org/record/8190968> (accessed on 18 August 2023).
30. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2014**, *2*, 5. [CrossRef]
31. Miller, C.E.; Tucker, A.W.; Zemlin, R.A. Integer Programming Formulation of Traveling Salesman Problems. *J. ACM* **1960**, *7*, 326–329. [CrossRef]
32. Gonzalez-Bermejo, S.; Alonso-Linaje, G.; Atchade-Adelomou, P. GPS: A New TSP Formulation for Its Generalizations Type QUBO. *Mathematics* **2022**, *10*, 416. [CrossRef]
33. Zhu, J.; Gao, Y.; Wang, H.; Li, T.; Wu, H. A Realizable GAS-Based Quantum Algorithm for Traveling Salesman Problem. *arXiv* **2022**, arXiv:2212.02735.
34. Glos, A.; Krawiec, A.; Zimborás, Z. Space-efficient binary optimization for variational computing. *arXiv* **2020**, arXiv:2009.07309.
35. Bakó, B.; Glos, A.; Salehi, O.; Zimborás, Z. Near-Optimal Circuit Design for Variational Quantum Optimization. *arXiv* **2022**, arXiv:2209.03386.
36. Albash, T.; Lidar, D.A. Adiabatic quantum computation. *Rev. Mod. Phys.* **2018**, *90*, 015002. [CrossRef]
37. Blekos, K.; Brand, D.; Ceschini, A.; Chou, C.H.; Li, R.H.; Pandya, K.; Summer, A. A Review on Quantum Approximate Optimization Algorithm and Its Variants. *arXiv* **2023**, arXiv:2306.09198.
38. Herrman, R.; Lotshaw, P.C.; Ostrowski, J.; Humble, T.S.; Siopsis, G. Multi-angle quantum approximate optimization algorithm. *Sci. Rep.* **2022**, *12*, 1–10.
39. Chandarana, P.; Hegade, N.N.; Paul, K.; Albarrán-Arriagada, F.; Solano, E.; del Campo, A.; Chen, X. Digitized-counterdiabatic quantum approximate optimization algorithm. *Phys. Rev. Res.* **2022**, *4*, 013141. [CrossRef]
40. Wurtz, J.; Love, P.J. Counterdiabaticity and the quantum approximate optimization algorithm. *Quantum* **2022**, *6*, 635. [CrossRef]
41. Zhu, L.; Tang, H.L.; Barron, G.S.; Calderon-Vargas, F.A.; Mayhall, N.J.; Barnes, E.; Economou, S.E. Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. *Phys. Rev. Res.* **2022**, *4*, 033029. [CrossRef]
42. Hadfield, S. On the Representation of Boolean and Real Functions as Hamiltonians for Quantum Computing. *ACM Trans. Quantum Comput.* **2021**, *2*, 1–21. [CrossRef]
43. Cruz, D.; Fournier, R.; Gremion, F.; Jeannerot, A.; Komagata, K.; Tomic, T.; Thiesbrummel, J.; Chan, C.L.; Macris, N.; Dupertuis, M.A.; et al. Efficient quantum algorithms for GHZ and W states, and implementation on the IBM quantum computer. *Adv. Quantum Technol.* **2019**, *2*, 1900015. [CrossRef]
44. Diker, F. Deterministic construction of arbitrary W states with quadratically increasing number of two-qubit gates. *arXiv* **2016**, arXiv:1606.09290.
45. Cervera-Lierta, A. Exact Ising model simulation on a quantum computer. *Quantum* **2018**, *2*, 114. [CrossRef]

46. LaRose, R.; Rieffel, E.; Venturelli, D. Mixer-phaser Ansätze for quantum optimization with hard constraints. *Quantum Mach. Intell.* **2022**, *4*, 17. [CrossRef]
47. Bärtschi, A.; Eidenbenz, S. Grover Mixers for QAOA: Shifting Complexity from Mixer Design to State Preparation. In Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, 12–16 October 2020; pp. 72–82. [CrossRef]
48. Wang, Z.; Rubin, N.C.; Dominy, J.M.; Rieffel, E.G. XY mixers: Analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A* **2020**, *101*, 012320. [CrossRef]
49. Borgsten, C. Quantum Approximate Optimization Using SWAP Gates for Mixing. Ph.D Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2021.
50. Powell, M.J.D. Direct search algorithms for optimization calculations. *Acta Numer.* **1998**, *7*, 287–336. [CrossRef]
51. Powell, M. A View of Algorithms for Optimization without Derivatives. *Math. Today* **2007**, *43*, 1–12.
52. Powell, M.J.D. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In *Advances in Optimization and Numerical Analysis*; Gomez, S., Hennart, J.P., Eds.; Springer: Dordrecht, The Netherlands, 1994; pp. 51–67. [CrossRef]
53. Spall, J. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control.* **1992**, *37*, 332–341. [CrossRef]
54. Spall, J. Accelerated second-order stochastic optimization using only function measurements. In Proceedings of the 36th IEEE Conference on Decision and Control, San Diego, CA, USA, 12 December 1997; Volume 2, pp. 1417–1424. [CrossRef]
55. Rakyta, P.; Zimborás, Z. Approaching the theoretical limit in quantum gate decomposition. *Quantum* **2022**, *6*, 710. [CrossRef]
56. Campos, E.; Rabinovich, D.; Akshay, V.; Biamonte, J. Training saturation in layerwise quantum approximate optimization. *Phys. Rev. A* **2021**, *104*, L030401. [CrossRef]
57. Atsushi, M.; Yudai, S.; Shigeru, Y. Problem-specific Parameterized Quantum Circuits of the VQE Algorithm for Optimization Problems. *arXiv* **2020**, arXiv:2006.05643.
58. Basili, R.; Qian, W.; Tang, S.; Castellino, A.; Eshaghian-Wilner, M.; Khokhar, A.; Luecke, G.; Vary, J.P. Performance Evaluations of Noisy Approximate Quantum Fourier Arithmetic. In Proceedings of the 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lyon, France, 30 May 2022–3 June 2022; pp. 435–444. [CrossRef]
59. Abrams, D.; Didier, N.; Johnson, B.; Silva, M.; Ryan, C. Implementation of XY entangling gates with a single calibrated pulse. *Nat. Electron.* **2020**, *3*, 744–750. [CrossRef]
60. Joanes, D.N.; Gill, C.A. Comparing measures of sample skewness and kurtosis. *J. R. Stat. Soc. Ser. D (Stat.)* **1998**, *47*, 183–189. [CrossRef]
61. Kokoska, S.; Zwillinger, D. *CRC Standard Probability and Statistics Tables and Formulae*; CRC Press: Boca Raton, FL, USA, 2000.
62. Kübler, J.M.; Arrasmith, A.; Cincio, L.; Coles, P.J. An Adaptive Optimizer for Measurement-Frugal Variational Algorithms. *Quantum* **2020**, *4*, 263. [CrossRef]
63. Qian, W.; Basili, R.; Eshaghian-Wilner, M.; Khokhar, A.; Luecke, G.; Vary, J.P. Comparative study on the variations of quantum approximate optimization algorithms to the Traveling Salesman Problem. In Proceedings of the 2023 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPS), St. Petersburg, FL, USA, 15–19 May 2023; IEEE Computer Society: Washington, DC, USA, 2023; pp. 541–551. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Asymmetric Measurement-Device-Independent Quantum Key Distribution through Advantage Distillation

Kailu Zhang ^{1,2,3,†}, Jingyang Liu ^{1,2,3,†}, Huajian Ding ^{1,2,3}, Xingyu Zhou ^{1,2,3}, Chunhui Zhang ^{1,2,3}
and Qin Wang ^{1,2,3,*}

- ¹ Institute of Quantum Information and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 1221014031@njupt.edu.cn (K.Z.); 2020010107@njupt.edu.cn (J.L.); 2019010103@njupt.edu.cn (H.D.); xyz@njupt.edu.cn (X.Z.); chz@njupt.edu.cn (C.Z.)
- ² “Broadband Wireless Communication and Sensor Network Technology” Key Lab of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
- ³ “Telecommunication and Networks” National Engineering Research Center, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
- * Correspondence: qinw@njupt.edu.cn
- † These authors contributed equally to this work.

Abstract: Measurement-device-independent quantum key distribution (MDI-QKD) completely closes the security loopholes caused by the imperfection of devices at the detection terminal. Commonly, a symmetric MDI-QKD model is widely used in simulations and experiments. This scenario is far from a real quantum network, where the losses of channels connecting each user are quite different. To adapt such a feature, an asymmetric MDI-QKD model is proposed. How to improve the performance of asymmetric MDI-QKD also becomes an important research direction. In this work, an advantage distillation (AD) method is applied to further improve the performance of asymmetric MDI-QKD without changing the original system structure. Simulation results show that the AD method can improve the secret key rate and transmission distance, especially in the highly asymmetric cases. Therefore, this scheme will greatly promote the development of future MDI-QKD networks.

Keywords: quantum key distribution; asymmetric MDI-QKD; advantage distillation technology

Citation: Zhang, K.; Liu, J.; Ding, H.; Zhou, X.; Zhang, C.; Wang, Q.

Asymmetric Measurement-Device-Independent Quantum Key Distribution through Advantage Distillation. *Entropy* **2023**, *25*, 1174. <https://doi.org/10.3390/e25081174>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 18 July 2023

Revised: 28 July 2023

Accepted: 3 August 2023

Published: 7 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Quantum key distribution (QKD) can unconditionally ensure the theoretical security of information transmission between two or more distant users with quantum mechanics. In the process of development from theory to practice, there are many challenges to realizing remote and secure quantum key distribution in the practical applications. With various theoretical ideas and experimental schemes being put forward, many challenges have been overcome. The BB84 protocol [1] proposed by Bennett realizes two-point communication and the Ekert91 and BBM92 protocols have been proposed successively [2,3]. Although QKD has been proven to have unconditional security in theory, imperfect devices can lead to some security loopholes that hinder the development of QKD protocols in practical applications. In practical applications, we often use weak coherent sources (WCSs) with multi-photon components, and Eve can eavesdrop with photon-number splitting (PNS) attacks [4]. Fortunately, the decoy-state method proposed [5,6] can solve PNS attacks and obtain rapid development both theoretically and experimentally [7–9]. Considering the imperfection of the detector, Lo firstly proposed the MDI-QKD protocol [10] which thoroughly solves the security loopholes mainly at the detection terminal. With the advantages of the MDI-QKD protocol, the MDI-QKD protocol attracts extensive attention and has been greatly studied in theory and experiments [11–18].

In previous work, the MDI-QKD was mainly studied in symmetric scenarios for simplicity. With the development of theory and technology, researchers have paid more

attention to the asymmetric MDI-QKD protocol in recent years. To achieve good interference at the detection terminal, Lo proposed an asymmetric seven-intensity MDI-QKD [19], which can improve the performance of MDI-QKD in practical asymmetric structures based on the four-intensity MDI-QKD [11]. Consequently, asymmetric MDI-QKD is more suitable for the common QKD networks. However, due to its asymmetric nature, its performance is inferior to that of the original symmetric scheme. Improving the performance of asymmetric MDI-QKDs has become an urgent problem that needs to be addressed.

Inspired by the advantage distillation (AD) method [20–23], we study the principle of the method and find that the AD method can be successfully applied to the asymmetric seven-intensity MDI-QKD protocol. Compared with the original protocol, the performance of the asymmetric protocol has been significantly improved, which provides another theoretical verification that the post-processing AD method can improve the performance of the QKD protocol. This method can divide the original key string into blocks of only a few bits to achieve a high key correlation and greatly improve the protocol’s performance. The paper is organized as follows: In Section 2, we review the asymmetric seven-intensity MDI-QKD protocol and introduce the protocol with AD. The results of numerical simulations are shown in Section 3. Finally, summaries are given in Section 4.

2. Methods

2.1. Asymmetric MDI-QKD

Here, we mainly describe the process of the asymmetric seven-intensity MDI-QKD protocol, which develops from the four-intensity symmetric protocol, as follows:

- (1). **State preparation.** Alice (Bob) randomly prepares the signal state only in Z basis with s_A (s_B), and prepares the decoy states only in X basis with intensities of w_A, v_A (w_B, v_B), satisfying the formula $w_A < v_A$ ($w_B < v_B$). When preparing the vacuum state of intensity o , Alice (Bob) does not choose any base. The prepared states will be sent to Charlie to perform measurement;
- (2). **Measurement.** Charlie performs the Bell state measurement (BSM) after receiving the quantum states sent from Alice and Bob;
- (3). **Announcement.** After Alice and Bob repeat the above steps and enough counting events are recorded, Charlie publicly announces the BSM results. Subsequently, they announce the selected bases and intensities;
- (4). **Parameter estimation.** After finishing the quantum transmission phase, Alice and Bob can estimate the lower bound of single-photon yield $Y_{11}^{Z,L}$ and the upper bound of single-photon error rate (QBER) $e_{11}^{X,U}$ using the decoy-state technology;
- (5). **Post-processing.** Alice and Bob perform key reconciliation and privacy amplification on the raw key data to obtain the final secret key.

The decoupled bases are used in the asymmetric seven-intensity MDI-QKD protocol, thus the protocol can perform decoy states in the X basis only to estimate $Y_{11}^{X,L}$ and can use $Y_{11}^{Z,L} = Y_{11}^{X,L}$ to obtain the single-photon yield in Z basis [11]. Then, the secret key rate can be calculated by the following formula [10,11,19]:

$$R = P_{s_A} P_{s_B} \left\{ (s_A e^{-s_A})(s_B e^{-s_B}) Y_{11}^{Z,L} [1 - h(e_{11}^{X,U})] - f_e Q_{s_A s_B}^Z h(E_{s_A s_B}^Z) \right\}, \quad (1)$$

where P_{s_A} and P_{s_B} each correspond to the probability that Alice or Bob emits the signal states of s_A or s_B , respectively. $Q_{s_A s_B}^Z$ and $E_{s_A s_B}^Z$ are the gain and QBER in the Z basis, $Y_{11}^{X,L}(e_{11}^{X,U})$ is the lower (upper) bound of single-photon yield (QBER), which can be estimated from the decoy-state technology, $h(x)$ is the binary entropy function, and f_e is the error correction efficiency.

Based on the asymmetric seven-intensity MDI-QKD protocol above, the performance can be further improved by optimization techniques such as joint estimations and collective constraints [11]. Referring to the joint estimations method, the common part \mathbb{H} is extracted from the following two parameters $Y_{11}^{X,L}, e_{11}^{X,U}$ to optimize the key rate. $e_{11}^{Z,U}$

is used in the following subsection. $Y_{11}^{X,L}$ is a piecewise function where $P_{v_A}^1 P_{w_A}^2 P_{w_B}^1 P_{v_B}^2 < P_{w_A}^1 P_{v_A}^2 P_{v_B}^1 P_{w_B}^2$ [19,24]. These parameters $Y_{11}^{X,L}$, $e_{11}^{X,U}$ and $e_{11}^{Z,U}$ can be estimated accurately by the decoy-state technology in the original MDI-QKD protocol [10,11]. The following formulas can estimate these parameters which lead to a much higher rate in distilling the secure final key:

$$Y_{11}^{X,L} = Y_{11}^{X,e} = \frac{P_{v_A}^1 P_{v_B}^2 Q_{w_A w_B} + P_{w_A}^1 P_{w_B}^2 P_{v_A}^0 Q_{ow_B} + P_{w_A}^1 P_{w_B}^2 P_{v_B}^0 Q_{v_A o}}{P_{w_A}^1 P_{v_A}^1 (P_{w_B}^1 P_{v_B}^2 - P_{w_B}^2 P_{v_B}^1)} - \frac{P_{w_A}^1 P_{w_B}^2 Q_{v_A v_B} + P_{w_A}^1 P_{w_B}^2 P_{v_A}^0 P_{v_B}^0 Q_{oo}}{P_{w_A}^1 P_{v_A}^1 (P_{w_B}^1 P_{v_B}^2 - P_{w_B}^2 P_{v_B}^1)} - \frac{P_{v_A}^1 P_{v_B}^2 \mathbb{H}}{P_{w_A}^1 P_{v_A}^1 (P_{w_B}^1 P_{v_B}^2 - P_{w_B}^2 P_{v_B}^1)}, \tag{2}$$

$$Y_{11}^{X,L} = Y_{11}^{X,f} = \frac{P_{v_B}^1 P_{v_A}^2 Q_{w_A w_B} + P_{w_B}^1 P_{w_A}^2 P_{v_A}^0 Q_{ow_B} + P_{w_B}^1 P_{w_A}^2 P_{v_B}^0 Q_{v_A o}}{P_{w_B}^1 P_{v_B}^1 (P_{w_A}^1 P_{v_A}^2 - P_{w_A}^2 P_{v_A}^1)} - \frac{P_{w_B}^1 P_{w_A}^2 Q_{v_A v_B} + P_{w_B}^1 P_{w_A}^2 P_{v_A}^0 P_{v_B}^0 Q_{oo}}{P_{w_B}^1 P_{v_B}^1 (P_{w_A}^1 P_{v_A}^2 - P_{w_A}^2 P_{v_A}^1)} - \frac{P_{v_B}^1 P_{v_A}^2 \mathbb{H}}{P_{w_B}^1 P_{v_B}^1 (P_{w_A}^1 P_{v_A}^2 - P_{w_A}^2 P_{v_A}^1)}, \tag{3}$$

$$e_{11}^{X,U} = \frac{T_{w_A w_B} (1 + \gamma \sqrt{1/(N_{xw_A w_B} T_{w_A w_B})}) - \mathbb{H}/2}{P_{w_A}^1 P_{w_B}^1 Y_{11}^{X,L}}, \tag{4}$$

$$e_{11}^{Z,U} = \frac{T_{s_A s_B} + P_{s_A}^0 P_{s_B}^0 T_{oo} - [P_{s_A}^0 T_{os_B} + P_{s_B}^0 T_{s_A o}]}{P_{s_A}^1 P_{s_B}^1 Y_{11}^{Z,L}}, \tag{5}$$

$$\mathbb{H} = P_{w_A}^0 Q_{ow_B} + P_{w_B}^0 Q_{w_A o} - P_{w_A}^0 P_{w_B}^0 Q_{oo}, \tag{6}$$

where $P_{l_A}^n (P_{l_B}^m)$ denotes the photon-number distribution of the source at Alice’s (Bob’s) side, $Q_{l_A l_B}$ and $T_{l_A l_B}$ are the gain and the total quantum bit errors [25], and \mathbb{H} is the combination of the gain of the decoy state and the vacuum state. γ is the standard error, and its value is set to 5.3 here. The expression for $Y_{11}^{X,L}$ is equal to $Y_{11}^{X,e}$ when $P_{v_A}^1 P_{w_A}^2 P_{w_B}^1 P_{v_B}^2 < P_{w_A}^1 P_{v_A}^2 P_{v_B}^1 P_{w_B}^2$, otherwise the expression equals $Y_{11}^{X,f}$ [19,24]. Considering the effect of statistical fluctuations on multiple observations, the method of collective constraints can provide tighter constraint conditions between different sources ($s_A, w_A, v_A, s_B, w_B, v_B, o$) than independent bounds. Thus, these parameters $Y_{11}^{X,L}$, $e_{11}^{X,U}$, $e_{11}^{Z,U}$, \mathbb{H} can be further optimized to achieve a higher key rate by the joint constraints method [8].

By the above formulas, we can calculate the final secret key rate of the asymmetric seven-intensity MDI-QKD protocol.

2.2. Asymmetric MDI-QKD with AD

Many previous works have demonstrated that the AD method can further improve the performance of QKD [20–23]. In this section, we improve the secure key rate and transmission distance of the asymmetric seven-intensity MDI-QKD protocol with the AD method. An additional AD method is performed between parameter estimation and post-processing step, and highly correlated bit pairs are discriminated from weakly correlated information. The security of AD method will be analyzed in an entanglement-based scheme. Alice prepares the quantum state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and sends the second particle to Bob through the quantum channel. Since Eve controls the quantum channel by certain value $\lambda_i (i = 0, 1, 2, 3)$, the quantum state shared between Alice and Bob after transmission can be expressed by the following formula:

$$\sigma_{AB} = \lambda_0 |\phi_0\rangle \langle \phi_0| + \lambda_1 |\phi_1\rangle \langle \phi_1| + \lambda_2 |\phi_2\rangle \langle \phi_2| + \lambda_3 |\phi_3\rangle \langle \phi_3|, \tag{7}$$

$$\begin{aligned}
 |\phi_0\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \\
 |\phi_1\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \\
 |\phi_2\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \\
 |\phi_3\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle),
 \end{aligned}
 \tag{8}$$

and $\lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 = 1$. For the quantum state σ_{AB} , the bit error rate of Alice and Bob’s measurements on different bases can be expressed as $\lambda_1 + \lambda_3 = e_1^x$ (four-state or six-state protocol), $\lambda_2 + \lambda_3 = e_1^z$ (four-state or six-state protocol), and $\lambda_1 + \lambda_2 = e_1^y$ (six-state protocol). Eve can steal information and reduce the key rate by choosing the certain value λ_i and the secret key rate can be given by [20]:

$$\begin{aligned}
 R &\geq \min_{\lambda_0, \lambda_1, \lambda_2, \lambda_3} [H(X|E) - H(X|Y)] \\
 &= \min_{\lambda_0, \lambda_1, \lambda_2, \lambda_3} [1 - (\lambda_0 + \lambda_1)h\left(\frac{\lambda_0}{\lambda_0 + \lambda_1}\right) - (\lambda_2 + \lambda_3)h\left(\frac{\lambda_2}{\lambda_2 + \lambda_3}\right) - h(\lambda_0 + \lambda_1)].
 \end{aligned}
 \tag{9}$$

In the AD method, Alice and Bob divide their own raw bits into blocks (x_1, \dots, x_b) and (y_1, \dots, y_b) of size b . Then, choosing a random binary value c , Alice sends $(x_1 \oplus c, \dots, x_b \oplus c)$ to Bob. Bob compares this bitstring with their bitstring (y_1, \dots, y_b) and accepts the security of information only if the results are either all zeros or all ones in the block. In the two cases accepted, Alice (Bob) saves the first bit x_1 (y_1) of the initial string as the raw key. Thus, AD can discern highly correlated bitstring from weakly correlated information as the final raw key. Obviously, the successful probability of the AD method on a certain block of size b can be calculated by:

$$P_{succ} = (\lambda_0 + \lambda_1)^b + (\lambda_2 + \lambda_3)^b.
 \tag{10}$$

After performing the AD step, the practical QBER value $\lambda_2 + \lambda_3$ in the Z basis can be replaced by $\frac{(\lambda_2 + \lambda_3)}{P_{succ}}$, and the practical QBER in the X basis also can be recalculated. The quantum state shared between Alice and Bob can be replaced by:

$$\sigma_{AB} = \bar{\lambda}_0|\phi_0\rangle\langle\phi_0| + \bar{\lambda}_1|\phi_1\rangle\langle\phi_1| + \bar{\lambda}_2|\phi_2\rangle\langle\phi_2| + \bar{\lambda}_3|\phi_3\rangle\langle\phi_3|,
 \tag{11}$$

$$\begin{aligned}
 \bar{\lambda}_0 &= \frac{(\lambda_0 + \lambda_1)^b + (\lambda_0 - \lambda_1)^b}{2P_{succ}}, \\
 \bar{\lambda}_1 &= \frac{(\lambda_0 + \lambda_1)^b - (\lambda_0 - \lambda_1)^b}{2P_{succ}}, \\
 \bar{\lambda}_2 &= \frac{(\lambda_2 + \lambda_3)^b + (\lambda_2 - \lambda_3)^b}{2P_{succ}}, \\
 \bar{\lambda}_3 &= \frac{(\lambda_2 + \lambda_3)^b - (\lambda_2 - \lambda_3)^b}{2P_{succ}}.
 \end{aligned}
 \tag{12}$$

The QKD protocol enhanced by the AD method can achieve the secret key at rate [20]:

$$\begin{aligned}
 R &\geq \max_b \frac{1}{b} P_{succ} \min_{\bar{\lambda}_0, \bar{\lambda}_1, \bar{\lambda}_2, \bar{\lambda}_3} [1 - (\bar{\lambda}_0 + \bar{\lambda}_1)h\left(\frac{\bar{\lambda}_0}{\bar{\lambda}_0 + \bar{\lambda}_1}\right) - (\bar{\lambda}_2 + \bar{\lambda}_3)h\left(\frac{\bar{\lambda}_2}{\bar{\lambda}_2 + \bar{\lambda}_3}\right) \\
 &\quad - h(\bar{\lambda}_0 + \bar{\lambda}_1)].
 \end{aligned}
 \tag{13}$$

Based on the previous analysis, the AD method can be combined with the QKD protocol. It has been widely used in other protocols in previous works. Similarly, the

AD method can be applied to further optimize the properties of quantum channels in the asymmetric MDI-QKD. When the AD method is combined with the asymmetric seven-intensity MDI-QKD protocol, the secret key rate can be estimated by the following formula:

$$R \geq P_{s_A} P_{s_B} \frac{1}{b} q_{succ} Q_{s_A s_B}^Z \left\{ \left(\frac{P_{11} Y_{11}^{Z,L}}{Q_{s_A s_B}^Z} \right)^b [1 - (\bar{\lambda}_0 + \bar{\lambda}_1) h\left(\frac{\bar{\lambda}_0}{\bar{\lambda}_0 + \bar{\lambda}_1}\right) - (\bar{\lambda}_2 + \bar{\lambda}_3) h\left(\frac{\bar{\lambda}_2}{\bar{\lambda}_2 + \bar{\lambda}_3}\right)] - f_e h(\bar{E}_{s_A s_B}^Z) \right\}, \tag{14}$$

$$P_{11} = s_A e^{-s_A} s_B e^{-s_B}, \tag{15}$$

$$q_{succ} = (E_{s_A s_B}^Z)^b + (1 - E_{s_A s_B}^Z)^b, \tag{16}$$

$$\bar{E}_{s_A s_B}^Z = \frac{(E_{s_A s_B}^Z)^b}{(E_{s_A s_B}^Z)^b + (1 - (E_{s_A s_B}^Z))^b}, \tag{17}$$

where P_{11} is the probability of both Alice and Bob’s signal states emitting single-photon events, q_{succ} is the successful probability of the AD method, $\bar{E}_{s_A s_B}^Z$ is the error rate value after the AD method, and e_{11}^x and e_{11}^z are the single-photon error rate in the X and Z bases, respectively. Note that $Y_{11}^{X,L}$, e_{11}^x , and e_{11}^z can be estimated with the decoy-state method.

3. Results

In this work, we explore the combination of a QKD and a post-processing method. We adopt the asymmetric seven-intensity MDI-QKD protocol and the AD method, which can improve the performance of asymmetric MDI-QKD protocol greatly. In this section, numerical simulations of the asymmetric seven-intensity MDI-QKD protocol with AD method are given and the simulation parameters are shown in Table 1. After analyzing the simulation results, we obtained the following significant research results.

Table 1. The basic system parameters used in our numerical simulations. η_D and Y_0 are the efficiency and dark count rate of detectors at Charlie’s side; e_d : the misalignment error of the QKD system; f_e : the error correction efficiency; N : the number of pulse pairs Alice and Bob send.

e_d	η_D	Y_0	f_e	N
0.5%	65%	8×10^{-7}	1.16	10^{11}

We analyze the secret key rate of the asymmetric MDI-QKD protocol with and without the AD method, and the corresponding comparison results are shown in Figure 1 under different conditions $L_{asy} = 0$ dB, 12 dB, 24 dB. The figure shows that the key rate with and without AD are consistent within a short distance. However, for example, the red line with $L_{asy} = 12$ dB, the AD method has a clear advantage at a transmission loss of about 33 dB, and a final transmission loss reaching 39 dB as well as the secret key rate showing a clear improvement. For a more obvious exploration of the reason, we present Figure 2 with respect to b . We can observe that, in the above example, the value of b at about 33 dB has changed from 1 to 2, indicating that the AD method begins to work. With the increase of transmission loss, the AD method requires a larger b value to obtain a tight correlation from weak correlation. Furthermore, the results of the above case are similar to the other two cases ($L_{asy} = 0$ dB, $L_{asy} = 24$ dB). Therefore, the AD method can improve the key generation rate of asymmetric MDI-QKD over a long distance.

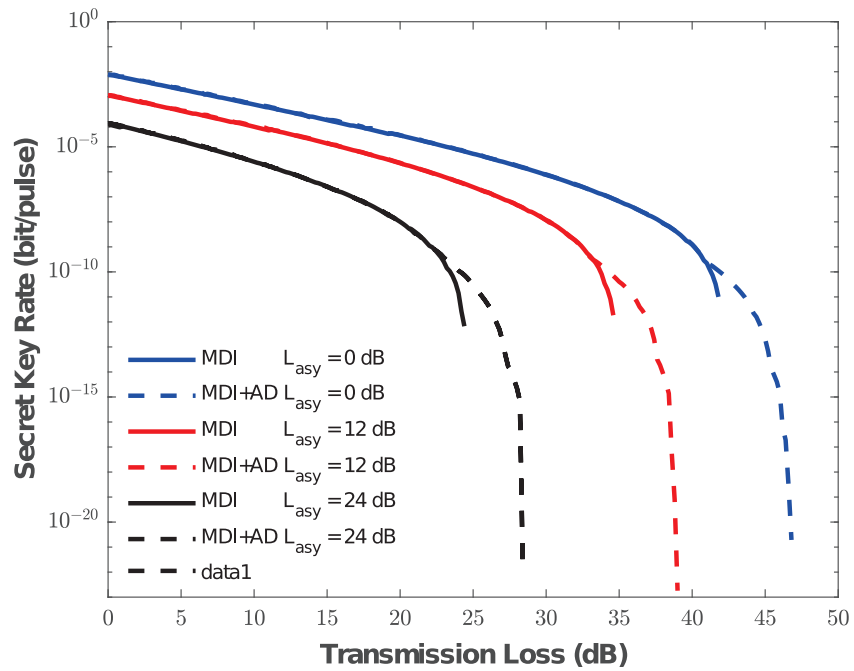


Figure 1. Comparison of the secret key generation rate versus the transmission loss. The value L_{asy} is the loss difference of Alice to Charlie and Bob to Charlie. The different colors represent loss difference, which is $L_{asy} = 0$ dB, $L_{asy} = 12$ dB, and $L_{asy} = 24$ dB, respectively. The solid line represents the secret key without the AD method, and the dotted line represents the secret key with the AD method.

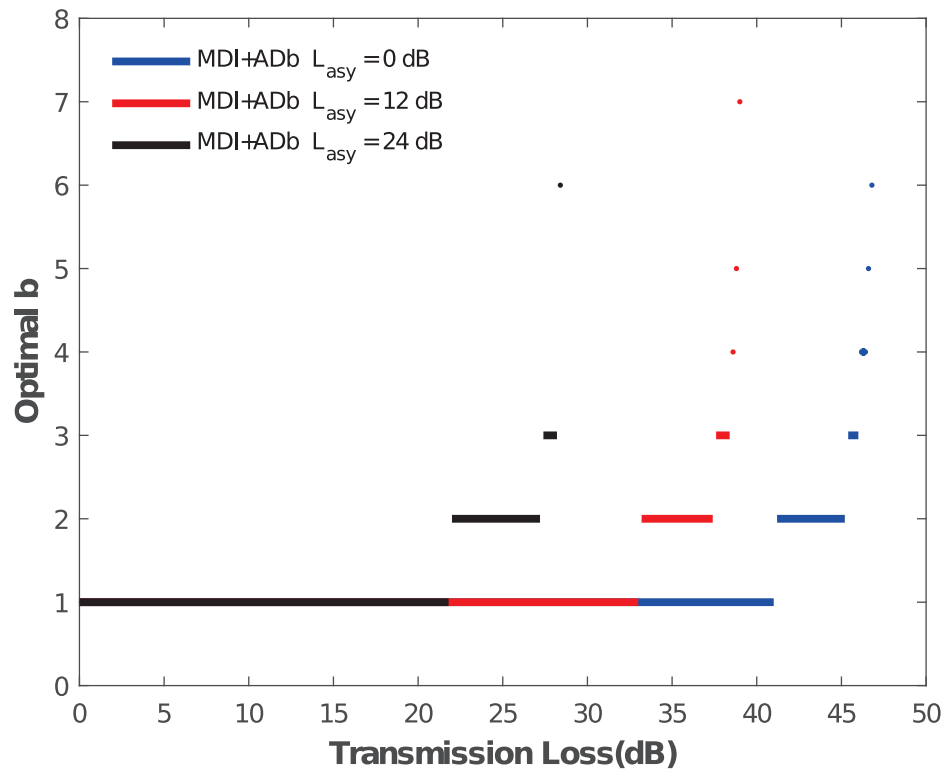


Figure 2. Results of the optimal b versus the transmission loss. The black, red, and blue represent the values $L_{asy} = 0$ dB, $L_{asy} = 12$ dB, and $L_{asy} = 24$ dB, respectively. When value b is not equal to 1, the AD method can further improve the secret key rate and transmission distance of the asymmetric MDI-QKD protocol.

Additionally, we also further investigate the specific effects of the AD method on an asymmetric MDI-QKD under various values L_{asy} , and the results are shown in Figure 3. We describe the meaning of Figure 3 and give a detailed definition of the improved percentage. Generally, when the degree of asymmetry is large, the deterioration of the key rate becomes more obvious. However, after the AD method is used, it can be clearly seen in Figure 3 that the improvement effect of the AD method becomes more obvious with the increase of the degree of asymmetry. For example, the improved percentage can reach about 35% when the value $L_{asy} = 35$ dB, which means that AD method can better solve some transmission performance bottlenecks of the entire network.

By the above analysis, the AD method indeed can increase the propagation distance when the number of pulse pairs $N = 10^{11}$. In order to further analyze the finite size effects, we give the simulation results in Figure 4 under different values of N when the value $L_{asy} = 12$ dB. As can be seen from Figure 4, the AD method improves the performance of the asymmetric MDI-QKD protocol under various finite-size effects. Even though there is a large statistical fluctuation when the number of pulse pairs $N = 10^{10}$, the AD method can still tolerate transmission losses of more than 5 dB, which means that AD method can also be more adaptable with finite-size cases.

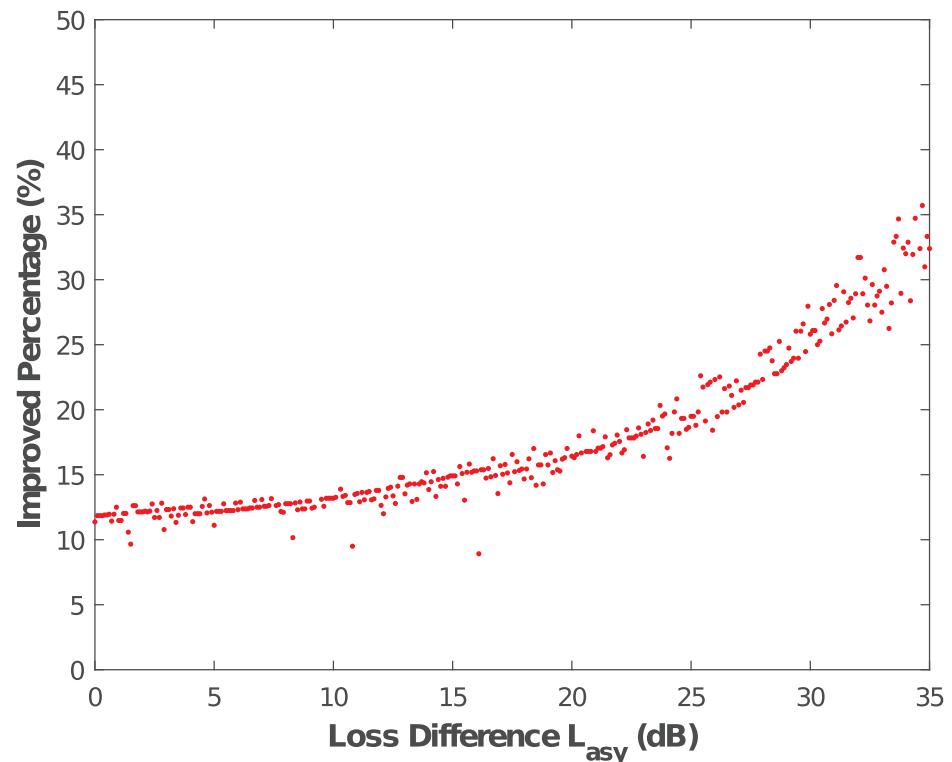


Figure 3. Results of the value L_{asy} versus the improved percentage. The improved transmission loss is the difference of the maximum transmission loss of the asymmetric MDI-QKD with and without the AD method, and we define the improved percentage as the difference divided by the latter. With the increasing degree of asymmetry, the improved percentage also becomes better.

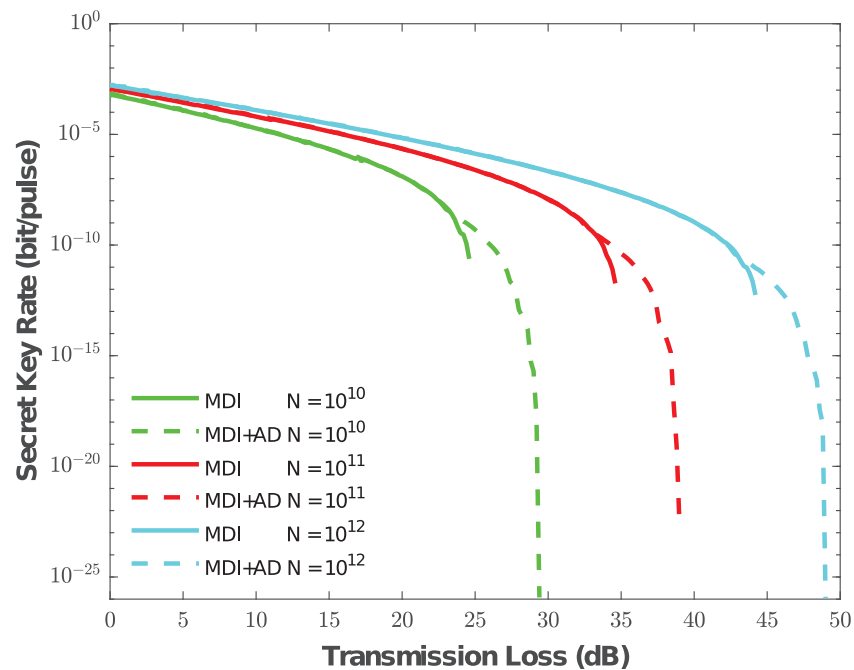


Figure 4. Comparison of the secret key generation rate versus the transmission loss when the value $L_{asy} = 12$ dB. The different colors represent the values $N = 10^{10}$, $N = 10^{11}$, and $N = 10^{12}$, respectively. The solid line represents the secret key without the AD method, and the dotted line represents the secret key with the AD method.

4. Conclusions

The AD method, a classical algorithm based on information theory, can be combined with QKD without changing the existing system structure. Specifically, the AD method can be combined with an asymmetric seven-intensity MDI-QKD to improve the robustness effectively, so as to distinguish and extract highly correlated bit pairs from the weakly correlated information as the final secret key. The AD method has a better performance for the asymmetric MDI-QKD protocol. The greater the degree of asymmetry, the better the improvement of the AD method. The AD method can also improve the performance of the asymmetric MDI-QKD protocol under various finite-size effects, and can be more adaptable with finite-size cases. Our work may play a role in measurement-device-independent networks.

Author Contributions: Conceptualization, J.L. and Q.W.; Methodology, K.Z., J.L., H.D., C.Z. and Q.W.; Software, K.Z., J.L., H.D. and C.Z.; Validation, K.Z., J.L., H.D., X.Z., C.Z. and Q.W.; Formal analysis, K.Z., J.L., X.Z. and C.Z.; Investigation, J.L. and C.Z.; Resources, Q.W.; Data curation, K.Z., J.L., X.Z. and C.Z.; Writing—original draft, K.Z. and X.Z.; Writing—review & editing, J.L., H.D., X.Z. and Q.W.; Visualization, K.Z. and J.L.; Supervision, X.Z. and Q.W.; Project administration, Q.W.; Funding acquisition, X.Z. and Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: We gratefully acknowledge the financial support from the National Natural Science Foundation of China (12074194, 12104240, 62101285); Industrial Prospect and Key Core Technology Projects of Jiangsu provincial key R&D Program (E2022071); Natural Science Foundation of Jiangsu Province (BK20192001, BK20210582); Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX20_0726); Natural Science Foundation of the Jiangsu Higher Education Institutions(21KJB140014), and NUPTSF (NY220122, NY220123).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bennett, C.H.; Brassard, G. Quantum cryptography: Public key distribution and coin tossing. In Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, 9–12 December 1984.
- Ekert, A.K. Quantum cryptography based on Bell’s theorem. *Phys. Rev. Lett.* **1991**, *67*, 661–663. [CrossRef] [PubMed]
- Bennett, C.H.; Brassard, G.; Mermin, N.D. Quantum cryptography without Bell’s theorem. *Phys. Rev. Lett.* **1992**, *68*, 557. [CrossRef] [PubMed]
- Lutkenhaus, N.; Jarma, M. Quantum key distribution with realistic states: Photon-number statistics in the photon-number splitting attack. *New J. Phys.* **2002**, *4*, 44. [CrossRef]
- Lo, H.K.; Ma, X.F.; Chen, K. Decoy State Quantum Key Distribution. *Phys. Rev. Lett.* **2005**, *94*, 230504. [CrossRef]
- Wang, X.B. Beating the photon-number-splitting attack in practical quantum cryptography. *Phys. Rev. Lett.* **2005**, *94*, 230503. [CrossRef]
- Wang, X.B. Three-intensity decoy-state method for device-independent quantum key distribution with basis-dependent errors. *Phys. Rev. A* **2013**, *87*, 12320. [CrossRef]
- Yu, Z.W.; Zhou, Y.H.; Wang, X.B. Statistical fluctuation analysis for measurement-device-independent quantum key distribution with three-intensity decoy-state method. *Phys. Rev. A* **2015**, *91*, 032318. [CrossRef]
- Zhang, C.H.; Zhang, C.M.; Wang, Q. Improving the Performance of Practical Decoy-State Measurement-Device-Independent Quantum Key Distribution with Biased Basis Choice. *Theor. Comput. Phys.* **2018**, *70*, 331. [CrossRef]
- Lo, H.K. Measurement-Device-Independent Quantum Key Distribution. *Phys. Rev. Lett.* **2012**, *108*, 130503. [CrossRef]
- Zhou, Y.H.; Yu, Z.W.; Wang, X.B. Making the decoy-state measurement-device independent quantum key distribution practically useful. *Phys. Rev. A* **2016**, *93*, 042324. [CrossRef]
- Liu, H.; Wang, W.; Wei, K.; Fang, X.T.; Li, L.; Liu, N.L.; Liang, H.; Zhang, S.J.; Zhang, W.; Li, H.; et al. Experimental Demonstration of High-Rate Measurement-Device-Independent Quantum Key Distribution over Asymmetric Channels. *Phys. Rev. Lett.* **2019**, *122*, 160501. [CrossRef]
- Chen, Y.P.; Liu, J.Y.; Sun, M.S.; Zhou, X.Y.; Zhang, C.H.; Li, J.; Wang, Q. Experimental measurement-device-independent quantum key distribution with the double-scanning method. *Opt. Lett.* **2021**, *46*, 3729–3732. [CrossRef]
- Jiang, C.; Yu, Z.W.; Hu, X.L.; Wang, X.B. Higher key rate of measurement-device-independent quantum key distribution through joint data processing. *Phys. Rev. A* **2021**, *103*, 012402. [CrossRef]
- Lu, F.Y.; Wang, Z.H.; Yin, Z.Q.; Wang, S.; Wang, R.; Guo, G.C.; Han, Z.F. Unbalanced-basis-misalignment-tolerant measurement-device-independent quantum key distribution. *Optica* **2022**, *9*, 886–893. [CrossRef]
- Cao, Y.; Li, Y.H.; Yang, K.X.; Jiang, Y.F.; Li, S.L.; Hu, X.L.; Abulizi, M.; Li, C.L.; Zhang, W.J.; Sun, Q.C.; et al. Long-Distance Free-Space Measurement-Device-Independent Quantum Key Distribution. *Phys. Rev. Lett.* **2020**, *125*, 260503. [CrossRef] [PubMed]
- Wei, K.J.; Li, W.; Tan, H.; Li, Y.; Min, H.; Zhang, W.J.; Li, H.; You, L.X.; Wang, Z.; Jiang, X.; et al. High-Speed Measurement-Device-Independent Quantum Key Distribution with Integrated Silicon Photonics. *Phys. Rev. X* **2020**, *10*, 031030. [CrossRef]
- Jie, G.; Yuan, F.; Lu, F.Y.; Wang, S.; Yin, Z.Q.; He, D.Y.; Chen, W.; Zhou, Z.; Wang, H.Z.; Teng, J.; et al. Robust and Adaptable Quantum Key Distribution Network without Trusted Nodes. *Optica* **2022**, *9*, 812.
- Wang, W.Y.; Xu, F.H.; Lo, H.K. Asymmetric Protocols for Scalable High-Rate Measurement-Device-Independent Quantum Key Distribution Networks. *Phys. Rev. X* **2019**, *9*, 041012. [CrossRef]
- Renner, R. Security of quantum key distribution. *Int. J. Quant. Inf.* **2008**, *6*, 1–127. [CrossRef]
- Wang, R.Q.; Zhang, C.M.; Yin, Z.Q.; Li, H.W.; Wang, S.; Chen, W.; Guo, G.C.; Han, Z.F. Phase-matching quantum key distribution with advantage distillation. *New J. Phys.* **2022**, *24*, 73049. [CrossRef]
- Li, H.W.; Zhang, C.M.; Jiang, M.S.; Cai, Q.Y. Improving the performance of practical decoy-state quantum key distribution with advantage distillation technology. *Commun. Phys.* **2022**, *5*, 53. [CrossRef]
- Li, H.W.; Wang, R.Q.; Zhang, C.M.; Cai, Q.Y. Improving the performance of twin-field quantum key distribution with advantage distillation technology. *arXiv* **2022**, arXiv:2202.10059.
- Xu, F.H.; Curty, M.; Qi, B.; Lo, H.K. Practical aspects of measurement-device-independent quantum key distribution. *New J. Phys.* **2013**, *15*, 113007. [CrossRef]
- Wang, Q.; Wang, X.B. Simulating of the measurement-device independent quantum key distribution with phase randomized general sources. *Sci. Rep.* **2014**, *4*, 4612. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

On the Applicability of Quantum Machine Learning

Sebastian RaubitzeK ^{1,2,*} and Kevin Mallinger ^{1,2}¹ Data Science Research Unit, TU Wien, Favoritenstrasse 9-11/194, 1040 Vienna, Austria² SBA Research gGmbH, Floragasse 7/5.OG, 1040 Vienna, Austria

* Correspondence: sebastian.raubitzeK@tuwien.ac.at

Abstract: In this article, we investigate the applicability of quantum machine learning for classification tasks using two quantum classifiers from the Qiskit Python environment: the variational quantum circuit and the quantum kernel estimator (QKE). We provide a first evaluation on the performance of these classifiers when using a hyperparameter search on six widely known and publicly available benchmark datasets and analyze how their performance varies with the number of samples on two artificially generated test classification datasets. As quantum machine learning is based on unitary transformations, this paper explores data structures and application fields that could be particularly suitable for quantum advantages. Hereby, this paper introduces a novel dataset based on concepts from quantum mechanics using the exponential map of a Lie algebra. This dataset will be made publicly available and contributes a novel contribution to the empirical evaluation of quantum supremacy. We further compared the performance of VQC and QKE on six widely applicable datasets to contextualize our results. Our results demonstrate that the VQC and QKE perform better than basic machine learning algorithms, such as advanced linear regression models (Ridge and Lasso). They do not match the accuracy and runtime performance of sophisticated modern boosting classifiers such as XGBoost, LightGBM, or CatBoost. Therefore, we conclude that while quantum machine learning algorithms have the potential to surpass classical machine learning methods in the future, especially when physical quantum infrastructure becomes widely available, they currently lag behind classical approaches. Our investigations also show that classical machine learning approaches have superior performance classifying datasets based on group structures, compared to quantum approaches that particularly use unitary processes. Furthermore, our findings highlight the significant impact of different quantum simulators, feature maps, and quantum circuits on the performance of the employed quantum estimators. This observation emphasizes the need for researchers to provide detailed explanations of their hyperparameter choices for quantum machine learning algorithms, as this aspect is currently overlooked in many studies within the field. To facilitate further research in this area and ensure the transparency of our study, we have made the complete code available in a linked GitHub repository.

Citation: RaubitzeK, S.; Mallinger, K. On the Applicability of Quantum Machine Learning. *Entropy* **2023**, *25*, 992. <https://doi.org/10.3390/e25070992>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 7 June 2023

Revised: 22 June 2023

Accepted: 26 June 2023

Published: 28 June 2023

Keywords: quantum machine learning; variational quantum circuit; quantum kernel estimator; Qiskit; Ridge; Lasso; XGBoost; LightGBM; CatBoost; classification; quantum computing; boost classifiers; neural networks



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Quantum computing has recently gained significant attention due to its potential to solve complex computational problems exponentially faster than classical computers [1]. Quantum machine learning (QML) is an emerging field that combines the power of quantum computing with traditional machine learning techniques to solve real-world problems more efficiently [2,3]. Various QML algorithms have been proposed, such as quantum kernel estimator [4] and variational quantum circuit [5,6], which have shown promising results in diverse applications, including pattern recognition and classification tasks [7–9].

In this study, we aim to compare QKE (quantum kernel estimator) and VQC (variational quantum circuit) with powerful classical machine learning methods such

as XGBoost [10], Ridge [11], Lasso [12], LightGBM [13], CatBoost [14], and MLP (multilayer perceptron) [15] on six benchmark datasets partially available in the Scikit-learn library [16] as well as artificially generated datasets. To ensure a fair comparison on the benchmark datasets, we perform a randomized search to optimize hyperparameters for each algorithm, thereby providing a comprehensive statistical comparison of their performance. Furthermore, we provide the full program code in a GitHub repository [17] to make our results reproducible and boost research that can potentially build on our approach.

Since quantum machines are not readily accessible, we can only compare these algorithms' performance on simulated quantum circuits. Although this approach does not reveal the full potential of quantum machine learning, it does highlight how the discussed quantum machine learning methods handle different levels of complexity inherent in the datasets. For this reason, we also developed a method to generate artificial datasets based on quantum mechanical concepts to provide a prototype for a particularly well-suited dataset for quantum machine learning. This will estimate the possible improvements that quantum machine learning algorithms can offer over classical methods in terms of accuracy and efficiency, considering the computational resources needed to simulate quantum circuits.

In this study, we address and partially answer the following research questions:

1. How do QKE and VQC algorithms compare to classical machine learning methods such as XGBoost, Ridge, Lasso, LightGBM, CatBoost, and MLP regarding accuracy and efficiency on simulated quantum circuits?
2. To what extent can a randomized search to find a suitable set of hyperparameters make the performance of quantum machine learning algorithms comparable to classical approaches?
3. What are the limitations and challenges associated with the current state of quantum machine learning, and how can future research address these challenges to unlock the full potential of quantum computing in machine learning applications?
4. Do quantum machine learning algorithms outperform regular machine learning algorithms on datasets constrained by the rules of quantum mechanics? Thus, do they provide a quantum advantage for datasets that exhibit strong symmetry properties in terms of adhering to Lie algebras?

The research presented in this article is partially inspired by the work of Zeguendry et al. [18], which offers an excellent review and introduction to quantum machine learning. However, their article does not delve into the tuning of hyperparameters for the quantum machine learning models employed, nor does it provide ideas on creating best-suited data for quantum machine learning classification tasks. We aim to expand the toolbox of quantum machine learning, first by discussing the space of Hyperparameters and second by providing a prototype for generating "quantum data". Furthermore, this analysis will help determine the current state of quantum machine learning performance and whether researchers should employ these algorithms in their studies.

We provide the entire program code of our experiments and all the results in a GitHub repository, ensuring the integrity of our findings, fostering research in this field, and offering a comprehensive code for researchers to test quantum machine learning on their classification problems. Thereby, a key contribution of our research is not only the provision of a single implementation of a quantum machine learning algorithm, but also the execution of a randomized search for potential hyperparameters of both classical and quantum machine learning models and a novel approach for generating artificial classification problems based on concepts inherent to quantum mechanics, i.e., Lie groups and algebras.

This article is structured as follows: Section 2 discusses relevant and related work. In Section 3, we describe, reference, and, to some degree, derive all employed techniques. We will not discuss the mathematical details of all employed algorithms here, but rather refer the interested reader to the referenced sources. Section 4 describes our performed experiments in detail, followed by the obtained results in Section 5, which also features a discussion of our findings. Finally, we conclude our findings in Section 6.

2. Related Work

Considerable research was conducted in recent years to advance quantum machine learning environments and their application field. This starts in the data encoding process, in which Schuld and Killoran [3] investigated quantum machine learning in feature Hilbert spaces theoretically. They proposed a framework for constructing quantum embeddings of classical data to enable quantum algorithms that learn and classify data in quantum feature spaces.

Further research was conducted on introducing novel architectural frameworks. For this, Mitarai et al. [19] presented a method called quantum circuit learning (QCL), which uses parameterized quantum circuits to approximate classical functions. QCL can be applied to supervised and unsupervised learning tasks, as well as reinforcement learning.

Havlíček et al. [4] introduced a quantum-enhanced feature space approach using quantum circuits. This work demonstrated that quantum computers can effectively process classical data with quantum kernel methods, offering the potential for exponential speedup in certain applications.

Furthermore, Farhi and Neven [20] explored the use of quantum neural networks for classification tasks on near-term quantum processors. They showed that quantum neural networks can achieve good classification performance with shallow circuits, making them suitable for noisy intermediate-scale quantum (NISQ) devices.

Other research focused on the advancement of applying quantum fundamentals on classical machine learning applications. Hereby, Reberntrost et al. [21] introduced the concept of a quantum support vector machine for big data classification. They showed that the quantum version of the algorithm can offer exponential speedup compared to its classical counterpart, specifically in the kernel evaluation stage.

To advance the application field of quantum machine learning, Liu and Reberntrost [22] proposed a quantum machine learning approach for quantum anomaly detection. They demonstrated that their method can efficiently solve classification problems, even when the data have a high degree of entanglement.

In this regard, it is worth mentioning the work of Broughton et al. [23] introduced TensorFlow Quantum, an open-source library for the rapid prototyping of hybrid quantum-classical models for classical or quantum data. They demonstrated various applications of TensorFlow Quantum, including supervised learning for quantum classification, quantum control, simulating noisy quantum circuits, and quantum approximate optimization. Moreover, they showcased how TensorFlow Quantum can be applied to advanced quantum learning tasks such as meta-learning, layer-wise learning, Hamiltonian learning, sampling thermal states, variational quantum eigensolvers, classification of quantum phase transitions, generative adversarial networks, and reinforcement learning.

In the review paper by Zeguendry et al. [18], the authors present a comprehensive overview of quantum machine learning from the perspective of conventional machine learning techniques. The paper starts by exploring the background of quantum computing, its architecture, and an introduction to quantum algorithms. It then delves into several fundamental algorithms for QML, which form the basis of more complex QML algorithms and can potentially offer performance improvements over classical machine learning algorithms. In the study, the authors implement three machine learning algorithms: quantum neural networks, quantum support vector machines, and variational quantum circuit. They compare the performance of these quantum algorithms with their classical counterparts on various datasets. Specifically, they implement quantum neural networks on a quantum computer to recognize handwritten digits and compare its performance to convolutional neural networks, stating the performance improvements by quantum machine learning.

Despite these advancements, it is important to note that some of the discussed papers may not have used randomized search CV from Scikit-learn to optimize the classical machine learning algorithms, thereby overstating the significance of quantum supremacy. Nevertheless, the above-mentioned works present a comprehensive overview of the state

of the art in quantum machine learning for classification, highlighting the potential benefits of using quantum algorithms in various forms and applications.

3. Methodology

This section presents our methodology for comparing the performance of classical and quantum machine learning techniques for classification tasks. Our approach is designed to provide a blueprint for future experiments in this area of research. We employ the Scikit-learn library, focusing on the inbuilt functions to select a good set of hyperparameters, i.e., `RandomizedSearchCV` to compare classical and quantum machine learning models. We also utilize the Qiskit library to incorporate quantum machine learning techniques into our experiments, [24]. The selected datasets for our study include both real-world and synthetic data, enabling a comprehensive evaluation of the classifiers' performance.

3.1. Supervised Machine Learning

Supervised machine learning is a subfield of artificial intelligence that focuses on developing algorithms and models to learn patterns and make decisions or predictions based on data [25,26]. The main goal of supervised learning is to predict labels or outputs of new, unseen data given a set of known input–output pairs (training data). This section briefly introduces several classical machine learning techniques used for classification tasks, specifically in the context of supervised learning. These techniques serve as a baseline to evaluate the applicability of quantum machine learning approaches, which are the focus of this paper. Furthermore, we will then introduce the employed quantum machine learning algorithms.

One of the essential aspects of supervised machine learning is the ability to predict/classify data. The models are trained using a labeled dataset, and then the performance of the models is evaluated based on their accuracy in predicting the labels of previously unseen test samples [27]. This evaluation is crucial to estimate the model's ability to generalize the learned information when making predictions on new, real-world data.

Various techniques, such as cross-validation and train-test splits, are often used to obtain reliable performance estimates of the models [28]. By comparing the performance of different models, researchers and practitioners can determine which model or algorithm is better suited for a specific problem domain.

3.2. Classical Supervised Machine Learning Techniques

The following list describes the employed algorithms that serve as a baseline for the afterwards described and later tested quantum machine learning algorithms.

- **Lasso and Ridge Regression/Classification:** Lasso (least absolute shrinkage and selection operator) and Ridge Regression are linear regression techniques that incorporate regularization to prevent overfitting and improve model generalization [11,12]. Lasso uses L1 regularization, which tends to produce sparse solutions, while Ridge Regression uses L2 regularization, which prevents coefficients from becoming too large. Both of these regression algorithms can also be used for classification tasks.
- **Multilayer Perceptron:** MLP is a type of feedforward artificial neural network with multiple layers of neurons, including input, hidden, and output layers [15]. MLPs are capable of modeling complex non-linear relationships and can be trained using backpropagation.
- **Support Vector Machines (SVM):** SVMs are supervised learning models used for classification and regression tasks [29]. They work by finding the optimal hyperplane that separates the data into different classes, maximizing the margin between the classes.
- **Gradient Boosting Machines:** Gradient boosting machines are an ensemble learning method that builds a series of weak learners, typically decision trees, to form a strong learner [30]. The weak learners are combined by iteratively adding them to the model while minimizing a loss function. Notable gradient boosting machines for classification tasks include XGBoost [10], CatBoost [14], and LightGBM [13]. These three algorithms

have introduced various improvements and optimizations to the original gradient boosting framework, such as efficient tree learning algorithms, handling categorical features, and reducing memory usage.

3.3. Quantum Machine Learning

Quantum machine learning is an emerging interdisciplinary field that leverages the principles of quantum mechanics and quantum computing to improve or develop novel algorithms for machine learning tasks [2]. This section introduces two key quantum machine learning techniques, Variational Quantum Circuit and Quantum Kernel Estimator, and discusses their connections to classical machine learning techniques. Additionally, we briefly introduce Qiskit Machine Learning, a Python package developed by IBM for implementing quantum machine learning algorithms. Furthermore, we want to mention the work done by [18] for a review of quantum machine learning algorithms and a more detailed discussion of the employed algorithms.

3.3.1. Variational Quantum Circuit (VQC)

VQC is a hybrid quantum-classical algorithm that can be viewed as a quantum analog of classical neural networks, specifically the multilayer perceptron [5,6]. VQC employs a parametrized quantum circuit, which is trained using classical optimization techniques to find the optimal parameters for classification tasks. The learned quantum circuit can then be used to classify new data points.

Figure 1 illustrates the schematic depiction of the variational quantum circuit, which involves preprocessing the data, encoding it onto qubits using a feature map, processing it through a variational quantum circuit (Ansatz), measuring the final qubit states, and optimizing the circuit parameters θ . Thus, the main building blocks of the VQC are as follows:

1. Preprocessing: The data are prepared and preprocessed before being encoded onto qubits.
2. Feature map encoding (yellow in the figure): The preprocessed data are encoded onto qubits using a feature map.
3. Variational quantum circuit (Ansatz) (steel-blue in the figure): The encoded data undergo processing through the variational quantum circuit, also known as the Ansatz, which consists of a series of quantum gates and operations.
4. Measurement (orange in the figure): The final state of the qubits is measured, providing probabilities for the different quantum states.
5. Parameter optimization (Optimizer): The variational quantum circuit is optimized by adjusting the parameters θ , such as the rotations of specific quantum gates, to improve the outcome/classification.

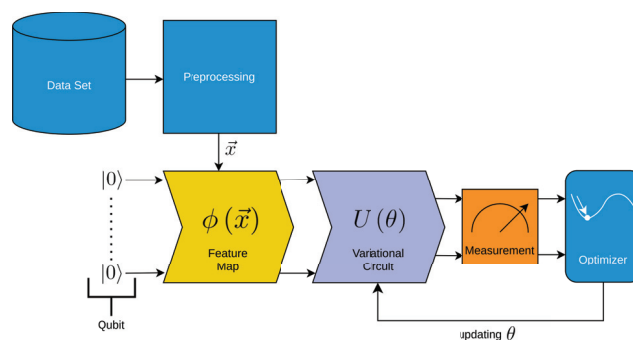


Figure 1. Schematic depiction of the variational quantum circuit. The VQC consists of several steps. We colored the steps that are similar to classical neural networks in light blue and the other steps in yellow, steel-blue, and orange.

3.3.2. Quantum Kernel Estimator

QKE is a technique that leverages the quantum computation of kernel functions to enhance the performance of classical kernel methods, such as support vector machines [4,31]. By computing the kernel matrix using quantum circuits, QKE can capture complex data relationships that may be challenging for classical kernel methods to exploit.

The main building blocks for the employed QKE, which are depicted in Figure 2 are as follows:

1. Data preprocessing: The input data are preprocessed, which may include tasks such as data cleaning, feature scaling, or feature extraction. This step ensures that the data are in an appropriate format for the following quantum feature maps.
2. Feature map encoding (yellow in the figure): The preprocessed data are encoded onto qubits using a feature map.
3. Kernel computation (steel-blue in the figure): Instead of directly computing the kernel matrix from the original data, a kernel function is precomputed using the quantum computing capabilities, meaning that the inner product of two quantum states is estimated on a quantum simulator/circuit. This kernel function captures the similarity between pairs of data points in a high-dimensional feature space.
4. SVM training: The precomputed kernel function is then used as input to the SVM algorithm for model training. The SVM aims to find an optimal hyperplane that separates the data points into different classes with the maximum margin.

Here, we need to mention that in the documentation of Qiskit machine learning, the developers provided a full QKE implementation without the need to use, e.g., Scikit-learn's SVM-implementation. However, as of the writing of this article, this estimator is no longer available in Qiskit machine learning. Thus, one needs to use a support vector machine implementation from other sources after precomputing the kernel on a quantum simulator.

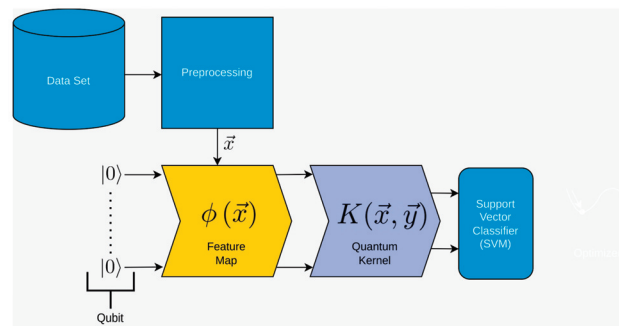


Figure 2. Schematic depiction of the quantum kernel estimator. The QKE consists of several steps. We colored the steps that are similar to classical support vector machines in light blue and the other steps in yellow and steel-blue. The employed QKE algorithm consists of a support vector machine algorithm with precomputed kernel, i.e., a classical machine learning method that leverages the power of quantum computing to efficiently compute the kernel matrix.

3.3.3. Qiskit Machine Learning

Qiskit Machine Learning is an open-source Python package developed by IBM for implementing quantum machine learning algorithms [24]. This package enables researchers and practitioners to develop and test quantum machine learning algorithms, including VQC and QKE, using IBM's quantum computing platform. It provides tools for building and simulating quantum circuits, as well as interfaces to classical optimization and machine learning libraries. Thus, we used this environment and the corresponding quantum simulators described in Appendix A for our experiments.

3.4. Accuracy Score for Classification

The accuracy score is a standard metric used to evaluate the performance of classification algorithms. We employed the accuracy score to evaluate all presented experiments.

It is defined as the ratio of correct predictions to the total number of predictions. The formula for the accuracy score is defined as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (1)$$

In Scikit-learn, the accuracy score can be computed using the `accuracy_score` function from the `'sklearn.metrics'` module [16]. For more information on the accuracy score and its interpretation, refer to the Scikit-learn documentation [16].

3.5. Datasets

In this study, we used six classification datasets from various sources. Two datasets are part of the Scikit-learn library, while the remaining four are obtained/fetched from OpenML. The datasets are described below:

1. **Iris Dataset:** A widely known dataset consisting of 150 samples of iris flowers, each with four features (sepal length, sepal width, petal length, and petal width) and one of three species labels (Iris Setosa, Iris Versicolor, or Iris Virginica). This dataset is included in the Scikit-learn library [16].
2. **Wine Dataset:** A popular dataset for wine classification, which consists of 178 samples of wine, each with 13 features (such as alcohol content, color intensity, and hue) and one of three class labels (class 1, class 2, or class 3). This dataset is also available in the Scikit-learn library [16].
3. **Indian Liver Patient Dataset (LPD):** This dataset contains 583 records, with 416 liver patient records and 167 non-liver patient records [32]. The dataset includes ten variables: age, gender, total bilirubin, direct bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT, and Alkphos. The primary task is to classify patients into liver or non-liver patient groups.
4. **Breast Cancer Coimbra Dataset:** This dataset consists of 10 quantitative predictors and a binary dependent variable, indicating the presence or absence of breast cancer [33,34]. The predictors are anthropometric data and parameters obtainable from routine blood analysis. Accurate prediction models based on these predictors can potentially serve as a biomarker for breast cancer.
5. **Teaching Assistant Evaluation Dataset:** This dataset includes 151 instances of teaching-assistant (TA) assignments from the Statistics Department at the University of Wisconsin-Madison, with evaluations of their teaching performance over three regular semesters and two summer semesters [35,36]. The class variable is divided into three roughly equal-sized categories ("low", "medium", and "high"). There are six attributes, including whether the TA is a native English speaker, the course instructor, the course, the semester type (summer or regular), and the class size.
6. **Impedance Spectrum of Breast Tissue Dataset:** This dataset contains impedance measurements of freshly excised breast tissue at the following frequencies: 15.625, 31.25, 62.5, 125, 250, 500, and 1000 KHz [37,38]. The primary task is to predict the classification of either the original six classes or four classes by merging the fibro-adenoma, mastopathy, and glandular classes whose discrimination is not crucial.

These datasets were selected for their diverse domains and varied classification tasks, providing a robust testing ground for the quantum classifiers we employed in our experiments. Furthermore, we used artificially generated datasets to control the number of samples. Here, Scikit-learn provides a valuable function called `make_classification` to generate synthetic classification datasets. This function creates a random n -class classification problem, initially creating clusters of points normally distributed about vertices of an n -informative-dimensional hypercube, and assigns an equal number of clusters to each class [16]. It introduces interdependence between features and adds further noise to the data. The generated data are highly customizable, with options for specifying the number of samples, features, informative features, redundant features, repeated features, classes, clusters per class, and more. For more details on the `make_classification` function and its

parameters, refer to the Scikit-learn documentation available on scikit-learn.org (accessed on 25 June 2023).

3.5.1. Data Obtained from Lie-Algebras

We construct another artificial dataset final dataset for our final evaluation; however, this time, we do this by using tools from the theory of Lie groups. The reason for employing these concepts is that we want to produce data that resembles the complexity inherent to the Qubit-Vectorspace of quantum machine learning and that, furthermore, is generated by applying transformations on vectors that are similar to the manipulations present in quantum machine learning algorithms, e.g., for the VQC, rotations of/around the Bloch-sphere. Thus, overall, we aim to provide random data for a classification task to show a case where the authors assume quantum machine learning algorithms can, because of their inherent structure, outperform classical machine learning algorithms, and thus, provide a prototype on the type of data specifically tailored to address the inherent structure of quantum machine learning. The theoretical foundations of this section are obtained from [39], and thus, the interested reader is referred to this book for a profound introduction to Lie groups. To further explain the employed ideas, we start by introducing the concept of a Lie group G and the corresponding Lie-algebra \mathfrak{g} .

A Lie group is a mathematical structure that captures the essence of continuous symmetry. Named after the Norwegian mathematician Sophus Lie, Lie groups are ubiquitous in many areas of mathematics and physics, including the study of differential equations, geometry, and quantum mechanics.

A Lie group is a set G that has the structure of both a smooth manifold and a group in such a way that the group operations (multiplication and inversion) are smooth. That is, a Lie group is a group that is also a differentiable manifold, such that the group operations are compatible with the smooth structure.

Thus, a Lie group is a set G equipped with a group structure (i.e., a binary operation $G \times G \rightarrow G, (g, h) \mapsto gh$ that is associative, an identity element $e \in G$, and an inversion operation $G \rightarrow G, g \mapsto g^{-1}$) and a smooth manifold structure such that the following conditions are satisfied:

1. The multiplication map $\mu : G \times G \rightarrow G$ defined by $\mu(g, h) = gh$ is smooth.
2. The inversion map $\iota : G \rightarrow G$ defined by $\iota(g) = g^{-1}$ is smooth.

Lie algebra is associated with each Lie group, a vector space equipped with a binary operation called the Lie bracket. The Lie algebra captures the local structure of the Lie group near the identity element, meaning that the Lie algebra of a Lie group G is the tangent space at the identity, denoted T_eG , equipped with the Lie bracket operation. The Lie bracket is defined in terms of the group operation and the differential.

There is a map from the Lie algebra to the Lie group called the exponential map, denoted $\exp : T_eG \rightarrow G$. The exponential map provides a way to generate new group elements from elements of the Lie algebra. In particular, given an element X of the Lie algebra, $\exp(X)$ is a group element close to the identity if X is ‘small’. We will exploit this concept to generate random data associated with a specific group:

We start with a set of generators T_a contained within the Lie-algebra \mathfrak{g} of a Lie group G , where $a = 1, 2, \dots, d_{\mathfrak{g}}$, i.e., the dimension of the Lie-algebra. We can then create elements $g \in G$ by employing:

$$g = e^{i \sum_a \theta_a T_a}, \quad \text{where } \theta_a \in [0, 2\pi] \quad . \quad (2)$$

We used the condition for our θ_a -values without loss of generality due to the periodicity of the exponential function. To generate our random data, we randomly choose our θ_a and create an element of our group. We then apply this element to a corresponding base vector of our vector space.

Specifically, in our example, we use the Lie-group $SU(2)$. The special unitary group of degree 2, denoted as $SU(2)$, is a Lie group of 2×2 unitary matrices with determinant 1.

$$SU(2) = \{U \in \mathbb{C}^{2 \times 2} : UU^\dagger = I, \det(U) = 1\} \tag{3}$$

The corresponding Lie algebra, $\mathfrak{su}(2)$, consists of 2×2 Hermitian traceless matrices, i.e., the Pauli matrices:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{4}$$

The commutation relations of the Pauli matrices form the structure of the $\mathfrak{su}(2)$ Lie algebra:

$$[\sigma_i, \sigma_j] = 2i\varepsilon_{ijk}\sigma_k \tag{5}$$

where $[\cdot, \cdot]$ denotes the commutator and ε_{ijk} is the Levi-Civita symbol.

To generate a classification dataset from this algebra, we use the following procedure:

1. Find a set of random parameters $\theta \in (0; \pi]$, $\phi \in (0; 2\pi]$, $\lambda \in (0; 2\pi]$;
2. We then create an element U of $SU(2)$ using these these randomly set parameters: $U = e^{i(\theta\sigma_1 + \phi\sigma_2 + \lambda\sigma_3)}$;
3. Next, we take one of the basevectors from \mathbb{C}^2 , denoted as \hat{v} to create a new complex vector \vec{v} using the previously obtained matrix U such that: $\vec{v} = U \cdot \hat{v}$;
4. This vector is then separated into four features F_j such that:

$$F_1 = \Re\{v_1\} \tag{6}$$

$$F_2 = \Im\{v_1\} \tag{7}$$

$$F_3 = \Re\{v_2\} \tag{8}$$

$$F_4 = \Im\{v_2\}, \tag{9}$$

where v_1 and v_2 denotes the individual components of the vector \vec{v} , and $\Re[\dots]$ and $\Im[\dots]$ denote their respective real and imaginary parts;

5. Finally, we assign a class label C to this collection of features such that:

$$C = \begin{cases} 0 & \text{if } \theta < \frac{\pi}{2} \\ 1 & \text{if } \theta > \frac{\pi}{2} \end{cases}, \tag{10}$$

and collect the features and the class label into one sample $[F_1, F_2, F_3, F_4, C]$. We repeat this process N_S times, starting with 1, where N_S is the number of samples that we want for our dataset.

Note that this approach can be extended to arbitrary Lie groups, given that one can construct or obtain a Lie group's generators.

4. Experimental Design

In this section, we describe our experimental design, which aims to provide a fair and comprehensive comparison of the performance of classical machine learning (ML) and quantum machine learning techniques, as discussed in Sections 3.2 and 3.3. Our experiments involve two main components: Firstly, assessing the algorithms' performance on artificially generated datasets with varying parametrizations, and secondly, evaluating the algorithms' performance on benchmark datasets using randomized search to optimize hyperparameters, ensuring a fair comparison. By carefully selecting our experimental setup, we avoid the issue of "cherry-picking" only a favorable subset of results, a common problem in machine learning, leading to heavily biased conclusions.

4.1. Artificially Generated Scikit Datasets

To generate the synthetic classification dataset, we utilized Scikit-learn's `make_classification` function. We employed two features and two classes while varying the number of samples to obtain a performance curve illustrating how the chosen algorithms' performance changes depending on the sample size.

We partitioned each dataset such that 20% of the original data were reserved as a test set to evaluate the trained algorithm, producing the accuracy score used for our assessment. Furthermore, each dataset was normalized such that all features are within the unit interval $[0, 1]$.

As a baseline, we employed the seven classical machine learning algorithms described in Section 3.2, namely Lasso, Ridge, MLP, SVM, XGBoost, LightGBM, and CatBoost. We used two different parameterizations for the classical machine learning algorithms for our comparisons. Firstly, we applied the out-of-the-box implementation without any hyperparameter optimization. Secondly, we used an optimized version of each algorithm found through Scikit-learn's `RandomizedSearchCV` by testing 20 different models.

We then examined 20 distinct parameter configurations, each for the VQC and QKE classifiers, randomly selected from a predefined parameter distribution. Appendix A discusses the parameter grids for all utilized algorithms and all experiments.

4.2. Artificially Generated SU(2) Datasets

For our synthetic SU(2) classification dataset, we used the concepts previously discussed in Section 3.5.1. We employed two complex features, i.e., resulting in four continuous real features, and two classes while varying the number of samples to obtain a performance curve illustrating how the chosen algorithms' performance changes depending on the sample size.

We partitioned each dataset such that 20% of the original data were reserved as a test set to evaluate the trained algorithm, producing the accuracy score used for our assessment. Furthermore, each dataset was normalized such that all features are within the unit interval $[0, 1]$.

As a baseline, we employed the seven classical machine learning algorithms described in Section 3.2, namely Lasso, Ridge, MLP, SVM, XGBoost, LightGBM, and CatBoost. We used two different parameterizations for the classical machine learning algorithms for our comparisons. Firstly, we applied the out-of-the-box implementation without any hyperparameter optimization. Secondly, we used an optimized version of each algorithm found through Scikit-learn's `RandomizedSearchCV` by testing 20 different models.

We then examined 20 distinct parameter configurations, each for the VQC and QKE classifiers, randomly selected from a predefined parameter distribution. Appendix A discusses the parameter grids for all utilized algorithms and all experiments.

4.3. Benchmark Datasets and Hyperparameter Optimization

Our last experiment was to test the two employed quantum machine learning algorithms against the classical machine learning algorithms on six benchmark datasets (Section 3.5). For this reason, we employed Scikit-learn's `RandomizedSearchCV` to test 20 randomly parameterized models for each algorithm to report the best of these tests. Again, we used a train-test-split to keep 20% of the original data to test the trained algorithm. Furthermore, each dataset was normalized such that all features are within the unit interval $[0, 1]$.

5. Results

In this section, we present the results of our experiments, comparing the performance of classical machine learning and quantum machine learning techniques on both artificially generated datasets and benchmark datasets (Section 3.5). By analyzing the results, we aim to draw meaningful insights into the strengths and weaknesses of each approach and provide a blueprint for future studies in the area. Everything was calculated on a Lenovo

ThinkCentre machine using an *Intel(R) Core(TM) i7-4770 CPU 3.40GHz* and 16GB RAM and Linux 20.04. We used python 3.6 and the included packages are the following:

- numpy version: 1.18.5
- sklearn version: 0.23.1
- catboost version: 0.26.1
- xgboost version: 1.2.1
- lightgbm version: 3.2.1
- qiskit version: {'qiskit-terra': '0.19.2', 'qiskit-aer': '0.10.3', 'qiskit-ignis': '0.7.0', 'qiskit-ibmq-provider': '0.18.3', 'qiskit-aqua': None, 'qiskit': '0.34.2', 'qiskit-nature': '0.3.1', 'qiskit-finance': None, 'qiskit-optimization': None, 'qiskit-machine-learning': '0.3.1'}
- qiskit_machine_learning version: 0.3.1

5.1. Performance on Artificially Generated Scikit Datasets

In this section, we compare the performance of quantum machine learning algorithms and classical machine learning algorithms on artificially generated classification datasets. The comprehensive experimental setup can be found in Section 4.1.

Regarding accuracy and runtime, our findings are presented in Tables 1 and 2 and Figures 3–5. The measured runtime includes hyperparameter tuning via randomized search and five-fold cross-validating, training, and testing the model.

While QML algorithms perform reasonably well, we observe that they are not a match for properly trained and/or sophisticated state-of-the-art classifiers. Even out-of-the-box implementations of state-of-the-art ML algorithms outperform QML algorithms on these artificially generated classification datasets.

The accuracy of the algorithms varies depending on the dataset size, with larger datasets posing more challenges. CatBoost performed best in our experiments, both out-of-the-box and when optimized in terms of high accuracy over all experiments. The quantum kernel estimator is the fifth-best algorithm overall in terms of accuracy, though it outperforms CatBoost regarding the runtime for CatBoost's optimized version. XGBoost and support vector classification (SVC) follow closely, with competitive performances in terms of accuracy. However, variational quantum circuit struggles to achieve high accuracy compared to sophisticated boosting classifiers or support vector machines. Furthermore, we observe the best performance in terms of runtime for the two linear models, Lasso and Ridge. We need to point out that Lasso and Ridge both feature increased runtimes for the datasets of size 50; this is most likely due to the optimizer needing an increased number of iterations due to the small number of samples and their relatively scattered distribution of data points.

Other algorithms, such as multilayer perceptron, Ridge regression, Lasso regression, and LightGBM, exhibit varying performances depending on dataset size and optimization. Despite some reasonable results from QKE, we conclude that classical ML algorithms, particularly sophisticated boosting classifiers, should be chosen to tackle similar problems due to their ease of implementation, better runtime, and overall superior performance.

In summary, while QML algorithms have shown some promise, they cannot yet compete with state-of-the-art classical ML algorithms on artificially generated classification datasets in terms of accuracy and runtime.

Table 1. This table presents the scores/accuracies of our experiments conducted on artificially generated classification datasets of varying sizes, e.g., 50 and 100. Given these different dataset sizes, this table is sorted in decreasing order of the average accuracy over all different sample sizes of each algorithm. The parametrization for the QKE is as follows: QKE, feature map, quantum simulator, C-Value for the SVM algorithm. The parametrization for the VQC is as follows: VQC, feature map, Ansatz, optimizer, quantum simulator. For the classical machine learning algorithms, *OutOfTheBox* means that we did not tune the hyperparameters of the employed algorithm and *RandomSearch* refers to hyperparameter optimization via a randomized search.

Algorithm/Parametrization	Size 50	Size 100	Size 250	Size 500	Size 1000	Size 1500	Size 2000	Average
OutOfTheBox, CatBoost, results	1.0	1.0	0.98	0.97	0.925	0.93	0.9425	0.963929
RandomSearch, CatBoost, results	1.0	1.0	0.96	0.96	0.935	0.936667	0.9425	0.962024
RandomSearch, SVM, results	1.0	1.0	0.94	0.96	0.945	0.93	0.9375	0.958929
RandomSearch, XGBoost, results	1.0	0.95	0.98	0.96	0.93	0.933333	0.9425	0.956548
QKE, PauliFeatureMap, statevector-simulator, 1000.0	1.0	1.0	0.96	0.93	0.93	0.93	0.925	0.953571
OutOfTheBox, XGBoost, results	1.0	0.95	0.94	0.96	0.91	0.936667	0.95	0.949524
OutOfTheBox, SVM, results	1.0	1.0	0.92	0.92	0.94	0.933333	0.93	0.949048
QKE, ZZFeatureMap, statevector-simulator, 177.82794100389228	1.0	1.0	0.94	0.93	0.915	0.926667	0.9225	0.947738
QKE, ZFeatureMap, statevector-simulator, 5.623413251903491	1.0	1.0	0.92	0.91	0.925	0.93	0.9375	0.946071
RandomSearch, MLP, results	1.0	1.0	0.94	0.88	0.905	0.933333	0.94	0.942619
OutOfTheBox, MLP, results	1.0	1.0	0.94	0.89	0.905	0.916667	0.9275	0.939881
OutOfTheBox, Ridge, results	1.0	1.0	0.94	0.88	0.9	0.896667	0.9025	0.93131
QKE, ZFeatureMap, qasm-simulator, 5.623413251903491	1.0	1.0	0.94	0.82	0.91	0.92	0.9025	0.9275
QKE, ZZFeatureMap, statevector-simulator, 31.622776601683793	1.0	0.95	0.92	0.88	0.88	0.926667	0.9175	0.924881
QKE, PauliFeatureMap, statevector-simulator, 5.623413251903491	1.0	0.95	0.92	0.85	0.895	0.93	0.92	0.923571
QKE, ZFeatureMap, statevector-simulator, 0.1778279410038923	1.0	0.95	0.9	0.88	0.9	0.92	0.9125	0.923214
QKE, ZFeatureMap, aer-simulator, 0.1778279410038923	1.0	0.95	0.9	0.87	0.905	0.92	0.9125	0.9225
RandomSearch, Ridge, results	1.0	1.0	0.9	0.88	0.88	0.893333	0.9025	0.922262
QKE, ZZFeatureMap, qasm-simulator, 5.623413251903491	1.0	0.95	0.92	0.86	0.89	0.91	0.9175	0.921071
QKE, PauliFeatureMap, qasm-simulator, 5.623413251903491	1.0	0.95	0.92	0.86	0.89	0.91	0.9175	0.921071
VQC, ZFeatureMap, EfficientSU2, COBYLA, statevector-simulator	1.0	0.95	0.9	0.9	0.92	0.893333	0.88	0.920476
RandomSearch, Lasso, results	1.0	1.0	0.94	0.82	0.895	0.893333	0.89	0.919762
VQC, ZFeatureMap, EfficientSU2, COBYLA, qasm-simulator	1.0	0.95	0.9	0.88	0.92	0.91	0.845	0.915
QKE, PauliFeatureMap, aer-simulator, 1.0	0.9	0.95	0.92	0.89	0.89	0.93	0.91	0.912857
VQC, ZFeatureMap, EfficientSU2, SPSA, qasm-simulator	1.0	0.95	0.9	0.86	0.925	0.91	0.845	0.912857
VQC, ZFeatureMap, EfficientSU2, COBYLA, aer-simulator	1.0	0.95	0.92	0.88	0.9	0.906667	0.8275	0.912024
VQC, ZFeatureMap, EfficientSU2, SPSA, statevector-simulator	1.0	0.95	0.92	0.87	0.89	0.89	0.835	0.907857
VQC, ZFeatureMap, RealAmplitudes, COBYLA, aer-simulator	1.0	0.95	0.9	0.86	0.905	0.85	0.865	0.904286
RandomSearch, LightGBM, results	0.4	1.0	0.98	0.95	0.93	0.933333	0.9475	0.877262
OutOfTheBox, LightGBM, results	0.4	1.0	0.96	0.94	0.925	0.936667	0.9375	0.87131
VQC, PauliFeatureMap, EfficientSU2, SPSA, qasm-simulator	0.9	0.75	0.9	0.84	0.89	0.86	0.8675	0.858214
VQC, ZFeatureMap, EfficientSU2, NFT, statevector-simulator	1.0	0.95	0.86	0.72	0.9	0.776667	0.77	0.85381
QKE, PauliFeatureMap, aer-simulator, 31.622776601683793	1.0	0.85	0.96	0.7	0.875	0.826667	0.735	0.849524
QKE, ZFeatureMap, aer-simulator, 31.622776601683793	1.0	1.0	0.88	0.62	0.835	0.736667	0.7475	0.83131
QKE, PauliFeatureMap, aer-simulator, 1000.0	1.0	0.85	0.96	0.58	0.87	0.826667	0.665	0.821667
VQC, PauliFeatureMap, EfficientSU2, SPSA, aer-simulator	0.8	0.75	0.9	0.73	0.845	0.86	0.8525	0.819643
VQC, PauliFeatureMap, EfficientSU2, NFT, statevector-simulator	0.8	0.65	0.9	0.8	0.84	0.783333	0.8475	0.802976
QKE, ZFeatureMap, qasm-simulator, 177.82794100389228	0.9	1.0	0.88	0.57	0.875	0.73	0.6375	0.798929

Table 1. Cont.

Algorithm/Parametrization	Size 50	Size 100	Size 250	Size 500	Size 1000	Size 1500	Size 2000	Average
VQC, ZZFeatureMap, EfficientSU2, COBYLA, aer-simulator	0.7	0.7	0.9	0.71	0.82	0.826667	0.835	0.784524
VQC, ZZFeatureMap, RealAmplitudes, COBYLA, qasm-simulator	0.8	0.7	0.9	0.62	0.775	0.816667	0.785	0.770952
VQC, ZZFeatureMap, RealAmplitudes, NFT, qasm-simulator	0.7	0.7	0.9	0.86	0.775	0.786667	0.535	0.750952
VQC, PauliFeatureMap, RealAmplitudes, NFT, qasm-simulator	0.6	0.7	0.9	0.49	0.8	0.763333	0.78	0.719048
VQC, ZZFeatureMap, RealAmplitudes, COBYLA, aer-simulator	0.5	0.65	0.84	0.73	0.83	0.83	0.575	0.707857
QKE, PauliFeatureMap, aer-simulator, 0.03162277660168379	0.4	0.35	0.9	0.65	0.86	0.923333	0.8275	0.701548
QKE, PauliFeatureMap, aer-simulator, 0.005623413251903491	0.4	0.35	0.9	0.49	0.75	0.766667	0.8275	0.640595
QKE, PauliFeatureMap, qasm-simulator, 0.005623413251903491	0.4	0.35	0.9	0.49	0.75	0.766667	0.8275	0.640595
QKE, ZFeatureMap, statevector-simulator, 0.005623413251903491	0.4	0.35	0.84	0.49	0.63	0.85	0.83	0.627143
VQC, ZFeatureMap, TwoLocal, SPSA, statevector-simulator	0.7	0.65	0.52	0.51	0.52	0.493333	0.58	0.567619
QKE, PauliFeatureMap, qasm-simulator, 0.001	0.4	0.35	0.9	0.49	0.48	0.753333	0.4975	0.552976
OutOfTheBox, Lasso, results	0.4	0.35	0.5	0.49	0.48	0.506667	0.4975	0.460595
VQC, ZZFeatureMap, TwoLocal, COBYLA, qasm-simulator	0.2	0.35	0.28	0.35	0.225	0.216667	0.3975	0.288452
VQC, PauliFeatureMap, TwoLocal, SPSA, qasm-simulator	0.2	0.35	0.26	0.38	0.185	0.223333	0.4	0.285476
VQC, PauliFeatureMap, TwoLocal, COBYLA, statevector-simulator	0.2	0.35	0.28	0.36	0.19	0.223333	0.39	0.284762
VQC, PauliFeatureMap, TwoLocal, SPSA, statevector-simulator	0.2	0.35	0.28	0.36	0.19	0.223333	0.39	0.284762

Table 2. This table presents the runtimes of our experiments conducted on artificially generated classification datasets of varying sizes, e.g., 50 and 100. Given these different dataset sizes, this table is sorted in increasing order of the average runtime over all different sample sizes of each algorithm. The measured runtime includes hyperparameter tuning via randomized search and five-fold cross-validating, training, and testing the model. The parametrization for the QKE is as follows: QKE, feature map, quantum simulator, C-Value for the SVM algorithm. The parametrization for the VQC is as follows: VQC, feature map, Ansatz, optimizer, quantum simulator. For the classical machine learning algorithms, *OutOfTheBox* means that we did not tune the hyperparameters of the employed algorithm and *RandomSearch* refers to hyperparameter optimization via a randomized search.

Algorithm/Parametrization	Size 50	Size 100	Size 250	Size 500	Size 1000	Size 1500	Size 2000	Average
OutOfTheBox, Lasso, results	0.001473	0.001162	0.001158	0.001123	0.001141	0.001153	0.001159	0.001196
OutOfTheBox, Ridge, results	0.002933	0.001553	0.001433	0.001894	0.002628	0.002575	0.002436	0.002207
OutOfTheBox, SVM, results	0.001021	0.000648	0.001039	0.002457	0.005501	0.017243	0.0295	0.008201
OutOfTheBox, XGBoost, results	0.016881	0.017187	0.022922	0.038751	0.05111	0.151807	0.120973	0.059947
OutOfTheBox, LightGBM, results	0.009655	0.024887	0.104107	0.124862	0.1898	0.489043	0.218343	0.165814
RandomSearch, Lasso, results	1.045328	0.113413	0.102258	0.105736	0.104031	0.120507	0.116006	0.243897
RandomSearch, Ridge, results	1.120708	0.122188	0.114706	0.175996	0.226949	0.255845	0.25067	0.323866
RandomSearch, SVM, results	1.06376	0.135593	0.163875	0.159699	0.203163	0.354172	0.442741	0.360429
OutOfTheBox, MLP, results	0.082953	0.091169	0.121317	0.232771	0.451674	0.947373	1.376965	0.472032
OutOfTheBox, CatBoost, results	0.389826	0.411965	0.654325	0.783825	0.867595	1.085298	1.1931	0.769419
RandomSearch, LightGBM, results	1.711872	0.376494	0.58387	0.704715	0.728305	0.897428	1.000039	0.857532
RandomSearch, XGBoost, results	1.572541	0.399174	0.441059	0.577969	0.99776	1.467667	1.352474	0.972663
VQC, ZFeatureMap, TwoLocal, SPSA, statevector-simulator	0.502447	0.82391	1.319602	2.9078	6.75953	11.81601	18.064725	6.027718
VQC, PauliFeatureMap, TwoLocal, COBYLA, statevector-simulator	0.536454	0.886945	1.757877	3.486975	8.137821	14.688881	22.79476	7.469959
VQC, PauliFeatureMap, TwoLocal, SPSA, statevector-simulator	1.981785	0.715829	1.621059	3.488372	8.517624	15.170185	22.300972	7.685118
VQC, PauliFeatureMap, TwoLocal, SPSA, qasm-simulator	0.750719	1.154406	2.53449	5.000262	11.265137	19.493945	29.031463	9.89006
VQC, ZZFeatureMap, TwoLocal, COBYLA, qasm-simulator	0.734865	1.097202	2.514703	4.990832	11.895971	19.283406	29.318269	9.976464
RandomSearch, MLP, results	3.899634	3.298337	5.003618	9.651274	14.729924	20.652249	31.202069	12.633872
QKE, ZFeatureMap, statevector-simulator, 0.1778279410038923	1.343983	0.802286	2.170829	5.965899	18.504546	36.659922	59.889941	17.905344

Table 2. Cont.

Algorithm/Parametrization	Size 50	Size 100	Size 250	Size 500	Size 1000	Size 1500	Size 2000	Average
QKE, ZFeatureMap, statevector-simulator, 0.005623413251903491	0.411296	0.697461	2.154164	6.122564	19.670819	37.297334	62.1901	18.363391
QKE, PauliFeatureMap, statevector-simulator, 1000.0	0.470933	0.956269	2.721257	7.2817	21.356298	40.130716	67.422908	20.048583
QKE, PauliFeatureMap, statevector-simulator, 5.623413251903491	0.501446	0.922237	2.775664	7.454642	21.780637	40.426036	66.758927	20.088513
QKE, ZFeatureMap, statevector-simulator, 5.623413251903491	0.378018	0.757363	2.141677	4.962464	19.901565	41.913003	71.453831	20.215417
QKE, ZZFeatureMap, statevector-simulator, 31.622776601683793	0.214386	0.567282	1.650304	5.302437	20.77629	42.614517	72.871078	20.570899
QKE, ZZFeatureMap, statevector-simulator, 177.82794100389228	0.461093	0.943574	2.780804	7.580857	22.906811	41.955521	68.045553	20.667745
RandomSearch, CatBoost, results	8.627878	10.873142	26.728395	35.20857	36.902272	56.253265	37.994929	30.369779
VQC, ZFeatureMap, RealAmplitudes, COBYLA, aer-simulator	47.438183	63.446748	192.148143	404.233954	1060.291657	1619.397205	2290.222381	811.025467
VQC, ZZFeatureMap, RealAmplitudes, COBYLA, qasm-simulator	43.113636	83.175558	166.040938	421.278374	1064.238564	1702.893006	2719.340939	885.725859
VQC, ZZFeatureMap, RealAmplitudes, COBYLA, aer-simulator	45.909504	83.201411	152.20265	509.1956	1158.902532	1654.065907	2603.942577	886.774312
VQC, ZFeatureMap, EfficientSU2, COBYLA, statevector-simulator	48.546243	81.030425	190.958188	402.121722	1044.855825	1807.676357	2751.241623	903.775769
VQC, ZFeatureMap, EfficientSU2, COBYLA, aer-simulator	57.728111	100.590997	240.174666	507.58709	1253.080578	2139.855218	3196.07247	1070.727019
VQC, ZFeatureMap, EfficientSU2, COBYLA, qasm-simulator	59.058898	100.862056	242.285405	507.171731	1262.650143	2151.503499	3191.745568	1073.611043
VQC, ZZFeatureMap, EfficientSU2, COBYLA, aer-simulator	59.651649	105.629842	254.918442	601.245125	1335.017904	2260.354294	3366.65501	1140.496038
QKE, ZFeatureMap, qasm-simulator, 177.82794100389228	4.589478	13.184805	82.633779	332.71327	1337.102907	3020.689579	5368.201509	1451.30219
QKE, ZZFeatureMap, qasm-simulator, 5.623413251903491	4.352785	15.921249	97.165028	390.472092	1573.103197	3549.629798	6282.670251	1701.902057
QKE, PauliFeatureMap, aer-simulator, 0.03162277660168379	3.549125	15.094144	98.970568	393.496921	1581.662241	3554.962927	6317.355669	1709.298799
QKE, PauliFeatureMap, aer-simulator, 0.005623413251903491	3.373257	15.311538	99.2351	390.52131	1574.108371	3555.3048	6339.026443	1710.982974
QKE, PauliFeatureMap, qasm-simulator, 0.005623413251903491	3.812115	19.479307	101.289711	404.432384	1636.24686	3642.937393	6307.605039	1730.828973
QKE, PauliFeatureMap, aer-simulator, 31.622776601683793	3.848578	17.062982	101.387533	408.69903	1635.863136	3674.976257	6555.811507	1771.092718
VQC, ZFeatureMap, EfficientSU2, NFT, statevector-simulator	98.831974	167.48274	394.378037	836.913451	2197.652135	3719.047116	5621.134708	1862.205737
VQC, PauliFeatureMap, EfficientSU2, NFT, statevector-simulator	103.914165	177.047181	423.423603	1014.963511	2338.078356	3953.861723	5905.433094	1988.10309
VQC, ZZFeatureMap, RealAmplitudes, NFT, qasm-simulator	105.987181	183.918751	427.016702	1036.605473	2366.463152	4052.521035	6042.538015	2030.721473
VQC, PauliFeatureMap, RealAmplitudes, NFT, qasm-simulator	103.625823	180.306618	425.488049	1041.160999	2371.366715	4044.856475	6048.573929	2030.768373
VQC, ZFeatureMap, EfficientSU2, SPSA, statevector-simulator	119.513477	200.101417	474.113288	1008.932874	2601.731917	4505.306268	6781.089745	2241.541284
VQC, ZFeatureMap, EfficientSU2, SPSA, qasm-simulator	145.295744	256.711762	609.791229	1272.675059	3150.527537	5366.116602	8009.649075	2687.25243
VQC, PauliFeatureMap, EfficientSU2, SPSA, aer-simulator	144.280811	259.102175	625.193096	1502.476923	3356.340799	5689.827615	8454.144295	2861.623673
VQC, PauliFeatureMap, EfficientSU2, SPSA, qasm-simulator	152.666649	269.680847	642.400747	1505.762521	3388.662998	5709.505826	8438.957709	2872.519614
QKE, ZFeatureMap, aer-simulator, 31.622776601683793	5.993241	25.852654	166.703792	669.201309	2934.169598	6729.31411	12,037.430687	3224.095056
QKE, PauliFeatureMap, qasm-simulator, 5.623413251903491	8.384715	32.795287	206.595473	890.414904	3753.488868	8537.768589	15,232.745542	4094.599054
QKE, PauliFeatureMap, qasm-simulator, 0.001	7.792093	32.566225	207.832614	896.042249	3778.324351	8610.335147	15,348.810142	4125.957546
QKE, ZFeatureMap, aer-simulator, 0.1778279410038923	10.511296	43.335078	276.810734	1111.545614	4799.032996	10,979.135601	19,768.073574	5284.063556
QKE, ZFeatureMap, qasm-simulator, 5.623413251903491	11.573929	43.186982	277.291314	1113.664313	4842.587094	10,978.908476	19,798.821156	5295.147609
QKE, PauliFeatureMap, aer-simulator, 1000.0	12.596938	51.788837	332.281104	1434.208601	5986.631006	13,592.866065	24,280.544075	6527.273804
QKE, PauliFeatureMap, aer-simulator, 1.0	12.261604	51.508959	332.561822	1423.111135	5984.902587	13,603.956887	24,362.83202	6538.733573

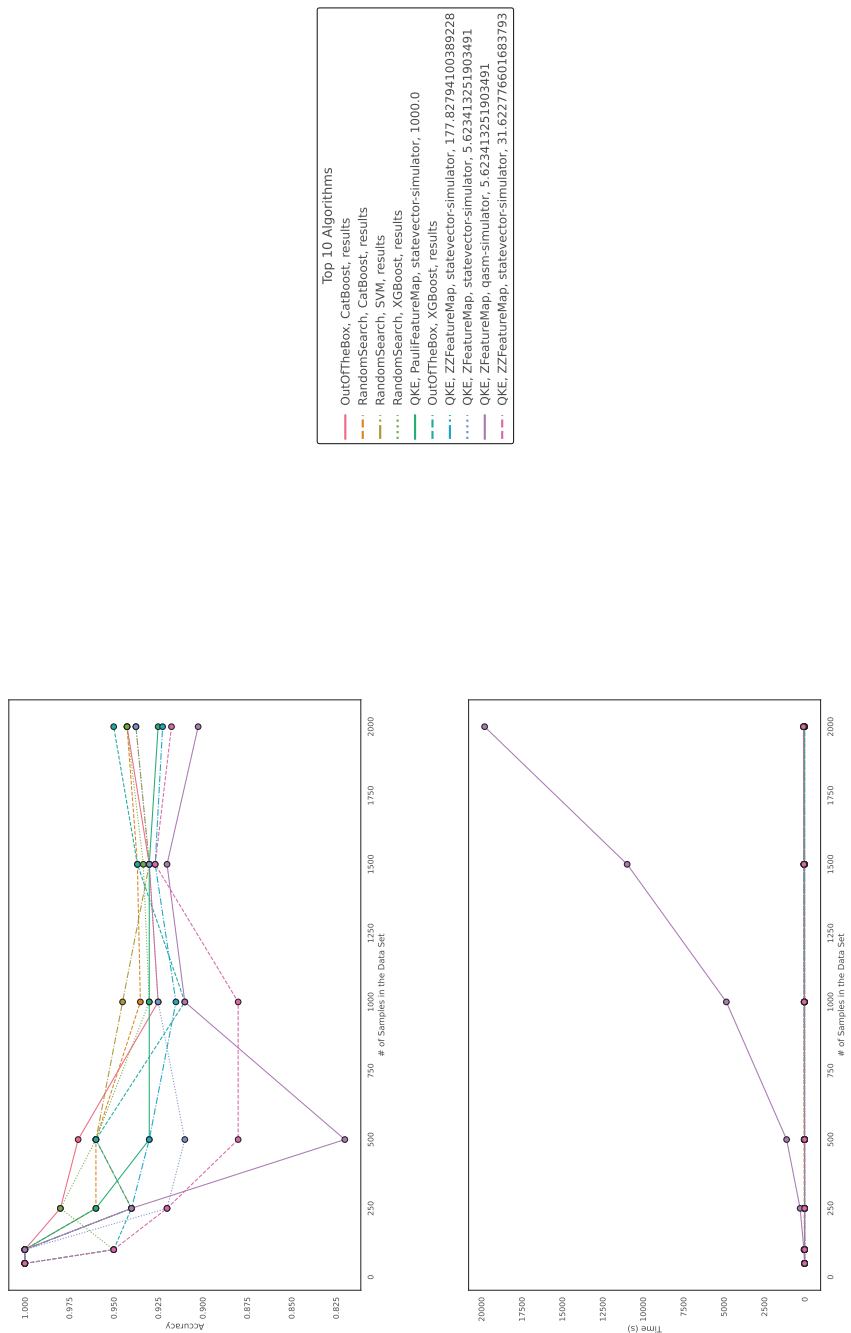


Figure 3. These figures depict the results from our experiments, comparing the five best QML and classical ML algorithms on artificially generated datasets in terms of accuracy. The upper part illustrates the accuracy of the algorithms on different sample sizes, while the lower part demonstrates how the runtimes change with increasing size of the test dataset. The right part contains the legend, indicating which algorithms were used, and more specifically, the different parametrizations of the employed quantum machine learning algorithms. Furthermore, the legend is sorted in decreasing order of the average accuracy of the employed algorithms. The parametrization for the QKE is as follows: QKE, feature map, quantum simulator, C-Value for the SVM algorithm.

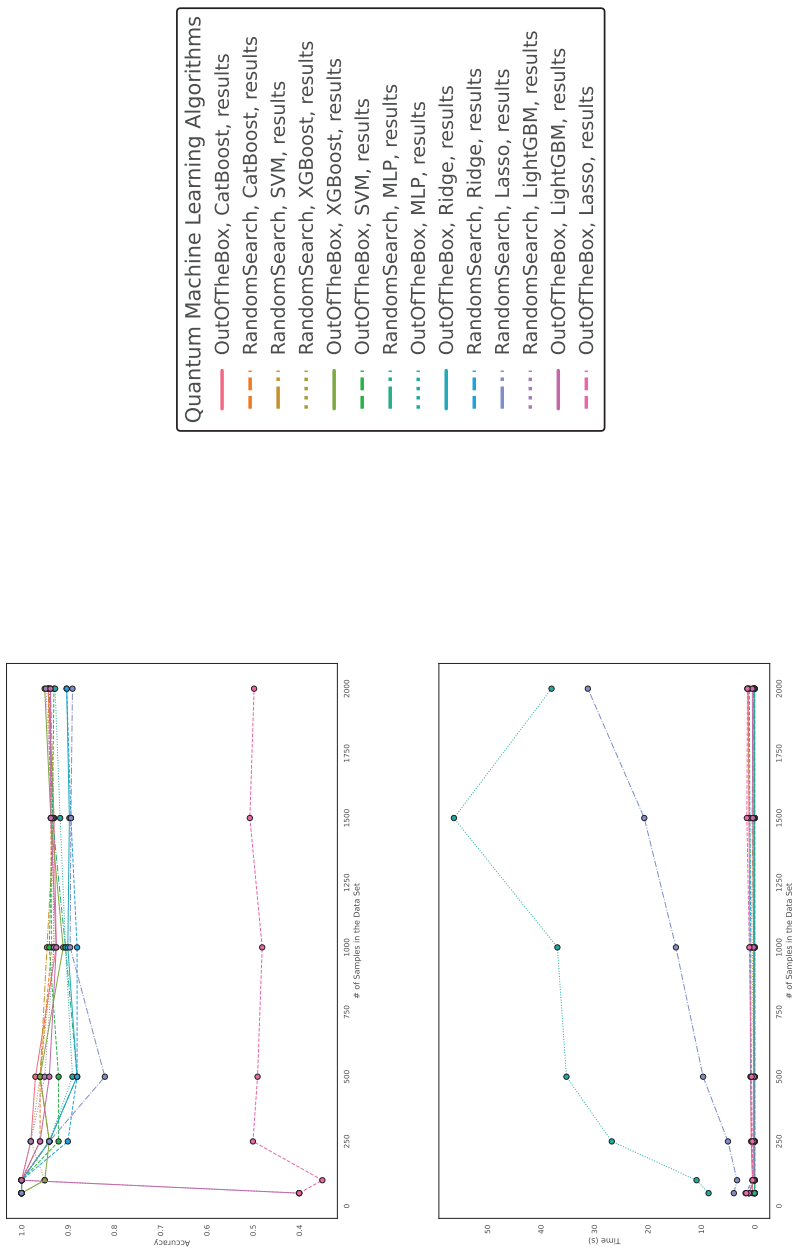


Figure 4. These figures depict the results from our experiments, comparing differently parameterized classical machine learning algorithms on artificially generated datasets. The upper part illustrates the behavior of the accuracies, while the lower part demonstrates how the run times change with the increasing size of the test dataset. The legend, indicating which algorithms were used, and more specifically, the different parametrizations of the employed machine learning algorithms. Furthermore, the legend is sorted in decreasing order of the average accuracy of the employed algorithms.

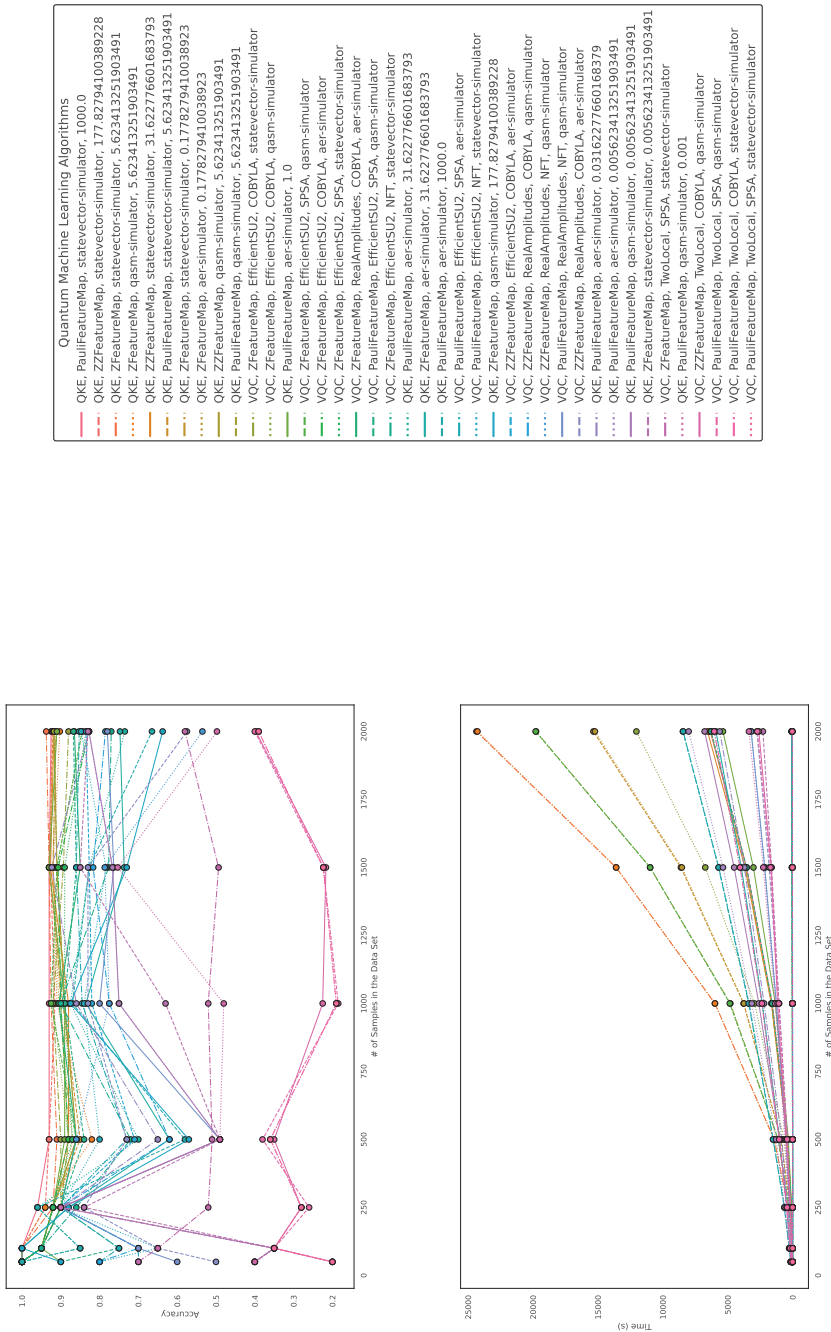


Figure 5. These figures depict the results from our experiments for the artificially generated datasets, comparing differently parameterized QML algorithms on artificially generated datasets. The **upper part** illustrates the behavior of the accuracies, while the **lower part** demonstrates how the runtimes change with the increasing size of the test datasets. The **right part** contains the legend, indicating which algorithms were used, and more specifically, the different parametrizations of the employed quantum machine learning algorithms. Furthermore, the legend is sorted in decreasing order of the average accuracy of the employed algorithms. The parametrization for the QKE is as follows: QKE, feature map, quantum simulator, C-Value for the SVM algorithm. The parametrization for the VQC is as follows: VQC, feature map, Ansatz, optimizer, quantum simulator.

5.2. Performance on Artificially Generated SU2 Datasets

In this section, we compare the performance of quantum machine learning algorithms and classical machine learning algorithms on artificially generated classification datasets based on Lie group structures. The detailed experimental setup can be found in Section 4.2.

Regarding accuracy and runtime, our findings are presented in Tables 3 and 4 and Figures 6–8. While QML algorithms perform reasonably well, we observe that they are not a match for properly trained and/or sophisticated state-of-the-art classifiers. Even out-of-the-box implementations of state-of-the-art ML algorithms outperform QML algorithms on artificially generated classification datasets that are particularly suited for QML.

The accuracy of the algorithms varies depending on the dataset size, with larger datasets providing increased accuracy for most algorithms. CatBoost performed best in our experiments, both out-of-the-box and when optimized in terms of high accuracy over all experiments. The quantum kernel estimator is the fifth-best algorithm overall in terms of accuracy. However, we observe that, on average, CatBoost with improved hyperparameters performs best over all experiments, but is outperformed by the best QKE implementation for 100 and 500 data points. Thus, we conclude that quantum kernel estimators can capture the complexity of this SU(2)-generated dataset, but overall, one is better off with an out-of-the-box CatBoost implementation. This means that we do not observe a quantum advantage for this type of data, but rather that the employed quantum kernel estimator behaves similarly to classical machine learning algorithms, i.e., it exhibits reasonable performance but does not perform best for all datasets, even the ones created by exploiting quantum symmetry properties.

Other algorithms, such as multilayer perceptron, Ridge regression, Lasso regression, and LightGBM, exhibit varying performances depending on dataset size and optimization. Despite some reasonable results from QKE, we conclude that classical ML algorithms, particularly sophisticated boosting classifiers, should be chosen to tackle similar problems due to their ease of implementation, better runtime, and overall superior performance. Furthermore, we again observe the best performance in terms of runtime for the two linear models, Lasso and Ridge. Moreover, again, we observe that Lasso and Ridge both feature increased runtimes for the datasets of size 50.

In summary, while QML algorithms have shown some promise, they cannot yet compete with state-of-the-art classical ML algorithms even on these SU(2)-datasets, where the authors intended to provide evidence for the quantum advantage for datasets generated from symmetry properties inherent to quantum mechanics.

Table 3. This table presents the scores/accuracies of our experiments conducted on classification datasets generated via SU(2) generators of varying sizes, e.g., 50 and 100. Given these different dataset sizes, this table is sorted in decreasing order of the average accuracy over all different sample sizes of each algorithm. The parametrization for the QKE is as follows: QKE, feature map, quantum simulator, C-Value for the SVM algorithm. The parametrization for the VQC is as follows: VQC, feature map, Ansatz, optimizer, quantum simulator. For the classical machine learning algorithms, *OutOfTheBox* means that we did not tune the hyperparameters of the employed algorithm and *RandomSearch* refers to hyperparameter optimization via a randomized search.

Algorithm/Parametrization	Size 50	Size 100	Size 250	Size 500	Size 1000	Size 1500	Size 2000	Average
RandomSearch, CatBoost, results	0.9	0.55	0.78	0.78	0.88	0.906667	0.915	0.815952
OutOfTheBox, CatBoost, results	0.6	0.7	0.76	0.85	0.895	0.906667	0.9375	0.807024
RandomSearch, XGBoost, results	0.6	0.65	0.74	0.84	0.87	0.86	0.9425	0.786071
OutOfTheBox, XGBoost, results	0.4	0.75	0.76	0.86	0.89	0.926667	0.9	0.78381
QKE, ZFeatureMap, statevector-simulator, 1000.0	0.7	0.8	0.74	0.79	0.8	0.806667	0.8475	0.783452
RandomSearch, LightGBM, results	0.3	0.6	0.74	0.87	0.89	0.903333	0.9175	0.745833
OutOfTheBox, LightGBM, results	0.3	0.6	0.74	0.9	0.85	0.923333	0.885	0.742619
QKE, ZZFeatureMap, statevector-simulator, 177.82794100389228	0.8	0.55	0.54	0.71	0.795	0.826667	0.85	0.724524

Table 3. Cont.

Algorithm/Parametrization	Size 50	Size 100	Size 250	Size 500	Size 1000	Size 1500	Size 2000	Average
QKE, PauliFeatureMap, aer-simulator, 5.623413251903491	0.7	0.65	0.44	0.73	0.725	0.683333	0.72	0.664048
QKE, ZZFeatureMap, qasm-simulator, 31.622776601683793	0.7	0.7	0.7	0.59	0.615	0.62	0.665	0.655714
QKE, ZZFeatureMap, statevector-simulator, 0.1778279410038923	0.8	0.5	0.62	0.55	0.645	0.693333	0.7375	0.649405
QKE, ZZFeatureMap, aer-simulator, 0.1778279410038923	0.4	0.6	0.7	0.67	0.625	0.713333	0.7075	0.630833
QKE, PauliFeatureMap, aer-simulator, 0.1778279410038923	0.5	0.65	0.64	0.63	0.615	0.68	0.7	0.630714
QKE, PauliFeatureMap, statevector-simulator, 0.1778279410038923	0.4	0.65	0.6	0.55	0.75	0.666667	0.73	0.620952
OutOfTheBox, MLP, results	0.8	0.55	0.52	0.63	0.555	0.56	0.62	0.605
VQC, ZZFeatureMap, EfficientSU2, SPSA, aer-simulator	0.7	0.5	0.6	0.6	0.605	0.576667	0.645	0.60381
VQC, ZZFeatureMap, EfficientSU2, COBYLA, qasm-simulator	0.6	0.6	0.62	0.54	0.6	0.606667	0.66	0.60381
OutOfTheBox, SVM, results	0.4	0.65	0.44	0.59	0.68	0.676667	0.695	0.590238
RandomSearch, SVM, results	0.2	0.7	0.8	0.65	0.78	0.503333	0.4725	0.586548
VQC, ZZFeatureMap, EfficientSU2, NFT, statevector-simulator	0.5	0.6	0.54	0.56	0.7	0.58	0.61	0.584286
VQC, ZZFeatureMap, RealAmplitudes, COBYLA, statevector-simulator	0.6	0.65	0.6	0.53	0.575	0.543333	0.5775	0.582262
VQC, PauliFeatureMap, EfficientSU2, COBYLA, aer-simulator	0.4	0.75	0.52	0.55	0.56	0.606667	0.6225	0.572738
VQC, PauliFeatureMap, RealAmplitudes, NFT, statevector-simulator	0.3	0.65	0.52	0.69	0.63	0.586667	0.62	0.570952
OutOfTheBox, Ridge, results	0.7	0.55	0.62	0.48	0.575	0.533333	0.525	0.569048
VQC, ZFeatureMap, RealAmplitudes, COBYLA, qasm-simulator	0.7	0.5	0.52	0.57	0.59	0.573333	0.5275	0.56869
VQC, ZZFeatureMap, EfficientSU2, NFT, aer-simulator	0.4	0.7	0.56	0.58	0.575	0.543333	0.6175	0.567976
QKE, ZZFeatureMap, aer-simulator, 31.622776601683793	0.4	0.55	0.6	0.62	0.625	0.596667	0.555	0.56381
QKE, PauliFeatureMap, aer-simulator, 31.622776601683793	0.6	0.45	0.6	0.56	0.57	0.62	0.5425	0.563214
VQC, ZZFeatureMap, RealAmplitudes, COBYLA, aer-simulator	0.6	0.65	0.48	0.55	0.535	0.573333	0.545	0.561905
QKE, ZFeatureMap, qasm-simulator, 177.82794100389228	0.6	0.7	0.5	0.52	0.5	0.556667	0.53	0.558095
VQC, ZFeatureMap, EfficientSU2, COBYLA, aer-simulator	0.7	0.6	0.5	0.48	0.525	0.48	0.615	0.557143
VQC, ZZFeatureMap, RealAmplitudes, NFT, qasm-simulator	0.6	0.4	0.58	0.47	0.66	0.573333	0.61	0.55619
VQC, ZFeatureMap, EfficientSU2, COBYLA, qasm-simulator	0.7	0.5	0.56	0.55	0.51	0.566667	0.505	0.555952
QKE, PauliFeatureMap, qasm-simulator, 0.1778279410038923	0.5	0.35	0.36	0.63	0.655	0.693333	0.7025	0.555833
QKE, ZZFeatureMap, qasm-simulator, 0.001	0.7	0.45	0.66	0.57	0.54	0.466667	0.4725	0.55131
VQC, ZFeatureMap, RealAmplitudes, SPSA, aer-simulator	0.7	0.5	0.5	0.59	0.495	0.516667	0.555	0.550952
VQC, ZFeatureMap, EfficientSU2, SPSA, statevector-simulator	0.3	0.8	0.44	0.59	0.55	0.563333	0.595	0.548333
VQC, PauliFeatureMap, RealAmplitudes, NFT, qasm-simulator	0.4	0.55	0.6	0.54	0.59	0.553333	0.58	0.544762
VQC, ZFeatureMap, RealAmplitudes, COBYLA, aer-simulator	0.8	0.45	0.48	0.48	0.52	0.513333	0.545	0.54119
VQC, ZFeatureMap, EfficientSU2, COBYLA, statevector-simulator	0.5	0.55	0.56	0.56	0.54	0.516667	0.56	0.540952
QKE, ZFeatureMap, aer-simulator, 1000.0	0.6	0.55	0.6	0.52	0.505	0.486667	0.52	0.540238
VQC, PauliFeatureMap, EfficientSU2, SPSA, qasm-simulator	0.4	0.45	0.56	0.64	0.575	0.533333	0.6075	0.537976
VQC, ZFeatureMap, RealAmplitudes, NFT, aer-simulator	0.5	0.4	0.62	0.52	0.58	0.533333	0.5875	0.534405
VQC, PauliFeatureMap, EfficientSU2, SPSA, aer-simulator	0.4	0.45	0.54	0.54	0.64	0.56	0.6	0.532857
QKE, ZFeatureMap, statevector-simulator, 0.1778279410038923	0.5	0.45	0.54	0.54	0.585	0.556667	0.555	0.532381
QKE, ZFeatureMap, aer-simulator, 1.0	0.4	0.5	0.4	0.63	0.56	0.603333	0.575	0.524048
QKE, ZFeatureMap, qasm-simulator, 1000.0	0.6	0.5	0.44	0.59	0.525	0.443333	0.5	0.514048
QKE, PauliFeatureMap, statevector-simulator, 0.03162277660168379	0.5	0.45	0.58	0.56	0.46	0.476667	0.5675	0.513452
RandomSearch, MLP, results	0.3	0.25	0.5	0.6	0.635	0.64	0.6675	0.513214
QKE, ZFeatureMap, aer-simulator, 177.82794100389228	0.4	0.65	0.54	0.46	0.495	0.48	0.565	0.512857
QKE, ZFeatureMap, qasm-simulator, 0.005623413251903491	0.5	0.65	0.48	0.47	0.515	0.48	0.485	0.511429
RandomSearch, Ridge, results	0.4	0.55	0.52	0.46	0.56	0.52	0.5675	0.511071
OutOfTheBox, Lasso, results	0.5	0.7	0.4	0.51	0.495	0.46	0.495	0.508571
RandomSearch, Lasso, results	0.6	0.45	0.44	0.45	0.515	0.533333	0.54	0.504048

Table 4. This table presents the scores/accuracies of our experiments conducted on classification datasets generated via SU(2) generators of varying sizes, e.g., 50 and 100. Given these different dataset sizes, this table is sorted in increasing order of the average runtime over all different sample sizes of each algorithm. The measured runtime includes hyperparameter tuning via randomized search and five-fold cross-validating, training, and testing the model. The parametrization for the QKE is as follows: QKE, feature map, quantum simulator, C-Value for the SVM algorithm. The parametrization for the VQC is as follows: VQC, feature map, Ansatz, optimizer, quantum simulator. For the classical machine learning algorithms, *OutOfTheBox* means that we did not tune the hyperparameters of the employed algorithm and *RandomSearch* refers to hyperparameter optimization via a randomized search.

Algorithm/Parametrization	Size 50	Size 100	Size 250	Size 500	Size 1000	Size 1500	Size 2000	Average
OutOfTheBox, Lasso, results	0.004103	0.000646	0.000661	0.001249	0.000804	0.000694	0.000691	0.001264
OutOfTheBox, Ridge, results	0.003733	0.002898	0.002786	0.029688	0.001899	0.001929	0.00178	0.006388
OutOfTheBox, SVM, results	0.00111	0.000919	0.002298	0.006078	0.012139	0.025935	0.04667	0.013593
RandomSearch, Lasso, results	1.055654	0.123047	0.103457	0.104252	0.117591	0.124301	0.115839	0.249163
RandomSearch, Ridge, results	1.084348	0.122741	0.138248	0.145562	0.156783	0.129915	0.14262	0.274317
OutOfTheBox, XGBoost, results	0.026616	0.040969	0.207265	1.348421	0.190274	0.123878	0.205496	0.306131
RandomSearch, SVM, results	1.133957	0.167929	0.131167	0.147877	0.233314	0.444834	0.438209	0.385327
OutOfTheBox, CatBoost, results	1.072432	0.439886	0.68483	0.813699	0.936663	1.13145	1.252242	0.904457
RandomSearch, XGBoost, results	1.555324	0.398761	0.517864	0.817118	1.79372	2.056436	2.289563	1.346969
OutOfTheBox, LightGBM, results	0.134499	0.935128	0.319397	1.125076	5.646488	4.908206	4.312101	2.482985
OutOfTheBox, MLP, results	0.447343	0.385258	0.303056	1.874662	2.165486	5.582801	6.986128	2.534962
RandomSearch, LightGBM, results	3.76752	0.726469	0.892496	1.349411	5.942432	5.569574	3.741842	3.141392
QKE, ZFeatureMap, statevector-simulator, 1000.0	0.588803	1.105112	3.115228	9.692818	24.488245	46.863381	75.605305	23.065556
QKE, ZZFeatureMap, statevector-simulator, 177.82794100389228	1.023377	2.014738	6.133744	14.08855	36.209869	63.510698	97.575217	31.508028
QKE, PauliFeatureMap, statevector-simulator, 0.1778279410038923	1.110054	2.154783	6.591036	14.487802	36.089562	62.912848	97.64273	31.569831
RandomSearch, MLP, results	16.90452	15.129156	5.671446	42.576796	38.564641	70.807035	64.602048	36.322235
QKE, ZFeatureMap, statevector-simulator, 0.1778279410038923	1.235019	1.984607	5.547594	15.333808	41.977686	79.052726	127.710628	38.977438
QKE, PauliFeatureMap, statevector-simulator, 0.03162277660168379	1.453579	2.804056	9.085313	19.440721	47.633347	81.740004	128.186636	41.477665
QKE, ZZFeatureMap, statevector-simulator, 0.1778279410038923	2.194236	4.856549	10.043207	20.391739	57.658834	97.515617	151.070676	49.104408
RandomSearch, CatBoost, results	18.350174	35.654742	40.725868	70.65956	68.788446	68.958949	43.685619	49.546194
VQC, ZFeatureMap, RealAmplitudes, COBYLA, qasm-simulator	55.500573	100.423827	241.59017	577.400258	1286.962867	2164.282246	3315.185871	1105.906545
VQC, ZFeatureMap, RealAmplitudes, COBYLA, aer-simulator	61.174412	116.296597	274.944359	672.215972	1509.987298	2624.710516	4058.819283	1331.164062
VQC, ZZFeatureMap, RealAmplitudes, COBYLA, statevector-simulator	68.600875	123.207802	380.446356	770.716185	1635.89919	2621.242822	3805.021206	1343.590634
VQC, PauliFeatureMap, EfficientSU2, COBYLA, aer-simulator	89.832315	163.70887	480.279572	975.291255	2084.174564	3407.844934	5050.433405	1750.223559
VQC, ZZFeatureMap, EfficientSU2, COBYLA, qasm-simulator	88.21128	163.988243	480.35886	973.614566	2068.950356	3425.564759	5057.849761	1751.219689
VQC, ZFeatureMap, EfficientSU2, COBYLA, qasm-simulator	85.847133	156.496491	381.392026	888.461174	2300.527629	3878.06995	6136.175247	1975.281379
VQC, ZFeatureMap, EfficientSU2, COBYLA, statevector-simulator	103.51928	191.066923	456.017277	1079.39181	2305.444005	3940.212229	5958.277453	2004.846997

Table 4. Cont.

Algorithm/Parametrization	Size 50	Size 100	Size 250	Size 500	Size 1000	Size 1500	Size 2000	Average
VQC, ZFeatureMap, RealAmplitudes, NFT, aer-simulator	111.03018	203.235006	491.978903	1181.668828	2620.544781	4428.84701	6770.538213	2258.263274
VQC, ZFeatureMap, EfficientSU2, COBYLA, aer-simulator	113.765615	205.589837	516.289881	1202.288662	2663.121947	4533.378097	6730.84663	2280.754381
VQC, ZZFeatureMap, RealAmplitudes, COBYLA, aer-simulator	111.686165	209.034186	638.294179	1296.48554	2869.708946	4781.981067	7074.925479	2426.016509
VQC, PauliFeatureMap, RealAmplitudes, NFT, qasm-simulator	163.24924	303.991817	936.898012	1915.698994	4145.466983	6900.43208	10,273.374065	3519.873027
VQC, ZFeatureMap, EfficientSU2, SPSA, statevector-simulator	190.048137	341.454534	823.875859	1930.637818	4188.237967	6999.258807	10,586.6448	3580.02256
VQC, PauliFeatureMap, RealAmplitudes, NFT, statevector-simulator	190.485358	349.906865	1108.180672	2248.345232	4814.143806	7883.610885	11,821.059517	4059.390334
VQC, ZFeatureMap, RealAmplitudes, SPSA, aer-simulator	195.45132	357.101921	856.679886	2110.857992	4766.449826	8178.593323	12,549.799371	4144.99052
VQC, ZZFeatureMap, RealAmplitudes, NFT, qasm-simulator	224.602174	405.497928	1270.741552	2674.141109	5725.217252	9676.191547	14,306.04881	4897.491482
VQC, ZZFeatureMap, EfficientSU2, NFT, statevector-simulator	243.54281	457.172266	1372.528644	2784.422174	5896.297166	9666.126191	14,179.331314	4942.774366
QKE, ZFeatureMap, aer-simulator, 1.0	9.847104	40.399745	258.379268	1171.612393	4847.638381	10,994.736629	19,569.132042	5270.249366
VQC, ZZFeatureMap, EfficientSU2, NFT, aer-simulator	259.240503	473.126474	1408.601057	2903.563737	6301.786963	10,332.959201	15,345.314436	5289.227482
VQC, PauliFeatureMap, EfficientSU2, SPSA, aer-simulator	316.140029	574.301472	1727.700192	3530.722918	6706.076042	9861.491595	14,726.388476	5348.974389
QKE, ZFeatureMap, aer-simulator, 1000.0	10.898171	46.773357	297.390107	1339.784466	5587.107904	11,607.623344	19,440.97092	5475.79261
VQC, ZZFeatureMap, EfficientSU2, SPSA, aer-simulator	243.897139	463.586231	1368.88324	2789.133991	6570.676821	11,456.705765	17,726.796588	5802.811396
VQC, PauliFeatureMap, EfficientSU2, SPSA, qasm-simulator	348.216144	639.131772	1898.469802	3945.807249	8437.848893	13,815.446673	20,511.034176	7085.136387
QKE, ZFeatureMap, qasm-simulator, 1000.0	11.579451	47.675482	344.543775	1568.903939	6191.830912	14,908.467701	27,002.766078	7153.681048
QKE, ZFeatureMap, aer-simulator, 177.82794100389228	14.163619	56.856619	359.793343	1620.482626	6647.990849	15,084.137764	26,961.326713	7249.250219
QKE, ZFeatureMap, qasm-simulator, 177.82794100389228	16.35717	77.129608	482.478877	2237.68152	9219.954344	18,899.046552	26,623.312487	8222.28008
QKE, ZFeatureMap, qasm-simulator, 0.005623413251903491	16.184459	68.030962	439.889123	2003.14586	8339.157072	18,939.866418	33,875.189822	9097.351959
QKE, PauliFeatureMap, aer-simulator, 31.622776601683793	16.822446	70.391611	549.285996	2267.794391	9148.306499	20,490.131389	36,687.638808	9890.05302
QKE, ZZFeatureMap, aer-simulator, 31.622776601683793	17.382234	70.921393	552.720236	2290.305118	9223.01824	20,681.450668	36,991.554065	9975.335993
QKE, ZZFeatureMap, aer-simulator, 0.1778279410038923	19.618006	80.653612	632.012298	2628.407038	9714.431489	20,666.725844	36,766.378776	10,072.603866
QKE, PauliFeatureMap, aer-simulator, 5.623413251903491	20.03461	81.805468	657.437384	2646.600018	10,751.043722	24,303.410594	42,050.186601	11,501.502628
QKE, ZZFeatureMap, qasm-simulator, 0.001	22.474871	94.5939	748.53639	3061.492908	11,095.037557	24,179.108494	42,833.544061	11,719.255454
QKE, PauliFeatureMap, aer-simulator, 0.1778279410038923	15.70449	64.293166	539.372432	2052.767245	10,360.381735	28,219.134103	53,610.138579	13,551.684536
QKE, PauliFeatureMap, qasm-simulator, 0.1778279410038923	28.777769	121.675706	961.248534	3951.052992	16,159.343561	35,692.431334	48,691.262451	15,086.541764
QKE, ZZFeatureMap, qasm-simulator, 31.622776601683793	28.201021	110.795119	877.141222	3647.413017	16,257.207805	38,819.796065	69,300.661749	18,434.459428

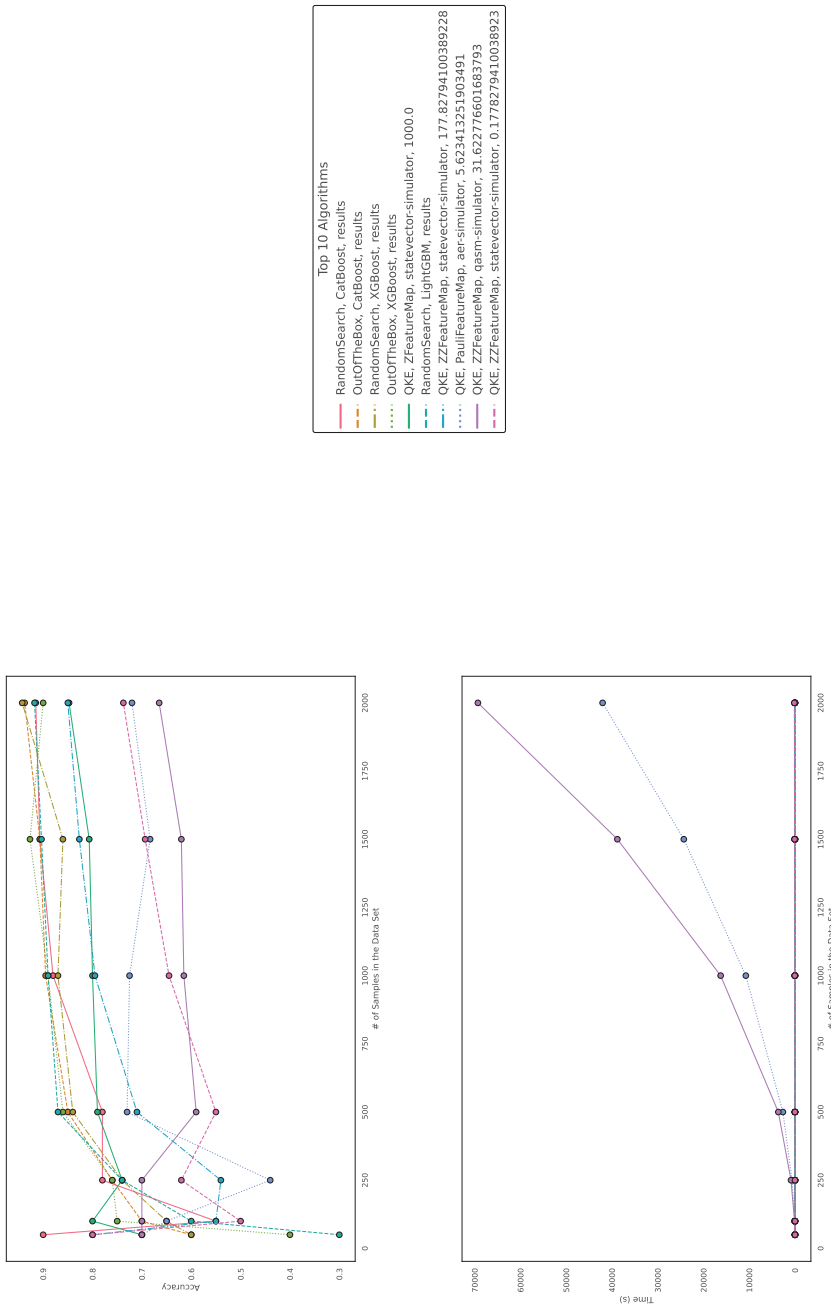


Figure 6. These figures depict the results from our experiments, comparing the five best QML and classical ML algorithms in terms of accuracy on datasets using the exponential map to create SU(2)-transformations on complex vectors. The **upper part** illustrates the accuracy of the algorithms on different sample sizes, while the **lower part** demonstrates how the runtimes change with the increasing size of the test dataset. The **right part** contains the legend, indicating which algorithms were used, and, more specifically, the different parametrizations of the employed quantum machine learning algorithms. Furthermore, the legend is sorted in decreasing order of the average accuracy of the employed algorithms. The parametrization for the QKE is as follows: QKE, feature map, quantum simulator, C-Value for the SVM algorithm.

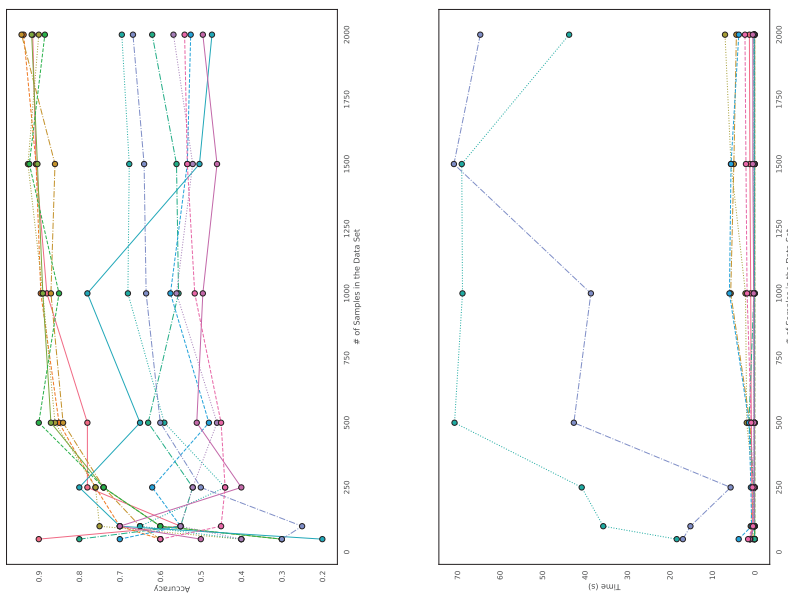


Figure 7. These figures depict the results from our experiments, comparing differently parameterized classical machine learning algorithms on the SU(2)-generated datasets. The upper part illustrates the behavior of the accuracies, while the lower part demonstrates how the run times change with the increasing size of the test dataset. The legend contains the legend, indicating which algorithms were used, and more specifically, the different parametrizations of the employed machine learning algorithms. Furthermore, the legend is sorted in decreasing order of the average accuracy of the employed algorithms.

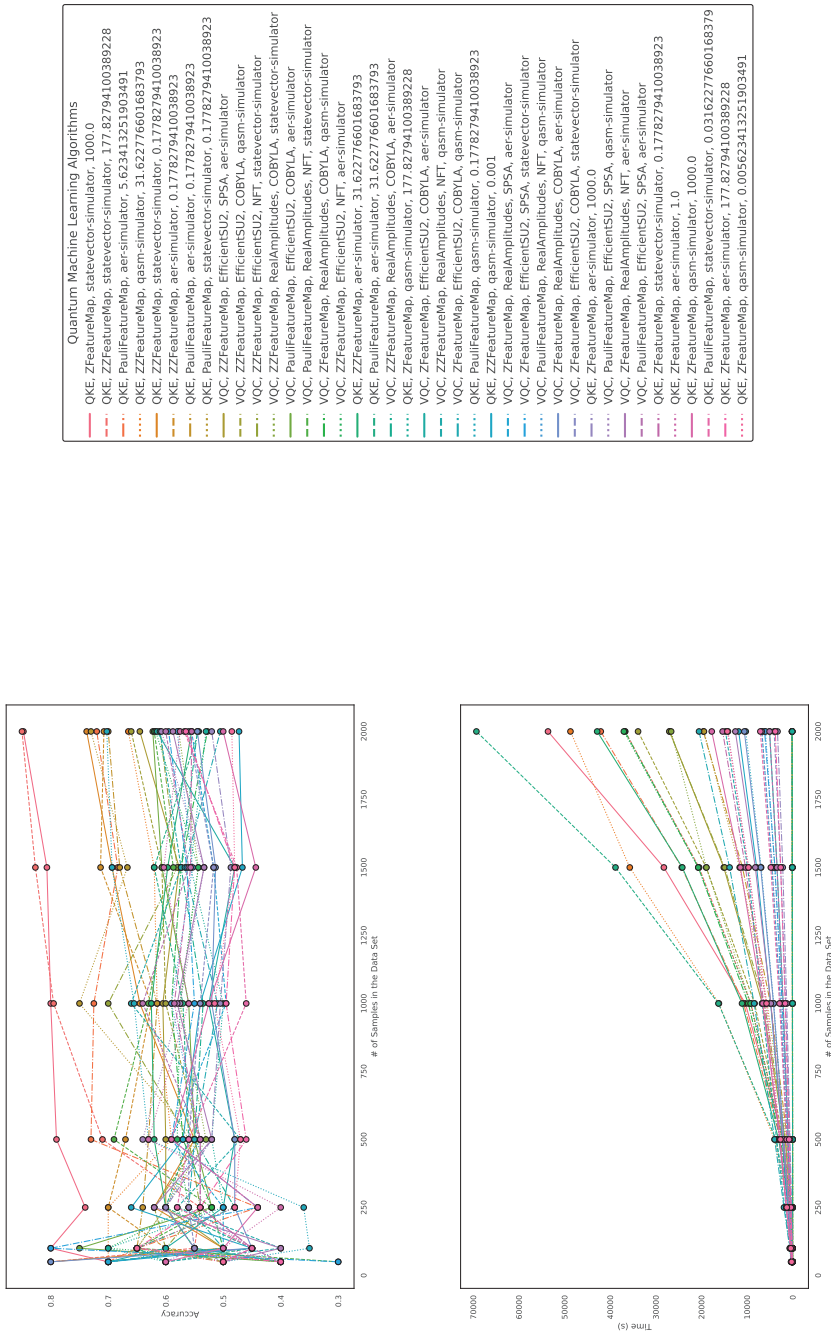


Figure 8. These figures depict the results from our experiments for the artificially generated datasets, comparing differently parameterized QML algorithms on the SU(2)-generated datasets. The **upper part** illustrates the behavior of the accuracies, while the **lower part** demonstrates how the runtimes change with the increasing size of the test datasets. The **right part** contains the legend, indicating which algorithms were used and, more specifically, the different parametrizations of the employed quantum machine learning algorithms. Furthermore, the legend is sorted in decreasing order of the average accuracy of the employed algorithms. The parametrization for the QKE is as follows: QKE, feature map, quantum simulator, C-Value for the SVM algorithm. The parametrization for the VOC is as follows: VOC, feature map, Ansatz, optimizer, quantum simulator.

5.3. Results on Benchmark Datasets

In this section, we discuss the performance of quantum machine learning and classical machine learning algorithms on six benchmark datasets described in Section 3.5. We include results for the quantum classifiers detailed in Section 3.3 and the classical machine learning classifiers discussed in Section 3.2. The scores/accuracies were obtained using randomized search cross-validation from Scikit-learn with 20 models and five-fold cross-validation.

Our results, shown in Table 5, display the best five-fold cross-validation scores (upper table) and the scores of the best model evaluated on an unseen test subset of the original data (lower table), which makes up 20% of the original data. We observe varying performances of the algorithms on these benchmark datasets.

Table 5. These tables present the scores/accuracies of our experiments conducted on publicly available classification datasets. The upper table displays the best five-fold cross-validation scores, obtained using randomized search cross-validation from Scikit-learn, which were employed to identify the optimal model. The lower table shows the scores of the best model evaluated on an unseen test subset of the original data. We include results for the six datasets described in Section 3.5, the quantum classifiers detailed in Section 3.3, and the classical machine learning classifiers discussed in Section 3.2.

Classifier\Dataset	Iris	Wine	ILPD	BC-Coimbra	TAE	Breast-Tissue
VQC	0.817	0.817	0.706	0.599	0.417	0.339
QKE	0.908	0.853	0.706	0.620	0.483	0.382
Ridge	0.914	0.875	0.080	0.053	0.053	<0.001
Lasso	0.914	0.870	0.085	0.004	0.004	<0.001
MLP	0.975	0.937	0.712	0.687	0.425	0.406
SVM	0.958	0.759	0.706	0.630	0.450	0.382
XGBoost	0.958	0.986	0.695	0.656	0.533	0.441
LightGBM	0.967	0.986	0.699	0.666	0.475	0.393
CatBoost	0.950	0.979	0.702	0.688	0.525	0.440

Classifier\Dataset	Iris	Wine	ILPD	BC-Coimbra	TAE	Breast-Tissue
VQC	0.767	0.639	0.744	0.541	0.388	0.334
QKE	1.0	0.833	0.744	0.792	0.613	0.409
Ridge	0.947	0.878	0.115	0.234	<0.001	<0.001
Lasso	0.945	0.882	0.115	0.296	<0.001	<0.001
MLP	1.0	1.0	0.769	0.875	0.387	0.455
SVM	1.0	0.972	0.743	0.875	0.355	0.455
XGBoost	1.0	1.0	0.735	0.917	0.533	0.441
LightGBM	1.0	1.0	0.752	0.917	0.419	0.455
CatBoost	1.0	1.0	0.744	0.917	0.645	0.545

Notably, both the variational quantum circuit and the quantum kernel estimator classifier show competitive performance on several datasets but do not consistently outperform classical ML algorithms. In particular, QKE achieves a perfect score on the Iris dataset, but its performance varies across the other datasets.

Classical ML algorithms, such as multilayer perceptron, support vector machines, XGBoost, LightGBM, and CatBoost, exhibit strong performance across all datasets, with some algorithms achieving perfect scores on multiple datasets. CatBoost consistently performs well, ranking as the top-performing algorithm on three of the six datasets. Ridge

and Lasso regression show high accuracy on Iris and Wine datasets but perform poorly on the others.

When comparing the runtimes of the experiments, as presented in Table 6, it becomes evident that QML algorithms take substantially longer to execute than their classical counterparts. For instance, the VQC and QKE classifiers take hours to days to complete on various datasets, whereas classical ML algorithms such as Ridge, Lasso, MLP, SVM, XGBoost, LightGBM, and CatBoost typically take seconds to minutes.

This significant difference in runtimes could be attributed to the inherent complexity and resource requirements of QML algorithms, which generally demand specialized quantum hardware and simulators. On the other hand, classical ML algorithms are optimized for execution on conventional hardware, making them more efficient and faster to run.

In conclusion, while QML algorithms such as VQC and QKE demonstrate potential in achieving competitive performance on certain datasets, their relatively longer runtimes and less consistent performance across the benchmark datasets may limit their practical applicability compared to classical ML algorithms. Classical ML algorithms, such as CatBoost, XGBoost, and LightGBM, continue to offer superior and more consistent performance with faster execution times, solidifying their place as reliable and powerful tools for classification tasks.

Table 6. This table presents the combined runtimes of our experiments conducted on well-known and publicly available classification datasets. The runtimes include both the five-fold randomized search cross-validation process from Scikit-learn, which was employed to identify the optimal model, and the evaluation of the best model on an unseen test subset of the original data. We include results for the six datasets described in Section 3.5, the quantum classifiers detailed in Section 3.3, and the classical machine learning classifiers discussed in Section 3.2.

Classifier\ Dataset	Iris	Wine	ILPD	BC-Coimbra	TAE	Breast-Tissue
VQC	3:32:16.547605	1 day, 13:56:59.455185	2 days, 23:03:26.398856	9:55:17.907443	2:46:25.921553	9:01:58.623806
QKE	2:03:57.921154	21:41:38.738255	7 days, 6:30:41.179676	5:02:26.430001	1:28:54.069725	3:37:05.655104
Ridge	0:00:00.175009	0:00:00.496771	0:00:00.399229	0:00:00.240857	0:00:00.209600	0:00:00.296966
Lasso	0:00:00.173051	0:00:00.181444	0:00:00.237455	0:00:00.192257	0:00:00.229508	0:00:00.225531
MLP	0:00:16.876288	0:00:10.477420	0:00:26.748907	0:00:10.951229	0:00:08.475263	0:00:13.729790
SVM	0:00:00.143353	0:00:00.165431	0:00:00.484485	0:00:00.180694	0:00:00.228508	0:00:00.226784
XGBoost	0:00:03.809085	0:00:04.030425	0:00:04.752627	0:00:02.744122	0:00:05.820371	0:00:06.864497
LightGBM	0:00:02.971164	0:00:03.180770	0:00:03.062553	0:00:01.462174	0:00:03.056615	0:00:04.540870
CatBoost	0:00:06.465975	0:00:18.511612	0:00:11.352944	0:00:07.460460	0:00:06.964821	0:00:26.639070

5.4. Comparison and Discussion

In this study, we have compared the performance of quantum machine learning and classical machine learning algorithms on six benchmark datasets and two types of artificially generated classification datasets. We included results for quantum classifiers, such as variational quantum circuit and quantum kernel estimator, and classical machine learning classifiers, such as CatBoost, XGBoost, and LightGBM. Our experiments showed that while QML algorithms demonstrate potential in achieving competitive performance on certain datasets, they do not consistently outperform classical ML algorithms. Additionally, their longer runtimes for the whole process, i.e., hyperparameter tuning via randomized search and five-fold cross-validation, the corresponding training and testing, and less consistent performance across the benchmark datasets, may limit their practical applicability compared to classical ML algorithms, which continue to offer superior and more consistent performance with faster execution times. Furthermore, we constructed artificial datasets with the structure and rulings of quantum Mechanics in mind, i.e., we used symmetry properties and unitary transformations to generate a classification dataset from SU(2)-matrices in order to demonstrate an advantage of quantum machine learning algorithms to tackle

problems with an inherent structure relatable to that of quantum circuits and quantum mechanics overall. However, also for these datasets, the employed quantum machine learning algorithms performed reasonably but did not outperform sophisticated boost classifiers. Thus, we cannot conclude a quantum advantage for these datasets.

It is essential to highlight that the QML algorithms' performance in our experiments was based on simulated quantum infrastructures. This is a significant limitation to consider, as the specific constraints and characteristics of the simulated hardware may influence the performance of these algorithms. Furthermore, given the rapid advancement of quantum technologies and hardware, this constraint might be obsolete in the near future.

The impact of quantum simulators, feature maps, and quantum circuits on the performance of quantum estimators stems from the fact that these components play crucial roles in shaping the behavior and capabilities of quantum machine learning algorithms. Quantum simulators, which emulate quantum systems on classical computers, introduce various levels of approximation and noise, leading to deviations from ideal quantum behavior. Different simulators may employ distinct algorithms and techniques, resulting in variations in performance.

Feature maps, responsible for encoding classical data into quantum states, determine how effectively the quantum system can capture and process information. The choice of feature map can greatly influence the ability of quantum algorithms to extract meaningful features and represent the data in a quantum-mechanical space.

Similarly, quantum circuits, composed of quantum gates and operations, define the computational steps performed on the encoded data. Different circuit designs and configurations can affect the expressiveness and depth of the quantum computation, potentially impacting the accuracy and efficiency of the quantum estimators.

Considering the diverse options for quantum simulators, feature maps, and quantum circuits, it becomes essential for researchers to provide detailed explanations of their hyperparameter choices. This entails clarifying the rationale behind selecting a specific simulator, feature map, or circuit design, as well as the associated parameters and their values. By providing such explanations, researchers can enhance the reproducibility and comparability of results, enabling the scientific community to better understand the strengths and limitations of different quantum machine learning algorithms.

Unfortunately, the current state of the field often overlooks the thorough discussion of hyperparameter choices in many studies. This omission restricts the transparency and interpretability of research outcomes and hinders the advancement of quantum machine learning. To address this issue, researchers should embrace a culture of providing comprehensive documentation regarding hyperparameter selection, sharing insights into the decision-making process, and discussing the potential implications of different choices.

By encouraging researchers to provide detailed explanations of hyperparameter choices and corresponding code, we can foster a more robust and transparent research environment in quantum machine learning. This approach enables the replication and comparison of results, promotes knowledge sharing, and ultimately contributes to the development of reliable and effective quantum machine learning algorithms. Additionally, our program code serves as introductory material, providing easy-to-use implementations and a foundation for comparing quantum machine learning and classical machine learning (CML) algorithms.

One possible direction for future research is exploring quantum ensemble classifiers and, consequently, quantum boosting classifiers, as suggested by Schuld et al. [40]. This approach might help in improving the capabilities of QML algorithms and make them more competitive with state-of-the-art classical ML algorithms in terms of high accuracies.

Finally, the relatively lower performance of the employed quantum machine learning algorithms compared to, for example, the employed boosting classifiers might be attributed to quantum machine learning, being constrained by specific rules of quantum mechanics.

In the authors' opinion, quantum machine learning might be constrained by the unitary transformations inherent in, for example, the variational quantum circuits. These

transformations are part of the unitary group $U(n)$. Thus, all transformations are constrained by symmetry properties. Classical machine learning models are not constrained by these limitations, meaning that, for instance, different activation functions in neural networks do not preserve certain distance metrics or probabilities when processing data. However, expanding the set of transformations of quantum machine learning and getting rid of possible constraints might improve the capabilities of quantum machine learning models such that these algorithms might be better capable of capturing the information of more complex data. However, this needs to be discussed in the context of quantum computers such that one determines what all possible transformations on a quantum computer are. This means that future research needs to consider the applicability of advanced mathematical frameworks for quantum machine learning regarding the formal requirements of quantum computers.

Furthermore, another constraint of quantum machine learning is that it, and quantum mechanics in general, relies on Hermitian matrices, e.g., to provide real-valued eigenvalues of observables. However, breaking this constraint might be another way to broaden the capabilities of quantum machine learning to better capture complexity, e.g., by using non-Hermitian kernels in a quantum kernel estimator. Here, we want to mention the book by Moiseyev [41], which introduces non-Hermitian quantum mechanics. Furthermore, quantum computers, in general, might provide a testing ground for non-Hermitian quantum mechanics in comparison to Hermitian quantum mechanics. However, at this point, this is rather speculative, but given that natural data are nearly always corrupted by noise and symmetries are never truly perfect in nature, breaking constraints and symmetries might be ideas to expand the capabilities of QML.

6. Conclusions

In this research, we have explored the applicability of quantum machine learning for classification tasks by examining the performance of variational quantum circuit and quantum kernel estimator algorithms. Our comparison of these quantum classifiers with classical machine learning algorithms, such as XGBoost, Ridge, Lasso, LightGBM, CatBoost, and MLP, on six benchmark datasets and artificially generated classification datasets demonstrated that QML algorithms can achieve competitive performance on certain datasets. However, they do not consistently outperform their classical ML counterparts, particularly with regard to runtime performance and accuracy. Quite the contrary, classical machine learning algorithms still demonstrate superior performance, especially in terms of increased accuracy, in most of our experiments. Furthermore, we cannot conclude a quantum advantage even for artificial data built by data manipulations inherent to quantum mechanics.

As our study's performance comparison relied on simulated quantum circuits, it is important to consider the limitations and characteristics of simulated hardware, which may affect the true potential of quantum machine learning. Given the rapid advancement of quantum technologies and hardware, these constraints may become less relevant in the future.

Quantum simulators, feature maps, and quantum circuits significantly influence quantum estimator performance; hence, a detailed discussion of the chosen hyperparameters is essential. The absence of such a discussion in current research limits the interpretation and replication of experiments. Thus, we aim to encourage transparency in decision-making processes to promote a robust research environment, aiding in knowledge sharing and the creation of reliable quantum machine learning algorithms.

Despite the current limitations, this study has shed light on the potential and challenges of quantum machine learning compared to classical approaches. Thus, by providing our complete code in a GitHub repository, we hope to foster transparency, encourage further research in this field, and offer a foundation for other researchers to build upon as they explore the world of quantum machine learning. Furthermore, the developed SU(2)-data creation might serve as a quantum data prototype for future experiments, and

both quantum and regular machine learning algorithms can be tested for their accuracy on datasets like these.

Future research should also consider exploring quantum ensemble classifiers and quantum boosting classifiers, as well as addressing the limitations imposed by the specific rules of quantum mechanics. By breaking constraints and symmetries and expanding the set of transformations in quantum machine learning, researchers may be able to unlock its full potential.

Author Contributions: Conceptualization, S.R. and K.M.; Methodology, S.R. and K.M.; Software, S.R. and K.M.; Validation, S.R. and K.M.; Formal analysis, S.R. and K.M.; Investigation, S.R. and K.M.; Resources, K.M.; Data curation, S.R. and K.M.; Writing—original draft, S.R. and Kevin Mallinger; Writing—review & editing, S.R. and K.M.; Visualization, S.R. and K.M. All authors have read and agreed to the published version of the manuscript.

Funding: Open Access Funding by TU Wien.

Acknowledgments: The authors acknowledge the funding by the TU Wien Bibliothek for financial support through its Open Access Funding Program.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Parametrization

This Appendix lists the parameter grids for all employed algorithms per the implementations from Scikit-learn and Qiskit [16,24]. Thus, for further explanations on the parameters and how they influence the discussed algorithm, the reader is referred to the respective sources, which we linked in Sections 3.2 and 3.3.

Appendix A.1. Ridge

```
param_grid = {
'alpha': [0.001, 0.01, 0.1, 1, 10, 100],
'fit_intercept': [True, False],
'normalize': [True, False],
'copy_X': [True, False],
'max_iter': [100, 500, 1000],
'tol': [1e-4, 1e-3, 1e-2],
'solver': ['auto', 'svd', 'cholesky', 'lsqr', 'sparse_cg', 'sag', 'saga'],
'random_state': [42]
}
```

Appendix A.2. Lasso

```
param_grid = {
'alpha': [0.001, 0.01, 0.1, 1, 10, 100],
'fit_intercept': [True, False],
'normalize': [True, False],
'precompute': [True, False],
'copy_X': [True, False],
'max_iter': [100, 500, 1000],
'tol': [1e-4, 1e-3, 1e-2],
'warm_start': [True, False],
'positive': [True, False],
'random_state': [42],
'selection': ['cyclic', 'random']
}
```

Appendix A.3. SVM

```
param_grid = {
  'C': [0.1, 1, 10, 100],
  'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
  'degree': [2, 3, 4],
  'gamma': ['scale', 'auto'],
  'coef0': [0.0, 1.0, 2.0],
  'shrinking': [True, False],
  'probability': [False],
  'tol': [1e-4, 1e-3, 1e-2],
  'cache_size': [200],
  'class_weight': [None, 'balanced'],
  'verbose': [False],
  'max_iter': [200, 300, 400],
  'decision_function_shape': ['ovr', 'ovo'],
  'break_ties': [False],
  'random_state': [42]
}
```

Appendix A.4. MLP

```
param_grid = {
  'hidden_layer_sizes': [(50,), (100,), (150,)],
  'activation': ['relu', 'tanh'],
  'solver': ['adam', 'sgd'],
  'alpha': [0.0001, 0.001, 0.01],
  'learning_rate': ['constant', 'invscaling', 'adaptive'],
  'max_iter': [200, 300, 400]
}
```

Appendix A.5. XGBoost

```
param_grid = {
  'max_depth': [3, 5, 7, 10],
  'learning_rate': [0.01, 0.05, 0.1, 0.2],
  'n_estimators': [50, 100, 150, 200],
  'subsample': [0.5, 0.8, 1],
  'colsample_bytree': [0.5, 0.8, 1]
}
```

Appendix A.6. LightGBM

```
param_grid = {
  'max_depth': [3, 5, 7, 10],
  'learning_rate': [0.01, 0.05, 0.1, 0.2],
  'n_estimators': [50, 100, 150, 200],
  'subsample': [0.5, 0.8, 1],
  'colsample_bytree': [0.5, 0.8, 1]
}
```

Appendix A.7. CatBoost

```
param_grid = {
  'iterations': [50, 100, 150, 200],
  'learning_rate': [0.01, 0.05, 0.1, 0.2],
  'depth': [3, 5, 7, 10],
  'l2_leaf_reg': [1, 3, 5, 7, 9],
}
```

Appendix A.8. QKE

For this Algorithm, we precomputed the kernel matrix using Qiskit and then performed the support vector classification via the vanilla SVM algorithm from Scikit-learn.

```
param_grid = {
  'feature_map': [PauliFeatureMap, ZFeatureMap, ZZFeatureMap],
  'quantum_instance': [
    QuantumInstance(Aer.get_backend('aer_simulator'), shots=1024),
    QuantumInstance(Aer.get_backend('qasm_simulator'), shots=1024),
    QuantumInstance(Aer.get_backend('statevector_simulator'), shots=1024)
  ],
  'C' : np.logspace(-3, 3, 9),
}
```

Appendix A.9. VQC

```
param_grid = {
  'feature_map': [PauliFeatureMap, ZFeatureMap, ZZFeatureMap],
  'ansatz': [EfficientSU2, TwoLocal, RealAmplitudes],
  'optimizer': [
    COBYLA(maxiter=max_iter),
    SPSA(maxiter=max_iter),
    NFT(maxiter=max_iter),
  ],
  'quantum_instance': [
    QuantumInstance(Aer.get_backend('aer_simulator'), shots=1024),
    QuantumInstance(Aer.get_backend('qasm_simulator'), shots=1024),
    QuantumInstance(Aer.get_backend('statevector_simulator'), shots=1024)
  ],
}
```

References

- Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed.; Cambridge University Press: Cambridge, MA, USA, 2011.
- Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [CrossRef] [PubMed]
- Schuld, M.; Sinayskiy, I.; Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **2015**, *56*, 172–185.
- Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. [CrossRef] [PubMed]
- Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
- Griol-Barres, I.; Milla, S.; Cebrián, A.; Mansoori, Y.; Millet, J. Variational Quantum Circuits for Machine Learning. An Application for the Detection of Weak Signals. *Appl. Sci.* **2021**, *11*, 6427. [CrossRef]
- Kuppusamy, P.; Yaswanth Kumar, N.; Dontireddy, J.; Iwendi, C. Quantum Computing and Quantum Machine Learning Classification – A Survey. In Proceedings of the 2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), Goa, India, 8–9 October 2022; pp. 200–204. [CrossRef]
- Blance, A.; Spannowsky, M. Quantum machine learning for particle physics using a variational quantum classifier. *J. High Energy Phys.* **2021**, *2021*, 212. [CrossRef]
- Abohashima, Z.; Elhoseny, M.; Houssein, E.H.; Mohamed, W.M. Classification with Quantum Machine Learning: A Survey. *arXiv* **2020**, arXiv:2006.12270.
- Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794. [CrossRef]
- Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67. [CrossRef]
- Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **1996**, *58*, 267–288. [CrossRef]
- Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 3149–3157.

14. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased Boosting with Categorical Features. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Montréal, ON, Canada December, 2–8 December 2018; Curran Associates Inc.: Red Hook, NY, USA, 2018; pp. 6639–6649.
15. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. *Learning Internal Representations by Error Propagation*; Technical report; California Univ San Diego La Jolla Inst for Cognitive Science: San Diego, CA, USA, 1985.
16. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 282–290.
17. Raubitzek, S. Quantum_Machine_Learning. *Preprints* **2023**, 2023050833. [CrossRef]
18. Zeguendry, A.; Jarir, Z.; Quafafou, M. Quantum Machine Learning: A Review and Case Studies. *Entropy* **2023**, *25*, 287. [CrossRef]
19. Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 032309. [CrossRef]
20. Farhi, E.; Neven, H. Classification with quantum neural networks on near term processors. *arXiv* **2018**, arXiv:1802.06002.
21. Reberntrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef] [PubMed]
22. Liu, D.; Reberntrost, P. Quantum machine learning for quantum anomaly detection. *Phys. Rev. A* **2019**, *100*, 042328. [CrossRef]
23. Broughton, M.; Verdon, G.; McCourt, T.; Martinez, A.J.; Yoo, J.H.; Isakov, S.V.; King, A.D.; Smelyanskiy, V.N.; Neven, H. TensorFlow Quantum: A Software Framework for Quantum Machine Learning. *arXiv* **2020**, arXiv:2003.02989.
24. Qiskit Contributors. Qiskit: An Open-source Framework for Quantum Computing. *Zenodo* **2023**. [CrossRef]
25. Bishop, C.M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*; Springer: Berlin/Heidelberg, Germany, 2006.
26. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; MIT Press: Cambridge, MA, USA, 2013.
27. Kotsiantis, S.B. Supervised Machine Learning: A Review of Classification Techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies*; IOS Press: Amsterdam, The Netherlands, 2007; pp. 3–24.
28. Refaeilzadeh, P.; Tang, L.; Liu, H. Cross-Validation. In *Encyclopedia of Database Systems*; Liu, L., ÖzSU, M.T., Eds.; Springer: Boston, MA, USA, 2009; pp. 532–538. [CrossRef]
29. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
30. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
31. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [CrossRef]
32. Ramana, B.V.; Babu, M.S.P.; Venkateswarlu, N.B. LPD (Indian Liver Patient Dataset) Data Set. 2012. Available online: [https://archive.ics.uci.edu/ml/datasets/ILPD+\(Indian+Liver+Patient+Dataset\)](https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset)) (accessed on 25 June 2023).
33. Patrício, M.; Pereira, J.; Crisóstomo, J.; Matafome, P.; Gomes, M.; Seïça, R.; Caramelo, F. Using Resistin, glucose, age and BMI to predict the presence of breast cancer. *BMC Cancer* **2018**, *18*, 29. [CrossRef] [PubMed]
34. Crisóstomo, J.; Matafome, P.; Santos-Silva, D.; Gomes, A.L.; Gomes, M.; Patrício, M.; Letra, L.; Sarmiento-Ribeiro, A.B.; Santos, L.; Seïça, R. Hyperresistinemia and metabolic dysregulation: A risky crosstalk in obese breast cancer. *Endocrine* **2016**, *53*, 433–442. [CrossRef] [PubMed]
35. Loh, W.Y.; Shih, Y.S. Split selection methods for classification trees. *Stat. Sin.* **1997**, *7*, 815–840.
36. Lim, T.S.; Loh, W.Y.; Shih, Y.S. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Mach. Learn.* **2000**, *40*, 203–228. [CrossRef]
37. Marques de Sá, J.; Jossinet, J. Breast Tissue Impedance Data Set. 2002. Available online: <https://archive.ics.uci.edu/ml/datasets/Breast+Tissue> (accessed on 25 June 2023).
38. Estrela da Silva, J.; Marques de Sá, J.P.; Jossinet, J. Classification of breast tissue by electrical impedance spectroscopy. *Med. Biol. Eng. Comput.* **2000**, *38*, 26–30. [CrossRef] [PubMed]
39. Georgi, H. *Lie Algebras in Particle Physics: From Isospin to Unified Theories*; CRC Press: Boca Raton, FL, USA, 2019. [CrossRef]
40. Schuld, M.; Petruccione, F. Quantum ensembles of quantum classifiers. *Sci. Rep.* **2018**, *8*, 2772. [CrossRef]
41. Moiseyev, N. *Non-Hermitian Quantum Mechanics*; Cambridge University Press: Cambridge, UK, 2011. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Quantum Image Encryption Based on Quantum DNA Codec and Pixel-Level Scrambling

Jie Gao ¹, YINUO Wang ², Zhaoyang Song ¹ and Shumei Wang ^{2,*}¹ School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266520, China² School of Science, Qingdao University of Technology, Qingdao 266520, China

* Correspondence: wangshumei@qut.edu.cn

Abstract: In order to increase the security and robustness of quantum images, this study combined the quantum DNA codec with quantum Hilbert scrambling to offer an enhanced quantum image encryption technique. Initially, to accomplish pixel-level diffusion and create enough key space for the picture, a quantum DNA codec was created to encode and decode the pixel color information of the quantum image using its special biological properties. Second, we used quantum Hilbert scrambling to muddle the image position data in order to double the encryption effect. In order to enhance the encryption effect, the altered picture was then employed as a key matrix in a quantum XOR operation with the original image. The inverse transformation of the encryption procedure may be used to decrypt the picture since all the quantum operations employed in this research are reversible. The two-dimensional optical image encryption technique presented in this study may significantly strengthen the anti-attack of quantum picture, according to experimental simulation and result analysis. The correlation chart demonstrates that the average information entropy of the RGB three channels is more than 7.999, the average NPCR and UACI are respectively 99.61% and 33.42%, and the peak value of the ciphertext picture histogram is uniform. It offers more security and robustness than earlier algorithms and can withstand statistical analysis and differential assaults.

Keywords: quantum information; DNA coding; quantum image encryption; quantum image scrambling

Citation: Gao, J.; Wang, Y.; Song, Z.; Wang, S. Quantum Image Encryption Based on Quantum DNA Codec and Pixel-Level Scrambling. *Entropy* **2023**, *25*, 865. <https://doi.org/10.3390/e25060865>

Academic Editors: Rosario Lo Franco, Giuliano Benenti and Brian R. La Cour

Received: 16 January 2023

Revised: 14 April 2023

Accepted: 22 May 2023

Published: 29 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the development of internet and communication technology, image has become the most widely used information transmission medium. Compared with text information, images contain more information. As a result, researchers suggest quantum image processing by extending the digital picture to the quantum computing framework [1,2].

Quantum image processing (QIP) is committed to using quantum computing technology to capture, restoration, and other classical image operations. its exponential storage capacity and parallelism give this technology a strong advantage in implementing operations requiring high real-time operations such as image retrieval and processing.

The special behavior of quantum particles is regarded as the rules of quantum physics and the tool of mathematical logic. In 1992, Lucien Hardy proposed Hardy's paradox while proving Bell's theorem [3], and continued to prove the nonlocality of two particles in 1993 [4]. Moreover, Shor [5] and Grover [6] created quantum algorithms using the same quantum computer property for integer factoring and database searching, respectively. These algorithms perform better than their traditional versions in terms of running time. References [5,6], which laid the foundation for diverse applications of quantum computing in the information sciences, served as an inspiration for a great number of researchers [7–16].

Quantum image encryption can be “unconditionally secure” based on the Heisenberg uncertainty principle because of quantum features such as quantum entanglement, coherence, parallelism, and superposition. Quantum picture encryption employs the “No-Cloning Theorem”, derived from the Heisenberg uncertainty principle, to encrypt image

data, whereas conventional encryption typically restricts the timeliness of decryption operations. That is, because the basis of replication is measurement, and because measurement often modifies the quantum state, it is impossible to accomplish the process of accurate duplication of any unknown quantum state in quantum mechanics.

To establish image protection in the sphere of digital pictures [17], the genuine and meaningful images are often transformed into meaningless forms. Today’s latest research hot topic is the quantum picture encryption technique created by fusing together quantum computing and digital imaging [18–20]. There are several quantum image representation techniques now being used [21], including FRQI [22], NEQR [23], MCQI [24], NASS [25,26], QUALPI [27], and others. Image information encryption has caught the attention of academics working in the area of quantum information processing. Recently, several quantum image encryption methods have been created, including a quantum image encryption scheme based on quantum image decomposition [28], an iterative extended Arnold transform-based quantum image encryption method, and a quantum image cyclic shift operation-based quantum image encryption strategy [29].

DNA coding has attracted wide attention because of its advantages such as large storage capacity and strong parallel processing ability. Compared with the traditional cryptography based on mathematical problems, DNA cryptography combines the two fields of mathematics and biology, which greatly enhances the security and robustness of DNA cryptography. In 1994, Adleman carried out the world’s first DNA computing experiment [30] and published related results in the journal *Science*. This result revealed that DNA molecules have computing power in addition to their stable genetic properties, and have since opened up a new information age [31–33]. At present, DNA coding is also gradually emerging in the field of encryption [34–37]. Scholars have proposed a classical image encryption algorithm that combines DNA coding technology with quantum walking [38].

In this research, a DNA coding technique and picture Hilbert scrambling were combined to develop a quantum image encryption scheme. The encryption technique uses Hilbert quantum image scrambling and quantum picture DNA coding and decoding. By closely integrating the two technologies, the goal of enhancing picture security may be achieved by more effectively reducing the high connection between neighboring pixels. We also developed the quantum DNA codec’s implementation circuit.

The rest of this article is structured as follows: Section 2 introduces the background. Section 3 shows the quantum circuit model. In Section 4, the flow of encryption and decryption algorithm is shown in detail. Section 5 introduces the theoretical analysis and experimental simulation. Finally, Section 6 draws a conclusion.

2. Related Work

2.1. Quantum Color Image Representation

NCQI is a quantum color digital image representation method proposed in 2017 [39]. We briefly reviewed the NCQI quantum representation model so as to introduce the quantum image encryption algorithm proposed in this paper.

The NCQI model of a $2^n \times 2^n$ image $|\psi\rangle$ can be mathematically expressed as follows:

$$|\psi\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} |c_{(y,x)}\rangle \otimes |yx\rangle, \tag{1}$$

where $|c_{(y,x)}\rangle$ represents the color value of the pixel, which can be encoded by binary sequence $R_{q-1} \cdots R_0 G_{q-1} \cdots G_0 B_{q-1} \cdots B_0$.

Every pixel contained in a color channel, which has a range of $[0, 2^q - 1]$, is represented by three components: the horizontal position X, the vertical position Y, and the color information $c_{(y,x)}$. The R, G, and B range $[0, 2^q - 1]$ of each channel is utilized to store picture data in an NCQI state of a color image using the $2n + 3q$ qubits.

Figure 1 is an example of a 4-by-4-color picture with the three channels, R, G, and B, with the range size $[0, 2^8 - 1]$, $n = 1$, and $q = 8$. The equation in Figure 1 states that

the whole NCQI is kept in a state of normalized quantum superposition, with each base standing in for a single pixel.

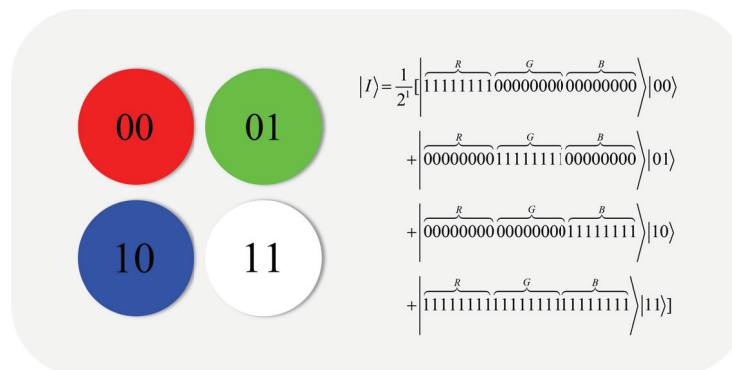


Figure 1. A color image and its quantum representation of NCQI.

2.2. DNA Coding Method and Operation

Adenine (A), thymine (T), cytosine (C), and guanine (G) are the four nucleotides that make up the molecular structure of deoxyribonucleic acid (DNA), which is based on the biological model (G). The DNA pairing rule states that A and T pair and C and G pair. Similarly, in the binary complementary calculation, 1 and 0 are complementary, and eight coding schemes which accord with the rules of a biological model were obtained by encoding each nucleic acid base with 2-bit binary number respectively, as shown in Table 1. Each RGB three-channel pixel in a color image is represented as a 24-bit binary sequence as part of the encryption procedure, where each color channel’s 8-bit binary sequence may be represented by four bases. For instance, scheme 1 will produce CACT if an image pixel’s R-channel gray value is 71, which is represented by the binary sequence 01000111.

Table 1. DNA coding rules.

	1	2	3	4	5	6	7	8
00	A	A	C	C	G	G	T	T
01	C	G	A	T	A	T	C	G
10	G	C	T	A	T	A	G	C
11	T	T	G	G	C	C	A	A

2.3. Quantum Hilbert Scrambling

With the development of quantum image processing, many image scrambling methods have emerged [40,41]. In this work, quantum Hilbert image scrambling [42] was used.

An original image of size $2^n \times 2^n$ can be regarded as a matrix. We call this matrix the starting matrix (or original matrix) S_n and used 1 to 2^{2n} to encode all pixels,

$$S_n = \begin{pmatrix} 1 & 2 & 3 & \dots & 2^n \\ 2^n + 1 & 2^n + 2 & 2^n + 3 & \dots & 2^{n+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 2^{2n-1} + 1 & 2^{2n-1} + 2 & 2^{2n-1} + 3 & \dots & 2^{2n} \end{pmatrix} \tag{2}$$

The arrangement of S_n is H_n . For example, $H_0 = (1)$, $H_1 = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$, $H_2 = \begin{pmatrix} 1 & 2 & 15 & 16 \\ 4 & 3 & 14 & 13 \\ 5 & 8 & 9 & 12 \\ 6 & 7 & 10 & 11 \end{pmatrix}$, where $H_n(i, j)$ represents the pixel at position (i, j) of the starting matrix S_n . Hilbert curves (see Figure 2) and scrambled images (see Figure 3) can be obtained along the H_n .

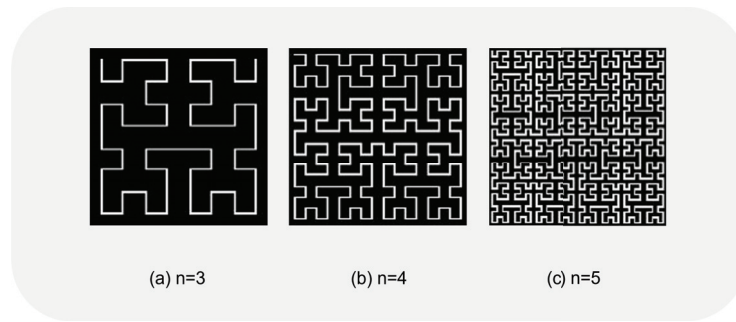


Figure 2. Hilbert curve.

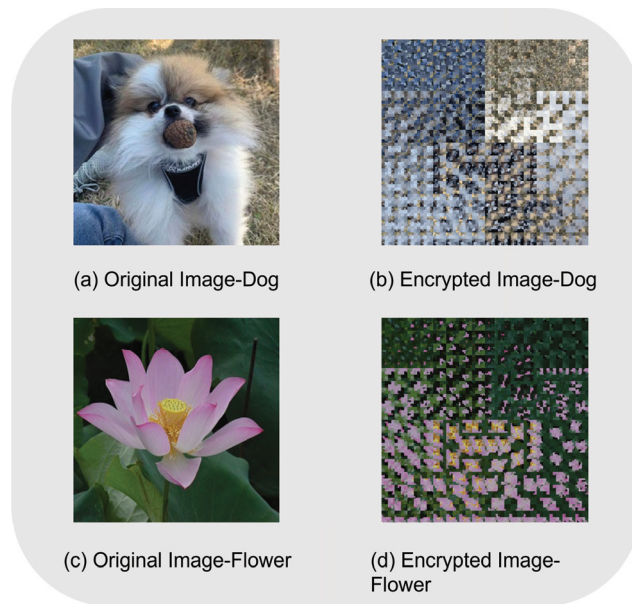


Figure 3. Results of performing a single Hilbert image scrambling.

This paper adopted the improved Hilbert scrambling recursive generation algorithm in [42]. If A is a matrix, then A^T represents its transposition, A^{ud} its upper and lower direction reversed, A^{ld} its left and right inversion, and A^{PP} its centre rotation matrix. For a quantum computer to implement Hilbert image scrambling,

$$H_{n+1} = \begin{cases} \begin{pmatrix} H_n & (H_n + 4^n E_n)^T \\ (H_n + 3 \times 4^n E_n)^{PP} & (H_n + 2 \times 4^n E_n)^T \end{pmatrix}, n \text{ is even} \\ \begin{pmatrix} H_n & (H_n + 3 \times 4^n E_n)^{PP} \\ (H_n + 4^n E_n)^T & (H_n + 2 \times 4^n E_n)^T \end{pmatrix}, n \text{ is odd} \end{cases}, \quad (3)$$

where n is a positive integer and the initial matrix is $H_1 = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$, $E_n = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$.

According to reference [42], initialization, and even and odd basic circuits, are also integrated circuits, and the process is described in Figure 4. The three parts that make up the three basic circuits are called three circuit modules.

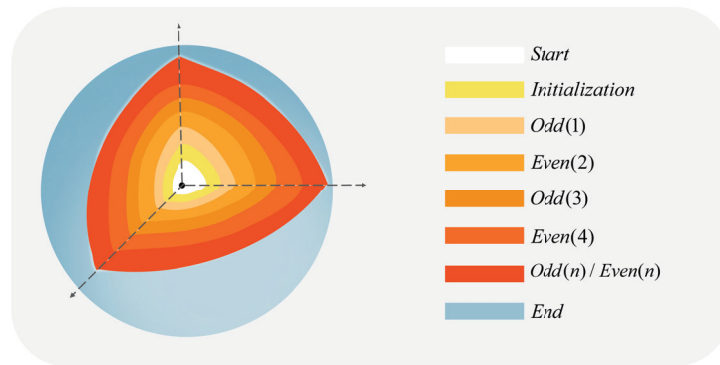


Figure 4. Flow chart of recursive generation algorithm.

2.4. Quantum XOR

According to reference [43], it needs to be divided into 2^{2^n} sub-operations $Y_{y,x}$ in order to implement the XOR operation on each pixel value of the quantum image. Use Y to represent a matrix of the same size as the image in the sub-operation:

$$Y = \begin{pmatrix} y_{0,1} & \cdots & y_{0,2^n-1} \\ \vdots & \ddots & \vdots \\ y_{2^n-1,0} & \cdots & y_{2^n-1,2^n-1} \end{pmatrix}, \tag{4}$$

where $y_{y,x} = m_{yx}^0 m_{yx}^1 m_{yx}^2 m_{yx}^3 \cdots m_{yx}^{2^3} m_{yx}^i \in 0, 1$, and the quantum gate operation sequence F is generated according to $y_{i,j}$:

$$F = \begin{pmatrix} b_{0,1} & \cdots & b_{0,2^n-1} \\ \vdots & \ddots & \vdots \\ b_{2^n-1,0} & \cdots & b_{2^n-1,2^n-1} \end{pmatrix}, \tag{5}$$

where $F_{yx} = V_{yx}^0 V_{yx}^1 \cdots V_{yx}^{2^3} V_{yx}^i$, $V_{yx}^i = \begin{cases} X, m_{yx}^i = 1 \\ I, m_{yx}^i = 0 \end{cases}$ represents the realization of the X-gate transformation or I-gate transformation of C_{yx}^i , respectively:

$$G_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, G_I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{6}$$

When F is applied to the entire image, there are:

$$\begin{aligned} F|I\rangle &= \prod_{x=0}^{2^n-1} \prod_{y=0}^{2^n-1} F_{yx}|I\rangle \\ &= \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} \otimes_{i=0}^{2^3} |C_{yx}^i \otimes m_{yx}^i\rangle |yx\rangle \\ &= \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} |f(y,x)\rangle |yx\rangle, \end{aligned} \tag{7}$$

where $|f(Y,X)\rangle$ represents the new pixel value after pixel scrambling, and $|C_{YX}\rangle$ is the pixel sequence.

3. Quantum Circuit Design

The design of the DNA codec simulator’s quantum circuit, which is a crucial component of our quantum picture encryption technique, is presented in this section.

3.1. Quantum DNA Codec Simulator

Based on the classical DNA coding technology and the quantum image representation of NCQI, a DNA codec simulation circuit for quantum images was designed.

In our proposed encryption algorithm, quantum DNA coding and decoding technology were used to encrypt pixel color information. NCQI representation can directly transform the color value information of a color image with three-channel color values in the range of $[0, 2^q - 1]$ of $2^n \times 2^n$ into a binary sequence of 3q bits, so we took every two lines as a combination to reflect the concept of bases in biology.

We encapsulated the whole quantum DNA codec into a black box. We only needed to input the binary sequence of the image into the black box and enter the sequence number of the coding and decoding scheme to complete the DNA codec operation of the quantum image. $D_{i,j}$ was used to represent the quantum DNA codec, where i is the coding scheme sequence number and j is the decoding scheme sequence number. As shown in Figure 5, six lines were added to reflect the sequence number of the codec scheme, and $|\psi_0\rangle$ and $|\psi_1\rangle$ were input lines as binary sequences. Three quantum lines b_0, b_1, b_2 can represent the numerical value of the decoding scheme sequence number. When designing quantum circuits, we used two auxiliary circuits. While reducing a lot of time complexity, we only added a little space complexity. Therefore, it can effectively reduce the circuit complexity and improve the operation efficiency. If we decode it with option 6, the circuit module of the scheme will be run directly through the three quantum lines b_0, b_1, b_2 . As a result, only the sequence number of the decoding scheme can be input to run the circuit to realize automatic decoding, and there is no need to select the circuit. Figure 6 shows the quantum circuits of seven decoding schemes encoded in Rule 1.

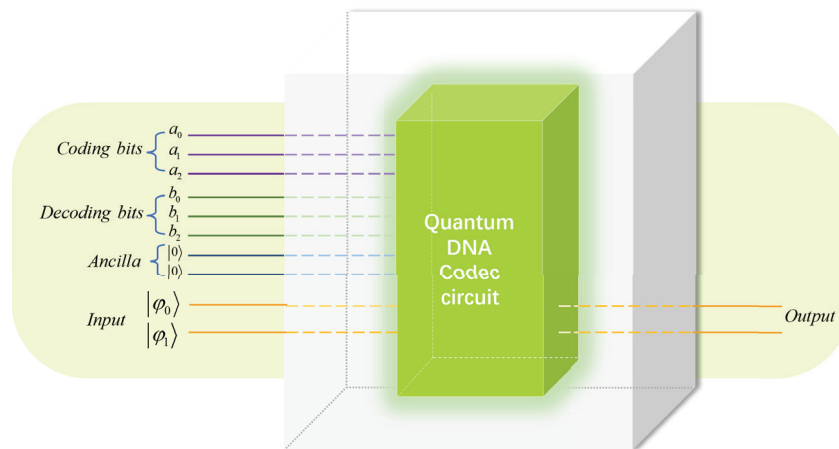


Figure 5. QuantumDNA codec analog circuit.

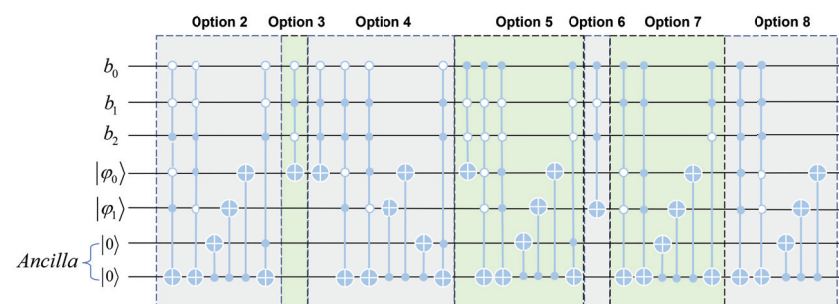


Figure 6. Seven kinds of quantum DNA codec simulators encoded by Rule 1.

In this paper, we designed the quantum circuits of seven kinds of decoders with scheme one as the coding scheme, and showed the process of transforming the same binary sequence from scheme one to the other seven schemes. As shown in Table 1, the sequence was first quantum DNA encoded according to scheme 1. If the second scheme is used for

decoding, it is necessary to reverse the two lines when the high qubits are different from the low qubits, that is, to realize the interchange between C and G. If the third scheme is used for decoding, it is necessary to reverse all the low qubits, that is, to realize the interchange between A and C and between T and G. If we use scheme 4 to decode, we need to reverse the high qubit and the low qubit at the same time and, if the high qubit is different from the low qubit, we need to flip the low qubit. If we use scheme 5 to decode, contrary to scheme 4, we need to flip high qubits when high qubits are different from low qubits, and if high qubits and low qubits flip low qubits at the same time. If decoding is carried out in scheme 6, each set of high qubits needs to be flipped. If we use scheme 7 to decode, when the high qubit and the low qubit are all flipped, the interchange between A and T will be realized. If we use scheme 8 to decode, it is necessary to reverse all the high and low qubits, that is, to realize the interchange between A and T and between C and G.

3.2. Hilbert Image Scrambling Quantum Circuit

The Hilbert scrambling operation of quantum image was divided into three steps: quantum image partition and Hilbert image scrambling parity operation, in which the parity operation is carried out alternately. The composition of these three basic circuits is described below. In this section, k is an integer and $0 \leq k \leq n - 1$.

3.2.1. Initialization Module

The initialization quantum module is beneficial to the segmentation of the quantum image and the formation of H_n , and the partition module PARTITION (k) plays a major role. The PARTITION (k) module can divide the input image of size $2^n \times 2^n$ into $2^{n-k-1} \times 2^{n-k-1}$ blocks of size $2^{k+1} \times 2^{k+1}$; the initialization module quantum circuit is shown in Figure 7a:

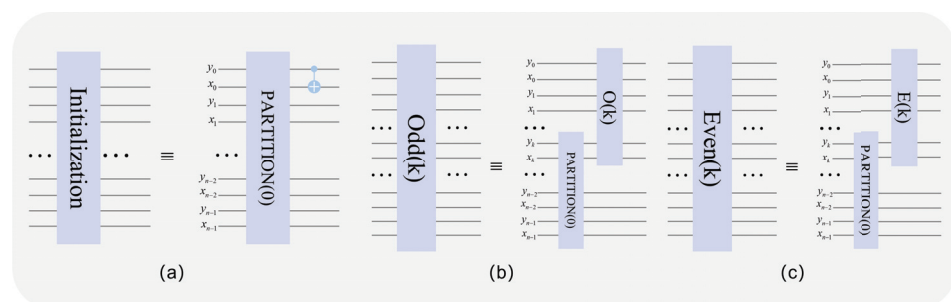


Figure 7. Initializing the quantum circuit. (a) implement the block function (b) implement the scrambling function when k is odd (c) realize the scrambling function when k is even

3.2.2. Odd(k) Module

The main function of the $Odd(k)$ module is to encrypt the pixel position information, where k is odd and $1 \leq k \leq n - 1$. The main role is the odd module $O(k)$ in the $Odd(k)$ module. Figure 7b shows the complete quantum circuit of the $O(k)$ module.

3.2.3. Even (k) Module

As with the function of the $Odd(k)$ module, the function of the Even (k) module is to transform the pixel position, where k is even and $2 \leq k \leq n - 1$. Figure 7c shows the complete Even (k) quantum circuit.

4. Encryption and Decryption of Quantum Images

Quantum image diffusion and scrambling are the two key components of this paper’s encryption phase. At the diffusion step, the picture is made confusing by using DNA coding and various decoding techniques, and the original image is quantum XOR coded. The approach employs iterative Hilbert scrambling during the scrambling stage to encrypt the image’s pixel location data.

4.1. Encryption Process

Using quantum Hilbert scrambling and DNA coding technology, we designed the following quantum image encryption method. The original input quantum image size is $2^n \times 2^n$ and the image representation is NCQI. The encryption flow chart is shown below. Figure 8 shows the encryption process of Rule 1.

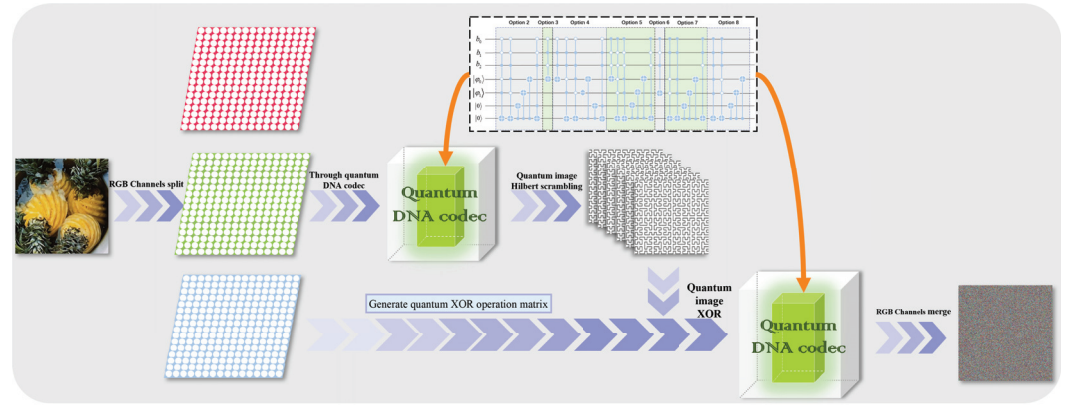


Figure 8. Algorithm flow chart applied to rule one.

Step 1: The pixel matrix of the original image is divided into three RGB channels, and the NCQI representation model is loaded as a quantum image.

$$|\psi_1\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} |c(y, x)\rangle \otimes |yx\rangle. \tag{8}$$

Step 2: The quantum color image is encoded and decoded through the quantum image DNA codec.

In the NCQI representation, the RGB color information will be input into the circuit in binary form, so this paper used rule 1 in Table 1 to encode the binary sequence, and then decodes the sequence according to rule 6, that is, the quantum image is input to the quantum DNA codec, the $D_{1,6}$ operation is performed, and the output is $|\psi_2\rangle$.

$$|\psi_2\rangle = D_{1,6}|\psi_1\rangle. \tag{9}$$

Step 3: Perform Hilbert quantum image scrambling with $|\psi_2\rangle$ iteration.

$|\psi_2\rangle$ has $2^n \times 2^n = 2^{2n}$ pixels, and if the original image pixel order is “1, 2, 3, 4, . . . , 2^{2n} ”, the partition module PARTITION (0) will separate the picture into 2×2 sub-images, that is, $\begin{pmatrix} a & a + 1 \\ a + 2 & a + 3 \end{pmatrix}$. The last two pixels of each sub-image are switched by the C-NOT gate, which will separate the picture into 2×2 sub-images.

The partition module PARTITION (1) divides the image into sub-images of 4×4 and so on, and operates in sequence $O(1), E(2), O(3), E(4), \dots, O(n - 1)/E(n - 1)$ until it is executed to PARTITION (n - 2).

Finally, the scrambled sub-image is restored to the original image size $2^n \times 2^n$ and named $|\psi_3\rangle$.

$$|\psi_3\rangle = Q_{2^n}|\psi_2\rangle, \tag{10}$$

where Q_{2^n} represents performing Hilbert quantum image scrambling on an image of size $2^n \times 2^n$.

Step 4: Between the original picture and the scrambled image, use quantum XOR coding.

Generate matrix Y_{YX} from the pixel color value of image $|\psi_1\rangle$ and convert each element into an octet binary,

$$Y_{YX} = \begin{pmatrix} y_{0,0} & \cdots & y_{0,2^n-1} \\ \vdots & \ddots & \vdots \\ y_{2^n-1,0} & \cdots & y_{2^n-1,2^n-1} \end{pmatrix}, \tag{11}$$

where $y_{yx} = m_{yx}^0 m_{yx}^1 m_{yx}^2 \dots m_{yx}^{23}, m_{yx}^i \in 0, 1$. According to matrix Y_{YX} , the quantum XOR operation matrix F is generated, which is the same as $y_{yx}, b_{yx} = V_{yx}^0 V_{yx}^1 V_{yx}^2 \dots V_{yx}^{23}$, where $V_{yx}^0 \sim V_{yx}^7$ controls the R channel in the quantum circuit, $V_{yx}^8 \sim V_{yx}^{15}$ controls the G channel, and $V_{yx}^{16} \sim V_{yx}^{23}$ controls the B channel, $V_{yx}^i = \begin{cases} X, m_{yx}^i = 1 \\ I, m_{yx}^i = 0 \end{cases}$ obtain image $|\psi_4\rangle$.

Step 5: Decode the image with different rules through the quantum image DNA codec simulator.

$|\psi_4\rangle$ is encoded and decoded by quantum DNA codec, perform the $D_{1,7}$ operation to get $|\psi_5\rangle$, and the encryption is completed.

$$|\psi_5\rangle = D_{1,7}|\psi_4\rangle. \tag{12}$$

4.2. Decryption Process

The reversibility of quantum circuits serves as the foundation for the quantum image decryption technique developed in this research. As a whole, the procedure is as follows:

Step 1: On the encrypted picture, perform the inverse quantum DNA coding and decoding procedure.

The quantum circuit module of the DNA encoder and the quantum DNA codec $D_{7,1}$ are both used to decrypt the encrypted picture to produce the result $|\psi_4\rangle$.

Step 2: Inverse quantum XOR coding.

$$|\psi_3\rangle = X_5^{-1}|\psi_4\rangle. \tag{13}$$

Step 3: The quantum image is iterated to perform Hilbert inverse scrambling.

Because the biggest difference between the quantum circuit and the classical circuit is that the quantum circuit is reversible, and there is no information loss in this process, the Hilbert image scrambling quantum circuit in this paper is reversible. We can input the scrambled image $|\psi_3\rangle$ and obtain the reconstructed image $|\psi_2\rangle$ using quantum Hilbert inverse scrambling.

$$|\psi_2\rangle = Q_{2^n}^{-1}|\psi_3\rangle. \tag{14}$$

Step 4: Pass $|\psi_2\rangle$ through quantum DNA codec $D_{6,1}$ to obtain the original image $|\psi_1\rangle$.

$$|\psi_1\rangle = D_{6,1}|\psi_2\rangle. \tag{15}$$

5. Safety Analysis

We performed simulation experiments in MATLAB and Python using classical computers since there are no quantum computers available. We did not take into account the impact of decoherence and inaccuracy in the quantum version while processing numerical data. In order to examine the encrypted data in this part, three color pictures of pineapples, roses, and plants with pixel sizes of 512×512 were utilized as the original image. The following summarizes the encryption and decryption simulation findings. Figure 9 shows the comparison of the results before and after the application of this algorithm.



Figure 9. Effect of encryption and decryption.

5.1. Histogram Analysis

One of the key indications needed to assess the security of encryption techniques is the histogram, which may considerably reflect the intensity distribution of picture pixels. The encrypted histograms of the three photos are shown in Figures 10–12.

In contrast to the non-uniform peak distribution of the plaintext histogram, which is seen in the image, the peak value of the histogram encrypted by this approach becomes uniform. As a result, the attacker is unable to obtain the picture data by studying the ciphertext image’s histogram.

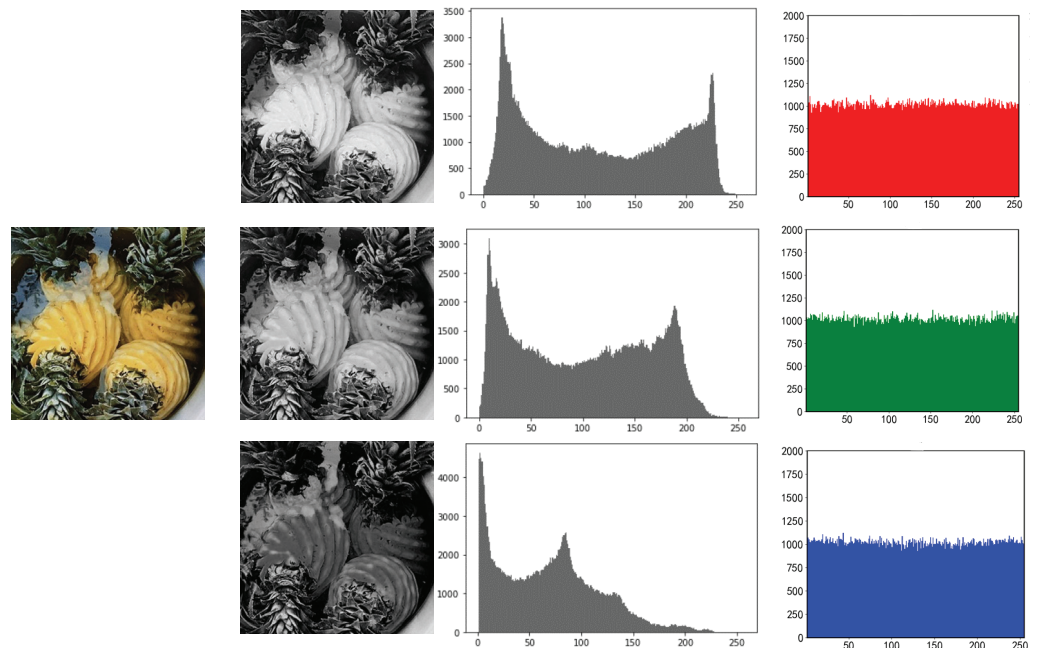


Figure 10. RGB three-channel histogram of pineapple before and after encryption.

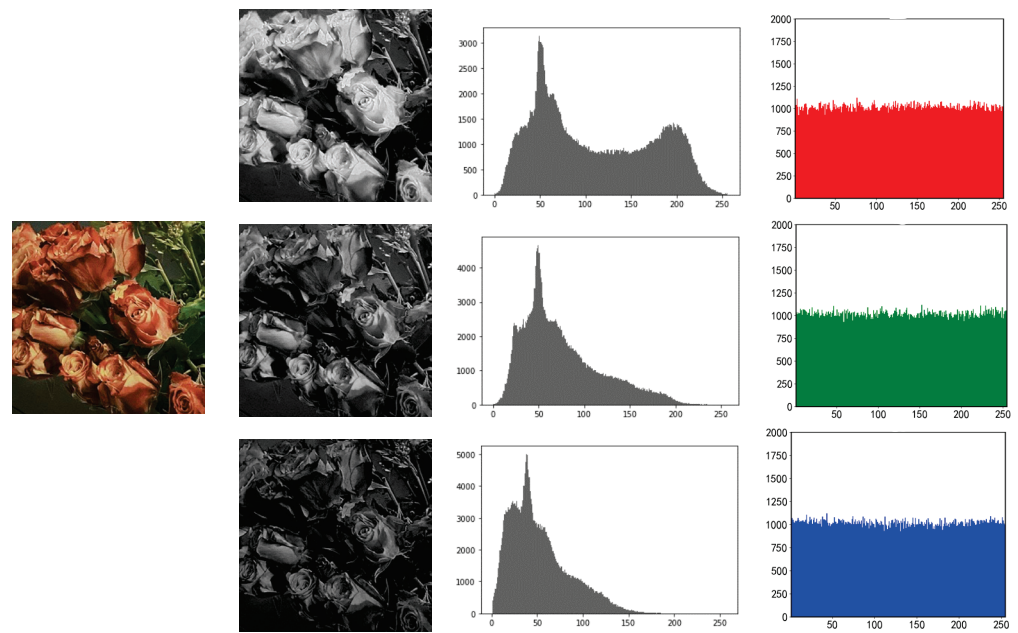


Figure 11. RGB three-channel histogram of rose before and after encryption.

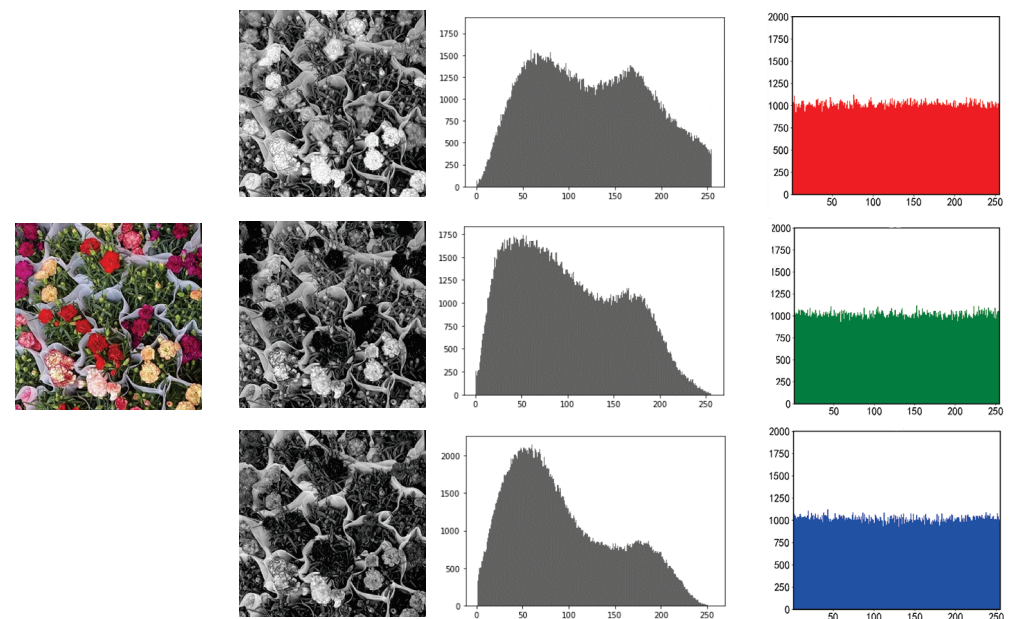


Figure 12. RGB three-channel histogram of plants before and after encryption.

5.2. Correlation Analysis of Adjacent Pixels

A crucial metric for determining the efficacy of the encryption technique is the correlation between neighboring pixels. As there is a significant connection between neighboring pixels in the original picture, a good encryption technique should minimize this correlation to zero. In this study, we utilized this coefficient to compare the correlation between neighboring pixels both before and after the method was applied

$$r = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}}, \tag{16}$$

where A and B represent the values of adjacent pixels, $\text{cov}(A, B)$ is the covariance of A and B , and $\sqrt{D(A)}$ and $\sqrt{D(B)}$ are the variances of A and B . In this section, the pixel correlation between the original image and the ciphertext image was analyzed horizontally,

vertically, and diagonally. The results are shown in Figures 13 and 14, and the specific data are reflected in Tables 2 and 3, where C-Image represents the ciphertext image.

The suggested encryption technique clearly creates a sizable correlation gap between the ciphertext picture and the original image based on the data shown in the chart, demonstrating the algorithm’s effectiveness.

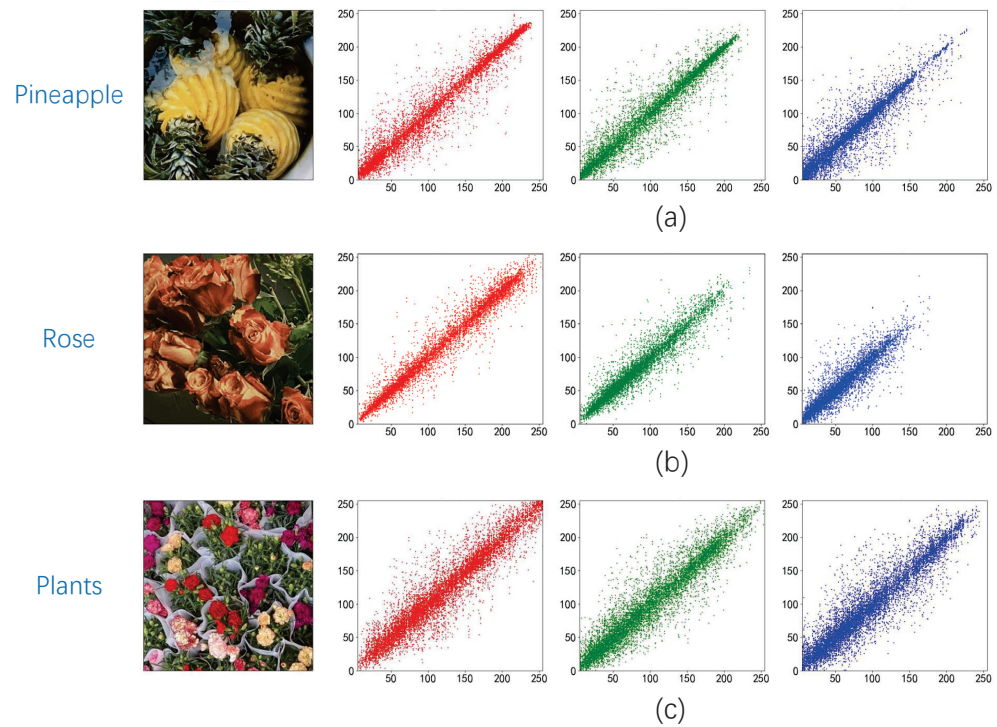


Figure 13. (a) Pineapple correlation analysis; (b) Rose correlation analysis; (c) Plants correlation analysis.

Table 2. Correlation analysis value of original image.

Image	Channel	Horizontal	Vertical	Diagonal
Pineapple	R	0.9849	0.9813	0.9833
	G	0.9753	0.9763	0.9588
	B	0.9597	0.9550	0.9251
Rose	R	0.9835	0.9844	0.9753
	G	0.9651	0.9643	0.9466
	B	0.9461	0.9446	0.9115
Plants	R	0.9539	0.9583	0.9256
	G	0.9556	0.9563	0.9238
	B	0.9478	0.9540	0.9148

Table 3. Three-channel correlation analysis of ciphertext images.

Image	Channel	Horizontal	Vertical	Diagonal
C-Pineapple	R	0.0002	0.0045	0.0051
	G	0.0026	0.0012	0.0044
	B	0.0037	0.0046	0.0029
C-Rose	R	0.0028	0.0049	0.0049
	G	0.0055	0.0023	0.0086
	B	0.0034	0.0053	0.0014
C-Plants	R	0.0010	0.0081	0.0004
	G	0.0052	0.0043	0.0025
	B	0.0057	0.0042	0.0033

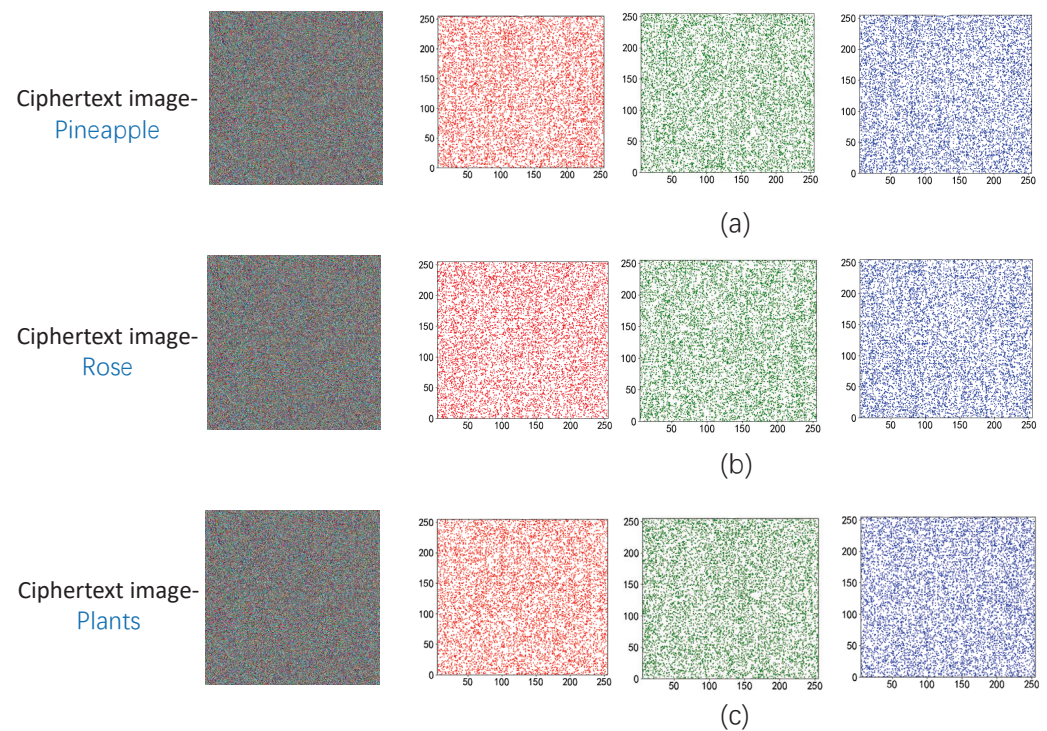


Figure 14. (a) Correlation analysis after pineapple encryption; (b) Correlation analysis after Rose encryption; (c) Correlation analysis after plants encryption.

5.3. Key Sensitivity Analysis

Two words that are often used to characterize the quantity of pixels and the average intensity of change between the original picture and the ciphertext image are NPCR and UACI. In accordance with the associated ideal value, the key sensitivity of the NPCR = 99.6094%, UACI = 33.4635% algorithm should be as high as possible; the more closely the numerical value resembles the ideal value, the stronger the security of the encryption technique should be. The data for this method’s NPCR and UACI are shown in Table 4. Table 5 compares our work numerically to the NPCR and UACI algorithms that have been proposed in different papers. This indicates very clearly how much more efficiently the technique used in this research can guarantee picture confidentiality.

Table 4. Three-channel average NPCR and UACI data.

Image	RGB Average NPCR	RGB Average UACI
C-Pineapple	99.6138%	33.4944%
C-Rose	99.6204%	33.5147%
C-Plants	99.6097%	33.5643%

Table 5. Comparison of information entropy of different algorithms.

Algorithm	Average NPCR	Average UACI
Proposed	99.61%	33.42%
Ref. [44]	99.61%	31.60%
Ref. [45]	99.57%	33.38%

5.4. Information Entropy

We often use information entropy to evaluate the unpredictability of the distribution of ciphertext pictures. The ciphertext image’s pixels may be distributed evenly via a decent picture encryption method, making the image more resistant to outside attacks. The perfect information entropy is eight. The image encryption effect is better and the value is more

closely aligned with the ideal value as the pixel distribution becomes more uniform. The information entropy of our recommended approach is shown in Table 6. The following table provides ample proof of the algorithm's strong security and robustness by showing that the average information entropy of RGB's three channels may reach 7.999.

Table 6. Information entropy data.

Ciphertext Image	R	G	B
Pineapple	7.99925	7.99901	7.99921
Rose	7.99910	7.99930	7.99889
Plants	7.99922	7.99895	7.99912

5.5. Key Space

The modified picture serves as the key matrix in the encryption procedure described in this research. The color picture is 512×512 in size, making its key space $2^{512 \times 512 \times 24}$ pixels, which is sufficient to stave against brute force assaults.

5.6. Scheme Reversibility Verification

Indicators used often to assess picture quality in the field of image processing include peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

5.6.1. Peak Signal-to-Noise Ratio

To evaluate the image's decryption quality, we employed PSNR. The floating-point value that PSNR returns will range from 30 to 50 if the two input photos are comparable; the higher the number, the greater the similarity. The PSNR values of plaintext pictures and encrypted images with a size of 512 to 512 are larger than 30 dB, according to simulation findings, and the average value is 43.4590. Table 7 displays the specific data. This demonstrates the algorithm's strong ability to aid in rebuilding.

5.6.2. Structural Similarity

The SSIM value ranges from 0 to 1. The value of SSIM increases with the degree of similarity between the two photos. The picture acquired using the image decryption approach suggested in this work was compared with the original image. Table 7 displays the specific data. The average value of SSIM determined by the simulation results is 0.980358, which shows that the technology has an excellent decryption and recovery effect.

Table 7. Image similarity analysis.

Name	PSNR	SSIM
Pineapple	43.7358	0.982998
Rose	42.9974	0.979976
Plants	43.6438	0.978102
Average	43.4590	0.980358

6. Conclusions

Quantum image processing is committed to the use of quantum computing technology to capture, restoration, and other classical image operations. Because of its exponential storage capacity and parallelism, this technology has a strong advantage in realizing real-time operations such as image retrieval and processing. In this paper, the circuit of a quantum DNA codec was designed, and the image information was encrypted by using the biological characteristics of DNA and the physical properties of quantum mechanics. At the end of this article, the combination of DNA technology and quantum image encryption was studied and verified. According to the simulation, average NPCR = 99.6094%, average NPCR = 33.4244%, the average information entropy of RGB three channels is more than 7.999, and the average value of SSIM determined by the simulation results is 0.980358.

These results unmistakably demonstrate the viability and effectiveness of the quantum picture encryption system presented in this research, which is based on DNA codec and Hilbert scrambling.

In this encryption scheme, a quantum DNA codec was designed to enable the biological field to participate in the quantum image encryption process. It is hoped that it can play a greater role in the later research. In the follow-up work, we hope to combine quantum random walk with DNA technology to realize the integration of physics and biology again. This will be the focus of our next paper.

Author Contributions: Conceptualization, Y.W.; Methodology, J.G. and Y.W.; Software, Z.S.; Validation, Z.S.; Formal analysis, J.G.; Data curation, Z.S.; Writing—original draft, J.G.; Writing—review & editing, J.G.; Visualization, J.G. and Y.W.; Supervision, Y.W. and S.W.; Project administration, S.W.; Funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Shandong Province, China (Grant Nos. ZR2021MF049), Joint Fund of Natural Science Foundation of Shandong Province (Grant Nos. ZR2022LLZ012), Joint Fund of Natural Science Foundation of Shandong Province (Grant Nos. ZR2021LLZ001), Project supported by the National Natural Science Foundation of China (Grant Nos. 11975132), National Natural Science Foundation of China (Grant Nos. 12005110) and the Natural Science Foundation of Shandong Province, China (Grant Nos. ZR2022JQ04).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Z.B.; Xu, M.Z.; Zhang, Y.N. Review of Quantum Image Processing. *Arch. Comput. Methods Eng.* **2022**, *29*, 737–761. [CrossRef]
2. Nielsen, M.A.; Chuang, I. Quantum computation and quantum information. *Am. J. Phys.* **2002**, *70*, 558–559
3. Hardy, L. Quantum mechanics, local realistic theories, and Lorentz-invariant realistic theories. *Phys. Rev. Lett.* **1992**, *68*, 2981. [CrossRef]
4. Hardy, L. Nonlocality for two particles without inequalities for almost all entangled states. *Phys. Rev. Lett.* **1993**, *71*, 1665. [CrossRef]
5. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
6. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
7. Ashikhmin, A.; Litsyn, S.; Tsfasman, M.A. Asymptotically good quantum codes. *Phys. Rev. A* **2001**, *63*, 032311. [CrossRef]
8. Ashikhmin, A.; Knill, E. Nonbinary quantum stabilizer codes. *IEEE Trans. Inf. Theory* **2001**, *47*, 3065–3072. [CrossRef]
9. Trugenberger, C.A. Probabilistic quantum memories. *Phys. Rev. Lett.* **2001**, *87*, 067901. [CrossRef]
10. Trugenberger, C.A. Phase transitions in quantum pattern recognition. *Phys. Rev. Lett.* **2002**, *89*, 277903. [CrossRef]
11. Venegas-Andraca, S.E.; Bose, S. Storing, processing, and retrieving an image using quantum mechanics. *Quantum Inf. Comput.* **2003**, *5105*, 137–147.
12. Klappenecker, A.; Rotteler, M. Discrete cosine transforms on quantum computers. In Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, Pula, Croatia, 19–21 June 2001; pp. 464–468.
13. Fijany, A.; Williams, C.P. Quantum wavelet transform: Fast algorithms and complete circuits. In *Quantum Computing and Quantum Communications*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 10–33.
14. Beach, G.; Lomont, C.; Cohen, C. Quantum image processing (quip). In Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop, Washington, DC, USA, 15–17 October 2003; pp. 39–44.
15. Caraiman, S.; Manta, V.I. New applications of quantum algorithms to computer graphic: The quantum random sample consensus algorithm. In Proceedings of the 6th ACM Conference on Computing Frontiers, Ischia, Italy, 18–20 May 2009; pp. 81–88.
16. Caraiman, S.; Manta, V.I. Image segmentation on a quantum computer. *Quantum Inf. Process.* **2015**, *14*, 1693–1715. [CrossRef]
17. Castleman, K.R. *Digital Image Processing*; Prentice Hall Press: Hoboken, NJ, USA, 1996.
18. Li, H.S.; Li, C.Y.; Chen, X.; Xia, H.Y. Quantum Image Encryption Algorithm Based on NASS. *Int. J. Theor. Phys.* **2018**, *57*, 3745–3760. [CrossRef]

19. Li, H.S.; Li, C.Y.; Chen, X.; Xia, H.Y. Quantum image encryption based on phase-shift transform and quantum Haar wavelet packet transform. *Mod. Phys. Lett. A* **2019**, *34*, 1950214. [CrossRef]
20. Yang, Y.G.; Xia, J.; Jia, X.; Zhang, H. Novel image encryption/decryption based on quantum Fourier transform and double phase encoding. *Quantum Inf. Process.* **2013**, *12*, 3477–3493. [CrossRef]
21. Yan, F.; Iliyasu, A.M.; Venegas-Andraca, S.E. A survey of quantum image representations. *Quantum Inf. Process.* **2016**, *15*, 1–35. [CrossRef]
22. Le, P.Q.; Dong, F.; Hirota, K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process.* **2011**, *10*, 63–84. [CrossRef]
23. Zhang, Y.; Lu, K.; Gao, Y.H.; Wang, M. NEQ: A novel enhanced quantum representation of digital images. *Quantum Inf. Process.* **2013**, *12*, 2833–2860. [CrossRef]
24. Sun, B.; Iliyasu, A.; Yan, F.; Dong, F.Y.; Hirota, K. An RGB multi-channel representation for images on quantum computers. *J. Adv. Comput. Intell. Inform.* **2013**, *17*, 404–415. [CrossRef]
25. Li, H.S.; Zhu, Q.X.; Zhou, R.G.; Lan, S.; Yang, X.J. Multi-dimensional color image storage and retrieval for a normal arbitrary quantum superposition state. *Quantum Inf. Process.* **2014**, *13*, 991–1011. [CrossRef]
26. Li, H.S.; Zhu, Q.X.; Lan, S.; Shen, C.Y.; Zhou, R.G.; Mo, J. Image storage, retrieval, compression and segmentation in a quantum system. *Quantum Inf. Process.* **2013**, *12*, 2269–2290. [CrossRef]
27. Zhang, Y.; Lu, K.; Gao, Y.H.; Xu, K. A novel quantum representation for log-polar images. *Quantum Inf. Process.* **2013**, *12*, 3103–3126. [CrossRef]
28. Zhang, J.L.; Huang, Z.J.; Li, X.; Wu, M.Q.; Wang, X.Y.; Dong, Y.M. Quantum Image Encryption Based on Quantum Image Decomposition. *Int. J. Theor. Phys.* **2021**, *60*, 2930–2942. [CrossRef]
29. Zhou, N.R.; Hu, Y.Q.; Gong, L.H.; Li, G.Y. Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations. *Quantum Inf. Process.* **2017**, *16*, 164. [CrossRef]
30. Adleman, L.M. Molecular computation of solutions to combinatorial problems. *Science* **1994**, *266*, 1021–1024. [CrossRef]
31. Leier, A.; Richter, C.; Banzhaf, W.; Rauhe, H. Cryptography with DNA binary strands. *Biosystems* **2000**, *57*, 13–22. [CrossRef]
32. Chang, W.L.; Guo, M.Y.; Ho, M.S.H. Fast parallel molecular algorithms for DNA-based computation. factoring integers. *IEEE Trans. Nanobiosci.* **2005**, *4*, 149–163. [CrossRef]
33. Basu, S.; Karuppiyah, M.; Nasipuri, M.; Halder, A.K.; Radhakrishnan, N. Bio-inspired cryptosystem with DNA cryptography and neural networks. *J. Syst. Archit.* **2019**, *94*, 24–31. [CrossRef]
34. Alghafis, A.; Firdousi, F.; Khan, M.; Batool, S.I.; Amin, M. An efficient image encryption scheme based on chaotic and Deoxyribonucleic acid sequencing. *Math. Comput. Simul.* **2020**, *177*, 441–466. [CrossRef]
35. Liu, Y.; Zhang, J.D. A Multidimensional Chaotic Image Encryption Algorithm based on DNA Coding. *Multimed. Tools Appl.* **2020**, *79*, 21579–21601. [CrossRef]
36. Wang, X.Y.; Zhang, Y.Q.; Bao, X.M. A novel chaotic image encryption scheme using DNA sequence operations. *Opt. Lasers Eng.* **2015**, *73*, 53–61. [CrossRef]
37. Zhang, X.Q.; Wang, X.S. Multiple-image encryption algorithm based on DNA encoding and chaotic system. *Multimed. Tools Appl.* **2019**, *78*, 7841–7869. [CrossRef]
38. Wang, Y.N.; Song, Z.Y.; Ma, H.Y. Color image encryption algorithm based on DNA code and alternating quantum random walk. *Acta Phys. Sin.* **2021**, *1*, 230302. [CrossRef]
39. Sang, J.Z.; Wang, S.; Li, Q. A novel quantum representation of color digital images. *Quantum Inf. Process.* **2017**, *16*, 42. [CrossRef]
40. Arnold, V.I.; Avez, A. *Ergodic Problems of Classical Mechanics*; Benjamin: Amsterdam, The Netherlands, 1968; Volume 9.
41. Wang, S.; Xu, X.S. A new algorithm of Hilbert scanning matrix and its MATLAB program. *J. Image Graph.* **2006**, *11*, 119–122.
42. Jiang, N.; Wang, L.; Wu, W.Y. Quantum Hilbert Image Scrambling. *Int. J. Theor. Phys.* **2014**, *53*, 2463–2484. [CrossRef]
43. Gong, L.H.; He, X.T.; Zhou, N.R. Quantum Image Encryption Algorithm Based on Quantum Image XOR Operations. *Int. J. Theor. Phys.* **2016**, *55*, 3234–3250. [CrossRef]
44. Vagish, K.D.; Rajakumaran, C.; Kavitha, R. Chaos based encryption of quantum images. *Multimed. Tools Appl.* **2020**, *79*, 23849–23860.
45. Li, C.; Yang, X.Z. An image encryption algorithm based on discrete fractional wavelet transform and quantum chaos. *Optik* **2022**, *260*, 169042. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Optimizing Quantum Classification Algorithms on Classical Benchmark Datasets

Manuel John ^{1,2}, Julian Schuhmacher ¹, Panagiotis Barkoutsos ^{1,†}, Ivano Tavernelli ¹ and Francesco Tacchino ^{1,*}

¹ IBM Quantum, IBM Research Europe—Zurich, 8803 Rüschlikon, Switzerland

² Institute for Theoretical Physics, ETH Zürich, 8093 Zurich, Switzerland

* Correspondence: fta@zurich.ibm.com

† Current Address: PASQAL SAS, 2 Av. Augustin Fresnel Palaiseau, 91120 Palaiseau, France.

Abstract: The discovery of quantum algorithms offering provable advantages over the best known classical alternatives, together with the parallel ongoing revolution brought about by classical artificial intelligence, motivates a search for applications of quantum information processing methods to machine learning. Among several proposals in this domain, quantum kernel methods have emerged as particularly promising candidates. However, while some rigorous speedups on certain highly specific problems have been formally proven, only empirical proof-of-principle results have been reported so far for real-world datasets. Moreover, no systematic procedure is known, in general, to fine tune and optimize the performances of kernel-based quantum classification algorithms. At the same time, certain limitations such as kernel concentration effects—hindering the trainability of quantum classifiers—have also been recently pointed out. In this work, we propose several general-purpose optimization methods and best practices designed to enhance the practical usefulness of fidelity-based quantum classification algorithms. Specifically, we first describe a data pre-processing strategy that, by preserving the relevant relationships between data points when processed through quantum feature maps, substantially alleviates the effect of kernel concentration on structured datasets. We also introduce a classical post-processing method that, based on standard fidelity measures estimated on a quantum processor, yields non-linear decision boundaries in the feature Hilbert space, thus achieving the quantum counterpart of the radial basis functions technique that is widely employed in classical kernel methods. Finally, we apply the so-called quantum metric learning protocol to engineer and adjust trainable quantum embeddings, demonstrating substantial performance improvements on several paradigmatic real-world classification tasks.

Keywords: quantum machine learning; quantum classification algorithms; quantum kernel methods

Citation: John, M.; Schuhmacher, J.; Barkoutsos, P.; Tavernelli, I.; Tacchino, F. Optimizing Quantum Classification Algorithms on Classical Benchmark Datasets. *Entropy* **2023**, *25*, 860. <https://doi.org/10.3390/e25060860>

Academic Editors: Brian R. La Cour and Giuliano Benenti

Received: 20 April 2023

Revised: 24 May 2023

Accepted: 24 May 2023

Published: 27 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Machine learning (ML) algorithms are ubiquitous in today's world. These techniques leverage the natural ability of computers to sieve through vast amounts of data with the aim of revealing the underlying patterns and to accomplish a wide range of tasks, such as image classification, automated generation of text and images or, more generally, decision making.

Quantum computers implement a novel information processing paradigm and may provide an alternative platform for executing machine learning algorithms, an approach known as Quantum Machine Learning (QML) [1,2]. The potential for using quantum computation in machine learning is essentially two-fold. On the one hand, one could, for example, leverage results from quantum optimization or quantum linear algebra [3–6] to increase the training efficiency of classical ML models [7,8]. On the other hand, native quantum models, such as quantum neural networks [9–17], could be engineered to directly carry out specific learning tasks on classical or quantum data and to analyze correlations that are hard to describe or capture classically [1].

Quantum kernel methods applying quantum feature maps naturally emerge from the second line of research. Here, classical input feature vectors are mapped to high-dimensional Hilbert spaces realized with feature-dependent preparation of quantum states [18–20]. Once such a quantum embedding is realized, a decision rule to carry out the desired classification task can be obtained directly from the fidelity between the encoded feature vectors [20] or by passing the resulting quantum kernel to a classical support vector machine [18,19]. Soon after the introduction of quantum kernel methods, it was shown that quantum kernel methods, equipped with the right quantum feature maps, can solve certain specifically designed problems more efficiently than any known classical counterpart [21], thus motivating a large body of research aimed at finding similar advantages in more generic and applied contexts [22–24], including for the following: data analysis for high-energy physics [25–27], quantum phase classification [28], fraud detection [29] and virtual screening for drug discovery [30]. While some promising examples were identified [27,30], only proof-of-principle results have been achieved so far, mostly based on empirical considerations. Moreover, only selected applications may be viable in the near future due to the presence of hardware noise, although several known error suppression and mitigation strategies [31–35] could be employed, and have, in fact, already been tested in the context of QML [12,36].

At the same time, it has recently become clear that the high expressivity of parameterized quantum circuits, together with other properties, such as entangling power or the structure of certain cost functions, can have unintended consequences. In fact, this not only hinders the trainability of variational quantum models—leading to the so-called barren plateau phenomenon [37–39]—but also has a negative impact on the capabilities of quantum kernel methods [23,40]. Some of the latest theoretical advancements in the literature have addressed precisely this class of problems: for example, both Kübler et al. [40] and Huang et al. [23] proposed projection-based approaches as well as ways to incorporate inductive biases, i.e., constraints on the range of representable functions. In parallel, Shaydulin et al. [41] proposed a strategy to tune the bandwidth of quantum kernels, which was shown to have effects on generalization performances [42]. However, the exponential concentration of the kernel values, due to high expressivity, entanglement, global measurements, and noise, prevents, in general, the application of fidelity and projection-based quantum kernels to higher numbers of qubits [43].

In this work, we discuss best practices to reduce the impact of known limitations of quantum kernel methods and we explore different general-purpose strategies to systematically enhance the performances of quantum classification algorithms, based on quantum feature maps, with a specific focus on paradigmatic real-world datasets. First, we propose a strategy to alleviate the problem of the exponential concentration in the presence of structured datasets. Our approach is related to the one described in Reference [41], but, instead of global rescaling of the input features, implied by tuning the kernel bandwidth, we employ a separate scaling factor for each feature. In practice, we identify a *domain of rotations* and normalize the input features so as to ensure that the arguments of each parameterized quantum gate in a feature map do not exceed a predefined range (e.g., $[-\pi, \pi]$). As a second step, we describe a classical post-processing procedure that, starting from the usual fidelity measurements between encoded quantum feature vectors, effectively engineers a continuous nearest-neighbor classification rule, and, therefore, enables non-linear decision boundaries in the Hilbert space. This extends the basic notion of quantum kernel and fidelity-based classifiers which, in the standard formulation, only make use of linear separating hyperplanes. Finally, we explore the concept of trainable quantum feature maps, originally introduced for some specific examples by Lloyd et al. [20] and Glick et al. [44]. In this case, we follow the intuition that a generic quantum feature map may not perform well across multiple datasets originating from a wide range of application domains, but should rather be, at least to a certain degree, tailored to the problem. We benchmark this procedure, known as quantum metric learning or quantum kernel alignment, on a collection of paradigmatic datasets of practical relevance.

The remainder of the paper is structured as follows. In Section 2, we introduce some basic concepts related to quantum classification and quantum kernel methods (Section 2.1). We describe the specific quantum algorithms employed in our work (Sections 2.2 and 2.3) and we provide information about the datasets considered in our numerical experiments (Section 2.4). We present our results in Section 3, while a discussion of their implications and some concluding remarks are contained in Section 4.

2. Materials and Methods

2.1. Quantum Classification Algorithms and Quantum Embeddings

Classification algorithms from the family of kernel methods rely on a function, called a *kernel*, that quantifies the similarity between data vectors x_i . For binary classification, the kernel function embeds the data vector into a high-dimensional feature space, where the two classes (ideally) become linearly separable. The success of classical kernel methods stems from the so-called *kernel trick*, which allows one to evaluate the kernel function without explicitly mapping the data to the high-dimensional feature space. The widely used radial basis function (RBF) kernel,

$$K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right), \quad (1)$$

is an example for which the effective feature space would be infinitely dimensional [45]. However, the kernel itself can be evaluated efficiently.

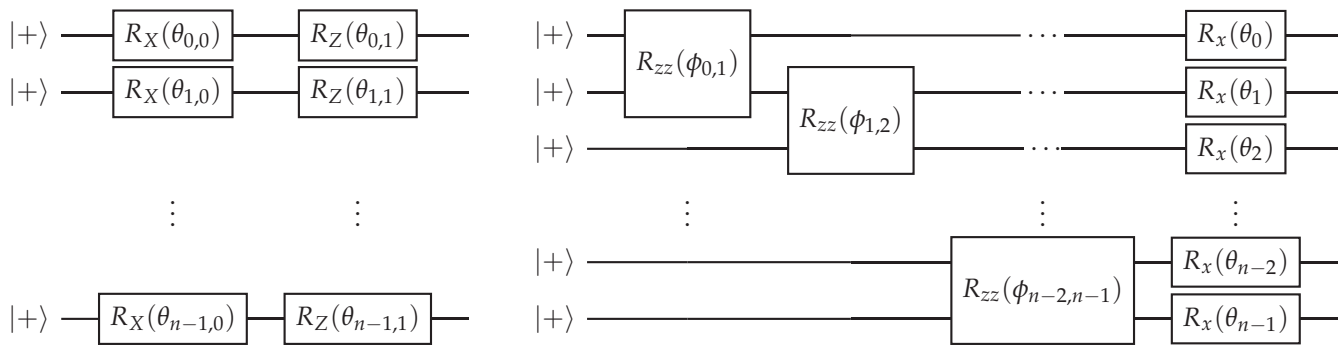
Quantum computers provide an alternative platform to implement kernel methods, since they provide an efficient way to access high-dimensional Hilbert spaces into which classical data can be embedded [20]. A feature vector x can be mapped into the space of n -qubit quantum states by using the entries of x as arguments of a parameterized quantum circuit $U(x)$ [18,19]. We denote the quantum state prepared by applying such a parameterized unitary to the zero state as

$$|x\rangle = U(x)|0\rangle. \quad (2)$$

The unitary $U(x)$ is often referred to as the quantum feature map or quantum embedding. The potential of using such a method originates from the fact that, in general, quantum embeddings cannot be efficiently simulated with classical computers [18,21]. This is a necessary, albeit not sufficient, condition for achieving quantum advantage with quantum kernel methods.

In Figure 1, we present the two feature maps that we apply in our study. The first one, denoted as *RXRZ embedding*, encodes the input features in a layer of single-qubit R_X rotations, followed by a layer of R_Z rotations (see Figure 1a). Here, with R_K , $K \in \{X, Y, Z\}$ we denote the standard Pauli rotation gates. The basic building block shown in Figure 1a can be repeated a number L of times, hence producing a L -layer version of the feature map. If we choose the parameters in the circuit proportional to the entries x_i of the data vector x (i.e., $\theta_{i,\{0,1\}} \sim x_i$) the required number of qubits corresponds to the dimension of the classical data vector. The concrete relation between the parameters and the input features is specified in Section 3.

The second feature map, denoted as *ZZ embedding*, and inspired by similar popular proposals in the literature [18,30,46], is illustrated in Figure 1b. Compared to the RXRZ embedding it additionally contains entangling operators between the qubits in order to capture correlations in the input features. As in the RXRZ case, the ZZ feature map can also be repeated for L layers. We choose the parameters of the two qubit R_{ZZ} gates proportional to the product of feature values (i.e., $\phi_{i,j} \sim x_i x_j$). The parameters of the subsequent layer of R_X rotations are proportional to a single feature value, identical to the RXRZ embedding. The concrete relation between the parameters and the input features is specified in Section 3.



(a) RXRZ feature map.

(b) ZZ feature map.

Figure 1. (a) One layer of the RXRZ feature map, which produces a classically simulatable output but serves as a reference for the ZZ embedding in (b). We optionally apply L layers of the depicted gates.

In the space of n -qubit quantum states a natural choice for the kernel function is the overlap between two embedded feature vectors \mathbf{x}_i and \mathbf{x}_j ,

$$K(\mathbf{x}_i, \mathbf{x}_j) = |\langle \mathbf{x}_i | \mathbf{x}_j \rangle|^2. \tag{3}$$

This quantity, also called *fidelity*, can be evaluated on a quantum computer by means of, for example, the so-called SWAP test [47], or the inversion test [18].

2.2. Quantum Fidelity and RBF Fidelity Classifiers

For binary classification between two classes A and B, an intuitive method to determine the class label of a new data point \mathbf{x} is the fidelity classifier [20]. Given access to reference data points belonging to the two classes (i.e., a training set), we calculate the average fidelity of $|\mathbf{x}\rangle$ with the embedded data points from classes A and B, denoted as $\{|\mathbf{a}\rangle\}$ and $\{|\mathbf{b}\rangle\}$, respectively. More concretely, the decision function of the fidelity classifier can be written as

$$f(\mathbf{x}) = \frac{1}{M_A} \sum_{\mathbf{a} \in A} |\langle \mathbf{x} | \mathbf{a} \rangle|^2 - \frac{1}{M_B} \sum_{\mathbf{b} \in B} |\langle \mathbf{x} | \mathbf{b} \rangle|^2, \tag{4}$$

where M_A is the number of reference points belonging to class A and M_B is the number of reference points in class B. If the decision function evaluates to a value $f(\mathbf{x}) > 0$ the data point \mathbf{x} is assigned to class A, and vice versa if $f(\mathbf{x}) < 0$. For this classifier, the corresponding hyperplane separating the two classes is linear in the Hilbert space [20]. In Figure 2a, we visualize the hyperplane obtained with the fidelity classifier for the binary classification between two classes of pure quantum states randomly chosen on the single-qubit Bloch sphere (both the states and the corresponding classes are selected/assigned randomly in this example).

To go beyond linear decision boundaries in the Hilbert space, we propose a classifier based on the following kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma(1-|\langle \mathbf{x}_i | \mathbf{x}_j \rangle|^2)}, \tag{5}$$

inspired by the classical RBF kernel presented in Equation (1). Here γ is a tunable hyperparameter. The classifier obtained with the decision function

$$f(\mathbf{x}) = \frac{1}{M_A} \sum_{\mathbf{a} \in A} e^{-\gamma(1-|\langle \mathbf{x} | \mathbf{a} \rangle|^2)} - \frac{1}{M_B} \sum_{\mathbf{b} \in B} e^{-\gamma(1-|\langle \mathbf{x} | \mathbf{b} \rangle|^2)}, \tag{6}$$

is denoted as *RBF fidelity classifier* in the following. Evaluating the decision function requires the same amount of quantum resources as for the fidelity classifier, since the exponentiation

is a simple post-processing of the fidelity values. We apply the same decision rule as for the fidelity classifier. Using the RBF fidelity for classification only considers data points within a neighborhood of $|x\rangle$, where the range of this neighborhood is determined by γ , and the fidelity is used as a distance metric. Since this approach is more flexible than the fidelity classifier, we expect that the RBF fidelity classifier will offer better classification performance. In Figure 2b, we visualize the non-linear hyperplane obtained with the RBF fidelity classifier for the same binary classification as for the fidelity classifier above.

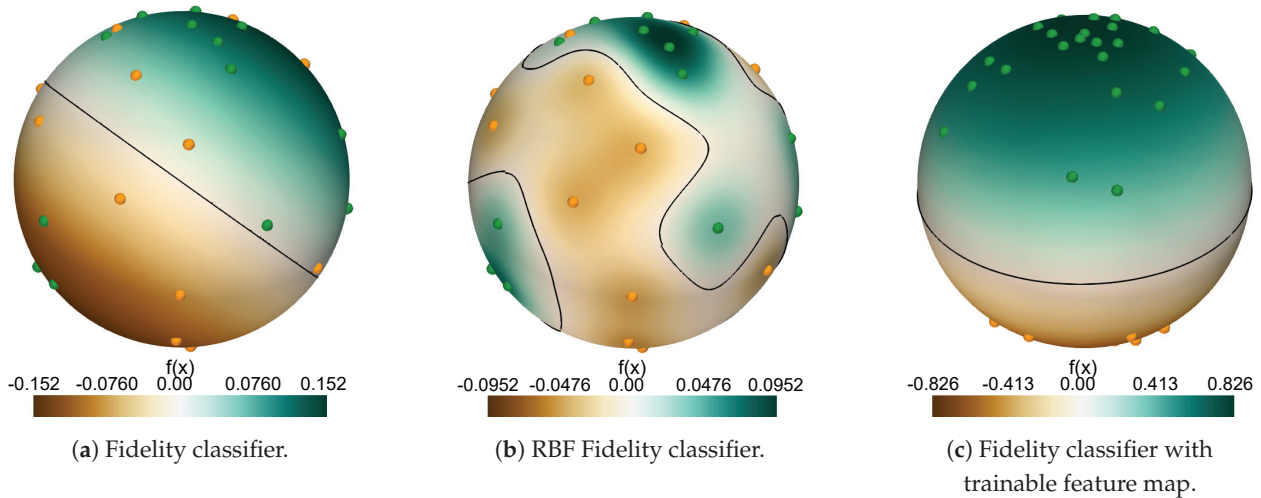


Figure 2. Illustration of a prototypical classification with (a) the fidelity classifier, (b) the RBF fidelity classifier (with $\gamma = 50$), and (c) the fidelity classifier with trainable feature map. (a,b) visualize the respective decision functions based on two classes of 25 randomly sampled points on the Bloch sphere. (c) shows the ideal outcome of quantum metric learning. The samples of the two embedded classes are mapped to opposing poles of the Bloch sphere. The black line corresponds to the decision boundary, where the decision function evaluates to zero.

2.3. Quantum Metric Learning

It has been shown in References [20,44] that adding trainable parts to the quantum feature map can lead to an improvement in the classification performance of quantum kernel and fidelity-based models. Instead of manipulating the separating hyperplane (as in the case of the RBF classifier presented in Section 2.2 above), these approaches manipulate the feature map by tailoring it to the considered classification task. Following the approach presented in Reference [20], called *quantum metric learning*, we can introduce trainable parameters $\alpha_{i,j}$ and β_i to the gates in the ZZ embedding,

$$\tilde{\phi}_{i,j} = \phi_{i,j} + \alpha_{i,j} \quad \tilde{\theta}_i = \theta_i + \beta_i. \tag{7}$$

Optimizing these parameters changes the form of the embedding of the data points in the Hilbert space, and, therefore, effectively modifies the relations between them.

The cost function used for the optimization of the parameters is the empirical risk computed for the decision function $f_{\alpha,\beta}(x)$,

$$I[f_{\alpha,\beta}] = -\frac{1}{M} \sum_{m=1}^M L(f_{\alpha,\beta}(x_m), y_m), \tag{8}$$

where $L(f_{\alpha,\beta}(x), y)$ is the loss function, x are the data samples, y the corresponding labels, the subscript (α, β) denotes the dependence on the trainable parameters and M is the total number of available data samples in the reference data set. Using the fidelity classifier and $L(f(x), y) = f(x) \cdot y$ as the loss function, the parameters are then optimized such that data points belonging to the same class are mapped to close regions in the Hilbert space, and data points belonging to different classes are mapped to distant regions in the Hilbert

space [20]. In Figure 2c, we illustrate the desired effect of optimizing the feature map. Ideally, optimizing the embedding parameters maps the two classes to opposing poles of the Bloch sphere. This allows for accurate classification using the linear hyperplane of the fidelity classifier.

2.4. Datasets

We evaluated the performance of the quantum classifiers introduced above on six different datasets, representing paradigmatic classification tasks from a broad range of application domains. The datasets and their characteristics are listed in Table 1.

Table 1. Datasets used in this study and their properties after cleaning the data (removing duplicates, defective samples, etc.).

Dataset	# Features	# Positives	# Negatives	Source	Description
MNIST	28×28	500	500	[48]	Grayscale images of hand-written digits (0's vs. 9's)
fMNIST	28×28	500	500	[49]	Grayscale images of clothing (T-shirts vs. dresses)
musk	166	207	269	[50,51]	Molecules occurring in different conformations (musk vs. non-musk)
sonar	60	97	111	[51,52]	Sonar signals (bounced off a metal cylinder vs. a roughly cylindrical rock)
cancer	30	212	357	[53]	Characteristics of breast cancer tumors (benign vs. malignant)
plasticc	67	500	500	[54]	Photometric LSST Astronomical Time-series Classification Challenge dataset. Pre-processed by [22] (type II vs. Ia supernovae)

3. Results

3.1. Pre-Processing

Before evaluating the performance of the classifiers on the different datasets introduced above, we first present a study on different pre-processing strategies for the data. After an initial cleaning of the datasets (i.e., removing duplicates, defective samples, etc.) we performed the following pre-processing steps. First, we standardized the data in each feature x_i , by subtracting its mean μ_i and dividing by its standard deviation σ_i

$$x_i \mapsto \frac{x_i - \mu_i}{\sigma_i}. \quad (9)$$

Both the mean and standard deviation were calculated over the training data set, and the respective transformation was then also applied to the test data set. The standardized features were then projected to an n -dimensional feature vector using principal component analysis (PCA) [55]. In the literature, these two steps often represent the full pre-processing pipeline. However, consistent with the observations made in Reference [41], we demonstrated that subsequent scaling or normalization of the features has a beneficial effect on the performance of the resulting classifier. We compared the following three cases: The first option directly uses the principal components as input to the feature maps. The second option applies a global scaling factor λ to each principal component $x_i \mapsto \lambda x_i$, as in Reference [41]. The optimal scaling factor is determined via cross-validation. As a third option, we propose normalizing each feature to a fixed interval $[a, b]$ with min–max normalization

$$x_i \mapsto a + \frac{(x_i - x_{i,\min})(b - a)}{x_{i,\max} - x_{i,\min}}, \quad (10)$$

where $x_{i,\min}$ and $x_{i,\max}$ are the minimal and maximal values of feature x_i over the training set, respectively. The motivation for the latter was to map all features to a suitable *domain of rotations*. This ensures that all arguments given as inputs to a parameterized feature map lie within the range $[-\pi, \pi]$ (or alternatives thereof), such that the mapping becomes injective. In practice, we achieve mapping to the domain of rotations by normalizing each feature x_i to the interval $[0, 1]$, and by selecting the parameters in the feature maps accordingly.

For the RXRZ feature map, presented in Figure 1a, we chose parameters $\theta_{i,0} = \pi x_i$ and $\theta_{i,1} = 2\pi x_i$, as this guaranteed that the arguments to the R_X and R_Z rotations were in $[0, 2\pi]$. For the ZZ feature map, introduced in Figure 1b, we chose the parameters $\phi_{ij} = 2\pi x_i x_j$ and $\theta_i = \pi x_i$, which guaranteed that the arguments of the two-qubit rotations were in $[0, 2\pi]$, and the arguments of the single-qubit rotations were in $[0, \pi]$. The same expressions for the θ and ϕ parameters, including the π or 2π factors, were also used for the other pre-processing strategies.

To compare the different pre-processing pipelines, we conducted the following experiment, inspired by a similar study performed in Reference [43]. As a starting point, we calculated the kernel matrix $K_{ij} = K(x_i, x_j) = |\langle x_i | x_j \rangle|^2$ of 800 data samples in the fMNIST dataset for increasing numbers of features n (and, hence, also increasing number of qubits), and for increasing numbers of layers L in the feature map. We then evaluated the average and the variance across the entries in the resulting kernel matrix. The corresponding results are displayed in Figure 3A1–A4. The average and variance are shown for the kernels produced by applying the pre-processing with a final normalization step (solid lines) and without a final normalization step (dotted lines), using the RXRZ embedding (panels A1 and A3), or the ZZ embedding (panels A2 and A4). In all cases, the average and the variance decayed exponentially if no normalization was applied. This agreed with the results in Reference [43], where the authors demonstrated that increasing the expressivity of a feature map, and employing global fidelity measurements on uniformly sampled inputs, led to exponential concentration of the resulting kernel entries. However, the exponential concentration, as well as the exponential decay of the average kernel entries, are less pronounced when normalization to the domain of rotations is applied. For the ZZ feature map, we even observed a modest increase in the variance when increasing the number of features.

In Figure 3B1–B4 we illustrate an intuitive explanation of the effect of the pre-processing. The panels show the distribution of the first three principal components subjected to the different pre-processing strategies. As a visual guideline, the interval $[-\pi, \pi]$ is highlighted with the red dashed lines. The distributions of the non-normalized principal components mostly lay outside the highlighted interval (panel B1). Periodic mapping (such as that enforced by Pauli rotation gates) of these distributions into the highlighted interval led to distributions resembling a uniform distribution (panel B2). The original structure of the dataset was, thus, lost and the embedded data points hardly represented the original dataset faithfully. In fact, under these conditions the distribution of the input features tended to quickly become close to uniform over one period of the encoding Pauli rotations, such that some of the concentration hypotheses made in Reference [43], particularly for the case of global fidelity kernels, were essentially met. Applying a scaling factor $\lambda = 0.1$ to all principal components (the approach used in Reference [41]) led, instead, to the distributions displayed in panel B3. We observed that the most important regions of the distribution now lay within the highlighted interval, such that most of the structure would be preserved, even under periodicity. However, only our proposed min–max inspired option, namely the normalization to the domain of rotations, fully restricted each principal component individually to the interval $[-\pi, \pi]$ (panel B4). Compared to the distribution in B1, a periodic mapping to the highlighted interval preserved the distribution from B4, as all values were already in the correct domain. In summary, careful pre-processing represents a fast, direct and problem-agnostic method to mitigate kernel concentration effects leveraging the intrinsic data distribution, whenever that is present.

In the next section we describe the effects of the pre-processing on the performance of the resulting classifier.

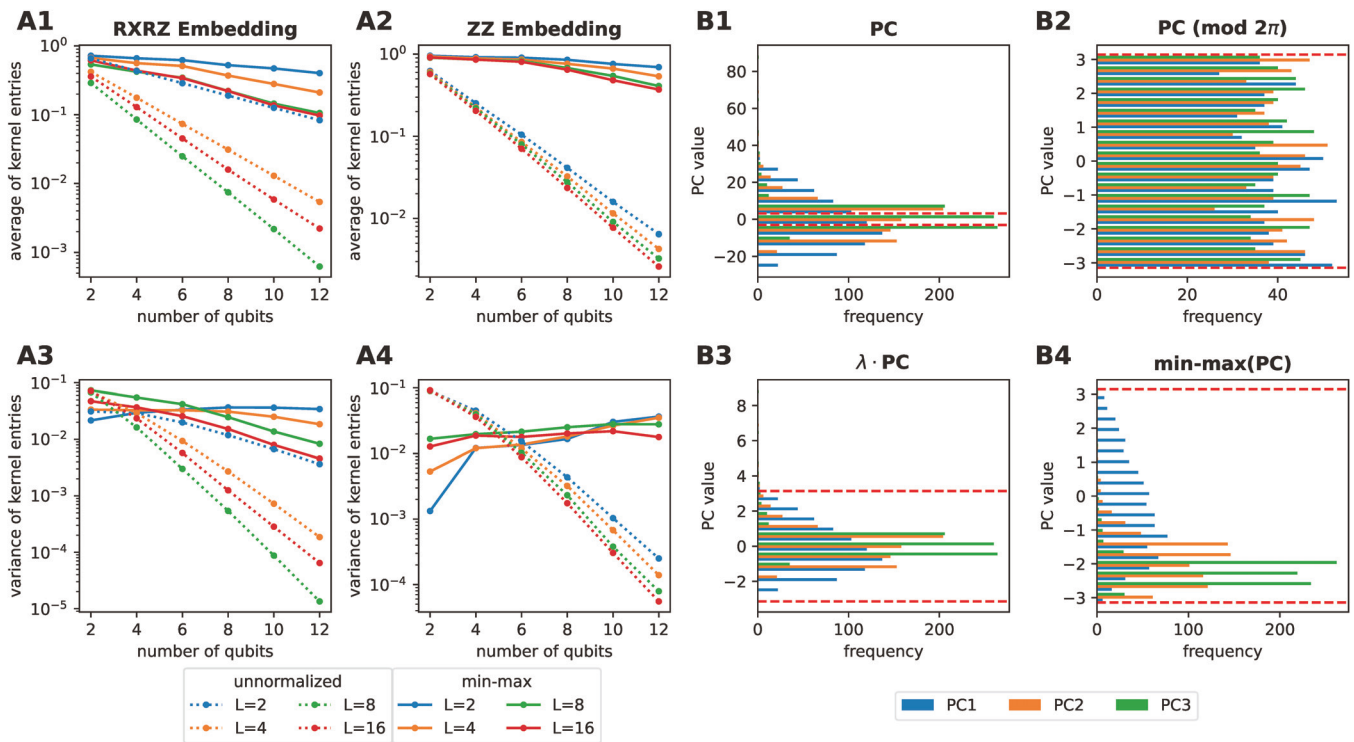


Figure 3. Illustration of the exponential concentration of the kernel values (panels (A1–A4)), and the effect of the pre-processing on the distributions of the input features (panels (B1–B4)). In panels (A1–A4), we used $L = 2, 4, 8, 16$ layers of the RXRZ or ZZ feature maps to embed 800 data samples in the fMNIST dataset. Panels (B1–B4) show the distribution of the first three principal components (PCs) of the considered data samples in the fMNIST data set, subject to different pre-processing strategies. The interval $[-\pi, \pi]$ is highlighted with red dotted lines. (B1) “Raw” principal components. (B2) “Raw” principal components, periodically mapped to the highlighted interval. (B3) Principal components scaled with $\lambda = 0.1$. (B4) Principal components normalized to the highlighted interval.

3.2. Classification

To evaluate the performance of the classifiers under study, we considered different standard metrics, namely the balanced accuracy, the receiver operator characteristic area under curve (ROC AUC) and the F1 score. The detailed definitions of these scoring methods are given in Appendix A. All the scores presented in the following were evaluated by averaging over 100 train–test splits (80–20%) of a specific dataset. In addition, if the scaling factor or RBF γ hyperparameters were to be determined, we performed five-fold cross validation on the train set of a split. The scaling factor was chosen from the interval $[10^{-3}, 1]$ and the γ parameter from $[10^{-5}, 10^3]$.

The ROC–AUC score and the balanced accuracy of all studied classifiers applied to the datasets in Table 1 are shown in Figure 4. All classifiers were built via the embedding of the pre-processed input features with the ZZ feature map. The blue, orange and green lines show the performances of the fidelity classifier resulting from the pre-processing of the data without normalization, scaling with an optimized scaling factor, or with normalization, respectively. For all datasets, applying no normalization to the principal components, resulted in classifiers that had, essentially, the same performance as a random classifier (value of 0.5 for both performance metrics). Applying a joint scaling factor to all principal components, led to considerable improvement in the performances of the classifiers for most datasets. However, in almost all cases, our proposed normalization to the domain of rotations (which effectively corresponds to applying an individual scaling factor to each feature), led to substantial improvements over the other two pre-processing strategies. For

the rest of the investigation, we, therefore, applied the normalization to the domain of rotations as the last step of the pre-processing pipeline.

As a next step, we look at the performance of the proposed RBF fidelity classifier (red lines in Figure 4). The effects of the simple post-processing of the fidelities, required to build the RBF classifier, were noticeable for all datasets. In fact, the RBF fidelity classifier performed better than the standard fidelity classifier (green lines) in almost all cases, with significant improvements for the fMNIST, cancer, and sonar datasets. Note that the cross-validation to find the best hyperparameter for the RBF fidelity classifier did not lead to an increase in the required quantum resources. The fidelity kernel only had to be evaluated once, and the cross-validation could then be performed classically.

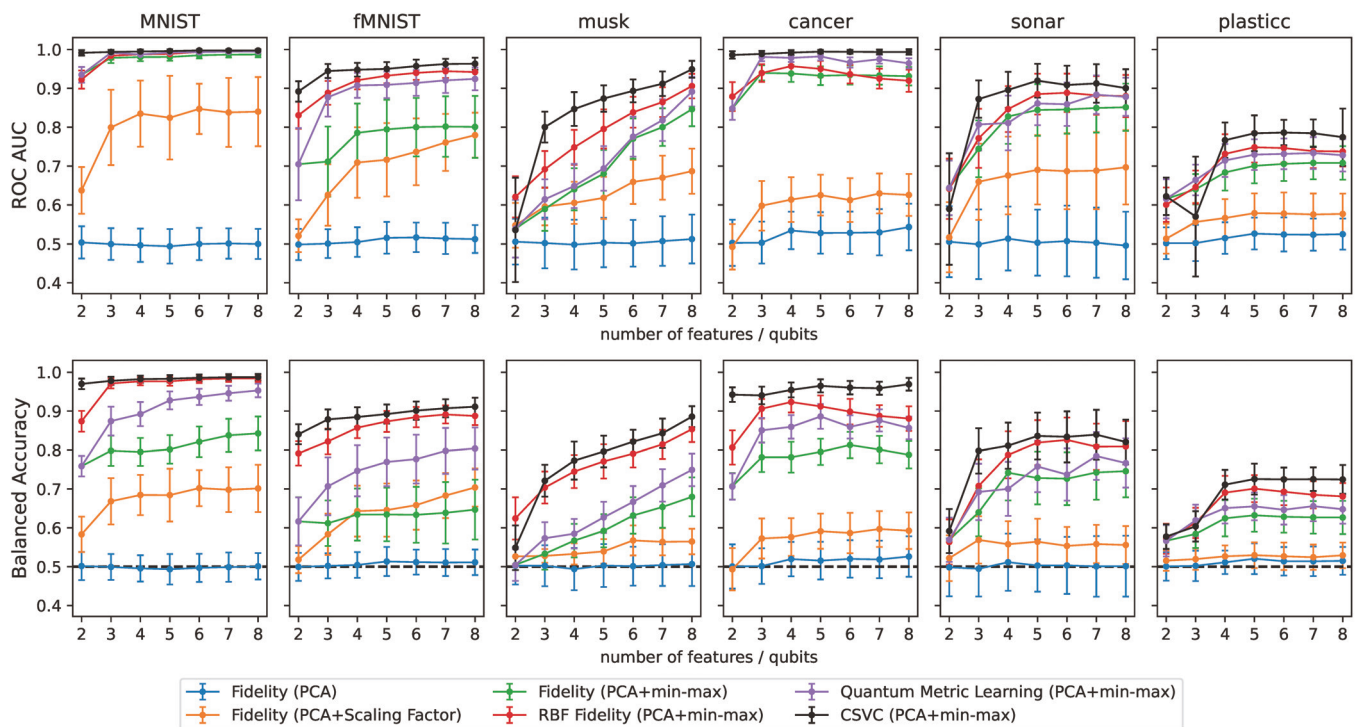


Figure 4. ROC–AUC and balanced accuracy for different datasets and increasing number of features. The blue, orange and green lines show the performance of the fidelity classifier resulting from the pre-processing of the input data without normalization, scaling with an optimized scaling factor, or with normalization, respectively. The red lines show the performance of the RBF fidelity classifier (using the same pre-processing steps as for the green line). The purple lines show the performance of the fidelity classifier with a trainable feature map (using the same pre-processing steps as for the green line). For all classifiers, the ZZ feature map (or its trainable version), was used for the embedding of the data points. The black lines show the performance of the classical support vector classifier. In the bottom row, the black dashed lines show the balanced accuracy achieved by a random classifier.

Finally, the performance of the fidelity classifier with a trainable feature map, is depicted in Figure 4 with the purple lines. The approach led to noticeable improvements over the standard fidelity classifier for the MNIST, fMNIST, musk and cancer datasets. The effect was less pronounced for the remaining datasets. Nonetheless, the results still showcase the potential of tailoring the feature map to the considered classification task. However, compared to the RBF fidelity classifier, the trainable fidelity classifier performed worse in most of the cases.

For completeness, we also provide, in Figure 4, the results obtained for all datasets with a classical Support Vector Classifier (CSVC) featuring either a linear or an RBF kernel (where all hyperparameters were chosen by five-fold cross-validation). While the performances of the CSVC still represented an upper bound, a quantum classifier with our proposed

improvements often achieved comparable classification scores. Overall, these findings suggest that our techniques could become part of a toolbox aimed at maximising empirical performances of quantum classification algorithms, and, hence, provide opportunities for quantum advantage, when scaling up the size of practical applications.

4. Discussion and Conclusions

The results presented in this study demonstrate the importance of proper data pre-processing in quantum machine learning, specifically in the context of quantum kernel methods for classification. Our experiments, conducted on a variety of structured real-world datasets, showed that the absence of a suitable normalization procedure could lead to essentially random quantum embeddings, characterized by a loss of the relationships between the data, exponential concentration in the kernel values and, most notably, poor classification performance. We illustrated the effect of different feature normalization strategies using various scaling methods, and demonstrated that our proposed normalization approach consistently led to improved performance across all tested datasets and all numbers of principal components. It is also worth mentioning that, while an optimized global scaling factor controlling the kernel bandwidth [41] can, in general, only be found through cross-validation, requiring several kernel evaluations for multiple train–test splits, the normalization approach is defined solely by the input data, and can, therefore, be applied without any substantial computational overhead.

We also investigated the effect of non-linear post-processing of the fidelity quantum kernel entries through exponentiation, yielding an original and effective RBF-like quantum kernel, and of quantum metric learning across a broad range of application domains. In both cases, for a representative collection of datasets, significant improvements, in terms of classification performances with respect to standard quantum methods, were observed. As a result, we conclude that these approaches constitute an effective and relatively inexpensive toolbox that could be applied in many realistic scenarios to systematically improve the performances of quantum classification algorithms.

Author Contributions: Conceptualization, M.J., J.S., P.B., I.T. and F.T.; formal analysis, M.J., J.S., P.B. and F.T.; investigation, M.J. and J.S.; data curation, M.J. and J.S.; writing—original draft preparation, M.J., J.S. and F.T.; writing—review and editing, P.B. and I.T.; visualization, M.J., J.S. and P.B.; supervision, P.B., I.T. and F.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the NCCR MARVEL, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 205602).

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request.

Acknowledgments: IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. The current list of IBM trademarks is available at <https://www.ibm.com/legal/copytrade>.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Performance Metrics

Most scoring methods can be derived from a so-called confusion matrix. For binary classification, such a confusion matrix summarizes the classification performance with four values:

- true positives (TP)—number of positive samples classified as positive
- false positives (FP)—number of negative samples classified as positive
- true negatives (TN)—number of negative samples classified as negative
- false negatives (FN)—number of positive samples classified as negative

For clarity, we also introduce the total number of positive samples P and the total number of negative samples N .

The most intuitive method to evaluate performance is to look at how many samples out of the entire test set were correctly classified. Accuracy is then defined as

$$a = \frac{TP + TN}{TP + FP + TN + FN} = \frac{TP + TN}{P + N}. \quad (A1)$$

This score is well suited for those cases in which the dataset is balanced (same number of positives and negatives) and if both classes are of equal importance. However, it can become a misleading figure of merit on imbalanced datasets. As an alternative, balanced definition of accuracy takes into account an average over each class. For binary classification, we define the accuracy on the positive samples as the true positive rate $TPR = TP/P$ and the accuracy on the negative samples as the true negative rate $TNR = TN/N$. The balanced accuracy can then be defined as

$$a_{\text{balanced}} = \frac{TPR + TNR}{2}. \quad (A2)$$

The Receiver Operator Characteristic Area Under Curve (ROC–AUC) is a metric that, in addition to classification of performances, also conveys information about the robustness of the model. In a nutshell, the ROC follows the true positive rate and the false positive rate while continuously moving the decision boundary of the classifier from an extreme condition where all data are classified as negative to the opposite scenario, where all data are considered positive. The area under the ROC curve is then a measure of the overall quality of the classifier. A completely random classifier on a balanced dataset achieves an ROC–AUC of 0.5.

The F1 score is the harmonic mean of the precision (how many out of all positively classified samples are positive) and the true positive rate (how many out of the positive samples were classified as positive). Ideally, our classifier would achieve a high score on both the precision and the TPR—In fact, an ideal model only classifies positive data points as positives and at the same time finds all the positives. Typically there is a trade-off between the two requirements, which is summarized by the F1 score

$$F1 = \frac{2TP}{(TP + FP) + (TP + FN)} \quad (A3)$$

As this is a robust scoring method, we used it to evaluate the cross-validation on all datasets.

References

1. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [CrossRef] [PubMed]
2. Cerezo, M.; Verdon, G.; Huang, H.Y.; Cincio, L.; Coles, P.J. Challenges and opportunities in quantum machine learning. *Nat. Comput. Sci.* **2022**, *2*, 567–576. [CrossRef]
3. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
4. Durr, C.; Hoyer, P. A quantum algorithm for finding the minimum. *arXiv* **1996**, arXiv:quant-ph/9607014.
5. Farhi, E.; Goldstone, J.; Gutmann, S.; Lapan, J.; Lundgren, A.; Preda, D. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **2001**, *292*, 472–475. [CrossRef]
6. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [CrossRef]
7. Neven, H.; Denchev, V.S.; Rose, G.; Mcready, W.G. Training a large scale classifier with the quantum adiabatic algorithm. *arXiv* **2009**, arXiv:0912.0779.
8. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef]
9. Schuld, M.; Sinayskiy, I.; Petruccione, F. The quest for a Quantum Neural Network. *Quantum Inf. Process.* **2014**, *13*, 2567–2586. [CrossRef]
10. Farhi, E.; Neven, H. Classification with quantum neural networks on near term processors. *arXiv* **2018**, arXiv:1802.06002.
11. Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **2019**, *4*, 043001. [CrossRef]

12. Tacchino, F.; Barkoutsos, P.; Macchiavello, C.; Tavernelli, I.; Gerace, D.; Bajoni, D. Quantum implementation of an artificial feed-forward neural network. *Quantum Sci. Technol.* **2020**, *5*, 044010. [CrossRef]
13. Mangini, S.; Tacchino, F.; Gerace, D.; Bajoni, D.; Macchiavello, C. Quantum computing models for artificial neural networks. *EPL Europhys. Lett.* **2021**, *134*, 10002. [CrossRef]
14. Tacchino, F.; Mangini, S.; Barkoutsos, P.K.; Macchiavello, C.; Gerace, D.; Tavernelli, I.; Bajoni, D. Variational Learning for Quantum Artificial Neural Networks. *IEEE Trans. Quantum Eng.* **2021**, *2*, 1–10. [CrossRef]
15. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
16. Abbas, A.; Sutter, D.; Zoufal, C.; Lucchi, A.; Figalli, A.; Woerner, S. The power of quantum neural networks. *Nat. Comput. Sci.* **2021**, *1*, 403–409. [CrossRef]
17. Liu, J.; Najafi, K.; Sharma, K.; Tacchino, F.; Jiang, L.; Mezzacapo, A. Analytic Theory for the Dynamics of Wide Quantum Neural Networks. *Phys. Rev. Lett.* **2023**, *130*, 150601. [CrossRef]
18. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. [CrossRef]
19. Schuld, M.; Killoran, N. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [CrossRef]
20. Lloyd, S.; Schuld, M.; Ijaz, A.; Izaac, J.; Killoran, N. Quantum embeddings for machine learning. *arXiv* **2020**, arXiv:2001.03622.
21. Liu, Y.; Arunachalam, S.; Temme, K. A rigorous and robust quantum speed-up in supervised machine learning. *Nat. Phys.* **2021**, *17*, 1013–1017. [CrossRef]
22. Peters, E.; Caldeira, J.; Ho, A.; Leichenauer, S.; Mohseni, M.; Neven, H.; Spentzouris, P.; Strain, D.; Perdue, G.N. Machine learning of high dimensional data on a noisy quantum processor. *npj Quantum Inf.* **2021**, *7*, 161. [CrossRef]
23. Huang, H.Y.; Broughton, M.; Mohseni, M.; Babbush, R.; Boixo, S.; Neven, H.; McClean, J.R. Power of data in quantum machine learning. *Nat. Commun.* **2021**, *12*, 2631. [CrossRef] [PubMed]
24. Jerbi, S.; Fiderer, L.J.; Poulsen Nautrup, H.; Kübler, J.M.; Briegel, H.J.; Dunjko, V. Quantum machine learning beyond kernel methods. *Nat. Commun.* **2023**, *14*, 517. [CrossRef]
25. Wu, S.L.; Sun, S.; Guan, W.; Zhou, C.; Chan, J.; Cheng, C.L.; Pham, T.; Qian, Y.; Wang, A.Z.; Zhang, R.; et al. Application of quantum machine learning using the quantum kernel algorithm on high energy physics analysis at the LHC. *Phys. Rev. Res.* **2021**, *3*, 033221. [CrossRef]
26. Schuhmacher, J.; Boggia, L.; Belis, V.; Puljak, E.; Grossi, M.; Pierini, M.; Vallecorsa, S.; Tacchino, F.; Barkoutsos, P.; Tavernelli, I. Unravelling physics beyond the standard model with classical and quantum anomaly detection. *arXiv* **2023**, arXiv:2301.10787.
27. Woźniak, K.A.; Belis, V.; Puljak, E.; Barkoutsos, P.; Dissertori, G.; Grossi, M.; Pierini, M.; Reiter, F.; Tavernelli, I.; Vallecorsa, S. Quantum anomaly detection in the latent space of proton collision events at the LHC. *arXiv* **2023**, arXiv:2301.10780.
28. Sancho-Lorente, T.; Román-Roche, J.; Zueco, D. Quantum kernels to learn the phases of quantum matter. *Phys. Rev. A* **2022**, *105*, 042432. [CrossRef]
29. Grossi, M.; Ibrahim, N.; Radescu, V.; Loredò, R.; Voigt, K.; Von Altrock, C.; Rudnik, A. Mixed Quantum–Classical Method for Fraud Detection With Quantum Feature Selection. *IEEE Trans. Quantum Eng.* **2022**, *3*, 1–12. [CrossRef]
30. Mensa, S.; Sahin, E.; Tacchino, F.; Barkoutsos, P.K.; Tavernelli, I. Quantum machine learning framework for virtual screening in drug discovery: A prospective quantum advantage. *Mach. Learn. Sci. Technol.* **2023**, *4*, 015023. [CrossRef]
31. Li, Y.; Benjamin, S.C. Efficient Variational Quantum Simulator Incorporating Active Error Minimization. *Phys. Rev. X* **2017**, *7*, 021050. [CrossRef]
32. Temme, K.; Bravyi, S.; Gambetta, J.M. Error Mitigation for Short-Depth Quantum Circuits. *Phys. Rev. Lett.* **2017**, *119*. [CrossRef] [PubMed]
33. Endo, S.; Benjamin, S.C.; Li, Y. Practical Quantum Error Mitigation for Near-Future Applications. *Phys. Rev. X* **2018**, *8*, 031027. [CrossRef]
34. Earnest, N.; Tornow, C.; Egger, D.J. Pulse-efficient circuit transpilation for quantum applications on cross-resonance-based hardware. *Phys. Rev. Res.* **2021**, *3*, 043088. [CrossRef]
35. Kim, Y.; Wood, C.J.; Yoder, T.J.; Merkel, S.T.; Gambetta, J.M.; Temme, K.; Kandala, A. Scalable error mitigation for noisy quantum circuits produces competitive expectation values. *Nat. Phys.* **2023**, *19*, 752–759. [CrossRef]
36. Melo, A.; Earnest-Noble, N.; Tacchino, F. Pulse-efficient quantum machine learning. *arXiv* **2022**, arXiv:2211.01383.
37. McClean, J.R.; Boixo, S.; Smelyanskiy, V.N.; Babbush, R.; Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **2018**, *9*, 4812. [CrossRef]
38. Cerezo, M.; Sone, A.; Volkoff, T.; Cincio, L.; Coles, P.J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat. Commun.* **2021**, *12*, 1791. [CrossRef]
39. Thanasilp, S.; Wang, S.; Nghiem, N.A.; Coles, P.J.; Cerezo, M. Subtleties in the trainability of quantum machine learning models. *arXiv* **2021**, arXiv:2110.14753.
40. Kübler, J.; Buchholz, S.; Schölkopf, B. The inductive bias of quantum kernels. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12661–12673.
41. Shaydulin, R.; Wild, S.M. Importance of kernel bandwidth in quantum machine learning. *Phys. Rev. A* **2022**, *106*, 042407. [CrossRef]
42. Canatar, A.; Peters, E.; Pehlevan, C.; Wild, S.M.; Shaydulin, R. Bandwidth enables generalization in quantum kernel models. *arXiv* **2022**, arXiv:2206.06686.

43. Thanasilp, S.; Wang, S.; Cerezo, M.; Holmes, Z. Exponential concentration and untrainability in quantum kernel methods. *arXiv* **2022**, arXiv:2208.11060.
44. Glick, J.R.; Gujarati, T.P.; Corcoles, A.D.; Kim, Y.; Kandala, A.; Gambetta, J.M.; Temme, K. Covariant quantum kernels for data with group structure. *arXiv* **2021**, arXiv:2105.03406.
45. Shashua, A. Introduction to machine learning: Class notes 67577. *arXiv* **2009**, arXiv:0904.3664.
46. Farhi, E.; Harrow, A.W. Quantum Supremacy through the Quantum Approximate Optimization Algorithm. *arXiv* **2019**, arXiv:1602.07674.
47. Barenco, A.; Berthiaume, A.; Deutsch, D.; Ekert, A.; Jozsa, R.; Macchiavello, C. Stabilization of quantum computations by symmetrization. *SIAM J. Comput.* **1997**, *26*, 1541–1557. [CrossRef]
48. Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
49. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv* **2017**, arXiv:1708.07747.
50. Dietterich, T.G.; Lathrop, R.H.; Lozano-Pérez, T. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.* **1997**, *89*, 31–71. [CrossRef]
51. Romano, J.D.; Le, T.T.; La Cava, W.; Gregg, J.T.; Goldberg, D.J.; Chakraborty, P.; Ray, N.L.; Himmelstein, D.; Fu, W.; Moore, J.H. PMLB v1. 0: An open-source dataset collection for benchmarking machine learning methods. *Bioinformatics* **2022**, *38*, 878–880. [CrossRef]
52. Gorman, R.P.; Sejnowski, T.J. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Netw.* **1988**, *1*, 75–89. [CrossRef]
53. Wolberg, W.; Mangasarian, O.; Street, N. Breast Cancer Wisconsin (Diagnostic). In *UCI Machine Learning Repository*; UCI: Irvine, CA, USA, 1995. [CrossRef]
54. Kessler, R.; Narayan, G.; Avelino, A.; Bachelet, E.; Biswas, R.; Brown, P.; Chernoff, D.; Connolly, A.; Dai, M.; Daniel, S.; et al. Models and simulations for the photometric LSST astronomical time series classification challenge (PLAsTiCC). *Publ. Astron. Soc. Pac.* **2019**, *131*, 094501. [CrossRef]
55. Jolliffe, I.T. *Principal Component Analysis for Special Types of Data*; Springer: Berlin/Heidelberg, Germany, 2002.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Link Prediction with Continuous-Time Classical and Quantum Walks

Mark Goldsmith^{1,2,*}, Harto Saarinen^{1,2,*}, Guillermo García-Pérez^{1,2,3,4}, Joonas Malmi^{1,3,4}, Matteo A. C. Rossi^{1,3,5,6} and Sabrina Maniscalco^{1,2,3,4,5,6}

¹ Algorithmiq Ltd., Kanavakatu 3 C, FI-00160 Helsinki, Finland

² Complex Systems Research Group, Department of Mathematics and Statistics, University of Turku, FI-20014 Turku, Finland

³ QTF Centre of Excellence, Department of Physics, Faculty of Science, University of Helsinki, FI-00014 Helsinki, Finland

⁴ InstituteQ-The Finnish Quantum Institute, University of Helsinki, FI-00014 Helsinki, Finland

⁵ QTF Centre of Excellence, Department of Applied Physics, Aalto University, FI-00076 Aalto, Finland

⁶ InstituteQ-The Finnish Quantum Institute, Aalto University, FI-00076 Aalto, Finland

* Correspondence: mark@algorithmiq.fi (M.G.); harto@algorithmiq.fi (H.S.)

Abstract: Protein–protein interaction (PPI) networks consist of the physical and/or functional interactions between the proteins of an organism, and they form the basis for the field of network medicine. Since the biophysical and high-throughput methods used to form PPI networks are expensive, time-consuming, and often contain inaccuracies, the resulting networks are usually incomplete. In order to infer missing interactions in these networks, we propose a novel class of link prediction methods based on continuous-time classical and quantum walks. In the case of quantum walks, we examine the usage of both the network adjacency and Laplacian matrices for specifying the walk dynamics. We define a score function based on the corresponding transition probabilities and perform tests on six real-world PPI datasets. Our results show that continuous-time classical random walks and quantum walks using the network adjacency matrix can successfully predict missing protein–protein interactions, with performance rivalling the state-of-the-art.

Keywords: link prediction; protein–protein interaction networks; random walks; quantum walks

Citation: Goldsmith, M.; Saarinen, H.; García-Pérez, G.; Malmi, J.; Rossi, M.A.C.; Maniscalco, S. Link Prediction with Continuous-Time Classical and Quantum Walks.

Entropy **2023**, *25*, 730. <https://doi.org/10.3390/e25050730>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 14 March 2023

Revised: 21 April 2023

Accepted: 26 April 2023

Published: 28 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The link prediction problem has long been an active area of research, with applications ranging from friendship recommendation in social networks [1–3] to finding missing interactions between proteins [4,5]. In this paper, we were interested in the latter. For general surveys in link prediction, we refer to [6–8].

One particularly successful class of link prediction methods is those based on random walks [5,9,10]. Random walk algorithms have been explored more generally throughout the field of network science, and many different applications exist. These include the ranking of web pages using PageRank [11,12], collaborative filtering [13], and computer vision [14]. Many random walk link prediction algorithms have also been studied [5,15]. These methods typically rely on discrete-time random walks.

In contrast, in this paper, we propose a class of link prediction methods based on continuous-time random walks. Moreover, the continuous-time setting allowed us to propose a new link prediction method using quantum walks, which closely resembles the classical method described here.

Continuous-time quantum walks, initially proposed in [16], are the quantum analogues of continuous-time classical random walks, which describe the propagation of a particle over a discrete set of positions. Together with their discrete-time counterpart [17], they have received much attention for their applications in quantum information processing [18,19], quantum computation [20], and quantum transport [21]. However, only

a few recent methods have attempted to use quantum walks for link prediction, using their discrete-time [22] and continuous-time [23] variations. While the methods described here are *quantum-inspired*, since they were implemented classically, we can foresee that these will be even more efficient if run on quantum devices. Continuous-time quantum walks have already been implemented on various physical platforms [24], including optical setups [25–29] and superconducting devices [30,31], and they can also be simulated on gate-based quantum computers [32,33].

In order to evaluate our proposed methods, we conducted experiments on several networks and found that both the classical and quantum walks outlined here are particularly good at finding missing links in protein–protein interaction (PPI) networks. Protein–protein interactions play a critical role in all cellular processes, ranging from cellular division to apoptosis. Elucidating and analysing PPIs is thus essential to understand the underlying mechanisms in biology and, eventually, to unveil the molecular roots of human disease [34]. Indeed, this has been a major focus of research in recent years, providing a wealth of experimental data about protein associations [35,36]. Current PPI networks, called interactomes, have been constructed using a number of techniques, but despite the enormous advancement, the current coverage of PPIs is still rather poor (for example, it is estimated that only around 10% of interactions in humans are currently known [37]). Additionally, despite considerable improvements in high-throughput (HTP) techniques, they are still prone to spurious errors and systematic biases, yielding a significant number of false positives and false negatives. This limitation impedes our ability to assess the true quality and coverage of the interactome.

Recently, a number of algorithms have been developed to predict protein–protein interactions. In a recent study by Kovács et al. [4] (see also [38,39]), a novel PPI-specific link predictor was proposed. Their link predictor was biologically motivated by the so-called L3 principle, and it was shown to be superior to other general link predictors when applied to PPI data. The exceptional success of the L3 framework is rooted in its ability to capture the structural and evolutionary principles that drive PPIs. The results of Kovács and collaborators proved that, contrary to the current network paradigm, interacting proteins are not necessarily similar and similar proteins do not necessarily interact, questioning the traditional validation strategy based on the biological similarity of the predicted protein pairs.

However, the L3 link prediction method, considered the most-successful to date for PPIs, as well as most other existing link prediction methods are not without limitations. The most-common approaches cannot find interactions for self-interacting proteins or links between proteins that have long shortest paths between them. Given the low coverage of the current PPI databases, this can be a significant drawback. It is, therefore, highly desirable to complement the existing frameworks with methods relying on the exploration of the whole network, and consequently be able to predict edges whose corresponding nodes may be far away in the network. Thus, we propose novel quantum- and classical-random-walk-based link prediction methods that can potentially traverse the entire network and simultaneously predict self-edges.

2. Materials and Methods

Consider a network modelled by an undirected and unweighted graph $G = (V, E)$, where V is the set of nodes of size n and E is the set of edges. We allowed for the existence of self-edges, so that for any node i , the edge (i, i) may or may not be present in E . The *adjacency matrix* of G is the $n \times n$ matrix defined by

$$A = (A_{ij}) = \begin{cases} 1, & \text{if } (i, j) \in E, \\ 0, & \text{if } (i, j) \notin E. \end{cases}$$

The *graph Laplacian* is defined as $L = D - A$, where D is the *degree matrix* defined by $D = \text{diag}(\sum_j A_{1j}, \dots, \sum_j A_{nj})$.

The link prediction problem is to infer missing links in a network G , using only the information provided by the structure of G . Thus, a link prediction algorithm typically gives a ranking of all the non-edges (pairs of nodes that are not directly connected in G) based on some proposed scoring scheme.

We now present a rather general scoring scheme for ranking the non-edges of a graph based on state transition probabilities resulting from quantum and classical random walks; the precise details of the walks we employed are described in the next subsections. For now, it suffices to consider the notion of a probability transition matrix that evolves over time, denoted by $P(t)$; for a graph G , the probability of the walker being at node v at time t , given that it began at node u , is thus $P_{uv}(t)$. For a fixed time t , we define the score $S(i, j; t)$ between two non-adjacent nodes i and j at time t to be

$$S(i, j; t) = \begin{cases} P_{ij}(t)(k_i + k_j) & i \neq j \quad (1) \\ \frac{1}{2} \sum_{u \in N(i)} P_{iu}(t) & i = j, \quad (2) \end{cases}$$

where $N(v)$ denotes the set of nodes adjacent to v (possibly including v itself) and $k_v = \sum_j A_{vj}$ is the degree of node v . Equations (1) and (2) handle the cases of distinct nodes and self-edges, respectively. The scoring scheme in Equation (1) is based on the intuition that two nodes i and j should likely be connected if the walk is more likely to move from i to j than to other nodes. We also scale these probabilities by the node degrees so that high-degree nodes have a higher preference, similar to the preferential attachment link prediction method [40,41]. Further, Equation (2) claims that the properties of the walker in the neighbourhood of the node determines the likelihood of a self-edge. While the score in Equation (1) is superficially similar to the one proposed in [5], the fact that we use continuous-time walks leads to several key differences: the continuous-time nature of our method allows for a wider range of time parameters t to use; in the continuous-time setting, there is symmetry in the transition probabilities, i.e., $P_{ij}(t) = P_{ji}(t)$ for all nodes i, j ; finally, there is a close relationship in the implementation of classical and quantum walks in the continuous-time setting.

Regardless of which type of walk is used, we must choose a value t , representing the time duration of the walk. We start the walk at time $t_0 = 0$ and let it run for a time t , at which point we extract the scores for the target edges from the probability distributions. In the case of a continuous-time classical random walk, the expected time it takes for a random walker to leave a node i is $1/k_i$. This motivates the idea that the amount of time we let the walk run should be related to the degree distribution of the network. In our experiments, we tested a few small multiples of the value $1/\langle k \rangle$, where $\langle k \rangle$ is the average node degree in the graph, and report the value yielding the best results (see the results in Section 3).

2.1. Continuous-Time Random Walks

A continuous-time (classical) random walk (CRW) is a Markov process with state space V characterised by an initial distribution $\mathbf{p}(0)$ over the set of nodes and a rate matrix Q that has null row sum $\sum_k Q_{jk} = 0$ for all j . Here, we considered edge-based random walks [42] (as opposed to node-based), which are characterised by setting $Q = -L$, where L is the Laplacian of the underlying graph. In this case, the evolution of the probability vector $\mathbf{p}(t)$ is governed by the equation:

$$\mathbf{p}(t) = \mathbf{p}(0)\mathbf{P}(t), \tag{3}$$

where $\mathbf{P}(t) = e^{-tL}$ is the probability transition matrix, which has the elements $P_{ij}(t) = \langle j | e^{-itL} | i \rangle$, where i and j are standard basis vectors.

Intuitively, the random walker operates as follows. Every edge of the graph is associated with an independent Poisson process with unit intensity. When the walker is at some

node, it will remain there until one of the Poisson processes at an incident edge jumps, at which point, the walker follows that edge to the corresponding neighbour, and the process repeats. Note that this implies that, on average, a random walker will spend less time waiting at a higher-degree node than at a lower-degree node. Furthermore, this method will assign non-zero probabilities to all pairs of nodes in a connected component, due to the continuous-time nature of the walk.

2.2. Continuous-Time Quantum Walks

In contrast to a classical random walk, a quantum walk on a network evolves according to the laws of quantum physics. A major implication of this is that the trajectories of the walker across the network can interfere constructively or destructively. This interference causes the evolution of the quantum walker to sometimes be significantly different from the classical one [17,43].

A continuous-time quantum walk (QW) [16] on a graph G is defined by considering the Hilbert space \mathcal{H} spanned by the orthonormal vectors $\{|i\rangle\}_{i=1}^n$, corresponding to the n nodes of the graph and the unitary transformation $U(t)$. This transformation implies that the state vector in \mathcal{H} at a time t after starting from initial time $t_0 = 0$ is given by the evolution:

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle, \quad (4)$$

where $U(t) = e^{-itH}$ is the unitary evolution operator and H is the Hamiltonian. In general, the Hamiltonian H can be almost any Hermitian matrix related to G as long as it describes the structure of the network [19], but the most-common choices are the graph adjacency matrix A or the Laplacian L [44]. We also note that, in the classical random walk, the rate matrix Q is required to have a null row sum so that it is probability-conserving, and thus, the Laplacian L is a valid choice. However, for quantum walks, no such restriction exists, and a wider range of walks can be considered by modifying the Hamiltonian, as long as it remains Hermitian [45]. For example, the graph adjacency matrix can be used as a Hamiltonian, but not as a classical rate matrix since its rows do not sum to zero. In this paper, we used both the adjacency and Laplacian matrices as the Hamiltonians separately and, therefore, can compare different realisations of quantum walks for the link prediction task.

In order to obtain a probability transition matrix analogous to the one in Equation (3), we must take the square of the modulus of the entries of $U(t)$. The entries of the probability transition matrix are given by

$$P_{ij}(t) = |\langle j|e^{-itH}|i\rangle|^2. \quad (5)$$

These transition probabilities can then be used to compute scores for non-edges as described in Equations (1) and (2) above. Note that, contrary to the classical case, where randomness comes from stochastic transitions between states, in the quantum walk, the state transitions are deterministically governed by the Schrödinger equation, and the randomness results from the measurement and collapse of the wave function.

Our motivation for the usage of continuous- rather than discrete-time walks is three-fold: there is a close resemblance between the classical and quantum versions via the matrix exponential, which allows both methods to be easily compared; having a real, rather than an integer-valued hyperparameter t allows for a wider range of results to be explored and also permits non-zero scores to be assigned to all pairs of non-neighbouring nodes within a connected component. We emphasise that the usage of continuous-time quantum walks for link prediction is a new direction of research, with very few studies conducted so far. The method proposed in [23], in particular, appears to be competitive with some state-of-the-art link prediction methods in certain real networks. While some aspects of their algorithm are similar to the quantum version of our algorithm, the implementation details and calculation of the link prediction scores are very different. Moreover, their algorithm requires entanglement with an additional ancilla. While this would be feasible

in a hypothetical implementation on a quantum computer, the typical sizes of relevant real networks are far beyond the capabilities of current and near-term quantum hardware. Simulations on classical computers are required, but the presence of the extra ancilla increases the complexity of the simulations.

2.3. Datasets and Metrics

We tested our link prediction methods on six different PPI networks. Four networks were *Homo sapiens* (human) PPI networks: we used the physical, multi-validated interactions from v4.4.219 of BioGRID [46], the high-quality binary and co-complex interactions from the HINT database [47], the interactions proven by 2 or more pieces of experimental evidence from APID [48,49] (downloaded on 1 March 2023), and the experimentally validated interactions from the Integrated Interactions Database (IID) [50], Version 2021-05. Furthermore, we also tested our methods on the interactions of the organism *Saccharomyces cerevisiae* (yeast) from BioGRID and HINT just described.

Some statistics of these networks are listed below in Table 1, and their degree distributions are shown in Figure 1. We observed from these statistics that the networks have high clustering and that they are very sparse. Furthermore, the networks are approximately scale-free [51], which is typical of biological networks. One distinguishing feature of PPI networks compared to most other complex networks is that they contain self-edges, which represent the ability of a protein to interact with itself.

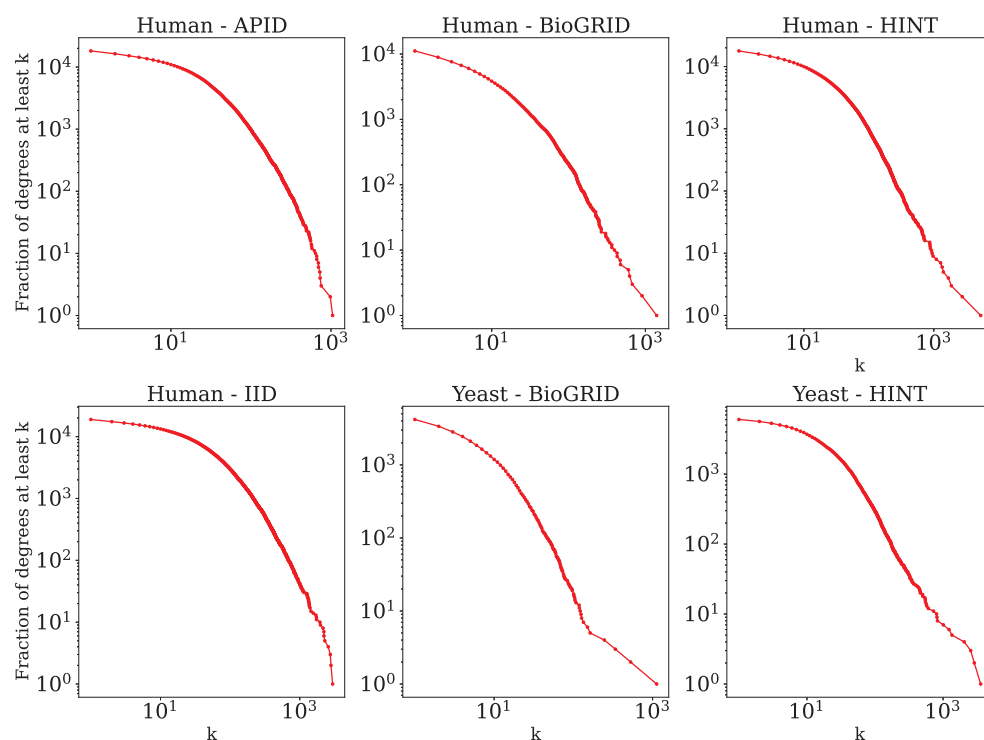


Figure 1. Complementary cumulative degree distributions. For each degree value k (x -axis), the proportion of nodes with degree greater than or equal to k (y -axis) is shown, each on a logarithmic scale.

Since the ground truth of the considered PPI networks is of course unknown, we proceeded to test the algorithms using cross-validation. For each dataset, we randomly removed $P\%$ of the edges in the original network, for $P \in \{10, 20, 30, 40, 50\}$, and reserved these edges as positive test cases. All of the non-edges (including self-edges that are not present in the network) were used as negative testing data. These positive and negative edges were used to evaluate the methods, and the remaining $(100 - P)\%$ existing edges were used for running the models in question. In other words, after removing the $P\%$ of the edges, the non-edges were ranked by sorting them in descending order according to their

scores, and the edges with higher scores were deemed most likely to exist. This ranking was then compared to the evaluation set to see how well the positive test cases were ranked. This process was repeated 10 times for each P , and the results of the accuracy metrics were averaged (see the results in Section 3).

Table 1. Some properties of the networks that were tested. $|V|$: number of nodes, $|E|$: number of edges, $\langle k \rangle$: average degree, ρ : network density, C : average clustering, A : assortativity, SIPs: number of self-interacting proteins (self-edges).

Network	$ V $	$ E $	$\langle k \rangle$	ρ	C	A	SIPs
Yeast-BioGRID	4186	20,053	9.581	0.002	0.306	−0.080	826
Yeast-HINT	6025	92,201	30.606	0.005	0.304	−0.129	1837
Human-BioGRID	11,134	79,536	14.287	0.001	0.200	−0.063	1254
Human-HINT	17,818	256,972	28.844	0.002	0.129	−0.059	5223
Human-APID	18,173	265,216	29.188	0.002	0.086	−0.082	2488
Human-IID	18,925	560,628	59.247	0.003	0.126	−0.085	4684

In order to compare the rankings of the edges of the methods under consideration, we used the areas under the precision–recall and receiver operator characteristic curves, two metrics that are typically used in link prediction and other binary classification problems. Hence, we define

$$\begin{aligned} \text{true positive rate} = \text{recall} &= \frac{TP}{TP + FN}, \\ \text{precision} &= \frac{TP}{TP + FP}, \\ \text{false positive rate} &= \frac{FP}{FP + TN}, \end{aligned}$$

where TP = true positive, FP = false positive, FN = false negative, and TN = true negative. In order to calculate each of these from the rankings, a threshold that serves as a cut-off rule has to be selected (the predictions above the thresholds are classified as positive and below it as negative). Our two metrics were calculated by varying this threshold through the rankings. Firstly, we considered the area under the precision–recall curve (AuPR). Precision–recall curves plot the recall on the x -axis against precision on the y -axis. In order to reduce this curve to a single number, the area under the curve is used, and this also circumvents the problem of choosing an arbitrary score threshold at which to distinguish predicted positives from negatives. Note that the AuPR focuses only on performance relative to the positive class, an important consideration when the ratio of positive cases to negative cases is small, as is the case in most networks and especially in PPI networks (these networks are extremely sparse; see Table 1). As a secondary metric, we considered the area under the receiver operating characteristic curve (AuROC) [52], which plots the false positive rate versus the recall. It can be interpreted as the probability that the classifier will rank a positive case, chosen uniformly at random from the positive set, higher than a negative one, chosen uniformly at random from the negative set [53]. Thus, a random classifier has an AuROC equal to half and a perfect classifier has an AuROC equal to one. We emphasise that the AuPR is widely accepted as the preferred metric for link prediction, due to the large class imbalance mentioned above [54,55].

3. Results

In order to test our methods, we selected five other popular link prediction methods to compare against: $L3$ relies on a weighted counting of paths of length three and was designed specifically to predict links in PPI networks [4]; *preferential attachment* (PA) defines a score between two disconnected nodes by multiplying their degrees [40,41]; *common neighbours* (CN) is a straightforward heuristic that assigns a score to the node pair (u, v) defined by the number of neighbours that u and v have in common; *Adamic-Adar* (AA) is

an adaptation of the common neighbours idea, but adds more weight to less-connected neighbours [1]; the *structural perturbation method* (SPM) uses perturbations of the adjacency matrix of a graph in order to estimate its predictability [56]. While the SPM has shown great success as a general link prediction method [6,57], it is yet to be tested extensively on PPI networks. For the SPM, we used $p^H = 0.1$ and averaged the results over 10 runs, as was performed in the original paper [56].

The following tables show the average AuPR and AuROC values for the six different networks described in Section 2.3. Each value was averaged over 10 runs (10 randomly selected edge removals), and the highest value for each network is shown in bold. We compared three variations of our proposed methods, labelled as “QW-A”, “QW-L”, and “CRW”, referring to quantum walks using the network adjacency matrix as the Hamiltonian, quantum walks using the network Laplacian matrix as the Hamiltonian, and classical random walks, respectively.

For completeness, in Figures 2–7, we also include plots showing the relationship of the area under the precision–recall curve and area under the ROC curve as a function of the edge removal fraction.

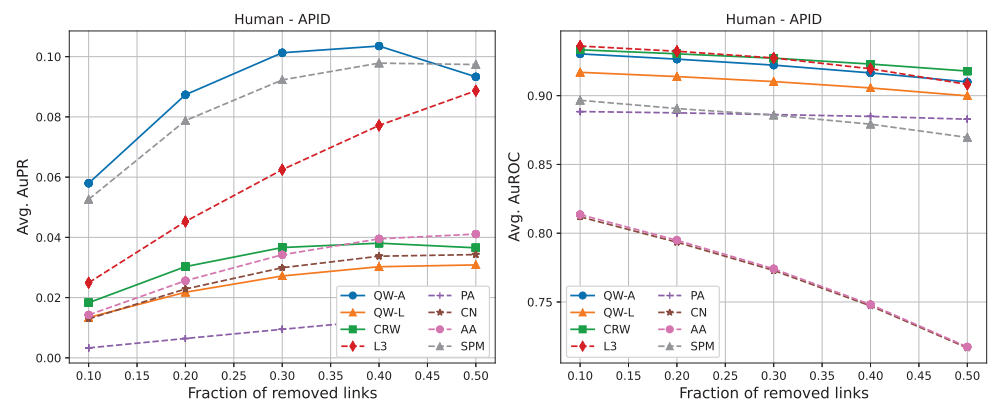


Figure 2. Average areas under the precision–recall curve (**left**) and average areas under the receiver operating characteristic curve (**right**) as a function of the fraction of true links that were removed from the APID *Homo sapiens* PPI network [48,49]. Plotted values are the averages over 10 runs. Our walks used a hyperparameter of $t = 3/\langle k \rangle$.

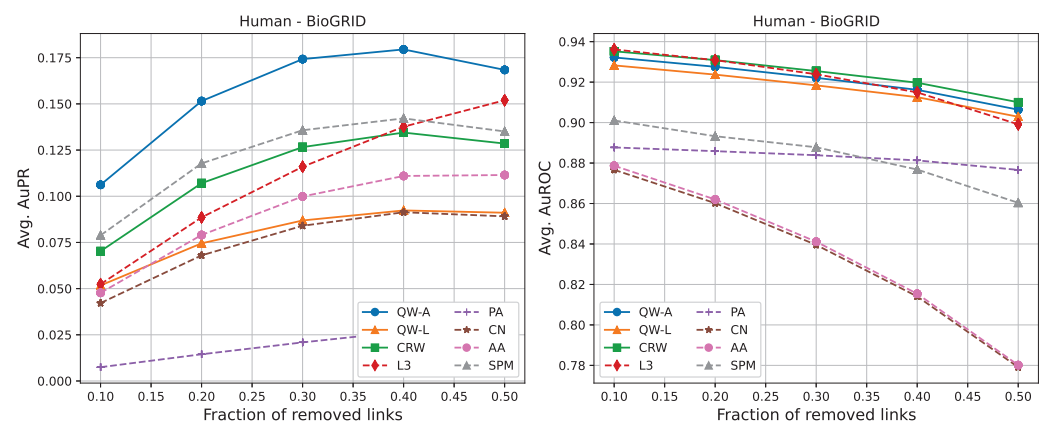


Figure 3. Average areas under the precision–recall curve (**left**) and average areas under the receiver operating characteristic curve (**right**) as a function of the fraction of true links that were removed from the BioGRID *Homo sapiens* PPI network [46]. Plotted values are the averages over 10 runs. Our walks used a hyperparameter of $t = 2/\langle k \rangle$.

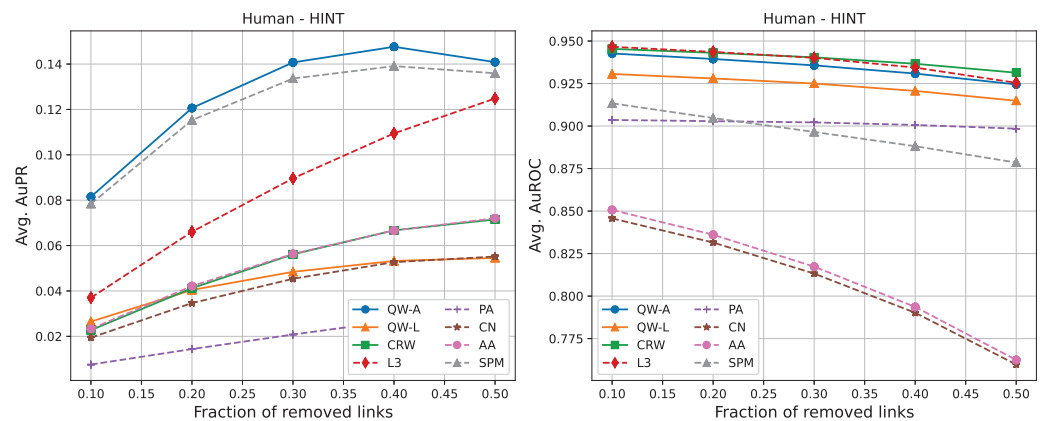


Figure 4. Average areas under the precision–recall curve (**left**) and average areas under the receiver operating characteristic curve (**right**) as a function of the fraction of true links that were removed from the HINT *Homo sapiens* PPI network [47]. Plotted values are the averages over 10 runs. Our walks used a hyperparameter of $t = 3/\langle k \rangle$.

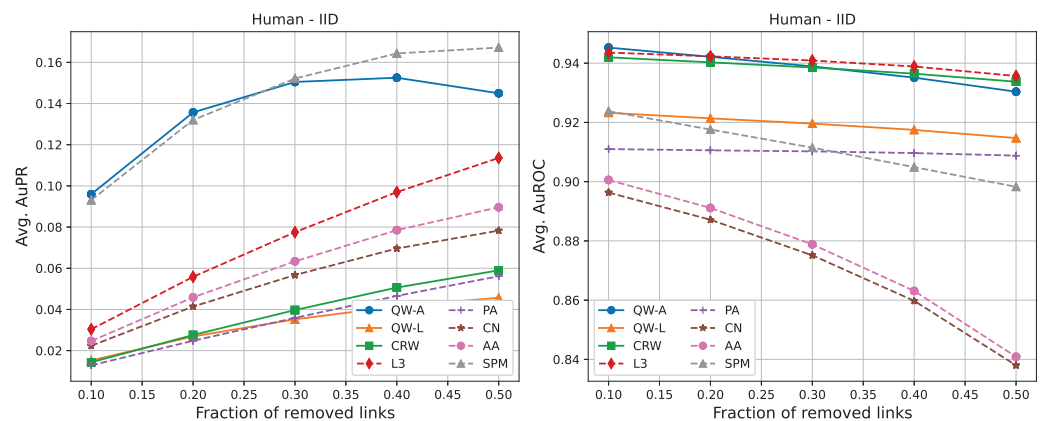


Figure 5. Average areas under the precision–recall curve (**left**) and average areas under the receiver operating characteristic curve (**right**) as a function of the fraction of true links that were removed from the IID *Homo sapiens* PPI network [50]. Plotted values are the averages over 10 runs. Our walks used a hyperparameter of $t = 4/\langle k \rangle$.

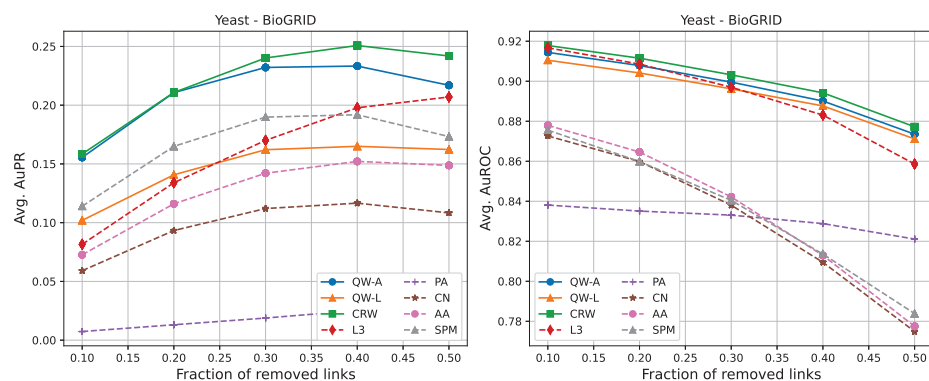


Figure 6. Average areas under the precision–recall curve (**left**) and average areas under the receiver operating characteristic curve (**right**) as a function of the fraction of true links that were removed from the BioGRID *Saccharomyces cerevisiae* PPI network [46]. Plotted values are the averages over 10 runs. Our walks used a hyperparameter of $t = 2/\langle k \rangle$.

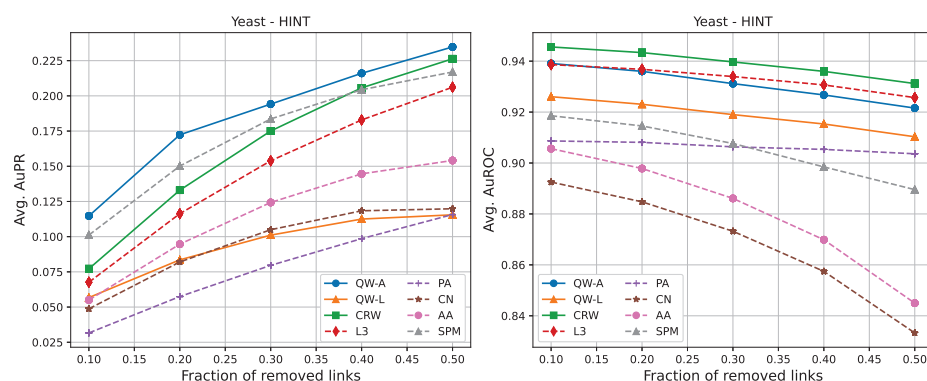


Figure 7. Average areas under the precision–recall curve (**left**) and average area under the receiver operating characteristic curve (**right**) as a function of the fraction of true links that were removed from the HINT *Saccharomyces cerevisiae* PPI network [47]. Plotted values are the averages over 10 runs. Our walks used a hyperparameter of $t = 2/\langle k \rangle$.

In terms of area under the precision–recall curve (AuPR), the quantum walk with the adjacency Hamiltonian (QW-A) showed the best results overall. When 10% of the edges were removed, the QW-A had a higher average AuPR than all other benchmarked methods. This also held when 50% of the edges were removed, except in three cases. For the secondary metric, AuROC, the three best methods appeared to be QW, CRW, and L3; while L3 had the highest AuROC in half of the networks at the 10% removal level by a small margin, CRW had the highest AuROC at the 50% level in all but one network.

4. Discussion

The experimental results in the previous section showed that our methods performed well on a variety of PPI networks. In particular, we saw that our quantum walk with the adjacency Hamiltonian method yielded the best overall performance of all algorithms tested with respect to the area under the precision–recall curve. Furthermore, the adjacency Hamiltonian always beat the Laplacian as the better choice when comparing the results of quantum walks. One possible explanation for this is that the inclusion of node degrees on the diagonal of the Hamiltonian for the Laplacian matrix caused walkers to remain at nodes for longer periods of time, thus preventing them from adequately exploring the rest of the network. In order to explore this further, in Figure 8, we show the distribution of the return probabilities $P_{ii}(t)$ over all nodes i for the various networks studied. Indeed, we see that the QW-L had a large spike close to 1.0 for all of the networks, indicating that the majority of nodes were never departed from when using the Laplacian Hamiltonian. In order to verify that this claim holds for other values of t , in Figure 9, we compare the return probabilities, averaged over all nodes, for various values of t . We see that the QW-L always had the largest average return probability, while the QW-A had an average return probability that was less than the QW-L, but larger than the CRW.

Comparing the QW-A to the CRW, Tables 2 and 3 above show that the former had a higher area under the precision–recall curve for all networks, except the Yeast-BioGRID network. One interesting property of this network is that it has the highest proportion of self-edges (826 self-interacting proteins out of 4186 proteins; see Table 1) of all the networks considered. In order to test the hypothesis that the CRW performs better when the proportion of self-edges is high, we repeated our experiments on the Yeast-BioGRID network, but this time did not use any self-edges for scoring. We found that the change in AuPR was negligible and that the CRW still had a slightly higher AuPR than the QW-A. Therefore, we do not believe that the high proportion of self-edges plays a significant role in explaining the better performance of the CRW for this network.

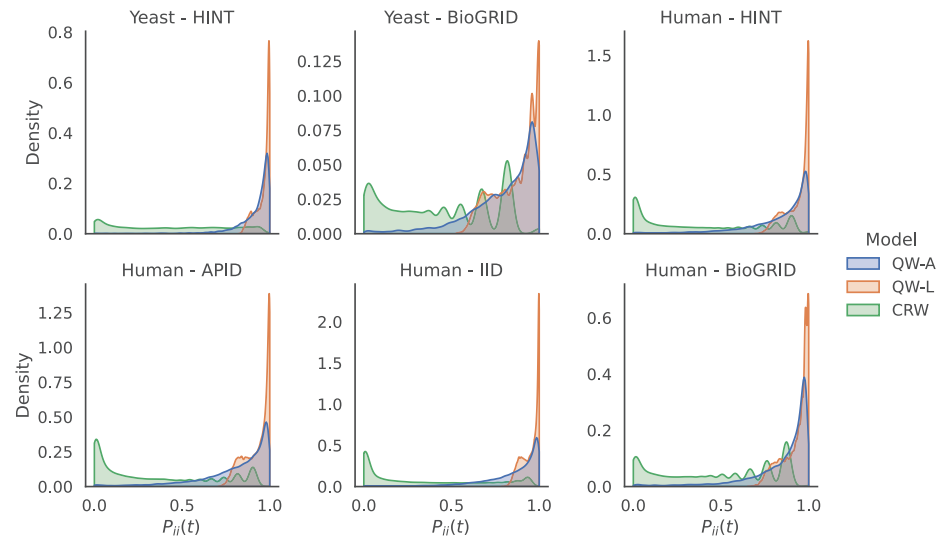


Figure 8. Comparison of return probabilities for the quantum and classical random walk methods on the 6 networks studied. For each network, we show kernel density estimations of the return probabilities $P_{ii}(t)$, for every node i . The values of t used are those for which the AuPRs and AuROCs were presented above.

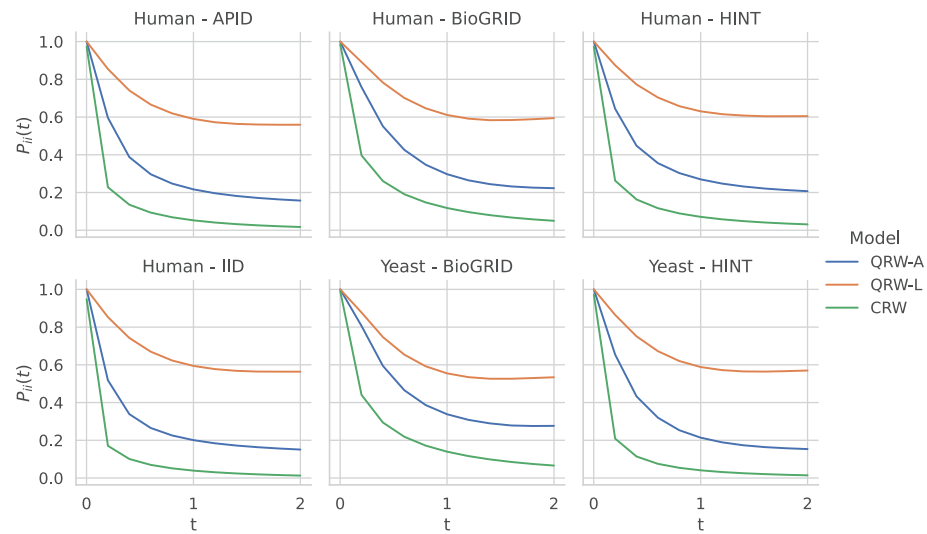


Figure 9. Comparison of return probabilities for the different quantum and classical random walk methods on the 6 networks studied. For each network, we show the the average value of $P_{ii}(t)$, averaged over all nodes, for values of t in the range $(0, 2)$.

Another possible explanation for the higher AuPR of the CRW on the Yeast-BioGRID network may be due to its relatively high clustering compared to the other networks. In order to test this hypothesis, we used a theoretical model to generate scale-free networks with tunable average clusterings [58]. Using this model, we generated scale-free networks with a variety of average clusterings while holding the average degree constant, up to minor random fluctuations. We then used these networks to run the QW-A and CRW using the same cross-validation method described above, with half the edges being removed for testing, in order to compare their performance. In Figure 10, we see that, in all four cases, there was indeed a trend confirming that the QW-A has a better performance when clustering is low, while CRW performs better when clustering is higher. While these theoretically generated networks may not be accurate models of true PPI networks, the effect of clustering on classical and quantum walks remains an interesting topic for future research.

Table 2. Area under the precision–recall curve for 10% edge removals, averaged over 10 runs. The highest AuPR for each network is shown in bold.

AuPR: 10% Removal								
Network	QW-A	QW-L	CRW	L3	PA	CN	AA	SPM
Human-APID	0.058	0.013	0.018	0.025	0.003	0.013	0.014	0.053
Human-BioGRID	0.106	0.052	0.070	0.052	0.007	0.042	0.048	0.079
Human-HINT	0.081	0.026	0.023	0.037	0.008	0.019	0.023	0.078
Human-IID	0.096	0.015	0.014	0.030	0.013	0.022	0.025	0.093
Yeast-BioGRID	0.156	0.102	0.158	0.082	0.007	0.059	0.073	0.114
Yeast-HINT	0.115	0.057	0.077	0.068	0.032	0.049	0.055	0.101

Table 3. Area under the precision–recall curve for 50% edge removals, averaged over 10 runs. The highest AuPR for each network is shown in bold.

AuPR: 50% Removal								
Network	QW-A	QW-L	CRW	L3	PA	CN	AA	SPM
Human-APID	0.093	0.031	0.037	0.089	0.015	0.034	0.041	0.097
Human-BioGRID	0.168	0.091	0.129	0.152	0.032	0.089	0.111	0.135
Human-HINT	0.141	0.055	0.072	0.125	0.033	0.055	0.072	0.136
Human-IID	0.145	0.046	0.059	0.114	0.056	0.078	0.090	0.167
Yeast-BioGRID	0.217	0.162	0.242	0.207	0.030	0.108	0.149	0.173
Yeast-HINT	0.235	0.116	0.226	0.206	0.116	0.120	0.154	0.217

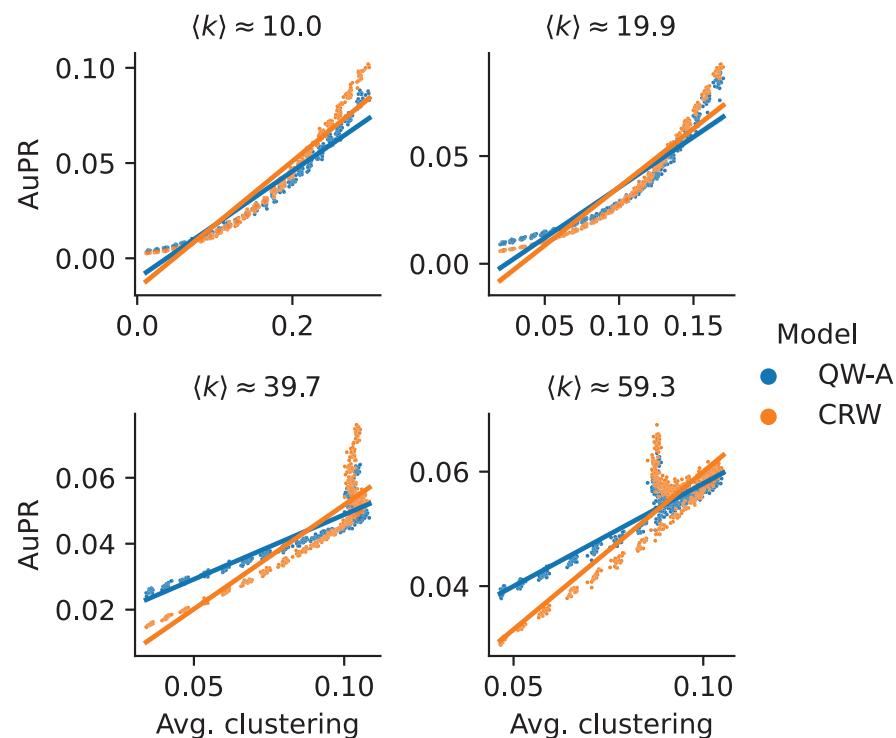


Figure 10. AuPRs for scale-free networks with tunable clusterings. Four different settings are shown, corresponding to different (approximate) average degrees. In each setting, the average clustering was varied to produce different networks. In the resulting networks, half of the edges were reserved for testing, and the remaining network was used to run the QW-A and CRW link prediction methods on. Each point corresponds to the AuPR of a generated network; solid lines show linear fits. Each plot title shows the average degree $\langle k \rangle$, averaged over all networks.

Finally, we mention a few points about the computational complexity of our algorithm and its implementation. The bottleneck of our algorithm, in either the classical or quantum case, is the computation of the matrix exponential appearing in Equations (3) and (5), which is a very well-studied problem with a long history [59]. Our experiments were performed using the “matrix_exp” function in PyTorch [60], which is an implementation of the Taylor polynomial approximation algorithm described in [61]. The problem was thus reduced to a constant number of matrix multiplications, another well-studied problem that can be solved more quickly than the naive $O(n^3)$ method, for example with Strassen’s algorithm or its variations [62]. It is also worth noting that, in this implementation of matrix exponentiation, and many others, the norm of the matrix being exponentiated has an impact on running time, so that using a small t , as tends to be the case in our algorithm, may help in this regard.

In order to compare the running times of the link prediction methods studied here, each method was implemented in python 3.10 and vectorised where possible. The methods were then run on the six networks described in Section 2.3, without removing any edges. The experiments were carried out on a setup consisting of 16 cores and 112 GB of RAM. The results of the running times are shown in Table 4. In general, L3, PA, CN, and AA had the fastest running times, but had low AuPR when compared to the QW-A and SPM (Figures 2–7, left). Of the general link prediction methods, the SPM typically had the highest AuPR, but it is computationally demanding due to the need to calculate the eigenvectors and eigenvalues of the perturbed adjacency matrix many times. The QW-A is indeed the most-promising of the methods considered, since its runtime was several times faster than the SPM, while outperforming the SPM in every case, except two, in which case, the QW-A had the higher AuROC (Figures 2 and 5 and Tables 5 and 6).

Table 4. Average runtimes (in minutes) with standard deviations (over 10 runs) on each of the human PPI networks studied. The choice of hyperparameter t for the quantum and classical walks was the same as was reported in the Results Section.

Model	Human				Yeast	
	APID	BioGRID	IID	HINT	BioGRID	HINT
QW-A	4.15 ± 0.05	1.05 ± 0.01	5.39 ± 0.14	4.52 ± 0.03	0.13 ± 0.00	0.39 ± 0.00
QW-L	4.69 ± 0.03	1.2 ± 0.01	6.03 ± 0.14	5.02 ± 0.03	0.14 ± 0.00	0.44 ± 0.00
CRW	3.23 ± 0.05	0.82 ± 0.02	4.43 ± 0.05	3.52 ± 0.08	0.05 ± 0.00	0.17 ± 0.00
L3	0.54 ± 0.05	0.1 ± 0.01	1.15 ± 0.04	0.55 ± 0.03	0.01 ± 0.00	0.1 ± 0.00
PA	0.23 ± 0.03	0.04 ± 0.01	0.33 ± 0.03	0.18 ± 0.03	0.01 ± 0.00	0.03 ± 0.00
CN	0.23 ± 0.04	0.04 ± 0.01	0.39 ± 0.04	0.21 ± 0.03	0.01 ± 0.00	0.03 ± 0.00
AA	0.27 ± 0.05	0.05 ± 0.01	0.41 ± 0.03	0.24 ± 0.03	0.01 ± 0.00	0.04 ± 0.00
SPM	27.28 ± 1.27	6.38 ± 0.03	29.68 ± 0.50	24.67 ± 0.11	0.84 ± 0.01	2.67 ± 0.01

Table 5. Area under the receiver operating characteristic curve for 10% edge removals, averaged over 10 runs. The highest AuROC for each network is shown in bold.

Network	AuROC: 10% Removal							
	QW-A	QW-L	CRW	L3	PA	CN	AA	SPM
Human-APID	0.930	0.917	0.933	0.936	0.888	0.812	0.814	0.897
Human-BioGRID	0.932	0.928	0.935	0.936	0.888	0.877	0.879	0.901
Human-HINT	0.943	0.931	0.945	0.947	0.904	0.846	0.851	0.913
Human-IID	0.945	0.923	0.942	0.944	0.911	0.896	0.901	0.924
Yeast-BioGRID	0.914	0.911	0.918	0.917	0.838	0.873	0.878	0.876
Yeast-HINT	0.939	0.926	0.946	0.939	0.909	0.893	0.906	0.919

Table 6. Area under the receiver operating characteristic curve for 50% edge removals, averaged over 10 runs. The highest AuROC for each network is shown in bold.

Network	AuROC: 50% Removal							
	QW-A	QW-L	CRW	L3	PA	CN	AA	SPM
Human-APID	0.910	0.900	0.918	0.908	0.883	0.717	0.717	0.870
Human-BioGRID	0.906	0.903	0.910	0.899	0.877	0.779	0.780	0.860
Human-HINT	0.924	0.915	0.931	0.925	0.898	0.760	0.762	0.879
Human-IID	0.930	0.915	0.934	0.936	0.909	0.838	0.841	0.898
Yeast-BioGRID	0.874	0.871	0.877	0.859	0.821	0.775	0.777	0.784
Yeast-HINT	0.922	0.910	0.931	0.926	0.904	0.833	0.845	0.890

5. Conclusions

Although experimental methods have greatly improved in the past ten years, most interactomes remain far from being complete. It is, therefore, important to discover new computational methods for inferring interactions from incomplete datasets. We described a class of algorithms based on continuous-time walks that rank among the best link prediction methods tested on PPI networks.

Furthermore, the continuous-time quantum walks described here are among the first successful quantum-inspired link prediction methods. Although we found that using the reciprocal of the average degree provided a good time length for which to run the walks, many further options can still be explored: using cross-validation to choose a more optimal value or using times that depend on the walker's location are immediate candidates. Another open direction of research involves the choice of the Hamiltonian. Our experimental results demonstrated a strong sensitivity on the Hamiltonian used for controlling the quantum walks. While the adjacency matrix yielded better results than the Laplacian on the networks we tested, it would be beneficial to understand why this is the case. This also indicates the potential for improvement if better Hamiltonians can be found for the purpose of link prediction. Further investigations in this direction may yield better methods and insights into both networks being studied and the quantum walks being employed.

Author Contributions: M.G. and H.S. conceived of the algorithm. G.G.-P., M.A.C.R. and S.M. designed and directed the research. M.G. and J.M. implemented the algorithms and ran the simulations. M.G., J.M. and H.S. wrote the first version of the manuscript. All authors contributed to the scientific discussions and to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: M.G., H.S., S.M., and G.G.-P. acknowledge support from the Emmy.network foundation. S.M. and M.A.C.R. acknowledge financial support from the Academy of Finland via the Centre of Excellence program (Project No. 336810 and Project No. 336814). G.G.-P. acknowledges financial support from the Academy of Finland via the Postdoctoral Researcher program (Project No. 341985).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The datasets used in this study are available upon request from the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Adamic, L.A.; Adar, E. Friends and neighbors on the web. *Soc. Netw.* **2003**, *25*, 211–230. [CrossRef]
- Murata, T.; Moriyasu, S. Link prediction of social networks based on weighted proximity measures. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'07), Fremont, CA, USA, 2–5 November 2007; pp. 85–88.
- Leskovec, J.; Huttenlocher, D.; Kleinberg, J. Predicting positive and negative links in online social networks. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, CA, USA, 26–30 April 2010; pp. 641–650.
- Kovács, I.A.; Luck, K.; Spirohn, K.; Wang, Y.; Pollis, C.; Schlabach, S.; Bian, W.; Kim, D.K.; Kishore, N.; Hao, T.; et al. Network-based prediction of protein interactions. *Nat. Commun.* **2019**, *10*, 1–8.
- Liu, W.; Lü, L. Link prediction based on local random walk. *EPL Europhys. Lett.* **2010**, *89*, 58007. [CrossRef]

6. Kumar, A.; Singh, S.S.; Singh, K.; Biswas, B. Link prediction techniques, applications, and performance: A survey. *Phys. Stat. Mech. Its Appl.* **2020**, *553*, 124289.
7. Martínez, V.; Berzal, F.; Talavera, J.C.C. A Survey of Link Prediction in Complex Networks. *ACM Comput. Surv.* **2017**, *49*, 69:1–69:33. [CrossRef]
8. Zhou, T. Progresses and challenges in link prediction. *iScience* **2021**, *24*, 103217.
9. Che, Y.; Cheng, W.; Wang, Y.; Chen, D. A Random Walk with Restart Model Based on Common Neighbors for Predicting the Clinical Drug Combinations on Coronary Heart Disease. *J. Healthc. Eng.* **2021**, *2021*, 4597391. [CrossRef]
10. Zhou, Y.; Wu, C.; Tan, L. Biased random walk with restart for link prediction with graph embedding method. *Phys. A Stat. Mech. Its Appl.* **2021**, *570*, 125783. [CrossRef]
11. Brin, S.; Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. Isdn Syst.* **1998**, *30*, 107–117.
12. Das Sarma, A.; Molla, A.R.; Pandurangan, G.; Upfal, E. Fast distributed pagerank computation. In Proceedings of the International Conference on Distributed Computing and Networking, Mumbai, India, 3–6 January 2013; pp. 11–26.
13. Fous, F.; Pirotte, A.; Renders, J.M.; Saerens, M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 355–369.
14. Pan, J.Y.; Yang, H.J.; Faloutsos, C.; Duygulu, P. Automatic multimedia cross-modal correlation discovery. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 653–658.
15. Tong, H.; Faloutsos, C.; Pan, J.Y. Fast random walk with restart and its applications. In Proceedings of the Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006; pp. 613–622.
16. Farhi, E.; Gutmann, S. Quantum computation and decision trees. *Phys. Rev. A* **1998**, *58*, 915–928. [CrossRef]
17. Aharonov, Y.; Davidovich, L.; Zagury, N. Quantum random walks. *Phys. Rev. A* **1993**, *48*, 1687–1690. [CrossRef] [PubMed]
18. Kempe, J. Quantum random walks: An introductory overview. *Contemp. Phys.* **2003**, *44*, 307–327. [CrossRef]
19. Venegas-Andraca, S.E. Quantum walks: A comprehensive review. *Quantum Inf. Process.* **2012**, *11*, 1015–1106. [CrossRef]
20. Childs, A.M. Universal computation by quantum walk. *Phys. Rev. Lett.* **2009**, *102*, 180501. [CrossRef]
21. Mülken, O.; Blumen, A. Continuous-time quantum walks: Models for coherent transport on complex networks. *Phys. Rep.* **2011**, *502*, 37–87. [CrossRef]
22. Qian, J.; Yang, L.; Yu, Z.; Liu, S. Link prediction using discrete-time quantum walk. *Teh. Vjesn.* **2017**, *24*, 1329–1334. [CrossRef]
23. Moutinho, J.A.P.; Melo, A.; Coutinho, B.; Kovács, I.A.; Omar, Y. Quantum link prediction in complex networks. *Phys. Rev. A* **2023**, *107*, 032605. [CrossRef]
24. Manouchehri, K.; Wang, J. *Physical Implementation of Quantum Walks*; Springer: Berlin/Heidelberg, Germany, 2014.
25. Young, A.W.; Eckner, W.J.; Schine, N.; Childs, A.M.; Kaufman, A.M. Tweezer-programmable 2D quantum walks in a Hubbard-regime lattice. *Science* **2022**, *377*, 885–889. [CrossRef]
26. Wang, K.; Shi, Y.; Xiao, L.; Wang, J.; Joglekar, Y.N.; Xue, P. Experimental realization of continuous-time quantum walks on directed graphs and their application in PageRank. *Optica* **2020**, *7*, 1524–1530. [CrossRef]
27. Tang, H.; Lin, X.F.; Feng, Z.; Chen, J.Y.; Gao, J.; Sun, K.; Wang, C.Y.; Lai, P.C.; Xu, X.Y.; Wang, Y.; et al. Experimental two-dimensional quantum walk on a photonic chip. *Sci. Adv.* **2018**, *4*, eaat3174. [CrossRef] [PubMed]
28. Peruzzo, A.; Lobino, M.; Matthews, J.C.F.; Matsuda, N.; Politi, A.; Poullos, K.; Zhou, X.Q.; Lahini, Y.; Ismail, N.; Wörhoff, K.; et al. Quantum Walks of Correlated Photons. *Science* **2010**, *329*, 1500–1503. [CrossRef] [PubMed]
29. Preiss, P.; Ma, R.; Tai, E.; Lukin, A.; Rispoli, M.; Zupancic, P.; Lahini, Y.; Islam, R.; Greiner, M. Strongly correlated quantum walks in optical lattices. *Science* **2015**, *347*, 1229–1233. [CrossRef] [PubMed]
30. Gong, M.; Wang, S.; Zha, C.; Chen, M.C.; Huang, H.L.; Wu, Y.; Zhu, Q.; Zhao, Y.; Li, S.; Guo, S.; et al. Quantum walks on a programmable two-dimensional 62-qubit superconducting processor. *Science* **2021**, *372*, 948–952. [CrossRef] [PubMed]
31. Yan, Z.; Zhang, Y.R.; Gong, M.; Wu, Y.; Zheng, Y.; Li, S.; Wang, C.; Liang, F.; Lin, J.; Xu, Y.; et al. Strongly correlated quantum walks with a 12-qubit superconducting processor. *Science* **2019**, *364*, 753–756. [CrossRef] [PubMed]
32. Loke, T.; Wang, J.B. Efficient quantum circuits for continuous-time quantum walks on composite graphs. *J. Phys. Math. Theor.* **2017**, *50*, 055303. [CrossRef]
33. Qiang, X.; Loke, T.; Montanaro, A.; Aungskunsiri, K.; Zhou, X.; O'Brien, J.L.; Wang, J.B.; Matthews, J.C.F. Efficient quantum walk on a quantum processor. *Nat. Commun.* **2016**, *7*, 11511. [CrossRef]
34. Vidal, M.; Cusick, M.E.; Barabási, A.L. Interactome networks and human disease. *Cell* **2011**, *144*, 986–998. [CrossRef]
35. Stelzl, U.; Worm, U.; Lalowski, M.; Haenig, C.; Brembeck, F.H.; Goehler, H.; Stroedicke, M.; Zenkner, M.; Schoenherr, A.; Koeppen, S.; et al. A human protein–protein interaction network: A resource for annotating the proteome. *Cell* **2005**, *122*, 957–968. [CrossRef]
36. Rolland, T.; Taşan, M.; Charlotiaux, B.; Pevzner, S.J.; Zhong, Q.; Sahni, N.; Yi, S.; Lemmens, I.; Fontanillo, C.; Mosca, R.; et al. A proteome-scale map of the human interactome network. *Cell* **2014**, *159*, 1212–1226. [CrossRef]
37. Luck, K.; Kim, D.K.; Lambourne, L.; Spirohn, K.; Begg, B.E.; Bian, W.; Brignall, R.; Cafarelli, T.; Campos-Laborie, F.J.; Charlotiaux, B.; et al. A reference map of the human binary protein interactome. *Nature* **2020**, *580*, 402–408. [CrossRef]
38. Yuen, H.Y.; Jansson, J. Better Link Prediction for Protein-Protein Interaction Networks. In Proceedings of the 2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE), Cincinnati, OH, USA, 26–28 October 2020; pp. 53–60. [CrossRef]

39. Yuen, H.Y.; Jansson, J. Normalized L3-based link prediction in protein protein interaction networks. *BMC Bioinform.* **2023**, *24*, 59. [CrossRef] [PubMed]
40. Liben-Nowell, D.; Kleinberg, J. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* **2007**, *58*, 1019–1031. [CrossRef]
41. Barabási, A.L.; Jeong, H.; Néda, Z.; Ravasz, E.; Schubert, A.; Vicsek, T. Evolution of the social network of scientific collaborations. *Phys. A Stat. Mech. Its Appl.* **2002**, *311*, 590–614. [CrossRef]
42. Masuda, N.; Porter, M.A.; Lambiotte, R. Random walks and diffusion on networks. *Phys. Rep.* **2017**, *716*, 1–58. [CrossRef]
43. Childs, A.M.; Farhi, E.; Gutmann, S. An Example of the Difference Between Quantum and Classical Random Walks. *Quantum Inf. Process.* **2002**, *1*, 35–43. [CrossRef]
44. Thomas G. Wong, L.T.; Nahimov, N. Laplacian versus adjacency matrix in quantum walk search. *Quantum Inf. Process.* **2016**, *15*, 4029–4048. [CrossRef]
45. Childs, A.M.; Goldstone, J. Spatial search by quantum walk. *Phys. Rev. A* **2004**, *70*, 022314. [CrossRef]
46. Stark, C.; Breitkreutz, B.J.; Reguly, T.; Boucher, L.; Breitkreutz, A.; Tyers, M. BioGRID: A general repository for interaction datasets. *Nucleic Acids Res.* **2006**, *34*, D535–D539. [CrossRef]
47. Das, J.; Yu, H. HINT: High-quality protein interactomes and their applications in understanding human disease. *BMC Syst. Biol.* **2012**, *6*, 1–12. [CrossRef]
48. Alonso-López, D.; Campos-Laborie, F.J.; Gutiérrez, M.A.; Lambourne, L.; Calderwood, M.A.; Vidal, M.; De Las Rivas, J. APID database: Redefining protein–protein interaction experimental evidences and binary interactomes. *Database* **2019**, *2019*, baz005. [CrossRef] [PubMed]
49. Alonso-Lopez, D.; Gutiérrez, M.A.; Lopes, K.P.; Prieto, C.; Santamaría, R.; De Las Rivas, J. APID interactomes: Providing proteome-based interactomes with controlled quality for multiple species and derived networks. *Nucleic Acids Res.* **2016**, *44*, W529–W535. [CrossRef]
50. Kotlyar, M.; Pastrello, C.; Sheahan, N.; Jurisica, I. Integrated interactions database: Tissue-specific view of the human and model organism interactomes. *Nucleic Acids Res.* **2016**, *44*, D536–D541. [CrossRef]
51. Barabási, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [CrossRef]
52. Hanley, J.A.; McNeil, B.J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **1982**, *143*, 29–36. [CrossRef]
53. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [CrossRef]
54. Armengol, E.; Boixader, D.; Grimaldo, F. Evaluating link prediction on large graphs. In *Artificial Intelligence Research and Development: Proceedings of the 18th International Conference of the Catalan Association for Artificial Intelligence*; IOS Press: Amsterdam, The Netherlands, 2015; Volume 277.
55. Saito, T.; Rehmsmeier, M. The precision–recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE* **2015**, *10*, e0118432. [CrossRef]
56. Lü, L.; Pan, L.; Zhou, T.; Zhang, Y.C.; Stanley, H. Toward link predictability of complex networks. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, 201424644. [CrossRef]
57. Zeng, X.; Liu, L.; Lü, L.; Zou, Q. Prediction of potential disease-associated microRNAs using structural perturbation method. *Bioinformatics* **2018**, *34*, 2425–2432. [CrossRef]
58. Holme, P.; Kim, B.J. Growing scale-free networks with tunable clustering. *Phys. Rev. E* **2002**, *65*, 026107. [CrossRef]
59. Moler, C.; Van Loan, C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.* **2003**, *45*, 3–49. [CrossRef]
60. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
61. Bader, P.; Blanes, S.; Casas, F. Computing the matrix exponential with an optimized Taylor polynomial approximation. *Mathematics* **2019**, *7*, 1174. [CrossRef]
62. Strassen, V. Gaussian Elimination is not Optimal. *Numer. Math.* **1969**, *13*, 354–356. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Generation of Pseudo-Random Quantum States on Actual Quantum Processors

Gabriele Cenedese ^{1,2}, Maria Bondani ^{3,*}, Dario Rosa ^{4,5} and Giuliano Benenti ^{1,2,6}

¹ Center for Nonlinear and Complex Systems, Dipartimento di Scienza e Alta Tecnologia, Università degli Studi dell'Insubria, Via Valleggio 11, 22100 Como, Italy

² Istituto Nazionale di Fisica Nucleare, Sezione di Milano, Via Celoria 16, 20133 Milano, Italy

³ Istituto di Fotonica e Nanotecnologie, Consiglio Nazionale delle Ricerche, Via Valleggio 11, 22100 Como, Italy

⁴ Center for Theoretical Physics of Complex Systems, Institute for Basic Science (IBS), Daejeon 34126, Republic of Korea

⁵ Basic Science Program, Korea University of Science and Technology (UST), Daejeon 34113, Republic of Korea

⁶ NEST-CNR Istituto Nanoscienze, 56126 Pisa, Italy

* Correspondence: maria.bondani@uninsubria.it

Abstract: The generation of a large amount of entanglement is a necessary condition for a quantum computer to achieve quantum advantage. In this paper, we propose a method to efficiently generate pseudo-random quantum states, for which the degree of multipartite entanglement is nearly maximal. We argue that the method is optimal, and use it to benchmark actual superconducting (IBM's *ibm_lagos*) and ion trap (IonQ's *Harmony*) quantum processors. Despite the fact that *ibm_lagos* has lower single-qubit and two-qubit error rates, the overall performance of *Harmony* is better thanks to its low error rate in state preparation and measurement and to the all-to-all connectivity of qubits. Our result highlights the relevance of the qubits network architecture to generate highly entangled states.

Keywords: quantum computing; NISQ devices; random quantum circuits

1. Introduction

Quantum computers working with approximately 50–100 qubits could perform certain tasks beyond the capabilities of current classical supercomputers [1,2], and quantum advantage for particular problems has recently been claimed [3–5], although later simulations on classical supercomputers [6,7] have almost closed the quantum advantage gap. As a general remark, quantum advantage can only be achieved if the precision of the quantum gates is sufficiently high and the executed quantum algorithm generates a sufficiently large amount of entanglement that can overcome classical tensor network methods [8]. Therefore, for quantum algorithms, multipartite (many-qubit) entanglement is the key resource to achieve exponential acceleration over classical computation. Unfortunately, existing noisy intermediate-scale quantum (NISQ) devices suffer from various noise sources such as noisy gates, coherent errors, and interactions with an uncontrolled environment. Noise limits the size of quantum circuits that can be reliably executed, so achieving quantum advantage in complex and practically relevant problems is still a formidable challenge. It is therefore important to benchmark the progress of currently available quantum computers [9–11].

Quantifying entanglement is a demanding task [12,13]. In particular, the characterization of multipartite entanglement is not a simple matter, since, as the number of subsystems increases, we should consider all possible non-local correlations among parties in order to obtain a complete description of entanglement. Moreover, tomographic state reconstruction requires a number of measures that grows exponentially with the number of qubits [14]. Finally, there is no unique way to characterize multipartite entanglement [13].

On the other hand, bipartite entanglement can be probed by means of entanglement entropies. In particular, we can consider the *second order Rényi entropy* of the reduced density

Citation: Cenedese, G.; Bondani, M.; Rosa, D.; Benenti, G. Generation of Pseudo-Random Quantum States on Actual Quantum Processors. *Entropy* **2023**, *25*, 607. <https://doi.org/10.3390/e25040607>

Academic Editor: Masahito Hayashi

Received: 16 February 2023

Revised: 20 March 2023

Accepted: 29 March 2023

Published: 3 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

matrix for any of the subsystems. If it is larger than the entropy of the entire system, we can conclude that bipartite entanglement exists between the two subsystems. If the overall state is pure, the second-order Rényi entropy is directly a measure of bipartite entanglement. In that case, in order to quantify the amount of multipartite entanglement, one can look at the distribution of the Rényi entropy of a subsystem over all possible bipartitions of the total system. For example, Facchi et al. proposed [15] a method based on the probability density of bipartite entanglement between two parts of the total system; one expects that multipartite entanglement will be large when bipartite entanglement is large and does not depend on the bipartition, namely when its probability density is a narrow function centered at a large value.

Computing entanglement entropies requires the knowledge of the density matrix of the system. Unfortunately, probing the density matrix is also a challenging problem, especially as the dimension of the system increases. For this reason, it is necessary to indirectly estimate the entropy, for instance, using the method proposed by Brydges et al. [16] via randomized measurements.

For random pure quantum states, the entanglement content is almost maximal and the purity (and so the second-order Rényi entropy) probability distribution is well known. Unlike simpler states such as W and GHZ, for which the entanglement content is essentially independent of the dimension of the system, for random states, the average multipartite entanglement is an extensive quantity. Moreover, random states are relevant in the study of the complexity of quantum circuits [17] and black holes [18] and for benchmarking quantum hardware [9,19].

The purpose of this paper was to investigate strategies to efficiently generate highly entangled states and then find a way to quantify the actual amount of entanglement achieved in state-of-the-art quantum hardware. In particular, we propose a method (hereafter referred to as *direct method*) to efficiently generate pseudo-random quantum states for n qubits, approximating true random states to the desired accuracy by means of layers where a random permutation of the qubits is followed by two-qubit random state generation. We provide numerical evidence that this method converges to true n -qubit random states by increasing the number of layers as fast as the circuit implementing two-qubit random unitary gates using the KAK parametrization of $SU(4)$ (*KAK method*) [20], but with reduced cost in terms of the number of CNOT gates. We also argue that the proposed method is optimal for pseudo-random quantum state generation. Finally, we implement the method to benchmark actual quantum processors. In particular, two different realizations of quantum hardware are compared: IBM's superconducting-based devices and IonQ's trapped ion-based devices. We show that, despite the fact that superconducting devices have smaller error rates than IonQ for one- and two-qubit gates, the overall performance is better in trapped ion devices. This is mainly due to the complete connectivity of these machines, which allows avoiding noisy SWAP gates to implement qubit permutations. Our results highlight the importance of quantum hardware architecture in the implementation of quantum algorithms.

This paper is organized as follows. In Section 2, we discuss and compare methods for the generation of pseudo-random states. In Section 3, we apply the direct method in real quantum hardware and compare the results for IBMQ and IonQ devices with the second-order Rényi entropy estimated via the method of Ref. [16]. Finally, our conclusions are drawn in Section 4.

2. Generation of Pseudo-Random Quantum States

In this section, we briefly discuss methods of generating pseudo-random states, starting with the exact strategy and ending with our proposal, which will be numerically verified by comparison with the standard KAK method.

Let $|\psi\rangle$ be a pure state that belongs to the Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, where \mathcal{H}_A and \mathcal{H}_B are spanned, respectively, by $\{|i_A\rangle\}_{1 \leq i_A \leq N_A}$ and $\{|i_B\rangle\}_{1 \leq i_B \leq N_B}$. A and B are two

bipartitions of the entire system. Assuming that, without loss of generality, $N_A \leq N_B$, the state admits a Schmidt decomposition [1]:

$$|\psi\rangle = \sum_{i=1}^{N_A} \sqrt{x_i} |a_i\rangle \otimes |b_i\rangle, \tag{1}$$

where $\{|a_i\rangle\}$ and $\{|b_i\rangle\}$ are suitable orthonormal sets for \mathcal{H}_A and \mathcal{H}_B , respectively, which depend on the particular state $|\psi\rangle$, and the scalars x_i , known as the Schmidt coefficients for $|\psi\rangle$, are real, non-negative, and unique up to reordering. These coefficients can be used to quantify the bipartite entanglement via the second-order Rényi entropy

$$S^{(2)}(\rho_A) = -\log_2[R(\psi)], \tag{2}$$

with the reduced purity $R(\psi)$ of the state given by

$$R(\psi) = \text{Tr}(\rho_A^2) = \sum_{i=1}^{N_A} x_i^2, \tag{3}$$

where ρ_A is the reduced density matrix (with respect to \mathcal{H}_B) of the overall state ρ :

$$\rho_A = \text{Tr}_B(\rho) = \text{Tr}_B(|\psi\rangle\langle\psi|). \tag{4}$$

Hereafter, we shall focus on the purity, which is trivially related to the second-order Rényi entropy.

In the case of a random state, the cumulants of the purities' probability distributions can be exactly calculated [21,22], more details on which can be found in Appendix A. In particular, the first cumulants are given by

$$\mu_{N_A N_B} \equiv \langle R \rangle = \frac{N_A + N_B}{1 + N_A N_B}, \tag{5}$$

$$\sigma_{N_A N_B}^2 \equiv \langle (R - \langle R \rangle)^2 \rangle = \frac{2(N_A^2 - 1)(N_B^2 - 1)}{(1 + N_A N_B)^2(2 + N_A N_B)(3 + N_A N_B)}, \tag{6}$$

and they will be used later to verify the quality of random state generation.

In order to generate a true n -qubit random state, the ideal (and only) rigorous way would be to apply a random unitary operator, with respect to the Haar measure of the unitary group $SU(N = 2^n)$ (neglecting the global phase of no physical significance). Unfortunately, the implementation of such an operator acting on the n -qubit Hilbert space requires a number of elementary quantum gates that is exponential in the number of qubits [1].

On the other hand, it has been proven that the sequences of random single qubit gates followed by a two-qubit local interaction (which can be an $SU(4)$ random unitary operator, or more simply a single CNOT gate) generate pseudo-random unitary operators which approximate, to the desired accuracy, the entanglement properties of true n -qubit random states [23–27]. However, the random $SU(4)$ strategy depicted in Figure 1, used for example in [9], performs better than a single CNOT in terms of convergence rate [28], with the cost of using three CNOTs instead of just one, as we will see below.

Hence, the problem now turns to find an efficient way (in sense that will be clarified later) to generate random $SU(4)$ operators.

To this end, one possible strategy would be to use Hurwitz's parametrization of the unitary group, $SU(N)$, for the specific case of $N = 4$ [25,29]. However, this approach has the disadvantage of requiring a large number of CNOTs—16 for the particular case of $SU(4)$ —which are usually the main source of errors in NISQ devices [30].

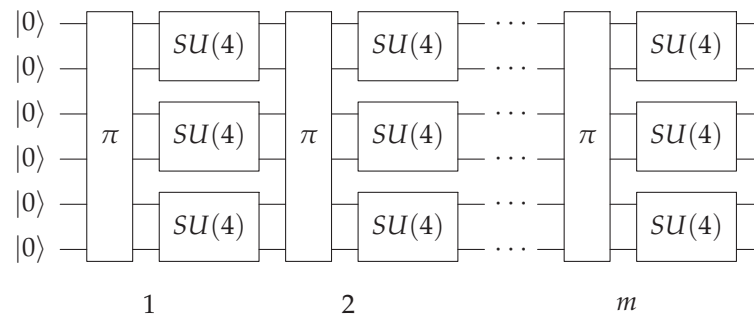


Figure 1. The pseudo-random state generator circuit consists of m layers of random permutations of the qubit labels, followed by random two-qubit gates. When the circuit width n is odd, one of the qubits is idle in each layer. In this figure, a circuit with $n = 6$ qubits width is shown for illustration purposes.

2.1. Cartan’s KAK Decomposition of the Unitary Group

An alternative approach consists of using the Cartan’s KAK decomposition of a semi-simple Lie group \mathbf{G} (in this case $SU(2^n)$) which parametrizes the group in terms of the subgroups’ elements [31]. The case of $SU(4)$ of interest here is described in Appendix B, and is the optimal construction [20] to implement a generic two-qubit gate, using at most 3 CNOT and 15 single-qubit gates.

2.2. Direct Generation of Two-Qubit Random Quantum States

Is the Cartan decomposition the most efficient way to generate a two-qubit random state? Let us think in terms of free parameters. The Cartan’s KAK decomposition is the optimal (in terms of the number of CNOT and single-qubit rotations) way to construct random $SU(4)$ operators via quantum circuits. It requires 15 single-qubit rotations and so 15 independent real parameters (as one expects, since $\dim(SU(4)) = 15$). On the other hand, a normalized random two-qubit state $|\psi\rangle$ depends, up to a global phase, on six independent real parameters. This suggests that, in some ways, it could be possible to build any two-qubit state (starting from some fiducial state) with at most six independent rotations and one CNOT (needed to entangle the system).

This expectation is confirmed [21] by the quantum circuit depicted in Figure 2, producing $|\psi\rangle$ from an initial state $|00\rangle$. How can this circuit be achieved? Starting from a state $|\psi\rangle$, and transforming it by the inverse of the circuit of Figure 2, one can end up with $|00\rangle$, specifying how the angles θ are obtained. Any two-qubit state can in fact be written, using the Schmidt decomposition as a sum of two product terms:

$$|\psi\rangle = \sqrt{x_1}|a\rangle|b\rangle + \sqrt{x_2}|a^\perp\rangle|b^\perp\rangle, \tag{7}$$

where $|a\rangle$ and $|b\rangle$ are single-qubit states (of the first and second qubit, respectively), and $|a^\perp\rangle$ and $|b^\perp\rangle$ are single-qubit states orthogonal to $|a\rangle$ and $|b\rangle$, respectively (i.e., $\{|a\rangle, |a^\perp\rangle\}$ and $\{|b\rangle, |b^\perp\rangle\}$ are the Schmidt bases of the Hilbert spaces of the two qubits). The idea is, starting from this decomposition, to obtain the state $|00\rangle$ using unitary operations, and then, taking the inverse transformation, one can obtain the desired result. The angle θ_4 is chosen such that the R_z rotation of angle $-\theta_4$ eliminates a relative phase between the coefficients of the expansion of $|a\rangle$ into $|0\rangle$ and $|1\rangle$ (note that, because the circuit is considered in the reverse direction, the angles of rotations have opposite signs). A subsequent R_y rotation with angle $-\theta_3$ results in the transformation $|a\rangle \rightarrow |0\rangle$ (up to a global phase). Similarly, rotations of angles $-\theta_6$ and $-\theta_5$ rotate $|b\rangle$ into $|0\rangle$. After applying rotations of angles $-\theta_3, -\theta_4, -\theta_5$, and $-\theta_6$, the state has become, up to a global phase, of the form:

$$|\psi\rangle = \cos \theta_1|00\rangle + e^{i\theta_2} \sin \theta_1|11\rangle. \tag{8}$$

Finally, the R_z rotation of $-\theta_2$ eliminates the relative phase between $|00\rangle$ and $|11\rangle$. The CNOT brings the second qubit to $|0\rangle$, and the last rotation of angle θ_1 on the first qubit yields the final state $|00\rangle$.

In order to obtain a random state, it is necessary to know how to randomly sample the various angles θ_i , that is, it is necessary to know their probability distributions with respect to some measure of the state space, associated with the parametrization provided by Figure 2. Formally, a quantum state $|\psi\rangle$ can be considered as an element of the complex projective space $\mathbb{C}P^{N-1}$, with $N = 2^n$ being the Hilbert space dimension for n qubits [32]. The natural Riemannian metric on $\mathbb{C}P^{N-1}$ is the Fubini–Study metric induced by the unitarily invariant Haar measure on $U(N)$. This is the only metric invariant under unitary transformations. Thus, the unnormalized joint probability distribution is simply obtained by calculating the determinant of the metric tensor with the parametrization [33]. The idea is to use these more efficient “operators” D to construct the n -qubit pseudo-random states, although formally they do not map the entire Bloch sphere if the initial state is not $|00\rangle$. Consider, for example, a dimensionally simpler case: from the north pole of a sphere, it is possible to reach any other point by making only two rotations. Thus, carefully choosing the distribution of the rotation angles, it is possible to uniformly map every point of the sphere, but this is no longer valid if the starting point is changed, where the worst case scenario is a point on the equator.

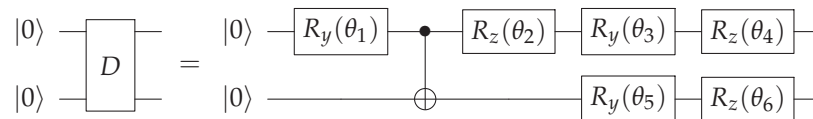


Figure 2. A circuit for two-qubit random state generation. Rotations R_k are obtained by exponentiating the corresponding Pauli matrices σ_k .

In general, one expects that the error committed in sampling the Bloch space is small and everything converges to a random state anyway (see below). In Figure 3, the circuit used to generate the random state (in a similar way to Figure 1) with this method is shown, which will be referred to from now on as the direct method.

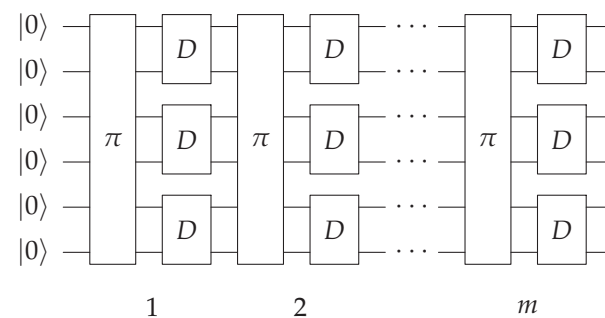


Figure 3. The pseudo-random state generator circuit consists of m layers of random permutations of the qubit labels, followed by random D gates. In this figure, a circuit with $n = 6$ qubit’s width is shown for illustrative purposes.

2.3. Comparison of KAK and Direct Method

In general, for an n -qubit state, there are $\binom{n}{n_a}$ ways to construct a bipartition in n_a and $n_b = n - n_a$ qubits ($N_A = 2^{n_a}$, $N_B = 2^{n_b}$). Clearly, n_a can be any natural number from 1 to n . For the sake of clarity, let us consider, for example, a four-qubit state, wherein the qubits are labeled $\{0, 1, 2, 3\}$. The $n_a = 2$ bipartition can be obtained in $\binom{4}{2} = 6$ different ways by tracing out the pair of qubits $\{0, 1\}$, $\{0, 2\}$, $\{0, 3\}$, $\{1, 2\}$, $\{1, 3\}$ or $\{2, 3\}$. In the case of a random state, these partitions are equivalent, i.e., the value of purity is independent of the choice of the subset of qubits traced out.

Given a quantum state generated as shown in Figure 1 (KAK method) and Figure 3 (direct method), and taking an ensemble of N_e states, we numerically estimate the mean value

$(\mu_{2^{n_a} 2^{n-n_a}})_e$ and the variance $(\sigma_{2^{n_a} 2^{n-n_a}}^2)_e$ of the purities of the generated pseudo-random quantum state. Simulations are performed using the Python library *Qiskit*, particularly the system density matrix, which is computed using the built-in state-vector simulator. In order to evaluate how well the states are generated, the idea is to calculate the relative error of the mean value Δ_μ and the variance Δ_{σ^2} , which are averaged over each possible bipartition of the number of qubits:

$$\overline{\Delta_\mu} = \frac{1}{n-1} \sum_{n_a=1}^{n-1} \frac{|(\mu_{2^{n_a} 2^{n-n_a}})_e - \mu_{2^{n_a} 2^{n-n_a}}|}{\mu_{2^{n_a} 2^{n-n_a}}}, \tag{9}$$

$$\overline{\Delta_{\sigma^2}} = \frac{1}{n-1} \sum_{n_a=1}^{n-1} \frac{|(\sigma_{2^{n_a} 2^{n-n_a}}^2)_e - \sigma_{2^{n_a} 2^{n-n_a}}^2|}{\sigma_{2^{n_a} 2^{n-n_a}}^2}. \tag{10}$$

The sum on n_a is up to $n - 1$, since $n_a = n$ means tracing out the whole system, i.e., calculating the purity of the whole state, which, being pure, has a unit mean and zero variance. In the previous formulas, the quantities $\mu_{2^{n_a} 2^{n-n_a}}$ and $\sigma_{2^{n_a} 2^{n-n_a}}^2$ are the expected values for a true random state, as shown in Equations (5) and (6), respectively.

The averaged relative error for the mean and variance is plotted as a function of the number of steps (i.e., of layers) of the generating circuit and the size of the statistical ensemble, for the cases of $n = 4, 6$, and 8 qubits. Numerical data from Figure 4 suggest that the direct and the KAK methods are basically equivalent in terms of speed of convergence to the expectation values of random states. Notice that the number of steps required for convergence grows as $\sim n$, since at least $n(n - 1)/2$ two-qubit gates are required in order to entangle all qubit pairs, and for each step, $n/2$ two-qubit gates are applied.

In Figure 5, the mean value and the variance of the purities are shown as a function of the number of qubits in a partition n_a for systems with different size n . The moments are estimated considering an ensemble of $N_e = 100$ pseudo-random states generated using the direct method with 20 steps.

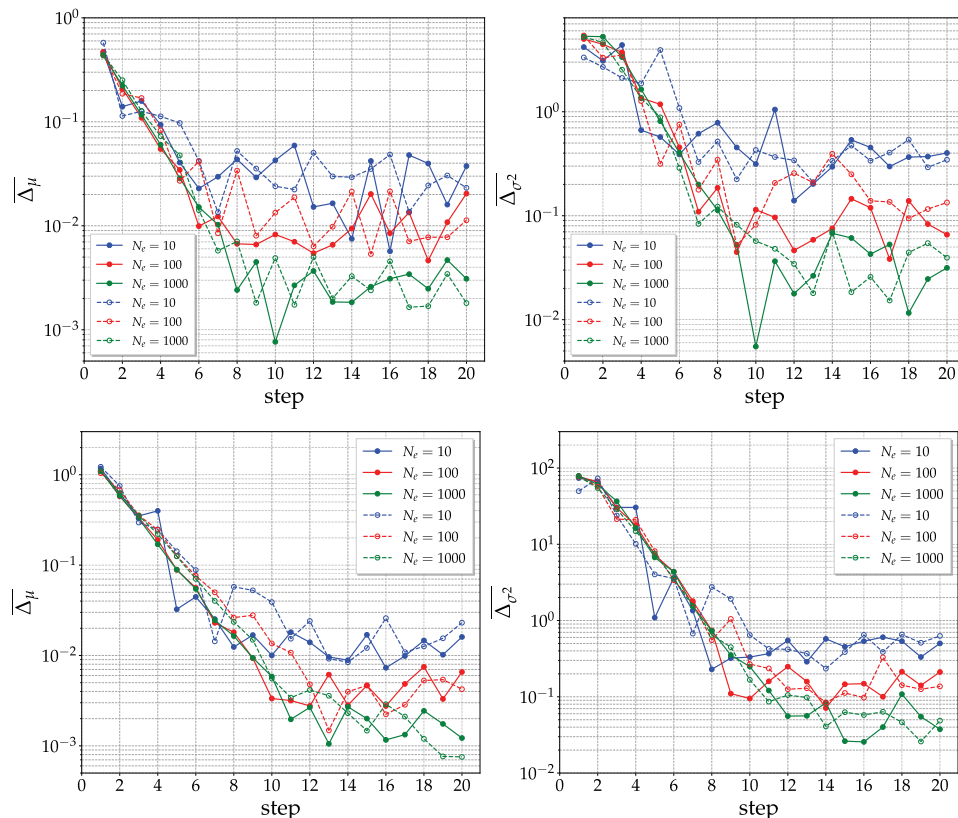


Figure 4. Cont.

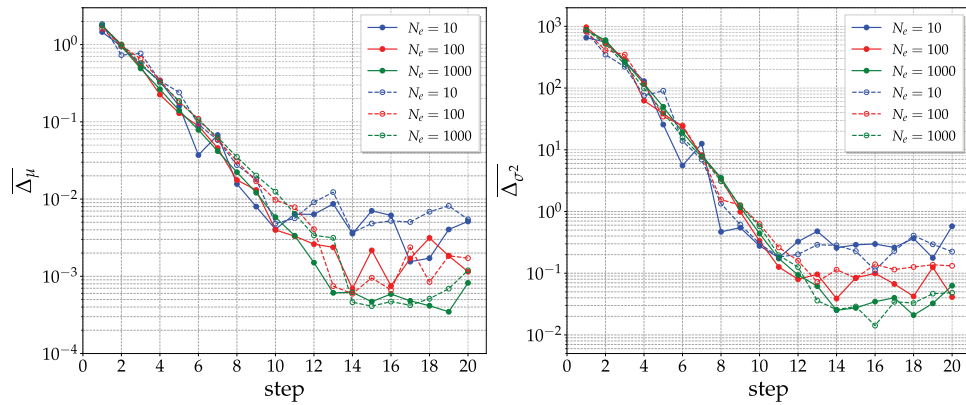


Figure 4. Average mean value relative error (left) and average variance relative error (right) for purities as a function of the number of steps (i.e., layers in the quantum circuit) and the ensemble size for 4-qubit (top), 6-qubit (middle), and 8-qubit (bottom) pseudo-random quantum state. The solid lines represent the direct method while the dashed lines represent the KAK method.

The convergence to the true random state expected values improve as the dimension of the system increases. Indeed, for higher dimensions, the entanglement content is highly typical, i.e., it is possible to show that the entanglement distribution for a random state becomes strongly peaked in the limit of a large number of qubits. This concentration of the measure explains the better convergence for higher-dimensional cases [34].

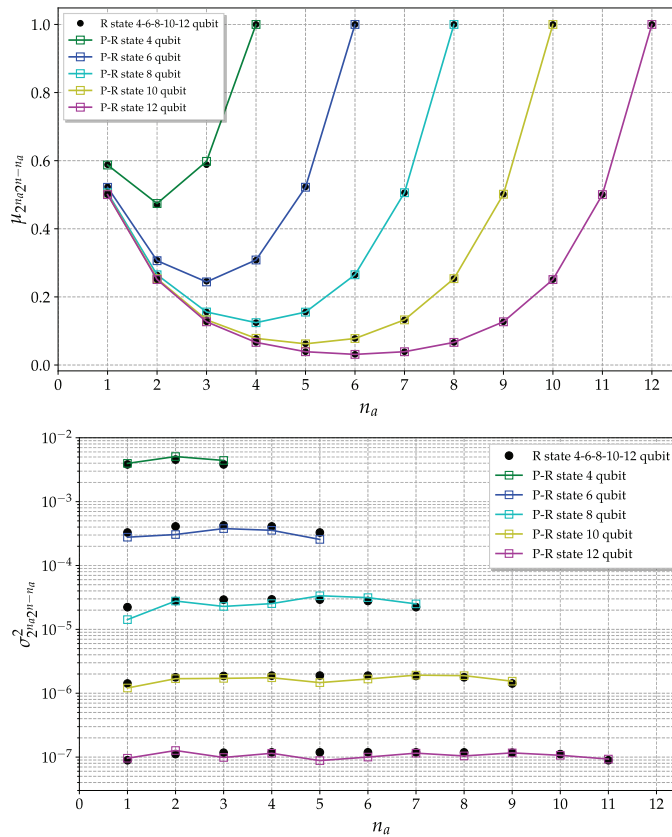


Figure 5. Purity mean value (top) and variance (bottom) of a pseudo-random quantum state plotted as a function of partition size. The various colors represent systems of different dimensions (number of qubits). The black dots are the expected values for a true random state. Here is shown the direct method with 20 steps and $N_e = 100$.

3. Results on Actual Quantum Hardware

The circuits we implemented on real quantum hardware (IBM's *ibm_lagos* and IonQ's *Harmony*, a visualization of which is given in Figure 6) are slightly different from that shown in Figure 3. First of all, given the available resources, only circuits with four and six qubits were considered. In order to limit circuit depth, the random permutation gates are avoided, and instead, since all the qubits must be entangled with each other, the D (or $SU(4)$) gates are applied to qubit pairs labeled as $\{(0,1), (2,3), (0,2), (1,3)\}$ (for the four-qubit case) and $\{(0,1), (2,3), (4,5), (1,2), (3,4), (0,5), (0,3), (1,4), (5,2)\}$ (for the six-qubit case). The purities of a random state are estimated using measurements along randomly rotated axes, following the method proposed by Brydges et al. [16].

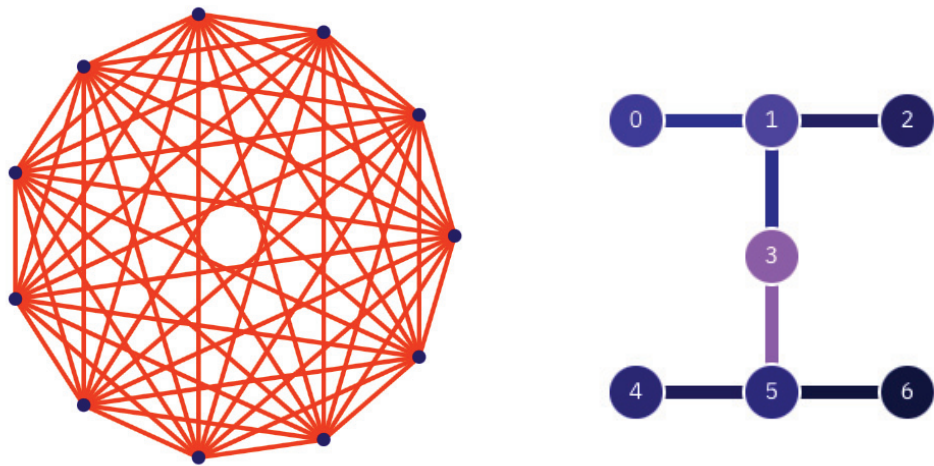


Figure 6. Architectures of the quantum processors used in this work. The circles represent the qubits while the lines represent the physical connection between them. On the left, we have the architecture of IonQ's *Harmony*, which clearly shows the complete connectivity of ion-based devices. On the right, we have *ibm_lagos*. Here, the color scheme (blue for min, violet for max) refers to the single-qubit (color of the circles) and two-qubit (color of the lines) error rates. These are purely indicative since the rates change upon every calibration of the device.

The ensemble of random states is $N_e = 10$ wide, and for each state, $N_m = 20$ random measurement axes are taken in order to estimate the purities. Each of these 200 circuits is followed by a measurement in the standard computational basis, and each circuit is repeated $N_s = 1000$ times (number of shots, limited by the available budget for IonQ) in order to estimate the outcome probabilities of each element of the computational basis for each circuit.

Note that IBM's quantum computers are nominally calibrated once over a 24-h period, and the system properties update once this calibration sequence is complete. Calibration plays a critical role in quantum circuit execution, since the properties of the systems are utilized for noise-aware circuit mapping and optimization (transpilation). Due to the daily calibration, it is difficult to compare the results obtained in different days on the same hardware. For this reason, all comparative results with the same quantum computer herein were taken with the same calibration data (i.e., the same day).

3.1. Comparison between Hardware Platforms

From the extensive tests performed in the literature [30] (see Table 1), we know that IBM's *ibm_lagos* has a better performance than IonQ's *Harmony* as far as mean fidelities for one- and two-qubit gates are considered. On the other hand, IonQ's *Harmony* is preferable when state preparation and measurement (SPAM) fidelities are considered. More importantly, IonQ's *Harmony* has the advantage of an all-to-all connectivity. This latter point is very relevant, because IBM's quantum processors need SWAP gates to implement D (or $SU(4)$) gates between qubits that aren't connected. Moreover, a SWAP

gate is not a native gate on the IBMQ devices, and must be decomposed into three CNOT gates. Being the product of three CNOT gates, SWAP gates are expensive operations to perform on a noisy quantum device.

Table 1. Table of quantum processing units (QPUs) evaluated in [30] using the quantum volume (QV) protocol. Values of QV, as well as single-qubit (1Q) gate, two-qubit (2Q) gate and state preparation and measurement (SPAM) fidelities are all vendor-provided metrics. The mean gate and SPAM fidelities are computed in [30] across all operations of the same type available on the device during the whole QV circuit execution duration. The number of edges for each backend was simply counted as the number of connections between qubits.

Vendor	Backend	QV	QPU			Fidelity		
			# Qubit	Topology	# Edges	2Q Gate	1Q Gate	SPAM
IBM Q	<i>ibm_lagos</i>	32	7	Falcon r5.11H	6	0.9924	0.9998	0.9862
IonQ	<i>Harmony</i>	8 *	11	All-to-All	55	0.96541	0.9972	0.99709

* The QV value for IonQ’s *Harmony* is the one measured in [30], since IonQ does not provide it.

The results obtained using the direct method are shown in Figure 7, both for IBMQ and IonQ. As can be seen from the figure, particularly in the IonQ case, the purity of the whole state is greater than the bipartitions’ reduced purities, with the exception of the $n_a = 1$ case for the IBMQ. This is equivalent to saying that the entropies of the parts are greater than the entropy of the whole state, which is a signature of bipartite entanglement in the system. Despite the fact that superconductor devices have lower error rates than IonQ for single-qubit and two-qubit gates, the overall purity is higher in trapped ion devices. This is mainly due to the complete connectivity of these machines, which allows avoiding noisy SWAP gates, in addition to the better SPAM fidelities of the ion-based device.

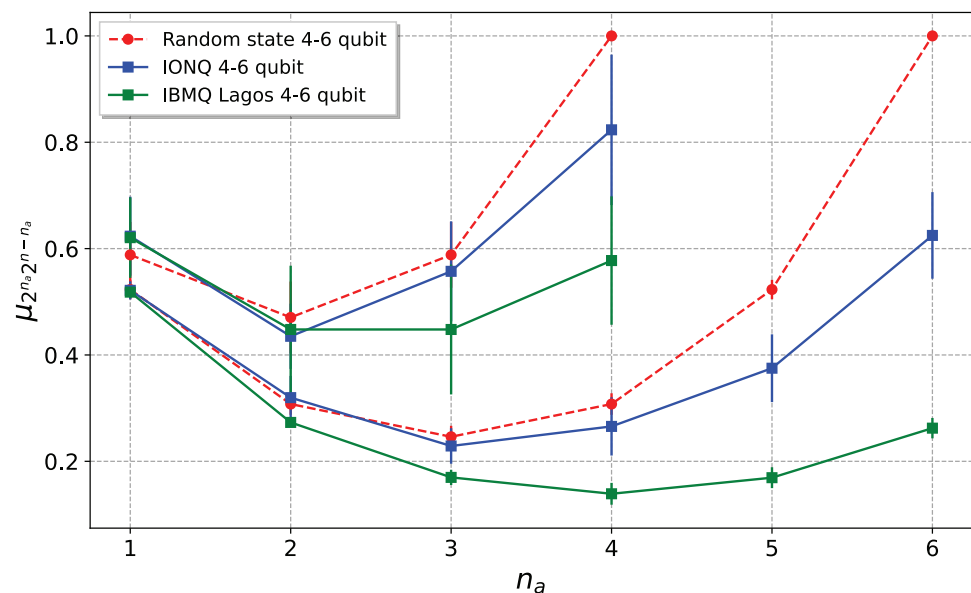


Figure 7. Comparison between the purities of a 4- and 6-qubit pseudo-random quantum states, generated in the two different realizations of a quantum computer investigated, with the direct method. In green, the superconductor IBM’s *ibm_lagos* is shown, while IonQ’s *Harmony* is shown in blue. Red curves give the results for ideal random states. Data were obtained on 10 September 2022, for *ibm_lagos* and on 24 July 2022, for *Harmony*.

3.2. Entanglement Evolution

To investigate the survival of entanglement in an operating quantum computer, we iterate the above circuit for the generation of pseudo-random quantum states for a number

of steps. In Figure 8, we consider *ibm_lagos* and $n = 4$ qubits, and show the purities as a function of the number of steps, for subgroups of n_a qubits. We can see that the purity of the overall system (ideally pure) is clearly higher than the purities of subsystems with $n_a = 2$ and $n_a = 3$ qubits up to 4 steps. For longer evolution times, the purity of the overall system drops below those of subsystems, and there is evidence, at least for $n_a = 1, 2$, of convergence to the purity for a maximally mixed state, equal to $1/2^{n_a}$. These values are smaller than those for pseudo-random states reported in Equation (5). Overall, the above remarks point to a vanishing entanglement content in the quantum hardware after 4–5 steps.

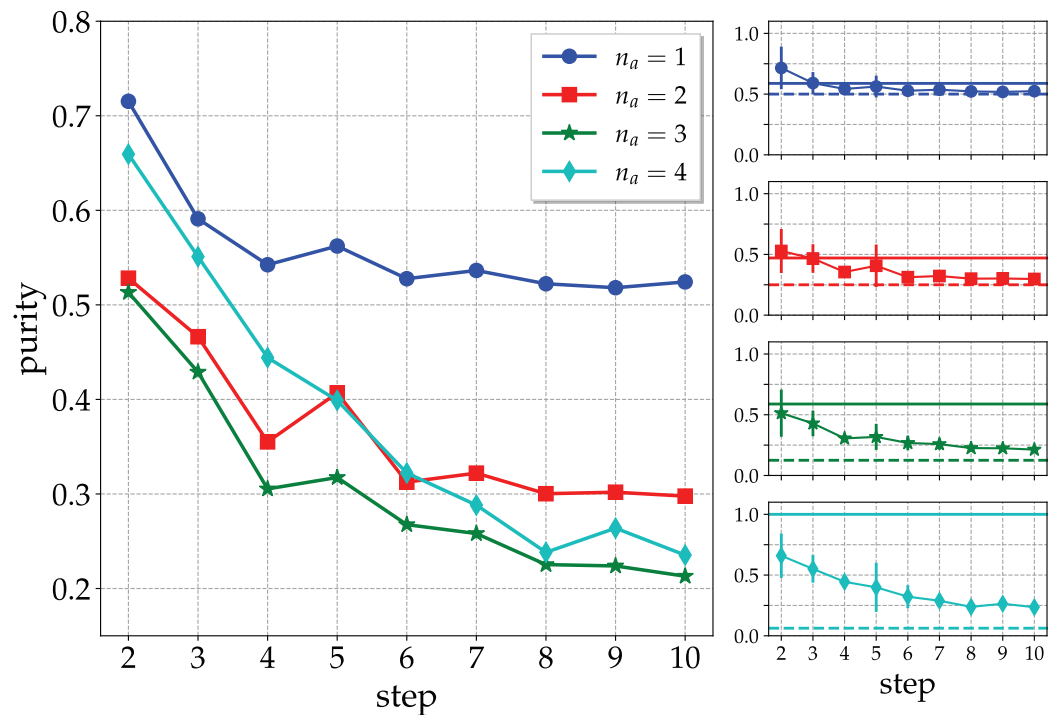


Figure 8. Evolution of the entanglement content of a pseudo-random quantum state generated by the circuit described in Figure 3 as a function of the number of layers (steps). The panels on the right show the individual curves, with the horizontal solid lines highlighting the purity expectation values for a true random state. The horizontal dashed lines refer to the purity of a maximally mixed state. Data taken from *ibm_lagos* on 29 January 2023.

4. Conclusions

We investigated the generation of random states for which the entanglement content is almost maximal on a quantum computer. We proposed a method in which the obtained pseudo-random states converge to true random states by concatenating layers in which random permutations of the qubit labels are followed by the generation of random states for pairs of qubits. We argue that our method is optimal, and that the number of CNOT gates is greatly reduced with respect to circuits implementing two-qubit random unitary gates. The effectiveness of our method has been tested in the current implementations of quantum hardware, both for superconducting and ion trap quantum processors. In the latest implementation, we highlighted the advantages of the all-to-all connectivity of qubits.

With regard to the attainment of the maximal entanglement of quantum states, it would be interesting to study the class of maximally multipartite n -qubit states proposed by Facchi et al. [35]. More generally, multipartite entanglement optimization is a difficult task, which could at the same time be an ideal testbed for investigating the complexity of quantum correlations in many-body systems and for developing variational hybrid quantum-classical algorithms [36–38].

Author Contributions: G.C. performed quantum simulations by coding actual IBM quantum processors. G.B. supervised the work with inputs from M.B. and D.R. All authors discussed the results and contributed to writing and revising the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: G.C. and G.B. acknowledge the financial support of the INFN through the project QUANTUM. D.R. acknowledges the support from the Institute for Basic Science in Korea (IBS-R024-D1).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used and analyzed in the current study is available from the corresponding author upon reasonable request.

Acknowledgments: We acknowledge use of the IBM Quantum Experience and the access to IonQ machines supported by Amazon Web Services. The views expressed in this work are those of the authors and do not reflect the official policy or position of AWS, IBM, and IonQ companies.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Random State Purities Moments

Recalling that $|\psi\rangle$ is a pure state that belongs to the Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, where \mathcal{H}_A and \mathcal{H}_B are spanned, respectively, by $\{|i_A\rangle\}_{1 \leq i_A \leq N_A}$ and $\{|i_B\rangle\}_{1 \leq i_B \leq N_B}$, A and B are two bipartitions of the entire system. The state, assuming $N_A \leq N_B$, admits a Schmidt decomposition [1]:

$$|\psi\rangle = \sum_{i=1}^{N_A} \sqrt{x_i} |a_i\rangle \otimes |b_i\rangle, \tag{A1}$$

where $\{|a_i\rangle\}$ and $\{|b_i\rangle\}$ are suitable basis sets for \mathcal{H}_A and \mathcal{H}_B .

For a pure random state, the Schmidt coefficients x_i are distributed according to the density [22]:

$$P(x_1, \dots, x_{N_A}) = \mathcal{N} \prod_{1 \leq i < j \leq N_A} (x_i - x_j)^2 \prod_{1 \leq k \leq N_A} x_k^{N_B - N_A} \delta\left(1 - \sum_{i=1}^{N_A} x_i\right), \tag{A2}$$

for $x_i \in [0, 1]$ and some normalization factor \mathcal{N} . From this distribution, it is possible to calculate the n -th moment of the purities, defined as [21]:

$$\begin{aligned} \langle R^n \rangle &= \mathcal{N} \int_0^1 dx_1 \dots dx_{N_A} (x_1^2 + x_2^2 + \dots + x_{N_A}^2)^n P(x_1, \dots, x_{N_A}) = \\ &= \frac{(N_A N_B - 1)!}{(N_A N_B + 2n - 1)!} \sum_{n_1 + n_2 + \dots + n_{N_A} = n} \frac{n!}{n_1! n_2! \dots n_{N_A}!} \times \\ &\times \prod_{n_i \neq 0} \left[\frac{(N_B + 2n_i - i)! (N_A + 2n_i - i)!}{(N_B - i)! (N_A - i)! (2n_i)!} \prod_{j=1}^{i-1} \left(1 - \frac{2n_j}{2n_i + j - i}\right) \right]. \end{aligned} \tag{A3}$$

Out of this last formula, it is easy to calculate the cumulants shown in Equations (5) and (6).

Appendix B. Cartan’s KAK Decomposition of the Unitary Group

The Cartan’s KAK decomposition can be used for constructing an optimal quantum circuit for achieving a general two-qubit quantum gate, up to a global phase, which requires at most 3 CNOT and 15 elementary one-qubit gates from the family $\{R_y, R_z\}$, i.e., single-qubit rotations obtained by exponentiating the corresponding Pauli matrices. It can be proven that this construction is optimal in the sense that there is no smaller circuit using the same family of gates, which achieves this operation [20].

Following the general prescription [39,40], one can decompose every $SU(4)$ element as depicted in Figure A1, where $A_j \in SU(2)$ are single-qubit unitaries decomposable into elementary one-qubit gates according to the well-known Euler decomposition. Note that, in order to randomly extract one of these operators, the angles of the single-qubit rotations must be extracted uniformly with respect to the Haar measure of the unitary group.

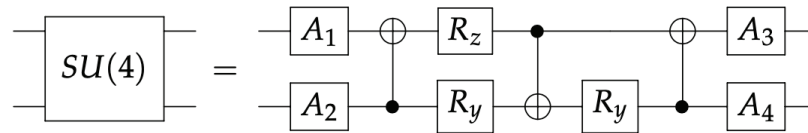


Figure A1. A quantum circuit implementing a two-qubit unitary gate using the KAK parametrization of $SU(4)$.

References

- Benenti, G.; Casati, G.; Rossini, D.; Strini, G. *Principles of Quantum Computation and Information (A Comprehensive Textbook)*; World Scientific: Singapore, 2019.
- Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
- Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef] [PubMed]
- Zhong, H.S.; Wang, H.; Deng, Y.H.; Chen, M.C.; Peng, L.C.; Luo, Y.H.; Qin, J.; Wu, D.; Ding, X.; Hu, Y.; et al. Quantum computational advantage using photons. *Science* **2020**, *370*, 1460–1463. [CrossRef] [PubMed]
- Daley, A.J.; Bloch, I.; Kokail, C.; Flannigan, S.; Pearson, N.; Troyer, M.; Zoller, P. Practical quantum advantage in quantum simulation. *Nature* **2022**, *607*, 667–676. [CrossRef]
- Liu, Y.A.; Liu, X.L.; Li, F.N.; Fu, H.; Yang, Y.; Song, J.; Zhao, P.; Wang, Z.; Peng, D.; Chen, H.; et al. Closing the “Quantum Supremacy” Gap: Achieving Real-Time Simulation of a Random Quantum Circuit Using a New Sunway Supercomputer. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, St. Louis, MI, USA, 18 November 2021; Association for Computing Machinery: New York, NY, USA, 2021. [CrossRef]
- Bulmer, J.F.F.; Bell, B.A.; Chadwick, R.S.; Jones, A.E.; Moise, D.; Rigazzi, A.; Thorbecke, J.; Haus, U.U.; Vaerenbergh, T.V.; Patel, R.B.; et al. The boundary for quantum advantage in Gaussian boson sampling. *Sci. Adv.* **2022**, *8*, eabl9236. [CrossRef]
- Zhou, Y.; Stoudenmire, E.M.; Waintal, X. What Limits the Simulation of Quantum Computers? *Phys. Rev. X* **2020**, *10*, 041038. [CrossRef]
- Cross, A.W.; Bishop, L.S.; Sheldon, S.; Nation, P.D.; Gambetta, J.M. Validating quantum computers using randomized model circuits. *Phys. Rev. A* **2019**, *100*, 032328. [CrossRef]
- Pizzamiglio, A.; Chang, S.Y.; Bondani, M.; Montangero, S.; Gerace, D.; Benenti, G. Dynamical Localization Simulated on Actual Quantum Hardware. *Entropy* **2021**, *23*, 654. [CrossRef]
- Keenan, N.; Robertson, N.; Murphy, T.; Zhuk, S.; Goold, J. Evidence of Kardar-Parisi-Zhang scaling on a digital quantum simulator. *arXiv* **2022**, arXiv:2208.12243.
- Plenio, M.B.; Virmani, S. An Introduction to Entanglement Measures. *Quantum Info. Comput.* **2007**, *7*, 1–51. [CrossRef]
- Horodecki, R.; Horodecki, P.; Horodecki, M.; Horodecki, K. Quantum entanglement. *Rev. Mod. Phys.* **2009**, *81*, 865–942. [CrossRef]
- Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2000.
- Facchi, P.; Florio, G.; Pascazio, S. Probability-density-function characterization of multipartite entanglement. *Phys. Rev. A* **2006**, *74*, 042331. [CrossRef]
- Brydges, T.; Elben, A.; Jurcevic, P.; Vermersch, B.; Maier, C.; Lanyon, B.P.; Zoller, P.; Blatt, R.; Roos, C.F. Probing Rényi entanglement entropy via randomized measurements. *Science* **2019**, *364*, 260–263. [CrossRef] [PubMed]
- Brandão, F.G.; Chemsassy, W.; Hunter-Jones, N.; Kueng, R.; Preskill, J. Models of Quantum Complexity Growth. *PRX Quantum* **2021**, *2*, 030316. [CrossRef]
- Hayden, P.; Preskill, J. Black holes as mirrors: Quantum information in random subsystems. *J. High Energy Phys.* **2007**, *2007*, 120. [CrossRef]
- Choi, J.; Shaw, A.L.; Madjarov, I.S.; Xie, X.; Finkelstein, R.; Covey, J.P.; Cotler, J.S.; Mark, D.K.; Huang, H.Y.; Kale, A.; et al. Preparing random states and benchmarking with many-body quantum chaos. *Nature* **2023**, *613*, 468–473. [CrossRef]
- Vatan, F.; Williams, C. Optimal quantum circuits for general two-qubit gates. *Phys. Rev. A* **2004**, *69*, 032315. [CrossRef]
- Giraud, O. Distribution of bipartite entanglement for random pure states. *J. Phys. A Math. Theor.* **2007**, *40*, 2793. [CrossRef]
- Lloyd, S.; Pagels, H. Complexity as thermodynamic depth. *Ann. Phys.* **1988**, *188*, 186–213. [CrossRef]
- Emerson, J.; Weinstein, Y.S.; Saraceno, M.; Lloyd, S.; Cory, D.G. Pseudo-Random Unitary Operators for Quantum Information Processing. *Science* **2003**, *302*, 2098–2100. [CrossRef]
- Emerson, J.; Livine, E.; Lloyd, S. Convergence conditions for random quantum circuits. *Phys. Rev. A* **2005**, *72*, 060302. [CrossRef]
- Weinstein, Y.S.; Hellberg, C.S. Entanglement Generation of Nearly Random Operators. *Phys. Rev. Lett.* **2005**, *95*, 030501. [CrossRef] [PubMed]

26. Dahlsten, O.C.O.; Oliveira, R.; Plenio, M.B. The emergence of typical entanglement in two-party random processes. *J. Phys. A Math. Theor.* **2007**, *40*, 8081. [CrossRef]
27. Oliveira, R.; Dahlsten, O.C.O.; Plenio, M.B. Generic Entanglement Can Be Generated Efficiently. *Phys. Rev. Lett.* **2007**, *98*, 130502. [CrossRef]
28. Žnidarič, M. Optimal two-qubit gate for generation of random bipartite entanglement. *Phys. Rev. A* **2007**, *76*, 012318. [CrossRef]
29. Pozniak, M.; Życzkowski, K.; Kus, M. Composed ensembles of random unitary matrices. *J. Phys. A Math. Gen.* **1998**, *31*, 1059. [CrossRef]
30. Pelofske, E.; Bärttschi, A.; Eidenbenz, S. Quantum Volume in Practice: What Users Can Expect From NISQ Devices. *IEEE Trans. Quantum Eng.* **2022**, *3*, 1–19. [CrossRef]
31. Humphreys, J.E. *Introduction to Lie Algebras and Representation Theory*; Springer Science & Business Media: Cham, Switzerland, 2012; Volume 9.
32. Bengtsson, I.; Życzkowski, K. *Geometry of Quantum States: An Introduction to Quantum Entanglement*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2017. [CrossRef]
33. Giraud, O.; Žnidarič, M.; Georgeot, B. Quantum circuit for three-qubit random states. *Phys. Rev. A* **2009**, *80*, 042309. [CrossRef]
34. Dahlsten, O.C.O.; Lupo, C.; Mancini, S.; Serafini, A. Entanglement typicality. *J. Phys. A Math. Theor.* **2014**, *47*, 363001. [CrossRef]
35. Facchi, P.; Florio, G.; Parisi, G.; Pascazio, S. Maximally multipartite entangled states. *Phys. Rev. A* **2008**, *77*, 060304. [CrossRef]
36. McClean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **2016**, *18*, 023023. [CrossRef]
37. Moll, N.; Barkoutsos, P.; Bishop, L.S.; Chow, J.M.; Cross, A.; Egger, D.J.; Filipp, S.; Fuhrer, A.; Gambetta, J.M.; Ganzhorn, M.; et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Sci. Technol.* **2018**, *3*, 030503. [CrossRef]
38. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
39. Khaneja, N.; Glaser, S.J. Cartan decomposition of SU (2n) and control of spin systems. *Chem. Phys.* **2001**, *267*, 11–23. [CrossRef]
40. Khaneja, N.; Brockett, R.; Glaser, S.J. Time optimal control in spin systems. *Phys. Rev. A* **2001**, *63*, 032308. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

A Variational Quantum Linear Solver Application to Discrete Finite-Element Methods

Corey Jason Trahan ^{1,*}, Mark Loveland ¹, Noah Davis ² and Elizabeth Ellison ¹

¹ Information and Technology Laboratory, U.S. Army Engineer Research and Development Center, Vicksburg, MS 39180, USA

² Applied Research Laboratories, The University of Texas at Austin, Austin, TX 78713, USA

* Correspondence: corey.j.trahan@erdc.dren.mil

Abstract: Finite-element methods are industry standards for finding numerical solutions to partial differential equations. However, the application scale remains pivotal to the practical use of these methods, even for modern-day supercomputers. Large, multi-scale applications, for example, can be limited by their requirement of prohibitively large linear system solutions. It is therefore worthwhile to investigate whether near-term quantum algorithms have the potential for offering any kind of advantage over classical linear solvers. In this study, we investigate the recently proposed variational quantum linear solver (VQLS) for discrete solutions to partial differential equations. This method was found to scale polylogarithmically with the linear system size, and the method can be implemented using shallow quantum circuits on noisy intermediate-scale quantum (NISQ) computers. Herein, we utilize the hybrid VQLS to solve both the steady Poisson equation and the time-dependent heat and wave equations.

Keywords: quantum computing; quantum variational algorithm; finite-element methods; Poisson equation; heat equation; quantum algorithms

1. Introduction

Quantum computing has reached a new era where theory is transitioning into practice as quantum computers and simulators become more widespread and available to the scientific community. This transition has encouraged algorithmic exploration, with an intent toward showing “quantum supremacy” or “quantum advantage”. Quantum advantage refers to the demonstrated and measured success in processing a real-world problem faster on a quantum computer than on a classic computer. Quantum supremacy [1], on the other hand, refers to the demonstrated and measured ability to process **any problem** faster on a quantum computer, regardless of its real-world applicability [2].

In 2019, Arute et al. [3] claimed to have achieved quantum supremacy using a programmable superconducting processor by “performing a series of operations in 200 s that would take a supercomputer about 10,000 years to complete”. In December 2020, a group based out of the University of Science and Technology of China (USTC) led by Jian-Wei Pan claimed quantum supremacy by implementing Gaussian boson sampling on 76 photons with their photonic quantum computer [4]. The paper states that to generate the number of samples the quantum computer generates in 20 s, a classical supercomputer would require 600 million years of computation. Although these supremacy claims have been the source of much recent debate, mostly with respect to whether or not their classical comparisons are the most efficient, it is clear that we are on the threshold of a new age of computation, heralded by today’s noisy intermediate-scale quantum (NISQ) hardware.

Today’s NISQ computers are limited in scalability because they are (1) subject to noise and thus not fault-tolerant, and (2) they are qubit-limited (usually meaning less than 100 qubits). Regarding the latter, however, the number of qubits on modern day quantum computers is rapidly growing, with IBM projecting a remarkable 1121 qubit system in

Citation: Trahan, C.J.; Loveland, M.; Davis, N.; Ellison, E. A Variational Quantum Linear Solver Application to Discrete Finite-Element Methods. *Entropy* **2023**, *25*, 580. <https://doi.org/10.3390/e25040580>

Academic Editor: Rosario Lo Franco

Received: 21 February 2023

Revised: 21 March 2023

Accepted: 22 March 2023

Published: 28 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

2023. Although this exponential qubit growth is vital in the near term, “shot” noise arising from the Heisenberg uncertainty principle and zero-point thermal fluctuations cause a phenomenon called decoherence, which may ultimately prevent scalability to larger qubit applications. Quantum systems achieve their notable advantage over classical ones via entanglement, a process by which a pure state quantum system develops a probability distribution over multiple classical outcomes. Entanglement gives quantum computers the ability to process and store exponentially more information than a classical computer. Noise, however, introduces errors that cause decoherence in the entanglement and can significantly degrade the performance of NISQ computers [5–8]. In fact, much of today’s quantum computing efforts are in noise mitigation [9–14]. In January 2022, for example, a group of scientists from the University of Chicago and Purdue University collaborated on a new promising noise control technique: Instead of directly trying to measure the noise, they constructed a unique “fingerprint” of the noise on a quantum computer as it was seen by a program run on the computer [15]. This approach shows promise for mitigating the noise problem, as well as suggesting ways that users could actually turn noise into an advantage.

Despite these drawbacks, NISQ computers remain promising in application areas such as quantum chemistry, cybersecurity, drug development, financial modeling, traffic optimization, weather forecasting, climate change prediction, artificial intelligence and machine learning. Over the last few years, quantum hardware has become available to the average researcher, mostly through two types of cloud computing. The first type is cloud services providing access to a single company’s collection of quantum devices. The Qiskit cloud service offered by IBM Quantum [16] is the premier example of this. On the other hand, there are multi-platform services such as Amazon Braket [17] that work as intermediaries to give users options to access quantum devices owned by multiple vendors. In most cases, cloud computing interfaces for quantum devices are implemented in Python to provide starting points for accessing working quantum devices. Introductory resources for algorithmic understanding and design are also widely available to the public. For example, the IBM Qiskit textbook [18] provides a college-level introduction to quantum information with integrated programming exercises, the Codebook by Xanadu [19] provides an introductory course built around the PennyLane package, allowing for differentiable programming of quantum computers, and QBraid is an online platform for developing quantum software with introductory quantum tutorials [20].

As near-term supremacy does not mean utility, many today utilize these current cloud resources for the investigation of quantum advantages for practical problems. NISQ computers must be restricted to “shallow” circuits for noise control. These circuits have a minimal number of qubits that are more easily controlled. One way of keeping quantum circuits shallow, for example, is by combining quantum and classical algorithms so that only the computationally intensive portion of the problem is implemented on the quantum computer, thereby offering some degree of quantum speed-up or advantage while maintaining shallow circuits amenable to NISQ computers. This type of hybrid set-up is somewhat analogous, for example, to classical GPU acceleration. Recently, hybrid methods such as these have been utilized for near-term acceleration of machine learning and optimization problems [21–29]. A number of quantum algorithms for machine learning are based on the idea of amplitude encoding, which associates the amplitudes of a quantum state with the inputs and outputs of computations [24,30,31]. Since a state of m qubits is described by 2^m complex amplitudes, this information encoding can allow for an exponentially compact representation. Intuitively, this corresponds to associating a discrete probability distribution over binary random variables with a classical vector. The goal of algorithms based on amplitude encoding is to formulate quantum algorithms whose resources grow polynomially in the number of qubits m , which amounts to a logarithmic time complexity in the number of amplitudes and therefore the dimension of the input.

Many quantum machine learning algorithms are based on variations in the quantum algorithm for linear systems of equations [32] (colloquially called HHL after the paper’s

authors) which, under specific conditions, perform a matrix inversion using an amount of physical resources growing only logarithmically in the dimensions of the matrix. One of these conditions is that a Hamiltonian which, entry-wise, corresponds to the matrix can be simulated efficiently, which is known to be possible if the matrix is sparse [33] or low in rank [23]. Quantum matrix inversion can be applied to machine learning methods in which the training reduces to solving a linear system of equations, such as in least squares linear regression [30,31], the least squares version of support vector machines [24], and Gaussian processes [34].

For suitably conditioned linear systems, the HHL algorithm scales logarithmically in n , suggesting the possibility of exponential speed-up over classical systems [32], which holds promise for quantum computers beyond the NISQ era. In today's NISQ machines, however, shot noise has dramatically limited the size of the linear systems directly solvable by the HHL algorithm. To date, 2×2 systems have been solved by superconducting qubits [35,36], nuclear magnetic resonance [37], and photonic devices [38,39]. The largest system solved on a gate-based computer was an 8×8 problem using NMR [40].

Given today's NISQ limitations of the HHL algorithm, an alternative method for linear system solution has been proposed to gain a quantum advantage: variational hybrid quantum-classical algorithms (VHQCAs). VHQCAs are capable of providing an advantage to Shor's algorithm for factoring [41] and have gained momentum in the fields of chemistry [42–45], simulation [46–50], data compression [51], state diagonalization [52–54], compiling [55,56], quantum foundations [57], fidelity estimation [58], and meteorology [59]. The general VHQA algorithm reduces the quantum circuit depth by using a classical optimizer and only evaluating the cost/objective function on the quantum computer.

In this study, we continue to investigate quantum advantages in classical problems by utilizing a VHQA recently introduced by Bravo-Prieto et al. [60,61] called the variational quantum linear solver (VQLS) to obtain finite-element solutions to the Poisson, heat, and wave equations. The quantum/classical hybrid VQLS is a method for solving linear systems on near-term quantum computers which variationally prepares a quantum state $|x\rangle$ such that $\mathbf{A}|x\rangle \propto |b\rangle$. Bravo-Prieto et al. were able to derive a meaningful termination condition for VQLS that allows one to guarantee a desired solution precision with efficient quantum circuits to estimate the variational cost function C while providing evidence for the classical hardness of its estimation. Using Rigetti's quantum computer, the VQLS was used for solutions up to a problem size of 1024×1024 (10 qubits), which is the largest implementation of a linear system on quantum hardware to date. The time complexity of the VQLS was heuristically found to scale efficiently with the linear solution precision ϵ , the matrix condition number κ , and the linear system size N .

2. The Variational Quantum Linear Solver

The quantum/classical hybrid VQLS [60,61] algorithm attempts to find a solution to the linear system such that $\mathbf{A}|x\rangle \propto |b\rangle$ by minimizing a scalar cost function based on the scaled projection of $\mathbf{A}|x\rangle$ onto $|b\rangle$. The solution vector $|x\rangle$ is approximated with a wave function created through a quantum circuit ansatz. To prepare a linear system for VQLS solution, the matrix \mathbf{A} must be expressed as a linear combination of universal quantum gates. Additionally, the right-hand side (RHS) of the linear system must be transformed into a normalized quantum state $|b\rangle$, which can be generated by unitary operations U applied to the ground state of some number of qubits. We now discuss these elements of the VQLS in detail.

2.1. The Variational Ansatz

In the VQLS algorithm, $|x\rangle$ is prepared by acting on the $|0\rangle$ state with a trainable gate sequence $V(\alpha)$. The ansatz $V(\alpha)$ can be expressed in terms of L gates from a gate alphabet $A = G_k(\alpha)$ as

$$V(\alpha) = G_{k_L}(\alpha_L) \cdots G_{k_i}(\alpha_i) \cdots G_{k_1}(\alpha_1) \quad (1)$$

Here, $\vec{k} = (k_L, \dots, k_1)$ identifies the types of gates and their placement in the circuit (i.e., on which qubit they act), while α represents the continuous parameters over which optimization occurs. All results presented herein are based on a “fixed ansatz”, where \vec{k} is fixed over time and V is only optimized over α . Though it was not investigated in this study, variable ansatz optimization was shown to improve convergence in some cases in [52,62].

Training of the ansatz is performed layer by layer, just as in neural networks. The number of layers is decided by the user. Although the solution function space widens as the layers are increased, over-determined parameter optimization may become difficult and inefficient. The properties of a good ansatz are as follows: (1) the circuit is shallow, minimizing decoherence, (2) it has minimal optimization parameters, and (3) the ansatz should span the space where the solution lives. Of all the layer structures we tested, the ansatz given in [60] (shown in Figure 1) was the most optimal one. This ansatz begins with an initial y rotation (Ry) of each qubit before moving on to the layered portion of the circuit.

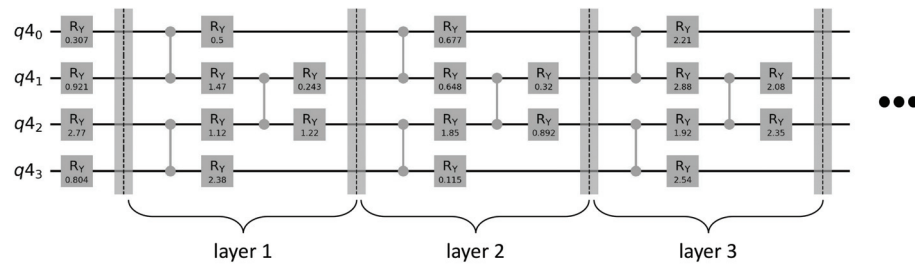


Figure 1. A four-qubit example of the fixed ansatz used for this study.

Each layer starts with alternating controlled-z (CZ) gates followed by Ry rotations on the controlled qubits. The CZ gates have the crucial function of entangling the qubits, which allows for an exponentially larger space representation than a purely classical cost evaluation. The Ry gates allow one to “search” the state space by varying the rotational parameters.

In this study, a range of layers were tested for each application of the VQLS. Some general guidelines for choosing the number of layers were found: (1) a greater number of layers was needed, as the problem’s dimensionality was increased (resulting in larger linear systems), and (2) a greater number of layers was required, as the number of terms in the Pauli decomposition of the stiffness matrix grew. These two factors greatly limited the size of the finite-element problems we could test at this time to a maximum of 10 nodes (8 internal nodes or 3 qubits).

2.2. Matrix Pauli Decomposition

In order to solve the linear system using the VQLS, the matrix must be represented as a linear combination of Hermitian unitary operators $\mathbf{A} = \sum_i c_i \mathbf{U}_i$, representing a system Hamiltonian where U_i represents the unitaries and c_i represents complex coefficients. Additional assumptions are that the matrix condition number $\kappa < \inf$ and $\|\mathbf{A}\| \leq 1$ and that the \mathbf{A}_i unitaries can be implemented with efficient quantum circuits. Typically, this decomposition consists of a linear combination of Kronecker products of the Identity and Pauli matrices, as these gates are widely used and recognized. These matrices and gates are defined as follows:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2}$$

For all application matrices herein, a recently proposed algorithm given in [63], which takes a square real symmetric matrix of an arbitrary size and decomposes it into a tensor product of Pauli spin matrices, was used. The routine was given by the authors in Python and is publicly available. The mathematical procedure for generating this decomposition

for a general-sized stiffness matrix, often encountered in discrete finite-element methods, is given in Appendix A.

2.3. Right-Hand Side Preparation

The VQLS requires that the linear system RHS be transformed into a normalized quantum state $|b\rangle$ generated by some series of unitary operations \mathbf{U} applied to the ground state of the qubits:

$$|b\rangle = \mathbf{U} |0\rangle \quad (3)$$

Again, we assume that \mathbf{U} can be efficiently implemented with a quantum circuit. For example, if the boundary conditions are homogeneous, and a reduced linear system is used which includes only the internal domain grid points, then the constant RHS wave function can be created by a quantum circuit which applies a Hadamard gate to each qubit:

$$|b\rangle = (\mathbf{H}_0 \mathbf{H}_1 \mathbf{H}_1 \cdots \mathbf{H}_{m-1}) |0\rangle \quad (4)$$

where m is the total number of qubits used to represent the reduced system. In general, however, the RHS vector of the linear systems will not be constant, and a vector-specific circuit must be generated. For the applications herein, we utilized the “isometry” package in Qiskit to produce the corresponding quantum state from a specific RHS vector. It is worth noting that more general, non-constant RHSs may lead to deeper, more complex circuitry that may affect the VQLS’s efficiency, since this circuit is evaluated in a controlled manner during each cost calculation.

3. Computational Details

The VQLS in this study was implemented in Python using IBM’s Qiskit [16]. Qiskit is an open source software development kit for working with OpenQASM and the IBM Q quantum processors. For prototypical applications, such as those needed for the early stages of this work, Qiskit offers a quantum computer simulator which allows the user to build and test quantum circuits on a local machine without the need for a quantum computer. The Qiskit package, along with its statevector simulator, can be imported into a Python script in the usual way. For all problems in this study, the Qiskit Aer simulator backend was used.

4. Training Algorithm

Scientific Python (SciPy) offers a variety of options for both constrained and unconstrained optimization of scalar objective/cost functions. The purpose of these optimizers is to update the parameters of the VQLS ansatz. Generally speaking, multi-variant objection function optimizers fall into two categories: gradient- and non-gradient-based optimization. Gradient-based methods, such as the Newton conjugate gradient method, use the objective function gradients (i.e., Jacobians or Hessians) to move in a descending direction toward a minima. Non-gradient methods, on the other hand, work by iteratively approximating the actual constrained optimization problem with linear programming problems. During an iteration, an approximating linear programming problem is solved to obtain a candidate for the optimal solution. The candidate solution is evaluated using the original objective and constraint functions, yielding a new data point in the optimization space. This information is used to improve the approximating linear programming problem used for the next iteration of the algorithm. When the solution cannot be improved anymore, the step size is reduced, refining the search. When the step size becomes sufficiently small, the algorithm finishes.

Previous studies have compared gradient- and non-gradient based optimization for a range of VQLS applications using quantum simulators, quantum simulators with shot noise, and fully quantum applications. In particular, in [64], it was shown that once shot noise is included in either the statevector simulator or real quantum application, gradient-based optimizers do not offer much of an advantage over non-gradient optimizers. A

popular choice for VQLS applications, for example, is the non-gradient based constrained optimization by linear approximation (COBYLA) method. Due to these previous findings and the complexity of including the objective function gradients, the COBYLA method was used for all applications herein, and gradient-based methods were not investigated.

5. Applications

5.1. Application 1: The Poisson Equation

For the first application, the QVA was used to solve the Dirichlet problem for the 1D Poisson equation, given in strong form by

$$-\Delta u(x) = f(x), u(x) \in \Omega, \tag{5}$$

where $u(0) = u_L$ and $u(1) = u_R$. The equivalent weak representation of this equation is obtained by taking Equation (5) and multiplying it by an arbitrary test function in the appropriate function space, followed by integrating by parts [65] to give

$$\int_{u_L}^{u_R} \frac{d\phi}{dx} \frac{du}{dx} dx = \int_{u_L}^{u_R} \phi f(x) dx \quad \forall \phi \in H_0^1(\Omega) \tag{6}$$

Here, $\phi(x)$ is the arbitrary test function in the appropriate Hilbert space, and the boundary term from integrating by parts vanishes since the test space H_0^1 has 0 trace. To discretize this equation, the standard Galerkin approximation with linear Lagrange polynomials is used on a uniform 1D grid of N points, where the i th nodal location is given by $x_i = ih$. Here, $h = 1/(N - 1)$ and $0 \leq i \leq (N - 1)$. Additionally, we define $n = N - 2$ as the internal node count. This discretization results in the linear system

$$\mathbf{K}\vec{u} = \vec{f} \tag{7}$$

where for linear, Lagrangian basis function support, \mathbf{K} is the typical tridiagonal “stiffness” matrix, \vec{u} is the solution vector, and \vec{f} is the right-hand side. When applying non-homogeneous Dirichlet boundary conditions, it is essential to manipulate this linear system to force the specified solution values on the domain endpoints, giving the following RHS:

$$\vec{f} = \begin{pmatrix} \int_0^1 \phi_1 f(x) dx + \int_0^1 \frac{\partial \phi_1}{\partial x} \frac{\partial \phi_0}{\partial x} u_L dx \\ \int_0^1 \phi_2 f(x) dx \\ \int_0^1 \phi_3 f(x) dx \\ \vdots \\ \int_0^1 \phi_n f(x) dx + \int_0^1 \frac{\partial \phi_n}{\partial x} \frac{\partial \phi_{n+1}}{\partial x} u_R dx \end{pmatrix} \tag{8}$$

For Dirichlet boundary conditions, a reduced system can be solved without the endpoints, since these are known. The reduced matrices were used for all applications herein to increase the grid resolution, since the qubit count was extremely limited. While obtaining the quantum wavefunction for the RHS of the homogeneous Poisson equation is relatively straightforward, heterogeneous boundaries or time-dependent solutions require more complex ways of calculating the RHS wavefunction on the fly. As mentioned, this was accomplished using Qiskit’s Isometry package. An example for creating a wavefunction from an arbitrary vector U is as follows:

```
qc = QuantumCircuit(4)
U = [0.1, 2, 2, 2, 2, 2, 2, 0.1]
U /= np.linalg.norm(U)
qc.isometry(U, [0, 1, 2], [])
qc = transpile(qc, basis_gates = ['u3', 'cx'], optimization_level=3)
```

This circuit is shown in Figure 2.

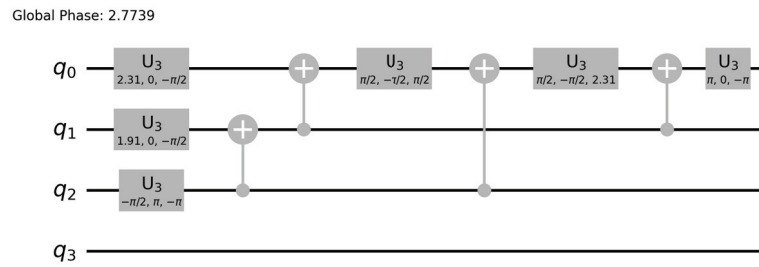


Figure 2. A quantum circuit representing $\vec{f}^T = [0.1, 2, 2, 2, 2, 2, 2, 0.1]$ found using Qiskit’s Isometry command.

5.1.1. Poisson Case 1: Parabolic Solution with Homogeneous Boundary Conditions

For the first Poisson test, a manufactured quadratic solution for Equation (5) was used to simplify the RHS preparation. The solution was given by

$$u(x) = a + b(x - x_0)^2 \tag{9}$$

where $a = g - b(-x_0)^2$, $u(0) = u(1) = g$ and $x_0 = 1/2$. The RHS of Equation (5) then simplifies to a constant $-2b$. For homogeneous boundary conditions, where $g = 0$, the reduced RHS of Equation (7) can be written as

$$\vec{f} = h[-2b \ -2b \ -2b \ -2b \ -2b \ -2b]^T \tag{10}$$

where h is the uniform grid spacing and T is the transpose.

This linear system was solved using the fixed-ansatz VQLS as described with the Pauli decomposition given in Appendix A and the right-hand side preparation detailed in Section 2.3. For the quantum simulator results without shot noise, errors arose only from discretization of spatial derivatives and the VQLS optimization. The number of qubits m in the VQLS determines the grid resolution such that the total number of nodes is $N = 2^m + 2$. Our attempts at optimization for anything greater than three qubits (eight nodes) took too long to simulate on a serial machine. This low qubit count leads to very coarse grids and noticeable discretization errors. To properly converge the discrete problem, finer grids were needed. All ansatz parameters were initialized randomly between $-\pi \leq \theta_k \leq \pi$, default optimizer tolerances of 10^{-4} were used, and the initial change to the variables in the COBYLA optimizer was set to $rhobeg = \pi$.

For the two-qubit homogeneous Poisson application, there were four internal and six total finite element nodes. The convergence results for the VQLS for a range of ansatz layers can be seen in Figure 3. These results were averaged over 20 runs, with solid lines indicating the average and variances shown with vertical bars. The two-qubit linear system’s stiffness was a 4×4 matrix. This figure shows that 2 layers were sufficient to successfully capture the solution to the default tolerance within 100 optimization steps. Note that the total number of optimization parameters N_θ varied as $N_\theta = m + 2(m - 1)(nlayers - 1)$, so for a two-qubit and two-layer network, there were four parameters to span the solution space. This figure shows that as the number of layers or parameters increased, the optimization converged slower, though still relatively fast when compared with the three-qubit problem. This was expected, however, since the solution test space dimensionality was increasing, and the variational algorithm had to span this space. For all layer cases, full solution convergence was achieved within the COBYLA tolerance using the statevector simulator in less than 100 iterations. Figure 4 plots the wall clock time in seconds versus the number of layers averaged over the 20 runs for the 2 qubit problem. From this figure, it is seen that the time it took to converge the solution was linearly proportional to the number of layers used in the variational ansatz.

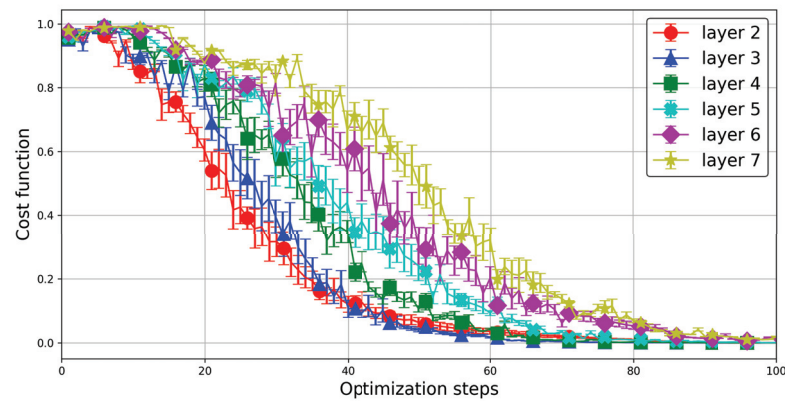


Figure 3. Two-qubit VQLS cost function results for the reduced Poisson problem with homogeneous Dirichlet boundary conditions. The results were averaged over 20 trial runs. Variances are shown by respective bars.

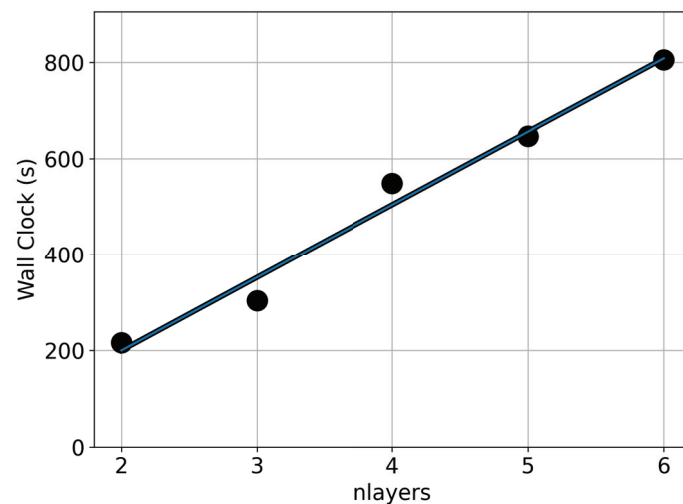


Figure 4. Wall clock time in seconds versus the number of layers for the two-qubit VQLS reduced Poisson problem with homogeneous Dirichlet boundary conditions.

Figure 5 displays the cost function of the 3 qubit statevector solution averaged over 10 runs. For this case, there were 8 internal nodes and 10 total, and the linear system stiffness was an 8×8 matrix. While the two-qubit results converged relatively fast for a small number of layers, this was not the case for the three-qubit application. Additionally, it took 4 or more layers for the cost function to converge within 1000 iterations. An interesting note from this figure is that the even-numbered layers performed notably better than the odd layers, with six layers converging in the least amount of time and most accurately. This can be seen more clearly in Figures 6 and 7, which display solution results and the grid root mean square errors averaged over all runs for each layer, respectively. Lastly, Figure 8 displays the time in seconds averaged over all 10 runs for each layer. Since the three-layer run never fully converged within the COBLYA default tolerance, it took the longest. All layers greater than three once again showed a linear increase in time as the layers were incremented.

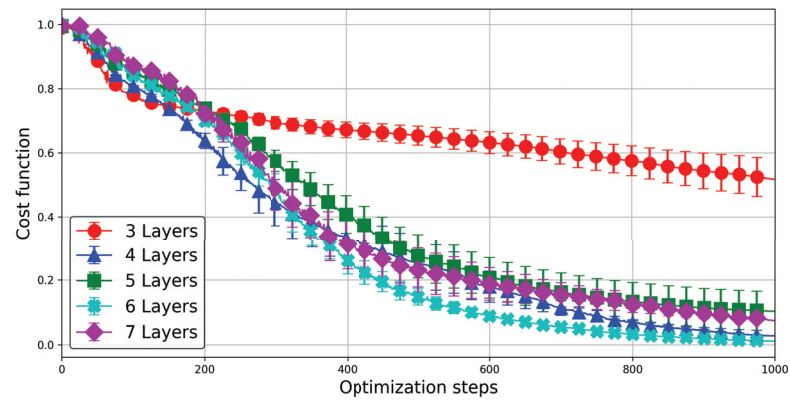


Figure 5. Three-qubit VQLS cost function results for the reduced Poisson problem with homogeneous Dirichlet boundary conditions. The results were averaged over 10 trial runs. Variances are shown by respective bars.

Figure 6 displays the VQLS versus the classical discrete solution for the three-qubit, eight-internal node problem. In this figure, we see the VQLS solution growing in accuracy as the number of ansatz layers is increased, as expected. In the right column, the VQLS solutions are plotted along with the analytic system solution. Note that the VQLS solution here is being compared to the discrete finite-element solution, and thus both include discretization errors which are not shown.

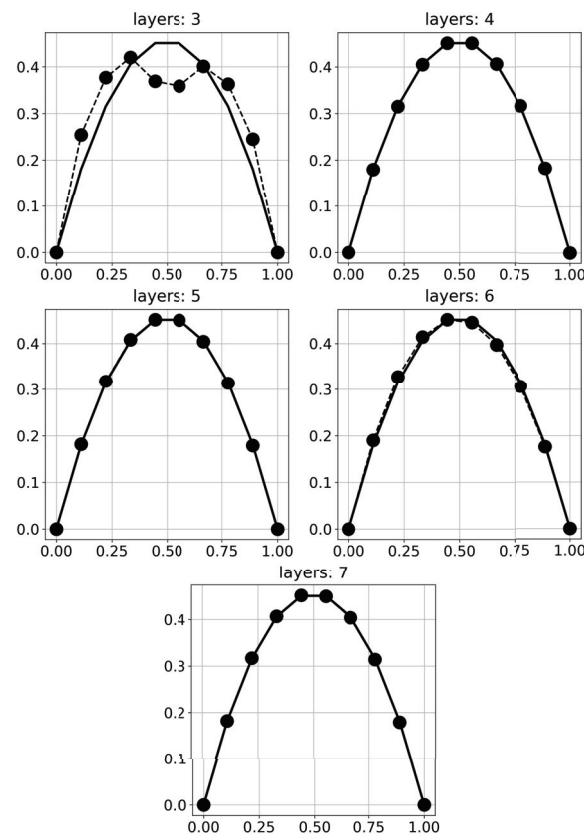


Figure 6. Three-qubit (eight-node) VQLS results (filled circles with dashed lines) for reduced Poisson problem with homogeneous Dirichlet boundary conditions. The classical discrete solution is shown with a solid black line.

For the VQLS results in Figures 3–8, a Qiskit statevector simulator was used so that the full wave function was known, eliminating measurement and sample errors from the

convergence figures. For a quantum calculation, however, measurements are necessary, and sampling errors can affect the classical optimizer convergence. Measurements occurred in the Hadamard tests of the cost calculations. Figure 9 shows the COBYLA cost convergence as the number of shots was increased. It was found that to achieve accurate and smooth convergence, at least 100,000 shots were needed for the 2 qubit VQLS system.

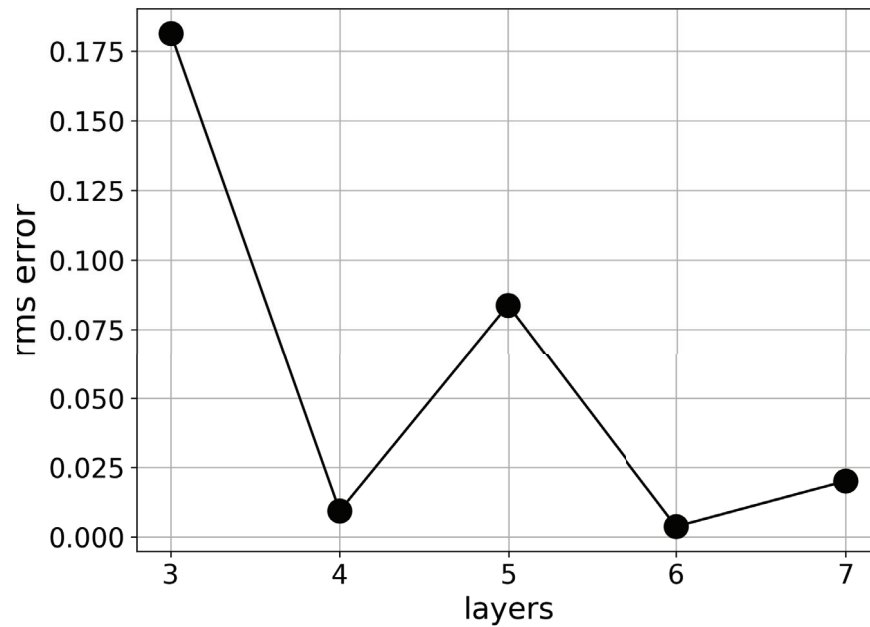


Figure 7. The root mean squared solution error versus the number of layers for the three-qubit VQLS reduced Poisson problem with homogeneous Dirichlet boundary conditions. Here, the errors were averaged over all 10 runs for each layer.

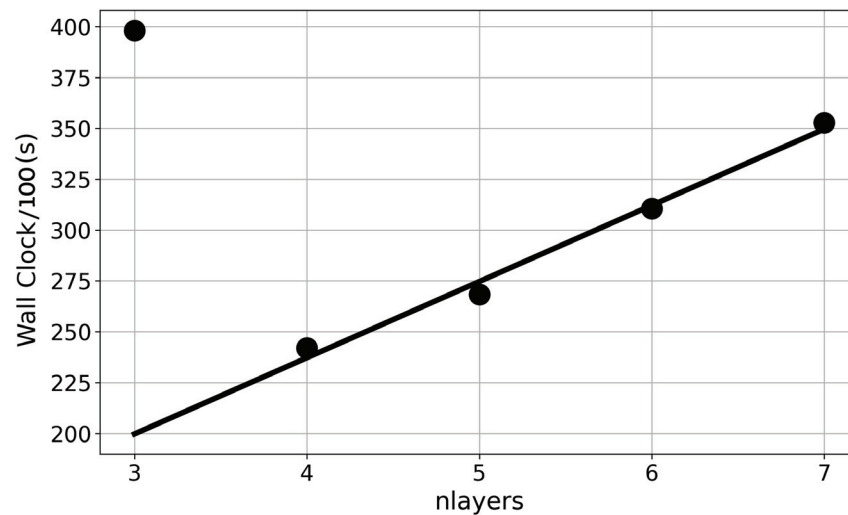


Figure 8. Wall clock time in seconds versus the number of layers for the three-qubit VQLS reduced Poisson problem with homogeneous Dirichlet boundary conditions. The three-layer run does not converge.

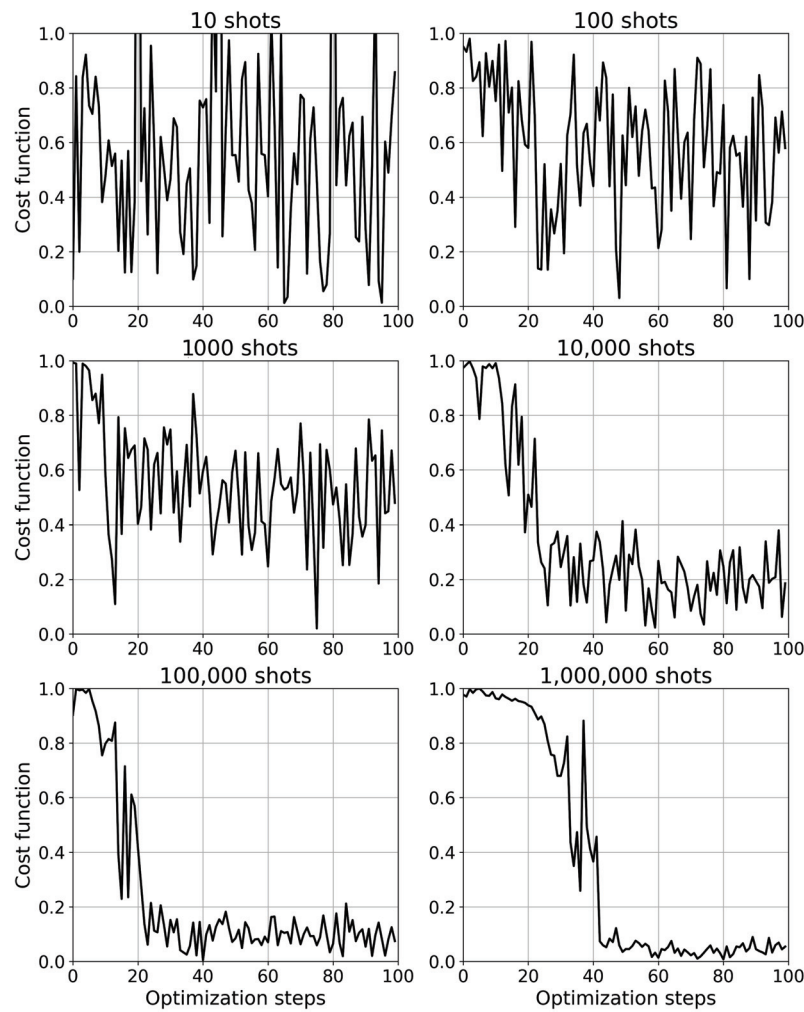


Figure 9. The COBYLA cost convergence for a range of shots in the two-qubit VQLS reduced Poisson problem with homogeneous Dirichlet boundary conditions.

5.1.2. Poisson Case 2: Cubic Solution with Non-Homogeneous B.C.

Next, we consider a non-symmetric cubic Poisson solution with non-homogeneous boundaries, which will further complicate the RHS vector as it modifies f to be

$$f_i = ah^2(6ih - 2) + u_i \quad i = 1, n \tag{11}$$

In this case, $u_i = u_L$ for $i = 1$, and $u_i = u_R$ for $i = n$. Note that normally, the RHS addition to the 1D case would be u_D/h , but both sides are multiplied by h in the discrete matrix solution.

The following cubic manufactured solution is used:

$$u(x) = a(-x^3 + x^2 + x + 1) \tag{12}$$

where $a = 1$ so that $u_L = 1$ and $u_R = 2$. The two-qubit Qiskit wavefunction simulator was used to calculate the discretized, finite-element VQLS results for 2–6 ansatz layers to investigate the layer count sensitivities to accuracy and convergence. For each layer count, five runs were executed, and the mean and standard deviation of the runs were calculated. Once again, all ansatz parameters were initialized randomly between $-\pi \leq \theta_k \leq \pi$, default tolerances of 10^{-4} were used, and the initial change to the variables in the COBYLA optimizer was set to $rhobeg = \pi$. Using these parameters, it was found that only two ansatz layers were needed to fully capture the solution, as can be seen in Figure 10. The best cost

convergence was also seen for two layers, shown in Figure 11. The cost curve variances did not show any obvious trend with the layer count.

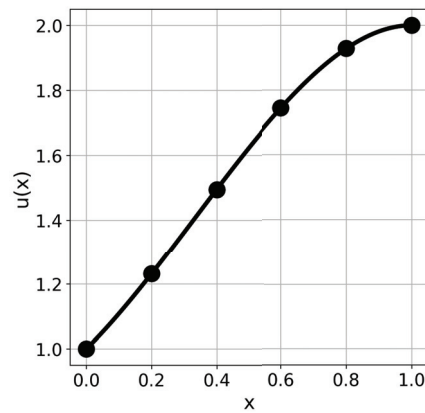


Figure 10. Two-qubit, two-layer solution (filled circles) along with the analytic solution (solid line) of Case 2: the cubic Poisson problem.

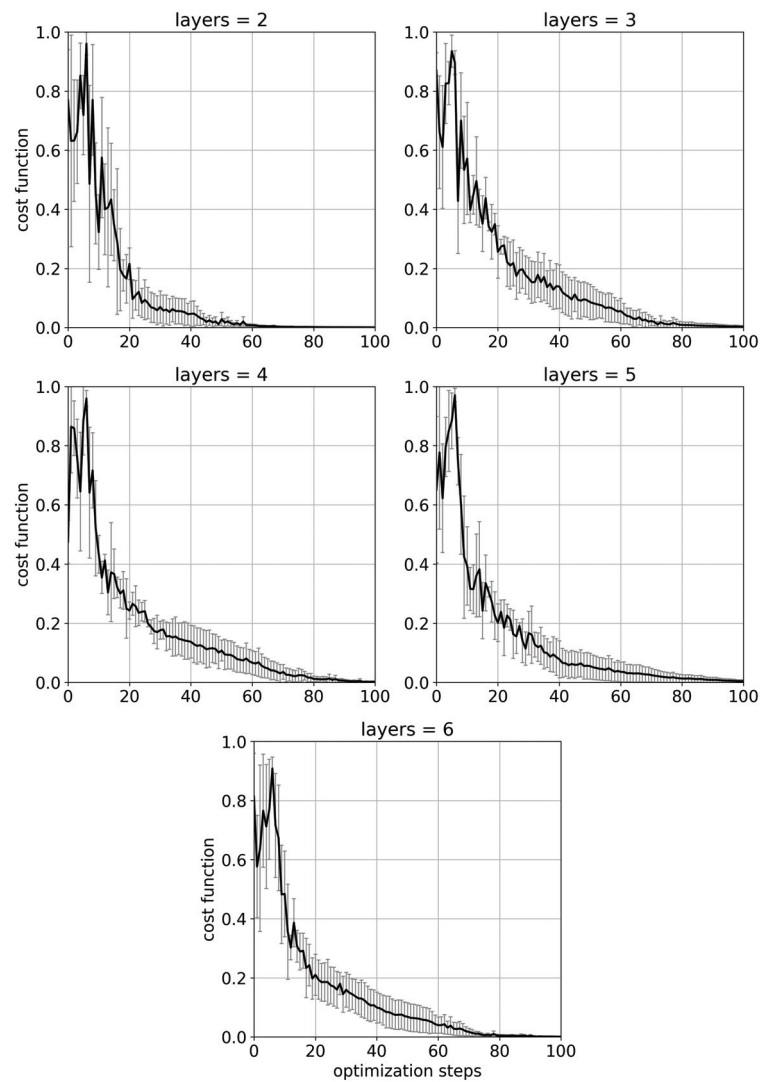


Figure 11. VQLS mean cost versus iteration or optimization count over a range of layers for the cubic Poisson problem. Variances are shown as curve error bars.

5.2. Application 2: The Heat Equation

The Poisson test cases were time-independent and required only one linear solve for the solution. In this section, however, the VQLS results are presented for the 1D time-dependent heat equation

$$\begin{aligned} \partial_t - \partial_{xx}u &= 0 \quad x \in (0,1) \\ u(x_L, t) &= u_L \\ u(x_R, t) &= u_R \\ u(x, 0) &= u_0 \end{aligned} \tag{13}$$

where x_L and x_R are the 1D domain endpoints. The weak form of this equation is

$$\int_{x_L}^{x_R} \partial_t u \phi \, dx + \int_{x_L}^{x_R} u_x \phi_x \, dx = 0 \quad \forall \phi \in H_0^1(0,1) \tag{14}$$

Discretizing in time with uniform time steps Δt and using the backward Euler approximation for the time derivative gives

$$\int_{x_L}^{x_R} u^{k+1} \phi \, dx - \int_{x_L}^{x_R} u^k \phi \, dx + \Delta t \int_{x_L}^{x_R} u_x^{k+1} \phi_x \, dx = 0 \tag{15}$$

where k is the discrete time step index such that $k = 1, nt$ and nt is the total number of time steps. This equation is made to hold for all test functions, giving the following finite-element (FE) backward Euler matrix equation:

$$(\mathbf{M} + \Delta t \mathbf{K}) \vec{u}^{k+1} = \mathbf{M} \vec{u}^k \tag{16}$$

For linear basis functions on a uniform grid of a spacing h , the matrix operators are

$$\mathbf{M} = \begin{bmatrix} \frac{2h}{3} & \frac{1h}{6} & 0 & 0 & \dots \\ \frac{1h}{6} & \frac{2h}{3} & \frac{1h}{6} & 0 & \dots \\ 0 & \frac{1h}{6} & \frac{2h}{3} & \frac{1h}{6} & 0\dots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{17}$$

$$\mathbf{K} = \begin{bmatrix} \frac{2}{h} & -\frac{1}{h} & 0 & 0 & \dots \\ -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & 0 & \dots \\ 0 & -\frac{1}{h} & \frac{2}{h} & -\frac{1}{h} & 0\dots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{18}$$

This equation gives a linear system $\mathbf{A} \vec{x} = \vec{b}$ at each time step such that

$$\mathbf{A} = \begin{bmatrix} \frac{2h}{3} + \frac{2\Delta t}{h} & \frac{1h}{6} + \frac{-1\Delta t}{h} & 0 & 0 & \dots \\ \frac{1h}{6} + \frac{-1\Delta t}{h} & \frac{2h}{3} + \frac{2\Delta t}{h} & \frac{1h}{6} + \frac{-1\Delta t}{h} & 0 & \dots \\ 0 & \frac{1h}{6} + \frac{-1\Delta t}{h} & \frac{2h}{3} + \frac{2\Delta t}{h} & \frac{1h}{6} + \frac{-1\Delta t}{h} & 0\dots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{19}$$

and $\vec{b} = \mathbf{M} \vec{u}^k$.

For verification of the FE-VQLS algorithm, a nonlinear solution was fabricated of the form

$$u(x, t) = \frac{1}{\sqrt{4\pi t}} \exp\left(-\frac{(x - 0.5)^2}{4t}\right) \tag{20}$$

on the domain $[0 \leq x \leq 1] \times [1 \leq t \leq 3]$. A uniform grid was created with $n = 2^m$ internal spatial grid points, $N = n + 2$ total spatial grid points, and $nt = 11$ time points. Figure 12 shows the two-qubit results (dashed lines and open circles) plotted against the analytic

solution (solid line). For these results, three layers were used, and *rhobeg* in the COBYLA method was set to $\pi/100$.

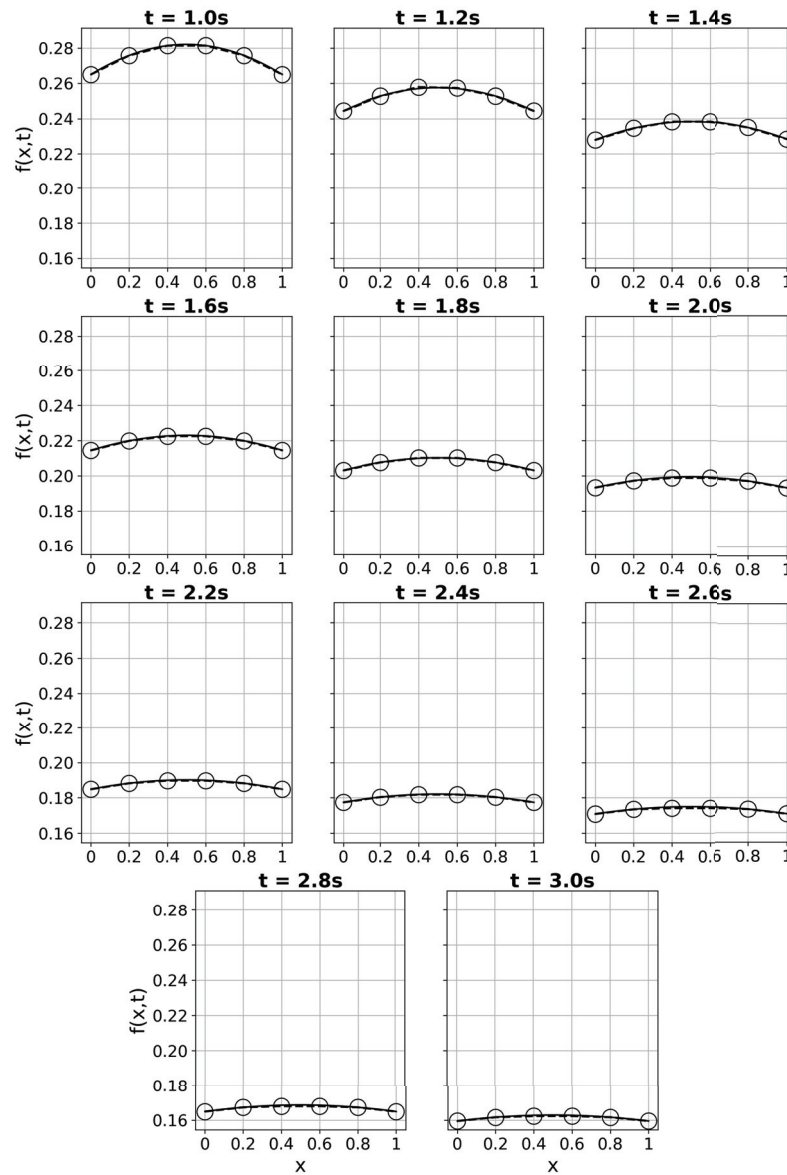


Figure 12. Analytic solution (solid line) versus two-qubit VQLS-based finite element results (dashed line with open circles) for the time-dependent heat equation at each time step.

The results in this figure show excellent agreement between the FE-VQLS and the analytic solution. It should be noted, however, that in order to obtain these results, the FE-VQLS solution had to be scaled appropriately at each time step, since the quantum results were only proportional to the solution. This could be accomplished by using the boundary conditions if they were non-homogeneous, and the system was not solved in a reduced way. However, since the reduced systems were used herein, the non-homogeneous boundaries were not included, and the ratio of the analytic and FE-VQLS solution of the first internal point was used for the scaling.

At each time step, the previous VQLS ansatz parameters were used to initialize the minimization procedure and speed up convergence. Ideally, the number of COBYLA iterations should decrease in time. This was seen for the two-qubit solution, as shown in Figure 13.

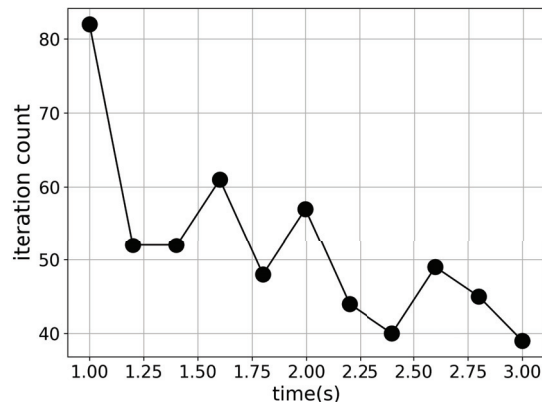


Figure 13. The COBYLA iteration count over time for the two-qubit solution of the heat equation.

5.3. Application 3: The Wave Equation

For the last application, we present the VQLS results for a 1D wave equation of the form

$$\begin{aligned}
 \partial_{tt}u - \partial_{xx}u &= 0 \quad x \in (0,1) \\
 u(x_L, t) &= u_L \\
 u(x_R, t) &= u_R \\
 u(x, 0) &= u_0
 \end{aligned}
 \tag{21}$$

where x_L and x_R are the endpoints of the 1D domain. The weak form of this equation is

$$\int_{x_L}^{x_R} \partial_{tt}u \phi \, dx + \int_{x_L}^{x_R} u_x \phi_x \, dx = 0 \quad \forall \phi \in H_0^1(0,1)
 \tag{22}$$

Discretizing in time with uniform time steps Δt and using a second-order difference approximation for the time derivative gives

$$\int_{x_L}^{x_R} u^{k+1} \phi \, dx - 2 \int_{x_L}^{x_R} u^k \phi \, dx + \int_{x_L}^{x_R} u^{k-1} \phi \, dx + \Delta t^2 \int_{x_L}^{x_R} u_x^{k+1} \phi_x \, dx = 0
 \tag{23}$$

Note here that we have treated the diffusion term implicitly. When applied to all test functions, this yields the matrix equation

$$(\mathbf{M} + \Delta t^2 \mathbf{K}) \vec{u}^{k+1} = \mathbf{M}(2\vec{u}^k + \vec{u}^{k-1})
 \tag{24}$$

where $\mathbf{A} = \mathbf{M} + \Delta t^2 \mathbf{K}$ and $\vec{b} = \mathbf{M}(2\vec{u}^k + \vec{u}^{k-1})$.

To test the VQLS, a non-separable solution of

$$u(x, t) = \sin(x + t)
 \tag{25}$$

was used on the domain $[0 \leq t \leq 1] \times [0 \leq x \leq 1]$. A total of $m = 2$ qubits ($n = 4$ internal points) were used for the matrix-reduced internal solve with three ansatz layers. The time step was set to $\Delta t = 0.1s$. As can be seen in Figure 14, the VQLS results agreed well with the analytic solution, and it is noted that a majority of the differences came from discretization and not from the VQLS procedure. The time-dependent COBYLA iteration count, which essentially represents the time evolution regularity of the ansatz parameters, can be seen in Figure 15. This figure shows a large initial iteration count associated with random sampling and a decrease in iteration count for each linear solve as the solution converged over time.

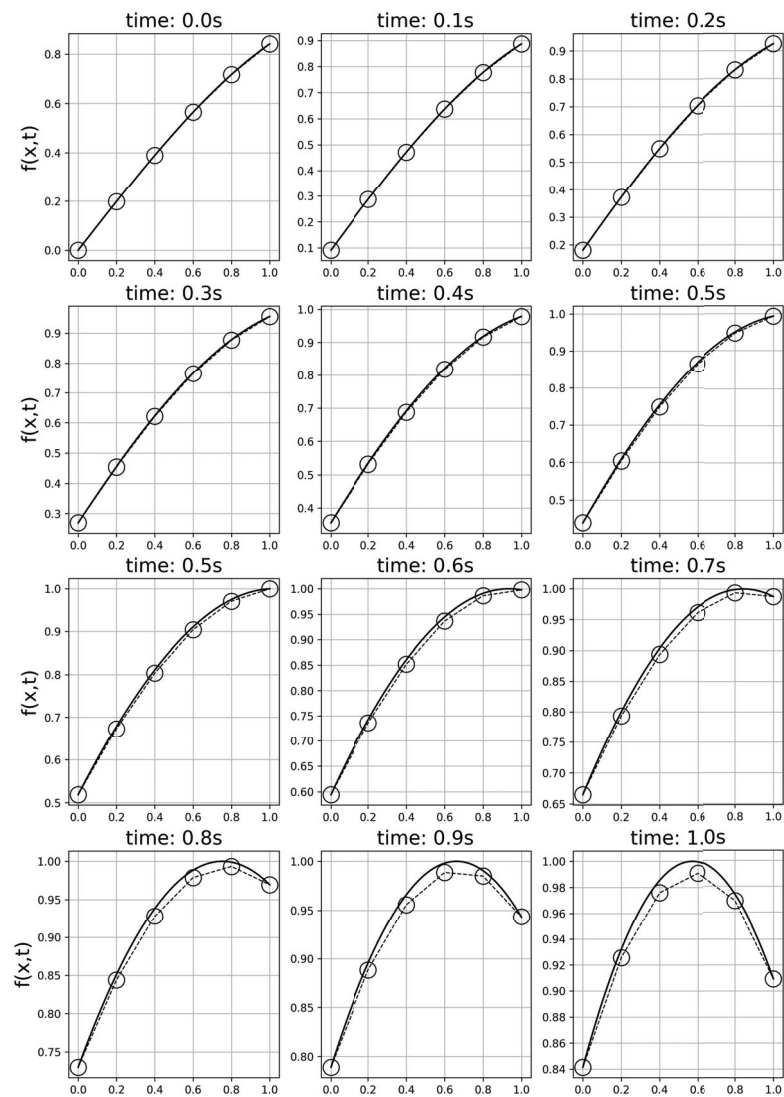


Figure 14. Analytic solution (solid line) versus two-qubit VQLS-based finite element results (dashed line with open circles) for the time-dependent wave equation at each time step.

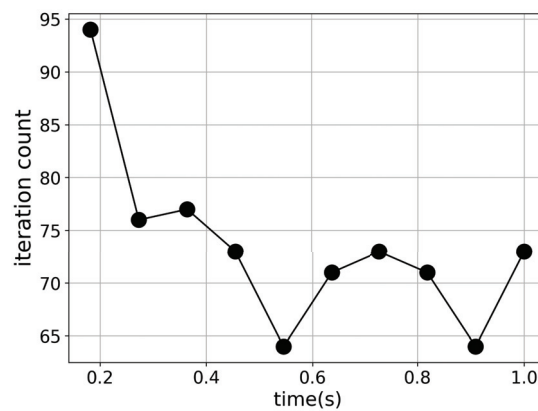


Figure 15. The COBYLA iteration count over time for the two-qubit solution of the wave equation.

6. Discussion

In this study, the variational quantum linear solver recently proposed by Bravo-Prieto et al. [60,61] was used to solve the linear systems obtained from finite-element discretization

of the time-independent Poisson and time-dependent heat and wave equations. Although the results presented focused on these equations, the tools of this effort can generally be used to solve any discretization of a partial differential equation that leads to a matrix solution. The key findings of this effort are that (1) the Qiskit Isometry command can be used to generate wavefunctions for arbitrary vectors, a vital component for solving time-dependent right-hand sides, (2) the quantum/classical hybrid variational solver can be used as a potential “accelerator” for discrete finite-element problems, (3) the large number of sampling shots and N^2 matrix gate Hadamard test evaluation requirements greatly affects qubit scalability and thus the finite element grid resolution, and (4) the minimization iteration count decreases over time as the solution converges, reflecting an ansatz parameter regularity. The latter point is particularly useful for initial value problems, where a set of initial ansatz parameters need only be found once and used thereafter.

Regarding scalability of the VQLS, although it was previously found in [61] that this method was scalable for up to 1024×1024 (10 qubit)-sized systems, that was certainly not the case for the practical linear systems herein, where the matrix and RHS required deeper circuits. Since each term in the stiffness Pauli decomposition requires a Hadamard test against the other terms, this size of the linear combination (circuit depth) directly affects the efficiency of the VQLS algorithm. For the 2 qubit, 6 node systems, the scalability more closely resembled that presented in Bravo-Prieto’s analysis, but the 4 qubit, 18 node systems required 16 Pauli terms in the linear combination, and the VQLS results converged too slowly to be practical.

Future work will include further investigation of (1) new ansatz and optimization options, (2) more efficient methods for creating arbitrary RHS vectors specifically for use in finite-element methods, and (3) the quantum hardware scalability and effect of quantum noise for applications of the FE-VQLS.

Author Contributions: Conceptualization, C.J.T.; Methodology, C.J.T. and M.L.; Validation, M.L.; Investigation, C.J.T., M.L., N.D. and E.E.; Data curation, E.E.; Writing—original draft, C.J.T. and M.L.; Writing—review & editing, C.J.T., M.L. and N.D.; Visualization, C.J.T.; Supervision, C.J.T.; Project administration, C.J.T.; Funding acquisition, C.J.T. All authors have read and agreed to the published version of the manuscript.

Funding: N.D. would like to acknowledge support from the Applied Research Laboratories at the University of Texas at Austin.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Preparation of the Stiffness Matrix

A crucial bottleneck of methods that simulate linear algebra computations with the amplitudes of quantum states is state preparation, which often requires one to initialize a quantum system in a state whose amplitudes reflect the features of the entire dataset. Although efficient methods for state preparation are known for specific cases [66,67], this step easily hides the complexity of the task [68,69].

In order to solve the linear system either directly or variationally on a quantum computer, the stiffness matrix \mathbf{K} must be represented as a linear combination of Hermitian unitary operators, $\mathbf{K} = \sum_i c_i U_i$, representing a system Hamiltonian where U_i represents the unitaries and c_i represents the complex coefficients. Additionally, we assume that the matrix condition number $\kappa < \infty$ and $\|\mathbf{A}\| \leq 1$ and that the \mathbf{A}_i unitaries can be implemented with efficient quantum circuits. Typically, this decomposition consists of a linear combination of Kronecker products of the Identity and Pauli matrices, as these gates are widely used and recognized. These matrices and gates are defined as follows:

$$\mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \mathbf{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \mathbf{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \mathbf{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{A1}$$

To find this linear combination, the tridiagonal matrix can be expressed recursively. Ignoring the diagonal component \mathbf{I}_{2^m} , suppose that we have an expression for the off-diagonal elements in the m qubit case

$$\mathbf{A}_m = \sum_{i=0}^{2^m-2} (|i\rangle \langle i+1| + |i+1\rangle \langle i|) \tag{A2}$$

which gives an expression such as

$$|0\rangle \langle 1| + |1\rangle \langle 2| + \dots + |2^m - 2\rangle \langle 2^m - 1| + h.c. \tag{A3}$$

Now, we write

$$\begin{aligned} \mathbf{I}_2 \otimes \mathbf{A}_m &= (|0\rangle \langle 0| + |1\rangle \langle 1|) \otimes \mathbf{A}_m \tag{A4} \\ &= \sum_{i=0}^{2^m-2} (|i\rangle \langle i+1| + |i+1\rangle \langle i|) + \sum_{i=2^m}^{2^{m+1}-2} (|i\rangle \langle i+1| + |i+1\rangle \langle i|) \\ &= \mathbf{A}_{m+1} - (|2^m - 1\rangle \langle 2^m| + |2^m\rangle \langle 2^m - 1|) \end{aligned}$$

and thus

$$\mathbf{A}_{m+1} = \mathbf{I}_2 \otimes \mathbf{A}_m + (|2^m - 1\rangle \langle 2^m| + |2^m\rangle \langle 2^m - 1|) \tag{A5}$$

In the second line of Equation (A4), tensoring $|1\rangle \langle 1|$ with the second sum puts a “1” bit ahead of every bit string, which shifts every index in the summand by 2^m . Then, by comparison with Equation (A2), this line is simply \mathbf{A}_{m+1} but missing a term connecting the two tri-diagonal block submatrices of $\mathbf{I}_2 \otimes \mathbf{A}_m$.

As an $m = 3$ example, it is first easy to find the off-diagonal solution for $m = 2$, given by

$$\begin{aligned} \mathbf{A}_2 &= \mathbf{I} \otimes \mathbf{X} + \frac{1}{2}(\mathbf{X} \otimes \mathbf{X} + \mathbf{Y} \otimes \mathbf{Y}) \tag{A6} \\ &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{aligned}$$

Using this result, Equation (A5) can be written as

$$\mathbf{A}_3 = \begin{pmatrix} \mathbf{A}_2 & 0 \\ 0 & \mathbf{A}_2 \end{pmatrix} + (|011\rangle \langle 100| + |100\rangle \langle 011|) \tag{A7}$$

Appendix A.1. Implementing the Recursion Using GHZ States

All that is needed now is an operator representing $(|2^m - 1\rangle \langle 2^m| + |2^m\rangle \langle 2^m - 1|)$. This turns out to be closely related to writing an $m + 1$ qubit GHZ state in terms of Pauli operators. The GHZ state $|\psi\rangle = |0\rangle^{\otimes(m+1)} + |1\rangle^{\otimes(m+1)}$ operator $|\psi\rangle \langle \psi|$ has two off-diagonal elements: one in the top left and one in the top right of the corresponding matrix. These elements can be permuting toward the center of the matrix with the operator

$$\begin{aligned} \mathbf{B}_{m+1} &= |0\rangle \langle 2^{m+1} - 1| + |2^{m+1} - 1\rangle \langle 0| \\ &= (|0\rangle \langle 1|)^{\otimes(m+1)} + (|1\rangle \langle 0|)^{\otimes(m+1)} \end{aligned} \tag{A8}$$

Thus, we have

$$\begin{aligned} (\mathbf{X} \otimes \mathbf{I}_{2^m}) \mathbf{B}_{m+1} (\mathbf{X} \otimes \mathbf{I}_{2^m}) &= (\mathbf{X} \otimes \mathbf{I}_{2^m}) ((|0\rangle \langle 1|)^{\otimes(m+1)} + (|1\rangle \langle 0|)^{\otimes(m+1)}) (\mathbf{X} \otimes \mathbf{I}_{2^m}) \\ &= (|10\dots 0\rangle \langle 01\dots 1| + |01\dots 1\rangle \langle 10\dots 0|) \\ &= |2^m\rangle \langle 2^m - 1| + |2^m - 1\rangle \langle 2^m| \end{aligned} \tag{A9}$$

Now, using the results of (GUHNE 2007), the center shift operator can be written as

$$\mathbf{B}_m = (|0\rangle \langle 1|)^{\otimes m} + (|1\rangle \langle 0|)^{\otimes m} \tag{A10}$$

$$= \frac{1}{2^{m-1}} \sum_{t=0}^{\lfloor m/2 \rfloor} (-1)^t \sum_{\pi} \mathbf{S}_{\pi} (\mathbf{X}^{\otimes(m-2t)} \otimes \mathbf{Y}^{\otimes 2t}) \tag{A11}$$

Here, the \mathbf{S}_{π} operator permutes m subsystems according to a permutation $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, m\}$, and the sum runs over all unique permutations π on size m sets. Using this formula along with Equation (A5) gives an analytic Pauli decomposition of the stiffness matrix.

Appendix A.2. Preparation of the $m = 3$ Stiffness Matrix

For the $m = 3$ qubit stiffness matrix (with $2^m = 8$ finite element nodes), Equation (A10) becomes

$$\mathbf{B}_3 = \frac{1}{4} (\mathbf{XXX} - \mathbf{XYY} - \mathbf{YXY} - \mathbf{YYX}) \tag{A12}$$

and therefore

$$(\mathbf{X} \otimes \mathbf{I}_4) \mathbf{B}_3 (\mathbf{X} \otimes \mathbf{I}_4) = \frac{1}{4} (\mathbf{XXX} - \mathbf{XYY} + \mathbf{YXY} + \mathbf{YYX}) \tag{A13}$$

Substituting this into Equation (A5) and adding the diagonal factor $2\mathbf{III}$ gives our final three-qubit, eight-node stiffness matrix Pauli decomposition as follows:

$$\begin{aligned} 2\mathbf{I}_8 - \mathbf{A}_3 &= 2\mathbf{I}_8 - [\mathbf{I}_2 \otimes \mathbf{A}_2 + (\mathbf{X} \otimes \mathbf{I}_4) \mathbf{B}_3 (\mathbf{X} \otimes \mathbf{I}_4)] \\ &= 2\mathbf{III} - [\mathbf{IIX} + \frac{1}{2}(\mathbf{IXX} + \mathbf{IYY}) + \frac{1}{4}(\mathbf{XXX} - \mathbf{XYY} + \mathbf{YXY} + \mathbf{YYX})] \end{aligned} \tag{A14}$$

Appendix A.3. Preparation of a General m Qubit Stiffness Matrix

Generalization of the above procedure for an m qubit stiffness matrix gives the following recursive procedure with $\mathbf{A}_1 := \mathbf{x}$:

$$\begin{aligned} \mathbf{A}_m &= \mathbf{I}_2 \otimes \mathbf{A}_{m-1} + \\ &(\mathbf{X} \otimes \mathbf{I}_{2^{m-1}}) \frac{1}{2^{m-1}} \sum_{t=0}^{\lfloor m/2 \rfloor} (-1)^t \sum_{\pi} \mathbf{S}_{\pi} (\mathbf{X}^{\otimes(m-2t)} \otimes \mathbf{Y}^{\otimes 2t}) + (\mathbf{X} \otimes \mathbf{I}_{2^{m-1}}) \end{aligned} \tag{A15}$$

The final finite-element stiffness matrix is then

$$\mathbf{K}_m = 2\mathbf{I}_{2^m} - \mathbf{A}_m \tag{A16}$$

References

1. Preskill, J. Quantum computing and the entanglement frontier. *arXiv* **2012**. arXiv:1203.5813. Available online: <http://arxiv.org/abs/1203.5813> (accessed on 20 February 2023).
2. Harrow, A.W.; Montanaro, A. Quantum computational supremacy. *Nature* **2017**, *549*, 203–209. [CrossRef] [PubMed]

3. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. Available online: <https://www.nature.com/articles/s41586-019-1666-5> (accessed on 20 February 2023). [CrossRef] [PubMed]
4. Connor, E. The New Light-Based Quantum Computer Jiuzhang Has Achieved Quantum Supremacy. 2020. Available online: <https://www.sciencenews.org/article/new-light-based-quantum-computer-jiuzhang-supremacy> (accessed on 20 February 2023).
5. Head-Marsden, K.; Flick, J.; Ciccarino, C.J.; Narang, P. Quantum information and algorithms for correlated quantum matter. *Chem. Rev.* **2021**, *121*, 3061–3120. [CrossRef]
6. Breuer, H.-P.; Petruccione, F. *The Theory of Open Quantum Systems*; Oxford University Press: Oxford, UK, 2007; ISBN 9780199213900.
7. Clerk, A.A.; Devoret, M.H.; Girvin, S.M.; Marquardt, F.; Schoelkopf, R.J. Introduction to quantum noise, measurement, and amplification. *Rev. Mod. Phys.* **2010**, *82*, 1155–1208. [CrossRef]
8. Lidar, D.A. Lecture notes on the theory of open quantum systems. *arXiv* **2019**, arXiv:1902.00967.
9. Krantz, P.; Kjaergaard, M.; Yan, F.; Orlando, T.P.; Gustavsson, S.; Oliver, W.D. A quantum engineer’s guide to superconducting qubits. *Appl. Phys. Rev.* **2019**, *6*, 021318. [CrossRef]
10. Kandala, A.; Temme, K.; Córcoles, A.D.; Mezzacapo, A.; Chow, J.M.; Gambetta, J.M. Error mitigation extends the computational reach of a noisy quantum processor. *Nature* **2019**, *567*, 491–495. [CrossRef]
11. McArdle, S.; Yuan, X.; Benjamin, S. Error-mitigated digital quantum simulation. *Phys. Rev. Lett.* **2019**, *122*, 180501. [CrossRef]
12. Smart, S.E.; Mazziotti, D.A. Quantum-classical hybrid algorithm using an error-mitigating n -representability condition to compute the mott metal-insulator transition. *Phys. Rev. A* **2019**, *100*, 022517. [CrossRef]
13. Smart, S.E.; Boyn, J.N.; Mazziotti, D.A. Resolving correlated states of benzyne with an error-mitigated contracted quantum eigensolver. *Phys. Rev. A* **2022**, *105*, 022405. [CrossRef]
14. Endo, S.; Cai, Z.; Benjamin, S.C.; Yuan, X. Hybrid quantum-classical algorithms and quantum error mitigation. *J. Phys. Soc. Jpn.* **2021**, *90*, 032001. [CrossRef]
15. Smart, S.E.; Hu, Z.; Kais, S.; Mazziotti, D.A. Relaxation of stationary states on a quantum computer yields a unique spectroscopic fingerprint of the computer’s noise. *Commun. Phys.* **2022**, *5*, 8. [CrossRef]
16. Aleksandrowicz, G.; Alexander, T.; Barkoutsosa, P.; Bello, L.; Ben-Haim, Y.; Bucher, D.; Cabrera-Hernández, F.; Carballo-Franquis, J.; Chen, A.; Chen, C.; et al. Qiskit: An Open-Source Framework for Quantum Computing. 2019. Available online: <https://doi.org/10.5281/zenodo.2562111> (accessed on 10 May 2022)
17. Amazon, Amazon Braket. Available online: <https://aws.amazon.com/braket/> (accessed on 10 May 2022).
18. IBM, Learning Quantum Computation Using Qiskit. Available online: <http://qiskit.org/textbook> (accessed on 1 July 2021).
19. Albornoz, C.; Alonso, G.; Andrenkov, P.A.M.; Asadi, A. Another, Xanadu Quantum Codebook. 2021. Available online: <https://codebook.xanadu.ai> (accessed on 1 July 2022).
20. Qbraid. Qbraid: Cloud-Based ide for Quantum Computing. Available online: <https://qbraid.com> (accessed on 10 July 2022).
21. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [CrossRef]
22. Pudenz, K.L.; Lidar, D.A. Quantum adiabatic machine learning. *Quantum Inf. Process.* **2013**, *12*, 2027–2070. [CrossRef]
23. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **2014**, *10*, 631–633. [CrossRef]
24. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef] [PubMed]
25. Schuld, M.; Sinayskiy, I.; Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **2015**, *56*, 172–185. [CrossRef]
26. Altaisky, M.V.; Zolnikova, N.N.; Kaputkina, N.E.; Krylov, V.A.; Lozovik, Y.E.; Dattani, N.S. Towards a feasible implementation of quantum neural networks using quantum dots. *Appl. Phys. Lett.* **2016**, *108*, 103108. [CrossRef]
27. Dunjko, V.; Taylor, J.M.; Briegel, H.J. Framework for learning agents in quantum environments. *arXiv* **2015**, arXiv:1507.08482.
28. Alvarez-Rodriguez, U.; Lamata, L.; Escandell-Montero, P.; Martín-Guerrero, J.D.; Solano, E. Supervised quantum learning without measurements. *Sci. Rep.* **2017**, *7*, 1–9.
29. Lamata, L. Basic protocols in quantum reinforcement learning with superconducting circuits. *Sci. Rep.* **2017**, *7*, 1–10.
30. Wiebe, N.; Braun, D.; Lloyd, S. Quantum algorithm for data fitting. *Phys. Rev. Lett.* **2012**, *109*, 050505. [CrossRef] [PubMed]
31. Schuld, M.; Sinayskiy, I.; Petruccione, F. Prediction by linear regression on a quantum computer. *Phys. Rev. A* **2016**, *94*, 022342. [CrossRef]
32. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [CrossRef] [PubMed]
33. Berry, D.W.; Childs, A.M.; Kothari, R. Hamiltonian simulation with nearly optimal dependence on all parameters. In Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 17–20 October 2015; pp. 792–809. [CrossRef]
34. Zhao, Z.; Fitzsimons, J.K.; Fitzsimons, J.F. Quantum-assisted Gaussian process regression. *Phys. Rev. A* **2019**, *99*, 052331. [CrossRef]

35. Zheng, Y.; Song, C.; Chen, M.C.; Xia, B.; Liu, W.; Guo, Q.; Zhang, L.; Xu, D.; Deng, H.; Huang, K.; et al. Solving systems of linear equations with a superconducting quantum processor. *Phys. Rev. Lett.* **2017**, *118*, 210504. [CrossRef]
36. Lee, Y.; Joo, J.; Lee, S. Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience. *Sci. Rep.* **2019**, *9*, 4778. [CrossRef] [PubMed]
37. Pan, J.; Cao, Y.; Yao, X.; Li, Z.; Ju, C.; Chen, H.; Peng, X.; Kais, S.; Du, J. Experimental realization of quantum algorithm for solving linear systems of equations. *Phys. Rev. A* **2014**, *89*, 022313. [CrossRef]
38. Cai, X.-D.; Weedbrook, C.; Su, Z.-E.; Chen, M.-C.; Gu, M.; Zhu, M.-J.; Li, L.; Liu, N.; Lu, C.; Pan, J. Experimental quantum computing to solve systems of linear equations. *Phys. Rev. Lett.* **2013**, *110*, 30501. [CrossRef]
39. Barz, S.; Kassal, I.; Ringbauer, M.; Lipp, Y.O.; Dakić, B.; Aspuru-Guzik, A.; Walther, P. A two-qubit photonic quantum processor and its application to solving systems of linear equations. *Sci. Rep.* **2014**, *4*, 6115. [CrossRef] [PubMed]
40. Wen, J.; Kong, X.; Wei, S.; Wang, B.; Xin, T.; Long, G. Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing. *Phys. Rev. A* **2019**, *99*, 012320. [CrossRef]
41. Anschuetz, E.; Olson, J.; Aspuru-Guzik, A.; Cao, Y. Variational quantum factoring. In *Quantum Technology and Optimization Problems*; Feld, S., Linnhoff-Popien, C., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 74–85.
42. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.; Zhou, X.; Love, P.J.; Aspuru-Guzik, A.; O’Brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213. [CrossRef] [PubMed]
43. Cao, Y.; Romero, J.; Olson, J.P.; Degroote, M.; Johnson, P.D.; Kieferova, M.; Kivlichan, I.D.; Menke, T.; Peropadre, B.; Sawaya, N.P.D.; et al. Quantum chemistry in the age of quantum computing. *Chem. Rev.* **2019**, *119*, 10856–10915. [CrossRef]
44. Higgott, O.; Wang, D.; Brierley, S. Variational quantum computation of excited states. *Quantum* **2019**, *3*, 156. [CrossRef]
45. Jones, T.; Endo, S.; McArdle, S.; Yuan, X.; Benjamin, S.C. Variational quantum algorithms for discovering hamiltonian spectra. *Phys. Rev. A* **2019**, *99*, 062304. [CrossRef]
46. Li, Y.; Benjamin, S.C. Efficient variational quantum simulator incorporating active error minimization. *Phys. Rev. X* **2017**, *7*, 021050. [CrossRef]
47. Kokail, C.; Maier, C.; van Bijnen, R.; Brydges, T.; Joshi, M.K.; Jurcevic, P.; Muschik, C.A.; Silvi, P.; Blatt, R.; Roos, C.F.; et al. Self-verifying variational quantum simulation of lattice models. *Nature* **2019**, *569*, 55–360. [CrossRef]
48. Heya, K.; Nakanishi, K.M.; Mitarai, K.; Fujii, K. Subspace variational quantum simulator. *arXiv* **2019**, arXiv:1904.08566.
49. Cirstoiu, C.; Holmes, Z.; Iosue, J.; Cincio, L.; Coles, P.J.; Sornborger, A. Variational fast forwarding for quantum simulation beyond the coherence time. *npj Quantum Inf.* **2020**, *6*, 82. [CrossRef]
50. Yuan, X.; Endo, S.; Zhao, Q.; Li, Y.; Benjamin, S.C. Theory of variational quantum simulation. *Quantum* **2019**, *3*, 191. [CrossRef]
51. Romero, J.; Olson, J.P.; Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol.* **2017**, *2*, 045001. [CrossRef]
52. LaRose, R.; Tikku, A.; O’Neel-Judy, É.; Cincio, L.; Coles, P.J. Variational quantum state diagonalization. *npj Quantum Inf.* **2019**, *5*, 57. [CrossRef]
53. Bravo-Prieto, C.; Garcí a-Martín, D.; Latorre, J.I. Quantum singular value decomposer. *Phys. Rev. A* **2020**, *101*, 062310. [CrossRef]
54. Cerezo, M.; Sharma, K.; Arrasmith, A.; Coles, P.J. Variational quantum state eigensolver. *npj Quantum Inf.* **2022**, *8*, 113. [CrossRef]
55. Khatri, S.; LaRose, R.; Poremba, A.; Cincio, L.; Sornborger, A.T.; Coles, P.J. Quantum-assisted quantum compiling. *Quantum* **2019**, *3*, 140. [CrossRef]
56. Jones, T.; Benjamin, S.C. Robust quantum compilation and circuit optimisation via energy minimisation. *Quantum* **2022**, *6*, 628. [CrossRef]
57. Arrasmith, A.; Cincio, L.; Sornborger, A.T.; Zurek, W.H.; Coles, P.J. Variational consistent histories as a hybrid algorithm for quantum foundations. *Nat. Commun.* **2019**, *10*, 3438. [CrossRef]
58. Cerezo, M.; Poremba, A.; Cincio, L.; Coles, P. J. Variational quantum fidelity estimation. *Quantum* **2020**, *4*, 248. [CrossRef]
59. Koczor, B.; Endo, S.; Jones, T.; Matsuzaki, Y.; Benjamin, S. Variational-state quantum metrology. *New J. Phys.* **2020**, *22*, 083038. [CrossRef]
60. Bravo-Prieto, C.; LaRose, R.; Cerezo, M.; Subasi, Y.; Cincio, L.; Coles, P.J. Variational Quantum Linear Solver. *arXiv* **2019**, arXiv:1909.05820. [CrossRef]
61. Bravo-Prieto, C.; LaRose, R.; Cerezo, M.; Subaşı, Y.; Cincio, L.; Coles, P.J. Variational quantum linear solver: A hybrid algorithm for linear systems. *Bull. Am. Phys. Soc.* **2020**, arXiv:1909.05820v2. [CrossRef]
62. Cincio, L.; Subaşı, Y.; Sornborger, A.T.; Coles, P.J. Learning the quantum algorithm for state overlap. *New J. Phys.* **2018**, *20*, 13022. [CrossRef]
63. Pesce, R.M.N.; Stevenson, P.D. H2zixy: Pauli spin matrix decomposition of real symmetric matrices. *arXiv* **2021**, arXiv:2111.00627.
64. Pellow-Jarman, A.; Sinayskiy, I.; Pillay, A.; Petruccione, F. A comparison of various classical optimizers for a variational quantum linear solver. *Quantum Inf. Process.* **2021**, *20*, 202. [CrossRef]
65. Hughes, T.J. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*; Courier Corporation: Chelmsford, MA, USA, 2012.
66. Soklakov, A.N.; Schack, R. Efficient state preparation for a register of quantum bits. *Phys. Rev. A* **2006**, *73*, 012307. [CrossRef]

67. Giovannetti, V.; Lloyd, S.; Maccone, L. Quantum Random Access Memory. *Phys. Rev. Lett.* **2008**, *100*, 160501. [CrossRef]
68. Aaronson, S. Read the fine print. *Nat. Phys.* **2015**, *11*, 291–293. [CrossRef]
69. Bang, J.; Dutta, A.; Lee, S.W.; Kim, J. Optimal usage of quantum random access memory in quantum machine learning. *Phys. Rev. A* **2019**, *99*, 012326. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

A More General Quantum Credit Risk Analysis Framework

Emanuele Dri ^{1,*}, Antonello Aita ², Edoardo Giusto ¹, Davide Ricossa ³, Davide Corbelleto ³,
Bartolomeo Montrucchio ¹ and Roberto Ugoccioni ³

¹ Dipartimento di Automatica e Informatica (DAUIN), Politecnico di Torino, 10129 Torino, Italy

² IBM Italia, 20090 Milano, Italy

³ Intesa Sanpaolo, 10121 Torino, Italy

* Correspondence: emanuele.dri@polito.it

Abstract: Credit risk analysis (CRA) quantum algorithms aim at providing a quadratic speedup over classical analogous methods. Despite this, experts in the business domain have identified significant limitations in the existing approaches. Thus, we proposed a new variant of the CRA quantum algorithm to address these limitations. In particular, we improved the risk model for each asset in a portfolio by enabling it to consider multiple systemic risk factors, resulting in a more realistic and complex model for each asset's default probability. Additionally, we increased the flexibility of the loss-given-default input by removing the constraint of using only integer values, enabling the use of real data from the financial sector to establish fair benchmarking protocols. Furthermore, all proposed enhancements were tested both through classical simulation of quantum hardware and, for this new version of our work, also using QPUs from IBM Quantum Experience in order to provide a baseline for future research. Our proposed variant of the CRA quantum algorithm addresses the significant limitations of the current approach and highlights an increased cost in terms of circuit depth and width. In addition, it provides a path to a substantially more realistic software solution. Indeed, as quantum technology progresses, the proposed improvements will enable meaningful scales and useful results for the financial sector.

Keywords: quantum computing; algorithms; scalability; credit risk analysis; quantum finance

Citation: Dri, E.; Aita, A.; Giusto, E.; Ricossa, D.; Corbelleto, D.; Montrucchio, B.; Ugoccioni, R. A More General Quantum Credit Risk Analysis Framework. *Entropy* **2023**, *25*, 593. <https://doi.org/10.3390/e25040593>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 15 March 2023

Revised: 26 March 2023

Accepted: 29 March 2023

Published: 31 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

This paper is intended as an extension of previous work [1]. In particular, this extended version includes new data, analysis, and theoretical developments not present in the original paper and that derive from recently acquired access to 7- and 27-qubit QPUs. The authors believe that these additions provide significant new insights and perspectives regarding the enhanced algorithm and its practical implementation.

Quantum Finance and Credit Risk Analysis

The field of quantum finance aims to use quantum computing to solve a variety of computational problems in finance more effectively than classical methods [2,3]. In recent years, researchers have focused on achieving a quantum advantage in credit risk analysis (CRA) [1,4]. CRA is a crucial risk management tool that assesses the risk of loss from a debtor's insolvency [5]. Classically, Monte Carlo methods are commonly used in the field to estimate economic capital, which is the amount of capital needed to ensure a company remains solvent based on its risk profile. Essentially, these estimation techniques depend on obtaining numerical results through repetitive random sampling [6]. A practical example of their utilization is the computation of the value at risk (VaR), a statistic that quantifies how much a set of investments might lose (with a given probability) over a defined time-frame [7]. This metric is broadly used for the assessment of EC, but in most cases, no closed-form solution currently exists for computing it [8].

However, Monte Carlo simulations are computationally expensive due to the rare-event simulation problems inherent in credit risk evaluation [9]. Additionally, Monte Carlo simulations can only generate pseudo-random variables, and the quality of the simulation can be compromised by the appearance of patterns [10].

To overcome these limitations, researchers have explored new methods, such as those based on quantum computing, which can naturally generate true random samples due to the probabilistic nature of qubits [11]. Moreover, quantum amplitude estimation (QAE) has shown promise in estimating the value at risk and offers a quadratic speedup over classical Monte Carlo methods [4,12].

However, the existing quantum algorithm for CRA [4] is based on the Basel II framework, built on an ASFR (asymptotic single-factor risk) model [13], which assumes a borrower will default if the value of its assets falls below the value of its liabilities [13]. A visual representation is provided in Figure 1. While this model is useful, it is not optimal, especially for complex credit risk portfolios. In fact, though it helps to reserve an EC amount that suits every default scenario, it is intended to be a standard tool for CRA and therefore it is deliberately conservative [14]. Large financial institutions use custom models that consider several risk factors instead of just one since this refinement allows them to reserve a more precise amount to cover potential losses [15]. To address this aspect, we proposed modifications to the existing quantum algorithm to handle increased complexity in the assets’ default model, while preserving the advantages of quantum computing in terms of needed (quantum) samples.

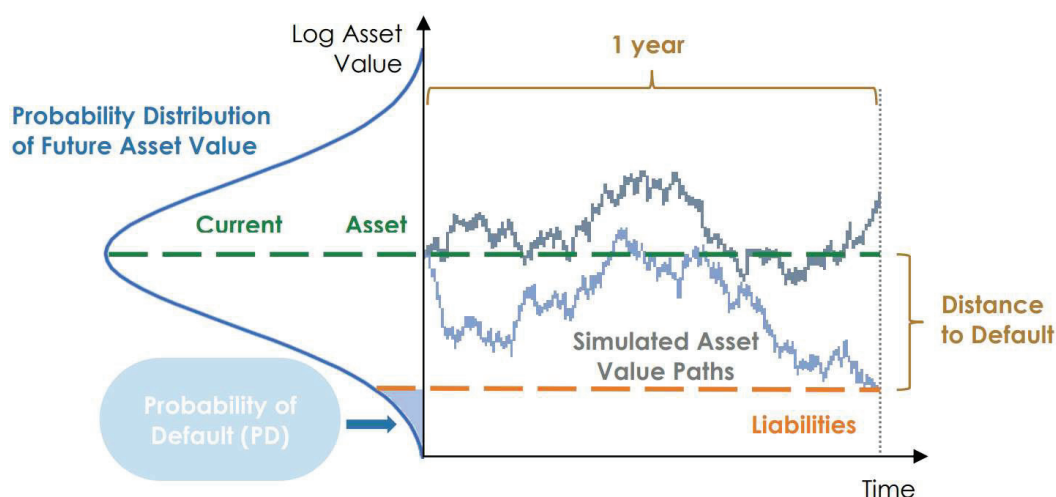


Figure 1. Asset-based default model.

Additionally, we presented a solution to encode non-integer values for the loss-given-default input parameter to use real-world data and provide a fair comparison with traditional benchmarks.

We are now able to provide experimental results for the enhanced version of the original quantum algorithm, not only through classical simulation of quantum hardware but also from cloud access to IBM QPUs with 7 and 27 superconducting qubits.

In the following sections, we first introduce the use of quantum amplitude estimation for CRA. We then present the proposed modifications to the existing algorithm to address outstanding issues, including the Basel II model’s limitations and the encoding of non-integer values. Lastly, we present the new results of simulation experiments and of executions on real quantum devices obtained by running the experiments on IBM devices from the *Researchers* program [16] and from the *pay-as-you-go* service [17].

2. Methods

As described in [1], credit risk can be evaluated through three primary measures: the probability of default (PD), the loss given default (LGD), and the economic capital (E_{cap}). The PD represents the likelihood that the debtor will become insolvent, while the LGD is the estimated loss following the insolvency of the counterparty. The expected loss is another commonly used risk measure, which depends on both the PD and LGD, as an increase in either quantity results in a higher expected loss. Multiplying the PD and LGD values gives the expected loss for each exposure. This measure is additive, so the expected loss for a portfolio of n assets is the sum of each exposure's expected loss.

$$\mathbb{E}[\mathcal{L}] = \sum_{k=1}^n PD_k \cdot LGD_k \tag{1}$$

E_{cap} is the third measure used to assess credit risk. It is defined as the amount of equity that a financial institution will maintain to manage the risk of credit losses in its portfolio. The economic capital, which is the VaR (quantile of losses at a certain confidence level α) minus the total expected loss, is determined based on the distribution of losses.

$$E_{cap} = VaR_{\alpha} - \mathbb{E}[\mathcal{L}] \tag{2}$$

The expected loss is already taken into account in financial reports for financial institutions, so it is subtracted from the VaR and thus not factored into the EC. Therefore, the economic capital is used to measure unexpected or extreme values of losses rather than average losses.

2.1. SOTA Quantum Credit Risk Analysis

The quantum amplitude estimation (QAE) algorithm [18] provides a quadratic speedup compared to classical Monte Carlo methods [12]. QAE has been utilized to determine VaR in prior research [4]. A variant of QAE called iterative QAE (IQAE) has recently been proposed as well [19]. This variant reduces the number of required qubits and gates while maintaining the quadratic speedup (up to a logarithmic factor) over classical methods.

In order to exploit the speedup guaranteed by the QAE algorithm, the problem under consideration has to be mapped to a Hermitian operator \mathcal{A} acting on $n + 1$ qubits. This \mathcal{A} operator is constructed in the following way:

$$\mathcal{A}|0\rangle_{n+1} = \sqrt{1-a}|\psi_0\rangle_n|0\rangle + \sqrt{a}|\psi_1\rangle_n|1\rangle \tag{3}$$

where $a \in [0, 1]$ represents the probability of measuring the last qubit in the quantum state $|1\rangle$. The last qubit is in fact the one identifying the property of interest. The QAE algorithm permits us to effectively estimate the value of a . The reader can refer to [4,12,18] for additional information on QAE.

In prior research [4], QAE has been utilized to determine the cumulative distribution function (CDF) of the total loss \mathcal{L} and construct a Hermitian operator \mathcal{A} such that $a = \mathbb{P}[\mathcal{L} \leq x]$ for a given $x \geq 0$. Then, a bisection search is applied in order to locate the smallest $x_{\alpha} \geq 0$ such that $\mathbb{P}[\mathcal{L} \leq x_{\alpha}] \geq \alpha$, implying that $x_{\alpha} = VaR_{\alpha}$. Thus, the aim when calculating VaR_{α} is to identify the minimum threshold for which the estimated probability is greater than or equal to α .

To map the CDF of the total loss to a Hermitian operator \mathcal{A} , three operators are usually required:

- \mathcal{U} , which loads the domain-dependent uncertainty model.
- \mathcal{S} , which computes the total loss over n_S qubits.
- \mathcal{C} , which flips a target qubit if the total loss is equal to or lower than a certain threshold x .

Operator \mathcal{C} is used to execute the bisection search needed to compute the VaR.

For what concerns the default model, the framework implemented in [4] is similar to the Basel II internal-ratings-based (IRB) method known as the Gaussian conditional independence model [20,21]. In compliance with this model, all losses can be represented by $L_k = LGD_k \cdot X_k$, where $X_k \in \{0, 1\}$ is a related Bernoulli random variable. The probability for asset k to default is the probability that $X_k = 1$. According to the Basel II approach, assuming a latent random variable \mathcal{Z} (also referred to as a systemic risk factor) with a realization z , the Bernoulli random variables $X_k \mid \mathcal{Z} = z$ are considered independent. However, their default probabilities PD_k depend on z while \mathcal{Z} adheres to a standard normal distribution. The default probability $PD_k(z)$ is given by

$$PD_k(z) = F\left(\frac{F^{-1}(p_k^0) - \sqrt{\rho_k}z}{\sqrt{1 - \rho_k}}\right) \tag{4}$$

where p_k^0 represents the default probability for $z = 0$, F represents the CDF of the standard normal distribution, and $\rho_k \in [0, 1)$ determines the sensitivity of X_k to \mathcal{Z} [4].

2.2. Multiple Risk Factors

In the original single-factor model presented in [4], the default probabilities of the counterparts are encoded in a qubit register on which one Y -rotation $R_Y(\theta_{p_0}^k)$ per qubit is applied with angle $\theta_{p_0}^k = 2 \arcsin\left(\sqrt{p_k^0}\right)$. These rotations comprise the loading operator \mathcal{U} introduced in [4].

The original implementation also makes use of a register with n_Z qubits. This register encodes a truncated and discretized version of \mathcal{Z} using the method proposed in [22]. In this way, we include systemic risk in the quantum uncertainty model, using the realization of \mathcal{Z} to prepare the qubits representing the counterparties through controlled rotations with angles $\theta_p^k(z) = 2 \arcsin\left(\sqrt{PD_k(z)}\right)$.

As stated in Section 1, the single-factor model, as implemented in the Basel II framework, is intentionally designed to be conservative [14]. However, it has been recognized that this model has limitations, prompting large financial institutions to seek alternatives to measure risk more accurately. The most common approach extends the single-factor model and employs multiple systemic risk factors. This extension aims to directly attribute default correlations and probabilities to the risk factors, thereby capturing a more realistic depiction of credit risk. The extended model provides a significant advantage by utilizing real-time information about the credit cycle, which enhances the accuracy of the underlying credit risk assessments. As a result, this approach represents a fundamental departure from the uncorrelated defaults inherent in the base model, and it can capture the linkages between economic and financial market factors. In light of these considerations, it is evident that the extended model constitutes a valuable tool for improving risk management practices in the financial sector, particularly for assessing the credit risk of large and complex financial institutions. Furthermore, this approach can reduce uncertainties about the parameters needed for portfolio models' value-at-risk calculations [15], which is particularly critical for risk-sensitive regulatory capital requirements. Thus, this extended model is widely adopted as a tool for more accurate risk measurement in the financial sector.

In the proposed implementation of the model described above, each risk factor \mathcal{Z}_i still adheres to a standard normal distribution and presents a weight α_i computed by financial institutions, taking into account possible correlation effects among the different factors considered [23]. Therefore, the default probability depends on a random variable \mathcal{Y} , which is a linear combination of the R risk factors considered.

$$\mathcal{Y} = \sum_{i=1}^R \alpha_i \mathcal{Z}_i \tag{5}$$

From a practical standpoint, this model comprises multiple latent random variables whose realizations, when appropriately combined, determine the probability of default for each asset.

$$PD_k(z) = F\left(\frac{F^{-1}(p_k^0) - \sqrt{\rho_k} \sum_{i=1}^R \alpha_i z_i}{\sqrt{1 - \rho_k}}\right) \tag{6}$$

However, with increased complexity comes the need for alternative approaches to implement a quantum multi-factor version of the canonical uncertainty model. To address this challenge, we proposed two alternatives, each with unique advantages and limitations.

The *first alternative* for encoding systemic risk factors in quantum financial applications involves the use of multiple quantum registers. In this approach, each systemic risk factor, denoted as Z_i , is assigned its own register, with the values in these registers corresponding to multiple normal standard distributions. The realization of these distributions controls one linear rotation for each asset in the portfolio, with each rotation being weighted by the corresponding α_i using the slope of the rotation. This produces a set of rotations that are used to encode the default probability of each asset in the portfolio, as in the original algorithm. The circuit corresponding to this process is illustrated in Figure 2.

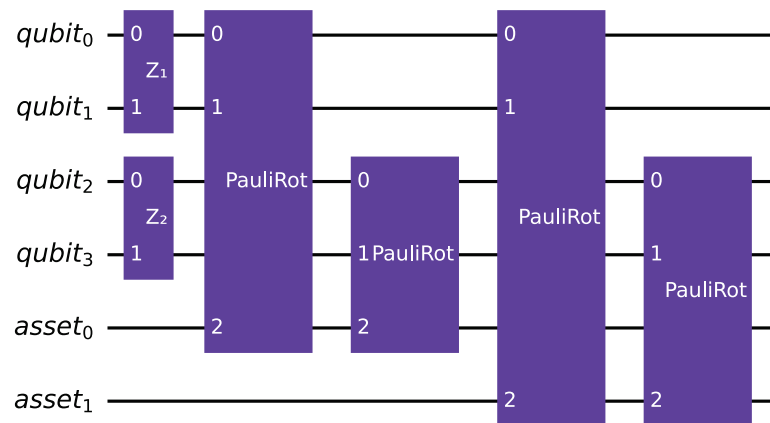


Figure 2. An instance of the multi-factor version of the quantum circuit that encodes the canonical uncertainty model, using multiple rotations. The example involves $K = 2$ assets and $n_z = 2$, which means that two qubits are used to encode each normal standard distribution. The example also takes into account two risk factors ($R = 2$).

While this alternative only requires a limited number of extra qubits to represent the various risk factors, it does entail a significant increase in the number of gates required to implement the encoding process. Specifically, each additional risk factor considered will necessitate K new controlled linear rotations. This increased number of gates is because each risk factor requires a separate register and rotation, which in turn requires additional controlled operations in the circuit.

The *second alternative* for implementing a quantum multi-factor version of the canonical uncertainty model involves a single quantum register encoding a random variable \mathcal{N} that follows a multivariate normal distribution. A sum register is employed to add up the values taken by the normal distributions, corresponding to the marginal distributions of the multivariate distribution, with each marginal distribution representing a risk factor. The resulting value is used to perform a single linear rotation for each asset, to encode its default probability in the target qubit. The circuit corresponding to this second process is illustrated in Figure 3.

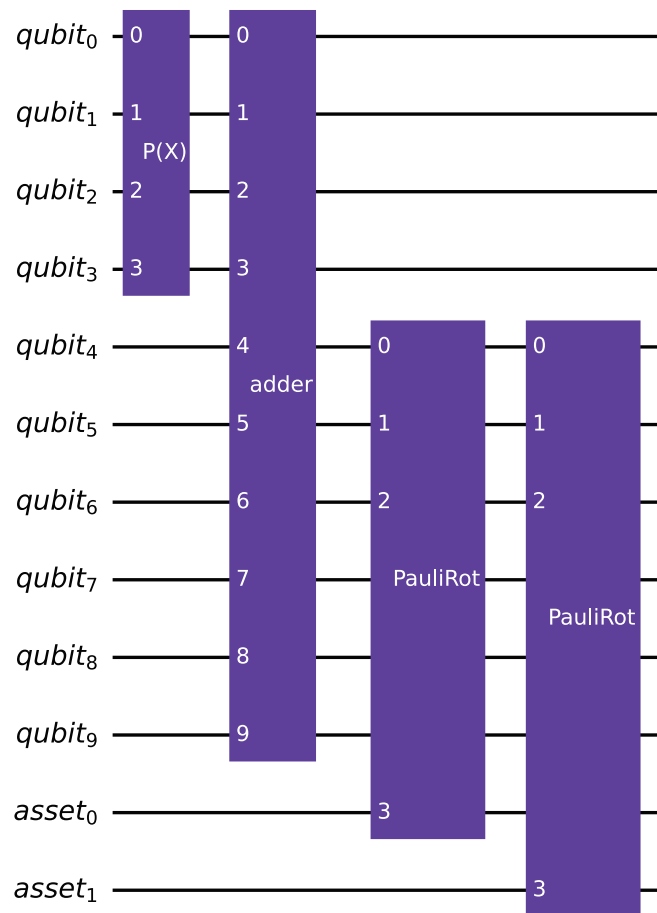


Figure 3. An instance of the multi-factor version of the quantum circuit that encodes the canonical uncertainty model. It has identical parameters to the circuit illustrated in Figure 2 but uses only one rotation per asset.

However, since a single rotation is performed per asset, accounting for all the risk factors, the multivariate normal distribution, in this case, is non-standard. This is because it is not possible to encode the weights in the slope of the rotations. Instead, the covariance matrix of the distribution is used to encode the α weights. This approach has a significant drawback, as it requires the same α vector for all the assets.

Despite this limitation, this approach reduces the circuit depth compared to the previous one, as only one rotation is required for encoding the asset’s default probability. However, the method incurs overhead in terms of the required qubits due to the presence of an extra sum register. Nevertheless, this overhead becomes negligible in a scenario with portfolios composed of thousands of assets.

For a more detailed evaluation of the qubits and gates required by the various approaches, we refer the reader to Section 4.1.

Both multi-factor approaches provide an advantage over the Basel II single-factor model by using actual information about the point in time of the credit cycle. Uncertainties about the parameters needed for value-at-risk calculations in portfolio models can thus be reduced.

2.3. Arbitrary LGD

One limitation of current implementations of quantum credit risk algorithms is the constraint on LGD parameters, which can only assume integer values due to the use of

a weighted sum register in the operator \mathcal{S} that computes the total loss. The function \mathcal{S} operates as follows:

$$\mathcal{S} : |x_1, \dots, x_K\rangle_K |0\rangle_{n_S} \mapsto |x_1, \dots, x_K\rangle_K |LGD_1 x_1 + \dots + LGD_K x_K\rangle_{n_S} \tag{7}$$

Here, $x_k \in \{0, 1\}$ denotes the possible realizations of X_k , while the loss given default of each asset is implemented using the weights of the *WeightedAdder* register provided by Qiskit [24,25], which are limited to integer values. We also require $n_S = \lfloor \log_2(LGD_1 + \dots + LGD_K) \rfloor + 1$ qubits to represent all possible values of the sum of losses given default in the second register.

This constraint is particularly limiting considering the small number of currently available qubits. For instance, using three assets with LGD values in the order of 10^5 , around 20 qubits would be needed just for the sum register. To allow for more realistic input data, we proposed an alternative version of the algorithm that eliminates the \mathcal{S} operator. In particular, we modified the \mathcal{C} operator using a circuit that implements a piecewise linear function $\hat{f} : 0, \dots, 2^n - 1 \rightarrow [0, 1]$ on qubit amplitudes [4,26,27]. The modified \mathcal{C} operator is defined as:

$$F|x\rangle|0\rangle = \sqrt{1 - \hat{f}(x)}|x\rangle|0\rangle + \sqrt{\hat{f}(x)}|x\rangle|1\rangle \tag{8}$$

where $|x\rangle$ is an n -qubit state. This new approach allows the operator to directly read defaulted qubits from the X-register and associate them with the corresponding total loss. The objective qubit is flipped only if the total loss is less than or equal to the given level x set by the current bisection search step. Essentially, the operator reads the X-register as a binary number, and then the specific total loss associated with that binary number is compared with x to determine if the objective qubit should be flipped.

In the next section, we apply this improved algorithm to an illustrative example using both classical simulations of quantum hardware and real quantum computers.

3. Results

In this section, we present the results of experiments conducted on toy models that illustrate the proposed improvements.

The chosen numeric values for the LGD parameters demonstrate the increased flexibility allowed by our approach compared to the previous one. Each latent random variable \mathcal{Z}_k was modeled using two qubits. No qubits were needed for the sum register as it is not required for the proposed algorithm.

3.1. Noiseless Simulation

The noiseless experiment utilized the multiple-rotations scheme with $K = 2$ assets and two systemic risk factors ($R = 2$). Table 1 provides the values of the parameters used in the experiments. To simulate the experiment, the circuit for \mathcal{A} was supplied to the iterative amplitude estimation sub-routine implemented in Qiskit [24]. We performed the bisection search using the result to find VaR_α , with $\alpha = 0.95$. For the iterative quantum amplitude estimation, we set a target precision of $\epsilon = 0.002$ and a 99% target confidence interval. This resulted in an average of approximately 50,000 quantum samples used by the IQAE algorithm to achieve the desired precision and confidence. The entire experiment required 9 qubits that were first simulated (without noise simulation) on classical computers using the simulation back-ends provided by Qiskit [24]. The resulting loss distribution is displayed in Figure 4. Additionally, Figure 5 shows the corresponding CDF and the target level for the value at risk.

Table 1. Problem parameters for the two-asset example (noiseless simulation).

Asset Number k	Loss Given Default LGD_k	Default Prob. p_k^0	Sensitivity ρ_k	Risk Factor Weights $(\alpha_1, \alpha_2)_k$
1	1000.5	0.15	0.1	0.35, 0.2
2	2000.5	0.25	0.05	0.1, 0.25

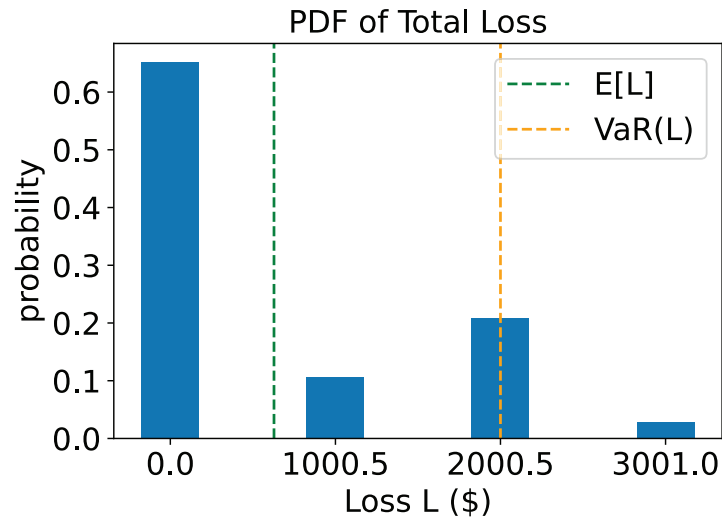


Figure 4. Noiseless simulation: probability distribution function of total loss. The green dashed line shows the expected loss while the orange dashed line shows the value at risk.

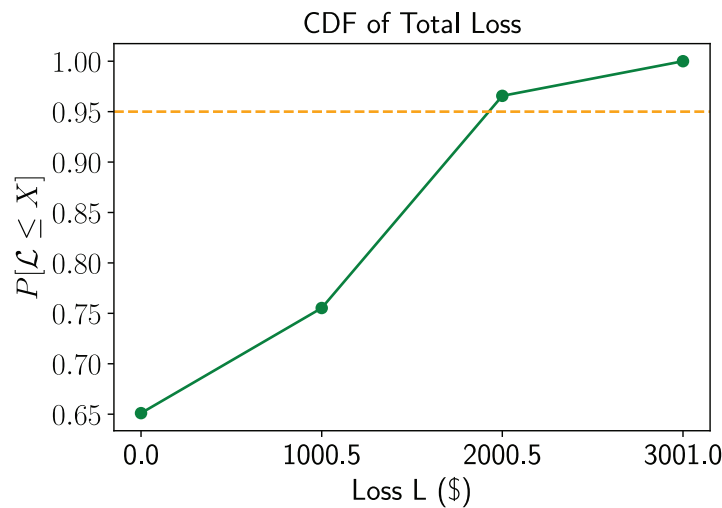


Figure 5. Noiseless simulation: CDF of total loss \mathcal{L} in green and target level of 95 percent in orange.

3.2. Real Hardware and Noisy Simulations

The experiments described in this section aimed to approach as closely as possible the practical implementation of the algorithm on actual quantum hardware. The results can provide a reference point for future research works that may want to evaluate technological improvements. For this reason, the data used were specifically generated by domain experts who took into account what realistic and reliable values for input measurements could be. All the relevant data are available in a public repository [28].

For these experiments, we tested various configurations of the multiple-rotations model on several quantum processors. For each configuration, the experiment was executed both on the actual machine and classically via simulation of the machine’s noise model. This

was done to understand the effect of the QPU's quantum volume and of its topology on the output, as well as to validate the noise models through simulation. As expected, the circuit sizes (especially in terms of depth) make the effects of decoherence on the results evident for all configurations, which hinders the proper extrapolation of the target measurement. Nevertheless, these findings allow for a baseline for future works focused on providing solutions in this regard, by reducing the circuit depth or exploiting more stable qubits.

The configurations considered involved 2 to 4 assets and 1 to 3 systemic risk factors. The required number of qubits varied from a minimum of 7 to a maximum of 13. The quantum processors used were as follows:

- **ibm_perth** and **ibm_lagos**, each with 7 qubits and a quantum volume of 32.
- **ibm_canberra** and **ibm_algiers**, each with 27 qubits and quantum volumes of 32 and 128, respectively.

The topology for these architectures is shown in Appendix A.

The aggregate results of the simulated experiments are displayed in Figure 6, while those related to real hardware executions are shown in Figure 7. As mentioned earlier, the depth of the circuit does not allow for the extrapolation of the correct expected value from the computation. For this reason, we deemed it essential to study the effect of noise on the circuit and observe its behavior. To investigate this aspect, we plotted the ratio between the estimated expected loss and the maximum possible loss (which coincides with the sum of the LGDs of the various counterparties) on the x-axis. It should be noted that we used the expected loss as the output metric, estimated directly using the objective qubit.

The complete and non-aggregate results, as well as the code used to generate them, are available in a public repository [28].

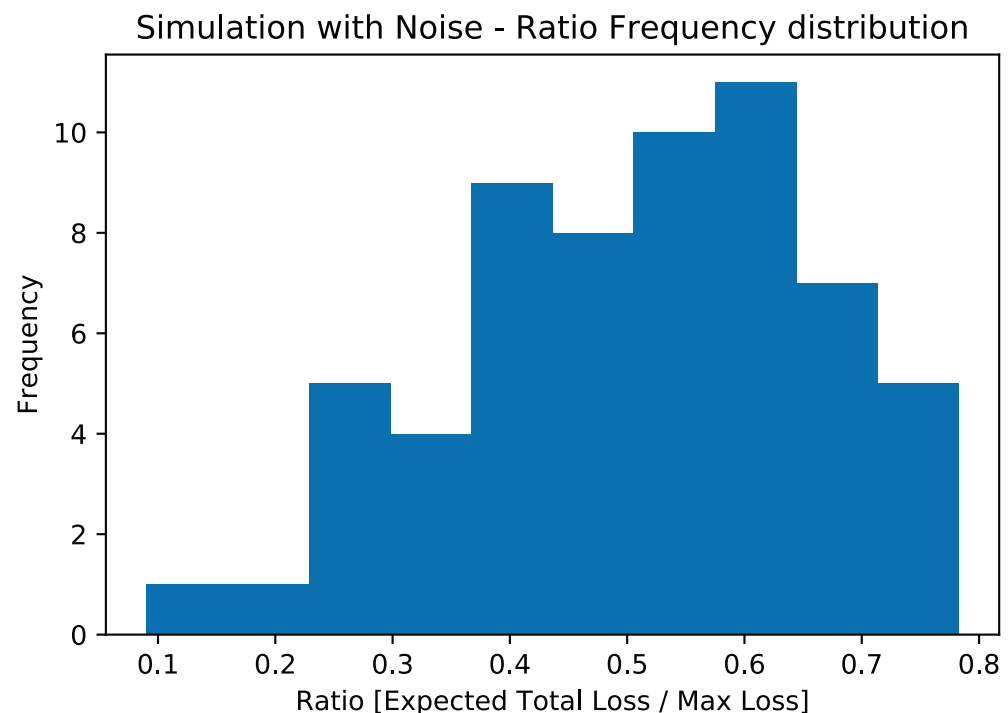


Figure 6. Ratio frequency distribution for the experiments conducted on classical machines, simulating the effects of noise thanks to noise models from the quantum experiments.

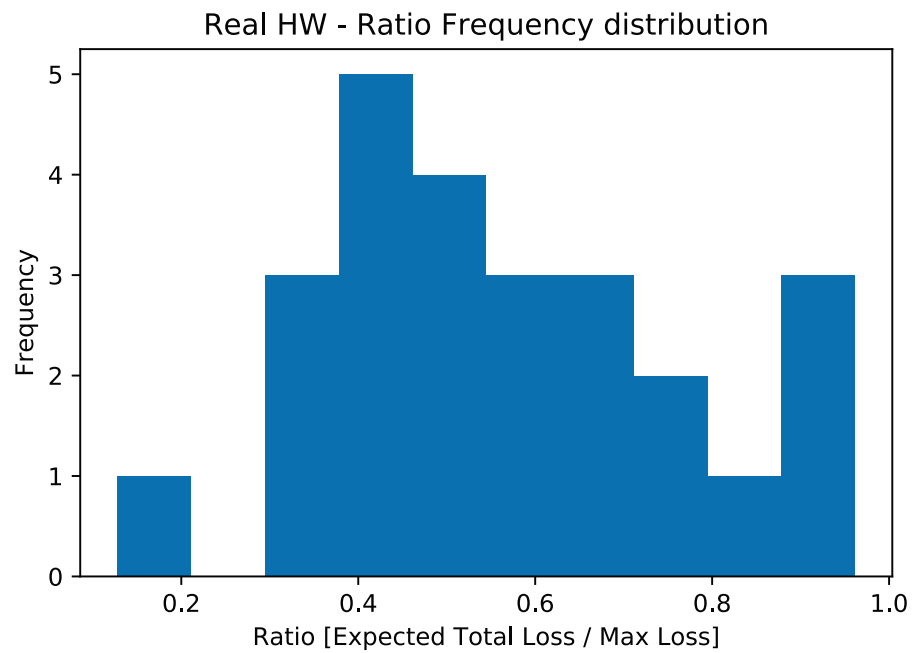


Figure 7. Ratio frequency distribution for the experiments conducted on quantum hardware.

4. Discussion

Noise is one of the major challenges facing quantum computing, as it can cause errors in qubits. The sources of noise can vary, from environmental factors such as temperature and electromagnetic radiation to decoherence and imperfections in the hardware itself. As a result, researchers have been actively investigating ways to characterize and mitigate the impact of noise on qubits [29–32].

What we observed, both by simulating quantum machines with their respective noise and by directly performing experiments on QPUs, was that the estimation of the expected total loss tended to converge towards half of the maximum possible loss (see Figures 6–8) regardless of the actual expected result. This configuration would correspond to a scenario in which the default probability of counterparties is exactly 50%. This is related to the loss of information due to the execution exceeding the qubits’ coherence time.

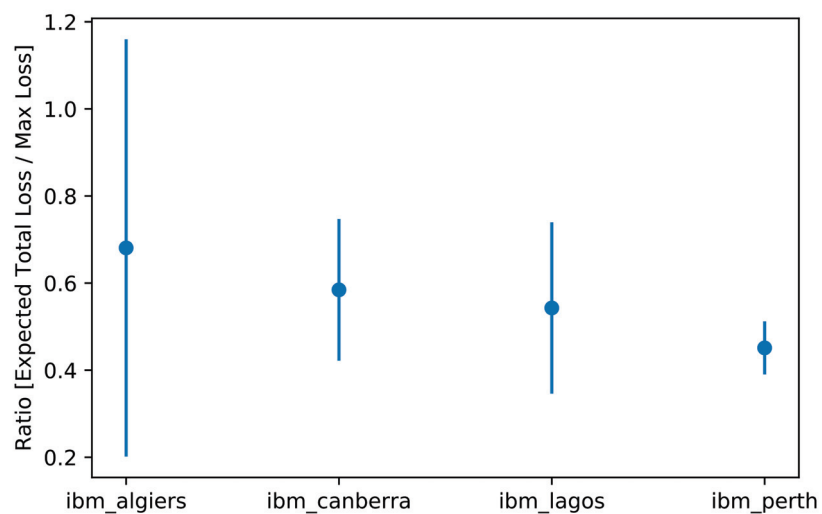


Figure 8. Detailed representation of results on real hardware by architecture.

A potentially interesting aspect that emerges from the analysis concerns **ibm_algiers**: this machine with a higher quantum volume (128) shows a greater variance around the central value, a potential indication of how the continuous improvement of this particular dimension suggests a future successful application of this and other algorithms on quantum machines.

4.1. Scalability and Complexity

While the proposed multiple-rotations variant of the quantum model presents an advantage over the original implementation [4] in terms of qubits required for small values of K and R , this advantage disappears when the algorithm scales to a realistic setting with thousands of assets and tens of factors. At this scaling, the overhead derived from the presence of the sum register becomes negligible, as the number of qubits it requires scales logarithmically as $O(\log_2(\sum_{i=1}^K LGD_i))$. However, using the Qiskit *LinearAmplitudeFunction* register [26] requires one additional qubit for each asset taken into account, thereby doubling the increase in terms of qubits that each additional asset entails. From a practical perspective, this translates into an increase in the width of the circuit with respect to the number of assets K that approaches $O(2K)$ instead of $O(K)$, which was the rate for the implementation in [4]. Moreover, for both proposed variants, the number of required qubits increases linearly with the number of factors, proportional to n_Z .

Regarding algorithm complexity, the iterative QAE introduced in [19] and used as a subroutine for our algorithm has a number of queries bounded by

$$\frac{1.4}{\epsilon} \ln \left(\frac{2}{\gamma} \log_2 \left(\frac{\pi}{4\epsilon} \right) \right), \quad (9)$$

where $1 - \gamma \in (0, 1)$ is the required confidence level and $\epsilon > 0$ is the target estimation error. Thus, if we set as an example $1 - \gamma = 99.9\%$ and $\epsilon = 0.05\%$, we need around 28 thousand applications of the Grover operator.

The main advantage of IQAE is that it does not increase the number of required qubits, as it does not require performing quantum phase estimation and still provides convergence proofs (which are instead missing for many of the other variants of the original QAE algorithm, such as the one in [33]).

For what concerns the uncertainty model, extrapolating from [4], the standard implementation of \mathcal{U} would require first K uncontrolled Y-rotations followed by $n_Z K * R$ controlled Y-rotations. As in the original analysis, we ignore the preparation of \mathcal{U}_Z as it can be performed efficiently and does not depend on K . It is important to underline the possibility of implementing \mathcal{U} more efficiently by duplicating the Z-qubits w times. Multiple copies of Z allow us to parallelize the preparation of the qubits representing the counterparties, achieving a depth of $(n_Z K * R) / w$ controlled Y-rotations. For further information and analysis, we refer the reader to [12] and particularly to [4], which contains an exhaustive analysis dedicated to the number of gates required for the original implementation.

For our implementations, we highlight the increase in terms of gates needed due to the use of the *LinearAmplitudeFunction* class. This circuit uses controlled linear rotations and comparator registers to implement the piecewise linear function on qubit amplitudes [26]. The number of such registers (and thus of the required gates) increases as $O(2^K)$. Thus, we observe a significant increase in terms of circuit depth in order to allow arbitrary values for the LGD parameters. However, alternative methods are already being proposed that can decrease the circuit depth needed for encoding the uncertainty model. In particular, in [34], the authors propose an alternative loading method based on quantum generative adversarial networks with encouraging results in terms of saved quantum resources. Moreover, in [35], a novel promising approximate quantum compiling approach is presented. This method would significantly lower the number of physical operations needed to implement complex quantum operators, such as the *LinearAmplitudeFunction*.

5. Conclusions

In this paper, we offered solutions to address the limitations of the quantum credit risk analysis algorithm, making it a more effective tool for future advancements in quantum computing technology. We illustrated our proposal and presented the results of several tests (both on quantum and classical hardware) that show the capabilities of our approach and the remaining challenges in terms of scalability and execution on actual QPUs.

The analysis highlights the need for further improvements in qubit coherence since our proposed measures require significantly more gates and qubits at scale than the previous implementation.

Thanks to the improvements proposed, our architecture can take non-integer values for the LGD vector, increasing input flexibility and allowing the use of real-world data. Additionally, our new uncertainty model with multiple risk factors corresponds to the framework most commonly used by big entities in the financial sector [36,37]. These enhancements allow for the creation of new benchmarks for the quantum model. These benchmarks should aim to enable a fair comparison with the classical algorithms currently used by financial institutions and, most importantly, will be able to use the same data for accurate comparison.

In conclusion, it is important to mention that while the proposed quantum credit risk model has the potential to improve the classical CRA process, its integration into production environments will require further research and development. In particular, the integration with existing data pipelines and possibly the design of an end-to-end digital twin will be necessary to evaluate the performance of the quantum model. Additionally, new regulations and legal requirements may be needed for the adoption of quantum algorithms in sensitive financial applications. These considerations highlight the importance of continued collaboration between researchers, financial institutions, and regulatory bodies to ensure the responsible and effective deployment of quantum technologies in the financial industry.

Author Contributions: Conceptualization, E.D., E.G., D.R., D.C. and R.U.; methodology, E.D.; software, E.D. and A.A.; validation, E.D., A.A., D.R. and R.U.; formal analysis, D.R.; investigation, E.D., A.A., D.R. and E.G.; resources, B.M., D.C. and R.U.; data curation, A.A.; writing—original draft preparation, E.D.; writing—review and editing, E.G., A.A. and B.M.; visualization, E.D. and A.A.; supervision, B.M., D.C., and R.U.; project administration, B.M. and D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in QVaR at <https://doi.org/10.5281/zenodo.7729267>, reference number [28].

Acknowledgments: The authors would like to thank the people at Intesa Sanpaolo and IBM Italia involved in the realization of this work. They provided inestimable insights regarding the use case and, most importantly, access to cutting-edge QPUs that made possible the development of this article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

QPU	Quantum processing unit
CRA	Credit risk analysis
VaR	Value at risk
QAE	Quantum amplitude estimation
PD	Probability of default
LGD	Loss given default

Appendix A. Quantum Processor Topologies

Appendix A.1. *ibm_lagos* and *ibm_perth*

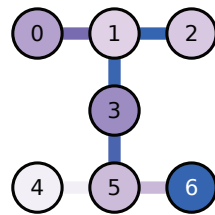


Figure A1. Topology for the Perth and Lagos IBM architectures.

Appendix A.2. *ibm_algiers* and *ibm_canberra*

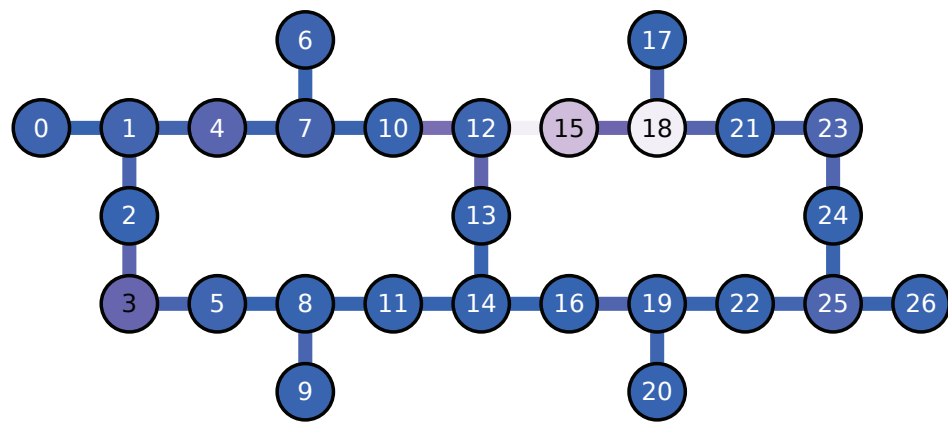


Figure A2. Topology for the Algiers and Canberra IBM architectures.

References

- Dri, E.; Giusto, E.; Aita, A.; Montrucchio, B. Towards practical Quantum Credit Risk Analysis. *J. Phys. Conf. Ser.* **2022**, *2416*, 12002. [CrossRef]
- Orús, R.; Mugel, S.; Lizaso, E. Quantum computing for finance: Overview and prospects. *Rev. Phys.* **2019**, *4*, 100028. [CrossRef]
- Egger, D.J.; Gambella, C.; Marecek, J.; McFaddin, S.; Mevissen, M.; Raymond, R.; Simonetto, A.; Woerner, S.; Yndurain, E. Quantum Computing for Finance: State-of-the-Art and Future Prospects. *IEEE Trans. Quantum Eng.* **2020**, *1*, 3030314. [CrossRef]
- Egger, D.J.; Gutiérrez, R.G.; Mestre, J.C.; Woerner, S. Credit Risk Analysis Using Quantum Computers. *IEEE Trans. Comput.* **2021**, *70*, 2136–2145. [CrossRef]
- Gestel, T.V.; Baesens, B. *Credit Risk Management*; Oxford University Press: Oxford, UK, 2008. [CrossRef]
- Hong, L.J.; Hu, Z.; Liu, G. Monte Carlo Methods for Value-at-Risk and Conditional Value-at-Risk: A Review. *ACM Trans. Model. Comput. Simul.* **2014**, *24*, 2661631. [CrossRef]
- Jorion, P. *Value at Risk*, 3rd ed.; McGraw-Hill: New York, NY, USA, 2006. [CrossRef]
- Pykhtin, M. Multi-factor adjustment. *Risk* **2004**, *17*, 85–90.
- Glasserman, P. *Monte Carlo Methods in Financial Engineering*; Stochastic Modelling and Applied Probability; Springer: New York, NY, USA, 2010.
- Danilowicz, R.L. Demonstrating the Dangers of Pseudo-Random Numbers. *SIGCSE Bull.* **1989**, *21*, 46–48. [CrossRef]
- Gupta, M.; Nene, M.J. Random Sequence Generation using Superconducting Qubits. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 640–645. [CrossRef]
- Woerner, S.; Egger, D.J. Quantum risk analysis. *NPJ Quantum Inf.* **2019**, *5*, 15. [CrossRef]
- Lütkebohmert, E. The Asymptotic Single Risk Factor Model. In *Concentration Risk in Credit Portfolios*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 31–42. [CrossRef]
- Jongh, R.D.; Verster, T.; Reynolds, E.; Joubert, M.; Raubenheimer, H. A Critical Review Of The Basel Margin Of Conservatism Requirement In A Retail Credit Context. *Int. J. Bus. Econ. Res. (IBER)* **2017**, *16*, 257–274. [CrossRef]
- Hamerle, A.; Liebig, T.; Roesch, D. Credit Risk Factor Modeling and the Basel Ii IRB Approach. *SSRN Electron. J.* **2003**, *2*, 2793952. [CrossRef]
- Researchers Program. Available online: <https://quantum-computing.ibm.com/programs/researchers> (accessed on 10 March 2023).

17. Qiskit Runtime: A Cloud-Native, Pay-As-You-Go Service for Quantum Computing. 2022. <https://www.ibm.com/cloud/blog/how-to-make-quantum-a-pay-as-you-go-cloud-service> (accessed on 10 March 2023).
18. Brassard, G.; Høyer, P.; Mosca, M.; Tapp, A. Quantum amplitude amplification and estimation. In *Quantum Computation and Information*; Contemporary Mathematics; American Mathematical Society: Providence, RI, USA, 2002; Volume 305, pp. 53–74.
19. Grinko, D.; Gacon, J.; Zoufal, C.; Woerner, S. Iterative quantum amplitude estimation. *NPJ Quantum Inf.* **2021**, *7*, 52. [CrossRef]
20. Rutkowski, M.; Tarca, S. Regulatory capital modeling for credit risk. *Int. J. Theor. Appl. Financ.* **2015**, *18*, 1550034. [CrossRef]
21. De Basilea, C.d.S.B. *Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework—Comprehensive Version*; BIS: Basel, Switzerland, 2006.
22. Grover, L.; Rudolph, T. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv* **2002**, arXiv:quant-ph/0208112.
23. Chen, N.F.; Roll, R.; Ross, S. Economic Forces and the Stock Market. *J. Bus.* **1986**, *59*, 383–403. [CrossRef]
24. Aleksandrowicz, G.; Alexander, T.; Barkoutsos, P.; Bello, L.; Ben-Haim, Y.; Bucher, D.; Cabrera-Hernández, F.J.; Carballo-Franquis, J.; Chen, A.; Chen, C.F.; et al. *Qiskit: An Open-Source Framework for Quantum Computing*; Zenodo: Geneva, Switzerland, 2019. [CrossRef]
25. WeightedAdder—Qiskit 0.36.1 Documentation. Available online: <https://qiskit.org/documentation/stubs/qiskit.circuit.library.WeightedAdder.html> (accessed on 10 March 2023).
26. LinearAmplitudeFunction—Qiskit 0.36.1 Documentation. Available online: <https://qiskit.org/documentation/stubs/qiskit.circuit.library.LinearAmplitudeFunction.html> (accessed on 10 March 2023).
27. Gacon, J.; Zoufal, C.; Woerner, S. Quantum-Enhanced Simulation-Based Optimization. In Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, 12–16 October 2020; IEEE: New York, NY, USA, 2020. [CrossRef]
28. Dri, E.; Aita, A. QVaR, 2022. [CrossRef]
29. Clerk, A.A.; Devoret, M.H.; Girvin, S.M.; Marquardt, F.; Schoelkopf, R.J. Introduction to quantum noise, measurement, and amplification. *Rev. Mod. Phys.* **2010**, *82*, 1155–1208. [CrossRef]
30. Harper, R.; Flammia, S.T.; Wallman, J.J. Efficient learning of quantum noise. *Nat. Phys.* **2020**, *16*, 1184–1188. [CrossRef]
31. Oliveira, D.; Giusto, E.; Dri, E.; Casciola, N.; Baheri, B.; Guan, Q.; Montrucchio, B.; Rech, P. QuFI: A Quantum Fault Injector to Measure the Reliability of Qubits and Quantum Circuits. In Proceedings of the 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Baltimore, MD, USA, 27–30 June 2022; pp. 137–149. [CrossRef]
32. Endo, S.; Benjamin, S.C.; Li, Y. Practical Quantum Error Mitigation for Near-Future Applications. *Phys. Rev. X* **2018**, *8*, 031027. [CrossRef]
33. Suzuki, Y.; Uno, S.; Raymond, R.; Tanaka, T.; Onodera, T.; Yamamoto, N. Amplitude estimation without phase estimation. *Quantum Inf. Process.* **2020**, *19*, 75. [CrossRef]
34. Zoufal, C.; Lucchi, A.; Woerner, S. Quantum Generative Adversarial Networks for learning and loading random distributions. *NPJ Quantum Inf.* **2019**, *5*, 103. [CrossRef]
35. Madden, L.; Simonetto, A. Best Approximate Quantum Compiling Problems. *ACM Trans. Quantum Comput.* **2022**, *3*, 3505181. [CrossRef]
36. Bindseil, U.; Sotamaa, K.; Amado, R.; Honings, N.; Chiappa, G.; Boux, B.; Föttinger, W.; Ledoyen, P.; Schwartzlose, H.; van der Hoorn, H.; et al. *The Use of Portfolio Credit Risk Models in Central Banks*; Occasional Paper Series 64; European Central Bank: Main, Germany, 2007.
37. Balzarotti, V.; Falkenheim, M.; Powell, A. On the Use of Portfolio Risk Models and Capital Requirements in Emerging Markets. *World Bank Econ. Rev.* **2002**, *16*, 197–211. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Dynamic Asset Allocation with Expected Shortfall via Quantum Annealing

Hanjing Xu ¹, Samudra Dasgupta ^{2,3,4}, Alex Pothen ¹ and Arnab Banerjee ^{2,3,*}

¹ Department of Computer Science, Purdue University, West Lafayette, IN 47906, USA; xu675@purdue.edu (H.X.)

² Department of Physics, Purdue University, West Lafayette, IN 47906, USA

³ Oak Ridge National Laboratory, Quantum Computing Institute, Oak Ridge, TN 37831, USA

⁴ Bredeben Center, University of Tennessee, Knoxville, TN 37996, USA

* Correspondence: arnabb@purdue.edu

Abstract: Recent advances in quantum hardware offer new approaches to solve various optimization problems that can be computationally expensive when classical algorithms are employed. We propose a hybrid quantum-classical algorithm to solve a dynamic asset allocation problem where a target return and a target risk metric (expected shortfall) are specified. We propose an iterative algorithm that treats the target return as a constraint in a Markowitz portfolio optimization model, and dynamically adjusts the target return to satisfy the targeted expected shortfall. The Markowitz optimization is formulated as a Quadratic Unconstrained Binary Optimization (QUBO) problem. The use of the expected shortfall risk metric enables the modeling of extreme market events. We compare the results from D-Wave's 2000Q and Advantage quantum annealers using real-world financial data. Both quantum annealers are able to generate portfolios with more than 80% of the return of the classical optimal solutions, while satisfying the expected shortfall. We observe that experiments on assets with higher correlations tend to perform better, which may help to design practical quantum applications in the near term.

Keywords: portfolio optimization problem; Quadratic Unconstrained Binary Optimization (QUBO); quantum annealing; hybrid algorithm

Citation: Xu, H.; Dasgupta, S.; Pothen, A.; Banerjee, A. Dynamic Asset Allocation with Expected Shortfall via Quantum Annealing. *Entropy* **2023**, *25*, 541. <https://doi.org/10.3390/e25030541>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 2 February 2023

Revised: 1 March 2023

Accepted: 17 March 2023

Published: 21 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

We describe a hybrid quantum-classical algorithm to solve a dynamic asset allocation problem where the targeted return and expected-shortfall (ES)-based risk appetite are specified. Since both the return as well as the shortfall are functions of the chosen asset allocation, we treat the return as a constraint in a modified Markowitz framework, and optimize the allocation strategy to meet the requirements of the expected shortfall using an iterative procedure that solves the Markowitz Optimization problem at each iteration. The latter optimization problem is solved by a Quadratic Unconstrained Binary Optimization (QUBO) formulation on a quantum annealer, while the iterative procedure to compute the shortfall is performed by a classical algorithm.

Quantum annealing offers a highly parallelized approach for solving optimization problems by using quantum tunneling from a manifold of high-energy solutions to the ground state. A common approach to embed the optimization problem into an Ising quantum annealer is to convert it to a QUBO problem [1–4]. Several examples have been explored so far in the literature, including the maximum clique [5], scheduling [6] and graph coloring problems [7], among others [8].

The portfolio optimization problem, introduced by Harry Markowitz [9] in 1952, investigates how investors could use the power of diversification to optimize portfolios by minimizing risk, and serves as a foundation for later models, such as the Black–Litterman model [10]. The original Markowitz Optimization problem used volatility as the measure

of the risk. However, it is now known that volatility changes with time [11]; hence, treating it as a constant is risky and sub-optimal. Furthermore, it often fails to characterize the market during extreme events or “shocks”, for example, the 2008 mortgage crisis which led to an abrupt collapse of the market with the insolvency of Lehman Brothers. As a result, modern finance practitioners prefer to use a time-varying risk metric such as stochastic volatility, Value-at-Risk (VaR) or the expected shortfall. The latter is defined as the average loss that can be expected when the loss has already exceeded a specific threshold [12]. The advantages of expected shortfall over other risk measurements such as volatility or Value-at-Risk are discussed in [11].

It is NP-Hard to solve general quadratic optimization problems [13]. For convex quadratic optimization problems such as the portfolio optimization problem, however, there exist polynomial time algorithms that takes $O(n^{7/2}L)$ time [14], where n is the number of variables and L bounds the number of digits for each integer. Hence it is prohibitive to solve large-scale portfolio optimization problems exactly using classical methods due to the high time complexity. Hence as more versatile and scalable quantum computing devices—currently quantum annealers—enter the market, we explore solving the portfolio optimization problem on two such machines available today using QUBO formulations.

In Grant et al. [15], the authors have benchmarked the performance of a D-Wave 2000Q quantum annealer on solving the Markowitz Optimization Problem with a relatively small size of 20 logical variables and random data. Our study has the following novel contributions:

- We demonstrate how optimization problems with non-polynomial constraints such as the Expected Shortfall could be solved with a hybrid quantum-classical, iterative approach that requires no additional qubits. An alternative approach would encode such constraints directly into a QUBO by converting them first to a multilinear polynomial through Fourier analysis [16], and then to a quadratic polynomial using methods described in [17–19]. However, in this approach, in the worst-case the number of binary variables will grow exponentially due to non-trivial higher-order terms generated from the Fourier expansion, which severely limits the problem sizes that we can solve on the current generation of quantum hardware.
- To the best of our knowledge, quantum computing has not been employed prior to this study for solving Expected-Shortfall based dynamic asset-allocation problems [12]. Previous approaches (e.g., [15]) have employed the classical Mean-variance framework. However, static variance is no longer used in modern finance as it is well known that volatility fluctuates with time and hence it needs to be modeled in a statistical framework that captures non-stationarity. Moreover, industrial practitioners prefer tail-risk measures such as Value at Risk and Expected Shortfall (the latter is considered cutting edge in risk management) since true risk is associated with the fluctuations in the negative return, and is not symmetric with respect to positive and negative returns (i.e., no one minds a surprise positive return).
- Thirdly, this is one of the first papers that uses quantum computing for portfolio optimization using real financial data (using ETF and currency data) on a real quantum computer (i.e., not simulation) in an accurate industry setting. Previous approaches have used random data (e.g., [15]).

We further explored our algorithm’s performance on two generations of quantum annealers offered by D-Wave, with up to 115 logical variables. We provide experimental results on both the Advantage (Pegasus topology) and 2000Q (Chimera topology) D-Wave quantum annealers. The results are generally close to the optimal portfolios obtained by classical optimization methods, in terms of final returns and Sharpe ratios (return/standard deviation of the return in a time period).

The paper is structured as follows. Section 2 defines the expected shortfall based dynamic asset allocation problem and lays out a hybrid algorithm for solving it. Section 3 provides the technical background on D-Wave’s quantum annealer and maps the Mean-Variance Markowitz problem on it. Section 4 discusses the experimental results on both

D-Wave 2000Q and Advantage systems. Section 5 states our conclusions and lists future research directions.

2. The Problem of Dynamic Asset Allocation

The problem of dynamic asset allocation is to allocate/invest an amount of money into N assets, while satisfying an expected return and keeping the risk below a given threshold. To make the problem more specific, we need to describe the input data and variables.

- The historical return matrix R is obtained from Yahoo Finance [20–25] for the assets mentioned in Section 4.2 with N rows and T_{total} columns, where N is the number of assets and T_{total} is the number of days data are collected. We divide the return matrix R into periods of T days and index the data for each time period, for example, R_t represents the return data from t -th time period.
- The vector of asset means μ_t is computed from R_t .
- The co-variance matrix C_t calculated from the matrix R_t as

$$C_{t,i,j} = \frac{(e_i^T R_t - \mu_{t,i} \mathbf{1}) \cdot (e_j^T R_t - \mu_{t,j} \mathbf{1})}{T - 1}, \tag{1}$$

where e_j is the column vector of all zeros except with a one at the j -th position.

Asset allocation is especially interesting to financial practitioners during a time period with unpredictable market turbulence with goal of minimizing risk while achieving a target return. The risk is upper bounded by a consumer-driven risk appetite. A data sheet of the assets' daily returns (profit one can earn if buying an asset the previous day and selling the next) for the previous three months is available. The risk threshold can be set using observed market metrics from a volatile time period, for example, the 2008 market crash. The algorithm uses the assets' historical return data to estimate the trends of the assets' performance and their correlations. Options for risk measurements include:

- Volatility: the standard deviation of the portfolio return.
- Value-at-Risk at level α : the smallest number y such that the probability that a portfolio does not lose more than $y\%$ of total budget is at least $1 - \alpha$.
- Expected Shortfall at level α : the expected return from the worst $\alpha\%$ cases. It is defined as follows:

$$ES_\alpha(w_t, R_t) = \text{mean}(\text{lowest } \alpha\% \text{ from } w_t^T R_t). \tag{2}$$

We focus on the expected shortfall as our risk measurement for the rest of the paper as it is the modern approach preferred by practitioners (as mentioned earlier in Section 1). The problem can be expressed as follows where the weight vector w indicates what fraction of the budget is invested in each asset:

(P1) Minimize the expected shortfall $ES_\alpha(w_t, R_t)$ under the constraints that the expected return is satisfied, the variance of the portfolio is small, and all assets are invested.

It is possible to write the expected shortfall based portfolio optimization as a linear program [26], but it requires adding $N + 1$ variables and $2N$ constraints where N is the number of assets. Since the expected shortfall cannot be expressed by a quadratic formulation natively, we opt to use it as a convergence criterion instead of including it in the optimization problem directly. To justify this approach, assuming that the assets' historical returns follow a Gaussian distribution, we can approximate the expected shortfall of a given portfolio P by:

$$ES_\alpha(P) = \mu + \sigma \frac{\phi(\Phi^{-1}(\alpha))}{1 - \alpha}, \tag{3}$$

where μ is the expected return, σ is the volatility of the portfolio, and $\phi(x)$ and $\Phi(x)$ are the Gaussian probability distribution and cumulative distribution functions, respectively [27].

The expected shortfall has positive correlation with the volatility and in turn, the variance of the portfolio ([11,28]).

Hence we propose a bilevel optimization approach described in Figure 1 to solve Problem (P1).

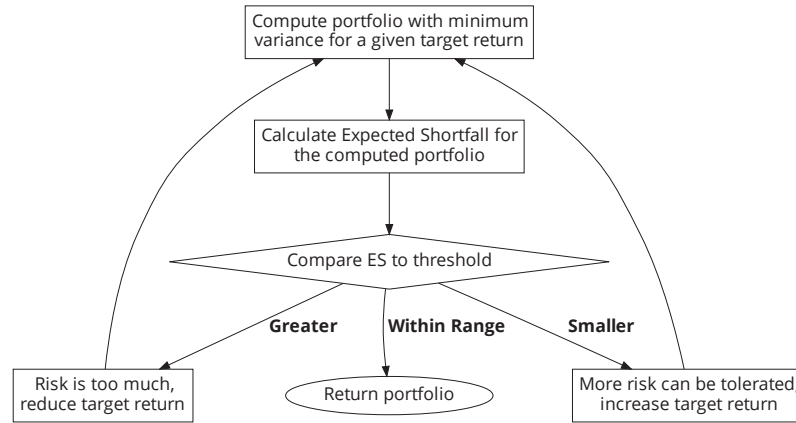


Figure 1. A flowchart for the proposed algorithm for computing optimal portfolio with a threshold on the expected shortfall.

Given a balance sheet of the assets’ returns in the history, we create multiple time periods each with T days. After picking target return p_t for one of the time periods t , we choose a reference asset that is representative of the portfolio, and set a target expected shortfall for that asset computed from its volatility in the year 2008, its volatility in the time period t , and its shortfall in 2008. (The precise expression is included in Algorithm 1). Then we use the Markowitz Optimization problem [9] to allocate assets within the portfolio in order to minimize volatility with the constraint that the target return is met. Next we compute the expected shortfall from the current allocation of assets. If the target expected shortfall is not met by the current allocation, then we adjust the target return value and iteratively solve the Markowitz Optimization problem. We terminate when the target expected shortfall is met, or the target return cannot be met as the maximum return of all assets is smaller than the target return.

Now we describe the Markowitz portfolio optimization procedure. Its QUBO formulation will be provided in the next section. The Markowitz Optimization problem can be expressed by the quadratic optimization problem

$$\begin{aligned}
 \min_w \quad & w_t^T C_t w_t \\
 \text{s.t.} \quad & \mu_t^T w_t = p_t, \\
 & \sum_i w_{t,i} = 1, w_{t,i} \geq 0 \quad \forall i.
 \end{aligned} \tag{4}$$

where p_t is the target portfolio return during t -th time period. The constraint $\mu_t^T w_t = p_t$ ensures that the target return is met, $\sum w_{t,i} = 1$ indicates we want to invest all of the resources, and $w_{t,i} \geq 0$ means short selling is not allowed. With all the constraints satisfied, we minimize $w_t^T C_t w_t$, that is, the variance of the portfolio at $t - th$ time period. However, with our bilevel optimization, we treat the constraints as soft constraints, that is, small violations of their values are permitted. The optimizer can return portfolios with small variance even when the expected return falls short of the target; if the sum of weight is not equal to 1, we can scale the weights of the assets to sum to 1.

Algorithm 1: Expected Shortfall based Dynamic Asset Allocation during t .

```

Input:  $\mu_t, R_t, \sigma_{ref}, \sigma_{ref_t}, ES_{ref}, \alpha$ 
Output:  $w_t$ 
Let  $p_t = \text{mean}(\mu_t)$ ; ▷ Initialize the target return
Let  $EST_t = \frac{\sigma_{ref}}{\sigma_{ref_t}} ES_{ref}$ ; ▷ Initialize the target expected shortfall
Let  $C_t = \text{cov}(R_t)$ ; ▷ Compute the co-variance matrix from the returns
while True do
  if  $p > \max(\mu_t)$  then
    | Return 0; ▷ Return constraint cannot be satisfied
  Solve Equation (4) for  $w_t$ ;
  Let  $ES_t = ES_\alpha(w_t, R_t)$ 
  if  $\frac{|ES_t|}{|EST_t|} > 1 + \epsilon$  then
    | Let  $p_t = p_t(1 - \delta)$ ; ▷ Decrease target return for lower risks
  else if  $\frac{|ES_t|}{|EST_t|} < 1 - \epsilon$  then
    | Let  $p_t = p_t(1 + \delta)$ ; ▷ Increase target return as there is room for
    | more risks
  else
    | Return  $w_t$ ;
end

```

Algorithm 1 provides the pseudo-code for the algorithm for expected shortfall based asset allocation. Here σ_{ref} is the volatility of a reference asset’s returns during the market crash in 2008; the reference asset is chosen from among the assets to be representative of the market trend, for example, SPDR S&P 500 ETF Trust (SPY). The variable σ_{ref_t} is the volatility of the reference asset’s returns during the time window t ; ES_{ref} is reference asset’s expected shortfall during the market crash; EST_t is the target expected shortfall at time window t ; α is the risk level parameter; ES_t is the expected shortfall for the computed portfolio during the optimization process at time window t ; ϵ is the error tolerance parameter; and δ is the momentum parameter that is adjusted dynamically. Figure 2 shows the ratio between the variance and expected shortfall in different iterations of Algorithm 1 from ETFs consisting of 6 assets whose returns were obtained from December 2019 to May 2020. The monotonic one-to-one tracking justifies why optimization problems with expected shortfall constraint can be solved iteratively using the Markowitz Mean-Variance framework.

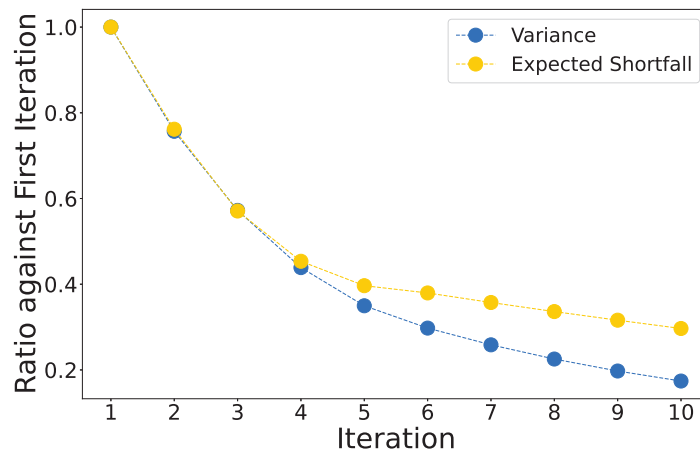


Figure 2. The y-axis tracks the ratio between the variance and expected shortfall with $\alpha = 5$ in later iterations against their respective values in the first iteration of Algorithm 1 running on 6 ETF assets. The expected shortfall decreases at a different rate from the variance but each iteration in the algorithm is guaranteed to make progress towards the target expected shortfall, which ensures convergence.

3. A Hybrid Quantum Classical Algorithm

3.1. Algorithm Overview

We will use a hybrid quantum classical algorithm to solve the quadratic optimization problem given in Equation (4) with a quantum annealer backend.

Quantum annealing (QA) [29–31] is the quantum analog of the classical annealing where the disorder is introduced quantum mechanically instead of thermally via applying the Pauli matrix x on every qubit as in Equation (5).

$$\mathcal{H}_I = - \sum_{i=1}^N \sigma_i^x, \tag{5}$$

This Hamiltonian does not commute with the problem Hamiltonian in Equation (6)

$$\mathcal{H}_P = - \sum_i h_i \sigma_i^z - \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z, \tag{6}$$

where σ_i^z is the Pauli matrix z acting on qubit i , h_i is the magnetic field on qubit i and J_{ij} defines the coupling strength between qubits i and j [32]. The spin configurations of the ground states of Equation (6) also minimize the Ising model problem:

$$\min_s E(s) = - \sum_i h_i s_i - \sum_{i,j} J_{ij} s_i s_j, \quad s_i \in \{-1, 1\}, \tag{7}$$

where s_i is the spin, h is the external longitudinal magnetic field strength vector and the matrix J represents the coupler interactions. Moreover, the general two-dimensional Ising problem within a magnetic field is NP-hard [33]. And in the case of spin-glass three-dimensional Ising model with lattice size of $N = lmn$, the complexity is $O(2^{mn})$ [34], which is NP-Hard as well.

During the QA process, combining both Hamiltonians in Equations (5) and (6), at time t the system evolves under the following Hamiltonian:

$$\mathcal{H}(t) = A\left(\frac{t}{T}\right)\mathcal{H}_I + B\left(\frac{t}{T}\right)\mathcal{H}_P. \tag{8}$$

Here T is the total annealing time and the system is initialized to the ground state of \mathcal{H}_I , which is a superposition of all qubits in the z basis. Functions $A\left(\frac{t}{T}\right)$ and $B\left(\frac{t}{T}\right)$ describe the change of influences from disorder and problem Hamiltonians on the system. \mathcal{H}_I dominates \mathcal{H}_P initially and slowly (adiabatically) changes to the opposite while the influence of \mathcal{H}_I vanishes at the end of the annealing process, thus removing disorder from the system. The system will then settle into one of the low energy states.

Due to unavoidable experimental compromises [35], QA serves as an intermediate step towards universal adiabatic quantum computation (AQC) [36,37] as the system evolves under a time-dependent Hamiltonian

$$\mathcal{H} = [1 - s(t)]\mathcal{H}_I + s(t)\mathcal{H}_P, \tag{9}$$

where $s(t)$ changes from 0 to 1. When conditions on internal energy gap and time scales are met [38], the system will remain in its ground state at all times, which is different from QA.

A Quadratic Unconstrained Binary Optimization (QUBO) problem of the form

$$\min_x Q(x) = \sum_i h_i x_i + \sum_{i,j} J_{ij} x_i x_j, \quad x_i \in \{0, 1\} \tag{10}$$

aims to minimize a mathematical function with linear and quadratic terms; here any combination of $x_i \in \{0, 1\}, \forall i$ is feasible. It can be converted to the Ising model shown in Equation (7) by a one-to-one mapping of the variables: $x_i = \frac{1+s_i}{2}$. We will use the QUBO

formulation for the rest of the paper but note that quantum annealers from D-Wave require the QUBO problems to be transformed into Ising models before execution.

Consider a standard binary optimization problem with a linear or quadratic objective function $f(x)$ and linear constraints $Ax = b$, where $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^{m \times 1}$.

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & Ax = b, \\ & \text{where } x \in \{0, 1\}^{n \times 1}. \end{aligned} \tag{11}$$

We can rewrite it as a QUBO

$$Q(s) = f(x) + \lambda(Ax - b)^T(Ax - b) \tag{12}$$

to be minimized by quantum annealers with a large enough $\lambda \in \mathbb{R}^+$ to guarantee that the constraint is satisfied in the optimal solutions.

We will now discuss how to convert the Markowitz Optimization problem with continuous variables in Equation (4) to a QUBO problem.

First we write Equation (4) as an unconstrained optimization problem with penalty coefficients λ_1 and λ_2 (the subscripts t are dropped for better readability):

$$Q = \left(\sum_i^n \sum_j^n C_{i,j} w_i w_j \right) + \lambda_1 \left(\sum_i^n \mu_i w_i - p \right)^2 + \lambda_2 \left(\sum_i^n w_i - 1 \right)^2, \tag{13}$$

where λ_1 and λ_2 scale the constraint penalties. Minimizing Equation (13) is equivalent to

$$\begin{aligned} \min Q = \left(\sum_i^n \sum_j^n C_{i,j} w_i w_j \right) + \lambda_1 \left[\left(\sum_i^n \mu_i w_i \right)^2 - 2p \sum_i^n \mu_i w_i \right] \\ + \lambda_2 \left[\left(\sum_i^n w_i \right)^2 - 2 \sum_i^n w_i \right], \end{aligned} \tag{14}$$

after expanding the squared terms and eliminating the constants. When the constraints are satisfied exactly, we have

$$\lambda_1 \left[\left(\sum_i^n \mu_i w_i \right)^2 - 2p \sum_i^n \mu_i w_i \right] = -\lambda_1 p^2, \tag{15}$$

and

$$\lambda_2 \left[\left(\sum_i^n w_i \right)^2 - 2 \sum_i^n w_i \right] = -\lambda_2. \tag{16}$$

We use k binary variables $x_{i,1} \dots x_{i,k} \in \{0, 1\}$ to approximate each continuous variable w_i in Equation (4) with a finite geometric series

$$w_i = \sum_{a=1}^k 2^{-a} x_{i,a}. \tag{17}$$

The larger k is, the more precision w_i has. However, larger k also widens the differences between the coupler strengths— J terms in Equation (10). Although the coupler strengths for D-Wave annealers can be set at any double-precision floating point number between -1 and 1 , precision errors may pose a challenge due to integrated control errors (ICE) [39]. In our experiments, we set $k = 5$ which empirically gives us the best approximations to the optimal solutions for Equation (4). For larger k we risk the errors dominating the

coupler coefficients, rendering those additional qubits unreliable. We set λ_1 to p^{-2} and λ_2 to 1 to bound the penalty terms in Equation (14) to -2 . Additionally, we scale the objective by a factor of λ_3 to around 1 such that penalty terms in the optimal solutions to Equation (14) remain relatively small while not overwhelming the objective. If penalties dominate the objective, it may introduce numerous local minima to the energy landscape and the optimizer will suffer from barren plateaus. Alternatively, if the objective dominates penalties, constraints will be violated significantly. The soft constraints enable us to obtain better portfolios as presented in Section 4.4.

Substituting Equation (17) in to Equation (14), we have the final binary optimization formalism

$$\begin{aligned}
 f(x) = & \left[\left(\sum_i^n \sum_a^k \mu_i 2^{-a} x_{i,a} \right)^2 - 2p \sum_i^n \sum_a^k \mu_i 2^{-a} x_{i,a} \right] \\
 & + p^{-2} \left[\left(\sum_i^n \sum_a^k 2^{-a} x_{i,a} \right)^2 - 2 \sum_i^n \sum_a^k 2^{-a} x_{i,a} \right] \\
 & + \lambda_3 \sum_i^n \sum_j^n \sum_a^k \sum_b^k C_{i,j} 2^{-a-b} x_{i,a} x_{j,b}, \quad (18)
 \end{aligned}$$

which is quantum-annealable as it only has linear and quadratic interactions.

3.2. Previous Work

Rosenberg et al. [40] solve the multi-period portfolio optimization problem using D-Wave’s quantum annealer:

$$\begin{aligned}
 \max_w & \sum_{t=1}^T (\mu_t^T w_t - \frac{\gamma}{2} w_t^T \Sigma_t w_t - \Delta w_t^T \Lambda_t \Delta w_t \\
 & + \Delta w_t^T \Lambda'_t \Delta w_t) \\
 \text{s.t.} & \sum_{n=1}^N w_{nt} = K, \forall t, \quad w_{nt} \leq K', \forall t, \forall n.
 \end{aligned} \quad (19)$$

Here T is the number of time steps, and N is the number of assets. At each time step t , μ_t represents the forecast returns, w_t are holdings for each asset, Σ_t is the forecast covariance matrix, Λ_t and Λ'_t are coefficients for transaction costs related to temporary and permanent market impacts, respectively, which penalize changes in the holdings if the corresponding terms are positive. Additionally, γ is the risk aversion factor.

Equation (19) seeks to maximize returns considering constraints on asset size. Specifically, the sum of asset holdings is constrained by K and the maximum allowed holdings of each asset is K' . For small problems ranging from 12 to 584 variables, D-Wave’s 512 and 1152-qubit systems are able to find optimal solutions with high probability.

Venturelli and Kondratyev [41] focus on the following QUBO problem where the task is to select M assets from a pool of N assets:

$$\min_q \sum_{i=1}^N a_i q_i + \sum_{i=1}^N \sum_{j=i+1}^N b_{ij} q_i q_j + P \left(M - \sum_{i=1}^N q_i \right). \quad (20)$$

The variable q_i is 1 if asset i is selected and 0 otherwise. The coefficient a_i indicates the attractiveness of the i -th asset and b_{ij} is the pairwise diversification penalties (positive) or rewards (negative). The penalty coefficient P scales the constraint on the number of selected assets to make sure it is satisfied in the optimal solution. The authors have explored

the benefits of reverse annealing on D-Wave systems, and report one to three orders of magnitude speed-up in time-to-solution with reverse annealing.

The problem considered by Phillipson and Bhatia [42] is similar to the Markowitz Optimization problem but with binary variables indicating asset selections instead of real weights. The authors report comparable results from D-Wave's hybrid solver to other state of the art classical algorithms and solvers including simulated annealing [43,44], genetic algorithm [45,46], linear optimization problems [47] and local search [48].

Grant et al. [15] benchmark the Markowitz Optimization problem on a D-Wave 2000Q processor with real weight variables and price data generated uniformly at random, and explore how embeddings, spin reversal and reverse annealing affect the success probability. Hegade et al. [49] solve the same problem with added counterdiabatic terms on circuit-based quantum computers and see improvements on success probabilities using digitized-adiabatic quantum computing (DAdQC) and Quantum Approximation Optimization Algorithm (QAOA) [49].

We extend the general QUBO formulation in [15] to solve the asset allocation problem, with the expected shortfall as the risk metric, using the Markowitz Optimization problem as a subroutine at each iterative step of the algorithm. We use our algorithms on real-world ETF and currency data. Additionally, we present the results on the newly-available Advantage processor and experiment on problems with up to 115 logical variables, up from 20 in [15].

4. Experimental Setup and Results

4.1. D-Wave Quantum Annealer

We start by discussing the latest quantum annealing technologies offered by D-Wave as the solvability of the problem is dependent on the architecture. D-Wave quantum annealers are specifically designed to solve Ising problems natively. Currently two types of quantum annealers are offered by D-Wave: 2000Q processor with Chimera topology and the Advantage processor with Pegasus topology. The latter was made publicly available in 2020 and it has more qubits (5760 vs. 2048) and better connectivity than the former. The qubits in the Chimera topology have 5 couplers per qubit while in the Pegasus topology they have 15 couplers per qubit [50]. It is not always possible to formulate an optimization problem to match the Chimera or Pegasus topologies exactly. Therefore minor embeddings are necessary to map the problems to D-Wave processors. Such embeddings usually require the users to map multiple physical qubits to one logical variable with constraints such that every qubit on the 'chain' behaves the same, which significantly reduces the total size of the problems that can be solved on the quantum annealers.

Furthermore, it is advisable to have uniform chain lengths (number of qubits representing a single variable) for more predictable chain dynamics during the anneal [51]. Algorithms in [52] detail such procedures for fully-connected graphs which is the underlying logical graph for the portfolio optimization problem. A full-yield 2000Q processor can map up to 64 logical variables and an Advantage processor can map around 180 logical variables. A comparison between the embedding of the two topologies is shown in Figure 3. In our experiments, we use the *find_clique_embedding* function from *dwave-system* to map fully-connected graphs to either the Chimera or the Pegasus topology.

4.2. Test Input and Annealer Parameters

We pick the top-six ETFs by trading volumes, EEM, QQQ, SPY, SLV, SQQQ and XLF, and six major currencies' USD exchange rates, AUD, EUR, GBP, CNY, INR and JPY, for most of the tests below. The reference assets for ETF and currency tests are SPY and EUR, respectively. For the tests in Section 4.5 we use 12 and 23 assets respectively and pick the top ETFs by trading volumes again. We choose the parameter α in the definition of expected shortfall to be 5%.

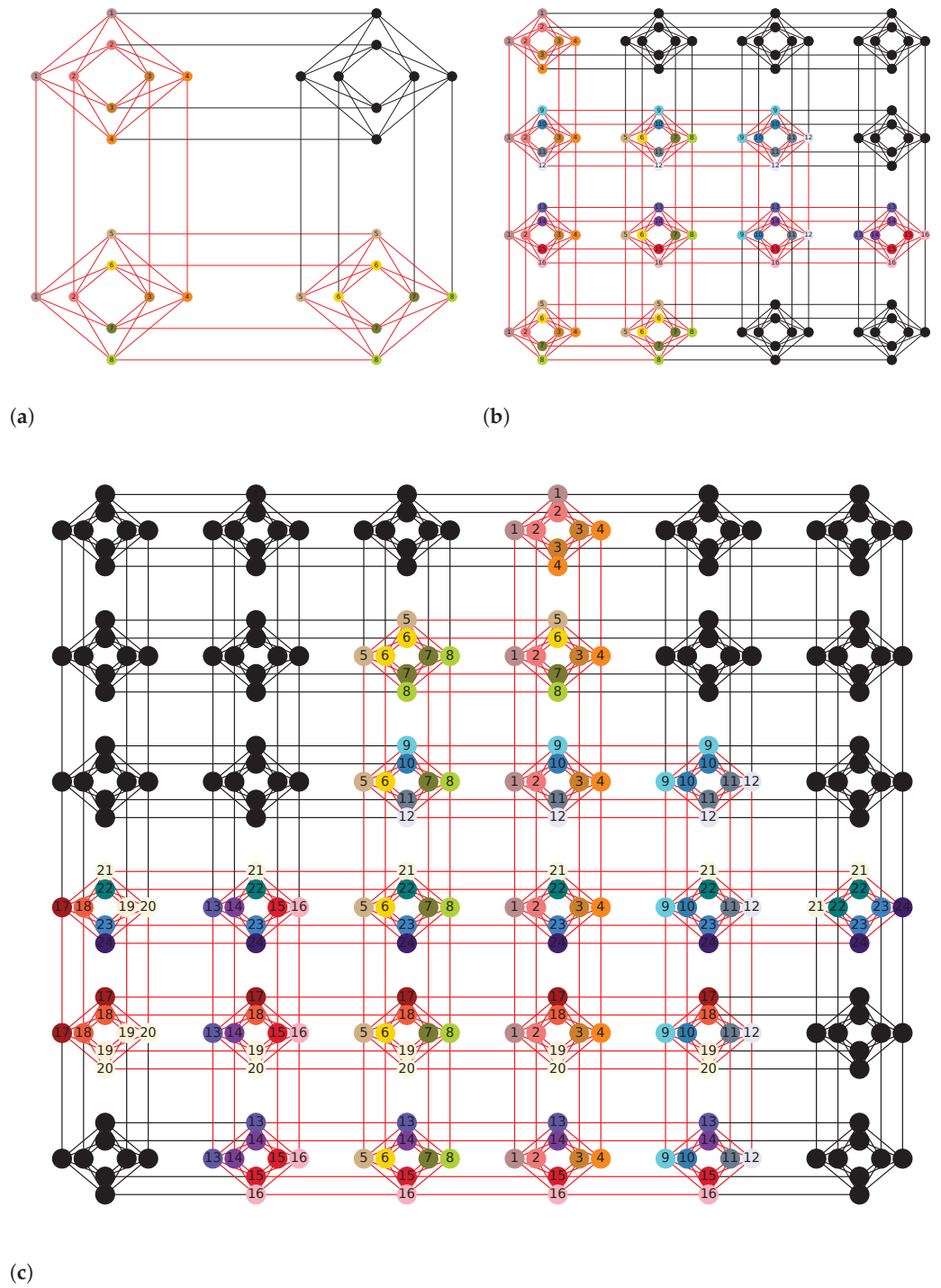


Figure 3. Cont.

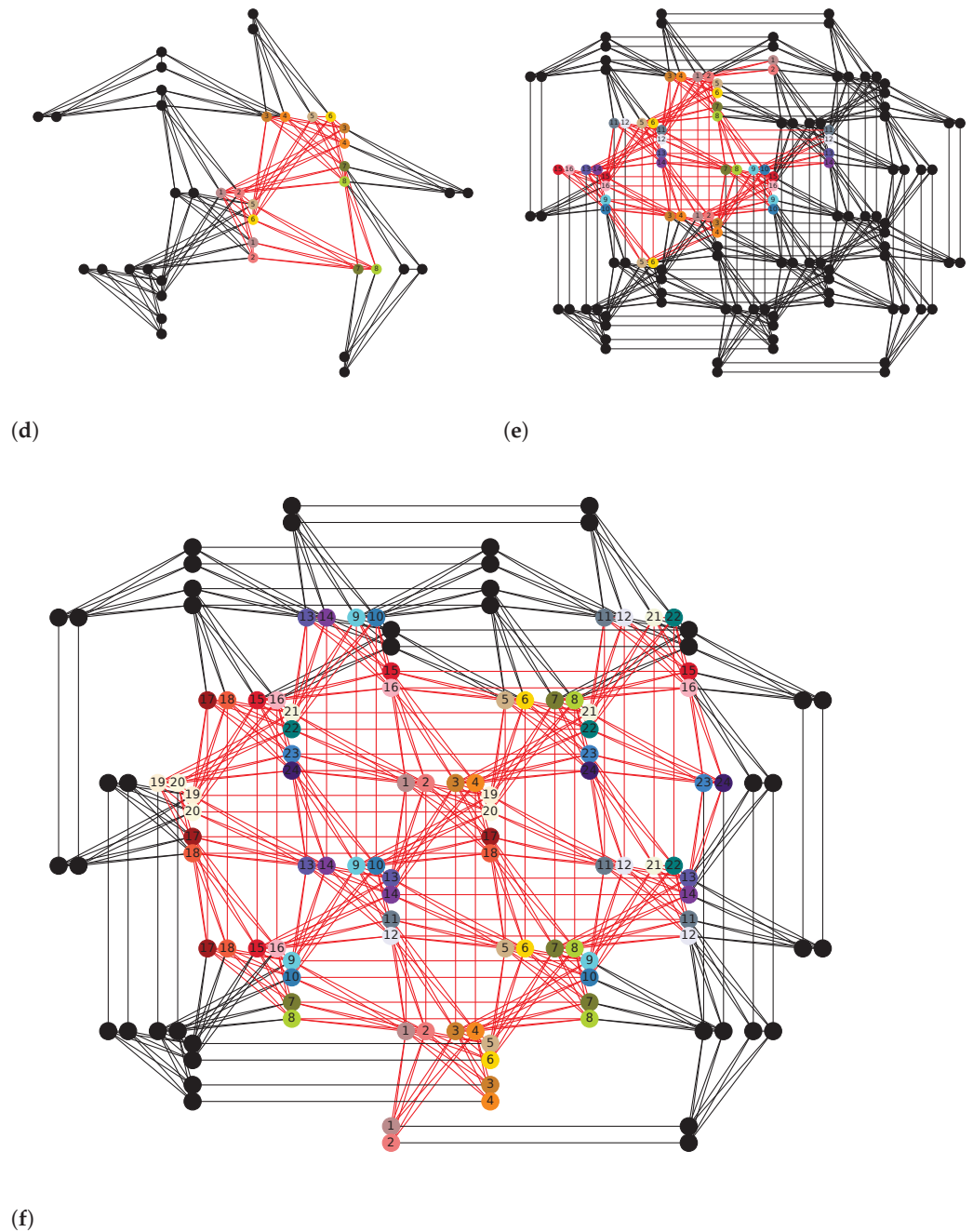


Figure 3. A comparison between minor embeddings on Chimera (2000Q) and Pegasus (Advantage) lattices of D-Wave processors for cliques (fully connected graphs) of different sizes. The vertices with the same color or label represent the same logical variable in Equation (7) and the chain length is defined as the number of qubits used to represent one logical variable. Each Chimera cell is a 4 by 4 complete bipartite graph ($K_{4,4}$) with 4 additional edges connecting neighboring cells. Each Pegasus cell has 24 qubits which include three $K_{4,4}$ graphs as in the Chimera cell and the cells are connected with each other using $K_{2,4}$ edges. To minor embed cliques of 8 vertices ($K = 8$), the chain length on the Chimera lattice is 3 while on the Pegasus lattice it is 2. With $K = 16$, the chain lengths are 5 and 2–3, respectively, and with $K = 24$, they are 7 and 3–4, respectively. This shows that Pegasus processor scales better for larger clique problems, which may lead to better performance. (a) Embedding $K_{8,8}$ on Chimera topology; (b) Embedding $K_{16,16}$ on Chimera topology; (c) Embedding $K_{24,24}$ on Chimera topology; (d) Embedding $K_{8,8}$ on Pegasus topology; (e) Embedding $K_{16,16}$ on Pegasus topology; (f) Embedding $K_{24,24}$ on Pegasus topology.

We can control a range of annealer parameters that may impact the solution quality in various degrees. Specifically, we set the number of spin reversal transforms [53] to 100 and readout thermalization to $100\ \mu\text{s}$ as suggested in [54,55]. The spin reversal transform flips the signs of 100 variables and coefficients of the Ising model, which leaves the ground state invariant. The goal is to average out the system errors thus improving the quality of the solutions [53]. $100\ \mu\text{s}$ readout thermalization allows the system enough time to cool back to the base temperature after each anneal. We set the annealing time at $1\ \mu\text{s}$ as longer annealing time sees no statistically significant improvements to the solutions similarly reported in [15]. Results from 2000Q and Advantage processors are both included in the following sections. Additionally, we report the results from D-Wave’s post processing utility on 2000Q processors, which decomposes the underlying graph induced by the QUBO into several low tree-width subgraphs [56], and then solves them exactly using belief propagation on junction trees [57].

We sample all QUBOs 30,000 times with both D-Wave backends and report the samples with the lowest objective value from Equation (17) each time. Figure 4 shows an example distribution of the samples.

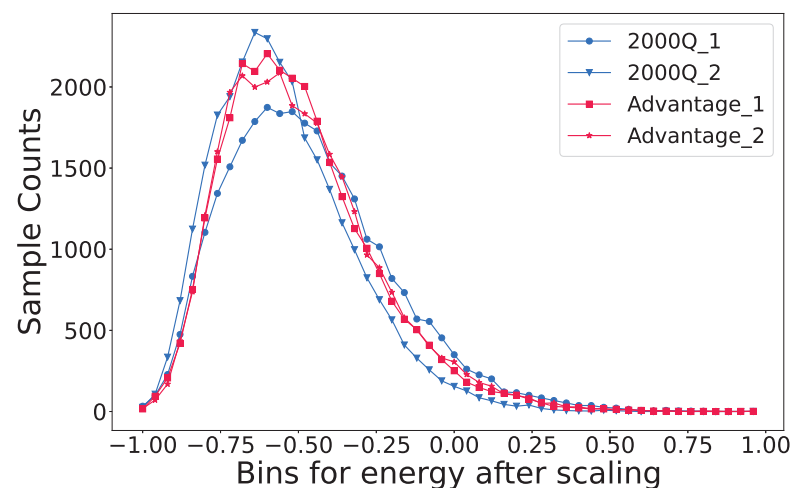


Figure 4. The distribution of the samples for 4 different QUBOs with both D-Wave backends. Each QUBO is sampled 30,000 times and the objective values of the samples is scaled to be between $(-1, 1)$. We divide the objective value range into 50 equally-spaced bins and count the number of samples in each bin. All four samples exhibit the Poisson distribution, and thus we only report the samples with the lowest objective value for the experiments in this section since they can be reproduced reliably.

4.3. Embedding Comparison on D-Wave Annealers

As discussed in Section 4.1, D-Wave quantum annealers require the problems to be minor embedded to the Chimera or Pegasus topology. For small problems this means there may be multiple valid embeddings and in this section we will measure how different embeddings can make an impact on the solution quality.

We compute four different embeddings that use different sets of physical qubits from both 2000Q and Advantage processors. Otherwise, the embedding graphs are the same, and hence they use the same number of qubits and chain lengths. We sample the same QUBO—first iteration of Algorithm 1 on the ETFs from December 19 to May 20—10 times with 10,000 samples each. We then pick the best solutions in terms of QUBO objective value from all 10 sample sets for each embedding and obtain their average and minimum values. Tables 1 and 2 report the results as ratios against the best objective values computed by simulated annealing for better readability. Since the objective values are negative, we compute ratios of the magnitudes instead.

Table 1. Embedding comparison on the 2000Q processor with 30 logical variables or 270 physical qubits after minor embedding. The objective value is calculated from Equation (18) and is normalized against the simulated annealer solving the same QUBO. All energies computed are negative, and their respective magnitudes are used for the comparison. The second embedding out of these four is able to find the solution with the better average and best objective value.

Embedding	Average Objective	Best Objective
1	95.66%	98.78%
2	96.83%	99.66%
3	96.53%	98.37%
4	96.21%	98.49%

Table 2. Embedding comparison on the Advantage processor with 30 logical variables or 134 physical qubits after minor embedding. Different embeddings on the Advantage processor show no statistically significant differences.

Embedding	Average Objective	Best Objective
1	99.25%	99.89%
2	99.52%	99.94%
3	99.22%	99.95%
4	99.48%	99.89%

We can see from Tables 1 and 2 that the impact that different embeddings make is statistically insignificant. However, it is clear that the Advantage processors have higher ratios than the 2000Q processors, which we will address next.

4.4. Annealing Results Comparison

We benchmark our algorithm on both simulated and quantum annealers using, as the baseline algorithm, a classical optimization solver, namely, *cvxpy* [58]. We create five ETF test datasets and four currency test datasets from 100 days of return data with different starting dates from 2010 to 2020.

The results are normalized against the optimal classical solution. The quantum algorithm fails to converge for the first two currency tests on the 2000Q processor, and the corresponding bars are missing in Figures 5 and 6.

In Figures 5 and 6, we used $k = 5$ binary variables to represent each asset weight. The simulated annealing results follow the optimal solutions closely in most tests. We note that in tests 2 and 5 from the ETF tests and tests 1, 2 and 3 from Currency tests, simulated annealing, and in some cases, quantum annealing produce portfolios of higher returns than those of the exact classical quadratic optimization problem solver. This is due to how Markowitz Optimization problems are formulated as QUBOs with discretized variables in Equation (18), which changes the optimization problem slightly, and also the optimal asset allocations. In test 4 from Currency tests, quantum annealers are able to find a portfolio with higher returns than simulated annealing as it returns a portfolio with a slightly increased risk that is still acceptable, but higher returns. This is not optimal in terms of QUBO objective values as the constraint penalty is now higher, yet the solution is still feasible. We also observe that the currency tests generally perform better than ETF tests on both quantum annealer backends. Figure 7 shows how quantum annealers perform with respect to the average of absolute correlation coefficients over all pairs of assets in each test. Higher correlation coefficients seem to lead to higher returns.

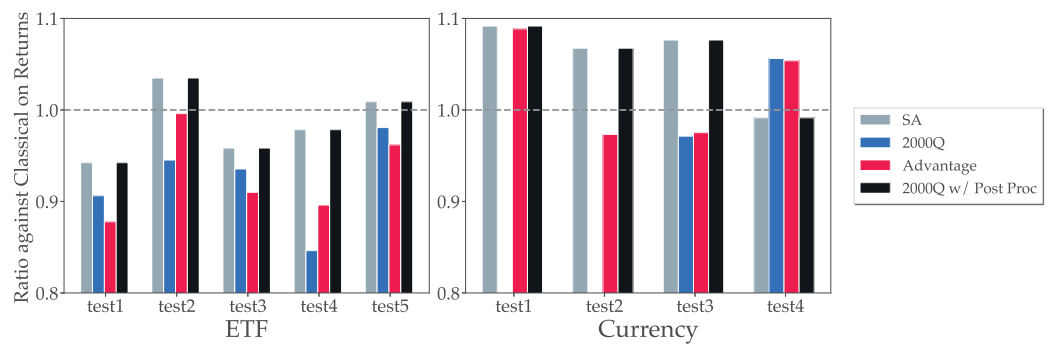


Figure 5. The comparison of the final returns between all four backends. A higher ratio means the backend can return portfolios with higher returns. Each test uses 100 days of return data with different starting dates from 2010 to 2020. The results from 2000Q with post processing yields identical results from simulated annealing. Both 2000Q and Advantage processors are able to compute returns that are consistently more than 80% of the optimal, except the two currency test cases where the algorithm fails to converge on the 2000Q.

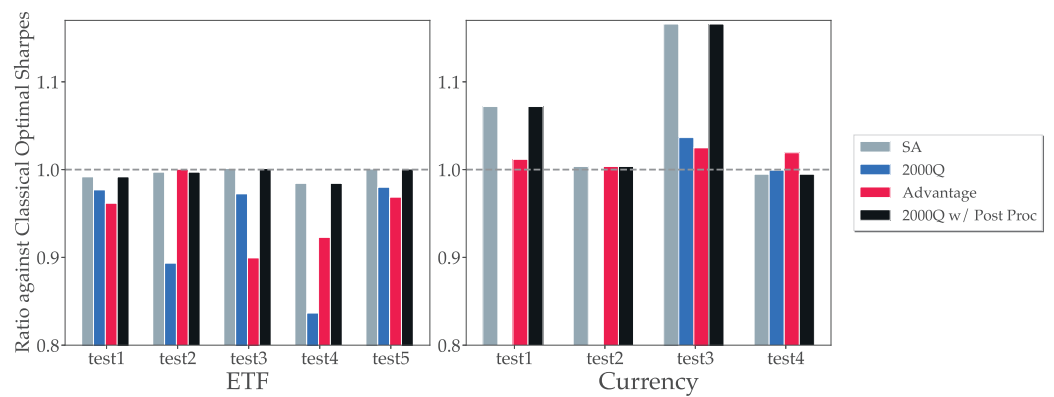


Figure 6. The comparison of the final Sharpe ratios between all four backends. Recall that the Sharpe ratio is the ratio of the return to the standard deviation of an asset for a set time period. Given a portfolio defined by the weight vector w , the Sharpe ratio of this portfolio is calculated as $\frac{\mu^T w}{\sqrt{w^T C w}}$. A higher ratio means the backend can return portfolios with higher Sharpe ratios. The results confirm that the portfolio variances returned by the quantum processors are close to the optimal results obtained from classical optimization methods, and it is effective to solve standard constrained optimization problems as a QUBO.

Although we acknowledge there may be other factors contributing to our observations that currency tests do better than ETF tests on for quantum annealers, Figure 7 implies that more correlated assets tend to perform better. Detailed analysis on which attributes of the assets have an impact on the quantum annealing performance and how much the impacts are requires more research in the future. Ref. [59] used machine learning models such as decision tree and regression to predict the accuracy of D-Wave’s quantum annealer on maximum clique problems.

4.5. State-of-the-Art on D-Wave Annealers

The embeddings of the six asset tests on both 2000Q and Advantage processors leave plenty of unused qubits. D-Wave’s clique embedding algorithm [52] suggests that we can embed fully connected graphs with 64 and 180 vertices to full-yield 2000Q and Advantage processors, respectively. Due to the defective qubits and connectors in the currently available Advantage processor, experimentally, we can embed only up to 119 qubits. This means we can solve portfolio optimization problems with 12 and 23 assets natively on 2000Q and Advantage processors, respectively.

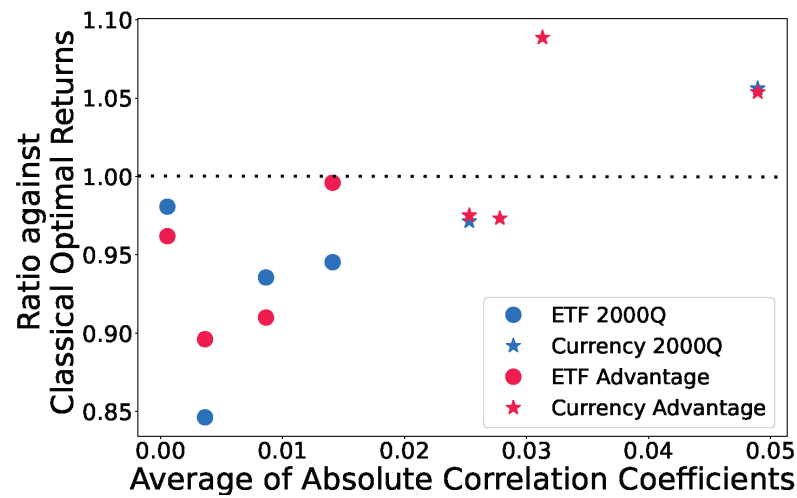


Figure 7. Final returns obtained from both quantum annealers against the average of the absolute correlation coefficients. The x-axis are the correlation coefficients of all N assets with each other computed using its daily returns from the chosen time periods and the y-axis is the ratio of the final returns against the classical optimal after Algorithm 1 converges using quantum annealers similar to Figure 5. The currency assets (stars) used in the tests all have higher correlation coefficients than those of the ETF assets, and generally yield better results.

On the 12 asset test shown in Figure 8, the 2000Q processor struggles to find the ground state as its embedding chain length reaches 16, while the Advantage processor provides results close to the simulated annealing and post-processed results. However, neither quantum annealer converges. Table 3 records the QUBO objective values of the last five iterations for the Advantage processor in this test. Although the objective values hardly differ, the solution quality is seemingly more sensitive to changes in the QUBO objective value for larger problems. A 0.1% change in the objective value leads to 30% difference in the portfolio variance. One potential reason is that larger problems have more assets that are less correlated, and as shown in Figure 7, smaller correlation coefficients generally equate to worse performance on quantum annealers. In this case, either the quantum annealers need to be more accurate to find the ground state, or our QUBO setup needs to be modified to account for higher asset counts.

Table 3. Objective values of last five iterations from simulated annealing and Advantage from the 12 asset test. This corroborates observations in Figure 8 that the Advantage processor is able to reach states with very good approximation ratios.

Last k Iteration	SA Objective	Advantage Objective	Difference
5	−1.026	−1.016	1.039%
4	−0.951	−0.950	0.092%
3	−0.879	−0.878	0.111%
2	−0.811	−0.809	0.170%
1	−0.746	−0.746	0.076%

For even larger problems of 23 assets, with the embedding chain lengths going up to 17, the Advantage processor fails to find the ground state by a large margin, as shown in Figure 9. Even though we can physically map a problem of this size, the results reflect the limitations of current-generation quantum annealers.

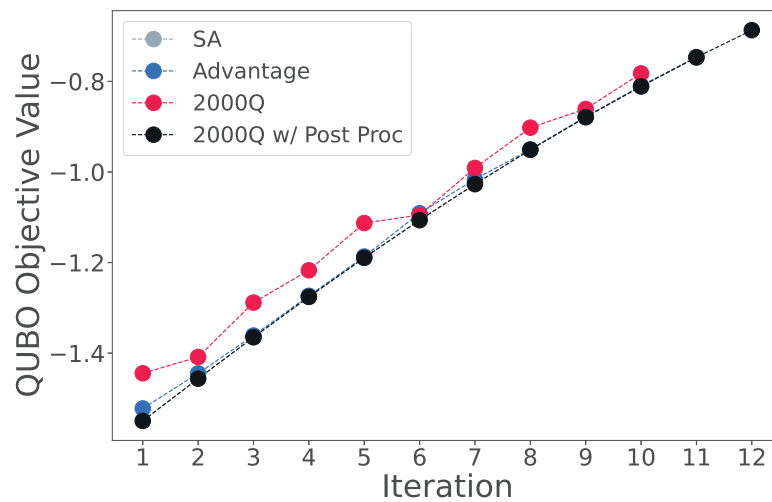


Figure 8. The objective comparison of the 12 asset test between all four backends. The solutions from 2000Q deviate from the ground states by a large margin, while the Advantage processor is able to keep up closely. Post-processing is able to improve the 2000Q results to once again match simulated annealing.

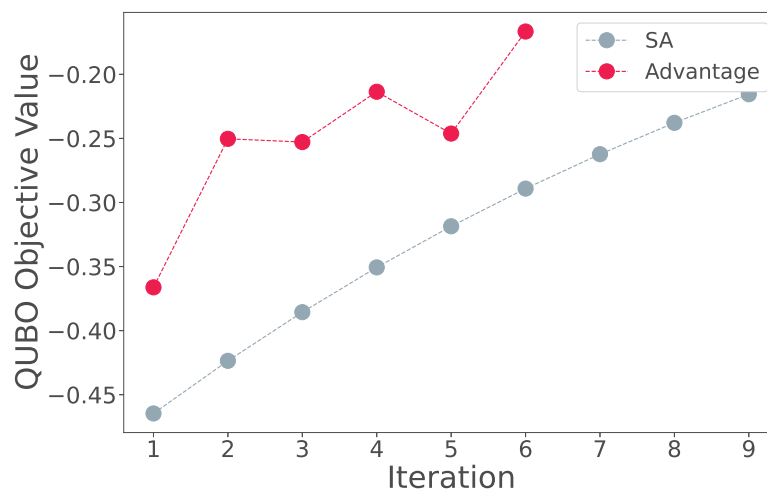


Figure 9. The objective comparison of the 23 asset test between simulated annealing and the Advantage processor. Due to the high chain lengths of the embedding, the Advantage processor fails to either reach the ground state or get close to it in all iterations, rendering the processor incapable of solving problems of such sizes.

5. Discussion

As newer quantum devices are released every year, it is important to design and benchmark algorithms across generations. As companies and researchers race to build the first quantum computer that can demonstrate quantum advantage on practical problems, different classes of quantum devices have emerged: general purpose quantum computers from IBM, Google, Honeywell, and others; the specialized quantum Ising machine from D-Wave; and quantum-inspired digital annealer from Fujitsu. These devices have different types of constraints due to different noise profiles, qubit connectivity, and/or implementable Hamiltonians, and none are perhaps at the scale and reliability needed to solve real-world problems at the edge of classical capability. Therefore, hybrid algorithms are needed to incorporate these quantum computers on practical problems with reasonable size.

In this paper, we have shown that it is not only possible to introduce such hybrid algorithm schemes that compute the optimal portfolios based on expected shortfall, but also highlighted where it is possible to reach working accuracy. We used a quantum annealer

to solve an asset allocation problem based on expected shortfall, employing a QUBO formulation of the Markowitz Optimization problem and interlacing it with a layer of classical decision-making. Here, we iteratively adjusted our problem Hamiltonian based on its feedback until the portfolio was within the desired risk threshold. The fact that both D-Wave 2000Q and Advantage quantum annealers performed reasonably well on the six-asset tests with portfolios' Sharpe ratios to above 80% of SA values is promising. Additionally encouraging is that the newer and more scalable Advantage processor achieved much better QUBO objective values on problems with 12 assets. Finally, we observed that both quantum annealers tended to obtain portfolios with higher returns on more correlated assets (Figure 7), which we believe should attract future research as it may help guide the application of quantum annealing on real-world applications in the near term.

Although the quantum annealers fell short on tests with more assets, we can remain optimistic about new hardware with more qubits, better connectivity, and lower noise in the near future. We also acknowledge the need to design algorithms that can scale with these new hardware, as we saw that the portfolio quality became increasingly sensitive to the QUBO objective values as we introduced more assets—results with 99.9% objective values of the optimal led to 30% more variance. Additionally, advances in gate-model quantum computers and combinatorial optimization algorithms [60,61] will provide other avenues for solving these problems. For example, it could be instructive to explore and compare to novel approaches, such as counterdiabatic techniques recently proposed for similar problems, but for gate-based systems [49].

Future research includes identifying subsets of problems that can be solved better on quantum devices, as we have discussed in Section 4. It is also important to find an efficient way to implement inequality constraints, as adding slack variables may not be the best choice in the QUBO. We also note that on specific test cases, the QUBO reformulation enables both simulated annealer and quantum annealers to find better portfolios than a classical convex optimizer *cvxpy*, by treating the constraints as soft. Other optimization problems might also benefit from QUBOs with soft constraints.

Author Contributions: Conceptualization, H.X., S.D. and A.B.; methodology, H.X. and S.D.; software, H.X.; validation, S.D., A.B. and A.P.; formal analysis, H.X., S.D., A.B. and A.P.; investigation, H.X.; resources, A.B.; data curation, H.X.; writing—original draft preparation, H.X.; writing—review and editing, H.X., S.D., A.B. and A.P.; visualization, H.X.; supervision, A.B. and A.P.; project administration, A.B. and A.P.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: Funding for S.D. was supported in part by the US Department of Energy, Advanced Scientific Computing Research program office Quantum Algorithms Team project ERKJ335, through subcontract number 4000175762 to Purdue University from Oak Ridge National Laboratory managed by UT-Battelle, LLC acting under contract DE-AC05-00OR22725 with the Department of Energy. A.B. was supported by Purdue University, College of Science, Startup funds, and H.X. was supported by College of Science, Quantum Seed Grant. The access to D-Wave for this research was funded by the U.S. Department of Energy under Contract No. DE-AC05-00OR22725 through the Oak Ridge Leadership Computing Facility (OLCF) at the Oak Ridge National Laboratory (ORNL).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data is available upon reasonable request.

Acknowledgments: We thank Travis Humble for overall suggestions and support for the project, and also Andrew King, Isil Ozfidan, and Erica Grant for helpful discussions. A.B. and S.D. thank the ORNL Quantum Computing Institute and the Purdue Quantum Science and Engineering Institute for their support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Glover, F.; Kochenberger, G.; Du, Y. A Tutorial on Formulating and Using QUBO Models. *arXiv* **2019**, arXiv:1811.11538.
2. Pastorello, D.; Blanzieri, E. Quantum Annealing Learning Search for Solving QUBO Problems. *Quantum Inf. Process.* **2019**, *18*, 303. [CrossRef]
3. Kochenberger, G.; Hao, J.K.; Glover, F.; Lewis, M.; Lü, Z.; Wang, H.; Wang, Y. The Unconstrained Binary Quadratic Programming Problem: A Survey. *J. Comb. Optim.* **2014**, *28*, 58–81. [CrossRef]
4. Lucas, A. Ising Formulations of Many NP Problems. *Front. Phys.* **2014**, *2*, 5. [CrossRef]
5. Djidjev, H.N.; Chapuis, G.; Hahn, G.; Rizk, G. Efficient Combinatorial Optimization Using Quantum Annealing. *arXiv* **2018**, arXiv:1801.08653.
6. Ikeda, K.; Nakamura, Y.; Humble, T.S. Application of Quantum Annealing to Nurse Scheduling Problem. *Sci. Rep.* **2019**, *9*, 12837. [CrossRef] [PubMed]
7. Titiloye, O.; Crispin, A. Quantum Annealing of the Graph Coloring Problem. *Discret. Optim.* **2011**, *8*, 376–384. [CrossRef]
8. Yarkoni, S.; Raponi, E.; Bäck, T.; Schmitt, S. Quantum Annealing for Industry Applications: Introduction and Review. *Rep. Prog. Phys.* **2022**, *85*, 104001. [CrossRef]
9. Markowitz, H. Portfolio Selection. *J. Financ.* **1952**, *7*, 77–91. [CrossRef]
10. Black, F.; Litterman, R.B. Asset Allocation: Combining Investor Views with Market Equilibrium. *J. Fixed Income* **1991**, *1*, 7–18. [CrossRef]
11. McNeil, A.J.; Frey, R.; Embrechts, P. *Quantitative Risk Management: Concepts, Techniques and Tools—Revised Edition*; Princeton University Press: Princeton, NJ, USA, 2015.
12. Dasgupta, S.; Banerjee, A. Quantum Annealing Algorithm for Expected Shortfall Based Dynamic Asset Allocation. *arXiv* **2020**, arXiv:1909.12904.
13. Pardalos, P.M.; Vavasis, S.A. Quadratic Programming with One Negative Eigenvalue Is NP-hard. *J. Glob. Optim.* **1991**, *1*, 15–22. [CrossRef]
14. Vavasis, S.A. Complexity Theory: Quadratic Programming. In *Encyclopedia of Optimization*; Floudas, C.A., Pardalos, P.M., Eds.; Springer US: Boston, MA, USA, 2001; pp. 304–307. [CrossRef]
15. Grant, E.; Humble, T.S.; Stump, B. Benchmarking Quantum Annealing Controls with Portfolio Optimization. *Phys. Rev. Appl.* **2021**, *15*, 014012. [CrossRef]
16. O'Donnell, R. *Analysis of Boolean Functions*; Cambridge University Press: Cambridge, UK, 2014. [CrossRef]
17. Dattani, N. Quadraticization in Discrete Optimization and Quantum Mechanics. *arXiv* **2019**, arXiv:1901.04405.
18. Verma, A.; Lewis, M.; Kochenberger, G. Efficient QUBO Transformation for Higher Degree Pseudo Boolean Functions. *arXiv* **2021**, arXiv:2107.11695.
19. Mandal, A.; Roy, A.; Upadhyay, S.; Ushijima-Mwesigwa, H. Compressed Quadraticization of Higher Order Binary Optimization Problems. In Proceedings of the 2020 Data Compression Conference (DCC), Snowbird, UT, USA, 24–27 March 2020; pp. 383–383. [CrossRef]
20. Yahoo Finance iShares MSCI Emerging Markets ETF (EEM). Available online: <https://finance.yahoo.com/quote/EEM/history?p=EEM> (accessed on 18 May 2020).
21. Yahoo Finance Invesco QQQ Trust (QQQ). Available online: <https://finance.yahoo.com/quote/QQQ/history?p=QQQ> (accessed on 18 May 2020).
22. Yahoo Finance iShares Silver Trust (SLV). Available online: <https://finance.yahoo.com/quote/SLV/history?p=SLV> (accessed on 18 May 2020).
23. Yahoo Finance SPDR S&P 500 ETF Trust (SPY). Available online: <https://finance.yahoo.com/quote/SPY/history?p=SPY> (accessed on 18 May 2020).
24. Yahoo Finance ProShares UltraPro Short QQQ (SQQQ). Available online: <https://finance.yahoo.com/quote/SQQQ/history?p=SQQQ> (accessed on 18 May 2020).
25. Yahoo Finance Financial Select Sector SPDR Fund (XLF). Available online: <https://finance.yahoo.com/quote/XLF/history?p=XLF> (accessed on 18 May 2020).
26. Uryasev, S.; Rockafellar, R.T. Conditional Value-at-Risk: Optimization Approach. In *Stochastic Optimization: Algorithms and Applications*; Uryasev, S., Pardalos, P.M., Eds.; Applied Optimization, Springer US: Boston, MA, USA, 2001; pp. 411–435. [CrossRef]
27. Norton, M.; Khokhlov, V.; Uryasev, S. Calculating CVaR and bPOE for Common Probability Distributions with Application to Portfolio Optimization and Density Estimation. *Ann. Oper. Res.* **2021**, *299*, 1281–1315. [CrossRef]
28. Bertsimas, D.; Lauprete, G.J.; Samarov, A. Shortfall as a Risk Measure: Properties, Optimization and Applications. *J. Econ. Dyn. Control* **2004**, *28*, 1353–1381. [CrossRef]
29. Brooke, J.; Bitko, D.; F., T.; Rosenbaum.; Aeppli, G. Quantum Annealing of a Disordered Magnet. *Science* **1999**, *284*, 779–781. [CrossRef]
30. Santoro, G.E.; Martoňák, R.; Tosatti, E.; Car, R. Theory of Quantum Annealing of an Ising Spin Glass. *Science* **2002**, *295*, 2427–2430. [CrossRef]
31. King, A.D.; Raymond, J.; Lanting, T.; Harris, R.; Zucca, A.; Altomare, F.; Berkley, A.J.; Boothby, K.; Ejtemaee, S.; Enderud, C.; et al. Quantum Critical Dynamics in a 5000-Qubit Programmable Spin Glass. *arXiv* **2022**, arXiv:2207.13800.

32. Venegas-Andraca, S.E.; Cruz-Santos, W.; McGeoch, C.; Lanzagorta, M. A Cross-Disciplinary Introduction to Quantum Annealing-Based Algorithms. *Contemp. Phys.* **2018**, *59*, 174–197. [CrossRef]
33. Barahona, F. On the Computational Complexity of Ising Spin Glass Models. *J. Phys. A Math. Gen.* **1982**, *15*, 3241–3253. [CrossRef]
34. Zhang, Z. Computational Complexity of Spin-Glass Three-Dimensional (3D) Ising Model. *J. Mater. Sci. Technol.* **2020**, *44*, 116–120. [CrossRef]
35. Vinci, W.; Lidar, D.A. Non-Stoquastic Hamiltonians in Quantum Annealing via Geometric Phases. *Npj Quantum Inf.* **2017**, *3*, 1–6. [CrossRef]
36. Farhi, E.; Goldstone, J.; Gutmann, S.; Sipser, M. Quantum Computation by Adiabatic Evolution. *arXiv* **2000**, arXiv:quant-ph/0001106.
37. Albash, T.; Lidar, D.A. Adiabatic Quantum Computation. *Rev. Mod. Phys.* **2018**, *90*, 015002. [CrossRef]
38. Born, M.; Fock, V. Beweis des Adiabatenatzes. *Z. für Phys.* **1928**, *51*, 165–180. [CrossRef]
39. Inc., D.W.S. The Practical Quantum Computing Company. *ICE: Dynamic Ranges in h and J Values*; 2021.
40. Rosenberg, G.; Haghnegahdar, P.; Goddard, P.; Carr, P.; Wu, K.; de Prado, M.L. Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer. *IEEE J. Sel. Top. Signal Process.* **2016**, *10*, 1053–1060. [CrossRef]
41. Venturelli, D.; Kondratyev, A. Reverse Quantum Annealing Approach to Portfolio Optimization Problems. *Quantum Mach. Intell.* **2019**, *1*, 17–30. [CrossRef]
42. Phillipson, F.; Bhatia, H.S. Portfolio Optimisation Using the D-Wave Quantum Annealer. In *Proceedings of the Computational Science—ICCS 2021*; Lecture Notes in Computer Science; Paszynski, M.; Kranzlmüller, D., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloat, P.M.A., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 45–59. [CrossRef]
43. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092. [CrossRef]
44. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
45. Liu, Y.J.; Zhang, W.G. A Multi-Period Fuzzy Portfolio Optimization Model with Minimum Transaction Lots. *Eur. J. Oper. Res.* **2015**, *242*, 933–941. [CrossRef]
46. Vercher, E.; Bermúdez, J.D. Portfolio Optimization Using a Credibility Mean-Absolute Semi-Deviation Model. *Expert Syst. Appl.* **2015**, *42*, 7121–7131. [CrossRef]
47. Mansini, R.; Ogryczak, W.; Speranza, M.G. Twenty Years of Linear Programming Based Portfolio Optimization. *Eur. J. Oper. Res.* **2014**, *234*, 518–535. [CrossRef]
48. Schaerf, A. Local Search Techniques for Constrained Portfolio Selection Problems. *Comput. Econ.* **2002**, *20*, 177–190. [CrossRef]
49. Hegade, N.N.; Chandarana, P.; Paul, K.; Chen, X.; Albarrán-Arriagada, F.; Solano, E. Portfolio Optimization with Digitized-Counterdiabatic Quantum Algorithms. *Phys. Rev. Res.* **2022**, *4*, 043204. [CrossRef]
50. Inc., D.W.S. The Practical Quantum Computing Company. The D-Wave Advantage System: An Overview. 2021. Available online: https://www.dwavesys.com/media/s3qbjp3s/14-1049a-a_the_d-wave_advantage_system_an_overview.pdf (accessed on 18 May 2020).
51. Venturelli, D.; Mandrà, S.; Knysh, S.; O’Gorman, B.; Biswas, R.; Smelyanskiy, V. Quantum Optimization of Fully Connected Spin Glasses. *Phys. Rev. X* **2015**, *5*, 031040. [CrossRef]
52. Boothby, T.; King, A.D.; Roy, A. Fast Clique Minor Generation in Chimera Qubit Connectivity Graphs. *Quantum Inf. Process.* **2016**, *15*, 495–508. [CrossRef]
53. Pelofske, E.; Hahn, G.; Djidjev, H. Optimizing the Spin Reversal Transform on the D-Wave 2000Q. In *Proceedings of the 2019 IEEE International Conference on Rebooting Computing (ICRC)*, San Mateo, CA, USA, 6–8 November 2019; pp. 1–8. [CrossRef]
54. Lanting, T.; Amin, M.H.; Baron, C.; Babcock, M.; Boschee, J.; Boixo, S.; Smelyanskiy, V.N.; Foygel, M.; Petukhov, A.G. Probing Environmental Spin Polarization with Superconducting Flux Qubits. *arXiv* **2020**, arXiv:2003.14244.
55. Pudenz, K.L. Parameter Setting for Quantum Annealers. In *Proceedings of the 2016 IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, USA, 13–15 September 2016; pp. 1–6. [CrossRef]
56. Markowitz, H.M. The Elimination Form of the Inverse and Its Application to Linear Programming. *Manag. Sci.* **1957**, *3*, 255–269. [CrossRef]
57. Jensen, F.V.; Lauritzen, S.L.; Olesen, K.G. Bayesian Updating in Causal Probabilistic Networks by Local Computations. *Comput. Stat. Q.* **1990**, *4*, 269–282.
58. Diamond, S.; Boyd, S.P. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *J. Mach. Learn. Res.* **2016**, *17*, 2909–2913.
59. Barbosa, A.; Pelofske, E.; Hahn, G.; Djidjev, H.N. Using Machine Learning for Quantum Annealing Accuracy Prediction. *Algorithms* **2021**, *14*, 187. [CrossRef]
60. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv* **2014**, arXiv:1411.4028.
61. Hadfield, S.; Wang, Z.; O’Gorman, B.; Rieffel, E.G.; Venturelli, D.; Biswas, R. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms* **2019**, *12*, 34. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Quantum Annealing in the NISQ Era: Railway Conflict Management

Krzysztof Domino ^{1,*}, Máttyás Koniorczyk ², Krzysztof Krawiec ³, Konrad Jałowiecki ⁴, Sebastian Deffner ^{5,6} and Bartłomiej Gardas ¹

¹ Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland

² Wigner Research Centre, Konkoly-Thege M. út 29-33, H-1525 Budapest, Hungary

³ Faculty of Transport and Aviation Engineering, Silesian University of Technology, 40-019 Katowice, Poland

⁴ Institute of Physics, University of Silesia, 41-500 Chorzów, Poland

⁵ Department of Physics, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

⁶ Instituto de Física ‘Gleb Wataghin’, Universidade Estadual de Campinas, Campinas 13083-859, SP, Brazil

* Correspondence: kdomino@iitis.pl

Abstract: We are in the noisy intermediate-scale quantum (NISQ) devices’ era, in which quantum hardware has become available for application in real-world problems. However, demonstrations of the usefulness of such NISQ devices are still rare. In this work, we consider a practical railway dispatching problem: delay and conflict management on single-track railway lines. We examine the train dispatching consequences of the arrival of an already delayed train to a given network segment. This problem is computationally hard and needs to be solved almost in real time. We introduce a quadratic unconstrained binary optimization (QUBO) model of this problem, which is compatible with the emerging quantum annealing technology. The model’s instances can be executed on present-day quantum annealers. As a proof-of-concept, we solve selected real-life problems from the Polish railway network using D-Wave quantum annealers. As a reference, we also provide solutions calculated with classical methods, including the conventional solution of a linear integer version of the model as well as the solution of the QUBO model using a tensor network-based algorithm. Our preliminary results illustrate the degree of difficulty of real-life railway instances for the current quantum annealing technology. Moreover, our analysis shows that the new generation of quantum annealers (the advantage system) does not perform well on those instances, either.

Keywords: railway dispatching problem; quadratic unconstrained binary optimization (QUBO); quantum annealing

Citation: Domino, K.; Koniorczyk, M.; Krawiec, K.; Jałowiecki, K.; Deffner, S.; Gardas, B. Quantum Annealing in the NISQ Era: Railway Conflict Management. *Entropy* **2023**, *25*, 191. <https://doi.org/10.3390/e25020191>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 30 November 2022

Revised: 11 January 2023

Accepted: 15 January 2023

Published: 18 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Concentrated efforts all around the globe [1–5] are pursuing the development of viable quantum technologies. However, the technological challenges are immense, and it may still take some time before the first fault-tolerant quantum computers may become available for practical applications [6]. Thus, it is of instrumental importance to not only build a quantum literate workforce [7] but also ensure investments are made in realistic and societally beneficial avenues for development [8].

Despite the fact that the first demonstrations of quantum advantage have been published [9], currently available hardware is still prone to noise. Thus, it has been argued that we are in the era of noisy-intermediate scale quantum (NISQ) technologies [10]. For instance, the D-Wave quantum annealer promises to deliver scalability beyond current classical hardware limitations. However, exploiting NISQ technologies often requires a different mathematical modeling framework. The D-Wave quantum annealer accepts an Ising spin-glass instance, possibly in the form of a quadratic unconstrained binary optimization (QUBO) problem equivalent to it, as its input and outputs solutions encoded in

spin configurations. High-quality solutions are expected to be computed by these devices in a reasonable time, even for problems of the size which already bear practical relevance (currently, up to 5000 variables on a sparse graph [11]). More importantly, an NISQ computer may not (yet) be able to outperform classical computers; however, seeking and demonstrating amendable applications provides the instrumental guiding principle for the development of purpose-specific devices with genuine quantum advantage.

To date, at least in public domain research, most of the studied “quantum” problems are not directly relevant to a particular industrial application but rather concern the solution of “classical” generic, computationally hard problems, such as, e.g., the traveling salesman problem or the graph coloring problem [12]; see, e.g., Ref. [13] for a comprehensive review. The present work belongs to a more practically motivated line of research: it is dedicated to making quantum computing more broadly accessible by demonstrating its applicability to a relevant problem from a field not directly related to physics: conflict management in railway operation. Railway operations involve a broad range of scheduling activities, ranging from provisional timetable planning over rolling stock circulation planning, crew scheduling and rostering, etc. to operational train dispatching in case of disturbances, such as, e.g., severe weather, unplanned events, and technological breakdown. Many of these tasks require solving computationally expensive and overall challenging combinatorial problems. Various consequences of improper planning, e.g., incorrect dispatching decisions can be severe in terms of resources (e.g., time costs, passengers’ satisfaction, financial loss).

In the domain of transportation research, the applicability of quantum annealing has only been demonstrated for very few problems. For instance, Stollenwerk et al. [14] recently addressed a class of simplified air traffic management problems of strategic conflict resolution. Their preliminary results show that some challenging problems can be solved efficiently with the D-Wave 2000Q machine, see Figure 1. As for the railway industry, to the best of our knowledge, a preliminary version of the present work [15] was the first to apply a quantum computing approach to a problem in railway optimization. As the citations to our e-print illustrate [16,17], this research direction is attracting increasing interest.

The main purpose of the present paper is to elucidate how railway management problems can be solved with the currently available hardware. Naturally, we do not expect the current generation of the D-Wave annealer to outperform the best available classical algorithms. Rather, the present work is of pedagogical and instructive value as it provides an entry point for transportation research into quantum computing and demonstrates the applicability of an NISQ computer. To this end, we solve the delay and conflict management on an existing Polish railway whose real-time solution is of paramount importance for the local community.

Realizing that especially in the NISQ era, there is still a significant language barrier between foundational quantum physics and real-life applications, the present paper strives to be as introductory and self-contained as possible. In particular, Section 2 provides a brief review of railway conflict management as well as quantum annealing. Our model of the “real” problem is then outlined in Section 3 before we discuss our findings in Section 4. The discussion is concluded with a few remarks on future research directions in Section 5. In the Supplemental Materials, we give a fully detailed description of our model. We include there also the description of a possible linear integer programming formulation that we use for comparison. The Supplemental Materials contains a number of additional particular instances and their solutions.

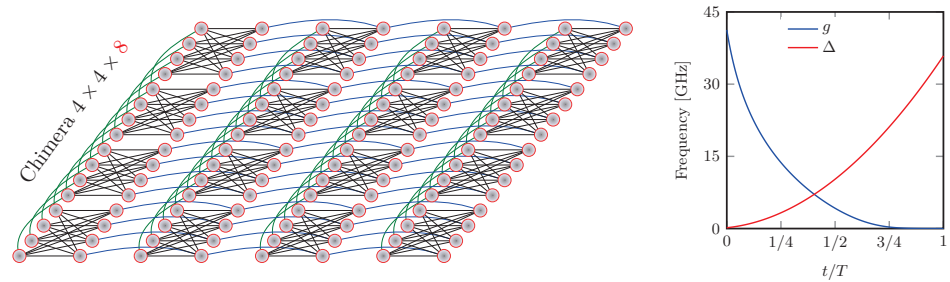


Figure 1. D-Wave processor specification. Left: An example of the Chimera topology, here composed of 4×4 (C_4) grid consisting of clusters (units cells) of 8 qubits each. The total number of variables (vertices) for this graph is $N = 4 \cdot 4 \cdot 8 = 128$. A graph’s edges indicate possible interactions between qubits. The maximum number of qubits is $N_{\max} = 2048$ for the Chimera C_{16} topology, whereas the total number of connections between them is limited to $6000 \ll N_{\max}^2$. Right: A typical annealing schedule controlling the evolution of a quantum processor, where T denotes the time to complete one annealing cycle (the annealing time). It ranges from microseconds ($\sim 2 \mu\text{s}$) to milliseconds ($\sim 2000 \mu\text{s}$). The parameters g and Δ are used in Equation (6).

2. Railway Dispatching Problem on Single-Track Lines

Railway dispatching problem management is a complex area of transportation research. Here, we focus on the delay management on single-track railways. This problem concerns the operative modifications of train paths upon disturbances in railway traffic. Incorrect decisions may cause the dispatching situation to deteriorate further by propagating the delay, resulting in unforeseeable consequences. Henceforth, we discuss this problem’s details and survey the relevant part of the literature. Although we focus on single-track railway lines, some considerations may also be applicable to multi-track railways [18].

2.1. Problem Description

Consider a part of a railway network in which the traffic is affected by a disturbance. As a result, some trains cannot run according to the original timetable. Hence, a new, feasible timetable should be created promptly, minimizing unwanted consequences of the delay. To be more specific, we are given a part of a railway network (referred to as the *network*), such as, e.g., those depicted in Figure 2a,b. The *network* is divided into *block sections* or *blocks* (This term originated in the railway signaling terminology. In general, it refers to a section of the railway line between two signal boxes.): sections which can be occupied by at most one train at a time. The block sections are labeled with numbers in the figures. We focus on single-track railway lines. These include *passing sidings* (referred to as *sidings*): parallel tracks, typically at stations; the *blocks* are labeled with upper indices in parentheses in the figures. Via the sidings, trains heading in opposite directions can meet and pass, while trains heading in the same direction can meet and overtake.

All trains run according to a *timetable*. Examples of timetables are illustrated in Figure 2c,d in the form of train diagrams, and they will be explained later in Section 4.1. The set of given time–location points of a given train are termed as the *train path*, which represents a train in a train diagram as points connected with straight lines. We assume that the initial timetable is *conflict free* and that it meets all feasibility criteria. The criteria may vary [19,20] depending on the railway network in question. The possible variants include technical requirements such as speed limits, dwell times, and other signaling-imposed requirements, as well as rolling stock circulation criteria and passenger demands for trains to meet. The railway delay management problem can be viewed from various perspectives, including that of a passenger, the infrastructure manager, or a transport operation company [19–21]. Here, we look at this problem from the perspective of the infrastructure manager, who is to make the ultimate decision about the modifications and is in the position to prioritize the requirements.

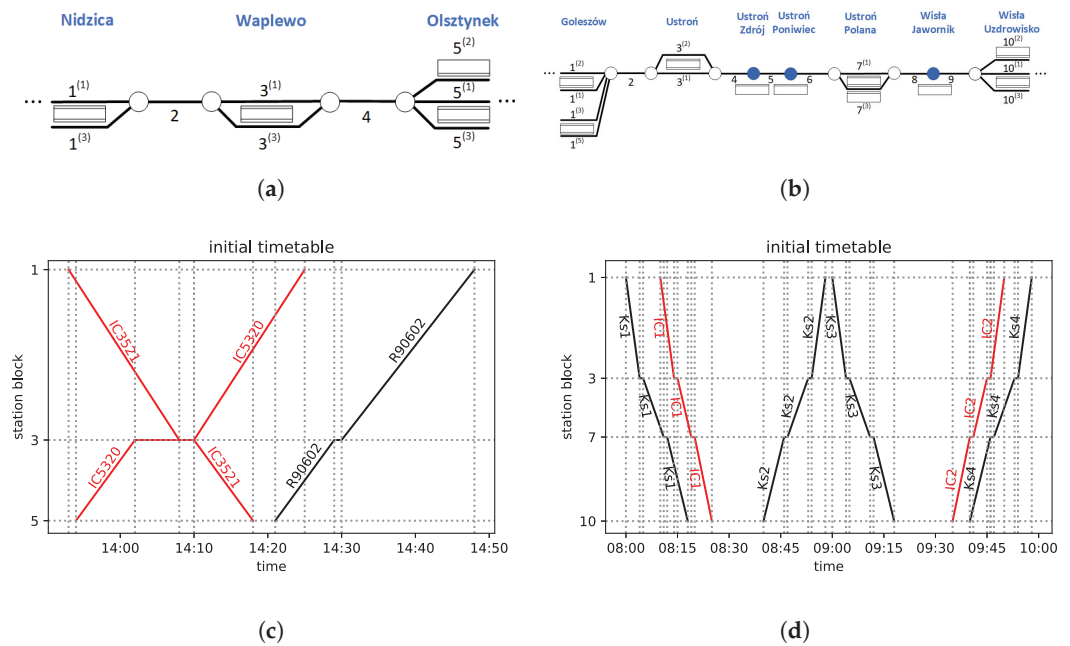


Figure 2. The railway line segments and their initial (undisturbed) timetables addressed in our calculations. The train diagrams in subfigures (c,d) represent train paths by connecting characteristic points of the location of trains at certain times by straight lines. Subfigures (a,b) represent the *network* topologies. The lines are the railway tracks. Their numbers represent blocks (as used, e.g., in the vertical axes of the train diagrams) and their upper indices in parentheses refer to the sidings (i.e., parallel tracks of stations). The rectangles represent the passenger platforms, circles represent the block boundaries (white: between a station and a line block, blue filled: between two line blocks). (a) Nidzica–Olsztynek section of railway line No. 216. (b) Golezów–Wiśła Uzdrowisko section of railway line No. 191. (c) Train diagram for the timetable of the line in (a). (d) Train diagram for the timetable of the line in (b).

In what follows, we assume that—for whatever reason—a delay occurs. Hence, some trains’ locations differ from the scheduled ones. A *conflict* is an inadmissible situation in which at least two trains are supposed to occupy the same block section. For instance, if an already delayed train would continue its trip according to the original plan shifted in time with the delay while the other trains would run according to the original timetable, multiple trains could meet in the same block, as illustrated in Figure 3a.

The objective of conflict management is thus to redesign the timetable to avoid conflicts (such as in Figure 3b in our example), and minimize delays. The overall delay of a train is the sum of two types of delays. A *primary delay* is caused by an initial disturbance directly, e.g., a particular train is delayed because of an engine breakdown. Such a delay cannot be avoided. Moreover, it has additional consequences as it propagates through the *network*. To separate the primary (unavoidable) part of the overall delay from the rest, which depends on dispatching decisions, the following consideration is commonly made. Obviously, given an actual location of trains, there is a minimal amount of time needed for each train to reach further destinations, e.g., due to speed limits, even when the train would not interact with any other train. The so-calculated delay is considered as the *primary delay*.

The delay of a train beyond the *primary delay* is termed as the secondary delay. These are induced by conflicts, i.e., interactions of trains, that have to be resolved by appropriate dispatching decisions. The objective of the optimization of these decisions is the minimization of a suitable function of secondary delays, e.g., their maximum or a weighted sum. Note that there are many other practically relevant options for the objective function [22], e.g., the total passenger delay or the cost of operations, and some of these are also in line with our approach.

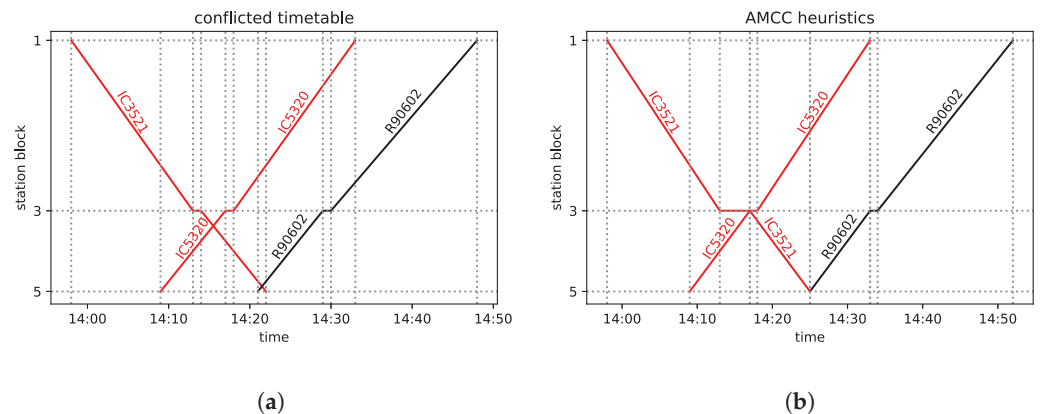


Figure 3. A possible solution of the conflict on line No. 216. (a) The conflicted diagram. All the three trains would meet in block 4 as it can be seen from the intersecting train paths. (b) The solution; FCFS, FLFS, and AMCC give the same outcome with a maximum secondary delay of 4 min.

The mathematical treatment of railway delay and conflict management leads to NP-hard problems (In computational complexity theory, NP-hardness, non-deterministic polynomial-time hardness, is the defining property of a class of problems that are informally “at least as hard as the hardest problems in NP, that is the class of problems that can be solved in polynomial time on a non-deterministic Turing machine [23]”). Certain simple variants are NP-complete [24]. It is broadly accepted that these problems are equivalent to job-shop models with blocking constraints [25], given the release and due dates of the jobs and depending on the requirements of the model and additional constraints such as recirculation or no-wait. The correspondence between the metaphors is the following. Trains are the jobs and block sections are the machines. Concerning the objective functions, the (weighted) total tardiness or make-span is typically addressed, which is the (weighted) sum of secondary delays or the minimum of the largest secondary delay in the railway context. So, with the standard notation of scheduling theory [26], our problem falls into the class $(J_m|r_i, d_i, block|\sum_j w_j T_j)$. Here, J_m stands for a job shop with multiple machines, r_i stands for the given release times, and d_i stands for the deadlines for each job, and *block* stands for the presence of blocking constraints (i.e., a job blocks a machine while it is processed). The objective, the third part of the triplet, is a total weighted tardiness.

2.2. Existing Algorithms

The following summary of railway dispatching and conflict management is focused on the works that are closely related to the problem addressed by us. A more comprehensive review of the huge literature on optimization methods applicable to railway management problems can be found in many related publications, notably in Refs. [20,22,27–31].

On a single-track line, the possible actions that can be taken to reschedule trains are the following: allocating new arrival and departure times, changing tracks and platforms, and reordering the trains by adjusting the meet-and-pass plans [20,22,32]. An important issue in modeling single-track lines is the handling of sidings (stations). A recent work of Lange and Werner [33] addressing the problem describes three approaches. In the *parallel machine approach*, it is assumed that each track within the siding corresponds to a separate machine in the job shop, thereby losing the possibility of flexible routing, i.e., changing track orders at a station afterward. In the *machine unit approach*, parallel tracks are treated as additional units of the same machine. Finally, in the *buffer approach*, the sidings at the same location are handled as buffers without internal structure and therefore not warranting the feasibility of track occupation planning at a station. We adopt the buffer approach in our model.

As to the nature of the decision variables, two major classes of models can be identified:

- *Order and precedence variables* prescribe the order in which a machine processes jobs, i.e., the order of trains passing a given block section in the railway dispatching problem on single-track lines.
- *Discrete time units*, in which the decision variables belong to discretized time instants; the binary variables describe whether the event happens at a given time.

These two approaches lead to different model structures, which are hard to compare. The *discrete time units* approach appears to be more suitable for a possible QUBO formulation, but it leads to many decision variables and thus worse scaling. On the other hand, the *order and precedence variables approach* can lead to a representation of the problem with alternative graphs [34,35], which is an intuitive picture. The solution of this problem representation leads to mixed-integer programs that can be solved, e.g., with iterative methods (such as branch-and-bound), but they are not ideal for a reformulation to QUBO. Time-indexed variables, on the other hand, can result in pure binary problems that are suitable for a transformation to QUBOs [36], so we follow the latter approach.

Returning to Ref. [33], the authors considered the problem adopting the *parallel machine approach* and the *machine unit approach with order and precedence variables*, addressing the problem $J_m|r_i, d_i, block, rcrc|\sum_j T_j$ in the standard notation of scheduling theory. In the case of the instances addressed in this reference, with 15 or more stations and 11 or more trains, the computational time of the presented classical algorithms is reported to be always higher than 10 min using CPLEX 12.6.1, IBM Armonk, New York, USA, which can be considered as a long time in a dispatching situation. These illustrate the limitations of the state-of-the-art classical algorithms.

In the present work, we will adopt slightly different constraints and objectives, namely, $J_m|r_i, d_i, block|\sum_j w_j T_j$. As to decision variables, we opt for discrete time units and time-indexed variables. (For the sake of completeness, in the Supplemental Materials, we demonstrate that the problem can also be encoded with precedence variables and handled by a linear solver.)

In Ref. [37], Zhou and Zhong considered the problem of timetabling on a single-track line. The starting times of trains and their stops are given, and a feasible schedule is to be designed to minimize the total running time of (typically passenger) trains. Although their problem, notably its objective function and the input, is different, the constraints are similar to those of our problem. The authors also deal with conflicts, dwell times, and minimum headway times for entering a segment of the railway line. They handle the problem with reference to resource-constrained project scheduling. Their decision variables are the discretized entry and leave times of the trains at the track segments, binary precedence variables describing the order of the trains passing a track segment, and time-indexed binary variables describing the occupancy of a segment by a given train at a given time. They introduce a branch-and-bound procedure with an efficiently calculable conflict-based bound in the bounding step to supplement the commonly used Lagrangian approach. They demonstrate its applicability to scheduling of up to 30 passenger trains for a 24-h periodic planning horizon on a line with 18 stations in China.

Harrod [38] proposed a discrete-time railway dispatching model, with a focus on conflict management. In this work, the train traffic flow is modeled as a directed hypergraph, with hyperarcs representing train moves with various speeds. This may be confined to an integer programming model with time-, train-, and hypergraph-related variables and a complex objective function covering multiple aspects. The model is demonstrated on an imaginary single-track line with long passing sidings at even-numbered block sections of up to 19 blocks in length. An intensive flow of trains at moderate speeds is examined. The model instances are solved with CPLEX in the order of 1000 s of computation time. As a practical application, a segment of a busy North American mainline is used, on which the model produced practically useful results. Bigger examples were also experimented with, leading to the conclusion that the approach is promising but that it needs more specialized technology than a standard mixed-integer programming (MIP) solver to be efficient.

Meng and Zhou [39] describe a simultaneous train rerouting and rescheduling model based on network cumulative flow variables. Their model also employs discrete-time-indexed variables. They implement a Lagrangian relaxation solution algorithm and make detailed experiments showing that their approach performs promisingly on a general n -track railway network. In the introduction of their article, they tabulate numerous timetabling and dispatching algorithms.

This brief survey of the extensive literature confirms that the problem of railway dispatching and conflict management is indeed a good candidate for demonstrating new computational technology capable of solving hard problems. Only very few models have been put into practice. The size and complexity of realistic dispatching problems make it challenging for the models to solve them with current computational technology.

2.3. Quantum Annealing and Related Methods

Let us now turn our attention to the main tools used in the present study: quantum annealing techniques. These have their roots in adiabatic quantum computing, which is a new computational paradigm [40] that, under additional assumptions, is equivalent [41] to the gate model of quantum computation [42] (provided that the specific interactions between quantum bits can be realized experimentally [43]). Thus, this emerging technology promises to tackle complicated (NP-hard in fact [44]) discrete optimization problems by encoding them in the ground state of a physical system: the Ising spin glass model [45]. Such a system is then allowed to reach its ground state “naturally” via an adiabatic-like process [46]. An ideal adiabatic quantum computer would in this way provide the exact optimum, whereas a *real* quantum annealer is a physical device that has noise and other inaccuracies. Hence, currently existing quantum annealers are paradigmatic examples of the NISQ era [10]. Their output is only a sample of candidates that is likely to contain the optimum. Quantum annealing can be therefore regarded as a heuristic approach, which will become increasingly accurate and efficient as the technology improves.

2.3.1. Ising-Based Solvers

The Ising model, introduced originally for the microscopic explanation of magnetism, has been in the center of the research interests of physicists ever since. It deals with a set of variables $s_i \in \{+1, -1\}$ (originally corresponding to the direction of microscopic magnetic momenta associated with spins). The configuration of N such variables is thus described by a vector $\mathbf{s} \in \{+1, -1\}^N$. The model then assigns an energy to a particular configuration:

$$E(\mathbf{s}) = \sum_{(i,j) \in E} J_{i,j} s_i s_j + \sum_{i \in V} h_i s_i, \quad (1)$$

where (V, E) is a graph whose vertices V represent the spins, the edges E define which spins interact, $J_{i,j}$ is the strength of this interaction, and h_i is an external magnetic field at spin i . Although the early studies of the model dealt with configurations in which the spins were arranged in a one-dimensional chain so that the coupling J was non-zero for nearest neighbors only, the model has been generalized in many ways, including the most general setting of an arbitrary (V, E) graph, i.e., incorporating the possibility of non-zero couplings for any i, j pair. Such a system is referred to as a spin glass in physics. For comprehensive reviews of generalizations of the Ising model, we refer to the literature [47,48].

From an operations research point of view, the physical model is interesting, since it describes a computational resource for optimization. The idea originates from the fact that in physics, the minimum energy configuration determines many properties of a material.

In mathematical programming, it is often more convenient to deal with 0–1 variables. By introducing new decision variables $\mathbf{x} \in \{0, 1\}$ so that

$$x_i = \frac{s_i + 1}{2}, \quad (2)$$

and the matrix

$$Q_{i,i} = 2 \left(h_i - \sum_{j=1}^n J_{i,j} \right), \quad Q_{i,j} = 4J_{i,j}, \tag{3}$$

the Ising objective in Equation (1) can be rewritten in the form of a QUBO:

$$\min \mathbf{x}^T Q \mathbf{x}, \quad \text{s.t. } \mathbf{x} \in \{0, 1\}^N. \tag{4}$$

Note that the transformation in Equations (2) and (3) is actually invertible (c.f. for instance, Ref. [49]). Hence, minimizing the Ising objective in Equation (1) is *equivalent* to solving a QUBO. In what follows, we will use the QUBO form only; it can be in fact submitted as it is to the solvers and commercial quantum devices directly. Moreover, the matrix Q can always be chosen to be symmetric, as $Q = (Q' + Q'^T)/2$ defines the same objective. Quantum annealers accept problems both in QUBO and Ising form and provide a non-deterministic output possibly containing the solution, as it will be discussed later.

A QUBO or Ising model can be also solved with other promising techniques. With the rapid development of quantum annealing technology, probabilistic *classical* accelerators have been under massive development. In recent years, a significant progress took place in the field of programmable gate array optimization solvers (digital annealers [50]), optical Ising simulators [51], coherent Ising machines [52], stochastic cellular automata [53], and, in general, those based on memristor electronics [54].

It is therefore vital to develop modeling strategies for quadratic binary optimization and to create novel techniques for analyzing the obtained results. This should progress similarly to how the powerful solvers for linear programs first started appearing: modeling strategies for linear programs as well as sensitivity analysis had been developed ahead of the creation of the hardware.

2.3.2. Quantum Annealing

An essential step in finding the minimum of an optimization problem (encoded in Equation (1)) efficiently is to map it to its quantum version. The mapping assigns a two-dimensional complex vector space to each spin, and a complete spin configuration becomes an element of the direct (tensor) products of these spaces. An orthonormal basis (ONB) is assigned to the -1 and $+1$ values of the variables; thus, the product of these vectors will be an ONB (called the “computational basis”) in the whole \mathbb{C}^{2^N} . The vectors with unit Euclidean norms are referred to as “states” of the system; they encode the physical configurations. The fact that the state can be an arbitrary vector and not only an element of the computational basis means that the quantum annealer can simultaneously process multiple configurations, i.e., inherent parallelism.

As to the objective function, the spin variables are replaced by their quantum counterpart: Hermitian matrices acting on the given spin’s \mathbb{C}^2 tensor subspace. The product of spins is meant to be the direct (tensor) product of the respective operators. Thus, the objective function Equation (1) turns into a Hermitian operator, which is referred to as the problem’s Hamiltonian:

$$\mathcal{H}_p := E(\hat{\sigma}^z) = \sum_{\langle i,j \rangle \in \mathcal{E}} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z + \sum_{i \in \mathcal{V}} h_i \hat{\sigma}_i^z, \tag{5}$$

whose lowest-energy eigenstate is commonly called the “ground state”. Above, σ_i^z denotes the Pauli z -matrix associated with the i th qubit. In the present case, it is an element of the computational basis, so it represents also the optimal configuration of the classical problem. Note that the energy of a physical system is related (via eigenvalues) to a Hermitian operator, which is called its Hamiltonian. Although it seems to be a significant complication to deal with \mathbb{C}^{2^N} instead of having 2^N binary vectors, it has important benefits, the most remarkable of which is that they model realistic physical systems.

The main idea behind quantum annealing is based on the celebrated adiabatic theorem [55]. Assume that a quantum system can be prepared in the ground state of an initial (“simple”) Hamiltonian \mathcal{H}_0 . Then, it will slowly evolve to the ground state of the final (“complex”) Hamiltonian \mathcal{H}_p in Equation (5) that can be harnessed to encode the solution of an optimization problem [45]. In particular, the dynamics of quantum annealers such as D-Wave 2000Q are governed by the following time-dependent Hamiltonian [46,56]:

$$\mathcal{H}(t)/(2\pi\hbar) = -g(t)\mathcal{H}_0 - \Delta(t)\mathcal{H}_{p'}, \quad t \in [0, T]. \quad (6)$$

Here, the original problem’s Hamiltonian in Equation (5) must be converted into a bigger one $\mathcal{H}_{p'}$, whose graph is compliant with what the existing hardware can realize: the “Chimera graph” in case of DWave 2000Q; see Figure 1. The original problem’s graph will be the minor of this graph. This procedure, called “minor embedding”, is standard in quantum annealing procedures (see also Section SIIC of the the Supplemental Materials for a simple graphical representation of this *Chimera embedding*).

Many relevant optimization problems are defined on dense graphs. Fortunately, even complete graphs can be embedded into a Chimera graph [57]. There is, however, substantial overhead, which effectively limits the size of the computational graph that can be treated with current quantum annealers [58,59]. This is considered as an engineering issue that will likely be overcome in the near future [11,60]. After the Chimera embedding, the Hamiltonian describing the system reads as follows:

$$\mathcal{H}_{p'} = \sum_{\langle i,j \rangle \in \mathcal{E}} J'_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z + \sum_{i \in \mathcal{V}} h'_i \hat{\sigma}_i^z, \quad \mathcal{H}_0 = \sum_i \hat{\sigma}_i^x, \quad (7)$$

where $\hat{\sigma}_i^x$ is the x Pauli matrix associated with the i th qubit. The annealing time T varies from microseconds ($\sim 2 \mu\text{s}$) to milliseconds ($\sim 2000 \mu\text{s}$) depending on the specific programmable schedule [46]. As shown in Figure 1, during the evolution, $g(t)$ varies from $g(0) \gg 0$ (i.e., all spins point in the x -direction) to $g(T) \approx 0$, whereas $\Delta(t)$ is changed from $\Delta(0) \approx 0$ to $\Delta(T) \gg 0$ (i.e., $\mathcal{H}(T) \sim \mathcal{H}_{p'}$). The Pauli operators $\hat{\sigma}_i^z$, $\hat{\sigma}_i^x$ describe the spin’s degrees of freedom in the z - and x -direction, respectively. Note that the Hamiltonian \mathcal{H}_p is classical in the sense that all its terms commute (which is the result of their multiplication, being independent of the order). Thus, as mentioned previously, its eigenstates translate directly to classical variables, $q_i = \pm 1$, which are introduced to encode discrete optimization problems.

The annealing time, T in Equation (6), is an important parameter of the quantum annealing process: it must be chosen so that the system reaches its ground state while the adiabaticity is at least approximately maintained. The adiabatic theorem gives us guidance in this respect. In the spectrum of the Hamiltonian in Equation (6), there is a difference between the energy of the ground state(s) and the energy of the state(s) just above it in energy scale. This difference is known as the (spectral) “gap”, and its minimum value in the course of the evolution determines the required computation time if certain additional conditions hold. Roughly speaking, the bigger the gap, the faster the quantum system reaches its ground state (the dependence is actually quadratic; see Ref. [61] for a detailed discussion). Thus, if the run time is not optimal, there is a finite probability of reading out an excited state instead of the true ground state.

The annealing time should be provided in advance to actually use a quantum annealer. As mentioned before, the time which would ensure that the ground state is likely to be in the resulting sample depends on the spectral gap, which is unknown. Its exact determination would be as hard as finding the actual optimum. Hence, in practice, a reasonable annealing time is determined from an educated guess, and the evolution is repeated reasonably many times, resulting in a *sample* of possible solutions (over different annealing times as well as other relevant parameters). The one with the lowest energy is considered to be the desired solution, albeit there is a finite probability that it is not the ground state. A quantum

annealer is thus a probabilistic and heuristic solver. Concerning the benchmarking of quantum annealers, consult also [62].

As a side note, it should be stressed that it is not always possible to maintain the adiabatic evolution. As an example, consider the second-order phase transition phenomenon [63–65], in which even a short-lasting lack of adiabaticity will result in the creation of topological defects preventing the system from remaining in its instantaneous ground state. This effect, on the other hand, is a clear manifestation of the quantum Kibble–Zurek mechanism Refs. [66–72] and can be used to detect departures from adiabaticity. Meanwhile, a system which would evolve adiabatically in the absence of interaction with its environment will keep evolving similarly to the ideal evolution also in certain noisy circumstances [73].

2.3.3. Classical Algorithms for Solving Ising Problems

An additional benefit from formulating problems in terms of Ising-type models is that the existing methods developed in statistical and solid-state physics for finding ground states of physical systems can also be used to solve a QUBO on classical hardware. Notably, variational methods based on finitely correlated states (such as matrix product states for 1D systems or projected entangled pair states suitable for 2D graphs) have had a very extensive development in the past few decades. A quantum information theoretic insight into density matrix renormalization group methods (DMRG [74])—being the most powerful numerical techniques in solid-state physics at that time—helped with proving the correctness of DMRG. These methods also led to a more general view of the problem [75], resulting in many algorithms that have potential applications in various problems, in particular solving QUBOs by finding the ground state of a quantum spin glass. We have used the algorithms presented in Ref. [76] to solve the models derived in the present manuscript.

Neither the quantum devices nor the mentioned classical algorithms do always provide the energy minimum and the corresponding ground state (as it is not trivial to reach it [77]) but possibly another eigenstate of the problem with an eigenvalue (i.e., a value of the objective function) close to the minimum. The corresponding states are referred to as “excited states”. There are problems related to the simulations of quantum systems with NISQ quantum hardware, where only the ground state is relevant [78]. Nevertheless, excited states can also encode valuable information. This is especially apparent, for instance, in the context of neural networks where sampling is more important than finding the ground state [79]. In many optimization problems, good but not optimal solutions also bear practical relevance. Another important point in interpreting the results of such a solver is the degeneracy of the solution: it can provide multiple equivalent optima at a time.

In analyzing these optima, it is helpful that for up to 50 variables, one can calculate the exact ground states and the excited states closest to them using a brute-force search on the spin configurations with GPU-based high-performance computers. In the present work, we also use such algorithms, in particular those introduced in Ref. [80] for benchmarking and evaluating our results for smaller examples. This way, we can compare the exact spectrum with the results obtained from the D-Wave quantum hardware and the variational algorithms.

3. Our Model

Here, we describe our model in brief. The Supplemental Materials provides a more detailed description. First, in Section 3.1, we encode constraints representing the railway operation scenario in the form of inequalities. To avoid continuous variables which would be incompatible with a QUBO solver, we use discretized time variables. The arising integer model is suitable to be turned into a purely 0-1 model adopting the discrete time unit approach, as described in Section 3.2. Then, in Section 3.3, the constrained 0-1 model is turned into the desired QUBO model using penalties. Finally, the QUBO model is converted to the Ising model as in Equation (1). This is completed automatically by the quantum

annealer’s software package (using the binary variable transformation via Equation (2) and QUBO matrix transformation in Equation (3); the QUBO and Ising models are equivalent and any of these formulations can be submitted to solvers such as D-Wave directly). While the quantum annealer is an Ising-system inside, the solutions it returns are mapped back to the 0-1 variables of the submitted QUBO model (c.f. Equation (2)). Given such a 0-1 solution vector, it defines the actual delay of each train at each station, as described in Section 3.2. Applying these delays to the given initial timetable (with *conflicts*), a solution in the form of the *conflict free* timetable is obtained: conflicts are resolved. The presented train diagrams are constructed in this way.

3.1. Integer Formulation of the Constraints

Let us return to our single-track *network: blocks* and trains. Observe that it is only the leave times of trains from station blocks that the dispatchers decide upon, as the trains cannot meet and pass or meet and overtake on a single-track line otherwise. Let us denote the station blocks by $s \in \mathcal{S}$ and the set of trains by $j \in \mathcal{J}$. We will formulate the problem entirely in terms of the secondary delays: $d_s(j, s)$ stands for the secondary delay of train j at the station block s . The detailed description in the Supplemental Materials makes it clear that these values, along with the original timetable and the technical data (i.e., the *network* topology and time required for each train to pass a block) fully determine a modified timetable. In what follows, this description will be referred to as the *delay representation*.

In order move toward a 0-1 model, we discretize the secondary delays requiring that

$$d_s(j, s) \in \mathbb{N}, \quad 0 \leq d_s(j, s) \leq d_{\max}(j), \tag{8}$$

where a reasonable upper bound $d_{\max}(j)$ can be obtained from some fast heuristics, and the time is measured in integer minutes from now on, which is a suitable scale for railway problems. When formulating constraints, it is better to work with the actual (discretized) delay $d(j, s) = d_s(j, s) + d_U(j, s)$ of train j at station block s , where $d_U(j, s)$ stands for the primary (unavoidable) delay. At this stage, we have defined a set of potential decision variables with finite ranges that already facilitate the formulation of a linear model for the problem, as shown in the Supplemental Materials.

As to the constraints, we consider the following ones, which cover the requirements of the particular railway operator. In the Supplemental Materials, we describe them in more detail while here, we give a brief summary:

The minimum passing time condition ensures that no block sections are passed by any train faster than allowed:

$$d(j, \rho_j(s)) \geq d(j, s) - \alpha(j, s, \rho_j(s)). \tag{9}$$

where $\rho_j(s)$ stands for the station block section succeeding s in train j -s sequence, while $\alpha(j, s, \rho_j(s))$ is the largest reserve the train can achieve by passing the *blocks* following s up to and together with the next station block $\rho_j(s)$ possibly faster than originally planned. The α values can be calculated in advance.

The single-block occupation ensures that at most one train can be present in a block section at a time.

$$d(j', s) \geq d(j, s) + \Delta(j, s, j', s) + \tau_{(1)}(j, s, \rho_j(s)). \tag{10}$$

where Δ is the difference of the leave times of two trains from the given *blocks*, whereas $\tau_{(1)}$ is the minimum time for train j to give way to another train going in the same direction in the route $s \rightarrow \rho_j(s)$. This condition is to be tested in this form for the pair of trains (j, j') if j leaves s before j' , i.e., $d(j', s) \geq d(j, s) + \Delta(j, s, j', s)$; otherwise, it has to be applied so that the order of the trains is reversed.

The deadlock condition ensures that no pairs of trains heading in the opposite direction will be waiting for each other to pass the same *blocks*:

$$d(j', \rho_j(s)) \geq d(j, s) + \Delta(j, s, j', \rho_j(s)) + \tau_{(2)}(j, s, \rho_j(s)), \tag{11}$$

where $\tau_{(2)}(j, s, \rho_j(s))$ is the minimum time required for train j to get from station block s to $\rho_j(s)$. Similarly to the previous condition, Equation (11) is to be applied for the pairs of trains (j, j') so that j' is supposed to leave the block $\rho_j(s)$ after the train j leaves s ; otherwise, the order of the trains is reversed.

The rolling stock circulation condition ensures the minimal technological time $R(j, j')$ for a given train set arriving as train j at its terminating station $s_{j,\text{end}}$ before operating again as train j' :

$$d(j', 1) > d(j, s_{j,\text{end}-1}) - R(j, j'). \tag{12}$$

Certainly, this condition has to hold for pairs of trains (j, j') which are operated with the same train set according to the rolling stock circulation plan.

There is an additional constraint: the capacity condition that could be also addressed; this would implement the buffer approach in our model. This is described in the Supplemental Materials, but we omit it here, as we will not include it in the calculation. It would increase additional complexity that would make our model intractable with current quantum hardware, so we opted for the verification of the solutions against this condition, thereby implementing the buffer approach. Having described the constraints, now, we formulate the model as a 0-1 program and define the objective function.

3.2. 0-1 Formulation

To turn our model into a 0-1 problem, we introduce our final decision variables

$$x_{s,j,d} = \begin{cases} 1, & d(j, s) = d \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

which take the value of 1 if the train j leaves station block s at delay d and zero otherwise. In this way, we have 0-1 variables with indices from a finite set. Observe that for constant d_{max} , the number of variables is proportional to the number of trains and the number of stations minus one (as we do not consider departing from the last station in our model).

As for the objective function, we opt for a weighted sum of delays:

$$f(\mathbf{x}) = \sum_{j \in J} \sum_{s \in S_j^*} \sum_{d \in A_{j,s}} f(d, j, s) \cdot x_{j,s,d}, \tag{14}$$

where $f(d, j, s)$ are the weights. Here, $S_j^* = S_j \setminus \{s_{j,\text{end}}\}$, where S_j stands for a sequence of station blocks the train runs through, $s_{j,\text{end}}$ stands for the last station of j , and $A_{j,s}$ is the respective range of delays. It is easy to see that the weights $f(d, j, s)$ can be chosen so that they depend on the secondary delays only; consult the Supplemental Materials for details.

As for the constraints, let us first assume that each train leaves each station block once and only once (recall that we do not allow for recirculation):

$$\forall_j \forall_{s \in S_j} \sum_{d \in A_{j,s}} x_{j,s,d} = 1. \tag{15}$$

The other constraints can be dealt with as follows.

The minimum passing time condition defined in Equation (9) becomes

$$\forall_j \forall_{s \in S_j^{**}} \sum_{d \in A_{j,s}} \left(\sum_{d' \in D(d) \cap A_{j,\rho_j(s)}} x_{j,s,d} x_{j,\rho_j(s),d'} \right) = 0, \tag{16}$$

where $D(d) = \{0, 1, \dots, d - \alpha(j, s, \rho_j(s)) - 1\}$, and $S_j^{**} = S_j \setminus \{s_{j,\text{end}}, s_{j,\text{end}-1}\}$.

The single-block occupation condition from Equation (10) follows that

$$\forall_{(j,j') \in \mathcal{J}^0(\mathcal{J}^1)} \forall_{s \in S_{j,j'}^*} \sum_{d \in A_{j,s}} \left(\sum_{d' \in B(d) \cap A_{j',s}} x_{j,s,d} x_{j',s,d'} \right) = 0, \tag{17}$$

where $B(d) = \{d + \Delta(j, s, j', s), d + \Delta(j, s, j', s) + 1, \dots, d + \Delta(j, s, j', s) + \tau_{(1)}(j, s, \rho_j(s)) - 1\}$ is a set of delays that violates the block occupation condition.

The deadlock condition is to be addressed for two trains heading in the opposite direction; from Equation (11), it follows that

$$\forall_{j \in \mathcal{J}^0(\mathcal{J}^1), j' \in \mathcal{J}^1(\mathcal{J}^0)} \forall_{s \in S_{j,j'}^*} \sum_{d \in A_{j,s}} \left(\sum_{d' \in C(d) \cap A_{j',\rho_j(s)}} x_{j,s,d} x_{j',\rho_j(s),d'} \right) = 0 \tag{18}$$

where $C(d) = \{d(j, s) + \Delta(j, s, j', \rho_j(s)), d(j, s) + \Delta(j, s, j', \rho_j(s)) + 1, \dots, d(j, s) + \Delta(j, s, j', \rho_j(s)) + \tau_{(2)}(j, s, \rho_j(s)) - 1\}$, and $\mathcal{J}^0, \mathcal{J}^1$ are explained in Equation (S1) of the Supplemental Materials. In Equations (17) and (18), each train is compared with a limited number of trains; this limit is imposed indirectly by fixed d_{max} . Hence, as a rough estimate of the number of terms in these two equations (for fixed d_{max}), one can claim that it is proportional to the number of trains times the number of stations minus one.

The rolling stock circulation condition is defined in Equation (12) and can be rewritten as

$$\forall_{j,j' \in \text{terminal pairs}} \sum_{d \in A_{j,s(j,\text{end}-1)}} \sum_{d' \in E(d) \cap A_{j',1}} x_{j,s(j,\text{end}-1),d} \cdot x_{j',s(j',1),d'} = 0, \tag{19}$$

where $E(d) = \{0, 1, \dots, d - R(j, j')\}$; this condition applies only for one station and a few selected trains.

The objective function in Equation (14) together with the constraints in Equations (15)–(19) comprise a quadratic constrained 0-1 formulation of our model. As an estimate of the number of variables and quadratic terms from Equations (13)–(18), one can conclude that (for fixed d_{max}), these are proportional to n.o. trains · (n.o. stations – 1).

3.3. QUBO Formulation: Penalties

Having formulated our problem as a constrained 0-1 program, we need to make it unconstrained to achieve a QUBO form—see Equation (4). This is usually completed with penalty methods [81]. It has been shown in [49] that all binary linear and quadratic programs translate to QUBO along some simple rules. (An alternative, symmetry-based approach [82] to constrained optimization has also been proposed in which the adiabatic quantum computer device is supposed to use a tailored \mathcal{H}_0 term in its dynamics of the model in Equation (7). As such a modification of the actual device is not available to us, we remain using penalty methods.)

The problems one faces with a quadratic 0-1 program require certain specific considerations when adopting the penalty method. Let us outline this approach with a focus on our problem. As we have a linear objective function Equation (14), it can be written as a quadratic function because the decision variables are binary:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} \mathbf{c}^T \mathbf{x} = \min_{\mathbf{x}} \mathbf{x}^T \text{diag}(\mathbf{c}) \mathbf{x}. \tag{20}$$

(A general QUBO can contain linear terms as well; however, the solver implementations accept a single matrix of quadratic coefficients [83], so transforming linear terms into quadratic ones is more a technical than a fundamental step.)

The constraints set out in Equations (16)–(19) have to be met for a feasible solution: they are *hard constraints*. To obtain an unconstrained problem, we define a penalty function

in the following way. We add the magnitude of the constraints' violation, multiplied by some well-chosen coefficient, to the objective function.

In particular, from Equations (16)–(19), we shall have quadratic constraints in the form of

$$\sum_{(i,j) \in \mathcal{V}_p} x_i x_j = 0, \tag{21}$$

excluding pairs of variables that are simultaneously 1. We can deal with such a constraint by adding to our objective the following terms:

$$\mathcal{P}_{\text{pair}}(\mathbf{x}) = p_{\text{pair}} \sum_{(i,j) \in \mathcal{V}_p} (x_i x_j + x_j x_i), \tag{22}$$

where p_{pair} is a positive constant. Additionally, from Equation (15), we have additional hard constraints in the form of:

$$\forall \mathcal{V}_s \quad \sum_{i \in \mathcal{V}_s} x_i = 1. \tag{23}$$

These constraints yield a linear expression that can be transformed into the following quadratic penalty function:

$$\mathcal{P}'_{\text{sum}}(\mathbf{x}) = \sum_{\mathcal{V}_s} p_{\text{sum}} \left(\sum_{i \in \mathcal{V}_s} x_i - 1 \right)^2. \tag{24}$$

Next, we replace the x_i s with x_i^2 s in the linear terms and omit the constant terms, as they provide only an offset to the solution, yielding:

$$\mathcal{P}_{\text{sum}}(\mathbf{x}) = \sum_{\mathcal{V}_s} p_{\text{sum}} \left(\sum_{i,j \in \mathcal{V}^{\times 2}, i \neq j} x_i x_j - \sum_{i \in \mathcal{V}_s} x_i^2 \right). \tag{25}$$

So, our effective QUBO representation is

$$\min_{\mathbf{x}} f'(\mathbf{x}) = \min_{\mathbf{x}} (f(\mathbf{x}) + \mathcal{P}_{\text{pair}}(\mathbf{x}) + \mathcal{P}_{\text{sum}}(\mathbf{x})), \tag{26}$$

which can be written in the form of Equation (4). We shall have many constraints similar in form to Equations (21) and (23), so we have one summed for each constraint in the objective. (It would also be possible to assign a separate coefficient to each of the constraints.)

Recall that in the theory of penalty methods [81] for continuous optimization, it is known that the solution of the unconstrained objective will tend to a feasible optimal solution of the original problem as the multipliers of the penalties (p_{sum} and p_{pair} in our case) tend to infinity, provided that the objective function and the penalties obey certain continuity conditions. As in our case, both the objective and the penalties are quadratic, and this convergence would be warranted for the continuous relaxation of the problem. Even though we have a 0-1 problem, if we had an infinitely precise solution of the QUBO, increasing the parameters would result in convergence to an optimal feasible solution.

However, somewhat similarly to the continuous case (in which the Hessian of the unconstrained problem diverges as the parameters grow, making the unconstrained problem numerically ill-conditioned), the properties of the actual computing approach or device make it more cumbersome to make a good choice of multipliers.

The parameters p_{sum} and p_{pair} have to be chosen so that the terms representing the constraints in this energy do not dominate the original objective function. If the penalties are too high, the objective is just a too small perturbation, which will be lost in the noise of the physical quantum computer or in the numerical errors of an algorithm modeling it. If, however, the penalty coefficients are too low, we obtain infeasible solutions. In the ideal case, there is a "feasibility gap" in the spectrum of solutions.

The multipliers can be assigned in an ad hoc manner by experimenting with the solution; however, a systematic, possibly problem-dependent approach to their appropriate assignment (as in the case of classical penalty methods; see [81]) would be highly desirable in order to make the QUBO more reliable and prevalent. The goal is to construct the QUBO representation in such a way that the energy landscape of the original problem is preserved. In particular, QUBO solutions that map to the feasible solutions of the original problem are expected to have lower energy than the infeasible ones. There are certain systematic methods for this: for instance, in case of linear constraints only, it is always possible to find the optimal penalty terms to separate between feasible and infeasible configurations [84]. In the present case study, we will determine penalties in an ad hoc manner based on numerical experience with the objective values and constraint violations.

Having a QUBO representation of the problem at hand (as well as an analogical Ising representation), let us turn our attention to the actual instances of our model and the results obtained for them.

4. Results

In this section, we discuss certain possible situations in train dispatching on the railway lines managed by the Polish state-owned infrastructure manager *PKP Polskie Linie Kolejowe S.A.* (*PKP PLK* in what follows). In particular, we consider two single-track railway lines:

- Railway line No. 216 (Nidzica–Olsztynek section);
- Railway line No. 191 (Goleszów–Wisła Uzdrowisko section).

Railway line No. 216 is of national importance. It is a single-track section of the passenger corridor Warsaw–Olsztyn, which has recently been modernized. There are both *Inter-City* (*IC*) and regional trains operating on the Nidzica–Olsztynek section of line No. 216. In this paper, we consider an official train schedule (as of April, 2020). The purpose of the analysis in this section is to demonstrate the application of our methodology to a real-life railway section.

Railway line No. 191 is of local importance. The main train service on the No. 191 railway line is Katowice–Wisła Głębce, which is operated by a local government-owned company called “*Koleje Śląskie*” (in English, Silesian Railways; abbreviated *KS*). There are a few *Inter-City* trains of higher priority there as well. Since 2020, the traffic at this section has been suspended due to comprehensive renovation works (a temporary rail replacement bus service is in operation). Our problem instances are based on the planned parameters of the line after its commissioning based on public procurement documents [85]. On the basis of these parameters, a cyclic timetable has been created. The aim of analyzing this case is to show the broader application possibilities of the methodology.

4.1. The Studied Network Segment

In Figure 2a, we present a segment of railway line No. 216 (Nidzica–Olsztynek section), and in Figure 2c, the analyzed part of the real timetable is depicted in the form of a train diagram.

In Figure 2a, three stations are presented. Block 1 represents Nidzica station, which has two platform edges numbered according to the rules of *PKP PLK*. Block 3 represents Waplewo station, with another two platform edges. Olsztynek station, with three platform edges, is represented by block 5. The model involves two line blocks with the labels 2 and 4. It is assumed that it takes the same amount of time to pass through a given station block regardless of which track the train uses. To leave the station, it is required that the subsequent block is free.

As to the trains, Figure 2c represents their planned paths. Three trains are modeled: the two *Inter-City* trains in red and the regional train in black. The scheduled meet-and-pass situations take place in Waplewo and Olsztynek (which might change in case of a delay). IC5320 leaves station block 5 (Olsztynek) at 13:54, has a scheduled stopover at station block 3 (Waplewo) from 14:02 to 14:10 to meet and pass IC3521, and finally arrives at station block 1 (Nidzica) at 14:25. As to the opposite direction, IC3521 leaves station block 1 at 13:53,

stops at station block 3 from 14:08 to 14:10, and arrives at station block 5 (Olsztynek) at 14:18. These two trains depart at the same time from station block 3 in opposite directions. The third train considered is R90602. It is scheduled to leave block 5 at 14:20 and stops at station block 3 (Waplewo) from 14:29 to 14:30, so it is scheduled to start occupying this track 19 min after both ICs left. It is behind IC5320 during the whole section and does not meet the IC train at all, so the original schedule is feasible and conflict free.

Now, let us add a 15-min delay to the departure time of IC5320 from station block 5 and 5-min delay to that of IC3521 from station block 1. The passing times were originally scheduled according to the maximum permissible speeds. The minimum waiting times at all the considered stations are 1 min regardless of the train type. This introduces the following situation: the two *Inter-City* trains and the regional train have a conflict at line block 4. This schedule will be referred to as the “conflicted diagram”—see Figure 3a. The resolution of this conflict requires making a decision at station blocks 3 and 5.

Let us now turn our attention to the other example. The line segment (a part of railway line No. 191) is presented in Figure 2b, while the considered train paths of the real timetable are shown in Figure 2d. There are four stations and another three stops for the passengers modeled. Block 1 represents Golezów station, which has four platform edges. Block 2 represents a line block between Golezów station and Ustroń station (which has two platform edges and is represented by block 3). Subsequently, we have three line blocks numbered 4, 5, and 6, with two stops for passengers: Ustroń Zdrój and Ustroń Poniwiec (with one platform edge each). Next, we have station block 7—Ustroń Polana, which has two platform edges. Between this station and Wisła Uzdrowisko station (numbered 10 with three platform edges), there are two more line blocks (8 and 9) with one stop for passengers (Wisła Jawornik). We assume that it takes exactly the same time to pass through a block regardless of the track used.

There are six trains, two *Inter-City* trains in red and four regional (*KS*) trains in black, as presented in Figure 2d. The regional trains serve all the stops and stations, while the *Inter-City* service stops only at stations. We consider Wisła Uzdrowisko (station block 10) to be a terminus for the *Inter-City* trains (however, it does not apply to the regional trains, which go farther). In this situation, there are no meet-and-pass situations at intermediate stations (Ustroń and Ustroń Polana) in the original timetable. Both *Inter-City* trains are served by the same train set, and the minimum service time is $R(j, j') = 20$ min at the terminus for ICs (block 10); see Condition SI.4 of the Supplemental Materials.

We analyze the following dispatching cases, which have been selected to demonstrate the algorithm behavior in various situations:

1. A moderate delay of the *Inter-City* train setting off from station block 1; see Figure S1a of the Supplemental Materials.
2. A moderate delay of all trains setting off from station block 1; see Figure S1b.
3. A significant delay of some trains setting off from station block 1; see Figure S1c.
4. A large delay of the *Inter-City* train setting off from station block 1; see Figure S1d.

The conflicted timetables of cases 1–4 are presented in Figure S1 of the Supplemental Materials.

4.2. Simple Heuristics

In the railway practice, conflicts are often resolved using simple heuristics: the First Come First Served (FCFS) and the First Leave First Served principles. A more complex one is AMCC (avoid maximum current C_{\max}) [34]. All of these heuristics are used to determine the order of trains when passing the blocks. In FCFS and FLFS, the way is given to the train that first arrives—or first leaves—the analyzed block section. In practice, the decisions based on both these heuristics are taken starting from the most urgent conflict. Next, since passing and overtaking is possible only at stations, so-called *implied selections* [35] are determined. The procedure is repeated as long as all the conflicts are solved.

The AMCC is a more complex approach whose objective is to minimize the maximum secondary delay of the trains; this objective will be referred to as the “AMCC objective” in

what follows. This is an intuitive procedure yet more sophisticated than FCFS and FLFS. To facilitate the comparison, stations are assigned an infinite capacity. Of course, solutions requiring a capacity higher than that of the given station must be rejected.

In the example presented in Figure 2a, for the conflicted timetable in Figure 3a, each of the heuristics returns the same solution; this is presented in Figure 3b. When comparing the FCFS with the FLFS, observe that in the conflicted timetable, three trains (IC5320, IC3521, R90602) are scheduled to occupy the block 4 simultaneously, which is forbidden.

To avoid the conflict, IC3521 is allowed to enter this block with a 3-min delay at 14:17 (as soon as IC5320 leaves the block), thus leaving the block at 14:25 instead of 14:22, which results in 3 min of secondary delay. Consequently, R9062 is allowed to enter the block not earlier than 14:25, which is an additional 4-min delay as compared with the conflicted timetable. Thus, the maximum secondary delay is 4 min, and the sum of the delays on entering the last block is 7 min. The maximum secondary delay is 4 min; it is the lowest possible one, so the solution is optimal with respect to the AMCC objective. Note that in case of such a simple situation, the use of the heuristics is very close to the enumeration and evaluation of all the possible solutions, and picking the best of these. This small example is included in order to have an instance that can be fully followed manually.

The other example—the disruptions of this presented in Figure 2b—is more complex, yet it is still solvable by a state-of-the-art quantum annealer. We do not discuss this example in detail; we only present the maximum secondary delay values in Table 1 for the discussed heuristics. Recall that we need an upper bound on the secondary delays to formulate our model; we opt for $d_{max} = 10$ on the basis of these data. The respective train diagrams are presented in Figures S2–S4 of the Supplemental Materials.

The values of the AMCC objective function are presented in Table 1; AMCC appears to find the optimum in these cases, thus providing a good enough reference for comparisons, albeit with an objective function different from that of ours. Our choice of the objective will be more flexible, thus leaving room for further non-trivial optimization.

Table 1. The maximum secondary delays, in minutes, resulting from simple heuristics. Observe that for each case, there are solutions far below $d_{max} = 10$.

Heuristics	Case 1	Case 2	Case 3	Case 4
FLFS	6	13	4	2
FCFS	5	5	5	2
AMCC	5	5	4	2

4.3. Quantum and Calculated QUBO Solutions

Our QUBO approach uses the objective function set out in Equation (26). This contains the feasibility conditions (*hard constraints*) and the objective function $f(\mathbf{x})$ of Equation (27). For the feasibility part, we need to determine $\tau_{(1)}(j, s)$, the minimum time for train j to give way to another train going in the same direction, and $\tau_{(2)}(j, s)$, the minimum time for train j to give way for the another train going in the opposite direction (see SI.2 and SI.3 of the Supplemental Materials).

As noted before, the QUBO objective function introduces flexibility in choosing the dispatching policy by setting the values of the penalty weights for the delays of the trains. In this way, almost any train prioritization is possible. To demonstrate this flexibility, we make the penalty values proportional to the secondary delays of the trains that enter the last station block. This is equivalent to the secondary delay on leaving the penultimate station block. Each train is assigned a weight w_j , yielding the form of Equation (S38) of the Supplemental Materials.

$$f(\mathbf{x}) = \sum w_j \cdot \frac{d(j, s^*) - d_U(j, s^*)}{d_{max}(j)} \cdot x_{j, s^*, d}, \tag{27}$$

where the sum is taken over $j \in J$ and $d \in A_{j,s^*}$ with $s^* = s_{(j,\text{end}-1)}$. Note that this objective coincides with that of the linear integer programming approach; see Equation (S28) of Supplemental Materials, which will be used for comparisons.

The following train prioritization is adopted. In the case of railway line No. 216, the *Inter-City* trains are assumed to have a higher value of the delay penalty weight $w = 1.5$, while the regional train is weighted $w = 1.0$. We assign the higher priority to the *Inter-City* train in accordance with the train prioritization rules in Poland (and in many other countries). In the other case (line No. 191), the priorities of trains heading toward block 10 (Wisła Uzdrowisko) are lower, weighted 0.9 for all the other trains in this direction. However, train priorities for the trains heading in the opposite direction (toward block 1—Goleszów—and beyond the analyzed section) have higher values: 1.0 for the regional trains and 1.5 for the *Inter-Cities*. Such a policy is motivated by the reluctance of letting the delays propagate across the Polish railway network—that regional trains proceed toward the main railway junction in the region’s major city (Katowice) and that the *Inter-City* train service is scheduled toward the state’s capital city (Warsaw). Observe that w_j is the highest possible penalty for the delay of a train j ; see Equation (27). In both these cases, the maximum of w_j is 1.5. Hence, the penalties for a infeasible solution should be higher (as discussed in Section 3.3. We set $p_{\text{pair}} = p_{\text{sum}} = 1.75 > 1.5$.

As mentioned before, the maximum secondary delay parameter d_{max} (which is assumed to be the same for all trains and all analyzed station blocks for sake of simplicity) cannot be smaller than the delay value returned by the AMCC heuristics. However, since the AMCC may not be optimal in terms of our objective function, we need to leave a margin for some larger values of the maximum secondary delay. On the other hand, since the system size grows with d_{max} , it must be limited enough to make the problem applicable to state-of-the-art quantum devices and classical algorithms motivated by them. Specifically, since we do not analyze the delays at the last station of the analyzed segment of the line, the required number of qubits will be approximately $(\text{number of station blocks} - 1) \cdot (\text{number of trains}) \cdot (d_{\text{max}} + 1)$.

In the case of railway line No. 216, we set $d_{\text{max}} = 7$, which is considerably larger than the AMCC solution. There are 48 logical quantum bits needed to handle this problem instance, making it suitable for both quantum annealing at the current state of the art, and the GPU-based implementation of the brute-force search for the low-energy spectrum (ground state and subsequent excited state) [80], which is possible with up to 50 quantum bits. The benefit of this possibility is that it provides an exact picture of the spectrum, which can be used as a reference when evaluating the heuristic results of approximate methods (tensor networks) or quantum annealing. This may guide the understanding of the results of the bigger instances, in which the brute-force exact search is not available.

There are many possible distinct solutions in the case of line No. 191, making the analysis more interesting from the dispatching point of view. We set $d_{\text{max}} = 10$: for a justification, see Table 1, and observe that d_{max} is considerably larger than the AMCC output. The $d_{\text{max}} = 10$ yields 198 logical quantum bits, which we were able to embed into a present-day quantum annealer, the D-Wave device DW-2000Q5, in most cases.

Recall that current quantum annealing devices are imperfect and often output excited states. The clue of our approach is that the excited states (e.g., returned by the quantum annealer) still represent the optimal dispatching solutions, provided that their corresponding energies are relatively small. The reason for this is that what really needs to be determined is the order of trains leaving from each station block (i.e., this is the decision to be made). What is crucial here is to determine all the meet-and-pass and the meet-and-overtake situations (in analogy with the determination of all the precedence variables in the linear integer programming approach). The exact time of leaving block sections is of secondary importance. Therefore, we consider those excited states that describe the same order of trains as the ground state to be equivalent to the actual ground state encoding the global optimum. As discussed in Section 3.3, our QUBO formulation problem ensures that those equivalent solutions are present in the low-energy spectrum.

4.3.1. Exact Calculation of the Low-Energy Spectrum

To demonstrate the aforementioned idea, we first present the results of the brute-force numerical calculations performed on a GPU architecture [80]. With this approach, the spectra of the smallest instances have been calculated exactly, providing some guidance for the understanding of the model behavior and parameter dependence, especially with respect to the low-energy part of the spectrum. Note that the brute-force search is actually an enumeration of all the possible configurations computed using a GPU. The method is suitable for small (i.e., up to 50 quantum bits) but otherwise arbitrary systems. No embedding is needed. To study (hard) penalties resulting from non-feasible solutions, apart from $p_{\text{pair}} = p_{\text{sum}} = 1.75$ in Equation (26), we use other higher penalties that are not equal to each other, $p_{\text{pair}} = 2.7$ and $p_{\text{sum}} = 2.2$.

Let us assume that the solution in Figure 3b is the optimal one. Here, the train IC3521 ($w = 1.5$) waits 3 min at block 3, while regional train R90602 ($w = 1.0$) waits 4 min at block 5, causing 4 min of secondary delay upon leaving block 3. This adds 1.214 to the objective. Concerning the feasibility terms in Equation (26), for a feasible solution $\mathcal{P}'_{\text{sum}} = 0$, while the linear constraint gives the negative offset to the energy. According to Equation (25), as we have three trains for which we analyze two stations, this negative offset is $\mathcal{P}_{\text{sum}} = -3 \cdot 2 \cdot p_{\text{pair}}$. Based on the feasibility terms set out Equation (26), this yields -10.5 for $p_{\text{sum}} = 1.75$ and -13.2 for $p_{\text{sum}} = 2.2$. This results in a ground-state energy of $f'(\mathbf{x}) = -9.286$ and $f'(\mathbf{x}) = -11.986$, respectively. Finally, in the ground-state solution shown in Figure 3b, the IC3521 train can leave the station block 1 with a secondary delay of 0, 1, 2, or 3, not affecting any delays of the trains leaving block 3. All these situations correspond to the ground state energy. Hence, our approach produces a 4-fold degeneracy of the ground state.

Low-energy spectra of the solutions and their degeneracy are presented in Figure 4a,b. All the solutions that are equivalent to the ground state from the dispatching point of view are marked in green. Infeasible excited state solutions (in which some of the feasibility conditions set out in Equation (26) are violated) are marked in red. In this example, we do not have feasible solutions that are not optimal, i.e., in which the order of trains at a station is different from the one in the ground-state solution.

In the case of line No. 191, a more detailed analysis of the low-energy spectra of the solutions was possible due to the generality of the brute-force simulation. The results are presented in Figure 4. We shall find later on that the D-Wave solutions were in the “green” tail of feasible solutions, but the high degeneracy of higher-energy states may impose some risk of the quantum annealing ending up in the more frequently appearing excited states (see Figure 5).

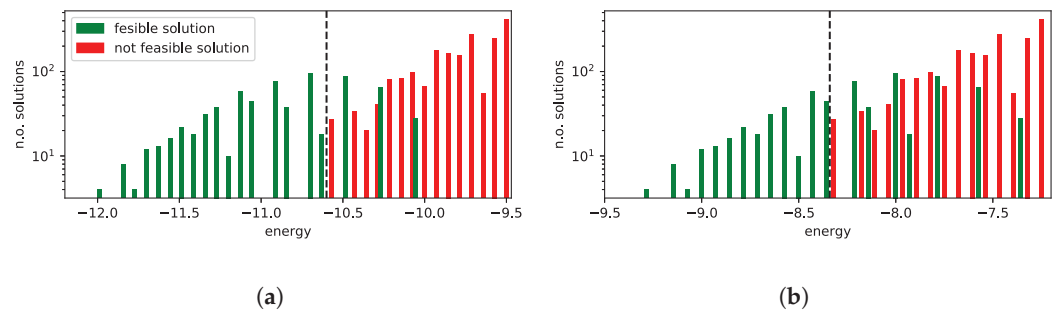


Figure 4. Spectra of the low-energy solutions for two penalty strategies of the brute-force (exact) solution. The black line separates the phase in which only feasible solutions appear. Observe the mixing phase, in which both feasible and unfeasible solutions occur. Here, p_{pair} and p_{sum} are penalties of the unconstrained problem expressed in the “logical” variables. The term $p_{\text{sum}} = (\sum_{i \in \mathcal{V}_s} x_i - 1)^2$, cf. Equation (24), ensures that each train leaves a station only once, whereas $p_{\text{pair}} = \sum_{(i,j) \in \mathcal{V}_p} (x_i x_j + x_j x_i)$, cf. Equation (25), imposes the following: minimal passing time constrain, single block occupation constrain, deadlock constrain, and rolling stock circulation constrain. (a) $p_{\text{pair}} = 2.7, p_{\text{sum}} = 2.2$. (b) $p_{\text{pair}} = p_{\text{sum}} = 1.75$.

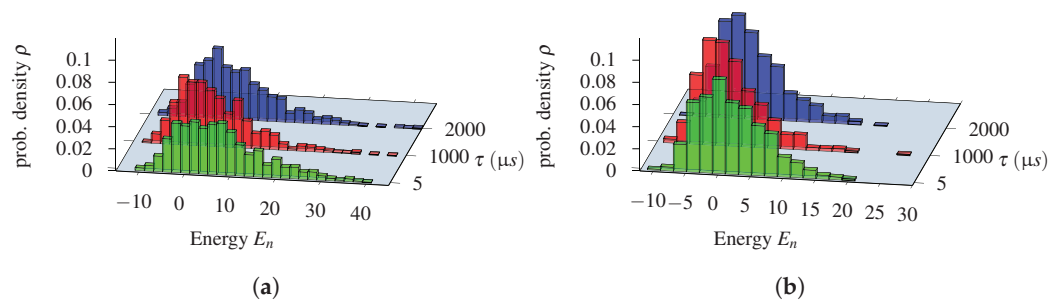


Figure 5. Distribution of the energies corresponding to the states (solutions), which are sampled by the D-Wave 2000Q quantum annealer of 48 logical quantum bits instance of line No. 216. In particular, 1000 samples were taken for each annealing time, and the strength of embedding was set to $css = 2.0$. This device is still very noisy and prone to errors, so the sample contains excited states. (a) QUBO param.: $p_{\text{pair}} = 2.7, p_{\text{sum}} = 2.2$. (b) $p_{\text{pair}} = p_{\text{sum}} = 1.75$.

4.3.2. Classical Algorithms for the Linear (Integer Programming) IP Model and QUBO

We expect classical algorithms for QUBOs to achieve the ground state of Equation (26) or at least low excited states equivalent to the ground state with respect to the dispatching problem. It is important to mention that hereafter, we embed the original QUBO into the Chimera graph (see Section 2.3.1). This makes the algorithm ready for processing on a real quantum annealer.

As to a simple example of the embedding, we refer to the problem with four quantum bits that has been discussed in Section SIIC of the Supplemental Materials. In that case, the mapping was trivial. In a case of six quantum bits, for instance (by setting $d_{\text{max}} = 2$), we will have additional terms in Equations (S41)–(S43) of the Supplemental Materials. Hence, the larger problems cannot be directly mapped onto the Chimera graph, so the embedding procedure is required, as illustrated in Figure 6. This illustrates the basic idea of how the embedding is performed in even larger models.

As to the model parameters, recall that for the particular QUBO, we have opted for $p_{\text{pair}} = p_{\text{sum}} = 1.75$ or $p_{\text{pair}} = 2.2, p_{\text{sum}} = 2.7$ for line No. 216 and $p_{\text{pair}} = p_{\text{sum}} = 1.75$ for line No. 191. Let us present the solutions of the two state-of-the-art numerical methods, which we shall later compare with the experimental results obtained by running the D-Wave 2000Q quantum annealers.

The first solver is developed ‘in-house’ and is based on tensor network techniques [76]. The solver is designed to efficiently sample high-quality solutions of certain spin-glass systems with the aim of solving hard optimization problems, and it has proven to be

applicable in our case. The idea behind this solver is to represent the probability of finding a given configuration by a quantum annealing processor as a PEPS tensor network. This allows an efficient bound-and-branch strategy to be applied in order to find $M \ll 2^N$ candidates for the low-energy states, where N is the number of physical quantum bits on the Chimera graph. In principle, such a heuristic method should work well for rather simple QUBO problems, i.e., those in which the Q matrix in Equation (4) has some identical or zero terms; this corresponds to the so-called weak entanglement regime. It can be shown that this is the case in our problem (see also the simple example of the Q matrix in Equation (S44) of Supplemental Materials), which makes the algorithm applicable in the present context. Furthermore, heuristic parameters such as the Boltzmann temperature (β) can be provided, allowing one to zoom in on the low-energy spectrum depending on the problem in question. We set $\beta = 4$, which is quite a typical setting, as discussed in [76]. Although even better solutions may potentially be achieved by further tuning this parameter, we demonstrated that this default setting is satisfactory from the dispatching point of view. The second classical solver is CPLEX [86] (version 12.9.0.0). In our work, we have used the DOCplex Mathematical Programming package (DOCplex.MP) for Python. In what follows, “CPLEX” refers to the QUBO solver of this package.

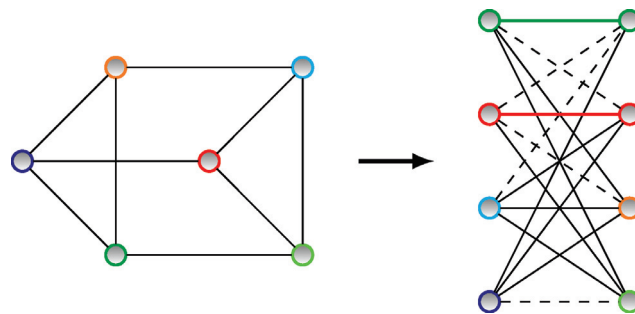


Figure 6. Embedding of a simple, six-qubit problem. **(Left)** graph of the original problem. **(Right)** problem embedded into a unit cell of Chimera. Here, different colors correspond to different logical variables. Apparently, the original problem does not map directly onto Chimera as it contains cycles of length 3. Therefore, two chains have to be introduced. Couplings corresponding to inner-chain penalties are marked with the same color as the variable to which they correspond.

In order to have a fair comparison with a traditional approach, we have also formulated our model as a linear integer program; this is described in Section SII of the Supplemental Materials in detail. We have implemented the linear integer model with the PuLP package [87] and solved with its default solver (CBC MILP Solver Version: 2.9.0). All instances were solved to the optimal solution in 0.03 s on an average computer. This was in line with our expectations, as our problems are small. Our goal is, however, not to outperform either CPLEX or the standard linear solver but to demonstrate the applicability of quantum hardware; at the present state of the art, we need the well-established solvers to produce results for comparisons and reference.

Concerning the results of the other railway line (No. 191), the values of the objective function in Equation (27) are given in Table 2. We also include the values of our objective function for the FLFS, FCFS, and AMCC optimal solutions.

The agreement with the linear integer programming approach provides the argument that the CPLEX results correspond to the ground state of the QUBO. We are interested in the results being equivalent to those of CPLEX and the linear solver from the dispatching point of view. These results are marked in blue in Table 2. The tensor network approach yields equivalent solutions to those of CPLEX. However, the tensor network sometimes returns excited states of the QUBO, as can be observed in case 3. The reason for this is that the tensor network method is based on approximations. This demonstrates that even some low-energy excited states encode a satisfactory solution. Interestingly, the results of the AMCC are also equivalent to those CPLEX in cases 2, 3, and 4 but different from

those in case 1. The reason is that the AMCC needs to have a specific objective function, whereas in our approach, we can choose this function more flexibly. Specifically, in case 1, the meet-and-pass situation of trains IC1 and Ks2 at station 10 yields the lowest maximum secondary delay, so it is optimal from the AMCC point of view. (Note that two trains have secondary delays: Ks2 and Ks3 in this case.) As discussed earlier, in this approach, Ks2 is prioritized, as it is the train leaving the modeled *network* segment, and one of the goals is to limit delays propagating further from this segment. The train diagrams based on the CPLEX solutions are depicted in Figure S5 of the Supplemental Materials.

In case 3, observe that the objective function in Table 2 from the tensor network solution differs from the minimum (yet the solution is still equivalent to the optimal one). To explain this, observe that there are numerous possibilities of additional train delays that do not affect the dispatching situation. An example of such a situation is a train having its stopover extended at the station with no meet and pass or meet and overtake. Such a situation increases the value of the objective but does not affect the optimal dispatching solution. The number of combinations here is high, and this is why such extended stopovers may be returned by the approximate algorithm. This is in contrast to the exact FCFS, FLFS, and AMCC heuristics, which do not allow for such unnecessary delays; the exact heuristics always return $f(x)$ that is the minimum for the particular dispatching solution. In case 3, the FCFS with $f(x) = 0.95$ does not give the optimal solution from the dispatching point of view, as opposed to the tensor network with $f(x) = 1.65$.

Table 2. The values of the objective function $f(x)$ for the solutions obtained by the classical calculation of the QUBO, linear integer programming approach, and all the heuristics. The blue color denotes equivalence from the dispatching point of view with the ground state of the QUBO or the output of the linear integer programming. The equivalence concerns the same order of trains at each station.

Method		Case 1	Case 2	Case 3	Case 4
QUBO approach	CPLEX	0.54	1.40	0.73	0.20
	tensor network	0.54	1.40	1.65	0.29
linear integer programming		0.54	1.40	0.73	0.20
Simple heuristics	AMCC	0.77	1.30	0.73	0.20
	FLFS	0.54	1.71	0.73	0.20
	FCFS	0.77	1.30	0.95	0.20

4.3.3. Quantum Annealing on the D-Wave Machine

As described in Section 2.3.2, physical quantum annealers are probabilistic. In particular, as the required time to drive the system into its ground state is unknown, the output is a sample of the low-energy spectrum from repeated annealing processes, hence it can be regarded as a heuristic. The solution is thus assumed to be the element of this sample with the lowest energy (in practice, these elements are not from the ground states but from low excited states). The likelihood of obtaining solutions with a lower energy (or the actual ground state) increases with the number of repetitions.

As already mentioned, qubits on the D-Wave’s chip are arranged into a Chimera graph topology. Furthermore, some nodes and edges may be missing on the physical device, making the topology different even from an ideal Chimera graph. This requires *minor embedding* of the problem, mapping logical qubits onto physical ones. To this end, multiple physical qubits are chained together to represent a single logical variable, which increases their connectivity at the cost of the number of available qubits. Such embedding is performed by introducing an additional *penalty term* that favors states in which the quantum bits in each chain are aligned in the same direction. The multiplicative factor governing this process is called the chain strength, and it should dominate all the coefficients present in the original problem. (Note that we encounter yet another penalty term at this point.)

In this work, we set this factor to the maximum absolute value of the coefficients of the original problem multiplied by a parameter that we call the *chain strength scale* (*css*). In our experiment, *css* ranged from 2.0 to 9.0. Another parameter is the annealing time (ranging from 5 to 2000 μ s). This is the actual duration of the physical annealing process.

The data flow after performing the calculation on the quantum hardware is the following. The raw solution returned by the hardware consists of the configurations of the physical spins or QUBO variables depending on the format of the submitted problem, the corresponding energies, multiplicities, and other technical parameters. Therefore, the solution has to be transformed to logical variables by reverting the embedding. The physical variables representing the same logical one have equal values ideally; however, in reality, a *chain break* can occur: some of them may have different values. These are resolved by “majority voting”. The transformation between Ising and QUBO variables and the conversion of the logical variables are all implemented in the software package supplied by D-Wave (even though the raw data are also available), so having submitted the QUBO, one obtains 0–1 solution vectors along with energies and multiplicities. From these, the *conflict free* timetable can be decoded, as already shown in Section 3.

In Figure 7c,d, we present the energies of the best outcomes of the D-Wave machine for line No. 216 and various annealing times. The green dots denote the feasible solutions (and equivalent to the optimal solution), while the red dots denote solutions that are not feasible. In general, the quality of a solution slightly improves with the annealing time; however, in large examples, the best results are achieved for a time between 1000 and 2000 μ s. This coincides with the observation in [58], in which quantum annealing on the D-Wave machine was performed on various problems too, and it was demonstrated that for a moderate problem size, the performance (in terms of the probability of success) improves with an annealing time of up to 1000 μ s. Hence, we have limited ourselves to the annealing times of the order of magnitude of 1000 μ s in analyzing larger examples.

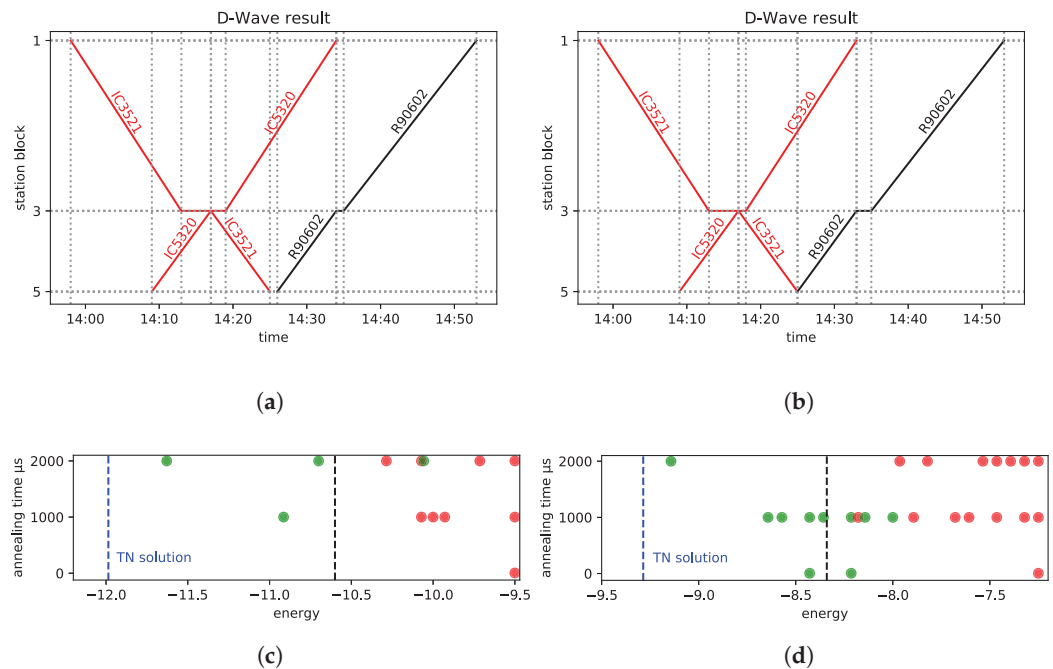


Figure 7. Train diagrams of the best D-Wave solutions, the lowest energies of the quantum annealing on the D-Wave machine (green: feasible, red: not feasible), and the optimal tensor network solution. The raw computational time on the D-Wave ($n.o. \text{ runs} \times \text{annealing time}$) was in the range 5×10^{-3} –2 s. (a) The optimal solution from (c). (b) The optimal solution from (d). (c) $p_{\text{pair}} = 2.7$, $p_{\text{sum}} = 2.2$. (d) $p_{\text{pair}} = p_{\text{sum}} = 1.75$.

Rather counterintuitively, setting lower penalty coefficients of $p_{\text{sum}} = p_{\text{pair}} = 1.75$ for the hard constraints resulted in samples containing more feasible solutions. For this reason,

we had kept this penalty setting for the analysis of the larger case. The embedding strength was set to $css = 2$ in this case, i.e., the lowest possible value. This has proven to be a good choice, as demonstrated in Figure 8. The best D-Wave solutions are presented in the form of train diagrams in Figure 7a,b.

The quality of the solutions in relation to the css strength in the various parameter settings is presented in Figure 8. We observed that in our cases, the quality of the solution degraded with an increase in css . This is unusual, as increasing the css strength typically yields more solutions without broken chains that do not need to be post-processed to obtain a feasible solution of the original problem. This may be caused by the fact that the large coupling of the embedding may cause the constraints to appear as a small perturbation in the physical QUBO. These perturbations, as discussed earlier, may be hidden in the noise of the D-Wave 2000Q annealer.

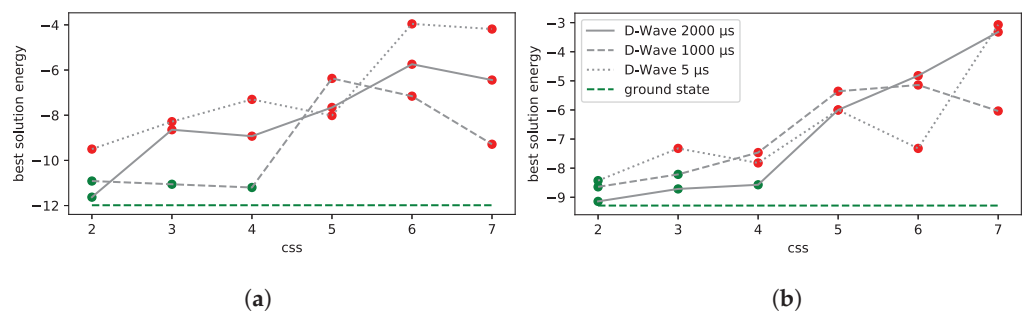


Figure 8. Line No. 216, with the minimal energy from the D-Wave quantum annealer, using 1000 runs. Green dots indicates the feasible solutions, while the red dots denote the unfeasible ones. In general, the energy rises as the css strength rises. We do not observe that the different settings of p_{pair} and p_{sum} improve the feasibility; see (a) The minimal energies vs. css for $p_{pair} = 2.2, p_{sum} = 2.7$. (b) The minimal energies vs. css for $p_{pair} = 1.75, p_{sum} = 1.75$.

Hence, we set $css = 2.0$ (the lowest possible value) for the further investigations. Some examples of the penalty and objective function values are presented in Table 3. Again, it appears that the higher the values of p_{sum} and p_{pair} , the higher the values of $f(x)$. This may be caused by the objective function being lost in the noise of the D-Wave 2000Q annealer.

For railway line No. 191, finding a feasible solution is more difficult. Hence, we increased the number of samples to 250,000. The results of the lowest energies and penalties are presented in Figure 9. We had to skip case 3 because the higher number of feasibility constraints prevented finding any embedding on a real Chimera. Interestingly, recall that we found the embedding for the ideal Chimera while simulating the solution (see Section 4.3.2). Hence, the failure in the case of the real graph is possibly due to the lack of certain required connections or nodes from the real Chimera. Finding the feasible solution in such a case (while having non-zero hard constraints penalties) is a problem for further research. One would expect that increasing the p_{pair} and p_{sum} parameters could be helpful. However, it may aggravate the objective function to an ever greater extent. In Figure 9b, the values of the objective function $f(x)$ are much higher than the optimal ones presented in Table 2.

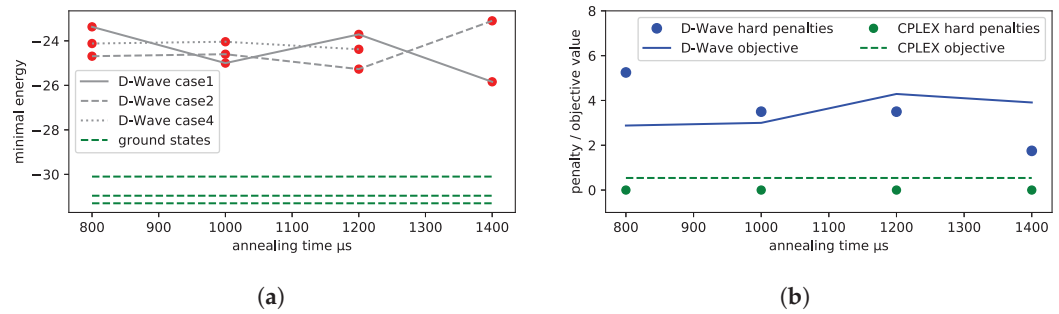


Figure 9. Line No. 191, with the minimal energy from the D-Wave annealer at 250 k runs, $css = 2.0$, and $p_{pair} = 1.75, p_{sum} = 1.75$. The output does not depend on the annealing time (in the investigated range) and is still far from the ground state. The raw computational time on the D-Wave (n.o. runs \times annealing time) was in the range 200–350 s. (a) Best D-Wave solutions (these are the lowest excited states we have recorded). Red dots indicate that the solutions are not feasible. (b) Comparison of the objective and hard penalty for the D-Wave outcome and the optimal solution calculated with CPLEX.

Table 3. Line No. 216, with the objective functions and penalties for violating the hard constraints: see Equation (S39) of Supplemental Materials. Output from the D-Wave quantum annealer for the annealing time of 2000 μs . If $f'(x) > 0$, the solution is not feasible. The $p_{sum} = p_{pair} = 1.75$ policy gives lower objectives.

css	p_{sum}, p_{pair}	Hard Constraints' Penalty $f'(x)$	$f(x)$
2	1.75, 1.75	0.0	1.36
2	2.2, 2.7	0.0	1.57
4	1.75, 1.75	0.0	1.93
4	2.2, 2.7	2.2	2.07
6	1.75, 1.75	5.25	0.43
6	2.2, 2.7	6.6	0.86

Although the solutions are not feasible, we can still select the two in which only one hard constraint is violated ($f'(x) = 1.75$); these are case 1 with an annealing time of 1400 μs and case 2 with an annealing time of 1200 μs . The train diagrams of these solutions are presented in Figure 10. Note that both these diagrams can easily be modified by the dispatcher to obtain a feasible solution. The case in Figure 10a can be amended by adding the lacking 1-min stay of Ks3 in station 7. This solution would not be optimal, and thus, it would be different from the optimal one obtainable with CPLEX, the tensor network solver, or FLFS. It would also differ from the non-optimal yet feasible ones returned by FCFS and AMCC. Similarly, the case in Figure 10b can be upgraded by shortening the stays of Ks3 and IC2 and letting them meet and pass at station 10. The so-obtained solution would be optimal.

At this point, a comment on the degeneracy of the ground state is in order. Clearly, in the instances related to the railway line 191, we are facing degenerate ground states. The reason for this is that in our model in Equation (27), the delay is penalized in the objective at the end of the trains' routes. Hence, there are many possibilities for trains to wait on various stations to meet the dispatching conditions. These choices lead to the same ground-state energy. What quantum annealing provides is a sample of the low-energy spectrum, possibly involving some of these ground states. In our case, however, the just mentioned intuitive interpretation of the solution enables us to find a practically useful and close-to-optimal solution.

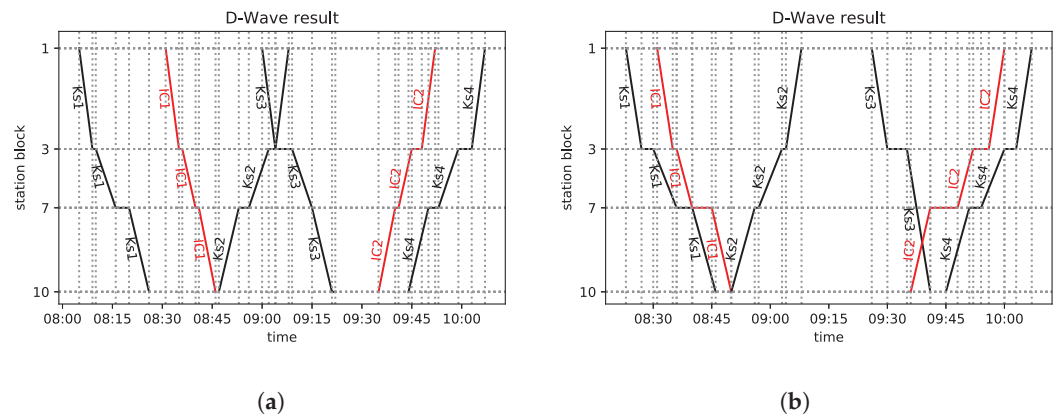


Figure 10. The best solutions obtained from the D-Wave quantum annealer for line No. 191. For case 1 (a), the annealing time is $t = 1400$. The solution is unfeasible since the stay of Ks3 at station 7 is below 1 min. If the solution is corrected (i.e., the stay is introduced), it loses its optimality and reflects a dispatching situation different from those obtained from FCFS, FLFS, AMCC, CPLEX, or the tensor network. For case 2 (b), $t = 1200$ is used. The solution is unfeasible as Ks3 does not stop at station 7; hence, Ks3 and IC2 are supposed to meet and pass between stations 7 and 10. It can, however, be amended to an optimal solution: shortening the stay of Ks3 at station 3 and shortening the stay of IC2 at station 7 (and 3 if necessary) result in a meet-and-pass situation at station 10, and this is optimal. (a) Case 1. (b) Case 2.

The real D-Wave quantum annealing is tied to some parameters of both the particular QUBO and the machine itself. In our experiments, we varied only the annealing time, number of performed reads and the chain strength scale (css), leaving all other parameters at their default values. We achieved the best results for a coupling constant $css = 2.0$ for the small example in Figure 8a; the same observation was made for the large example. This was not expected, as the coupling between quantum bits representing a single classical bit was rather weak. Here, we probably took advantage of the possible variations within the realization of a logical bit. This observation demonstrates that the embedding selection may be meaningful in searching for the convergence toward proper solutions lying in the low-energy part of the spectrum. For the small cases, we observed a feasible solution for a relatively small number of samples (equal to 1 k). For the larger case, we increased the number of samples to 250 k and still we did not reach any feasible solution. The conclusion is that the impact of the noise amplifies strongly with the size of the problem. The convergence of the best obtained solution toward the optimal one with the given sample size is complex, and an in-depth statistical analysis that samples the annealer’s real distribution is required.

As demonstrated in Figure 9b, in some cases, only a single hard constraint was violated. This may suggest that we are near the region of feasible solutions. However, the objective function values are still far from the optimal ones achieved by means of simulations (see Table 2). To elucidate the interplay between penalties, we refer to Figure 10, in which the solutions are not feasible but can be easily corrected by the dispatcher to obtain feasible ones. In Figure 10b the corrected solution would be optimal, while in Figure 10a, it would not be different from all the other achieved solutions. Hence, the current quantum annealer would rather sample the excited part of the QUBO spectrum, which can lead to unusual solutions. Such solutions, however, can be still be used by the dispatcher for some particular reason not encoded directly in the model. Such reasons include unexpected dispatching problems, rolling stock emergency, and non-standard requirements.

Let us also mention the characteristics of our QUBO problems as they are important features from the point of view of quantum methods. Table 4 summarizes the problem sizes and the densities of edges in the case of each problem instance.

Table 4. Graph densities for various problems. Since case 3 is supposed to be the most complicated one of cases 1–4, it has the largest graph density, #—number of.

Features	Line 216			Line 191		
	Case 1	Case 2	Case 3	Case 4	Enlarged	
problem size (# logical bits)	48	198	198	198	198	594
# edges	395	1851	2038	2180	1831	5552
density (vs. full graph)	0.35	0.095	0.104	0.111	0.094	0.032
embedding into	Chimera	Chimera	Chimera	Ideal Chimera	Chimera	Pegasus
approximate # physical bits	373	<2048	<2048	≈ 2048	<2048	<5760

4.4. Initial Studies on the D-Wave Advantage Machine

During the preparation of the present paper, a new quantum device, the D-Wave’s Advantage_system1.1 system (with an underlying topology code-named Pegasus [11]), became commercially available. Hence, we performed preliminary experiments with this new architecture to address a slightly larger example. To that end, we expanded our initial Golezów–Wiśła Uzdrowisko (line No. 191) problem instance to be 3 times bigger in size. Furthermore, we investigated nine trains in each direction.

The conflicts were introduced by assuming delays of 20, 25, or 30 min of certain trains entering block Section 1. The control parameters’ values $p_{\text{sum}} = p_{\text{pair}} = 1.75$ and $\text{css} = 2$ were not changed. As a result, the problem was mapped onto a QUBO with 594 variables and 5552 connections. The physical topology of the new system is different from that of its predecessor, so a different embedding was needed. This did not have any fundamental implications in our case; hence, we do not discuss its details here. Employing a strategy similar to the one used for our other calculations, we used the solution found by CPLEX as a reference for comparisons.

After performing 25 k runs, we reached a minimal energy of +75.28 with an annealing time of 1400 μs (the raw computational time on the D-Wave machine was 35 s). Unfortunately, this is not a feasible solution. The CPLEX calculations, on the other hand, resulted in an energy of −92.43 with an objective function value $f(x) = 2.07$ (see Figure 11). This is the same solution as the solution of the linear solver obtained using COIN-OR in 0.02 s. This solution is substantially better, and as it coincides with the linear solver’s output, it corresponds to the ground state. Our preliminary experiments indicate the need for a more detailed investigation of the new device’s behavior (and that of the current model) to determine whether obtaining solutions with the desired (better) quality is possible. A part of this problem will likely be eliminated simply by the technological development of the new annealer. For an intuitive justification, we refer to [78] and Figure 1 therein, in which an improvement in the performance between subsequent iterations within one generation of Chimera-based quantum annealers was observed. As discussed in Section 3.2, the number of logical bits (variables) and number of edges (quadratic terms) scales roughly as number of trains · (number of stations − 1) for constant d_{max} . Using this approximation, and referring to ref Table 4, for twice as many trains as in Figure 11 (roughly whole day of operation), we would have approximately 1200 logical bits and 11,000 edges. If we further enlarge the problem to the whole branch line (n.o. 157, 190, and 191) with eight stations, there would be approximately 2800 logical bits and 26000 edges.

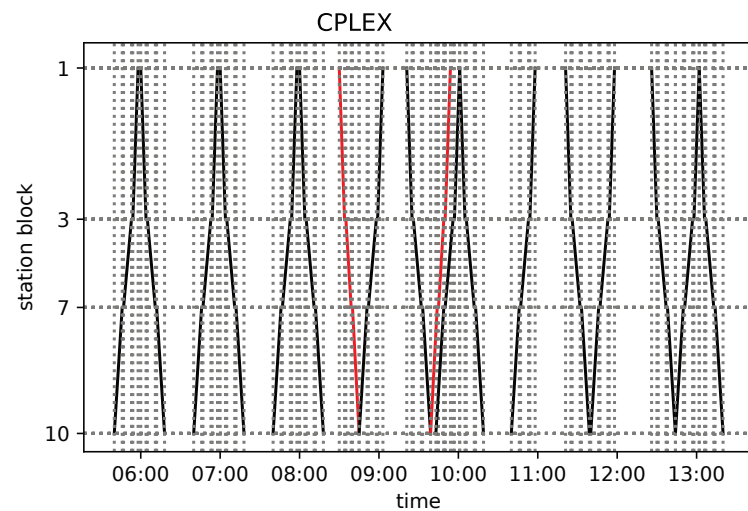


Figure 11. The CPLEX QUBO solution, coinciding with the linear model’s solution of the 18-train problem.

5. Discussion and Conclusions

The NISQ era [10] is here. Early generations of quantum computing hardware have become available that may serve as a stepping stone for the development of practically useful technologies that exhibit genuine quantum advantage. However, until such “real” quantum computers are available, it is imperative to demonstrate how and which real-world applications are amendable to be solved on quantum computing hardware. To this end, we have introduced a new approach to the single-track line dispatching problem that can be implemented on a real quantum annealing device (D-Wave 2000Q). Namely, we have addressed two particular real-life railway dispatching problems in Poland; many similar examples exist in other networks, too. Specifically, we have introduced a QUBO model of the problem that can be solved with quantum annealing, and we have benchmarked it against classical algorithms.

The first dispatching problem we considered (the Nidzica–Olsztynek section of line No. 216) was particularly small, and it was defined using only 48 logical quantum bits (which we were able to embed into 373 physical quantum bits of a real quantum processor). The final state reached by the quantum annealer for this problem was optimal for many parameter settings. This highlights that small-sized dispatching problems are already within reach of near-term quantum annealers. In addition, the limited size of the problem made it possible to analyze the QUBO with a greedy brute-force search algorithm, which revealed details of the behavior of the spectrum that cannot be exactly calculated for bigger instances.

Our second set of dispatching problems (the Goleiszów–Wisła Uzdrowisko section of line No. 191) was larger and needed 198 logical quantum bits. Here, the number of physical quantum bits depends on the number of constraints in each of the analyzed cases. We were able to embed all four dispatching cases of the No. 191 railway line into an ideal Chimera graph (2048 physical quantum bits) using a state-of-the-art embedding algorithm. We succeeded in solving these instances with classical solvers for QUBOs. Meanwhile, on the physical device (whose graph is not perfect and lacks several quantum bits and couplings), we were able to embed only three out of the four cases (case 3, with the highest number of conflicts, could not be embedded). We expect that such obstacles will become less restrictive as new embedding algorithms are being developed for both the current Chimera topology and the newest D-Wave Pegasus; see [88,89]. Therefore, it is not unreasonable to expect that the range of problems that can be embedded so that they can be solved on physical hardware will substantially increase in the near future. Unfortunately, the D-Wave 2000Q solutions of our second problem appeared to be far from optimal. This is attributable to the noise that is still present in the current quantum machine.

We have successfully solved our model using several algorithms for QUBOs running on classical computers, notably the novel tensor network method. This introduces additional possibilities, namely, that of QUBO modeling and the use of quantum-motivated classical algorithms. Although these possibilities obviously do not promise a breakthrough in scalability, they are essential for the validation and assessment of the results of real quantum annealing. In addition, they can yield practically useful results. In fact, hybrid quantum-classical computing is a promising avenue of research, which has recently seen significant development [90,91].

We are aware that the examples of the single-track railway dispatching problem discussed in the paper can be regarded as trivial from the point of view of professional dispatchers. This is also reflected by the efficiency of the conventional linear solver they may use. Our intention, however, was to provide a proof-of-concept demonstration of the applicability of quantum annealing in this field. This goal has been achieved: we have described a suitable model and succeeded in solving certain instances.

Due to the small size of the current quantum annealing processors, our implementation is limited: quantum annealing is an emerging technology. Owing to the significant efforts put into the development of quantum annealers, the addressable problem sizes are about to increase, and the quality of the samples will also improve. With the development of the technology, it is not far-fetched to realize that at some point, soon quantum annealers will be able to compete with or even outperform classical solvers. In particular, hybrid quantum-classical algorithms applied to the here presented type of model may even reach the size and complexity of the limitations of the state-of-the-art classical algorithms.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/e25020191/s1>. Figure S1: The conflicted timetables, various types of conflicts.; Figure S2: The FCFS solutions, some with a trouble-causing stopover of a particular train.; Figure S3: The FLFS solutions, some with a trouble-causing stopover of a particular train.; Figure S4: The AMCC solutions.; Figure S5: The CPLEX solutions: exact ground states of the QUBOs.; Figure S6: The tensor network solutions; Table S1: Notation summary.

Author Contributions: Conceptualization, K.D., M.K., K.K. and B.G.; Methodology, K.D., M.K., K.J. and S.D.; Software, K.D. and K.J.; Validation, M.K., K.J. and S.D.; Formal analysis, K.D., M.K. and B.G.; Investigation, K.D., M.K., K.K., K.J., S.D. and B.G.; Resources, K.K., K.J. and B.G.; Data curation, K.K. and K.J.; Writing—original draft, K.D., M.K., K.K. and B.G.; Writing—review & editing, K.D., M.K., K.K., S.D. and B.G.; Visualization, K.K. and K.J.; Supervision, K.D., S.D. and B.G.; Project administration, K.D.; Funding acquisition, S.D. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the National Research, Development, and Innovation Office of Hungary under project numbers K133882 and K124351 (M.K.); and the Foundation for Polish Science (FNP) under grant number TEAM NET POIR.04.04.00-00-17C1/18-00 (K.D. and B.G.); and the National Science Centre (NCN), Poland, under project number 2016/22/E/ST6/00062 (K.J.); and by the Silesian University of Technology Rector's Grant no. BKM-745/RT2/2021 12/020/BKM-2021/0213 and BKM-700/RT2/2022 12/020/BKM2022/0233 (K.K.).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The authors will supply data for any reasonable demand.

Acknowledgments: M.K. acknowledges the support of the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary. We gratefully acknowledge the support of NVIDIA Corporation, which donated the Titan V GPU used for this research. We acknowledge the cooperation with Koleje Śląskie sp. z o.o. (eng. Silesian Railways) and appreciate the valuable and substantive discussions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Raymer, M.G.; Monroe, C. The US National Quantum Initiative. *Quantum Sci. Technol.* **2019**, *4*, 020504. [CrossRef]
2. Riedel, M.; Kovacs, M.; Zoller, P.; Mlynek, J.; Calarco, T. Europe's Quantum Flagship initiative. *Quantum Sci. Technol.* **2019**, *4*, 020501. [CrossRef]
3. Yamamoto, Y.; Sasaki, M.; Takesue, H. Quantum information science and technology in Japan. *Quantum Sci. Technol.* **2019**, *4*, 020502. [CrossRef]
4. Sussman, B.; Corkum, P.; Blais, A.; Cory, D.; Damascelli, A. Quantum Canada. *Quantum Sci. Technol.* **2019**, *4*, 020503. [CrossRef]
5. Roberson, T.M.; White, A.G. Charting the Australian quantum landscape. *Quantum Sci. Technol.* **2019**, *4*, 020505. [CrossRef]
6. Sanders, B.C. *How to Build a Quantum Computer*; IOP Publishing: Bristol, UK, 2017; pp. 2399–2891. [CrossRef]
7. Aiello, C.D.; Awschalom, D.D.; Bernien, H.; Brower, T.; Brown, K.R.; Brun, T.A.; Caram, J.R.; Chitambar, E.; Felice, R.D.; Edmonds, K.M.; et al. Achieving a quantum smart workforce. *Quantum Sci. Technol.* **2021**, *6*, 030501. [CrossRef]
8. Roberson, T.; Leach, J.; Raman, S. Talking about public good for the second quantum revolution: analysing quantum technology narratives in the context of national strategies. *Quantum Sci. Technol.* **2021**, *6*, 025001. [CrossRef]
9. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef]
10. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
11. Dattani, N.; Szalay, S.; Chancellor, N. Pegasus: The second connectivity graph for large-scale quantum annealing hardware. *arXiv* **2019**, arXiv:1901.07636.
12. Więckowski, A.; Deffner, S.; Gardas, B. Disorder-assisted graph coloring on quantum annealers. *Phys. Rev. A* **2019**, *100*, 062304. [CrossRef]
13. Sax, I.; Feld, S.; Zielinski, S.; Gabor, T.; Linnhoff-Popien, C.; Mauerer, W. Approximate approximation on a quantum annealer. In Proceedings of the 17th ACM International Conference on Computing Frontiers, Catania, Italy, 11–13 May 2020; pp. 108–117.
14. Stollenwerk, T.; O'Gorman, B.; Venturelli, D.; Mandrà, S.; Rodionova, O.; Ng, H.; Sridhar, B.; Rieffel, E.G.; Biswas, R. Quantum Annealing Applied to De-Conflicting Optimal Trajectories for Air Traffic Management. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 285–297. [CrossRef]
15. Domino, K.; Koniarczyk, M.; Krawiec, K.; Jałowicki, K.; Gardas, B. Quantum computing approach to railway dispatching and conflict management optimization on single-track railway lines. *arXiv* **2020**, arXiv:2010.08227.
16. Grozea, C.; Hans, R.; Koch, M.; Riehn, C.; Wolf, A. Optimising Rolling Stock Planning including Maintenance with Constraint Programming and Quantum Annealing. *arXiv* **2021**, arXiv:2109.07212.
17. Yarkoni, S.; Huck, A.; Schülldorf, H.; Speitkamp, B.; Tabrizi, M.S.; Leib, M.; Bäck, T.; Neukart, F. Solving the Shipment Rerouting Problem with Quantum Optimization Techniques. In *Lecture Notes in Computer Science*; Springer International Publishing: Cham, Switzerland, 2021; pp. 502–517. [CrossRef]
18. Cacchiani, V.; Huisman, D.; Kidd, M.; Kroon, L.; Toth, P.; Veelenturf, L.; Wagenaar, J. An overview of recovery models and algorithms for real-time railway rescheduling. *Transp. Res. Part B Methodol.* **2014**, *63*, 15–37. [CrossRef]
19. Törnquist, J.; Persson, J.A. N-tracked railway traffic re-scheduling during disturbances. *Transp. Res. Part B Methodol.* **2007**, *41*, 342–362. [CrossRef]
20. Lamorgese, L.; Mannino, C.; Pacciarelli, D.; Krasemann, J.T. Handbook of Optimization in the Railway Industry. In *Train Dispatching*; Springer International Publishing: Cham, Switzerland, 2018; pp. 265–283. [CrossRef]
21. Jensen, J.; Nielsen, O.; Prato, C. Passenger Perspectives in Railway Timetabling: A Literature Review. *Transp. Rev.* **2016**, *36*, 500–526. [CrossRef]
22. Wen, C.; Huang, P.; Li, Z.; Lessan, J.; Fu, L.; Jiang, C.; Xu, X. Train Dispatching Management With Data-Driven Approaches: A Comprehensive Review and Appraisal. *IEEE Access* **2019**, *7*, 114547–114571. [CrossRef]
23. Van Leeuwen, J. *Handbook of Theoretical Computer Science (vol. A) Algorithms and Complexity*; MIT Press: Cambridge, MA, USA, 1991.
24. Cai, X.; Goh, C.J. A fast heuristic for the train scheduling problem. *Comput. Oper. Res.* **1994**, *21*, 499–510. [CrossRef]
25. Szpigel, B. Optimal train scheduling on a single line railway. *Oper. Res.* **1973**, *72*, 343–352.
26. Pinedo, M.L. *Scheduling: Theory, Algorithms, and Systems*, 3rd ed.; Springer Publishing Company, Incorporated: Berlin/Heidelberg, Germany, 2008.
27. Cordeau, J.F.; Toth, P.; Vigo, D. A Survey of Optimization Models for Train Routing and Scheduling. *Transp. Sci.* **1998**, *32*, 380–404. [CrossRef]
28. Törnquist, J. Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. In Proceedings of the 5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05), Schloss Dagstuhl-Leibniz-Zentrum für Informatik, Palma de Mallorca, Spain, 14 September 2005.
29. Dollevoet, T.; Huisman, D.; Schmidt, M.; Schöbel, A. Delay Propagation and Delay Management in Transportation Networks. In *Handbook of Optimization in the Railway Industry*; Springer International Publishing: Cham, Switzerland, 2018; pp. 285–317. [CrossRef]
30. Corman, F.; Meng, L. A Review of Online Dynamic Models and Algorithms for Railway Traffic Management. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1274–1284. [CrossRef]
31. Cacchiani, V.; Toth, P. Nominal and robust train timetabling problems. *Eur. J. Oper. Res.* **2012**, *219*, 727–737. [CrossRef]

32. Hansen, I. State-of-the-art of railway operations research. In *Timetable Planning and Information Quality*; WIT Press: Wessex, UK, 2010; pp. 35–47.
33. Lange, J.; Werner, F. Approaches to modeling train scheduling problems as job-shop problems with blocking constraints. *J. Sched.* **2018**, *21*, 191–207. [CrossRef]
34. Mascis, A.; Pacciarelli, D. Job-shop scheduling with blocking and no-wait constraints. *Eur. J. Oper. Res.* **2002**, *143*, 498–517. [CrossRef]
35. D’Ariano, A.; Pacciarelli, D.; Pranzo, M. A branch and bound algorithm for scheduling trains in a railway network. *Eur. J. Oper. Res.* **2007**, *183*, 643–657. [CrossRef]
36. Venturelli, D.; Marchand, D.J.J.; Rojo, G. Quantum Annealing Implementation of Job-Shop Scheduling. *arXiv* **2015**, arXiv:1506.08479.
37. Zhou, X.; Zhong, M. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transp. Res. Part B Methodol.* **2007**, *41*, 320–341. [CrossRef]
38. Harrod, S. Modeling Network Transition Constraints with Hypergraphs. *Transp. Sci.* **2011**, *45*, 81–97. [CrossRef]
39. Meng, L.; Zhou, X. Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables. *Transp. Res. Part B Methodol.* **2014**, *67*, 208–234. [CrossRef]
40. Kadowaki, T.; Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **1998**, *58*, 5355–5363. [CrossRef]
41. Aharonov, D.; van Dam, W.; Kempe, J.; Landau, Z.; Lloyd, S.; Regev, O. Adiabatic quantum computation is equivalent to standard quantum computation. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy, 17–19 October 2004; pp. 42–51. [CrossRef]
42. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*; Cambridge University Press: Cambridge, UK, 2010. [CrossRef]
43. Biamonte, J.D.; Love, P.J. Realizable Hamiltonians for universal adiabatic quantum computers. *Phys. Rev. A* **2008**, *78*, 012352. [CrossRef]
44. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2014**, *2*, 5. [CrossRef]
45. Albash, T.; Lidar, D.A. Adiabatic quantum computation. *Rev. Mod. Phys.* **2018**, *90*, 015002. [CrossRef]
46. Lanting, T.; Przybysz, A.J.; Smirnov, A.Y.; Spedalieri, F.M.; Amin, M.H.; Berkley, A.J.; Harris, R.; Altomare, F.; Boixo, S.; Bunyk, P.; et al. Entanglement in a quantum annealing processor. *Phys. Rev. X* **2014**, *4*, 021041. [CrossRef]
47. Wu, F.Y. The Potts model. *Rev. Mod. Phys.* **1982**, *54*, 235–268. [CrossRef]
48. Castellani, T.; Cavagna, A. Spin-glass theory for pedestrians. *J. Stat. Mech.* **2005**, *2005*, P05012. [CrossRef]
49. Glover, F.; Kochenberger, G.; Du, Y. Quantum Bridge Analytics I: A tutorial on formulating and using QUBO models. *Ann. J. Oper. Res.* **2019**, *17*, 335–371. [CrossRef]
50. Aramon, M.; Rosenberg, G.; Valiante, E.; Miyazawa, T.; Tamura, H.; Katzgraber, H.G. Physics-Inspired Optimization for Quadratic Unconstrained Problems Using a Digital Annealer. *Front. Phys.* **2019**, *7*, 48. [CrossRef]
51. Pierangeli, D.; Rafayelyan, M.; Conti, C.; Gigan, S. Scalable spin-glass optical simulator. *arXiv* **2020**, arXiv:2006.00828.
52. Yamamoto, Y.; Aihara, K.; Leleu, T.; Kawarabayashi, K.; Kako, S.; Fejer, M.; Inoue, K.; Takesue, H. Coherent Ising machines—Optical neural networks operating at the quantum limit. *Npj Quantum Inf.* **2017**, *3*, 49. [CrossRef]
53. Fukushima-Kimura, B.H.; Handa, S.; Kamakura, K.; Kamijima, Y.; Sakai, A. Mixing time and simulated annealing for the stochastic cellular automata. *arXiv* **2020**, arXiv:2007.11287.
54. Cai, F.; Kumar, S.; Van Vaerenbergh, T.; Sheng, X.; Liu, R.; Li, C.; Liu, Z.; Foltin, M.; Yu, S.; Xia, Q.; et al. Power-efficient combinatorial optimization using intrinsic noise in memristor Hopfield neural networks. *Nat. Electron.* **2020**, *3*, 409–418. [CrossRef]
55. Avron, J.E.; Elgart, A. Adiabatic Theorem without a Gap Condition. *Commun. Math. Phys.* **1999**, *203*, 445–463. [CrossRef]
56. Ozfidan, I.; Deng, C.; Smirnov, A.Y.; Lanting, T.; Harris, R.; Swenson, L.; Whittaker, J.; Altomare, F.; Babcock, M.; Baron, C.; et al. Demonstration of nonstoquastic Hamiltonian in coupled superconducting flux qubits. *arXiv* **2019**, arXiv:1903.06139.
57. Choi, V. Minor-embedding in adiabatic quantum computation: I. The parameter setting problem. *Quantum Inf. Process.* **2008**, *7*, 193–209. [CrossRef]
58. Hamerly, R.; Inagaki, T.; McMahon, P.L.; Venturelli, D.; Marandi, A.; Onodera, T.; Ng, E.; Langrock, C.; Inaba, K.; Honjo, T.; et al. Experimental investigation of performance differences between coherent Ising machines and a quantum annealer. *Sci. Adv.* **2019**, *5*, aau0823. [CrossRef]
59. King, A.D.; Bernoudy, W.; King, J.; Berkley, A.J.; Lanting, T. Emulating the coherent Ising machine with a mean-field algorithm. *arXiv* **2018**, arXiv:1806.08422v1.
60. Onodera, T.; Ng, E.; McMahon, P.L. A quantum annealer with fully programmable all-to-all coupling via Floquet engineering. *arXiv* **2019**, arXiv:math-ph/0409035.
61. Childs, A.M.; Farhi, E.; Preskill, J. Robustness of adiabatic quantum computation. *Phys. Rev. A* **2001**, *65*, 012322. [CrossRef]
62. Katzgraber, H.G.; Hamze, F.; Zhu, Z.; Ochoa, A.J.; Munoz-Bauza, H. Seeking Quantum Speedup Through Spin Glasses: The Good, the Bad, and the Ugly. *Phys. Rev. X* **2015**, *5*, 031026. [CrossRef]
63. Sachdev, S. *Quantum Phase Transitions*; Cambridge University Press: Cambridge, UK, 2011.
64. Dziarmaga, J. Dynamics of a quantum phase transition: Exact solution of the quantum Ising model. *Phys. Rev. Lett.* **2005**, *95*, 245701. [CrossRef] [PubMed]

65. Dziarmaga, J. Dynamics of a quantum phase transition and relaxation to a steady state. *Adv. Phys.* **2010**, *59*, 1063–1189. [CrossRef]
66. Kibble, T.W.B. Topology of cosmic domains and strings. *J. Phys. A Math. Gen.* **1976**, *9*, 1387. [CrossRef]
67. Kibble, T.W.B. Some implications of a cosmological phase transition. *Phys. Rep.* **1980**, *67*, 183–199. [CrossRef]
68. Zurek, W.H. Cosmological experiments in superfluid helium? *Nature* **1985**, *317*, 505. [CrossRef]
69. Francuz, A.; Dziarmaga, J.; Gardas, B.; Zurek, W.H. Space and time renormalization in phase transition dynamics. *Phys. Rev. B* **2016**, *93*, 075134. [CrossRef]
70. Deffner, S. Kibble-Zurek scaling of the irreversible entropy production. *Phys. Rev. E* **2017**, *96*, 052125. [CrossRef]
71. Gardas, B.; Dziarmaga, J.; Zurek, W.H. Dynamics of the quantum phase transition in the one-dimensional Bose-Hubbard model: Excitations and correlations induced by a quench. *Phys. Rev. B* **2017**, *95*, 104306. [CrossRef]
72. Gardas, B.; Dziarmaga, J.; Zurek, W.H.; Zwolak, M. Defects in Quantum Computers. *Sci. Rep.* **2018**, *8*, 4539. [CrossRef]
73. Venuti, L.C.; Albash, T.; Lidar, D.A.; Zanardi, P. Adiabaticity in open quantum systems. *Phys. Rev. A* **2016**, *93*, 032118. [CrossRef]
74. Schollwöck, U. The density-matrix renormalization group. *Rev. Mod. Phys.* **2005**, *77*, 259–315. [CrossRef]
75. Verstraete, F.; Cirac, J.I. Matrix product states represent ground states faithfully. *Phys. Rev. B* **2006**, *73*, 094423. [CrossRef]
76. Rams, M.M.; Mohseni, M.; Eppens, D.; Jałowiecki, K.; Gardas, B. Approximate optimization, sampling, and spin-glass droplet discovery with tensor networks. *Phys. Rev. E* **2021**, *104*, 025308. [CrossRef] [PubMed]
77. Czartowski, J.; Szymański, K.; Gardas, B.; Fyodorov, Y.V.; Życzkowski, K. Separability gap and large-deviation entanglement criterion. *Phys. Rev. A* **2019**, *100*, 042326. [CrossRef]
78. Jałowiecki, K.; Więckowski, A.; Gawron, P.; Gardas, B. Parallel in time dynamics with quantum annealers. *Sci. Rep.* **2020**, *10*, 13534. [CrossRef]
79. Gardas, B.; Rams, M.M.; Dziarmaga, J. Quantum neural networks to simulate many-body quantum systems. *Phys. Rev. B* **2018**, *98*, 184304. [CrossRef]
80. Jałowiecki, K.; Rams, M.M.; Gardas, B. Brute-forcing spin-glass problems with CUDA. *Comput. Phys. Commun.* **2021**, *260*. [CrossRef]
81. Luenberger, D.; Ye, Y. *Linear and Nonlinear Programming*; International Series in Operations Research & Management Science; Springer International Publishing: Berlin/Heidelberg, Germany, 2015.
82. Hen, I.; Spedalieri, F.M. Quantum Annealing for Constrained Optimization. *Phys. Rev. Appl.* **2016**, *5*, 034007. [CrossRef]
83. DWave Ocean Software Documentation. Available online: <https://docs.ocean.dwavesys.com/en/stable> (accessed on 29 June 2020).
84. Gusmeroli, N.; Wiegele, A. EXPEDIS: An exact penalty method over discrete sets. *Discret. Optim.* **2022**, *44*, 100622.
85. PKP Polskie Linie Kolejowe S.A. Public Procurement Website. Available online: <https://zamowienia.plk-sa.pl/> (accessed on 3 February 2020)
86. CPLEX Optimizer. Available online: <https://www.ibm.com/analytics/cplex-optimizer> (accessed on 29 June 2020).
87. Optimization with PuLP. Available online: <https://coin-or.github.io/pulp> (accessed on 15 February 2021).
88. Zbinden, S.; Bärttschi, A.; Djidjev, H.; Eidenbenz, S. Embedding Algorithms for Quantum Annealers with Chimera and Pegasus Connection Topologies. In Proceedings of the International Conference on High Performance Computing, Frankfurt, Germany, 22–25 June 2020; pp. 187–206.
89. Pelofske, E.; Hahn, G.; Djidjev, H. Decomposition Algorithms for Solving NP-hard Problems on a Quantum Annealer. *J. Signal Process. Syst.* **2021**, *93*, 405–420. [CrossRef]
90. Endo, S.; Cai, Z.; Benjamin, S.C.; Yuan, X. Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation. *J. Phys. Soc. Jpn.* **2021**, *90*, 032001. [CrossRef]
91. Ding, Y.; Chen, X.; Lamata, L.; Solano, E.; Sanz, M. Implementation of a Hybrid Classical-Quantum Annealing Algorithm for Logistic Network Design. *SN Comput. Sci.* **2021**, *2*, 68. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance

Congcong Feng ^{1,2,†}, Bo Zhao ^{2,3,†}, Xin Zhou ², Xiaodong Ding ² and Zheng Shan ^{2,3,*}

¹ School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450002, China

² State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

³ Songshan Laboratory, Zhengzhou 450001, China

* Correspondence: zzzhengming@163.com

† These authors contributed equally to this work.

Abstract: The K-nearest neighbor (KNN) algorithm is one of the most extensively used classification algorithms, while its high time complexity limits its performance in the era of big data. The quantum K-nearest neighbor (QKNN) algorithm can handle the above problem with satisfactory efficiency; however, its accuracy is sacrificed when directly applying the traditional similarity measure based on Euclidean distance. Inspired by the Polar coordinate system and the quantum property, this work proposes a new similarity measure to replace the Euclidean distance, which is defined as Polar distance. Polar distance considers both angular and module length information, introducing a weight parameter adjusted to the specific application data. To validate the efficiency of Polar distance, we conducted various experiments using several typical datasets. For the conventional KNN algorithm, the accuracy performance is comparable when using Polar distance for similarity measurement, while for the QKNN algorithm, it significantly outperforms the Euclidean distance in terms of classification accuracy. Furthermore, the Polar distance shows scalability and robustness superior to the Euclidean distance, providing an opportunity for the large-scale application of QKNN in practice.

Keywords: quantum computation; quantum machine learning; K-nearest neighbor algorithm; quantum K-nearest neighbor algorithm

Citation: Feng, C.; Zhao, B.; Zhou, X.; Ding, X.; Shan, Z. An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance. *Entropy* **2023**, *25*, 127. <https://doi.org/10.3390/e25010127>

Academic Editors: Brian R. La Cour and Giuliano Benenti

Received: 4 December 2022

Revised: 4 January 2023

Accepted: 4 January 2023

Published: 8 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Machine learning has made remarkable achievements in various artificial intelligence applications, such as object detection [1–4], image classification [5–8], and natural language processing [9–11]. However, in the era of big data, we are facing the problem of rapid growth in the amount and type of data. We urgently need to find more high-efficiency computing methods. The quantum system with natural parallelism looks like a good choice. With the in-depth study of quantum technology, many quantum algorithms showing quantum superiority have been proposed [12–15]. Researchers found that quantum and machine learning algorithms can be combined to improve the performance of the algorithm. The concept of quantum machine learning was born [16,17]. Many quantum machine learning algorithms [18–22] are significantly better than their classical counterparts. In this context, a KNN algorithm with a simple idea but high time complexity has attracted the interest of researchers. It requires little to no prior knowledge when classifying [23]. Similarity calculation and K-nearest neighbors search are two important parts of KNN. In recent years, many quantum methods for these two processes have been proposed. In 2001, Harry Bruerman et al. proposed the swap test quantum circuit for calculating the cosine distance of two vectors [24]. In 2013, Lloyd et al. proposed a quantum Euclidean distance estimator based on the swap test circuit [25]. Based on this, Wiebe et al. proposed a quantum nearest neighbor algorithm [26] in 2014 and used Dürr and Høyer’s algorithm for finding the minimum value in a database [27] to find the nearest neighbor. For non-

numerical data, a quantum K-nearest neighbor algorithm based on Hamming distance is proposed [28,29].

The similarity measure, which affects the accuracy of the algorithm classification, lies at the heart of the K-nearest neighbor algorithm [30]. A similarity measure is used to measure how similar two things are [31]. To date, many similarity measures have been proposed, such as Euclidean distance, cosine distance, Hamming distance, and so on. However, there is no one similarity distance measure that can best solve all problems [31]. Choosing an appropriate similarity measure will significantly improve the K-nearest neighbor algorithm’s classification accuracy. The Euclidean distance is the most frequently applied similarity measure. However, the result of the quantum Euclidean distance estimator has poor stability, and there is a significant difference with the actual result [32]. Therefore, we need to find a new similarity measure to replace the use of Euclidean distance in QKNN.

In machine learning, a sample is usually regarded as a vector with both magnitude and direction. Inspired by this, in addition to using Cartesian coordinates, we can also use Polar coordinates to represent a sample. So, we propose a new similarity measure that we call Polar distance, which considers both angular and module length information. The cosine theorem shows that the Euclidean distance is a combination of angular and module length information. The Polar distance introduces an adjustable parameter to adjust the ratio of angular and module length information according to the specific application. Then, we propose a quantum circuit to calculate the Polar distance. The frame diagram of the quantum part of the QKNN algorithm is shown in Figure 1. We optimize Figure 1b in this work. The following is a list of our major contributions:

- (1) We propose a new similarity measure, called Polar distance, which integrates both angular and module length information and combines the two proportionally according to practical applications. Its classification accuracy in KNN is comparable to that of Euclidean distance;
- (2) We design a quantum circuit to calculate the Polar distance. Compared with the quantum Euclidean distance estimator, it can directly obtain the desired results and has less difference with the classical results;
- (3) We carry out KNN and QKNN(quantum simulation) experiments on different datasets. The KNN’s experimental results show that Polar distance is comparable to Euclidean distance in classification accuracy. The QKNN’s experimental results show that Polar distance is better than Euclidean distance in classification accuracy.

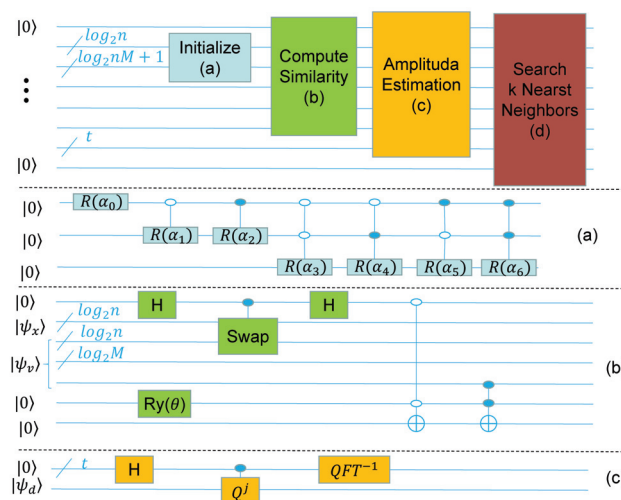


Figure 1. Frame diagram of quantum part of QKNN algorithm: (a) initialize any quantum state circuit, where α is the calculated parameter based on the quantum state vector representation; (b) quantum circuit for calculating Polar distance, where $|\psi_x\rangle$ represents the test sample and $|\psi_v\rangle(|\psi_v\rangle = |v_i\rangle|i\rangle|d_{r_i}\rangle)$ represents the entangled superposition state of the module length similarity and the training set; (c) amplitude estimation circuit diagram.

2. Materials and Methods

2.1. KNN

K-nearest neighbor algorithm is a supervised machine learning algorithm [23]. Its general idea is: if most of the K-nearest samples to a sample belong to a given category in the feature space, then the sample belongs to that category as well. The whole process of KNN is shown in Algorithm 1. Its main steps are as follows: first, calculate the similarity between the test sample and all training samples; then find the k training samples that are most like the test sample; finally, according to the category of k training samples, the category of the test sample is determined according to the principle that the minority obeys the majority. For example, as shown in Figure 2, the question mark represents the unknown test sample, and the red circle and blue cross denote two categories of training samples. At $k = 1$, the category of the question mark is consistent with the red circle category. When $k = 5$, the category of the question mark is consistent with the blue cross category. Obviously, the classification results will be affected by k value. Furthermore, similarity measure is another factor influencing the results of classification.

Algorithm 1 KNN

Input: A test sample and some training samples

Output: The test sample's category

- 1: **for** number of training samples **do**
 - 2: calculate the similarity between the test sample and a training sample
 - 3: **end for**
 - 4: find the k training samples that are most like the test sample
 - 5: determine test sample's category
-

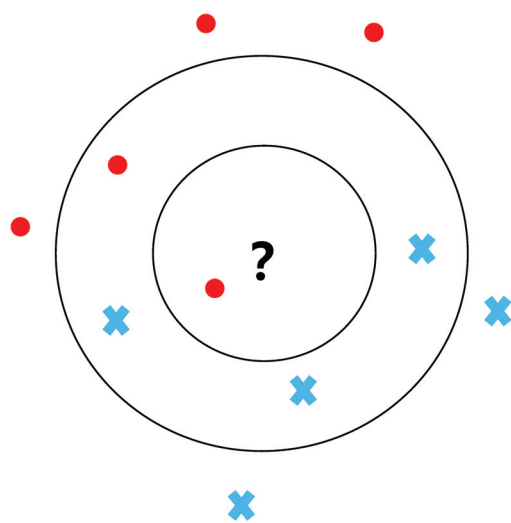


Figure 2. Schematic of the KNN algorithm.

2.2. QKNN

The quantum K-nearest neighbor algorithm is consistent with the overall idea of the classical K-nearest neighbor algorithm. Quantum K-nearest neighbor algorithm quantizes the part of K-nearest neighbor algorithm with high time complexity. It uses the natural parallelism of quantum computing to reduce the time complexity of the algorithm. As shown in Figure 1, the quantum part of the quantum K-nearest neighbor proposed in this paper consists of four parts: Initialize, Compute Similarity, Amplitude Estimation, and Search K-Nearest Neighbors. Finally, the test sample category is determined by the classical method. We describe these five parts in detail later.

2.2.1. Initialize

In order to process classical data using the quantum system, we need to encode classical data into quantum state. At present, there are many methods to encode classical data into quantum states [33–35]. The coding methods can be divided into two categories: using the amplitude of the quantum state to encode information and directly using the quantum state to encode information. Amplitude coding is one of the common coding methods in quantum machine learning algorithms [36]. In this paper, we also use amplitude coding. Its main idea is to use the amplitude of quantum states to represent classical data. In order to represent the classical vector by amplitude, we must first normalize the vector so that the vector module length is 1. After that, we should ensure that the dimension of the vector is 2^n , and n is the number of qubits required to encode the vector. When the vector does not meet this condition, it is completed by supplementing 0. Take vector \vec{a} as an example.

$$\vec{a} = (a_0, a_1, \dots, a_{2^n-1}) \tag{1}$$

Its quantum representation is as follows:

$$|\psi_a\rangle = \sum_{i=0}^{2^n-1} \frac{a_i}{\sqrt{|a_0|^2 + |a_1|^2 + \dots + |a_{2^n-1}|^2}} |i\rangle \tag{2}$$

Then, we need to initialize the register of n qubits as $|\psi_a\rangle$. The initial state is $|0 \dots 0\rangle$, we start from the high position, and the quantum circuit is shown in Figure 1a. The Ry represents single-qubit rotation about the Y-axis, the solid point represents 1 control, and the hollow point represents 0 control.

2.2.2. Compute Similarity

Inspired by Polar coordinates, we propose a parametric similarity measure that combines cosine similarity and module length similarity, which we call Polar distance.

$$d = d_c \cdot (1 - \omega) + d_r \cdot \omega \tag{3}$$

Among them, d_c and d_r represent cosine similarity and module length similarity, respectively, ω represents an adjustable parameter, and the value range of the three is $[0, 1]$. As d increases, the similarity between the two samples is stronger, otherwise this similarity becomes weaker. We can adjust the value of ω to improve the classification accuracy according to the actual application. The formula for calculating the cosine similarity is as follows (θ represents the angle between two vectors):

$$d_c = 0.5 \cdot (1 + \cos^2\theta) \tag{4}$$

The formula for calculating the module length similarity is as follows:

$$d_r = 1 - |r_x - r_v| \tag{5}$$

where r_x refers to the module length of the test sample and r_v represents the module length of the training sample. The closer the d_r value is to 1, the greater the similarity between the two samples, and the closer the d_r value is to 0, the smaller the similarity between the two samples.

As shown in Figure 1b, the similarity calculation is divided into two steps: first, the swap test circuit is applied to calculate the cosine similarity [24], and then the weighted summation circuit proposed by us is used to realize the weighted summation of the cosine similarity and the module length similarity.

Next, we take the calculation of the similarity between two samples x and v as an example to introduce the calculation process in detail. The initial state of the quantum system is:

$$|s_0\rangle = |0\rangle|x\rangle|v\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle + \sqrt{\omega}|1\rangle)|0\rangle \tag{6}$$

First of all, after a Hadamard gate is utilized, the state of the quantum system becomes:

$$|s_1\rangle = |+\rangle|x\rangle|v\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle + \sqrt{\omega}|1\rangle)|0\rangle \tag{7}$$

Then, after the usage of CSWAP gate, the state of the quantum system transforms into:

$$|s_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle|x\rangle|v\rangle + |1\rangle|v\rangle|x\rangle)(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle + \sqrt{\omega}|1\rangle)|0\rangle \tag{8}$$

At the third stage, another Hadamard gate is applied. The state of the quantum system is given as:

$$|s_3\rangle = \frac{1}{2}(|0\rangle|x\rangle|v\rangle + |1\rangle|x\rangle|v\rangle + |0\rangle|v\rangle|x\rangle - |1\rangle|v\rangle|x\rangle) (\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle + \sqrt{\omega}|1\rangle)|0\rangle \tag{9}$$

At the fourth stage, the extended general Toffoli gate is applied, and the state of the quantum system reads:

$$|s_4\rangle = \frac{1}{2}|0\rangle|x\rangle|v\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle|1\rangle + \sqrt{\omega}|1\rangle|0\rangle) + \frac{1}{2}|1\rangle|x\rangle|v\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle|0\rangle + \sqrt{\omega}|1\rangle|0\rangle) + \frac{1}{2}|0\rangle|v\rangle|x\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle|1\rangle + \sqrt{\omega}|1\rangle|0\rangle) + \frac{1}{2}|1\rangle|v\rangle|x\rangle(\sqrt{1-d_r}|0\rangle + \sqrt{d_r}|1\rangle)(\sqrt{1-\omega}|0\rangle|0\rangle + \sqrt{\omega}|1\rangle|0\rangle) \tag{10}$$

At the fifth stage, the Toffoli gate is applied, and the state of the quantum system becomes:

$$|s_5\rangle = \frac{1}{2}|0\rangle|x\rangle|v\rangle\sqrt{1-d_r}|0\rangle\sqrt{1-\omega}|0\rangle|1\rangle + \frac{1}{2}|0\rangle|x\rangle|v\rangle\sqrt{1-d_r}|0\rangle\sqrt{\omega}|1\rangle|0\rangle + \frac{1}{2}|0\rangle|x\rangle|v\rangle\sqrt{d_r}|1\rangle\sqrt{1-\omega}|0\rangle|1\rangle + \frac{1}{2}|0\rangle|x\rangle|v\rangle\sqrt{d_r}|1\rangle\sqrt{\omega}|1\rangle|1\rangle + \frac{1}{2}|1\rangle|x\rangle|v\rangle\sqrt{1-d_r}|0\rangle\sqrt{1-\omega}|0\rangle|0\rangle + \frac{1}{2}|1\rangle|x\rangle|v\rangle\sqrt{1-d_r}|0\rangle\sqrt{\omega}|1\rangle|0\rangle + \frac{1}{2}|1\rangle|x\rangle|v\rangle\sqrt{d_r}|1\rangle\sqrt{1-\omega}|0\rangle|0\rangle + \frac{1}{2}|1\rangle|x\rangle|v\rangle\sqrt{d_r}|1\rangle\sqrt{\omega}|1\rangle|1\rangle + \frac{1}{2}|0\rangle|v\rangle|x\rangle\sqrt{1-d_r}|0\rangle\sqrt{1-\omega}|0\rangle|1\rangle + \frac{1}{2}|0\rangle|v\rangle|x\rangle\sqrt{1-d_r}|0\rangle\sqrt{\omega}|1\rangle|0\rangle + \frac{1}{2}|0\rangle|v\rangle|x\rangle\sqrt{d_r}|1\rangle\sqrt{1-\omega}|0\rangle|1\rangle + \frac{1}{2}|0\rangle|v\rangle|x\rangle\sqrt{d_r}|1\rangle\sqrt{\omega}|1\rangle|1\rangle - \frac{1}{2}|1\rangle|v\rangle|x\rangle\sqrt{1-d_r}|0\rangle\sqrt{1-\omega}|0\rangle|0\rangle - \frac{1}{2}|1\rangle|v\rangle|x\rangle\sqrt{1-d_r}|0\rangle\sqrt{\omega}|1\rangle|0\rangle - \frac{1}{2}|1\rangle|v\rangle|x\rangle\sqrt{d_r}|1\rangle\sqrt{1-\omega}|0\rangle|0\rangle - \frac{1}{2}|1\rangle|v\rangle|x\rangle\sqrt{d_r}|1\rangle\sqrt{\omega}|1\rangle|1\rangle \tag{11}$$

Finally, the last qubit is measured. The probability of getting state $|1\rangle$, which measures the last qubit with the basis state $|1\rangle$, is given by:

$$p(|1\rangle) = \left(\frac{1}{2} + \frac{1}{2}|\langle x|v\rangle|^2\right)(1-\omega) + d_r\omega \tag{12}$$

2.2.3. Amplitude Estimation

There are two methods to obtain the results of similarity calculation of quantum circuits. The first method is to obtain statistical results through multiple measurements. The disadvantage of this method is that it cannot determine the accuracy of the results. The other method is to use an amplitude estimation algorithm [37]. Amplitude estimation can control the accuracy of the results by adjusting the number of qubits. In addition, the use of amplitude estimation is more convenient for subsequent quantum steps. So, we use amplitude estimation. In this article, we only introduce the usage of amplitude estimation. For more details, please refer to [37]. The function of amplitude estimation is to calculate the value of a in Equation (13). The circuit of amplitude estimation is shown in Figure 1c. The first step is to initialize the two registers with status $|0\rangle A|0\rangle$. The second step is to apply QFT to the first register. The third step is to apply a controlled Q^j ($Q = -AS_0A^{-1}S_X$). The fourth step is to apply QFT^{-1} to the first register. The fifth step is to measure the first register and denote the outcome $|y\rangle$. Finally, calculate the amplitude $a = \sin(\pi \frac{y}{2^t})$. We can control the precision of the result by adjusting the number of qubits t . The higher the value of t , the higher the precision of the result. On the contrary, the lower the precision. The functions of unitary operators A , S_0 and S_X are as follows:

$$A|0\rangle = a|\psi\rangle + \sqrt{1 - a^2}|\psi_{\perp}\rangle \tag{13}$$

$$S_0 = I - 2|0\rangle\langle 0| \tag{14}$$

$$S_X = I - 2|\psi\rangle\langle\psi| \tag{15}$$

In order to apply the quantum algorithm for finding the K-nearest neighbors, we need to make the amplitude estimation step reversible. Wiebe et al. call this form of amplitude estimation coherent amplitude estimation [26]. This results in a state that is, up to local isometries, approximately

$$\frac{1}{\sqrt{M}} \sum_{j=0}^M |j\rangle | |x - v_j| \rangle \tag{16}$$

2.2.4. Search K-Nearest Neighbors

Searching K-nearest neighbors is a part of a KNN with high time complexity. The appearance of the Grover algorithm opens up a new idea for the unordered search problem [12]. Dürr proposed a quantum algorithm [38] for finding k -minimum values in 2004. Miyamoto proposed a quantum algorithm to find K -minimum values with another idea in 2019 [39]. Both their algorithms are capable of finding k -minimum values in M data with a time complexity of $O(\sqrt{kM})$. Miyamoto's algorithm is simpler and easier to implement. Here, we will present their method. He introduced a parameter t , which is used to find out k values that are less than t . The quantum algorithm for finding k-Minima is summarized as follows:

- (1) Apply algorithm [27] for finding minimum and record the last k indexes of finding minimum algorithm process;
- (2) Use binary search to find the minimum algorithm record of the threshold index t that meets the condition that the number less than t is close to k . The quantum counting algorithm is used to determine whether the condition is satisfied;
- (3) Apply Grover algorithm to search k values that are less than t .

According to [29], the time complexity of the first step is $O(\sqrt{M})$. The second step combines the quantum counting algorithm with the classical binary search. Its time com-

plexity is $O(\sqrt{M} \log k)$. Finally, the time complexity of searching K indexes is $O(\sqrt{kM})$. To sum up, the overall time complexity of the algorithm is given as:

$$(\sqrt{M}) + O(\sqrt{M} \log k) + O(\sqrt{kM}) = O(\sqrt{kM}) \tag{17}$$

2.2.5. Determine Category

Finally, we determine the category of the test sample according to the k most similar training samples. Suppose the number of each category in the k most similar training samples is k_i . The category of test samples is consistent with the index of $\max(k_i)$. However, in practical application, $\max(k_i) = k_a = k_b (a \neq b)$ may take place, which makes it impossible to determine the type of the test sample. In this paper, once the onset of above issue, we make $k = k + 1$ until we can determine the category of the test sample.

3. Results

In this section, we first demonstrate theoretically that the Polar distance can be used as a measure of sample similarity. We then compare the performance of Polar distance and Euclidean distance in KNN on the Iris, Wine, Liver, and Overflow Vulnerability datasets. Finally, we compared the performance of Polar distance and Euclidean distance in QKNN on the same dataset. The accuracy of all experiments is the average of 30 10-fold cross-validations.

3.1. A New Similarity Distance Measure

Similarity measure is a metric for comparing the similarity of two samples. When comparing two samples, distance is usually used to determine their similarity. In this paper, we proposed a new similarity distance measure called Polar distance that considers both the information of angle and module length by combining them into a weighted value. In general, distance should meet the following three properties: non-negativity, symmetry, and trigonometric inequality. Derived by cosine similarity, the angle can be used as an index to measure the similarity. Here, we prove that module length can be used as an indicator for similarity measurement from the three properties of distance above. In this work, we define the module length distance of two samples A and B as:

$$|r_A - r_B| \tag{18}$$

The first is non-negativity and symmetry. Obviously,

$$|r_A - r_B| \geq 0 \tag{19}$$

$$|r_A - r_B| = |r_B - r_A| \tag{20}$$

Finally, it is proved that it satisfies the trigonometric inequality. Considering any three samples, such as A, B, and C, it is necessary to prove

$$|r_A - r_B| + |r_A - r_C| \geq |r_B - r_A + r_A - r_C| = |r_B - r_C| \tag{21}$$

Obviously, it satisfies the trigonometric inequality. Module length distance can be used as an indicator to measure similarity. In order to combine the angle and module length as indicators to measure similarity, we define the new similarity measurement method as the following form:

$$d = 0.5 \cdot (1 + \cos^2\theta) \cdot (1 - \omega) + (1 - |r_A - r_B|) \cdot \omega \tag{22}$$

where module length r_A and r_B are scaled so that their value range is $[0, 1]$. The ω values in this paper were determined using cross-validation. Specifically, the value of k under Euclidean distance is first determined using cross-validation. The value of k is

then held constant, and we determine the parameter ω for the Polar distance using the cross-validation method.

3.2. Polar Distance and Euclidean Distance in KNN

To verify that the Polar distance can replace the Euclidean distance in KNN, we first tested the classification accuracy of two similarity distance measures in KNN under different datasets. Iris and Wine are datasets with three classes. Overflow Vulnerability and Liver are datasets with two classes. There is not much difference in the classification accuracy of the two similarity distance measures, as shown in Figures 3–6. Table 1’s KNN column shows that the difference between the two similarity distance measures is still small in the best shape of the classification accuracy. Therefore, we consider that the two similarity distance measures are approximately equivalent in KNN.

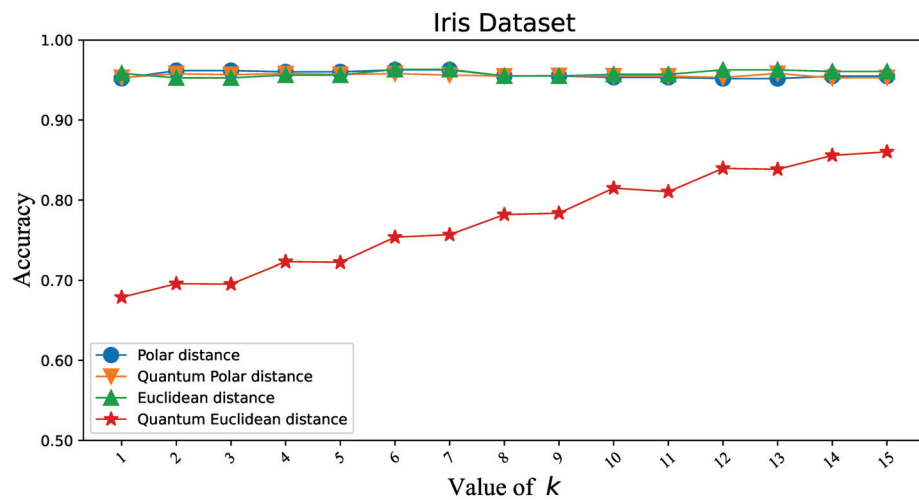


Figure 3. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on the Iris dataset.

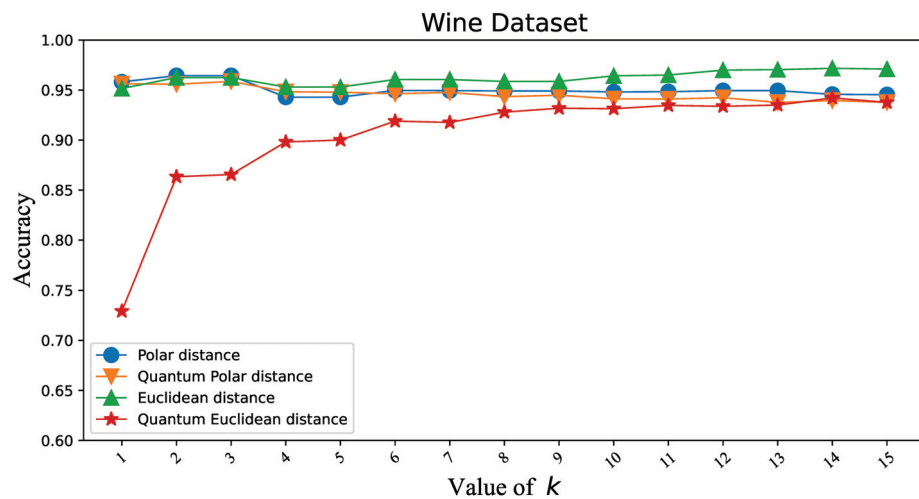


Figure 4. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on the Wine dataset.

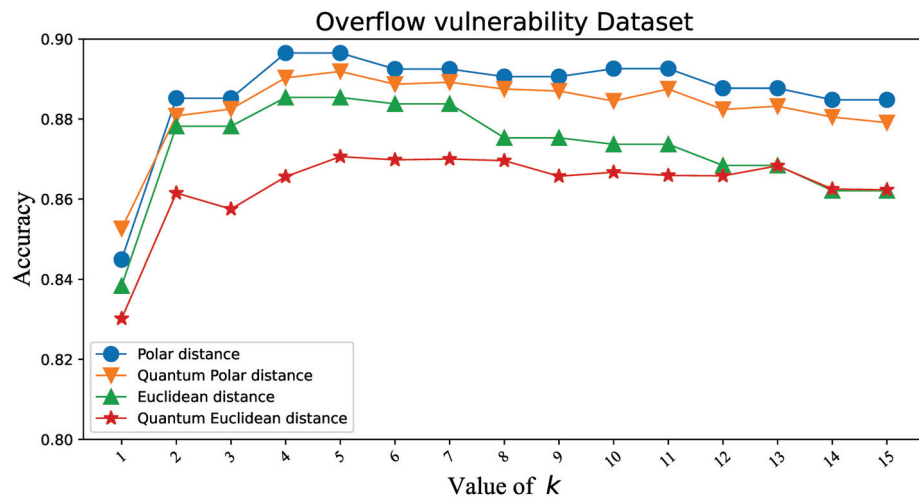


Figure 5. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on the Overflow Vulnerability dataset.

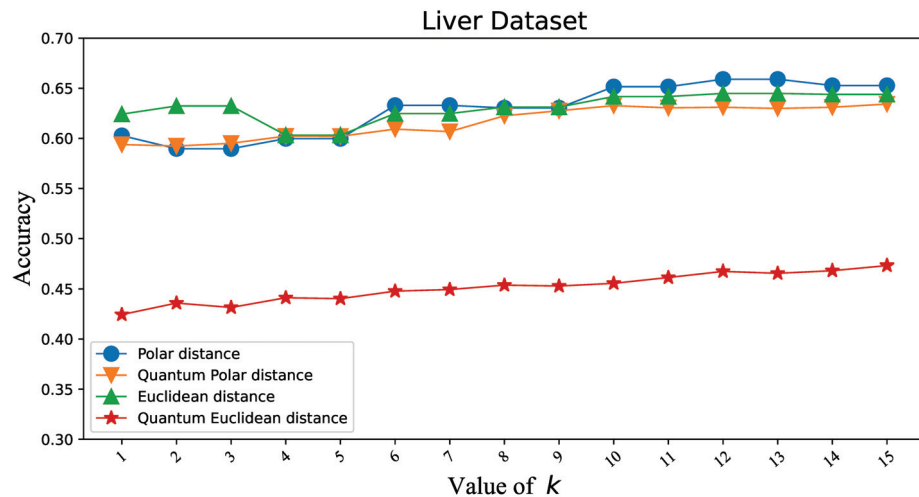


Figure 6. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on the Liver dataset.

Table 1. Classification accuracy corresponding to Polar distance and Euclidean distance in KNN and QKNN on four datasets.

Datasets	KNN		QKNN	
	Polar Distance	Euclidean Distance	Polar Distance	Euclidean Distance
Iris	96.27%	96.33%	95.82%	86.02%
Wine	96.44%	97.17%	95.86%	94.21%
Overflow	89.65%	88.54%	89.19%	87.06%
Liver	65.90%	64.48%	63.42%	47.33%

3.3. Polar Distance and Euclidean Distance in QKNN

To validate that the Polar distance can replace the Euclidean distance in QKNN, we make similar experiments as above. As shown in Figures 3–6, we can easily see that the gap between the Polar distance and quantum Polar distance is significantly smaller than the gap between the Euclidean distance and quantum Euclidean distance. From the results in the QKNN column of Table 1, the Polar distance we proposed is better than the Euclidean distance in classification accuracy. For the dataset of Iris, the accuracy of Polar distance

is 95.82%, achieving a 9.8% accuracy gain against Euclidean distance. For the dataset of Wine, the accuracy of Polar distance is 95.86%, achieving a 1.65% accuracy gain against Euclidean distance. For the dataset of Liver, the accuracy of Polar distance is 89.19%, achieving a 2.13% accuracy gain against Euclidean distance. For the dataset of Overflow Vulnerability, the accuracy of Polar distance is 63.42%, achieving a 16.09% accuracy gain against Euclidean distance. It is well known that there are deviations between quantum results and theoretical values. Although our QKNN experiments are performed by a quantum simulator, this deviation still exists due to Monte Carlo sampling. So why is the deviation from the quantum Euclidean distance greater? This starts with the calculation of the quantum Euclidean distance [32]. The formula for calculating the quantum Euclidean distance is as follows:

$$d = \sqrt{2 * (r_1^2 + r_2^2) * (2 * p(|0\rangle) - 1)} \quad (23)$$

Assume that the error of the quantum measurement result is δ . Obviously the error of the quantum Polar distance is δ . The errors in the quantum Euclidean data are as follows:

$$\frac{|\Delta d|}{d} = \frac{2\delta p}{\sqrt{2p-1} * (\sqrt{2p-1} + \sqrt{2 * (1 \pm \delta)p - 1})} \quad (24)$$

The error of the quantum Euclidean distance is $\frac{|\Delta d|}{d} = \frac{2p}{\sqrt{2p-1} * (\sqrt{2p-1} + \sqrt{2 * (1 \pm \delta)p - 1})}$ (the value of the equation is 1 to $+\infty$ since $p \in [0.5, 1]$) times that of the quantum Polar distance. The classical part of the quantum Euclidean distance estimator amplifies the error in the quantum part, so the results differ significantly from the true results. This leads to less satisfactory results for the quantum K-nearest neighbor (QKNN) algorithm based on Euclidean distance.

4. Discussion

In this paper, we proposed a new similarity distance measure to replace the Euclidean distance for use in QKNN. We call it Polar distance. From the experimental results, the Polar distance can achieve the following results in terms of classification accuracy:

- (1) The Polar and Euclidean distances are comparable in KNN;
- (2) The Polar distances are comparable in KNN and QKNN;
- (3) The Polar distances perform significantly better than the Euclidean distances in QKNN.

However, the disadvantage of the Polar distance is also obvious, namely the introduction of a new parameter ω . This not only increases the computational complexity but also makes Polar distance only applicable in supervised machine learning algorithms. We can try to find the right value of ω quickly by using gradient descent. Experiments have shown that the value of ω is not the same under different datasets. We can also delve into the relationship between the value of ω and the distribution of samples in the dataset to address the above problems. This is worthy of further study.

Author Contributions: Conceptualization, Z.S. and C.F.; methodology, C.F.; software, X.Z.; validation, C.F., B.Z. and X.D.; formal analysis, B.Z.; investigation, X.D.; resources, Z.S.; data curation, X.Z.; writing—original draft preparation, C.F.; writing—review and editing, B.Z.; visualization, X.Z.; supervision, Z.S.; project administration, Z.S.; funding acquisition, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Major Science and Technology Projects in Henan Province, China: 221100210600.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: We appreciate the support of the Nature Science Foundation of China (62006210, 62001284). In addition, we acknowledge the use of Origin Quantum services for this work. The views expressed are those of the authors and do not reflect the official policy or position of Origin Quantum or any other quantum team.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

KNN K-nearest neighbor
QKNN Quantum K-nearest neighbor

References

- Lin, T.-Y.; Dlloar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
- He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
- Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *6*, 1097–1105. [CrossRef]
- He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- Simonyan, K.; Zisserman, A. Very deep convolutional networks for large scale image recognition. In Proceedings of the International Conference on Learning Representations, San Diego, NV, USA, 7–9 May 2015.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
- Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
- Sak, H.; Senior, A.W.; Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic moduleling. In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Lukasz, K.; Polosukhin, I. Attention is all you need. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
- Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing, STOC '96, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219. [CrossRef]
- Shor, P. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
- Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [CrossRef]
- Jordan, S. The Quantum Algorithm Zoo. Available online: <http://math.nist.gov/quantum/zoo/> (accessed on 1 May 2022).
- Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* **2019**, *567*, 209–212. [CrossRef]
- Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [CrossRef]
- Chang, W.-L.; Chen, J.-C.; Chung, W.-Y.; Hsiao, C.-Y.; Wong, R.; Vasilakos, A.V. Quantum speedup and mathematical solutions of implementing bio-molecular solutions for the independent set problem on IBM quantum computers. *IEEE Trans. Nanobiosci.* **2021**, *20*, 354–376. [CrossRef]
- Wong, R.; Chang, W.-L. Fast Quantum Algorithm for Protein Structure Prediction in Hydrophobic-Hydrophilic modulel. *J. Parallel Distrib. Comput.* **2022**, *164*, 178–190. [CrossRef]
- Chang, W.-L.; Chen, J.-C.; Chung, W.-Y.; Hsiao, C.-Y.; Wong, R.; Vasilakos, A.V. Quantum Speedup for Inferring the Value of Each Bit of a Solution State in Unsorted Databases Using a Bio-Molecular Algorithm on IBM Quantum's Computers. *IEEE Trans. Nanobiosci.* **2022**, *21*, 286–293. [CrossRef]
- Wong, R.; Chang, W.-L. Quantum Speedup for Protein Structure Prediction. *IEEE Trans. Nanobiosci.* **2021**, *20*, 323–330. [CrossRef]

22. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big feature and big data classification. *Phys. Rev. Lett.* **2013**, *113*, 130503. [CrossRef]
23. Peterson, L.E. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
24. Buhrman, H.; Cleve, R.; Watrous, J.; de Wolf, R. Quantum fingerprinting. *Phys. Rev. Lett.* **2001**, *87*, 167902. [CrossRef]
25. Seth Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv* **2013**, arXiv:1307.0411.
26. Wiebe, N.; Kapoor, A.; Svore, K.M. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Inf. Comput.* **2015**, *15*, 316–356. [CrossRef]
27. Dürr, C.; Høyer, P. A Quantum Algorithm for Finding the Minimum. *arXiv* **1996**, arXiv:quant-ph/9607014.
28. Ruan, Y.; Xue, X.; Liu, H.; Tan, J.; Li, X. Quantum Algorithm for K-Nearest Neighbors Classification Based on the Metric of Hamming Distance. *Int. J. Theor. Phys.* **2017**, *56*, 3496–3507. [CrossRef]
29. Li, J.; Lin, S.; Yu, K.; Guo, G. Quantum K-nearest neighbor classification algorithm based on Hamming distance. *Quantum Inf. Process.* **2022**, *21*, 18. [CrossRef]
30. Abu Alfeilat, H.A.; Hassanat, A.; Lasassmeh, O.; Tarawneh, A.S.; Alhasanat, M.B.; Eyal Salman, H.S.; Prasath, V. Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data* **2019**, *7*, 221–248. [CrossRef]
31. Hassanat, A.B. Dimensionality Invariant Similarity Measure. *arXiv* **2014**, arXiv:1409.0923.
32. Getachew, A. Quantum K-medians Algorithm Using Parallel Euclidean Distance Estimator. *arXiv* **2020**, arXiv:2012.11139.
33. Kaye, P.; Mosca, M. Quantum Networks for Generating Arbitrary Quantum States. In Proceedings of the Optical Fiber Communication Conference and International Conference on Quantum Information, Anaheim, CA, USA, 17 March 2001.
34. Giovannetti, V.; Lloyd, S.; Maccone, L. Architectures for a quantum random access memory. *Phys. Rev. A* **2008**, *78*, 52310. [CrossRef]
35. Park, D.K.; Petruccione, F.; Rhee, J.-K.K. Circuit-Based Quantum Random Access Memory for Classical Data. *Sci. Rep.* **2019**, *9*, 3949. [CrossRef]
36. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 40504. [CrossRef]
37. Brassard, G.; Høyer, P.; Mosca, M.; Montreal, A.; Aarhus, B.U.; Waterloo, C.U. Quantum Amplitude Amplification and Estimation. *arXiv* **2000**, arXiv:quant-ph/0005055.
38. Dürr, C.; Heiligman, M.; Høyer, P.; Mhalla, M. Quantum Query Complexity of Some Graph Problems. *SIAM J. Comput.* **2004**, *35*, 1310–1328. [CrossRef]
39. Miyamoto, K.; Iwamura, M.; Kise, K. A Quantum Algorithm for Finding k -Minima. *arXiv* **2019**, arXiv:1907.03315.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Entanglement-Assisted Quantum Codes from Cyclic Codes

Francisco Revson F. Pereira ^{1,*} and Stefano Mancini ^{2,3}¹ IQM Quantum Computers, Nymphenburger Str. 86, 80636 Munich, Germany² School of Science and Technology, University of Camerino, I-62032 Camerino, Italy³ Istituto Nazionale di Fisica Nucleare, Sezione di Perugia, I-06123 Perugia, Italy

* Correspondence: revson.ee@gmail.com

Abstract: Entanglement-assisted quantum-error-correcting (EAQEC) codes are quantum codes which use entanglement as a resource. These codes can provide better error correction than the (entanglement unassisted) codes derived from the traditional stabilizer formalism. In this paper, we provide a general method to construct EAQEC codes from cyclic codes. Afterwards, the method is applied to Reed–Solomon codes, BCH codes, and general cyclic codes. We use the Euclidean and Hermitian construction of EAQEC codes. Three families have been created: two families of EAQEC codes are maximal distance separable (MDS), and one is almost MDS or almost near MDS. The comparison of the codes in this paper is mostly based on the quantum Singleton bound.

Keywords: quantum codes; Reed–Solomon codes; BCH codes; maximal distance separable; maximal entanglement

1. Introduction

Practical implementations of most quantum communication schemes and quantum computers will only be possible if such systems incorporate quantum-error-correcting codes. Quantum error correcting codes restore quantum states from corrupted by unwanted noisy action. One of the most known and used methods to create quantum codes from classical block codes is the CSS method [1]. Unfortunately, it requires (Euclidean or Hermitian) duality containing to one of the classical codes used. One way to overcome this constraint is via entanglement shared beforehand by the communicating parties. It is possible to show that such entanglement-assisted construction also improves the error-correction capability of quantum codes. These codes are called entanglement-assisted quantum error-correcting (EAQEC) codes. The first proposals of EAQEC codes were presented by Bowen [2] and Fattal et al. [3]. Then, Brun et al. [4] have developed an entanglement-assisted stabilizer formalism for these codes, which was recently generalized by Galindo et al. [5].

This formalism has created a method to construct EAQEC codes from classical block codes, which has lead to the construction of several families of EAQEC codes [6–13]. The majority of them utilize constacyclic codes [7,9,10,14] or negacyclic codes [8,9] as the classical counterpart. However, only a few of them have used cyclic codes and described the parameters of the quantum code constructed via the defining set of cyclic code. This can lead to a straightforward relation between the parameters of the classical and quantum codes and a method to create MDS EAQEC code. Li et al. used BCH codes to construct EAQEC codes via decomposing the defining set of the BCH code used [15]. Lu and Li constructed EAQEC codes from primitive quaternary BCH codes [16]. Recently, Lu et al. [9], using not cyclic but constacyclic MDS codes as the classical counterpart, proposed four families of MDS EAQEC codes.

Deriving EAQEC codes with different parameters provides a tool for reliable communication through quantum channels. In the quantum communication framework, sender and receiver are physically separated, which makes impossible the use joint unitary transformations. However, one can use other resources in order to maximize the code rate within

Citation: Pereira, F.R.F.; Mancini, S. Entanglement-Assisted Quantum Codes from Cyclic Codes. *Entropy* **2023**, *25*, 37. <https://doi.org/10.3390/e25010037>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 3 November 2022

Revised: 13 December 2022

Accepted: 21 December 2022

Published: 24 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

the constraint of high minimum distance, such as pre-shared entanglement. Although in a specified communication scenario one may aim at deriving a high rate code with a low error probability for the block code, due to the broad approach followed in this paper (where we do not design codes for a particular type of quantum channel), we shall use the minimum distance as performance measure for EAQEC codes.

The main goal of this paper is to describe any cyclic code, such as Reed–Solomon and BCH codes, under the same framework via defining set description, and to show, using two classical codes from one of these families, how to construct EAQEC codes from them. We use Euclidean and Hermitian methods to construct EAQEC codes. As it will be shown, EAQEC codes from Reed–Solomon codes are MDS codes, and the ones from BCH codes are new in two senses. The first one is that there is no work in the literature with the same parameters. The second one is that we use two BCH codes to derive the EAQEC code, which gives more freedom in the choice of parameters. Two more families of EAQEC codes are devised using the Hermitian construction. One of these families can generate codes which are almost MDS or almost near MDS; i.e., the Singleton defect for these codes is equal to one or two units. The last family created is maximally entangled and has a length proportional to a high power of the cardinality of the field. This family, when extending the block length, could approach the EA quantum hashing bound similarly to what happens to turbo codes in reference [17]. In fact, it was shown by Lai et al. [18] that maximal-entanglement EAQEC turbo codes get close to the EA quantum hashing bound. (By the EA quantum hashing bound is intended the quantum communication rate given by $1 - \frac{1}{2}[p(\log_2 3 - \log_2 p) - (1 - p) \log_2(1 - p)]$ for a depolarizing channel characterized by depolarizing parameter p [17].) Lastly, we would like to highlight that the description given in this paper gives a more direct relation between cyclic codes and the entanglement-assisted quantum codes constructed from them. Such a relation can be extended to constacyclic and negacyclic codes with a few adjustments.

The paper is organized as follows. In Section 2, we review Reed–Solomon and BCH codes and describe their parameters via a defining set. Additionally, we show construction methods of EAQEC codes from classical codes. Using these methods for cyclic classical codes, new EAQEC codes are constructed in Section 3. In Section 4, a comparison of these codes is presented via the quantum Singleton bound. In particular, we show families of MDS and almost MDS EAQEC codes. We also create a family of EAQEC codes which could approach the EA quantum hashing bound [17–19]. Lastly, the conclusion is presented in Section 5.

Notation

Throughout this paper, p denotes a prime number and $q \neq 2$ is a power of p . Let \mathbb{F}_q be the finite field with q elements. A linear code C with parameters $[n, k, d]_q$ is a k -dimensional subspace of \mathbb{F}_q^n with minimum distance d . For cyclic codes, $Z(C)$ denotes the defining set, and $g(x)$ is the generator polynomial. Lastly, an $[[n, k, d; c]]_q$ quantum code is a q^k -dimensional subspace of \mathbb{C}^{q^n} with minimum distance d that utilizes c pre-shared entangled pairs.

2. Preliminaries

In this section, we review some ideas related to linear complementary dual (LCD) codes, cyclic codes, and entanglement-assisted quantum codes. Before giving a description of LCD codes, we need to define the Euclidean and Hermitian dual of a linear code.

Definition 1. Let C be a linear code over \mathbb{F}_q with length n . The (Euclidean) dual of C is defined as

$$C^\perp = \{ \mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{x} \cdot \mathbf{c} = 0 \text{ for all } \mathbf{c} \in C \}. \tag{1}$$

If the finite field has cardinality equal to q^2 , an even power of a prime, then we can define the Hermitian dual of C . This dual code is defined by

$$C^{\perp_h} = \{ \mathbf{x} \in \mathbb{F}_{q^2}^n \mid \mathbf{x} \cdot \mathbf{c}^q = 0 \text{ for all } \mathbf{c} \in C \}, \tag{2}$$

where $\mathbf{c}^q = (c_1^q, \dots, c_n^q)$ for $\mathbf{c} \in \mathbb{F}_{q^2}^n$.

These types of dual codes can be used to derive quantum codes from the stabilizer formalism [1]. The requirement in this formalism is to the classical code to be self-dual; i.e., $C \subseteq C^\perp$ or $C \subseteq C^{\perp_h}$. However, there is a different relationship between a code and its (Euclidean or Hermitian) dual that can be interesting for constructing an EAQEC. This relation is complementary duality and is defined in the following.

Definition 2. The hull of a linear code C is given by $\text{hull}(C) = C^\perp \cap C$. The code is called linear complementary dual (LCD) code if the hull is trivial; i.e, $\text{hull}(C) = \{0\}$. Similarly, it is defined by $\text{hull}_H(C) = C^{\perp_h} \cap C$ and the idea of a Hermitian LCD code.

Now, we can define cyclic codes and some properties that can be used to extract the parameters of the quantum code constructed from them.

2.1. Cyclic Codes

A linear code C with parameters $[n, k, d]_q$ is called cyclic if for any codeword $(c_0, c_1, \dots, c_{n-1}) \in C$ implies $(c_{n-1}, c_0, c_1, \dots, c_{n-2}) \in C$. By defining a map from \mathbb{F}_q^n to $\mathbb{F}_q[x]/(x^n - 1)$, which takes $\mathbf{c} = (c_0, c_1, \dots, c_{n-1}) \in \mathbb{F}_q^n$ to $c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1} \in \mathbb{F}_q[x]/(x^n - 1)$, we can see that a linear code C is cyclic if and only if it corresponds to an ideal of the ring $\mathbb{F}_q[x]/(x^n - 1)$. Since any ideal in $\mathbb{F}_q[x]/(x^n - 1)$ is principal, any cyclic code C is generated by a polynomial $g(x)|(x^n - 1)$, which is called a generator polynomial. This polynomial is monic, and has the smallest degree among all the generators of C .

A characterization of the parameters of a cyclic code can be given from the generator polynomial and its defining set. For the description of this set, consider the following: Let $m = \text{ord}_n(q)$, α be a generator of the multiplicative group $\mathbb{F}_{q^m}^*$, and assume $\beta = \alpha^{\frac{q^m-1}{n}}$; i.e., β is a primitive n -th root of unity. Then, the defining set of C , which is denoted by $Z(C)$, is defined as $Z(C) = \{i \in \mathbb{Z}_n : c(\beta^i) = 0 \text{ for all } c(x) \in C\}$.

BCH and Reed–Solomon codes are particular cases of cyclic codes, where the generator polynomial has some additional properties. See Definitions 3 and 5.

Definition 3. Let $b \geq 0, \delta \geq 1$, and $\alpha \in \mathbb{F}_{q^m}$, where $m = \text{ord}_n(q)$. A cyclic code C of length n over \mathbb{F}_q is a BCH code with designed distance δ if

$$g(x) = \text{lcm}\{m_b(x), m_{b+1}(x), \dots, m_{b+\delta-2}(x)\}$$

where $m_i(x)$ is the minimal polynomial of α^i over \mathbb{F}_q . If $n = q^m - 1$, then the BCH code is called primitive, and if $b = 1$, it is called narrow-sense.

Before relating the parameters of an BCH code with the defining set, we need to introduce the idea of the cyclotomic coset. It comes from the observation that the minimal polynomial $m_i(x)$ of α^i can be the minimal polynomial of other powers of α . The reason for this is that α belongs to an extension of \mathbb{F}_q while the polynomial $m_i(x) \in \mathbb{F}_q[x]$. The set of all zeros of $m_i(x)$ in the field \mathbb{F}_{q^m} is given by the cyclotomic coset of i . Thus, the defining set of a BCH code C is the union of the cyclotomic cosets of $b, b + 1, \dots, b + \delta - 2$. The following definition describes this set.

Definition 4. The q -ary cyclotomic coset $\text{mod } n$ containing an element i is defined by

$$\mathbb{C}_i = \{i, iq, iq^2, iq^3, \dots, iq^{m_i-1}\}, \tag{3}$$

where m_i is the smallest positive integer such that $iq^{m_i} \equiv i \pmod n$.

For the parameters of a BCH code, it is shown that the dimension is equal to $n - |Z(C)|$ and the minimal distance of C is at least δ [20]. Thus, we can see that important properties of an BCH codes can be obtained from the defining set. The same characterization happens with Euclidean or Hermitian dual cyclic code. Propositions 1 and 2 focus on this.

Proposition 1 ([20], Proposition 4.3.8). Let C be a linear code of length n and defining set $Z(C)$. Then, the defining set of C^\perp is given by

$$Z(C^\perp) = \mathbb{Z}_n \setminus \{-i | i \in Z(C)\}$$

For BCH codes, the generator polynomial is given by the lcm of the minimal polynomials over \mathbb{F}_q of the elements α^j such that $j \in Z(C^\perp)$.

Proposition 2. Let C be a cyclic code over \mathbb{F}_{q^2} with defining set $Z(C)$. Then,

$$Z(C^{\perp_h}) = \mathbb{Z}_n \setminus \{-i | i \in qZ(C)\}.$$

Proof. Let $\mathbf{c} \in \mathbb{F}_{q^2}^n$ be a codeword of C . By expressing \mathbf{c}^q as a polynomial, we have that $c^{(q)}(x) = c_0^q + c_1^q x + \dots + c_{n-1}^q x^{n-1}$. Thus, $i \in \mathbb{Z}_n$ belongs to $Z(C^q)$ if and only if

$$\begin{aligned} c^{(q)}(\alpha^i) = 0 &\iff c_0^q + c_1^q \alpha^i + \dots + c_{n-1}^q \alpha^{i(n-1)} = 0 \\ &\iff (c_0^q + c_1^q \alpha^i + \dots + c_{n-1}^q \alpha^{i(n-1)})^q = 0 \\ &\iff c_0 + c_1 \alpha^{iq} + \dots + c_{n-1} \alpha^{iq(n-1)} = 0 \\ &\iff iq \in Z(C). \end{aligned}$$

This shows that $Z(C^q) = qZ(C)$. Since $C^{\perp_h} = (C^q)^\perp$, we have from Proposition 1 that $Z(C^{\perp_h}) = \mathbb{Z}_n \setminus \{-i | i \in qZ(C)\}$. \square

The other class of cyclic codes used in this paper, Reed–Solomon codes, can be viewed as a subclass of BCH codes. Thus, a similar characterization in terms of defining set can be given; see Definition 5 and Corollary 1. One property of such codes that makes them important is that they are maximal distance separable (MDS) codes; i.e., fixing the length and the dimension, they have the maximal minimal distance possible. As shown in Section 3, using such codes to construct EAQEC codes will result in MDS quantum codes.

Definition 5. Let $b \geq 0$, $n = q - 1$, and $1 \leq k \leq n$. A cyclic code $RS_k(n, b)$ of length n over \mathbb{F}_q is a Reed–Solomon code with minimal distance $n - k + 1$ if

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \dots (x - \alpha^{b+n-k-1}),$$

where α is a primitive element of \mathbb{F}_q .

A particular application of Proposition 1 to Reed–Solomon codes is described in Corollary 1, where the parameters and defining set of an Euclidean dual of a Reed–Solomon are derived.

Corollary 1. Let $RS_k(n, b)$ be a Reed–Solomon code. Then, its Euclidean dual can be described as

$$RS_k(n, b)^\perp = RS_{n-k}(n, n - b + 1)$$

In particular, the defining set of $RS_k(n, b)^\perp$ is given by $Z(RS_k(n, b)^\perp) = \{n - b + 1, n - b + 2, \dots, n - b + k\}$.

As will be shown in the next subsection, the amount of entanglement in a EAQEC code is computed from the dimension of the intersection between the two codes. Thus, the last proposition of this subsection addresses the subject.

Proposition 3 ([21], Exercise 239, Chapter 4). *Let C_1 and C_2 be cyclic codes with defining sets $Z(C_1)$ and $Z(C_2)$, respectively. Then, the defining set of $C_1 \cap C_2$ is given by $Z(C_1) \cup Z(C_2)$. In particular, $\dim(C_1 \cap C_2) = n - |Z(C_1) \cup Z(C_2)|$.*

2.2. Entanglement-Assisted Quantum Codes

Definition 6. *A quantum code \mathcal{Q} is called an $[[n, k, d; c]]_q$ entanglement-assisted quantum error-correcting (EAQEC) code if it encodes k logical qudits into n physical qudits using c copies of maximally entangled states and can correct $\lfloor (d - 1)/2 \rfloor$ quantum errors. A EAQEC code is said to have maximal entanglement when $c = n - k$.*

Formulating a stabilizer paradigm for EAQEC codes gives a way to use classical codes to construct this quantum codes [22]. In particular, we have the next two procedures by Galindo et al. [5].

Proposition 4 ([5], Theorem 4). *Let C_1 and C_2 be two linear codes over \mathbb{F}_q with parameters $[n, k_1, d_1]_q$ and $[n, k_2, d_2]_q$; and parity check matrices H_1 and H_2 , respectively. Then, there is a EAQEC code with parameters $[[n, k_1 + k_2 - n + c, d; c]]_q$, where $d = \min\{d_H(C_1 \setminus (C_1 \cap C_2^\perp)), d_H(C_2 \setminus (C_1^\perp \cap C_2))\}$, with d_H as the minimum Hamming weight of the vectors in the set, and*

$$c = \text{rank}(H_1 H_2^T) = \dim C_1^\perp - \dim(C_1^\perp \cap C_2) \tag{4}$$

is the number of required maximally entangled states.

Proposition 5 ([5], Proposition 3 and Corollary 1). *Let C be a linear codes over \mathbb{F}_{q^2} with parameters $[n, k, d]_q$, H be a parity check matrix for C , and H^* be the q -th power of the transpose matrix of H . Then, there is a EAQEC code with parameters $[[n, 2k - n + c, d'; c]]_q$, where $d' = d_H(C \setminus (C \cap C^{\perp h}))$, with d_H as the minimum Hamming weight of the vectors in the set, and*

$$c = \text{rank}(H H^*) = \dim C^{\perp h} - \dim(C^{\perp h} \cap C) \tag{5}$$

is the number of required maximally entangled states.

A measurement of goodness for a EAQEC code is the quantum Singleton bound (QSB). Let $[[n, k, d; c]]_q$ be a EAQEC code. Then, the QSB is given by

$$d \leq \left\lfloor \frac{n - k + c}{2} \right\rfloor + 1. \tag{6}$$

The difference between the QSB and d is called a quantum Singleton defect. When the quantum Singleton defect is equal to zero (resp. one), the code is called the maximum distance separable quantum code (resp. almost maximum distance separable quantum code), and it is denoted the MDS quantum code (resp. almost MDS quantum code).

3. New Entanglement-Assisted Quantum-Error-Correcting Cyclic Codes

In this section is shown the construction of EAQEC codes from the cyclic codes. We are going to make use of Euclidean and Hermitian constructions, which will give codes with different parameters when compared over the same field.

3.1. Euclidean Construction

A straightforward application of cyclic codes to the Proposition 4 via defining set description can produce some interesting results. See Theorem 1 and Corollary 2.

Theorem 1. Let C_1 and C_2 be two cyclic codes with parameters $[n, k_1, d_1]_q$ and $[n, k_2, d_2]_q$, respectively. Then, there is an EAQEC code with parameters $[[n, k_1 - |Z(C_1^\perp) \cap Z(C_2)|, \min\{d_1, d_2\}; n - k_2 - |Z(C_1^\perp) \cap Z(C_2)|]]_q$.

Proof. From Proposition 3, we have that $\dim(C_1^\perp \cap C_2) = n - |Z(C_1^\perp) \cup Z(C_2)| = n - |Z(C_2)| - |Z(C_1^\perp)| + |Z(C_1^\perp) \cap Z(C_2)| = k_2 - k_1 + |Z(C_1^\perp) \cap Z(C_2)|$. Thus, the amount of entanglement used in an EAQEC code constructed from these two cyclic codes can be computed from Proposition 4, which is $c = n - k_2 - |Z(C_1^\perp) \cap Z(C_2)|$. By substituting this value of c in the parameters of the EAQEC code in Proposition 4, we obtain an $[[n, k_1 - |Z(C_1^\perp) \cap Z(C_2)|, \min\{d_1, d_2\}; n - k_2 - |Z(C_1^\perp) \cap Z(C_2)|]]_q$ EAQEC code. \square

Corollary 2. Let C be a LCD cyclic code with parameters $[n, k, d]_q$. Then, there is a maximal entanglement EAQEC code with parameters $[[n, k, d; n - k]]_q$. In particular, if C is MDS, so is the EAQEC code derived from it.

Proof. Let $C_1 = C_2 = C$ in Theorem 1. Since C is LCD, $|Z(C_1^\perp) \cap Z(C_2)| = 0$. From Theorem 1, we have that there is an EAQEC code with parameters $[[n, k, d; n - k]]_q$. \square

Theorem 2. Let $C_1 = RS_{k_1}(n, b_1)$ and $C_2 = RS_{k_2}(n, b_2)$ be two Reed–Solomon codes over \mathbb{F}_q with $0 \leq b_1 \leq k_1, b_2 \geq 0$, and $b_1 + b_2 \leq k_2 + 1$. Then, we have two possible cases:

1. For $k_1 - b_1 \geq b_2$, there is an EAQEC code with parameters

$$[[n, b_1 + b_2 - 1, n - \min\{k_1, k_2\} + 1; n + b_1 + b_2 - k_1 - k_2 - 1]]_q;$$

2. For $k_1 - b_1 < b_2$, there is an EAQEC code with parameters

$$[[n, k_1, n - \min\{k_1, k_2\} + 1; n - k_2]]_q.$$

Proof. From Corollary 1, we have that $Z(C_1^\perp) = \{n - b_1 + 1, n - b_1 + 2, \dots, n - b_1 + k_1\}$. First of all, notice that the restriction $b_1 + b_2 \leq k_2 + 1$ implies that the first element in the defining set of $Z(C_1^\perp)$ comes after the last element in $Z(C_2)$. Since $0 \leq b_1 \leq k_1$, we have that $n - b_1 + k_1 \geq n$, which implies that the defining set for C_1^\perp equals to $Z(C_1^\perp) = \{n - b_1 + 1, n - b_1 + 2, \dots, n - 1, 0, 1, \dots, k_1 - b_1\}$. Thus, $Z(C_1^\perp)$ intersects $Z(C_2)$ if and only if $k_1 - b_1 \geq b_2$. In the case that it does, the intersection is equals to $Z(C_1^\perp) \cap Z(C_2) = k_1 - (b_1 + b_2) + 1$. The missing claims are obtained using these results in Theorem 1. \square

Corollary 3. Let $C = RS_k(n, b)$ be a Reed–Solomon code over \mathbb{F}_q with $0 < b \leq (k + 1)/2$ and $0 < k < n \leq q$. Then, there is an MDS EAQEC code with parameters $[[n, 2b - 1, n - k + 1; n + 2b - 2k - 1]]_q$. In particular, for $b = (k + 1)/2$, there is a maximal entanglement MDS EAQEC code.

Proof. Let $C_1 = C_2 = RS_k(n, b)$ in Theorem 2. Assuming $0 \leq b < (k + 1)/2$, we have that the classical codes fall in the first case of Theorem 2; and for $b = (k + 1)/2$, we are in the second case of Theorem 2. Thus, substituting the values of k_1, k_2 and b_1, b_2 by k and b , respectively; the result follows. \square

In a similar way, we can use BCH codes to construct EAQEC codes. The advantage in using BCH codes is that the length of the code is not bounded by the cardinality of the finite field used. However, creating classical or quantum codes from BCH codes which are MDS is a difficult task. Our proposal to have BCH codes as the classical counterpart in

this paper is to show how to use two BCH codes to construct EAQEC codes. In addition, it is also constructed maximal entanglement EAQEC codes. In order to do this, we show suitable properties concerning some cyclotomic cosets for $n = q^2 - 1$.

Lemma 1. Let $n = q^2 - 1$ with $q > 2$. Then, the q -ary coset \mathbb{C}_0 has one element, and $\mathbb{C}_i = \{i, iq\}$ for any $1 \leq i \leq q - 1$.

Proof. The first claim is trivial. For the second one, notice $iq^2 \equiv i \pmod{q^2 - 1}$. Thus, the only elements in \mathbb{C}_i are i and iq , for $1 \leq i \leq q - 1$. \square

From Lemma 1, we can construct EAQEC codes with length $n = q^2 - 1$. See Theorem 3.

Theorem 3. Let $n = q^2 - 1$ with $q > 2$. Assume a, b are integers such that $0 \leq a \leq q - 1$ and $1 \leq b \leq q$. Then, there is an EAQEC code with parameters

- $[[n, 2(q - b) - 1, b + 1; 2(q - a - 1)]]_q$, if $a \geq q - b$ and $b < q$;
- $[[n, 2a + 1, b + 1; 2b - \lfloor \frac{b}{q} \rfloor]]_q$, if $a < q - b$.

Proof. First of all, assume that C_1^\perp has a defining set given by $Z(C_1^\perp) = \cup_{i=0}^a \mathbb{C}_i$, and the defining set of C_2 is equal to $Z(C_2) = \cup_{i=1}^b \mathbb{C}_{q-i}$. From Lemma 1, we have that $|Z(C_1^\perp)| = 2a + 1$ and $|Z(C_2)| = 2b - \lfloor \frac{b}{q} \rfloor$. Thus, the dimensions of C_1 and C_2 are equal to $k_1 = |Z(C_1^\perp)| = 2a + 1$ and $k_2 = n - |Z(C_2)| = n - 2b + \lfloor \frac{b}{q} \rfloor$, respectively. To compute $|Z(C_1^\perp) \cap Z(C_2)|$, we have to consider two cases. If $a \geq q - b$, then we have that $Z(C_1^\perp) \cap Z(C_2) = \cup_{i=q-b}^a \mathbb{C}_i$, which has cardinality given by $|Z(C_1^\perp) \cap Z(C_2)| = 2(a - (q - b) + 1) - \lfloor \frac{b}{q} \rfloor$, because $|\mathbb{C}_0| = 1$. On the other hand, if $a < q - b$, then $|Z(C_1^\perp) \cap Z(C_2)| = 0$. Lastly, since $a, b \leq q$, $Z(C_1^\perp) = \cup_{i=0}^a \mathbb{C}_i$, and $n = q^2 - 1$ with $q > 2$, we can see that $d_1 > d_2 = b + 1$. Now, using these results in Theorem 1, we have that there is a EAQEC code with parameters $[[n, 2(q - b) - 1 + \lfloor \frac{b}{q} \rfloor, b + 1; 2(q - a - 1)]]_q$, if $a \geq q - b$, or a EAQEC code with parameters $[[n, 2a + 1, b + 1; 2b - \lfloor \frac{b}{q} \rfloor]]_q$. \square

3.2. Hermitian Construction

In the same way as before, it possible to use cyclic codes to construct EAQEC codes from the Hermitian construction method of Proposition 5. See the following theorem.

Theorem 4. Let C be a cyclic code with parameters $[n, k, d]_{q^2}$. Then there is an EAQEC code with parameters $[[n, k - |Z(C^{\perp h}) \cap Z(C)|, d; n - k - |Z(C^{\perp h}) \cap Z(C)|]]_q$.

Proof. First of all, from Proposition 3 we have $\dim(C^\perp \cap C) = n - |Z(C^\perp) \cup Z(C)| = n - |Z(C)| - |Z(C^{\perp h})| + |Z(C^{\perp h}) \cap Z(C)| = k - k + |Z(C^{\perp h}) \cap Z(C)| = |Z(C^{\perp h}) \cap Z(C)|$. Thus, $c = \dim(C^{\perp h}) - \dim(C^\perp \cap C) = n - k - |Z(C^{\perp h}) \cap Z(C)|$. Using a $[n, k, d]_{q^2}$ to construct a EAQEC codes via Proposition 5, we derive a code with parameters $[[n, k - |Z(C^{\perp h}) \cap Z(C)|, d; n - k - |Z(C^{\perp h}) \cap Z(C)|]]_q$. \square

Corollary 4. Let C be an LCD cyclic code with parameters $[n, k, d]_{q^2}$. Then there is a maximal entanglement EAQEC code with parameters $[[n, k, d; n - k]]_q$.

Proof. From the proof of Theorem 4, we have that $\dim(C^{\perp h} \cap C) = |Z(C^{\perp h}) \cap Z(C)|$. Since C is LCD, $|Z(C^{\perp h}) \cap Z(C)| = 0$, and the result follows from Theorem 4. \square

Differently from the construction of EAQEC code via Euclidean dual cyclic code, the construction via Hermitian dual can be more delicate, even for Reed–Solomon codes. Even so, we are going to show that is possible to construct EAQEC codes from a particular case of Reed–Solomon codes and some cyclic codes.

Theorem 5. Let q be a prime power and assume $C = RS_k(n, 1)$ is a Reed–Solomon code over \mathbb{F}_{q^2} with $k = qt + r < q^2$, where $t \geq 1$ and $0 \leq r \leq q - 1$, and $n = q^2$. Then we have the following:

- If $t \geq q - r - 1$, then there exists an MDS EAQEC code with parameters

$$[[q^2, (t + 1)^2 - 2(q - r) + 1, q(q - t) - r + 1; (q - t - 1)^2 + 1]]_q.$$

- If $t < q - r - 1$, then there exists an MDS EAQEC code with parameters

$$[[q^2, t^2 - 1, q(q - t) - r + 1; (q - t)^2 - 2r - 1]]_q.$$

Proof. Since $C = RS_k(n, 0)$, we have that $Z(C) = \{0, 1, 2, \dots, n - k - 1\}$. From the proof of Theorem 2, we also have that $Z(C^{\perp h}) = qZ(C^{\perp}) = \{q, 2q, \dots, kq\}$. From $n = q^2$ and $k = qt + r$, we can rewrite these two defining sets as $Z(C) = \{qi + j | 0 \leq i \leq q - t - 2, 0 \leq j \leq q - 1\} \cup \{(q - t - 1)q + j | 0 \leq j \leq q - r - 2\}$ and $Z(C^{\perp h}) = \{qi + j | 0 \leq i \leq q - 1, 0 \leq j \leq t - 1\} \cup \{qi + t | 0 \leq i \leq r\}$. Using this description, we can compute $|Z(C) \cap Z(C^{\perp h})|$. To do so, we have to consider two cases separately, $t \geq q - r - 1$ and $t < q - r - 1$. For the first case, the intersection is given by the following set $Z(C) \cap Z(C^{\perp h}) = \{qi + j | 0 \leq i \leq q - t - 2, 0 \leq j \leq t\} \cup \{(q - t - 1)q + j | 0 \leq j \leq q - r - 2\}$. Thus, $|Z(C) \cap Z(C^{\perp h})| = (q - t - 1)(t + 1) + q - r - 1$. Similarly for the case $t < q - r - 1$, we have $Z(C) \cap Z(C^{\perp h}) = \{qi + j | 0 \leq i \leq q - t - 1, 0 \leq j \leq t - 1\} \cup \{qi + t | 0 \leq i \leq r\}$, which implies $|Z(C) \cap Z(C^{\perp h})| = (q - t)t + r + 1$. Using these results and the fact that C has parameters $[q^2, k, q^2 - k + 1]_{q^2}$, in Theorem 4, we have that there exists a EAQEC code with parameters

- $[[q^2, (t + 1)^2 - 2(q - r) + 1, q(q - t) - r + 1; (q - t - 1)^2 + 1]]_q$, for $t \geq q - r - 1$; and
- $[[q^2, t^2 - 1, q(q - t) - r + 1; (q - t)^2 - 2r - 1]]_q$, for $t < q - r - 1$.

□

Theorem 6. Let $n = q^4 - 1$ and $q \geq 3$, a prime power. There exists an EAQEC code with parameters $[[n, n - 4(a - 1) - 3, d \geq a + 1; 1]]_q$, where $2 \leq a \leq q^2 - 1$.

Proof. Let C_a be a cyclic code with defining set $Z(C_a) = \mathbb{C}_0 \cup \mathbb{C}_{q^2+1} \cup (\cup_{i=2}^a \mathbb{C}_{q^2+a})$, for $2 \leq a \leq q^2 - 1$. From Ref. [23], we have that $\mathbb{C}_{q^2+1} = \{q^2 + 1\}$ and $\mathbb{C}_{q^2+a} = \{q^2 + a, 1 + aq^2\}$. It is trivial to show that $\mathbb{C}_0 = \{0\}$. From $-qZ(C_a) \cap Z(C_a) = \mathbb{C}_0$ [23], we can see that $Z(C_a^{\perp h}) \cap Z(C_a) = Z(C_a) \setminus \mathbb{C}_0$. Hence, $|Z(C_a^{\perp h}) \cap Z(C_a)| = 2(a - 1) + 1$. From the assumption of the defining set, the dimension and minimal distance of the classical code are $k = n - 2(a - 1) - 2$ and $d \geq a + 1$, respectively. Thus, using these quantities in Theorem 4, we have that there exists an EAQEC code with parameters $[[n, n - 4(a - 1) - 3, d \geq a + 1; 1]]_q$. □

Two important comments can be made about Theorem 6. Comparing the bound given for the minimal distance and the Singleton bound for EAQEC codes, we see that the difference between these two values is equal to $a - 1$. Thus, for lower values of a (such as $a = 2$ or $a = 3$), the EAQEC codes have a minimal distance, close to the optimum; e.g., if $a = 2$ (or $a = 3$), the family of EAQEC codes is almost MDS (or almost MDS). The second point is that the codes in Theorem 6 can be seen as a generalization of the result by Qian and Zhang [24].

In the following, we use LCD cyclic code to construct maximal entanglement EAQEC codes. The families obtained have an interesting range of possible parameters.

Theorem 7. Let q be a prime power, $m \geq 2$, $2 \leq \delta \leq q^{2\lceil \frac{m}{2} \rceil} + 1$, and $\kappa = q^{2m} - 2 - 2(\delta - 1 - \lfloor \frac{\delta - 1}{q^2} \rfloor)m$. Then,

- For m odd and $1 \leq u \leq q - 1$, there is a maximal entanglement EAQEC code with parameters $[[q^{2m} - 1, k, d \geq \delta + 1 + \lfloor \frac{\delta-1}{q} \rfloor; q^{2m} - 1 - k]]_q$, where

$$k = \begin{cases} \kappa, & \text{if } 2 \leq \delta \leq q^m - 1; \\ \kappa + u^2 m, & \text{if } uq^m \leq \delta \leq (u + 1)(q^m - 1); \\ \kappa + (u^2 + 2v + 1)m, & \text{if } \delta = (u + 1)(q^m - 1) + v + 1 \text{ for } 0 \leq v \leq u - 1; \\ \kappa + q^2 m, & \text{if } \delta = q^{m+1} \text{ or } q^{m+1} + 1. \end{cases} \quad (7)$$

- For m even, there is an maximal entanglement EAQEC code with parameters

$$[[q^{2m} - 1, \kappa, d \geq \delta + 1 + \lfloor \frac{\delta - 1}{q} \rfloor; 2(\delta - 1 - \lfloor \frac{\delta - 1}{q^2} \rfloor)m + 1]]_q. \quad (8)$$

Proof. From Li [25], we have that there are LCD cyclic codes with parameters $[q^{2m} - 1, k, \delta + 1 + \lfloor \frac{\delta-1}{q} \rfloor]_{q^2}$, where k is the same as in Equations (7) and (8) for odd and even m , respectively. Thus, by applying this LCD code to Corollary 4, we obtain the mentioned codes. \square

4. Code Examples

In Table 1, we present some MDS EAQEC codes obtained from Corollary 3 and Theorem 5. The codes in the first column are obtained from the Euclidean construction and the ones in the second column from the Hermitian construction. As can be seen, the latter one has a higher length within the same field. Thus, it can be used in applications where the underline quantum system has limited dimensions. On the other hand, the codes in the first column can have parameters that the ones from the Hermitian construction cannot. Thus, these two classes of EAQEC codes are suitable for specific applications.

The codes obtained from Corollary 3 and Theorem 5 are maximal entanglement EAQEC codes. We could use the dependency between the cardinality of the finite field and code parameters to derive new codes. In particular, this is not the case for the codes in Ref. [26], where the cardinality of the finite field must be two. Additionally, one cannot find in Ref. [9] codes similar to the ones on the left column of Table 1, since the codes in Ref. [9] request a number c of entangled pairs that can be only equal to one or two. For our codes with $c = 1$ or 4 , which can be used in a comparison with the codes in Ref. [9], we see that the codes $[[4, 3, 2; 1]]_4$ and $[[13, 9, 5; 4]]_{13}$ have parameters slightly worse than the codes $[[5, 4, 2; 1]]_7$ and $[[10, 9, 5; 4]]_3$, respectively. Lastly, if we do not take into consideration the cardinality of the field, we continue to see improvements in the code parameters. As an example, the code $[[16, 3, 9; 3]]_4$ has a higher rate (ratio between code dimension and code length) than the similar minimum distance code $[[31, 10, 10; 21]]_4$ given in Ref. [27].

Table 1. Some new MDS EAQEC codes from Reed–Solomon codes. The codes with a star \star are maximal entanglement MDS EAQEC codes.

New EAQEC codes—Corollary 3 $[[n, 2b - 1, n - k + 1; n + 2b - 2k - 1]]_q$ $0 < b \leq (k + 1)/2$ and $0 < k < n \leq q$	New EAQEC codes—Theorem 5 $[[q^2, t^2 - 1, q(q - t) - r + 1; (q - t)^2 - 2r - 1]]_q$ $qt + r < q^2$, where $1 \leq t < q - r - 1$ and $0 \leq r \leq q - 1$
Examples	
$\star[[3, 1, 3; 2]]_3$	$[[16, 3, 9; 3]]_4$
$\star[[4, 3, 2; 1]]_4$	$[[64, 35, 17; 3]]_8$
$\star[[7, 3, 5; 4]]_7$	$[[64, 15, 31; 11]]_8$
$\star[[8, 5, 4; 3]]_8$	$[[256, 196, 33; 3]]_{16}$
$\star[[11, 9, 3; 2]]_{11}$	$[[256, 120, 78; 18]]_{16}$
$\star[[13, 9, 5; 4]]_{13}$	$[[1024, 784, 129; 15]]_{32}$
$[[16, 13, 3; 2]]_{16}$	$[[1024, 624, 220; 38]]_{32}$

One family of EAQEC codes derived from BCH codes has been constructed; see Theorem 3. Some examples of these EAQEC codes are shown in Table 2. As can be seen in Table 1 in Ref. [28] (and the reference there in), the EAQEC codes derived from Theorem 3 have new parameters when compared with EAQEC codes known in the literature. Thus, though not having good parameters as the ones in our Table 1 in terms of quantum Singleton defect, these codes are new. One advantage of our codes with respect to the ones known in the literature is that, since they are constructed from two BCH codes, we have more freedom in the choice of parameters. The family of codes presented in Table 2 could be used in environments with low amounts of resources, since we have more freedom in the code parameters. As an example, the codes in Table 2 are longer than the codes in Table 1 for the same cardinality of the finite field, making the codes in Table 2 more favorable to environments where increasing the size of individual systems is less costly than composing such systems. Looking at the examples of Table 2, we see that there is no counterpart for the codes with parameters $[[63, 7, 5; 8]]_8$ and $[[255, 19, 7; 12]]_{16}$ in Ref. [26]. However, we did not obtain an improvement in rate when comparing the remaining codes in Table 2 with the codes shown in Ref. [29].

Table 2. Some new EAQEC codes from BCH codes.

New EAQEC codes—Theorem 3 $[[q^2 - 1, 2a + 1, b + 1; 2b - \lfloor \frac{b}{q} \rfloor]]_q$ $1 \leq b \leq q$ and $0 \leq a < q - b$
Examples
$[[15, 5, 2; 2]]_4$
$[[48, 9, 3; 4]]_7$
$[[63, 7, 5; 8]]_8$
$[[255, 19, 7; 12]]_{16}$

The remaining EAQEC codes constructed in this paper are the ones derived from cyclic codes that are neither Reed–Solomon nor BCH codes. Two families of such codes were created, both of them using Hermitian construction. Some examples of parameters that can be obtained from these codes are presented in Table 3. Codes in the first column are almost MDS or almost MDS—i.e., the Singleton defect, which is when the difference between the quantum Singleton bound (QSB) presented in Equation (6) and the minimal distance of the code is equal to one or two units. Lastly, we display in the second column of Table 3 some codes from Theorem 7. All codes in Theorem 7 are maximal entanglement. Thus, this family, when extending the block length, could approach the EA quantum hashing bound similarly to what happens to turbo codes in Ref. [17]. Having length proportional to a high power of the cardinality of the field, it is expected to achieve low error probability using these codes.

To compare the codes shown in Tables 2 and 3, we are going to use the concepts of ratio, given by k/n , and net ratio, given by $(k - c)/n$, where k, c , and n are the code dimension, the number of maximally entangled states, and code length, respectively. For the code $[[80, 50, 10; 30]]_3$, we see significant improvements in rate and net rate when comparing with the codes $[[73, 36, 10; 37]]_4$ and $[[89, 44, 10; 45]]_4$ shown in Ref. [27]. A similar conclusion is obtained for the comparison between our $[[255, 237, 7; 18]]_4$ and the code $[[217, 186, 6; 31]]_4$ shown in Ref. [27]. Lastly, we also have codes with no counterpart in Ref. [27], such as $[[80, 73, 3; 1]]_3$ and $[[255, 248, 3; 1]]_4$, due to large discrepancy in code parameters.

Table 3. Some EAQEC codes from cyclic codes via Hermitian construction.

New EAQEC codes—Theorem 6	New EAQEC codes—Theorem 7
Examples	
$[[80, 73, 3; 1]]_3$	$[[80, 42, 14; 38]]_3$
$[[80, 69, 4; 1]]_3$	$[[80, 50, 10; 30]]_3$
$[[255, 248, 3; 1]]_4$	$[[255, 193, 20; 62]]_4$
$[[255, 244, 4; 1]]_4$	$[[255, 237, 7; 18]]_4$

5. Conclusions

This paper has been devoted to the use of cyclic codes in the construction of EAQEC codes. General construction methods of EAQEC codes from cyclic codes via defining sets have been presented, using both Euclidean and Hermitian duals of the classical codes. As an application of these methods, five families of EAQEC codes were created. Two of them were derived from Reed–Solomon codes, which resulted in MDS codes. An additional family of almost MDS or near almost MDS EAQEC codes was derived from general cyclic codes. One of the remaining family used BCH codes as the classical counterpart. The construction of this family of EAQEC code used two BCH codes, which provided more freedom in the parameters of the quantum code. Lastly, we conjecture that the family of constructed EAQEC codes can achieve the hashing bound when extending their length. This is supported by the fact that the codes derived have maximal entanglement. Investigations (mainly numerical) along this line are left for future work.

Author Contributions: Conceptualization, F.R.F.P.; methodology, F.R.F.P. and S.M.; investigation, F.R.F.P. and S.M.; writing—original draft preparation, F.R.F.P.; writing—review and editing, F.R.F.P. and S.M.; supervision, S.M.; funding acquisition, F.R.F.P. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: F.R.F.P. was supported partly by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, grant no. 201223/2018-0. S.M. was supported by the European Union’s Horizon 2020 research and innovation programme, under grant agreement QUARTET no. 862644.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: F.R.F.P. is grateful to Ruud Pellikaan for proposing this research and for interesting discussions.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2011.
- Bowen, G. Entanglement required in achieving entanglement-assisted channel capacities. *Phys. Rev. A* **2002**, *66*, 052313-1–052313-8. [CrossRef]
- Fattal, D.; Cubitt, T.S.; Yamamoto, Y.; Bravyi, S.; Chuang, I.L. Entanglement in the stabilizer formalism. *arXiv* **2004**, arXiv:quant-ph/0406168.
- Brun, T.; Devetak, I.; Hsieh, M.H. Correcting Quantum Errors with Entanglement. *Science* **2006**, *314*, 436–439. [CrossRef] [PubMed]
- Galindo, C.; Hernando, F.; Matsumoto, R.; Ruano, D. Entanglement-assisted quantum error-correcting codes over arbitrary finite fields. *Quantum Inf. Process.* **2019**, *18*, 116. [CrossRef]
- Wilde, M.M.; Brun, T.A. Optimal entanglement formulas for entanglement-assisted quantum coding. *Phys. Rev. A* **2008**, *77*, 064302-1–064302-4. [CrossRef]
- Fan, J.; Chen, H.; Xu, J. Constructions of q -ary Entanglement-Assisted Quantum MDS Codes with Minimum Distance Greater than $q + 1$. *Quantum Inf. Comput.* **2016**, *16*, 423–434. [CrossRef]

8. Chen, J.; Huang, Y.; Feng, C.; Chen, R. Entanglement-assisted quantum MDS codes constructed from negacyclic codes. *Quantum Inf. Process.* **2017**, *16*, 303. [CrossRef]
9. Lu, L.; Ma, W.; Li, R.; Ma, Y.; Liu, Y.; Cao, H. Entanglement-assisted quantum MDS codes from constacyclic codes with large minimum distance. *Finite Fields Their Appl.* **2018**, *53*, 309–325. [CrossRef]
10. Chen, X.; Zhu, S.; Kai, X. Entanglement-assisted quantum MDS codes constructed from constacyclic codes. *Quantum Inf. Process.* **2018**, *17*, 273. [CrossRef]
11. Lu, L.; Li, R.; Guo, L. Entanglement-assisted quantum codes from quaternary codes of dimension five. *Int. J. Quantum Inf.* **2017**, *15*, 1750017. [CrossRef]
12. Guenda, K.; Jitman, S.; Gulliver, T.A. Constructions of good entanglement-assisted quantum error correcting codes. *Des. Codes Cryptogr.* **2018**, *86*, 121–136. [CrossRef]
13. Liu, X.; Yu, L.; Hu, P. New entanglement-assisted quantum codes from k -Galois dual codes. *Finite Fields Their Appl.* **2019**, *55*, 21–32. [CrossRef]
14. Koroglu, M.E. New entanglement-assisted MDS quantum codes from constacyclic codes. *Quantum Inf. Process.* **2019**, *18*, 44. [CrossRef]
15. Li, R.; Zuo, F.; Liu, Y. A study of skew asymmetric q^2 -cyclotomic coset and its application. *J. Air Force Eng. Univ. (Nat. Sci. Ed.)* **2011**, *12*, 87–89.
16. Lu, L.; Li, R. Entanglement-assisted quantum codes constructed from primitive quaternary BCH codes. *Int. J. Quantum Inf.* **2014**, *12*, 1450015. [CrossRef]
17. Wilde, M.M.; Hsieh, M.H.; Babar, Z. Entanglement-Assisted Quantum Turbo Codes. *IEEE Trans. Inf. Theory* **2014**, *60*, 1203–1222. [CrossRef]
18. Lai, C.Y.; Brun, T.A.; Wilde, M.M. Duality in Entanglement-Assisted Quantum Error Correction. *IEEE Trans. Inf. Theory* **2013**, *59*, 4020–4024. [CrossRef]
19. Li, R.; Guo, L.; Xu, Z. Entanglement-assisted quantum codes achieving the quantum Singleton bound but violating the quantum Hamming bound. *Quantum Inf. Comput.* **2014**, *14*, 1107–1116. [CrossRef]
20. Pellikaan, R.; Wu, X.W.; Bulygin, S.; Jurrius, R. *Codes, Cryptology and Curves with Computer Algebra*; Cambridge University Press: Cambridge, UK, 2017.
21. Huffman, W.C.; Pless, V. *Fundamentals of Error-Correcting Codes*; Cambridge University Press: Cambridge, UK, 2003.
22. Brun, T.A.; Devetak, I.; Hsieh, M.H. Catalytic Quantum Error Correction. *IEEE Trans. Inf. Theory* **2014**, *60*, 3073–3089. [CrossRef]
23. Guardia, G.G.L. Constructions of new families of nonbinary quantum codes. *Phys. Rev. A* **2009**, *80*, 042331-1–042331-11. [CrossRef]
24. Qian, J.; Zhang, L. Constructions of new entanglement-assisted quantum MDS and almost MDS codes. *Quantum Inf. Process.* **2019**, *18*, 71. [CrossRef]
25. Li, C. Hermitian LCD codes from cyclic codes. *Des. Codes Cryptogr.* **2018**, *86*, 2261–2278. [CrossRef]
26. Lu, L.; Li, R.; Guo, L.; Fu, Q. Maximal entanglement entanglement-assisted quantum codes constructed from linear codes. *Quantum Inf. Process.* **2015**, *14*, 165–182. [CrossRef]
27. Lv, L.; Li, R.; Fu, Q.; Li, X.; Li, X. Maximal entanglement entanglement-assisted quantum codes from quaternary BCH codes. In Proceedings of the IEEE Advanced Information Technology, Electronic and Automation Control Conference, Chongqing, China, 19–20 December 2015.
28. Luo, G.; Cao, X. Two new families of entanglement-assisted quantum MDS codes from generalized Reed–Solomon codes. *Quantum Inf. Process.* **2019**, *18*, 89. [CrossRef]
29. Guo, L.; Fu, Q.; Li, R.; Lu, L. Maximal entanglement entanglement-assisted quantum codes of distance three. *Int. J. Quantum Inf.* **2015**, *13*, 1550002-1–1550002-7. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Article

Binary Classification Quantum Neural Network Model Based on Optimized Grover Algorithm

Wenlin Zhao ¹, Yinuo Wang ², Yingjie Qu ², Hongyang Ma ² and Shumei Wang ^{2,*}

¹ School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266520, China

² School of Science, Qingdao University of Technology, Qingdao 266520, China

* Correspondence: wangshumei@qut.edu.cn

Abstract: We focus on the problem that the Grover algorithm is not suitable for the completely unknown proportion of target solutions. Considering whether the existing quantum classifier used by the current quantum neural network (QNN) to complete the classification task can solve the problem of the classical classifier, this paper proposes a binary quantum neural network classical model based on an optimized Grover algorithm based on partial diffusion. Trial and error is adopted to extend the partial diffusion quantum search algorithm with the known proportion of target solutions to the unknown state, and to apply the characteristics of the supervised learning of the quantum neural network to binary classify the classified data. Experiments show that the proposed method can effectively retrieve quantum states with similar features. The test accuracy of BQM retrieval under the depolarization noise at the 20th period can reach 97% when the depolarization rate is 0.1. It improves the retrieval accuracy by about 4% and 10% compared with MSE and BCE in the same environment.

Keywords: binary classification; Grover algorithm; QNN

Citation: Zhao, W.; Wang, Y.; Qu, Y.; Ma, H.; Wang, S. Binary Classification Quantum Neural Network Model Based on Optimized Grover Algorithm. *Entropy* **2022**, *24*, 1783. <https://doi.org/10.3390/e24121783>

Academic Editors: Brian R. La Cour and Giuliano Benenti

Received: 26 October 2022

Accepted: 29 November 2022

Published: 6 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Internet has many different kinds of data and information that are intersected and stored on social networks, prompting many different research fields to start to pay attention to social networks. Users on social networks obtain the resources that they need by visiting web pages [1–4]. The key to studying social networks is to analyze how these social networks are used. The data analyzed can be used to improve the social network itself, making it more convenient for users to browse the data required. They can also be used to analyze users' preferences to deliver advertisements at designated points. They can also be used to analyze user behavior and to predict the transactions that users participate in [5,6]. One of the main ways to analyze these results is to perform extensive data analysis on the weblogs of these sites [7,8]. Every time a user requests a page or some of the resources on that page, such as video, sound, etc., a new record is added to the weblog of the site [9]. This information contains information about the user's favourite pages (i.e., the most frequently visited pages), the sequence of visits to ordinary pages, and even hints at the user's characteristics. This information analysis method can be called web page using data mining (WUM) based on weblog information [10–12]. To use WUM, a sequence of interactions between a single user and a web page needs to be extracted. The resulting file should contain at least the following fields: the user's IP address, timestamp, requested resource, code for the result of the operation, the previous web address before entering the web page, and the browser used [13,14]. Using and analyzing this sequence, the pattern of the user's access to the web page can be obtained.

The value of big data is essentially reflected as follows: it provides new thinking and means for a human to understand complex systems. In theory, a virtual digital image of the natural world can be constructed by digitizing the real world on a sufficiently small

scale of time and space that carries the running rules of the real world. On the premise of sufficient computing power and efficient data analysis methods, an in-depth analysis of this virtual digital image will make it possible to understand and to discover the existing complex system's operation behavior, state, and law. Big data provides a new way of thinking and a new means for exploring objective laws and for transforming nature and society for human beings. Due to the high and increasing demand for big data mining and analysis work, the era is forced to gradually use more efficient quantum scientific research technology [15] to meet the gap of technology improvement means, and then to improve the efficiency of information extraction work and to promote the progress of more scientific research. In view of the above requirements and improved thinking, this paper makes use of the special advantages of quantum algorithms compared with traditional iterative algorithms [16], as a new field of quantum neural network, and studies the neural network autonomous learning scheme [17] for network log information extraction, which highlights the high-value prospects of the quantum field in big data mining and analysis.

This search hassle can be carried out in $O(N)$ with the use of Grover's quantum search algorithm. Grover's quantum search algorithm [18] makes use of the amplitude amplification approach in quantum computing to attain quadratic acceleration in unsorted search problems. Grover's quantum search algorithm has been efficiently applied on classical computer systems with the usage of a quantum laptop language (QCL). For an unordered database search, the Grover algorithm achieves quadratic acceleration compared with the classical algorithm. Then, analysis, induction, and variant research are carried out [19,20]. Except for the lower bound, Grover's methodology can be used for the case where the λ fraction of the target term is unknown. The fixed point strategy and the trial and error method are the two quantum search strategies that can be implemented.

The Binary QNN Model in this paper is a kind of model based on the Grover algorithm and the QNN supervised learning algorithm. First of all, this is based on the traditional Grover algorithm being analyzed and improved, using Younes' algorithm to improve the search algorithm efficiency, inserting this into the iterative learning process of the quantum neural network [21,22], and using quantum processes to promote the efficiency of the algorithm and neural network to realize multiple network synchronization searching and learning [23,24] for each iteration algorithm to improve the efficiency of solutions [25]. The quantum neural network learning scheme in this paper can be applied to quickly find the user's IP address in massive weblogs, and then accurately and efficiently identifying and classifying relevant and valuable information such as the IP addresses of network logs. The results can be sorted according to the number of search categories to identify and analyze user behavior accurately. It can not only quickly discover the person's record of specific interests, but also can divide the session of a single user, which ensures the accuracy of person document identification and improves the effectiveness of user activity queries.

This article is structured as follows. Section 2 introduces the trial-and-error method (Younes' algorithm) used in this paper, and the basic principles of the QNN supervised learning division task. Section 3 describes our binary QNN model. Section 4 describes the evolution process of the original dataset, and provides entropy analysis to show the advantages of this model. Section 5 summarizes and discusses the role of the proposed model in the development of user behavior pattern prediction.

2. Basic Conception

2.1. Supervised Learning Classification of QNN

Quantum Neural Network (QNN) is a new research field formed by the intersection of quantum physics, mathematics, computer science, information science, cognitive science, complexity science, and other disciplines. As the natural evolution of the traditional neural computing system, it makes full use of the great power of quantum computing to improve upon the information processing capacity of neural computing. By introducing the idea of the superposition of quantum states into the traditional feedforward neural network, QNN can train the quantum interval and quantify the uncertainty of the input

data of training samples. Different data will map to different magnitudes. A multi-layer excitation function is used to increase the fuzziness of the network, and to improve the accuracy and certainty of network pattern recognition [26]. Therefore, the research on the quantum neural network provides beneficial support for the combination of quantum computing and neural computing. The potential of quantum neural networks is that they take advantage of both quantum computing based on coherent superposition and neural computing based on parallel processing. For example, running a deep neural network (DNN) model on a device with limited computing power can be challenging because of the large computational power and memory requirements on the device. However, to solve this problem, a quantum neural network (QNN) has greater potential, which can save computing costs while ensuring the accuracy of DNN training.

QNN comprises quantum state preparation subcircuits and optimization tasks performed by classical controllers [27]. The fact that variable-component subcircuits utilized in QNN produce probability distributions that cannot be efficiently simulated is part of the evidence supporting the claim [28,29]. QNN's main application, similar to DNN's, is to tackle categorization tasks [30]. Practical challenges, such as recognizing handwritten digits and the features of many living creatures, can be categorized as categorization scenes [31,32].

A dataset is given

$$T = \{(x_i, y_i)\}_{i=0}^{N-1} \in (R^{N \times M}, \{0, 1\}^N) \quad (1)$$

According to N examples and M elements in the examples, a QNN is led to research $f_\theta(\cdot)$ to predict the label of a facts set T

$$\min_{\theta} \sum_{i=0}^{N-1} I_{y_i \neq f_\theta(x_i)} \quad (2)$$

where θ is the trainable parameter, and I_z is an indicator function whose value is 1 when the condition z is met; otherwise, it is zero. The quantum classifier realizes the data tag prediction function according to specific rules through the filtered data, and its basic principle is shown in Figure 1. We use quantum classifiers in the research section to specify a QNN for completing the classification task defined in Formula (2). Considering the binary task, it is necessary not only to find a decision rule in Formula (2), but also to output the index j satisfying the pre-determined black box function. Given a trained classifier $f_\theta(\cdot)$, both classical algorithms and previous quantum classifiers require at least $O(T)$ query complexity to find j .

The dataset T is constructed from a given qubit with adjustable interactions, where the qubit composition is represented by x_i and y_i . We learn the interaction from a given training set of each input–output relationship based on the classical backpropagation rule $f_\theta(\cdot)$, and taking x_i as the input to its rule, where the input–output relation is the data pair (x_i, y_i) that constitutes the dataset T . This learning process of qubits is viewed as the desired output algorithm behavior, that is, the quantum network “learns” an algorithm.

A notable theoretical result concerning quantum classifiers is the tradeoff between the computational cost and the training performance shown [33].

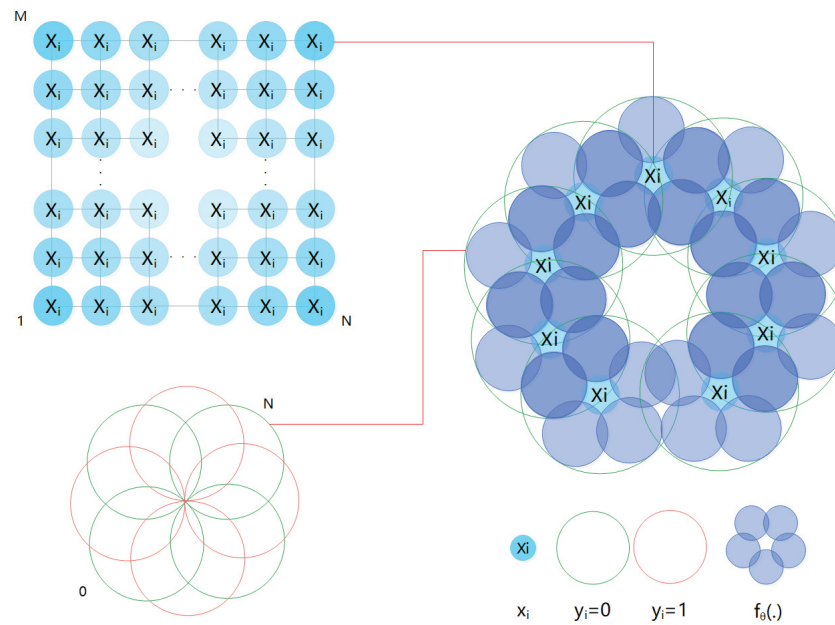


Figure 1. Basic principles of the quantum classifier. x_i randomly generated by the $R^{N \times M}$ matrix, and y_i , which can only be 0 or 1, forms N data pairs and then generates the dataset T . According to the classic backpropagation rule $f_\theta(\cdot)$, x_i is taken as the input to this rule. The qubit obtains the input–output relationship from the data pair (x_i, y_i) in the constructed dataset T , and learns the interaction in the training set of the relationship. In other words, the purpose of the quantum neural network is to use x_i as an input to learn $f_\theta(\cdot)$ rules.

2.2. Younes’ Algorithm

The methodology presented by Younes, Rowe J., and Miller J. [34] in Younes’ algorithm is used to carry out the quantum search by exploiting the local diffusion operator to overcome the soufflé problem in the Grover algorithm. It demonstrates that regardless of whether the number of matches is known, the entire range of $1 \leq M \leq N$ can be consistently handled. It lays the theoretical foundation of the binary QNN model.

In the $|0\rangle$ and $|1\rangle$ states, part of the diffusion operator Q_i system subspace of the entanglement in additional qubit workspace performs about the inverse operation of the mean, and the inverse operation of the phase shift is -1 . H is the Hadamard Gates denoted by $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. The diagonal representation of Q_i applied to the $n + 1$ qubit system is:

$$Q_i = (H^{\otimes n} \otimes I_1)(2|0\rangle\langle 0| - I_{n+1})(H^{\otimes n} \otimes I_1) \tag{3}$$

asthe $|0\rangle$ length is 2^{n+1} , I_1 is the unit of a 2×2 matrix.

Generally, quantum structures of the well-known size $n + 1$ can be expressed as:

$$|\psi\rangle = \sum_{p=0}^{N-1} \alpha_p(|p\rangle \otimes |0\rangle) + \sum_{p=0}^{N-1} \beta_p(|p\rangle \otimes |1\rangle) \tag{4}$$

Applying Q_i to $|\psi\rangle$ bits, we obtain

$$\sum_{p=0}^{N-1} \left(\frac{2}{N} \sum_{p=0}^{N-1} \alpha_p - \alpha_p \right) (|p\rangle \otimes |0\rangle) - \sum_{p=0}^{N-1} \beta_p (|p\rangle \otimes |1\rangle) \tag{5}$$

where $\frac{1}{N} \sum_{p=0}^{N-1} \alpha_j$ means the mean amplitude of subspace $\sum_{p=0}^{N-1} \alpha_p(|p\rangle \otimes |0\rangle)$. That is, the operator Q_i only performs the inversion of the means in the subspace and only changes the sign of the amplitude.

The H gate is utilized to the first n qubits to produce 2^n values to characterize the list. Then, we iteratively observe the Oracle feature U_f to map the goal in the list to 0 or 1, and we retail the outcomes such as $U_f|x, 0\rangle \rightarrow |x, f(x)\rangle$; the partial diffusion operator Q_i is applied, and this step is repeated q times. Finally, the first n qubits are measured.

The variety of iterations q has to be an integer to locate a healthy shut to the change in measurements.

Setting $q = \lfloor \frac{\pi}{2\theta} \rfloor$, as $|q - \bar{q}| \leq \frac{1}{2}, 0 < \theta \leq \frac{\pi}{2}$. As $\cos(\theta) = 1 - \frac{M}{N}$, we can find

$$\theta \geq \frac{\sqrt{2NM - M^2}}{N} \tag{6}$$

$$q = \lfloor \frac{\pi}{2\theta} \rfloor \leq O(\sqrt{\frac{N}{M}}) \tag{7}$$

It is proven that the algorithm can be handled in the range of $1 \leq M \leq N$ using the $O(\sqrt{\frac{N}{M}})$ fixed operator.

3. The Binary QNN Model

We simulate the creation of a binary analysis algorithm that uses quantum states to process information, as shown in Figure 2. The algorithm proposed in this paper is uniformly represented as a BQM field in the following content. As shown in Figure 2, BQM uses a specified variable component subcircuit U_c , and an H gate to replace the Oracle U_f . The variable component subcircuit U_c , based on the training data, can conditionally flip a flag qubit. The tagged qubit is then used as part of the H gate to guide a Grover search algorithm to identify the index of the specified example; i.e., the state of the tagged qubit, such as “0” or “1”, determines the probability of success in identifying the target index. BQM optimizes the trainable parameters of the variable component subcircuit U_c . When the corresponding training instance is positive, the success probability of sampling a target index is maximized. Otherwise, BQM minimizes the probability of the success of sampling target indicators. The design of our algorithm has some advantages in terms of query complexity by inheriting attributes from the Grover search algorithm and performing binary classification tasks on these attributes while allowing the setting of search constraints [35]. Under the above observation, a quantum classifier must have certain advantages [36–38].

3.1. Pretreatment Stage of a Dichotomous Task

In the pretreatment stage, a dichotomous task uses the dataset T defined in Equation (1) as the extended dataset \hat{T} .

To apply the Grover search algorithm to obtain index $i = K - 1$, for $K \in [N]$, the K^{th} pair data training rules T_k are as follows.

$$T_K = [(x_k^{(0)}, y_k^{(0)}), (x_k^{(1)}, y_k^{(1)}), \dots, (x_k^{(K-1)}, y_k^{(K-1)})] \tag{8}$$

The pair of data in T_k is like the K^{th} pair of \hat{T} ; this means that $(x_k^{(K-1)}, y_k^{(K-1)}) = (x_k, y_k)$. The first $K - 1$ pair of $T_k = \{(x_k^{(i)}, y_k^{(i)})\}_{i=0}^{K-2}$ uniformly samples from a subset of \hat{T} , when every label $\{y_k^{(i)}\}_{i=0}^{K-2}$ is the opposite of y_k .

$y_k \in \{0, 1\}$, $\hat{T}^{(0)}$ and $\hat{T}^{(1)}$ are constructed, which contain only those examples of \hat{T} with the labels ‘0’ and ‘1’, respectively. When $y_k = 0$, the pair samples before K from $\hat{T}^{(1)}$; it is same as for the situation where $y_k = 1$, in which the pair samples before K from $\hat{T}^{(0)}$.

Various quantum classifiers encode T_k into quantum states in different ways [39]. For the sake of notation, we indicate that $|\Phi^k\rangle$ that analogously connects with the K^{th} example is

$$|\Phi^k\rangle = U_{data}|0\rangle = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}} |g(x_i)\rangle |i\rangle \tag{9}$$

as $g(\cdot)$ is a coding operation.

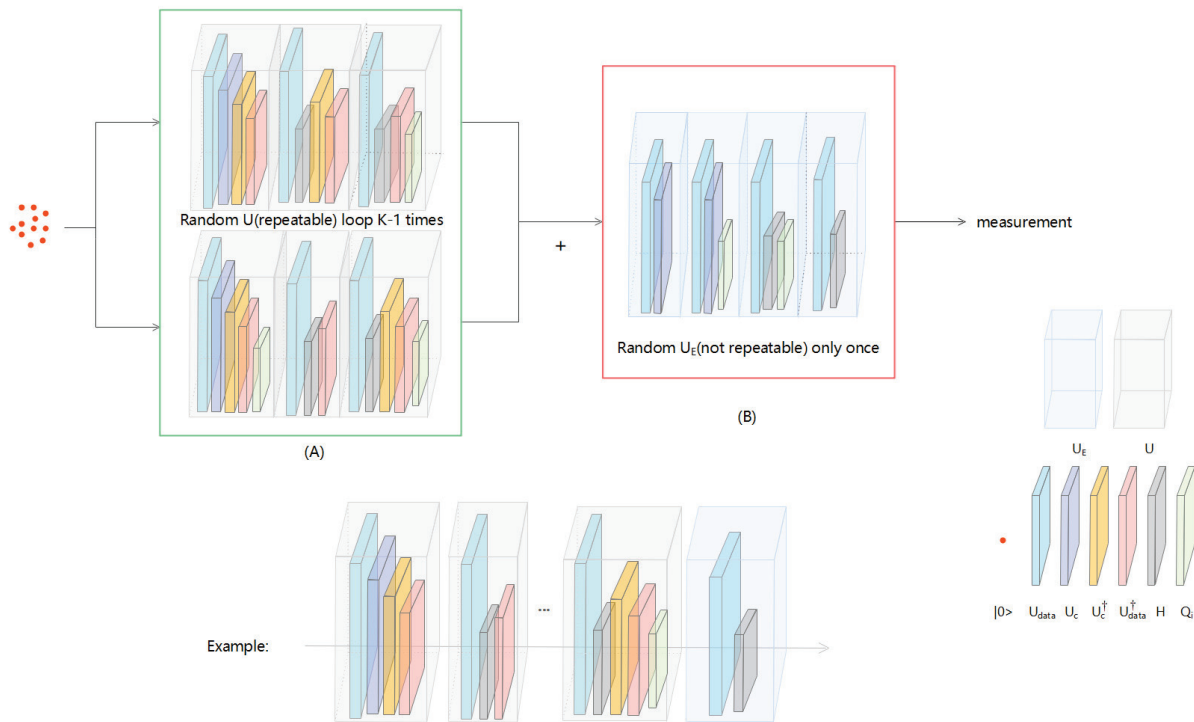


Figure 2. The paradigm of BQM. (A) The first $K-1$ loop uses U , defined in Equation (12), which consists of unitary operators (namely U_{data} , U_c , H , and Q_i). (B) The last cycle uses the unitary operation U_E defined in Equation (13). The qubit interacts with U_c and Q_i to form the feature register and data register.

3.2. The Training Process of the Learning Plan

Compared with the traditional Grover algorithm, combining the variational learning method and the Grover search algorithm produces quantum advantages [40–44]. The adopted variable component subcircuit U_c is designed to find a hyperplane to keep the last pair in the T_k away from the pair of samples before K .

For the variational quantum circuits $U_c(\theta)$ in BQM, a NISQ device scheme consists of a trainable single-qubit gate and two-qubit gates such as CNOT or CZ, which implement generation and discrimination tasks using variational hybrid quantum-classical algorithms. U_c is denoted as $U_c = \prod_{c=1}^C U(\theta^c)$, where each layer $U(\theta^c)$ contains $O(poly(N))$ parameterized single-qubit gates and at most, $O(poly(N))$ fixed two-qubit gates with the same layout.

In the optimal situation, given an initial state $|\Phi^k\rangle$ defined in Equation (9), U_c is applied to obtain the following goals:

1. If the pair of samples before K as T_k analogously connects with the label $y_k = 0$, the expected state is

$$(U_c \otimes I)(U_{data}|0\rangle)_{y_k=0} = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}} |0\rangle |i\rangle \tag{10}$$

- If the pair of samples before K as T_k analogously connects with the label $y_k = 1$, the expected state is

$$(U_c \otimes I)(U_{data}|0\rangle)_{y_k=1} = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}} |1\rangle |i\rangle \tag{11}$$

The label $y_k = 0$ (or $y_k = 1$) as the quantum state of the R_F characteristics register the first qubits $|0\rangle$ (or $|1\rangle$). As shown in Figure 2A, state $(U_c \otimes I)U_{data}|0\rangle$ was prepared. Our binary QNN model iteratively applied the H door to register by the characteristics of the first qubit and index on the index register control, using the U_{data} register and the U_c calculation characteristics, and to finish the first cycle, it applied the diffusion operator Q_i to the index register. All quantum processes, such as U , are part of a period.

$$U = Q_i \circ U_{data}^\dagger \circ (U_c \otimes I)^\dagger \circ H \circ (U_c \otimes I) \circ U_{data} \tag{12}$$

Define $Q_i = I \otimes (\frac{2}{K} \sum_i |i\rangle \langle i| - I)$. Except on a loop, the binary QNN model is repeatedly in the initial state $|0\rangle$ to U , and the application of the unitary operation is replaced with

$$U_E = Q_i \circ H \circ (U_c \otimes I) \circ U_{data} \tag{13}$$

The brown shade in Figure 2B shows this. According to the traditional Grover search algorithm [45], before making quantum measurements, the binary QNN model polls U and U_E for a total of $O(\sqrt{K})$ times.

3.3. The Evolution of the Quantum State

We analyze how quantum states evolve under $y_k = 0$ and $y_k = 1$:

- After interaction with unitary $U_c \otimes I$, using the Equation (10) input state $\Phi_k(y_k = 0)$, this state can be converted to $\frac{1}{\sqrt{K}} \sum_{i=0}^{K-1} |0\rangle |i\rangle$. For all computing in $i \in [K - 1]$, this means that the quantum operation $Q_i \circ U_{data}^\dagger \circ (U_c \otimes I)^\dagger$ does not change state.

$$\frac{1}{\sqrt{K}} (H^{\otimes n} \otimes I)(2|0\rangle \langle 0| - I_{n+1})(H^{\otimes n} \otimes I) \sum_{i=0}^{K-1} |0\rangle |i\rangle = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}} |0\rangle |i\rangle \tag{14}$$

When we measure the indicator register of the output state, the sampling $i \in [K - 1]$ for calculating the base i is distributed.

- After interaction with unitary $U_c \otimes I$ using the Equation (11) input state $\Phi_k(y_k = 1)$, this state can be converted to $\frac{1}{\sqrt{K}} \sum_{i=0}^{K-1} |1\rangle |i\rangle$.

Mathematically, the result state is generated after interaction with H

$$H \circ (U_c \otimes I)(U_{data}|0\rangle)_{y_k=1} = \frac{1}{\sqrt{K}} |0\rangle \sum_{i=0}^{K-2} |i\rangle - \frac{1}{\sqrt{K}} |1\rangle |i^*\rangle \tag{15}$$

where $|i^*\rangle$ for calculating the base $|K - 1\rangle$. The calculation operation $U_{data}^\dagger \circ (U_c \otimes I)^\dagger$ and the diffusion operation Q_i are used to increase $|i^*\rangle$ probability.

After the first cycle, the generated state is generated

$$U(U_{data}|0\rangle)_{y_k=1} = \frac{(K - 4)}{K\sqrt{K}} |0\rangle \sum_{i=0}^{K-2} |i\rangle + \frac{3K - 4}{K\sqrt{K}} |0\rangle |i^*\rangle \tag{16}$$

where Equation (12) defines U . According to Grover’s algorithm, the chance of sampling i^* will increase to $\frac{(3K-4)^2}{K^3}$.

3.4. The Loss Function

With the observation above leading to Theorem 1, the proof is given above:

Theorem 1. For BQM, under the optimal setting, if the label of the last item of T_k is $y_k = 1$, the probability of the sampling resulting in $i^* = K - 1$ is asymptotically 1.

Proof of Theorem 1. We discussed the case where the last entry in T_k has labels $y_k = 1$ and $y_k = 0$.

In the instance of $y_k = 0$, assuming that the label of the final item in T_k is $y_k = 0$, it is possible to determine from Equation (14) that after the first cycle, the generation state of BQM is

$$U_c|\Phi_k(y_k = 0)\rangle = U|0\rangle = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle|i\rangle \tag{17}$$

The chance of picking any index when U (apply to $|0\rangle$) is the same, according to the formula above. After U is applied to $|0\rangle$ via induction, with the migration with time n , the state changes as

$$\prod_{i=0}^N U^i|0\rangle = \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle|i\rangle \tag{18}$$

where the given N is any positive integer, and the probability of sampling $|i^*\rangle$ is $\frac{1}{K}$. In the last loop, the quantum operation U_E defined in Equation (13) is applied to state $\prod_{i=0}^N U^i|0\rangle$ and the resulting state is

$$\begin{aligned} U_E \prod_{i=0}^N U^i|0\rangle &= Q_i \circ H \circ (U_c \otimes I) \circ U_{data} \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle|i\rangle \\ &= \sum_{i=0}^{K-1} \frac{1}{\sqrt{K}}|0\rangle\Lambda \end{aligned} \tag{19}$$

where the first equality uses Equation (18); the second equation uses Equation (15) and exploits the application of the diffusion operator $Q_i = (H^{\otimes n} \otimes I_1)(2|0\rangle\langle 0| - I_{n+1})(H^{\otimes n} \otimes I_1)$, then $\Lambda = \frac{1}{\sqrt{K-1}} \sum_{i=0}^{K-2} |i\rangle + |i^*\rangle$.

In the instance of $y_k = 1$, assuming that the label of the last item in T_k is $y_k = 1$, it is possible to determine from Equation (16) that after the first cycle, the generation state of BQM is

$$U_c|\Phi_k(y_k = 1)\rangle = |0\rangle \otimes \left(\frac{(K-4)}{K\sqrt{K}} \sum_{i=0}^{K-2} |i\rangle + \frac{3K-4}{K\sqrt{K}} |i^*\rangle \right) \tag{20}$$

The chance of picking any index when U (apply to $|0\rangle$) is the same, according to the formula above. After U is applied to $|0\rangle$ by induction, with the migration with time n , the state changes as follows:

$$\prod_{i=0}^n U^i|0\rangle = |0\rangle \otimes \frac{1}{\sqrt{K}} \left[\hbar \sum_{i=0}^{K-2} |i\rangle + \lambda |i^*\rangle \right] \tag{21}$$

where given that n is any positive integer, $\hbar = \cos(2n\alpha) - \frac{1}{\sqrt{K-1}} \sin(2n\alpha)$, $\sin \alpha = \frac{1}{\sqrt{K}}$, $\lambda = \sqrt{K-1} \sin(2n\alpha) + \cos(2n\alpha)$. In the last loop, the quantum operation U_E defined in

Equation (13) is applied to state $\prod_{i=0}^N U^i |0\rangle$, and the resulting state is

$$\begin{aligned}
 U_E \prod_{i=0}^N U^i |0\rangle &= Q_i \circ H \circ (U_c \otimes I) \circ U_{data} |0\rangle \otimes \frac{1}{\sqrt{K}} \hbar \sum_{i=0}^{K-2} |i\rangle + \lambda |i^*\rangle \\
 &= Q_i \frac{[\hbar |0\rangle \sum_{i=0}^{K-2} |i\rangle + \lambda |1\rangle |i^*\rangle]}{\sqrt{K}} \\
 &= \frac{[(K-2)\hbar |0\rangle \sum_{i=0}^{K-2} |i\rangle + 2\sqrt{K-1}\lambda |1\rangle |i^*\rangle]}{\sqrt{K^3}}
 \end{aligned} \tag{22}$$

where the first equality uses Equation (21) and the second equation uses Equation (16) to design the feature register. It uses H to flip the phase of $|i\rangle$ whose first qubit of the feature register is $|1\rangle$, and the last equation comes from the application of the diffusion operator Q_i .

According to Equation (22), in the ideal situation, the probability of sampling i^* is near to 1 when $n \sim O(\sqrt{K})$, and then $(K-1) \sin(2n\alpha) + \sqrt{K-1} \cos(2n\alpha)$ is close to 1.

The result of Equation (19) shows that when $y_k = 0$, the probability of sampling i^* never increases. Thus, we can follow that the sampling probability of the result i^* asymptotically approaches one if and only if the label of the last term of T_k is $y_k = 1$. \square

According to Theorem 1 of the BQM's special property, the output distribution is different for different labels of the input T_k while performing the binary classification task. According to the analysis of Theorem 1 mentioned above, the calculation basis $i = K - 1$ will be present in the output state of the BQM; that is, $U_E U^{O(\sqrt{K})} |0\rangle$, which corresponds to $y_k = 1$, and its probability is close to 1. The matching output state for $y_k = 0$ will, however, include the same computational foundation $i \in [K - 1]$.

According to the mechanism of the Grover search algorithm, the loss function of BQM is deduced as

$$\min_{\theta} L(\theta) = s\left(\frac{1}{2} - y_k\right) \text{Tr}(|1\rangle\langle 1| \otimes H \otimes (|i^*\rangle\langle i^*|) \Delta_{\theta}) \tag{23}$$

where $s(\cdot)$ is the sign function, $\Delta_{\theta} = U_E U(\theta)^{O(\sqrt{K})} |0\rangle\langle 0| (U_E U(\theta)^{O(\sqrt{K})})^{\dagger}$, and $U(\theta)$ is defined in Equation (12).

The success probability of sampling i^* and obtaining the first feature qubit as '1'('0') is maximized (minimized) when $y_k = 1$ ($y_k = 0$), when faced with the challenge of minimizing the loss function $L(\theta)$.

3.5. Gradient-Based Parameter Optimization

The optimization method in this paper uses a multiple-layer parameterized quantum circuit (MPQC), according to the principle that the arrangement of quantum gates in each layer is the same [46], and the operation formed by the layer c is expressed as $U(\theta^c)$, produced by quantum states produced by MPQC

$$|\omega\rangle = \prod_{c=1}^C U(\theta^c) |0\rangle^{\otimes N} \tag{24}$$

where C is the total number of layers. BQM uses MPQC to construct U_c

$$U_c(\theta) = \prod_{c=1}^C U(\theta^c) \tag{25}$$

The circuit layout of $U(\theta^c)$ at layer l is shown in Figure 3. When the number of layers is C , the total number of trainable parameters of BQM is $2MC$.

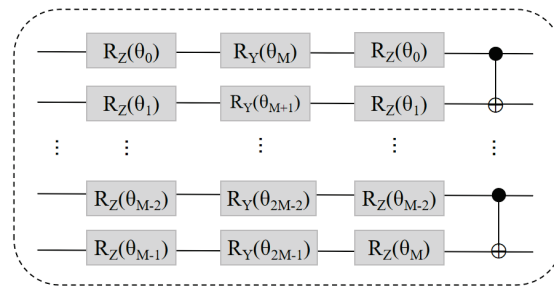


Figure 3. The realization of the l^{th} layer $U(\theta^c)$. It is assumed that the l^{th} layer $U(\theta^c)$ interacts with M qubits. Three trainable parameterized gates R_Z , R_Y , and R_Z are first applied to each qubit, followed by the $M - 1$ CNOT gates.

The update rules of BQM at the j^{th} iteration are as follows

$$\theta^{(j+1)} = \theta^{(j)} - \zeta \frac{L(\theta^{(j)}, T_j)}{\partial \theta} \tag{26}$$

where ζ is the learning rate. Given the explicit form of $L(\theta)$ in the defined Equation (23), the gradient of $L(\theta^{(j)}, T_j)$ can be rewritten as

$$\frac{\partial L(\theta^{(j)}, T_j)}{\partial \theta} = s\left(\frac{1}{2} - y_j\right) \frac{\partial Tr(\prod \Delta_{\theta^{(j)}})}{\partial \theta} \tag{27}$$

where y_j is the label of the last item in T_j , $s(\cdot)$ is the symbol function, and \prod is the measurement operator.

BQM employs a gradient-based method, according to the parameter displacement rule, to obtain the gradient $\frac{\partial Tr(\prod \Delta_{\theta^{(j)}})}{\partial \theta}$, to optimize θ . The parameter shift rule [47] iteratively calculates each gradient entry under its guiding principle.

For $e \in [2NC]$, only the e^{th} parameter is rotated by $\pm \frac{\pi}{2}$, i.e.,

$$\theta_{\pm}^{(j)} = [\theta_0^{(j)}, \dots, \theta_{e-1}^{(j)}, \theta_e^{(j)} \pm \frac{\pi}{2}, \theta_{e+1}^{(j)}, \dots, \theta_{2NC-1}^{(j)}] \tag{28}$$

Combining Equations (26)–(28), the update rule of BQM at the e^{th} iteration of the e item is

$$\theta_e^{(j+1)} = \theta_e^j - \zeta s\left(\frac{1}{2} - y_j\right) \frac{\partial Tr(\prod \Delta_{\theta_{\pm}^{(j)}})}{2} \tag{29}$$

where $\Delta_{\theta_{\pm}^{(j)}} = U_E U(\theta_{\pm}^{(j)})^{O(\sqrt{K})} |0\rangle\langle 0| (U_E U(\theta_{\mp}^{(j)})^{O(\sqrt{K})})^\dagger$.

3.6. Circuit Implementation of Label Prediction

After the training of BQM is completed, the trained U_c can use the corresponding circuit (as shown in Figure 4) to predict the label of an instance with $O(1)$ query complexity.

Denoting the new input as (x, y) , we encode a into quantum states using the same encoding method used during training; i.e., $|\tilde{\chi}\rangle = |g(x)\rangle$, then we apply the trained U_c to $|\tilde{\chi}\rangle$.

When the size of the dataset loaded by the binary QNN model is K , a well-trained binary QNN model can obtain the index with $O(\sqrt{\frac{K}{MT^2}})$ query complexity.

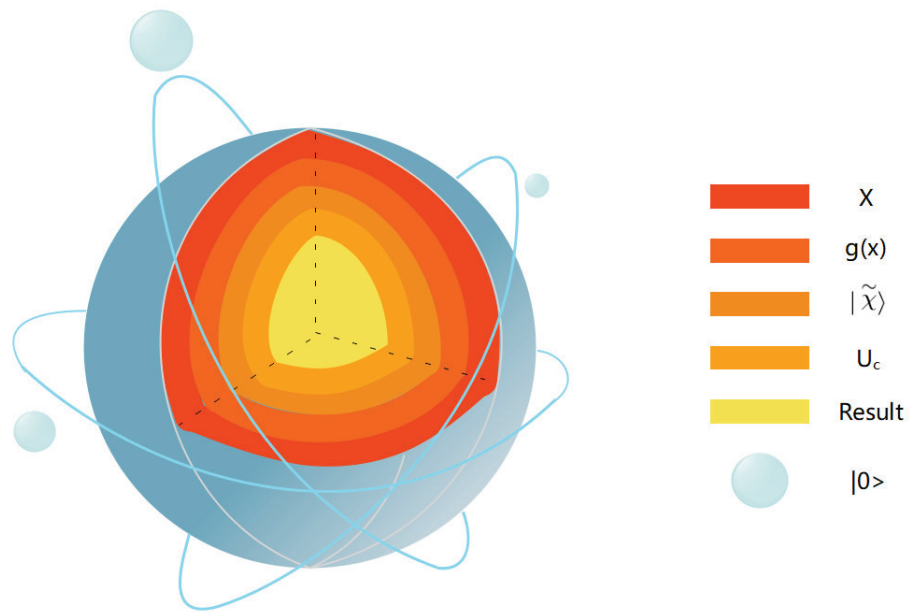


Figure 4. Circuit implementation of BQM prediction. Use $g(\cdot)$ as in the training process. The encoding method prepares the state $|g(x)\rangle$ and applies the trained variable component subcircuit U_c to $|\tilde{\chi}\rangle$.

3.7. Synthetic Construction of Datasets

Given the training example $x_i = (\alpha^{(j)}, \beta^{(j)}) \in \mathbb{R}^2$, the embedded function $f(\alpha^{(j)}, \beta^{(j)})$ used to encode x_i into a quantum state is represented as

$$f(\alpha^{(j)}, \beta^{(j)}) = (R(\gamma(\alpha^{(j)}, \beta^{(j)})) \otimes R(\gamma(\alpha^{(j)}, \beta^{(j)})))|0\rangle^{\otimes 2} \quad (30)$$

where $\gamma(\alpha^{(j)}, \beta^{(j)}) = (\alpha^{(j)}, \beta^{(j)})^2$ is a specified mapping function. The above formula means that $g(x_i)$ can be converted into a series of quantum operations, the implementation of which is shown in Figure 5a. To encode multiple training examples into quantum states simultaneously, we should treat $f(x_i)$ as a controlled version, the implementation of which is shown in Figure 5b.

3.8. The Details of BQM

The implementation of GBLS is shown in Figure 5c. In it, the data encoding the unitary U_{data} consists of a controlled set of $f(x_i)$ quantum operations. The implementation of encoding unitary U_{data} depends on the size of the batch B . For the quantum kernel classifier with BCE loss and MSE loss ($B = M$), it can be seen from Formula (30) that the unitary encoding is

$$U_{data} = R(\gamma(\alpha^{(j)}, \beta^{(j)})) \otimes R(\gamma(\alpha^{(j)}, \beta^{(j)})) \quad (31)$$

For a quantum kernel classifier with MSE loss ($B = M/4$), the implementation of encoding unitary U_{data} is the same as that of BQM, as shown in Figure 5.

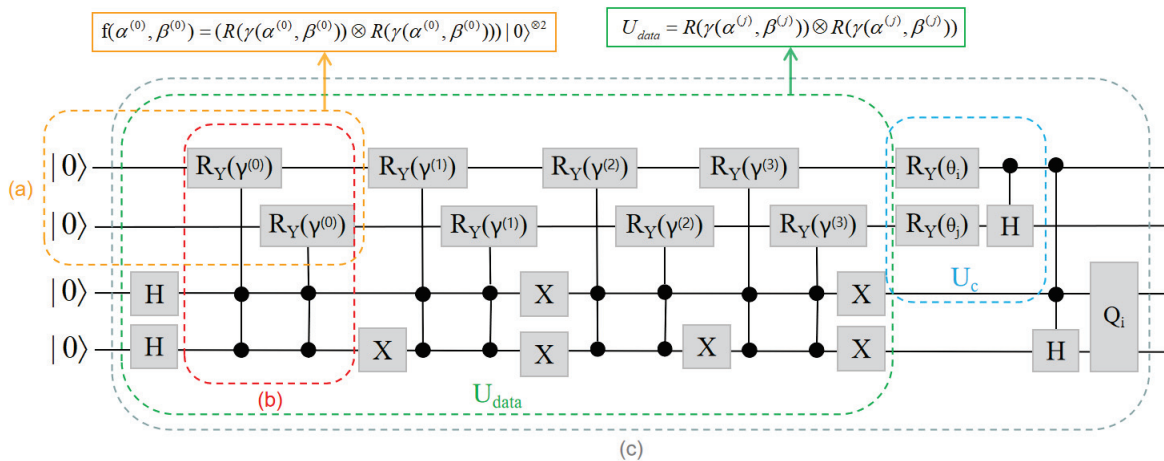


Figure 5. Implementation of BQM in numerical simulation. (a) The circuit implementation of the encoded unitary U_{data} corresponding to the feature map $f(x_i)$ is illustrated. (b) The realization of quantum operation $f(x_i)$. (c) The implementation of BQM, given input $T_k = \{x_i, x_j, x_m, x_n\}$.

4. Results

We will analyze the security of the proposed BQM, intuitively express the evolution and generation process of the dataset by using the dot plot, and evaluate the algorithm’s performance through entropy analysis and testing [48–50].

4.1. Dataset Evolution

This algorithm’s dataset generation process is shown in Figure 6. It uses the top $K - 1$ pair and the K^{th} pair in the original dataset for label classification and uniform sampling. It divides all data pairs into sub-datasets labeled as 1 (or 0) according to the y value of 0 (or 1).

The yellow cube in the figure represents the data point in the data pair whose value of y is 1, the blue cube represents the data point in the data pair whose value of y is 0, and the point whose circle in red represents the K^{th} data point. The specific rules are as follows:

1. Represents the original dataset when $y_k = 1$ as the ‘A’ cube or when $y_k = 0$ as the ‘B’ cube in the K^{th} pair of data (x_k, y_k) ;
2. Represents the uniform sampling from the sub-dataset labeled 1 in Figure 6a to generate a new dataset in Figure 6b;
3. Represents the uniform sampling from the sub-dataset labeled 0 in Figure 6a to generate a new dataset in Figure 6c.

4.2. Stability and Convergence Analysis

Define a utility-bound R as a utility measure to evaluate the distance between the optimization result and the stationary point in the optimized environment.

$$R = \mathbb{E}[||\nabla_{\theta}L(\theta^{(j)})||^2] \leq \varepsilon(j) \tag{32}$$

For the BQM quantum classifier with a depolarization noise setting, the utility bound of output $\theta^{(j)} \in R^l$ after j iterations is

$$\varepsilon(j) = O(poly(\frac{l}{j(1-P)^d}, \frac{l}{BK(1-P)^d}, \frac{l}{(1-P)^d})) \tag{33}$$

where P is the depolarization rate, l is the total number of trainable parameters, K is the number of measurements to estimate the quantum expected value, d is the circuit depth of the variable component sub-circuit, and B is the number of batches.

We use the decay rate of $\log(\varepsilon(j))$ to define the asymptotic convergence rate of this optimization algorithm [51,52]. According to Equation (33), the attenuation rate of $\log(\varepsilon(j))$ is slower than that of $-j$, which proves that this algorithm has a sublinear convergence rate.

When $B = M$, we input each sample T_j in turn to variable component subcircuits to obtain $\nabla L(\theta, T_j)$. Once the set $\{\nabla L(\theta, T_j)\}_{j=1}^M$ is collected, the gradient $\nabla L(\theta, T)$ can be estimated by $\frac{1}{M} \sum_{j=1}^M \nabla L(\theta, T_j)$. Assuming that the number of measurements required to estimate the derivative of the JTH parameter θ_j is K , the total number of measurements obtained is MK for $\frac{1}{M} \sum_{j=1}^M \nabla L(\theta, T_j)$. Therefore, the estimate of $\nabla L(\theta, T_j)$ with l parameters requires MKl measurements.

For the above definition of utility bound R , the results show that a large number of lot B can guarantee a better realization of utility bound R by increasing the total number of measurements.

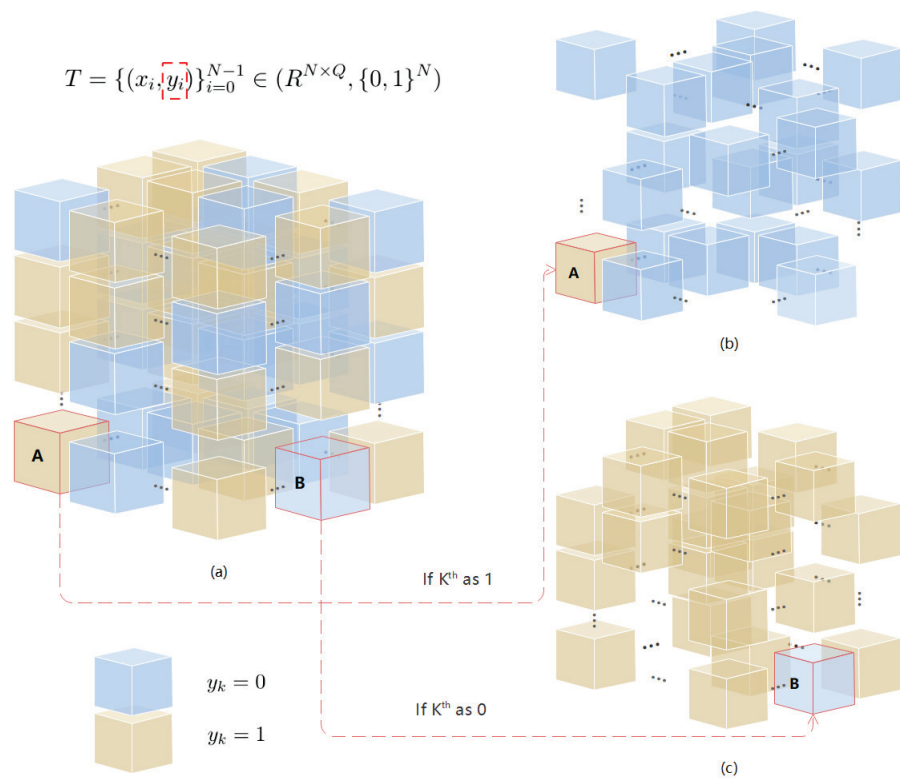


Figure 6. Dataset evolution. The yellow cube in the figure represents the data point in the data pair whose $y = 1$, the blue cube represents the data point in the data pair whose $y = 0$, and the point circle in red represents the K^{th} data point. (a) The traditional dataset when defined as T , which is defined in Equation (1). (b) When $K^{th} = 1$ (that is, the red grid labeled A), according to the generation formula, K^{th} is combined with the first $K - 1$ labels with $y = 0$, and the resulting dataset. (c) When $K^{th} = 0$ (that is, the red grid labeled B), according to the generation formula, K^{th} is combined with the first $K - 1$ labels with $y = 1$, and the resulting dataset.

4.3. Performance Analysis under Depolarization Noise

We employ depolarization channels to mimic the system noise, since the number of measurements and the quantum system’s noise are both constrained. We next examine how well BQM performs in the presence of depolarization noise [53].

If a quantum state is ω , we define the depolarizing channel ω_P acting on this quantum state as

$$\omega_P(\omega) = (1 - P)\omega + P \frac{I_l}{l} \tag{34}$$

where P is the depolarization rate, and l is the total number of trainable parameters.

We compare the performance of BQM and two other quantum kernel classifiers when quantum system noise and measurement delays are considered. Among them, BQM stands for a binary classification quantum neural network model based on the optimized Grover algorithm proposed in this paper. The two classifiers compared with BQM are defined as “BCE” and “MSE”, respectively. “BCE” stands for the quantum kernel classifier with binary cross-entropy loss, and “MSE” means the quantum kernel classifier with the mean square error loss ($B = N$). We simulated the statistics for each of the three classifiers by repeating the values 10 times. Figure 7 illustrates the simulation findings. After 20 periods, BQM, BCE, and MSE quantum classifiers achieve the same performances. It can be observed that the quantum classifiers with MSE loss have lower convergence speeds and larger variances than the BQM and BCE classifiers. This phenomenon reflects that using BQM for classification tasks with different batches is meaningful. In Table 1, we compare the average training and testing accuracies of the BQM, BCE, and MSE quantum classifiers in the last stage. Considering the measurement error and quantum gate noise, BQM still achieves a stable performance because of its minimal variance.

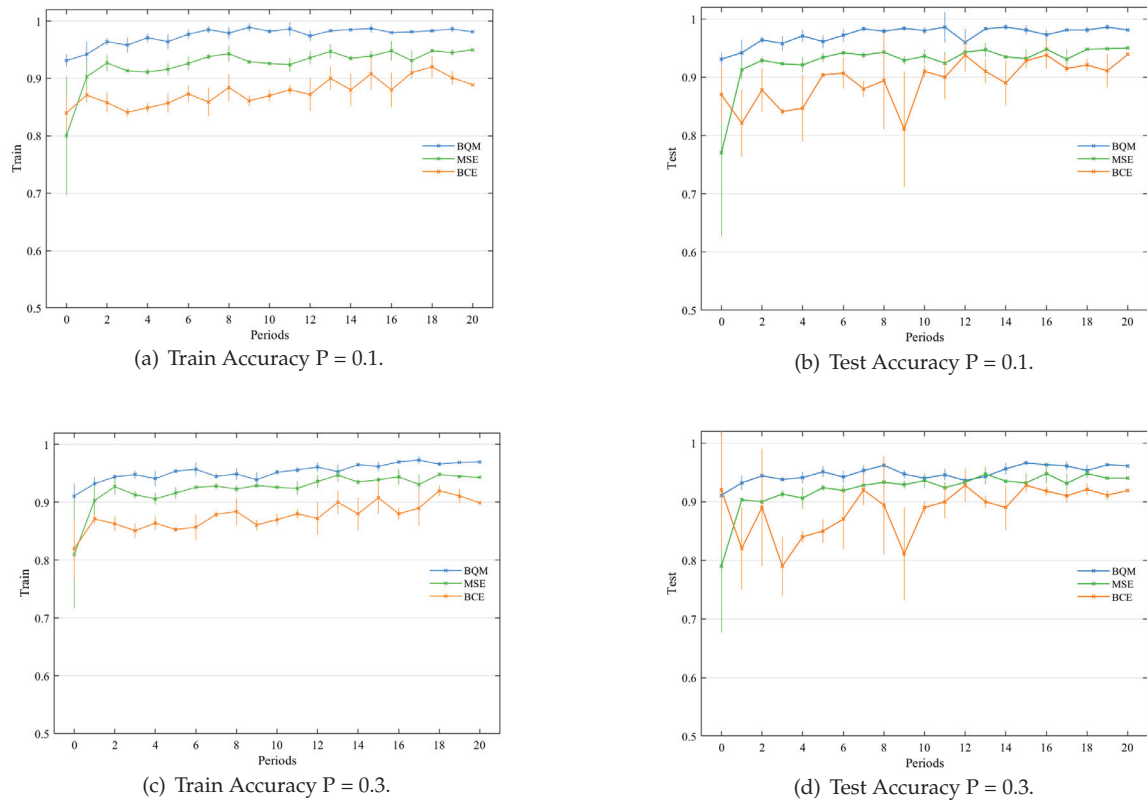


Figure 7. The performance of different quantum classifiers at the different depolarization rates ($P = 0.1, 0.3$). Depolarizing noise models extracted from quantum hardware are applied to the trainable unitary $U_c(\theta)$ of these three classifiers. The labels ‘BQM’, ‘BCE’, and ‘MSE’ refer to the proposed Grover-based quantum classifier, the quantum kernel classifier with BCE loss, and the quantum kernel classifier with mean square error loss. (a,b) shows the variation of the train and test accuracies of BQM and the quantum kernel classifier with BCE loss with a P value of 0.1. (c,d) show the variation of the train and test accuracies of BQM and the quantum kernel classifier with BCE loss when the P value is 0.3. Vertical bars reflect the variance of the train and test accuracy at each iteration.

The binary QNN model based on Grover, quantum kernel classifier BCE loss, and quantum kernel classifier mean square error loss is reflected by the labels ‘BQM’, ‘BCE’, and ‘MSE’. The train and test accuracies of the BQM quantum classifier are shown in the left and right figures. The vertical bar represents the train and test accuracy variation at

each iteration, where all hyperparameter settings are the same as those used in the above numerical simulation.

According to the numerical simulation results of the three quantum classifiers in Figure 7, BQM can obtain a good utility boundary R through some tests. When they achieve basically comparable performances, BQM reduces the number of measurements required by $K = 4$ times compared to quantum classifiers with BCE losses and MSE losses ($B = N$). This result shows that when N is larger, there is a large separation of computational efficiency between BQM and the previous $B = N$ quantum classifier.

The above data demonstrate that, when BQM is compared to the other two quantum classifiers, the number of measurements required by BQM is decreased by four times, demonstrating BQM’s efficacy.

Table 1. The average training and testing accuracies of BQM, BCE, and MSE quantum classifiers in the last stage. The value ‘ $a \pm b$ ’ means that the average precision is a and its variance is b . The labels ‘BQM’, ‘BCE’, and ‘MSE’ refer to the proposed Grover-based quantum classifier, the quantum kernel classifier with BCE loss, and the quantum kernel classifier with mean square error loss.

Algorithm’s Name	$P = 0.1$ (Train)	$P = 0.1$ (Test)	$P = 0.3$ (Train)	$P = 0.3$ (Test)
BCE	0.883 ± 0.034	0.871 ± 0.071	0.924 ± 0.042	0.799 ± 0.061
MSE	0.941 ± 0.017	0.938 ± 0.008	0.929 ± 0.034	0.917 ± 0.011
BQM	0.977 ± 0.011	0.971 ± 0.007	0.951 ± 0.027	0.949 ± 0.010

4.4. Complex Comparison Analysis

After receiving the encoded data in the variable component subcircuit of the quantum classifier, the measurement operator defines a query as one measurement. The iterative process of this algorithm is shown in Figure 8. According to the quantum classifier’s training mechanism, calculating the total number of measurements for variable component subcircuits is comparable to the query complexity of obtaining the gradient in a time frame.

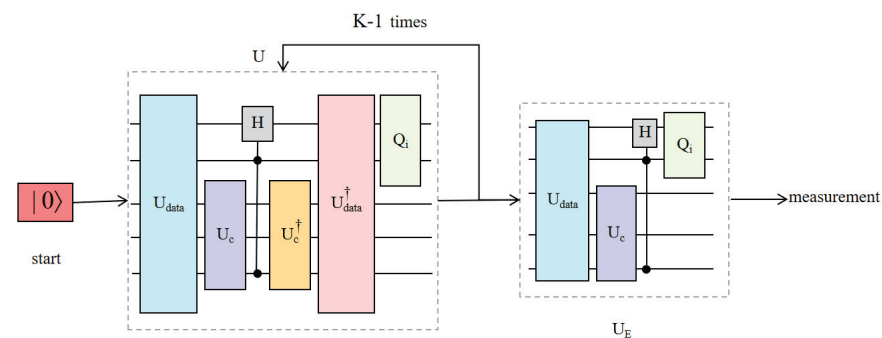


Figure 8. The iterated process of BQM algorithm. After receiving the encoded data in the variable component subcircuit of the quantum classifier, the measurement operator defines a query as one measurement.

The next step is to derive the number of measurements required for a quantum kernel classifier with BCE loss in one period. For the dataset T , the BCE loss is generated

$$L_{BCE} = -\frac{1}{N} \sum_{i=0}^{N-1} y_i \log(P(y_i)) + (1 - y_i) \log(1 - P(y_i)) \tag{35}$$

where y_i is the label of the i^{th} example, and $P(y_i)$ is the prediction probability of the label y_i ; the output of its quantum circuit is

$$P(y_i) = Tr((|1\rangle\langle 1|) \otimes H \otimes \nabla) \tag{36}$$

where $\nabla_{\theta} = |i^*\rangle\langle i^*|U_c(\theta)|0\rangle\langle 0|U_c(\theta)^\dagger$, $U_c(\theta)$ is defined in Equation (25), and $(|1\rangle\langle 1|) \otimes H \otimes (|i^*\rangle\langle i^*|) = \Pi$ is the measurement operator. According to the parameter displacement rule, the derivative of BCE loss is satisfied

$$\frac{\partial L_{\text{BCE}}}{\partial \theta_e} = \frac{1}{N} \sum_{i=0}^{N-1} \rho \frac{\text{Tr}(\Pi \nabla_{\theta_+}) - \text{Tr}(\Pi \nabla_{\theta_-})}{2} \tag{37}$$

where θ_{\pm} is defined in Equation (28), $\rho = \frac{1-y_i}{1-P(y_i)} - \frac{y_i}{P(y_i)}$. To obtain the gradient of BCE loss according to the above equation, we need to give each training example to the quantum kernel classifier to estimate $P(y_i)$, and then calculate the coefficient ρ .

BQM uses the superposition property of the loss function L defined in Equation (17) to obtain the gradient $\frac{\partial L}{\partial \theta_e}$. According to Equation (23), the gradient of BQM satisfies

$$\frac{\partial L(\theta, T_k)}{\partial \theta_e} = \frac{s}{2} \left(1 - y_k (\text{Tr}(\Pi \nabla_{\theta_+^{(k)}}) - \text{Tr}(\Pi \nabla_{\theta_-^{(k)}})) \right) \tag{38}$$

where y_k refers to the label of the last pair in the training example T_k . The gradient of T_k may be calculated using $2K$ measurements, where the first K measurement aims to approach $\text{Tr}(\Pi \nabla_{\theta_-^{(k)}})$ and the last K measurement aims to approximate $\text{Tr}(\Pi \nabla_{\theta_+^{(k)}})$, according to the equation above.

Complex comparison analysis determines the effect of the dataset size on the binary QNN model in this paper. The standard Grover search algorithm’s search complexity is $O(\sqrt{KI})$ for information data entries of size K and the total number of trainable parameters l . The optimal algorithm classification complexity value is $O(\sqrt{\frac{KI}{MT^2}})$, as can be shown in the following Table 2. The reduced query complexity of BQM implies a potential quantum advantage in completing the classification task.

Table 2. Query complexity in several algorithms. The notations T, K, M, and I refer to the batch size range, the wide variety of measurements used to estimate quantum expectation value, the complete variety of education examples, and the total number of trainable parameters.

Algorithm’s Name	Query Complexity
Grover	$O(\sqrt{KI})$
Younes’ algorithm	$O(\sqrt{\frac{KI}{M}})$
BCE	$O(KMI)$
MSE	$O(KMI)$
BQM	$O(\sqrt{\frac{KI}{MT^2}})$

5. Conclusions

As an essential source of information for value-added social media websites, user behavior patterns are the key to fast and accurate identification through session division to realize extensive data analysis. In this paper, based on the trial-and-error method of the Grover search algorithm, combined with the binary classification task of QNN supervised learning, the advantages of the Grover quantum algorithm are brought into play in the quantum classifier of the quantum neural network. The data are preprocessed by analyzing the network search data to realize the construction of the BQM algorithm. They lay the foundation for the development of user behavior pattern prediction. The experimental data show that the application effect of this algorithm has a more apparent accurate recognition rate than the other two classifiers, and it still has a prominent effect in the depolarized noise environment. It can play a supervisory role in the security detection of future users’ network search behaviors.

Author Contributions: Conceptualization, W.Z.; methodology, W.Z.; software, H.M.; validation, Y.W.; formal analysis, Y.Q.; data curation, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Nos. 11975132, 61772295), the Natural Science Foundation of Shandong Province, China (Nos. ZR2021MF049, ZR2019YQ01), and the Project of Shandong Provincial Natural Science Foundation Joint Fund Application (ZR202108020011).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Bao, Z.; Zhou, J.; Tay, Y.C. sonSQL: An Extensible Relational DBMS for Social Network Start-Ups. In Proceedings of the 32nd International Conference on Conceptual Modeling (ER 2013), Hong Kong, China, 11–13 November 2013; Volume 8217, pp. 495–498.
- Rosen, D.; Kim, B. Social networks and online environments: When science and practice co-evolve. *Soc. Netw. Anal. Min.* **2011**, *1*, 27–42. [CrossRef]
- Zhang, Y.; Ling, W. A comparative study of information diffusion in weblogs and microblogs based on social network analysis. *Chin. J. Libr. Inf. Sci.* **2012**, *5*, 51–66.
- Xu, L.; Jiang, C.; Wang, J.; Yuan, J.; Ren, Y. Information Security in Big Data: Privacy and Data Mining. *Chin. J. Libr. Inf. Sci.* **2014**, *2*, 2169–3536.
- Witten, I.; EibeFrank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, 1st ed.; China Machine Press: Beijing, China, 2005.
- Nasraoui, O. Web data mining: Exploring hyperlinks, contents, and usage data. *SIGKDD Explor. Newsl.* **2008**, *10*, 23–25. [CrossRef]
- Tang, H.S.; Yao, Y.W. Research on Decision Tree in Data Mining. *Appl. Res. Comput.* **2001**, *8*, 18.
- Chen, M.S.; Park, J.S.; Yu, P.S. Efficient data mining for path traversal patterns. *IEEE Trans. Knowl. Data Eng.* **1998**, *10*, 209–221. [CrossRef]
- Juntao, H.U.; Defeng, W.U.; Li, G.; Gan, Y. Architecture and Technique of Multimedia Data Mining. *Comput. Eng.* **2003**, *29*, 149–151.
- Shou, Y.J. Data Mining Technique. *Autom.-Petro-Chem. Ind.* **2000**, *6*, 38.
- Yang, Y.; Adelstein, S.J.; Kassis, A.I. Target discovery from data mining approaches. *Drug Discov. Today* **2009**, *14*, 147–154. [CrossRef]
- Mahmood, N.; Hafeez, Y.; Iqbal, K.; Hussain, S.; Aqib, M.; Jamal, M.; Song, O.Y. Mining Software Repository for Cleaning Bugs Using Data Mining Technique. *Comput. Mater. Contin.* **2021**, *69*, 873–893. [CrossRef]
- Helma, C.; Kramer, S. Machine Learning and Data Mining. *Predict. Toxicol.* **2005**, *42*, 223–254.
- Wang, Y.; Li, G.; Xu, Y.; Hu, J. An Algorithm for Mining of Association Rules for the Information Communication Network Alarms Based on Swarm Intelligence. *Math. Probl. Eng.* **2014**, *2014*, 894205. [CrossRef]
- Chen, Z.Y.; Qiu, T.H.; Zhang, W.B.; Ma, H.Y. Effects of initial states on the quantum correlations in the generalized Grover search algorithm. *Chin. Phys. B* **2021**, *30*, 080303. [CrossRef]
- Xu, Z.; Ying, M.; Valiron, B. Reasoning about Recursive Quantum Programs. *arXiv* **2021**, arXiv:2107.11679.
- Xue, Y.J.; Wang, H.W.; Tian, Y.B.; Wang, Y.N.; Wang, Y.X.; Wang, S.M. Quantum Information Protection Scheme Based on Reinforcement Learning for Periodic Surface Codes. *Quantum Eng.* **2022**, *2022*, 7643871. [CrossRef]
- Galindo, A.; Martín-Delgado, M.A. Family of Grover’s quantum-searching algorithms. *Phys. Rev. A* **2000**, *62*, 062303. [CrossRef]
- Ma, H.; Ma, Y.; Zhang, W.; Zhao, X.; Chu, P. Development of Video Encryption Scheme Based on Quantum Controlled Dense Coding Using GHZ State for Smart Home Scenario. *Wirel. Pers. Commun.* **2022**, *123*, 295–309. [CrossRef]
- Zhu, D.; Linke, N.M.; Benedetti, M.; Landsman, K.A.; Nguyen, N.H.; Alderete, C.H.; Perdomo-Ortiz, A.; Korda, N.; Garfoot, A.; Brecque, C.; et al. Training of quantum circuits on a hybrid quantum computer. *Sci. Adv.* **2019**, *5*, aaw9918. [CrossRef] [PubMed]
- Wang, H.W.; Xue, Y.J.; Ma, Y.L.; Hua, N.; Ma, H.Y. Determination of quantum toric error correction code threshold using convolutional neural network decoders. *Chin. Phys. B* **2022**, *31*, 010303. [CrossRef]
- Zheng, W.; Yin, L. Characterization inference based on joint-optimization of multi-layer semantics and deep fusion matching network. *PeerJ. Comput. Sci.* **2022**, *8*, e908. [CrossRef]
- Singh, M.P.; Radhey, K.; Rajput, B.S. Pattern Classifications Using Grover’s and Ventura’s Algorithms in a Two-qubits System. *Int. J. Theor. Phys.* **2018**, *57*, 692–705. [CrossRef]
- Huang, C.Q.; Jiang, F.; Huang, Q.H.; Wang, X.Z.; Han, Z.M.; Huang, W.Y. Dual-Graph Attention Convolution Network for 3-D Point Cloud Classification. *IEEE Trans. Neural Networks Learn. Syst.* **2022**, 1–13. [CrossRef] [PubMed]

25. Ding, L.; Wang, H.; Wang, Y.; Wang, S. Based on Quantum Topological Stabilizer Color Code Morphism Neural Network Decoder. *Quantum Eng.* **2022**, *2022*. [CrossRef]
26. Zhou, G.; Zhang, R.; Huang, S. Generalized Buffering Algorithm. *IEEE Access* **2021**, *99*, 1. [CrossRef]
27. Xu, Y.; Zhang, X.; Gai, H. Quantum Neural Networks for Face Recognition Classifier. *Procedia Eng.* **2011**, *15*, 1319–1323. [CrossRef]
28. Zhang, Y.C.; Bao, W.S.; Xiang, W.; Fu, X.Q. Decoherence in optimized quantum random-walk search algorithm. *Chin. Phys.* **2015**, *24*, 197–202. [CrossRef]
29. Tseng, H.-Y.; Tsai, C.-W.; Hwang, T.; Li, C.-M. Quantum Secret Sharing Based on Quantum Search Algorithm. *Int. J. Theor. Phys.* **2012**, *51*, 3101–3108. [CrossRef]
30. Chen, J.; Liu, L.; Liu, Y.; Zeng, X. A Learning Framework for n-Bit Quantized Neural Networks toward FPGAs. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1067–1081. [CrossRef]
31. Xiao, Y.; Huang, W.; Oh, S.K.; Zhu, L. A polynomial kernel neural network classifier based on random sampling and information gain. *Appl. Intell.* **2021**, *52*, 6398–6412. [CrossRef]
32. He, Z.; Zhang, H.; Zhao, J.; Qian, Q. Classification of power quality disturbances using quantum neural network and DS evidence fusion. *Eur. Trans. Electr. Power* **2012**, *22*, 533–547. [CrossRef]
33. Du, Y.; Hsieh, M.H.; Liu, T.; Tao, D. A Grover-search Based Quantum Learning Scheme for Classification. *New J. Phys.* **2021**, *23*, 20. [CrossRef]
34. Younes, A.; Rowe, J.; Miller, J. Enhanced quantum searching via entanglement and partial diffusion. *Phys. Nonlinear Phenom.* **2008**, *237*, 1074–1078. [CrossRef]
35. Liang, X.; Luo, L.; Hu, S.; Li, Y. Mapping the knowledge frontiers and evolution of decision making based on agent-based modeling. *Knowl. Based Syst.* **2022**, *250*, 108982. [CrossRef]
36. Liu, Z.; Chen, H. Analyzing and Improving the Secure Quantum Dialogue Protocol Based on Four-Qubit Cluster State. *Int. J. Theor. Phys.* **2020**, *59*, 2120–2126. [CrossRef]
37. Chakrabarty, I.; Khan, S.; Singh, V. Dynamic Grover search: Applications in recommendation systems and optimization problems. *Quantum Inf. Process.* **2017**, *16*, 153. [CrossRef]
38. Song, J.; Ke, Z.; Zhang, W.; Ma, Y.; Ma, H. Quantum Confidentiality Query Protocol Based on Bell State Identity. *Int. J. Theor. Phys.* **2022**, *61*, 52. [CrossRef]
39. Panella, M.; Martinelli, G. Neural networks with quantum architecture and quantum learning. *Int. J. Circuit Theory Appl.* **2011**, *39*, 61–77. [CrossRef]
40. Carlo, C.; Stefano, M. On Grover's search algorithm from a quantum information geometry viewpoint. *Phys. Stat. Mech. Its Appl.* **2012**, *391*, 1610–1625.
41. Ashraf, I.; Mahmood, T.; Lakshminarayanan, V. A Modification of Grover's Quantum Search Algorithm. *Photonics Optoelectron* **2012**, *1*, 20–24.
42. Long, G.; Liu, Y.; Loss, D. Search an unsorted database with quantum mechanics. *Front. Comput. Sci. China* **2007**, *1*, 247–271. [CrossRef]
43. Bin, C.; Zhang, W.; Wang, X.; Zhao, J.; Gu, Y.; Zhang, Y. A Memetic Algorithm Based on Two_Arch2 for Multi-depot Heterogeneous-vehicle Capacitated Arc Routing Problem. *Swarm Evol. Comput.* **2021**, *63*, 100864.
44. Ma, Z.; Zheng, W.; Chen, X.; Yin, L. Joint embedding VQA model based on dynamic word vector. *Peerj Comput. Sci.* **2021**, *7*, e353. [CrossRef] [PubMed]
45. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing; Association for Computing Machinery, New York, NY, USA, 3–5 May 1996; pp. 212–219.
46. Benedetti, M.; Garcia-Pintos, D.; Perdomo, O.; Leyton-Ortega, V.; Nam, Y.; Perdomo-Ortiz, A. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Inf.* **2019**, *5*, 45. [CrossRef]
47. Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 2309. [CrossRef]
48. Zhang, Y.J.; Tang, J. Analysis and assessment of network security situation based on cloud model. *Int. J. Theor. Phys.* **2014**, *36*, 63–67.
49. Tian, H.; Qin, Y.; Niu, Z.; Wang, L.; Ge, S. Summer Maize Mapping by Compositing Time Series Sentinel-1A Imagery Based on Crop Growth Cycles. *J. Indian Soc. Remote. Sens.* **2021**, *49*, 2863–2874. [CrossRef]
50. Qin, Y. Early-Season Mapping of Winter Crops Using Sentinel-2 Optical Imagery. *Remote. Sens.* **2021**, *13*, 3822.
51. Zheng, W.; Xun, Y.; Wu, X.; Deng, Z.; Chen, X.; Sui, Y. A Comparative Study of Class Rebalancing Methods for Security Bug Report Classification. *IEEE Trans. Reliab.* **2021**, *70*, 4. [CrossRef]
52. Wu, X.; Zheng, W.; Xia, X.; Lo, D. Data quality matters: A case study on data label correctness for security bug report prediction. *IEEE Trans. Softw. Eng.* **2021**, *48*, 2541–2556 [CrossRef]
53. Aghayar, K.; Rezaei Fard, E. Noisy Quantum Mixed State Pattern Classification Based on the Grover's Search Algorithm. *Int. J. Nanotechnol. Appl.* **2021**, *15*, 171–180.

Quantum Spatial Search with Electric Potential: Long-Time Dynamics and Robustness to Noise

Thibault Fredon ^{1,*}, Julien Zylberman ², Pablo Arnault ¹ and Fabrice Debbasch ^{2,*}

¹ Université Paris-Saclay, CNRS, ENS Paris-Saclay, INRIA, Laboratoire Méthodes Formelles, 91190 Gif-sur-Yvette, France

² Sorbonne Université, Observatoire de Paris, Université PSL, CNRS, LERMA, 75005 Paris, France

* Correspondence: fredonthibault@gmail.com (T.F.); fabrice.debbasch@gmail.com (F.D.)

Abstract: We present various results on the scheme introduced in a previous work, which is a quantum spatial-search algorithm on a two-dimensional (2D) square spatial grid, realized with a 2D Dirac discrete-time quantum walk (DQW) coupled to a Coulomb electric field centered on the node to be found. In such a walk, the electric term acts as the oracle of the algorithm, and the free walk (i.e., without electric term) acts as the “diffusion” part, as it is called in Grover’s algorithm. The results are the following. First, we run long time simulations of this electric Dirac DQW, and observe that there is a second localization peak around the node marked by the oracle, reached in a time $O(\sqrt{N})$, where N is the number of nodes of the 2D grid, with a localization probability scaling as $O(1/\ln N)$. This matches the state-of-the-art 2D-DQW search algorithms before amplitude amplification. We then study the effect of adding noise on the Coulomb potential, and observe that the walk, especially the second localization peak, is highly robust to spatial noise, more modestly robust to spatiotemporal noise, and that the first localization peak is even highly robust to spatiotemporal noise.

Keywords: quantum algorithms; quantum walks; quantum spatial search; noise

Citation: Fredon, T.; Zylberman, J.; Arnault, P.; Debbasch, F. Quantum Spatial Search with Electric Potential: Long-Time Dynamics and Robustness to Noise. *Entropy* **2022**, *24*, 1778. <https://doi.org/10.3390/e24121778>

Academic Editors: Giuliano Benenti and Brian R. La Cour

Received: 24 October 2022

Accepted: 30 November 2022

Published: 5 December 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Discrete-time quantum walks (DQWs) correspond to the one-particle sector of quantum cellular automata [1,2]. They can simulate numerous physical systems, ranging from particles in arbitrary Yang–Mills gauge fields [3] and massless Dirac fermions near black holes [4], to charged quantum fluids [5], see also Refs. [6–16] for other physics-oriented applications.

Moreover, DQWs can be seen as quantum analogs of classical random walks (CRWs) [17], and can be used to build spatial-search algorithms that outperform [18] those built with CRWs. Continuous-time quantum walks can also be used for such a purpose [19]. In three spatial dimensions, DQW-based algorithms [18,19] find the location of a marked node with a constant localization probability (We call “localization probability” the probability to be at the marked node, or nodes if there are several of them.) after $O(\sqrt{N})$ time steps, with N , the number of nodes of the three-dimensional grid, and this is exactly the bound reached by Grover’s algorithm [20–25]. However, no two-dimensional (2D) QW proposed so far reaches Grover’s lower bound.

The state-of-the-art result using a 2D DQW was obtained by Tulsi in Ref. [26]: Tulsi’s algorithm finds a marked node with a localization probability scaling as $O(1/\ln N)$ in $O(\sqrt{N})$ time steps, where N is the total number of nodes. To reach a probability independent of N , several amplitude amplification time steps have to be performed after the quantum-walk part. These extra time steps are Grover’s algorithm time steps, see Ref. [27]. Taking the amplitude amplification into account, Tulsi’s algorithm reaches an $O(1)$ localization probability after $O(\sqrt{N \ln N})$ time steps.

Other schemes of 2D DQW for spatial search have followed, such as the one by Roget et al. in Ref. [28], where the 2D DQW simulates a massless Dirac fermion on a grid

with defects. This scheme is inspired by physics, and it reaches Tulsi's bound using a coin of dimension 2 instead of 4. Recently, Zylberman and Debbasch introduced in Ref. [29] a new DQW scheme for 2D quantum spatial search. This scheme implements quantum search by simulating the dynamics of a massless Dirac fermion in a Coulomb electric field centered on the nodes to be found. We call this DQW "electric Dirac DQW" (We call "Dirac DQW" a DQW that has as a continuum limit the Dirac equation. Throughout this paper, the terminology "electric Dirac DQW" will always refer to a Dirac DQW coupled to a Coulomb electric potential, unless otherwise stated. In the literature, other types of electric potentials have been considered. The reason why we do not specify the term "Coulomb" in the present denomination "electric Dirac DQW" is because the idea we want to convey is that the marked node is encoded in the shape of the electric potential, but the precise form of the electric potential, e.g., here, the fact that it is a Coulomb potential, may not be that relevant.). In this walk, the oracle is a position-dependent *phase* .

This oracle is diagonal in the position basis and can be efficiently implemented on n qubits up to an error ϵ using $O(\frac{1}{\epsilon})$ primitive quantum gates [30]. This total number of quantum gates is independent of n and makes possible the implementation of the oracle on current Noisy Intermediate Scale Quantum (NISQ) devices and on future universal quantum computers.

Note also that the algorithm proposed in Ref. [29] actually constitutes a paradigm change in the construction of search algorithms, because it is based on the physically motivated idea that the position of the marked node can be encoded in the shape of an artificial force field, which acts on the quantum walker.

One of the main results of Zylberman and Debbasch's paper [29] is a localization probability, which displays a maximum in $O_{N \rightarrow \infty}(1)$ time steps, the localization probability scaling as $O(1/N)$ (a detailed analysis is presented in Section 3). Since it focuses on this result, Ref. [29] does not offer an analysis of the walk at times much larger than $O(1)$. Moreover, practical implementation not only on current NISQ devices, but also on future, circuit-based quantum computers, can only be envisaged if the algorithm is robust to noise (see, for example, Refs. [5,31–37]); this question is also not addressed in Ref. [29].

The aim of this article is to explore both aspects: long-time dynamics and robustness to noise. The main results are the following. First, the electric Dirac DQW exhibits a second localization peak at a time scaling as $O(\sqrt{N})$ with localization probability scaling as $O(1/\ln N)$. This makes this walk state-of-the-art for 2D DQW spatial search before amplitude amplification. Moreover, this second localization peak is highly robust to spatial noise. Finally, the peak is also robust to spatiotemporal noise, but not as much as it is to time-independent spatial noise.

The article is organized as follows. In Section 2, we offer a review of the electric Dirac DQW presented in Ref. [29]. In Section 3, we study in detail the first two maxima of the localization probability. We show that the first maximum, already analyzed in Ref. [29], is actually present up to $N = 9 \times 10^6 > 10^6 \simeq 2^{20}$ (have in mind that 20 is the current average number of working qubits on most IBM-Q platforms according to <https://quantum-computing.ibm.com/services/resources> accessed on 29 November 2022). We also present evidence for the scaling laws characterizing both the first peak and the second, long-time peak, which reaches Tulsi's state-of-the-art bound. In Section 4, we analyze the resources one needs to implement the quantum spatial search in terms of qubits and primitive quantum operations. In Section 5, we show that the walk, and in particular the second peak, have a good robustness to spatial oracle noise. We also show that the first peak is robust even to spatiotemporal noise. In Section 6, we propose an analysis of the walker's probability distributions. These probability distributions show that the spatial noise does not affect the shape of the peaks significantly. The peaks remain extremely high relative to the background, which shows not only good but high robustness of the peaks to spatial oracle noise. The probability distributions also show that the second peak is sharper than the first one.

2. Basics

2.1. Definition of the 2D Electric Dirac DQW

We consider a 2D square spatial grid with nodes indexed by two integers $(p, q) \in \llbracket 0, M \rrbracket^2$, where $M \in \mathbb{N}$ is the number of nodes along one dimension and $N = M^2$ is the total number of nodes. The time is also discrete and indexed by a label $j \in \mathbb{N}$. The walker is defined by its quantum state $|\Psi_j\rangle$ in the Hilbert space $\mathcal{H}_C \otimes \mathcal{H}_p$, where \mathcal{H}_C , called *coin space*, is the two-dimensional Hilbert space, which corresponds to the internal, coin degree of freedom, and \mathcal{H}_p , called *position space*, corresponds to the spatial degrees of freedom. The *wavefunction* of the state will be denoted as $\Psi_{j,p,q} \equiv \left(\psi_{j,p,q}^L \quad \psi_{j,p,q}^R \right)^\top$, where \top denotes the transposition. The discrete-time evolution of the walker is defined by the following one-step evolution equation,

$$\Psi_{j+1,p,q} = (\mathcal{U}\Psi_j)_{p,q}. \tag{1}$$

The one-step evolution operator, also called *walk operator*, \mathcal{U} , is defined by

$$\mathcal{U} := e^{-ie\phi} R(\theta^-) \mathcal{S}_2 R(\theta^+) \mathcal{S}_1, \tag{2}$$

where $\mathcal{S}_{1,2}$ are standard *shift operators*,

$$(\mathcal{S}_1 \Psi)_{p,q}^L := \psi_{p+1,q}^L \tag{3a}$$

$$(\mathcal{S}_1 \Psi)_{p,q}^R := \psi_{p-1,q}^R \tag{3b}$$

$$(\mathcal{S}_2 \Psi)_{p,q}^L := \psi_{p,q+1}^L \tag{3c}$$

$$(\mathcal{S}_2 \Psi)_{p,q}^R := \psi_{p,q-1}^R, \tag{3d}$$

$R(\theta)$ is a coin-space rotation, also called *coin operator*, defined by

$$R(\theta) := \begin{bmatrix} \cos \theta & i \sin \theta \\ i \sin \theta & \cos \theta \end{bmatrix}, \tag{4}$$

and

$$\theta^\pm := \pm \frac{\pi}{4} - \frac{\mu}{2}, \tag{5}$$

with μ , some real parameter. A schematic representation of a quantum circuit for \mathcal{U} is proposed in Figure 1. More details about the circuit are given in Section 4. A schematized picture of the walk operator is proposed in Figure 2.

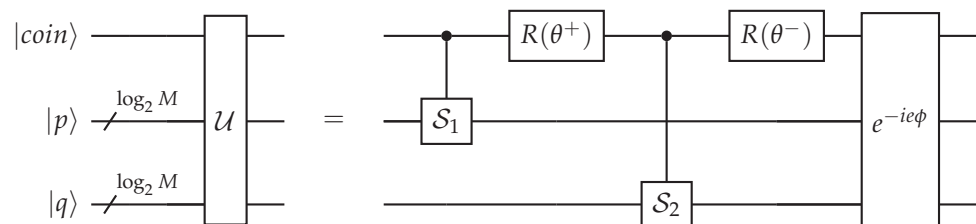


Figure 1. Quantum circuit of a single step operator \mathcal{U} .

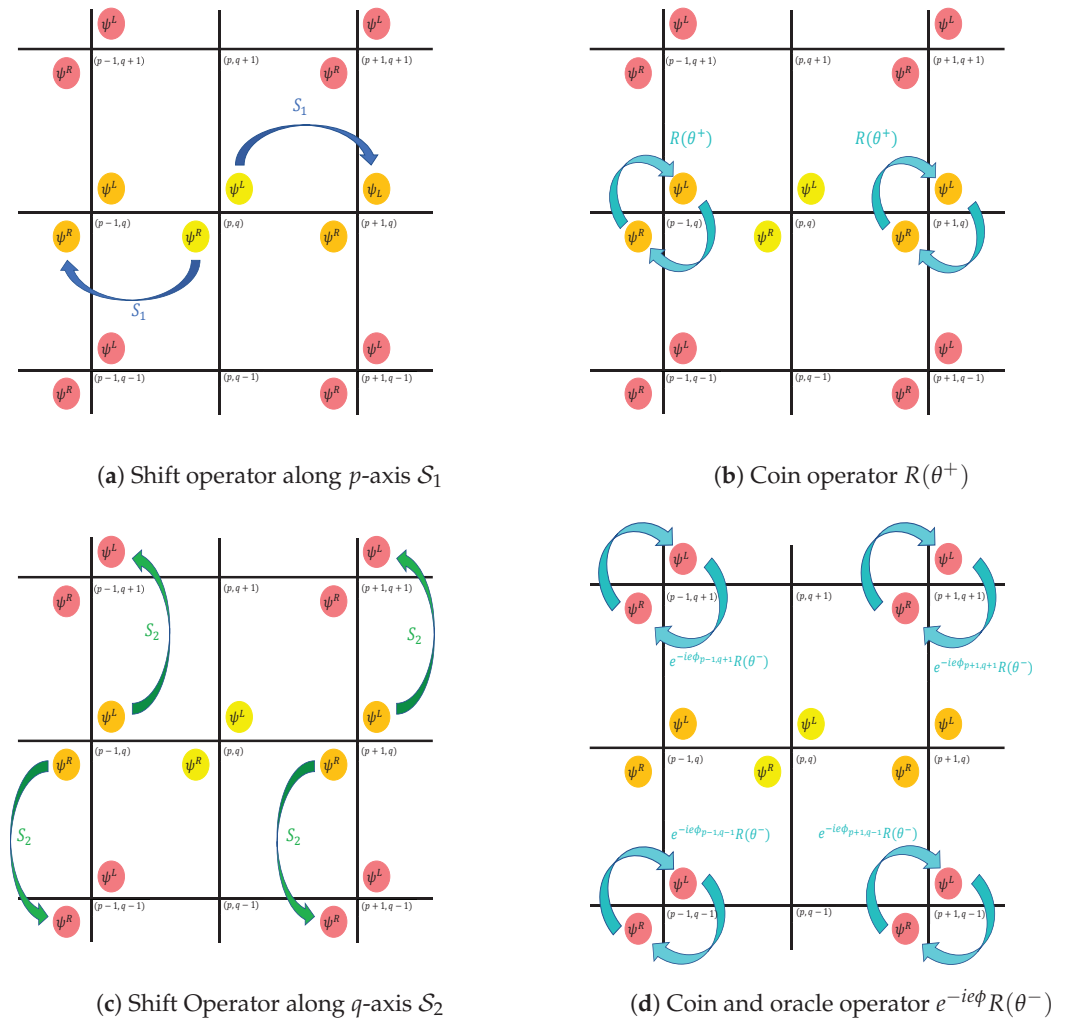


Figure 2. Schematic representation of the quantum walk scheme for one step, starting from position (p, q) . **(a)** First, the S_1 shift operator is applied, shifting the ψ^L component at position (p, q) to position $(p + 1, q)$ and the ψ^R component at position (p, q) to position $(p - 1, q)$. **(b)** Second, the rotation $R(\theta^+)$ is applied at positions $(p \pm 1, q)$, mixing the two components ψ^L and ψ^R (see Equation (4) with angle θ^+). **(c)** Third, the S_2 shift operator is applied, shifting the ψ^L components at positions $(p \pm 1, q)$ to positions $(p \pm 1, q + 1)$ and the ψ^R components at position $(p \pm 1, q)$ to position $(p \pm 1, q - 1)$. **(d)** Finally, the rotation $R(\theta^-)$ and the oracle $e^{-ie\phi}$ is applied. The two components ψ^L and ψ^R are first mixed by the rotation defined Equation (4) with angle θ^- and then multiplied by the position-dependent phase factor defined by the potential ϕ . In this scheme, these operations are illustrated by considering the components $\psi_{p,q}^L$ and $\psi_{p,q}^R$ at a node (p, q) . At the end of one step, the components $\psi_{p,q}^L$ and $\psi_{p,q}^R$ are spread at the nodes $(p + 1, q + 1)$, $(p + 1, q - 1)$, $(p - 1, q + 1)$, and $(p - 1, q - 1)$ in a unitary manner.

The operator $e^{-ie\phi}$ is diagonal in position space, i.e., it acts on Ψ_j as

$$(e^{-ie\phi}\Psi_j)_{p,q} = e^{-ie\phi_{p,q}}\Psi_{j,p,q}, \tag{6}$$

with $\phi : (p, q) \mapsto \phi_{p,q} \in \mathbb{R}$ some sequence of the lattice position, and e , a parameter that we can call the charge of the walker, see why further down. The sequence ϕ can be called the lattice electric potential for at least two reasons: (i) in the continuum limit (see below, Section 2.2), this sequence indeed becomes, mathematically, an electric potential coupled to the walker, who then obeys the Dirac equation, and (ii) beyond the continuum limit, it has been shown that similar 2D DQWs exhibit an exact lattice U(1) gauge invariance [38]

which, in the continuum limit, becomes the standard U(1) gauge invariance of the Dirac equation coupled to an electromagnetic potential.

2.2. Continuum Limit

We introduce a spacetime-lattice spacing ϵ , and coordinates $t_j := \epsilon j$, $x_p := \epsilon p$, and $y_q := \epsilon q$ [39,40]. We assume that $\Psi_{j,p,q}$ coincides with the value taken at point t_j , x_p , and y_q by a function Ψ of the continuous coordinates t , x , and y . We are interested in the dynamics followed by Ψ when $\epsilon \rightarrow 0$. Let us introduce the following continuum quantities,

$$m := \frac{\mu}{\epsilon} \tag{7a}$$

$$V(x_p, y_q) := \frac{\phi_{p,q}}{\epsilon}, \tag{7b}$$

which are, respectively, the mass and electric potential, see why just below.

Expand now Equation (1) in ϵ around $\epsilon = 0$. The walk operator, Equation (2), has been chosen so that (i) the zeroth-order terms give us $\Psi(t, x, y) = \Psi(t, x, y)$, i.e., the terms cancel each other, and (ii) the first-order terms deliver the well-known Dirac equation coupled to an electric potential V . This equation (in natural units where $c = 1$ and $\hbar = 1$) reads:

$$i\partial_t \Psi = \mathcal{H} \Psi, \tag{8}$$

where the Dirac Hamiltonian is

$$\mathcal{H} := \alpha^k (-i\partial_k) + m\alpha^0 + eV, \tag{9}$$

where summation over $k = 1, 2$ is implicitly assumed. The alpha matrices are

$$\alpha^0 := \sigma_x \tag{10a}$$

$$\alpha^1 := \sigma_z \tag{10b}$$

$$\alpha^2 := -\sigma_y, \tag{10c}$$

where the σ s are the Pauli matrices. Thus, this DQW, Equation (1), simulates the (1+2)D Dirac equation coupled to an electric potential, explaining why the ‘‘Dirac DQW’’ is called an electric DQW.

2.3. Coulomb Potential

As shown in Equation (8), the sequence $\phi = \epsilon V$ represents in the continuum limit the electric potential to which the walker is coupled. We choose V to be a Coulomb potential created by a point particle of charge Q at location (Ω_x, Ω_y) on the 2D plane:

$$eV(x, y) := \frac{eQ}{\sqrt{(x - \Omega_x)^2 + (y - \Omega_y)^2}}. \tag{11}$$

For the sake of simplicity, e will be set to -1 . As discussed in Ref. [29], one can take, without loss of generality (i) $(\Omega_x, \Omega_y) = (\frac{M}{2} - \frac{1}{2}, \frac{M}{2} - \frac{1}{2})$, which is called the *center*, and (ii) $\epsilon = 1$. The charge Q is set to 0.9, and $m = \mu = 0$. Notice that the center is not located on a node of the 2D lattice; it is at equal distance of the four nodes, namely, $(\frac{M}{2}, \frac{M}{2})$, $(\frac{M}{2} - 1, \frac{M}{2})$, $(\frac{M}{2}, \frac{M}{2} - 1)$, and $(\frac{M}{2} - 1, \frac{M}{2} - 1)$. With this choice of potential, the walk can be referred to as a ‘‘Coulomb walk’’.

2.4. Definition of the Spatial-Search Problem

The spatial-search problem is defined as follows. Consider at time $j = 0$ a fully delocalized walker on the grid, i.e., $\forall (p, q, a) \in \llbracket 0, M \rrbracket^2 \times \{L, R\}$, $\psi_{0,p,q}^a := \frac{1}{M\sqrt{2}}$. The

problem addressed by the Coulomb walk with this initial condition is: can the walker localize on the nodes where $\phi_{p,q}$ is at its extremum, that is, the four nodes around the center $(\Omega_x, \Omega_y) = (\frac{M}{2} - \frac{1}{2}, \frac{M}{2} - \frac{1}{2})$?

The first observable will be the probability of being on these nodes as a function of time and of the number of grid nodes:

$$P_j(N) := \sum_{(p',q') \in \{\pm\frac{1}{2}\}^2} \left\| \Psi_{j, \Omega_x+p', \Omega_y+q'}(N) \right\|^2, \tag{12}$$

which we call *localization probability*. It has been shown in Ref. [29] that the localization probability admits a first maximum at time $j_1(N) = 82$, independent of N . We now define long times as times t_j with j much larger than 82. The long-time behavior is studied below in Section 3.

We consider the *probability distribution* over space as second observable,

$$d_{j,p,q}(N) := \left\| \Psi_{j,p,q}(N) \right\|^2, \tag{13}$$

which is studied in Section 6.

The fully delocalized initial condition is common in spatial-search problems because of Grover’s algorithm [20]. Moreover, this initial condition can easily be implemented on a quantum circuit as a tower of Hadamard gates. Other initial superpositions for the coin part were considered in Ref. [29]. Now, the fully delocalized initial condition forces us to pay attention to boundary conditions. In our work, we choose periodic boundary conditions. From a computer science point of view, one can expect from a database to have a list of addresses, which are on a graph whose ends are connected, corresponding exactly to periodic boundary conditions.

3. Noiseless Case: Long Times

In Ref. [29], it is shown that for a ‘small’ grid (up to $N = 2.5 \times 10^5$), the first maximum occurs at $j_1(N) = 82 = O_{N \rightarrow \infty}(1)$ with a localization probability $P_{j_1(N)}(N)$ scaling as $O(1/N)$. According to Figure 3, the result $j_1(N) = O(1)$ actually holds up to $N = 9 \times 10^6 \simeq 10^7$. The left panel of Figure 4 shows that $P_{j_1(N)}(N) = O(1/N)$ is valid up to $N = 900 \times 900 \simeq 10^6$. Now, $P_j(N)$ with fixed N presents several other maxima as j varies, and Figure 4 shows in particular that there is a prominent second maximum. This second maximum occurs at a time $j_2(N)$ which, according to Figure 3 and to the right panel of Figure 4 scales as $O(\sqrt{N})$. The right panel of Figure 4 also shows the localization probability $P_{j_2(N)}(N) = O(1/\ln N)$. This result matches the state-of-the-art result in 2D DQW search algorithms before amplitude amplification [26].

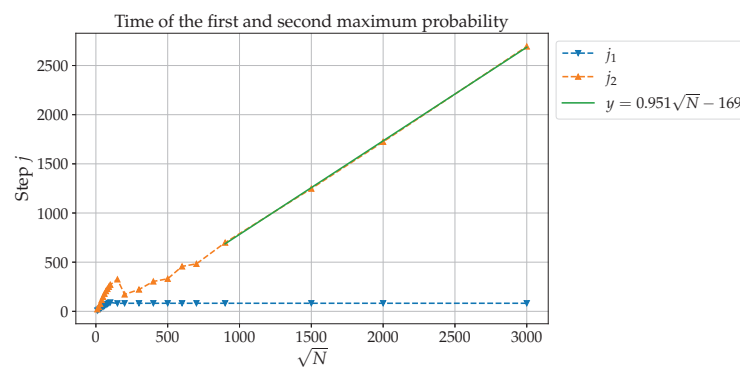


Figure 3. Times j_1 (blue) and j_2 (orange) at which the localization probability $P_j(N)$ reaches a maximum, plotted as a function of \sqrt{N} for $m = 0, e = -1$, and $Q = 0.9$.

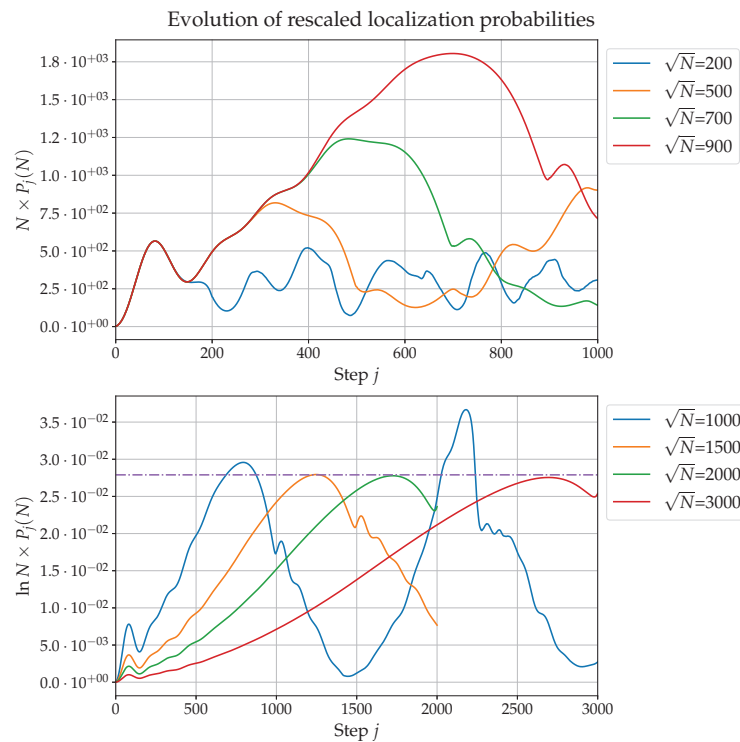


Figure 4. Rescaled localization probability $P_j(N)$ for $m = 0$, $e = -1$, and $Q = 0.9$. Left panel: $P_j(N) \times N$ as a function of j , for several values of N . Right panel: $P_j(N) \times \ln N$ as a function of j , for different values of N .

4. Resource Analysis

Since the evolution operator of the Coulomb walk is built out of two 1D shift operators, one for each spatial directions, the Coulomb walk only requires a 2-dimensional coin space. On the contrary, Tulsi’s walk (see Ref. [26]) uses a 2D shift operator, which requires a 4-dimensional Hilbert space for the coin, so encoding this walk requires one more qubit than encoding the walk studied in the present article. Also note that Tulsi’s algorithm also uses an ancilla qubit to allow a part of the probability amplitude to remain on the same site after one evolution step (technically, Tulsi’s walk uses a controlled shift operator and a controlled coin operator with respect to the ancilla). Thus, in total, Tulsi’s algorithm needs two more qubits than the Coulomb walk to perform a quantum spatial search on a database of the same size. Roget et al.’s walk, presented in Ref. [28], is a DQW—as is the Coulomb walk. It also uses two 1D shift operators and dispenses with the ancilla qubit. The difference with the Coulomb walk lies in the choice of oracle. The Coulomb walk uses an artificial electric field as oracle, while Roget et al.’s walk views the node to be found as a defect and therefore replaces on the defect the rotation $R(\theta)$ of Equation (4) by the identity operator.

A scheme implementing efficiently (up to a given precision ϵ) position-dependent diagonal unitaries similar to the electric potential oracle in Equation (6) can be found in J. Welsh et al. (Ref. [30]). The total number n of one-qubit and two-qubit quantum operations used in this scheme scales as $O(\frac{1}{\epsilon})$ and is actually independent of n . However, the implementation of the shift operators $S_{1,2}$ (see Equation (3)) requires a number of primitive quantum operations, which does depend on n and scales as $O(n^2)$ because implementing shift operators requires performing Quantum Fourier Transforms (QFTs) [41]. Note that each coin operation $R(\theta^-)$ and $R(\theta^+)$ in Equation (2) can be implemented as only one single quantum gate on the coin qubit.

5. Oracle Noise

Today, one of the main goals in quantum computing is having fault-tolerant algorithms, which can be implemented on NISQ devices [42–44].

In the scheme developed by Welch et al. in Ref. [30], the final quantum circuit of the oracle is composed of *CNOT* and R_Z . The rotation angles are only implementable up to a finite accuracy due to hardware limitations. This generates fluctuations in the potential ϕ and we model these fluctuations by a white noise. More precisely, we replace $\phi_{p,q}$ by $\phi_{j,p,q}^B = \phi_{p,q} + B_{j,p,q}$, where B is a white noise in all its variables. To make things as simple as possible, given a point (j, p, q) , $B_{j,p,q}$ is chosen randomly with uniform distribution in a certain interval $(-B_{\max}, B_{\max})$ independent of (j, p, q) . Noise that depends on time only does not modify the probability distribution. All noises considered in this article will therefore be space-dependent. We will first focus on time-independent, but space-dependent noise, and then switch to both time- and space-dependent noise.

Note that decoherence noise on the free-walk part and on Grover search has already been studied in Refs. [45–51].

The amplitude of the noise is best characterized by the noise-to-signal ratio:

$$r := \frac{B_{\max}}{\max_{p,q} |\phi_{p,q}|} . \tag{14}$$

5.1. Spatial Oracle Noise

In this subsection, all observables are averaged over 50 realizations of the noise. Figure 5 presents results obtained for $N = 200^2$ and $N = 500^2$. When the noise-to-signal ratio r is not too high, say $r \lesssim 0.5$, both peaks still exist and the second one occurs slightly later, with approximately the same time delay with respect to the noiseless situation. The amplitude of the peaks is also affected by the noise. In particular, for large enough N (see the right panel in Figure 5), the amplitude of the first peak decreases while the amplitude of the second peak actually increases. Thus, weak noise favors, and even enhances the second peak, at least for large enough values of N . Increasing the noise-to-signal ratio r erases the first peak and, to a certain extent, also the second one. Note however that, for large enough N , the probability $P_j^r(N)$ still exhibits a (rather flat) maximum in lieu of the second peak. So, in any case, noise favors the second peak. So, all in all, the algorithm studied in this article shows good to great robustness to spatial noise. It is also instructive to investigate this robustness through the probability distribution over space $d_{j,p,q}^r(N)$ and this is done in Section 6 below.

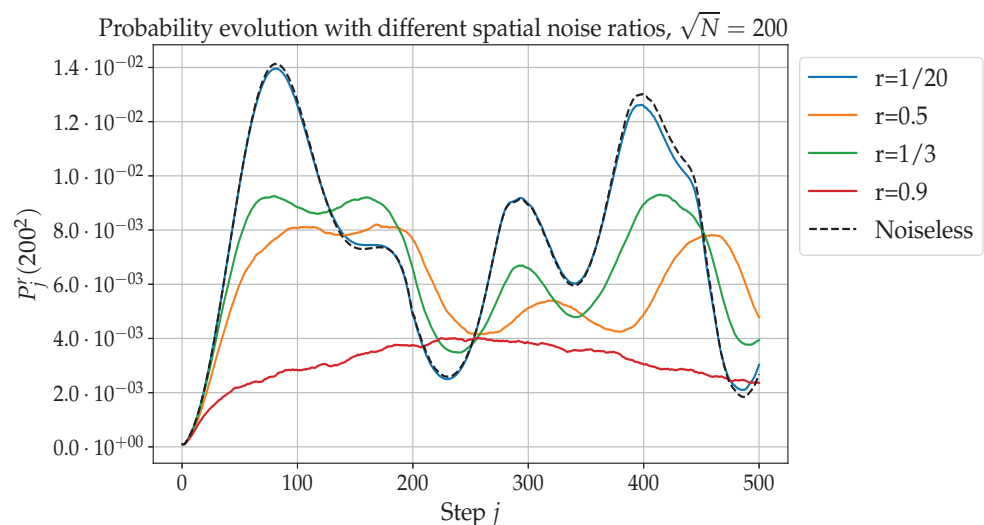


Figure 5. Cont.

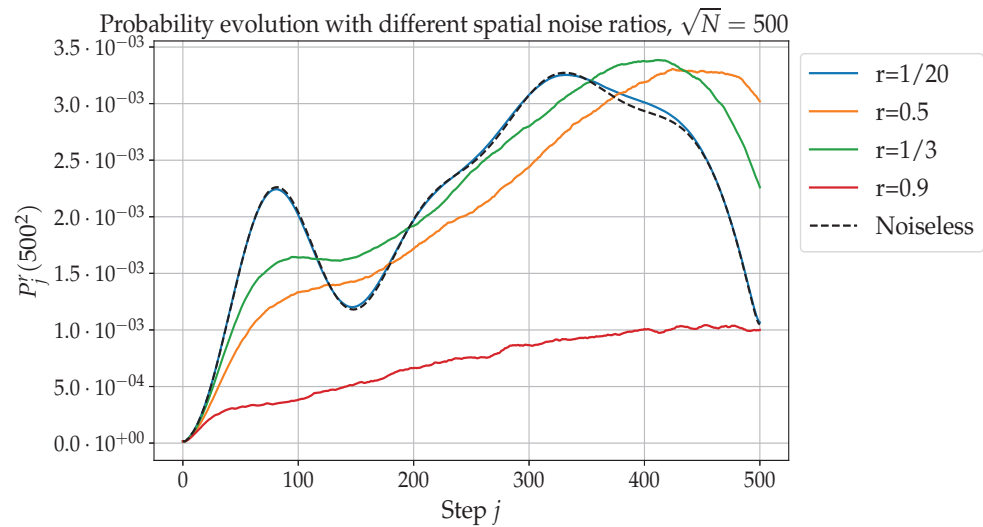


Figure 5. Localization probability $P_j^r(N)$ with spatial noise as a function of j , for different noise-to-signal ratios r , for $m = 0, e = -1, Q = 0.9$, and $\sqrt{N} = 200$ (top) and $\sqrt{N} = 500$ (bottom).

5.2. Spatiotemporal Oracle Noise

In this subsection, all observables are averaged over 10 realizations of the noise. Numerical results are presented in Figure 6. One first observes a global decreases in the localization probability, which gets globally lower with increasing r . However, it also appears that the first peak is less impacted by the noise than the rest of the curve, and especially the second peak. This can be understood in the following way. Since the noise we are considering is white in both space and time, the central limit theorem applies. The walk will therefore exhibit diffusive behavior in the ‘long’-time limit (see, for example, Ref. [49]).

However, the shorter the time, the less important the perturbation induced by the noise on the walk’s behavior. The striking robustness of the first peak, which always occur at $j = 82$, indicates that $j = 82$ is a ‘short’ time, at least for noise-to-signal ratios exceeding 0.5.

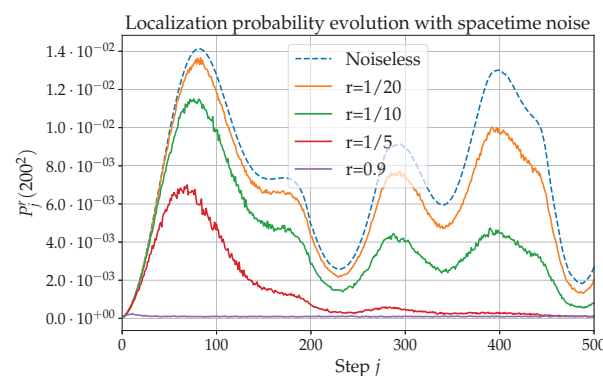


Figure 6. Localization probability $P_j^r(200^2)$ as a function of j for different spatiotemporal noise-to-signal ratios r , with $m = 0, e = -1$, and $Q = 0.9$.

6. Probability Distribution in Space

We now investigate the probability distribution $D_j = \{d_{j,p,q}, \{p, q\} \in \llbracket 0; M \rrbracket^2\}$ of the walk at the localization times j_1 and j_2 corresponding to the first and second peak. The height ratio η between the peak and the background is defined as

$$\eta_j(N) := \frac{d_{j, \frac{M}{2}-1, \frac{M}{2}-1}(N)}{d_{j,1,1}(N)}, \tag{15}$$

where $d_{j, \frac{M}{2}-1, \frac{M}{2}-1}(N)$ is the probability to be on one of the four nodes of interest (where the potential is maximum), and where $d_{j,1,1}(N)$ is the probability to be where the potential is the weakest.

6.1. Noiseless Case

The noiseless case is presented in Figure 7. The probability distributions are sharply peaked on the nodes of interest for both $j = j_1$ (top plots) and $j = j_2$ (bottom plots). For a small grid size (i.e., $\sqrt{N} = 200$), the height ratio is better for the first peak than for the second peak (the precise values are given in the figure caption). For a larger grid size (i.e., $\sqrt{N} = 500$ and $\sqrt{N} = 1000$), the height ratio of the first peak is important, but that of the second peak is substantially larger (see the figure caption).

6.2. Spatial Oracle Noise

Let us now investigate the probability density $D_j^r = \{d_{j,p,q}^r, \{p, q\} \in \llbracket 0, M \rrbracket\}$ in the presence of noise with noise-to-signal ratio r . Figure 8 displays D_j^r (top plots) and $D_j^r - D_j$ (bottom plots) at $j = j_1$ (left plots) and $j = j_2$ (right plots). On the top plots of Figure 8, where $D_j^{1/3}$ for $r = 1/3$ is plotted, one observes that the overall shapes of the peaks, and in particular their widths, are not affected by the noise. The height ratios (given in the caption of Figure 8) are still very large, even in the presence of a substantial amount of noise ($r = 1/3$). This shows *not only good, but high robustness* of the walk to spatial noise. Looking at the bottom plots of Figure 8 one observes that noise makes the first peak lower (two bottom left plots), but makes the second peak (two bottom right plots) higher for a small grid size ($M = 200$) or balanced between the four nodes of interest for a larger grid size ($M = 500$). These observations are of course consistent with the curves of Figure 5.

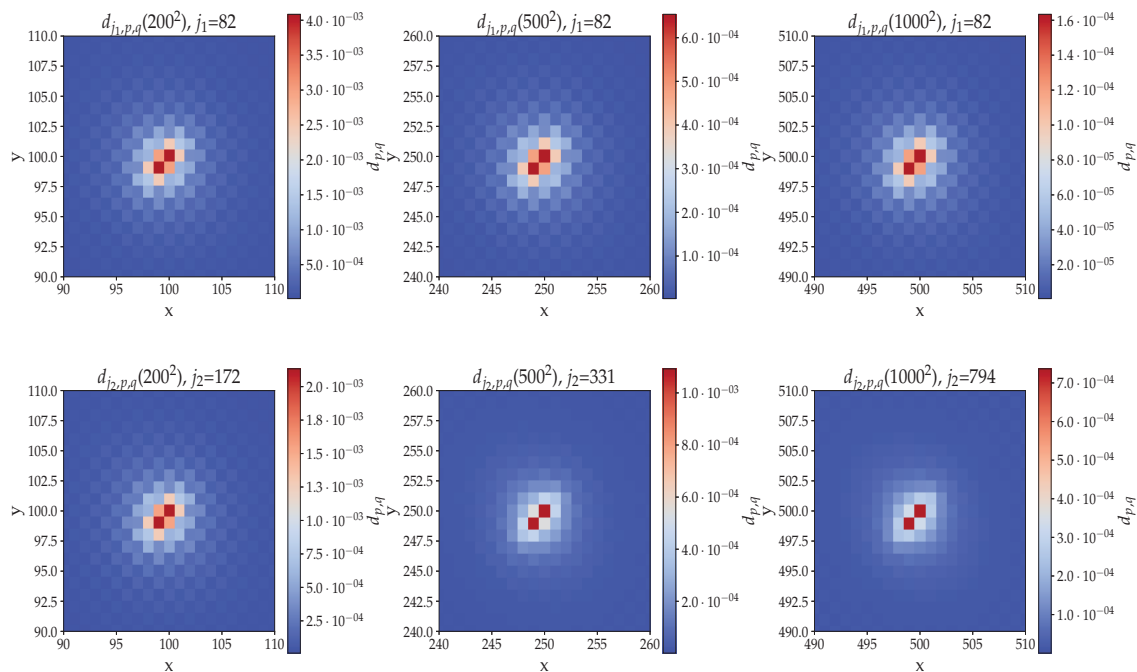


Figure 7. Probability distribution D_j in the noiseless case for $\sqrt{N} = 200, 500, 1000$, $j = j_1$ (top plots), and $j = j_2$ (bottom plots) and $m = 0, e = -1, Q = 0.9$. Height ratios for j_1 : $\eta_{j_1}(200^2) = 160$, $\eta_{j_1}(500^2) = 163$, and $\eta_{j_1}(1000^2) = 163$. Height ratios for j_2 : $\eta_{j_2}(200^2) = 127$, $\eta_{j_2}(500^2) = 242$, and $\eta_{j_2}(1000^2) = 1933$.

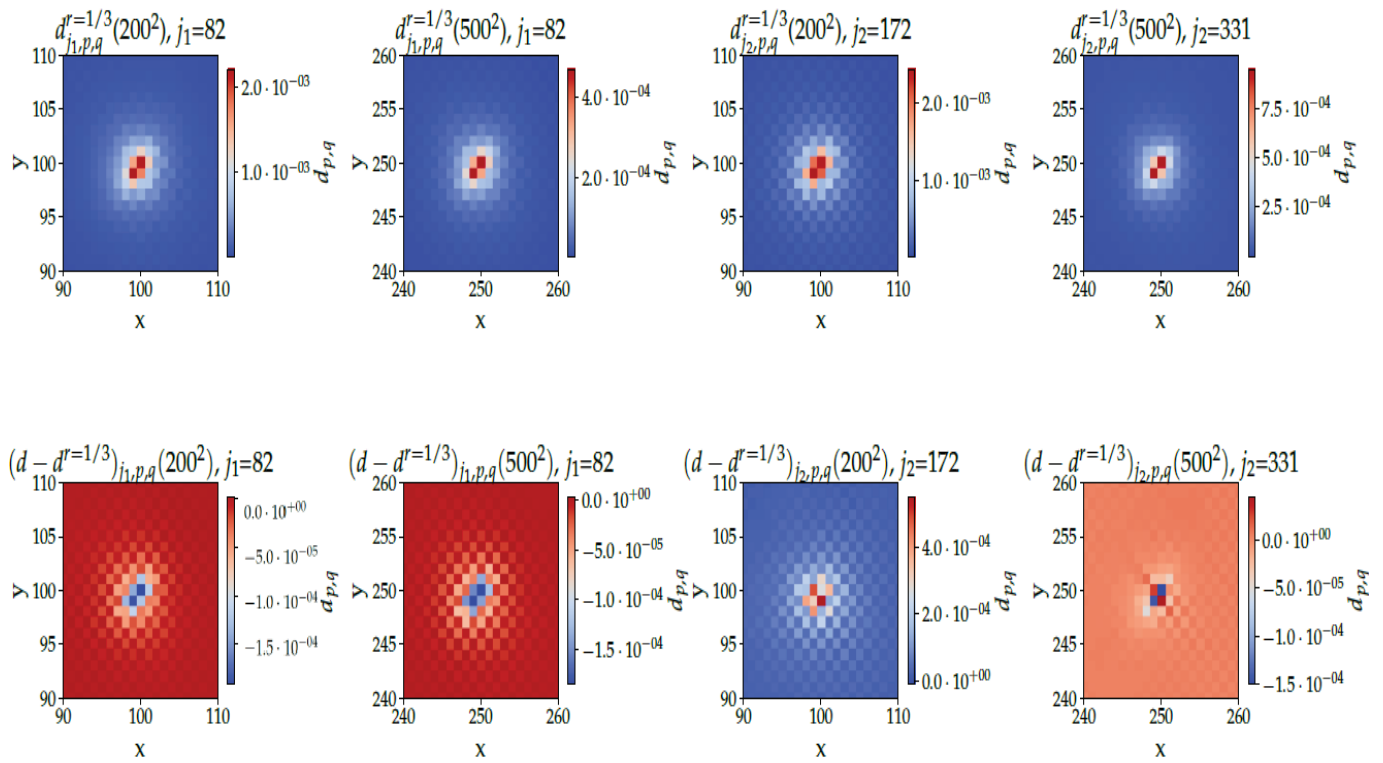


Figure 8. Top plots: Probability distribution $d_{j,p,q}^{r=1/3}(N)$ for $\sqrt{N} = 200$ and 500 , at j_1 (left plots) and j_2 (right plots), averaged over 50 realizations of the spatial noise, with $m = 0, e = -1, Q = 0.9$. Height ratios for j_1 : $\eta_{j_1}^{r=1/3}(200^2) = 87$ and $\eta_{j_1}^{r=1/3}(500^2) = 115$. Height ratios for j_2 : $\eta_{j_2}^{r=1/3}(200^2) = 142$ and $\eta_{j_2}^{r=1/3}(500^2) = 218$. Bottom plots: Difference $d_{j,p,q}^{r=1/3}(N) - d_{j,p,q}(N)$ between the noisy and the noiseless cases, for $m = 0, e = -1, Q = 0.9$.

7. Conclusions and Discussion

In this paper, we have shown that the 2D electric Dirac DQW presented in Ref. [29] has at least two different localization peaks: (i) one at short times ($O_{N \rightarrow \infty}(1)$ with N the number of nodes on the 2D grid), for which the localization probability scales as $O(1/N)$, and (ii) another at a time scaling as $O(\sqrt{N})$ with localization probability in $O(1/\ln N)$, which matches the state-of-the-art result in spatial search with 2D DQWs before amplitude amplification [26,28]. This dynamic was studied numerically up to $N = 9 \times 10^6 \simeq 2^{20}$.

This quantum spatial search also presents a memory advantage by formally requiring two qubits less than Tulsı’s algorithm. In terms of quantum operations, the oracle can be efficiently implemented on a quantum circuit up to an error ϵ using $O(\frac{1}{\epsilon})$ primitive quantum gates, allowing its implementation on current NISQ devices and future fault-tolerant universal quantum computers.

We have also explored the effect of oracle noise by adding a white noise to the electric potential. This white noise can be viewed, for example, as a model of the fluctuations induced by the finite accuracy implementation of the quantum rotations involved in the Oracle quantum circuit [30]. Our results demonstrate that the algorithm is highly robust to oracle noise. The second peak is not only highly robust to, but actually slightly amplified by, spatial noise. The second peak is admittedly less robust to spatiotemporal noise but the first peak turns out highly robust to this type of noise. This study is thus very encouraging for the future implementation of quantum spatial search with electric potential on universal quantum computers and NISQ devices.

Adapting to the present walk, the ancilla technique used in Tulsı’s walk may make the second peak appear sooner and might eventually help the walk reach Grover’s lower bound. Furthermore, studying the evolution of the localization probability under other kinds of noises is assuredly very promising to extend the robustness properties of the

quantum spatial search with electric potential. Finally, extending all results to higher dimensions and to walks using other fields such as oracle will certainly prove interesting.

Author Contributions: Conceptualization, T.F., J.Z., P.A. and F.D.; Methodology, T.F., J.Z., P.A. and F.D.; Software, T.F.; Validation, P.A. and F.D.; Formal analysis, T.F., J.Z., P.A. and F.D.; Investigation, T.F., J.Z., P.A. and F.D.; Resources, T.F., P.A. and F.D.; Data curation, T.F.; Writing—original draft, T.F., J.Z., P.A. and F.D.; Writing—review & editing, T.F., J.Z., P.A. and F.D.; Visualization, F.D.; Supervision, P.A. and F.D.; Project administration, P.A. and F.D.; Funding acquisition, P.A. and F.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available on reasonable demand at fredonthibault@gmail.com.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DQW	Discrete-Time Quantum Walk
QW	Quantum Walk
CQW	Continuous-Time Quantum Walk
NISQ	Noisy Intermediate Scale Quantum

References

1. Arrighi, P. An overview of quantum cellular automata. *Nat. Comput.* **2019**, *18*, 885–899. [CrossRef]
2. Farrelly, T. A review of Quantum Cellular Automata. *Quantum* **2020**, *4*, 368. [CrossRef]
3. Arnault, P.; Di Molfetta, G.; Brachet, M.; Debbasch, F. Quantum walks and non-Abelian discrete gauge theory. *Phys. Rev. A* **2016**, *94*, 012335. [CrossRef]
4. Di Molfetta, G.; Brachet, M.; Debbasch, F. Quantum walks as massless Dirac fermions in curved space-time. *Phys. Rev. A* **2013**, *88*, 042301. [CrossRef]
5. Zylberman, J.; Di Molfetta, G.; Brachet, M.; Loureiro, N.F.; Debbasch, F. Quantum simulations of hydrodynamics via the Madelung transformation. *Phys. Rev. A* **2022**, *106*, 032408. [CrossRef]
6. Berry, S.D.; Wang, J.B. Two-particle quantum walks: Entanglement and graph isomorphism testing. *Phys. Rev. A* **2011**, *83*, 042317. [CrossRef]
7. Ahlbrecht, A.; Alberti, A.; Meschede, D.; Scholz, V.B.; Werner, A.H.; Werner, R.F. Molecular binding in interacting quantum walks. *New J. Phys.* **2012**, *14*, 073050. [CrossRef]
8. Shikano, Y.; Wada, T.; Horikawa, J. Discrete-time quantum walk with feed-forward quantum coin. *Sci. Rep.-UK* **2014**, *4*, 4427. [CrossRef]
9. Bisio, A.; D’Ariano, G.M.; Perinotti, P.; Tosini, A. Weyl, Dirac and Maxwell Quantum Cellular Automata. *Found. Phys.* **2015**, *45*, 1203–1221. [CrossRef]
10. Di Molfetta, G.; Pérez, A. Quantum walks as simulators of neutrino oscillations in a vacuum and matter. *New J. Phys.* **2016**, *18*, 103038. [CrossRef]
11. Bisio, A.; D’Ariano, G.M.; Perinotti, P. Quantum cellular automaton theory of light. *Ann. Phys.-NY* **2016**, *368*, 177–190. [CrossRef]
12. Rakovszky, T.; Asbóth, J.K.; Alberti, A. Detecting topological invariants in chiral symmetric insulators via losses. *Phys. Rev. B* **2017**, *95*, 201407. [CrossRef]
13. Márquez-Martín, I.; Arnault, P.; Di Molfetta, G.; Pérez, A. Electromagnetic lattice gauge invariance in two-dimensional discrete-time quantum walks. *Phys. Rev. A* **2018**, *98*, 032333. [CrossRef]
14. Arrighi, P.; Di Molfetta, G.; Márquez-Martín, I.; Pérez, A. From curved spacetime to spacetime-dependent local unitaries over the honeycomb and triangular quantum walks. *Sci. Rep.-UK* **2019**, *9*, 10904. [CrossRef]
15. Jay, G.; Debbasch, F.; Wang, J.B. Dirac quantum walks on triangular and honeycomb lattices. *Phys. Rev. A* **2019**, *99*, 032113. [CrossRef]
16. Anglés-Castillo, A.; Pérez, A. A quantum walk simulation of extra dimensions with warped geometry. *Sci. Rep.-UK* **2022**, *12*, 1926. [CrossRef]
17. Kempe, J. Quantum random walks: An introductory overview. *Contemp. Phys.* **2003**, *44*, 307–327. [CrossRef]
18. Ambainis, A.; Kempe, J.; Rivosh, A. *Coins Make Quantum Walks Faster*; SODA ’05; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2005; pp. 1099–1108.
19. Childs, A.M.; Goldstone, J. Spatial search by quantum walk. *Phys. Rev. A* **2004**, *70*, 022314. [CrossRef]

20. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the ACM symposium on Theory of Computing—STOC '96, Philadelphia, PA, USA, 22–24 May 1996.
21. Lavor, C.; Manssur, L.R.U.; Portugal, R. Grover's Algorithm: Quantum Database Search. *arXiv* **2003**, arXiv:quant-ph/0301079.
22. Abal, G.; Donangelo, R.; Marquezino, F.L.; Portugal, R. Spatial search on a honeycomb network. *Math. Struct. Comp. Sci.* **2010**, *20*, 999–1009. [CrossRef]
23. Inui, N.; Konishi, Y.; Konno, N. Localization of two-dimensional quantum walks. *Phys. Rev. A* **2004**, *69*, 052323. [CrossRef]
24. Konno, N.; Obata, N.; Segawa, E. Localization of the Grover Walks on Spidernets and Free Meixner Laws. *Commun. Math. Phys.* **2013**, *322*, 667–695. [CrossRef]
25. Bezerra, G.A.; Lugão, P.H.G.; Portugal, R. Quantum-walk-based search algorithms with multiple marked vertices. *Phys. Rev. A* **2021**, *103*, 062202. [CrossRef]
26. Tulse, A. Faster quantum-walk algorithm for the two-dimensional spatial search. *Phys. Rev. A* **2008**, *78*, 012310. [CrossRef]
27. Brassard, G.; Høyer, P.; Mosca, M.; Tapp, A. Quantum amplitude amplification and estimation. *arXiv* **2002**, arXiv:quant-ph/0005055.
28. Roget, M.; Guillet, S.; Arrighi, P.; Di Molfetta, G. Grover Search as a Naturally Occurring Phenomenon. *Phys. Rev. Lett.* **2020**, *124*, 180501. [CrossRef]
29. Zylberman, J.; Debbasch, F. Dirac Spatial Search with Electric Fields. *Entropy* **2021**, *23*, 1441. [CrossRef]
30. Welch, J.; Greenbaum, D.; Mostame, S.; Aspuru-Guzik, A. Efficient quantum circuits for diagonal unitaries without ancillas. *New J. Phys.* **2014**, *16*, 033040. [CrossRef]
31. Nielsen, M.A.; Chuang, I.; Grover, L.K. Quantum Computation and Quantum Information. *Am. J. Phys.* **2002**, *70*, 558–559. [CrossRef]
32. Scherer, A.; Valiron, B.; Mau, S.C.; Alexander, S.; van den Berg, E.; Chapuran, T.E. Concrete resource analysis of the quantum linear-system algorithm used to compute the electromagnetic scattering cross section of a 2D target. *Quantum Inf. Process.* **2017**, *16*, 60. [CrossRef]
33. Portugal, R. *Quantum Walks and Search Algorithms*; Springer International Publishing: New York, NY, USA, 2018.
34. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
35. Zhang, K.; Rao, P.; Yu, K.; Lim, H.; Korepin, V. Implementation of efficient quantum search algorithms on NISQ computers. *Quantum Inf. Process.* **2021**, *20*, 233. [CrossRef]
36. Douglas, B.L.; Wang, J.B. Efficient quantum circuit implementation of quantum walks. *Phys. Rev. A* **2009**, *79*, 052335. [CrossRef]
37. Loke, T.; Wang, J.B. Efficient quantum circuits for continuous-time quantum walks on composite graphs. *J. Phys. A-Math. Theor.* **2017**, *50*, 055303. [CrossRef]
38. Arnault, P.; Debbasch, F. Quantum walks and discrete gauge theories. *Phys. Rev. A* **2016**, *93*, 052301. [CrossRef]
39. Shikano, Y. From Discrete Time Quantum Walk to Continuous Time Quantum Walk in Limit Distribution. *J. Comput. Nanos.* **2013**, *10*, 1558–1570. [CrossRef]
40. Arrighi, P.; Nesme, V.; Forets, M. The Dirac equation as a quantum walk: higher dimensions, observational convergence. *J. Phys. A-Math. Gen.* **2014**, *47*, 465302. [CrossRef]
41. Shakeel, A. Efficient and scalable quantum walk algorithms via the quantum Fourier transform. *Quantum Inf. Process.* **2020**, *19*, 323. [CrossRef]
42. Devitt, S.J.; Munro, W.J.; Nemoto, K. Quantum error correction for beginners. *Rep. Prog. Phys.* **2013**, *76*, 076001. [CrossRef]
43. Harper, R.; Flammia, S.T. Fault-Tolerant Logical Gates in the IBM Quantum Experience. *Phys. Rev. Lett.* **2019**, *122*, 080504. [CrossRef]
44. Roffe, J. Quantum error correction: an introductory guide. *Contemp. Phys.* **2019**, *60*, 226–245. [CrossRef]
45. Chandrashekar, C.M.; Srikanth, R.; Banerjee, S. Symmetries and noise in quantum walk. *Phys. Rev. A* **2007**, *76*, 022316. [CrossRef]
46. Banerjee, S.; Srikanth, R.; Chandrashekar, C.M.; Rungta, P. Symmetry-noise interplay in a quantum walk on an n-cycle. *Phys. Rev. A* **2008**, *78*, 052316. [CrossRef]
47. Alberti, A.; Alt, W.; Werner, R.; Meschede, D. Decoherence models for discrete-time quantum walks and their application to neutral atom experiments. *New J. Phys.* **2014**, *16*, 123052. [CrossRef]
48. Oliveira, A.C.; Portugal, R.; Donangelo, R. Decoherence in two-dimensional quantum walks. *Phys. Rev. A* **2006**, *74*, 012312. [CrossRef]
49. Di Molfetta, G.; Debbasch, F. Discrete-time quantum walks in random artificial gauge fields. *Quantum Stud. Math. Found.* **2016**, *3*, 293–311. [CrossRef]
50. Morley, J.G.; Chancellor, N.; Bose, S.; Kendon, V. Quantum search with hybrid adiabatic quantum-walk algorithms and realistic noise. *Phys. Rev. A* **2019**, *99*, 022339. [CrossRef]
51. Peng, Y.F.; Wang, W.; Yi, X.X. Discrete-time quantum walk with time-correlated noise. *Phys. Rev. A* **2021**, *103*, 032205. [CrossRef]

Article

Multi-Qubit Bose–Einstein Condensate Trap for Atomic Boson Sampling

Sergey Tarasov ¹, William Shannon ², Vladimir Kocharovsky ¹ and Vitaly Kocharovsky ^{2,*}

¹ Institute of Applied Physics, Russian Academy of Sciences, Nizhny Novgorod 603950, Russia

² Department of Physics and Astronomy and Institute for Quantum Science and Engineering, Texas A&M University, College Station, TX 77843-4242, USA

* Correspondence: vkochar@physics.tamu.edu

Abstract: We propose a multi-qubit Bose–Einstein-condensate (BEC) trap as a platform for studies of quantum statistical phenomena in many-body interacting systems. In particular, it could facilitate testing atomic boson sampling of the excited-state occupations and its quantum advantage over classical computing in a full, controllable and clear way. Contrary to a linear interferometer enabling Gaussian boson sampling of non-interacting non-equilibrium photons, the BEC trap platform pertains to an interacting equilibrium many-body system of atoms. We discuss a basic model and the main features of such a multi-qubit BEC trap.

Keywords: Bose–Einstein condensation; Gaussian boson sampling; quantum advantage; NP-hard problem

Citation: Tarasov, S.; Shannon, W.; Kocharovsky, V.; Kocharovsky, V. Multi-Qubit Bose–Einstein Condensate Trap for Atomic Boson Sampling. *Entropy* **2022**, *24*, 1771. <https://doi.org/10.3390/e24121771>

Academic Editors: Christos Volos, Karthikeyan Rajagopal, Sajad Jafari, Jacques Kengne and Jesus M. Munoz-Pacheco

Received: 29 October 2022
Accepted: 29 November 2022
Published: 3 December 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction to Quantum Statistical Physics of Atomic Boson Sampling in a BEC Trap

1.1. The Essence of the Problem

Recently, a stationary stochastic process of many-body fluctuations of the excited-atom occupations in a trapped Bose–Einstein-condensed gas has been suggested for quantum simulation of the $\#P$ -hard problem of boson sampling [1]. Such an atomic boson sampling, based on the Bose–Einstein-condensate (BEC) platform, is an alternative to a well-known photonic boson sampling based on the linear interferometer platform [2–25]. It has the potential to demonstrate quantum advantage [26–30] of the many-body interacting systems over classical computers. For a full and clear demonstration of a $\#P$ -hardness of computing atom-excitation sampling, a condensate should be nonuniformly spread over an entire BEC trap and provide, via an interparticle interaction, multimode Bogoliubov coupling between a large number of excited atom states. Moreover, all of the above parameters of the many-body system should be controllable in a wide range to ensure sufficient variability of the observed joint occupation statistics of the excited states or coarse-grained groups of excited states. So, there is an open problem of designing BEC traps most suitable for experimental studies of various phenomena associated with atomic boson sampling.

The present paper is devoted to this problem: We discuss a basic model of a potential design of the multi-qubit BEC trap that could provide the required conditions and be particularly suitable for atomic-boson-sampling experiments. It is inspired by an analogy with multi-qubit or multi-qudit systems [31,32] and could look like a system of a finite number, Q , of single-qubit or -qudit cells shown in Figure 1 in a two-dimensional (2D) case.

Remarkably, a direct measurement of fluctuations in a total occupation of the noncondensate in cold dilute gases has already been achieved [33,34]. Splitting the noncondensate into some parts associated with the groups of excited states and measuring atom-number fluctuations in the occupations of those parts is the next important step in the many-body statistical physics toward testing quantum advantage. It is beyond the bulk of previous studies of the BEC phenomena, which is devoted to the mean properties of the condensate

and quasiparticles, and it is also beyond the previous studies of fluctuations in the total occupation of the condensate (see, e.g., [34–42]). The atom-number fluctuations are especially important for the applications related to quantum information science and matter-wave interferometers [43], including Ramsey [44,45] and Mach–Zehnder [46] on-chip interferometers. In the literature, there are also other interesting discussions of the atom-number fluctuations associated with a subvolume of a BEC trap [47,48], BEC collapse [49], and squeezed states [44].

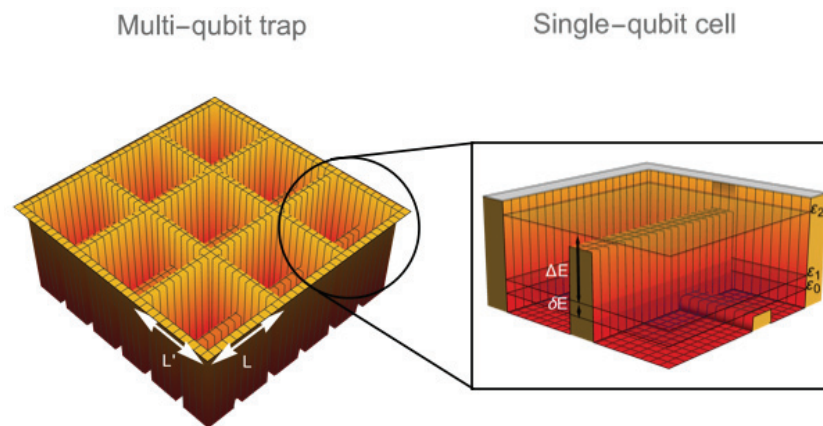


Figure 1. Two-dimensional (2D) model of a BEC trap made up of Q single-qubit (or single-qudit) cells of size $L \times L'$ each contributing with two (or more) lower energy levels to the lower-energy miniband of the multi-qubit (or multi-qudit) trap. This miniband is separated from the higher energy levels by an energy gap ΔE much larger than the lower-energy splitting δE . For presentation purposes, an inhomogeneous underlying (background) potential, designed for controlling the condensate profile and Bogoliubov couplings, as well as the high potential walls at the outer borders of the trap are not shown.

1.2. What Is the Atomic Boson Sampling?

In statistical theory, sampling is a selection of events (subsets) from within a sample space of all possible outcomes (or results or sample points) to mimic the characteristics of the probability distribution in a probability space (a probabilistic model). Atomic boson sampling means sampling from the excited-state occupations of identical Bose atoms subject to interparticle scattering (interaction) in a trapped Bose–Einstein condensed gas within a statistical ensemble of a given experimental setup. The atoms forming the condensate are not counted. One can consider the integer occupations $n_k = 0, 1, 2, \dots$ and their joint probability distribution $\rho(\{n_k\})$ for the individual orthonormal excited states $\{\phi_k(\mathbf{r}) | k = 1, 2, \dots\}$, orthogonal to the condensate wave function, or for groups of such excited states. The simultaneous measurement of their occupations has to be completed by multiple detectors via projecting atoms onto preselected subsets (groups) of the excited states. The latter subsets determine the sampling probability distribution in question.

Condensed-matter statistical physics of a mesoscopic system of N atoms confined in a trap is highly nontrivial due to an interaction between massive atoms taking place on a background of the Bose–Einstein condensate formed by the same interacting atoms via spontaneous symmetry breaking at a critical temperature T_c . Quantum many-body fluctuations in this system remain \sharp P-hard for computing [1] even in equilibrium and even within the grand-canonical-ensemble [50] and Bogoliubov–Popov approximations [51–53]. For simplicity’s sake, we adopt these approximations in the present paper and assume that the temperature is well below the critical region of the BEC phase transition, $T \ll T_c$.

The computational \sharp P-hardness of atomic boson sampling is a real property of the interacting BEC gas, not just a feature of the Bogoliubov–Popov approximation. It follows from the exact non-perturbative theory of critical fluctuations in BEC, which is based on the non-polynomial diagram technique [54–56] and also leads to the representation of the joint

probability distribution of the occupations of the bare excited atomic states in terms of the hafnian of a matrix associated with a correlation matrix which is similar to the one given by the stated approximation. In other words, the $\#P$ -hardness is a robust property of atomic boson sampling in the sense that it manifests itself (survives) even in the Bogoliubov–Popov, i.e., the first order with respect to the interaction parameter, mean-field approximation.

Each act of atomic boson sampling occurs by means of a measurement of the occupations of excited states or groups of them. One can employ, say, a simultaneous optical multi-detector imaging. Then, the system of interacting atoms returns back to its equilibrium state and becomes ready for the next act of sampling/measurement. In a sense, the system of atoms resets itself. This is true both if the atoms were reloaded into the trap after being released from the trap in the case of a trap-destruction measurement or if the atoms were not removed from the trap. Thus, the atomic boson sampler is not a quantum simulator of some input signal or some artificial, controlled process. The system of atoms in the BEC trap equipped with the atom-number detectors is just a quantum generator of random strings of atom-excitation occupations based on the natural process of persistent equilibrium fluctuations.

Surprisingly, atomic boson sampling has the potential to demonstrate a quantum advantage that is similar to the one of Gaussian boson sampling of non-interacting photons in a linear interferometer.

1.3. Comparison with Photonic Boson Sampling in a Linear Interferometer

Physics of photonic boson sampling in a linear interferometer, widely studied in the past decade [2–25], looks significantly simpler since photons are non-interacting, massless and enter the interferometer in a given quantum (e.g., Fock or squeezed) state from the external sophisticated on-demand light sources. Importantly, atomic boson sampling from an equilibrium BEC trap does not require external sources of atoms in a prescribed quantum state because the interacting atoms generate squeezing by themselves. Thus, in order to demonstrate the quantum advantage in a linear interferometer, one has to prepare the system of photons in a prescribed nonequilibrium state, whereas a usual thermal state of atoms in the BEC trap is enough in the case of atomic boson sampling. The reason for switching from the original proposal [3] to the Gaussian boson sampling protocol was an absence of the on-demand single-photon sources and availability of the reliable on-demand sources of input photons in the Gaussian/squeezed states based on the process of a parametric down-conversion [4,5,23].

The computational $\#P$ -hardness of the joint excited-state occupation fluctuations in the BEC trap exists by itself and does not require an adjustment or fine tuning of any input, processing, interaction or coupling parameters. On the contrary, Gaussian boson sampling in a linear interferometer, like other usually discussed nonequilibrium quantum processors and simulators, requires a lossless propagation through a system of beam couplers, splitters, and phase shifters as well as external sources of photons in squeezed states. Moreover, the main limiting factor for a demonstration of quantum advantage in photonic sampling is an exponential growth of photon losses on the input–output propagation with an increasing number of optical channels and coupling elements. Such a limiting factor is absent in the case of atomic boson sampling.

1.4. Why Atomic Boson Sampling Is $\#P$ -Hard for Computing?—A Brief Theory

Despite a number of outstanding differences stated above, both atomic and photonic boson samplings belong to the same $\#P$ -hard complexity class. The fact is that in both cases, the sampling (i.e., joint boson-occupation) probability distribution is determined by a hafnian (or, sometimes, a permanent) of matrices built from an appropriate covariance matrix G of the boson creation and annihilation operators. The above fact follows from the analytical calculation of the characteristic function (that is, Fourier transform) of the joint occupation probability distribution by means of the Wigner transform technique and application of the Hafnian Master Theorem that gives an explicit Taylor expansion

of this characteristic function via the aforementioned hafnian [1,57]. It is well known that for a general $n \times n$ matrix, computing the hafnian or permanent is #P-hard [58–60], that is, requires an exponentially large number of operations $\sim n^3 2^{n/2}$ or $\sim n^2 2^n$, respectively, for the best general-purpose algorithms [60–63]. Contrary to a determinant that can be computed in polynomial time $\sim n^3$ via Gaussian elimination, the hafnian and permanent matrix functions are intimately related to the analysis of the #P-hard problems [26,64].

Yet, the computational #P-hardness (which is the basis of quantum advantage) of atomic boson sampling pertains only if the matrices under the hafnian and the covariance matrix G from which they are built are variable/controllable in a wide range by varying the trapping potential, interparticle interaction (via Feshbach resonance [65]), temperature, number of trapped atoms and other parameters of the BEC trap [66]. In other words, those matrices should not be restricted to a narrow subset of matrices for which the hafnian could be computed in polynomial time by some efficient approximation or algorithm, such as a fully polynomial random approximation scheme (FPRAS) [59], a recurrence method [67], etc.

In order to understand how to satisfy this requirement in the design of the BEC trap, we need to know how the covariance matrix G depends on the trap parameters. Fortunately, we have an analytical formula for the G via the matrix R of the Bogoliubov transformation,

$$G = RDR^\dagger + \frac{1}{2}(RR^\dagger - \mathbb{1}); \quad G \equiv \left\langle : \left(\begin{matrix} \hat{\mathbf{a}}^\dagger \\ \hat{\mathbf{a}} \end{matrix} \right) \left(\begin{matrix} \hat{\mathbf{a}}^\dagger \\ \hat{\mathbf{a}} \end{matrix} \right)^T : \right\rangle, \quad D = \begin{bmatrix} D_1 & 0 \\ 0 & D_1 \end{bmatrix}, \quad D_1 = \bigoplus_j \frac{1}{e^{E_j/T} - 1}. \quad (1)$$

Here, the boldface operator vectors $\hat{\mathbf{a}}^\dagger = (\hat{a}_1^\dagger, \hat{a}_2^\dagger, \dots)^T$ and $\hat{\mathbf{a}} = (\hat{a}_1, \hat{a}_2, \dots)^T$ are the column-vectors of the creation and annihilation operators for bare excited atom states, respectively. The superscripts T and † denote transpose and conjugate transpose, respectively. The angles stand for the statistical average. The colons denote the normal ordering of operators, meaning that all creation operators stand to the left from the annihilation ones. The 2×2 block-diagonal matrix D include the Bose–Einstein thermal occupation numbers for the quasiparticles of eigenenergies $\{E_j\}$. The matrix R performs the Bogoliubov transformation

$$\begin{pmatrix} \hat{\mathbf{a}}^\dagger \\ \hat{\mathbf{a}} \end{pmatrix} = R \begin{pmatrix} \hat{\mathbf{b}}^\dagger \\ \hat{\mathbf{b}} \end{pmatrix}, \quad R = \begin{bmatrix} U^* & 0 \\ 0 & U \end{bmatrix} \begin{bmatrix} \cosh r^* & (e^{i\theta} \sinh r)^* \\ e^{i\theta} \sinh r & \cosh r \end{bmatrix}, \quad (2)$$

from the representation of the excited-particle field operator in terms of the quasiparticle creation and annihilation operators $\hat{\mathbf{b}}^\dagger = (\hat{b}_1^\dagger, \hat{b}_2^\dagger, \dots)^T$ and $\hat{\mathbf{b}} = (\hat{b}_1, \hat{b}_2, \dots)^T$ to the representation in terms of the bare-particle creation and annihilation operators, that is,

$$\hat{\psi}_{\text{ex}}(\mathbf{r}) = \sum_{k \neq 0} \phi_k(\mathbf{r}) \hat{a}_k = \sum_j (u_j(\mathbf{r}) \hat{b}_j + v_j^*(\mathbf{r}) \hat{b}_j^\dagger). \quad (3)$$

Within the mean-field Bogoliubov–Popov approximation [51–53], adopted in the present paper, a Bose–Einstein-condensed gas is described via the Hamiltonian given by a quadratic form of the bare-particle creation and annihilation operators

$$\begin{aligned} \hat{H} = & \sum_{k,k'} \left(\hat{a}_k^\dagger \hat{a}_{k'} + \frac{1}{2} \right) \int \phi_k^* \left(-\frac{\hbar^2 \Delta}{2m} + U(\mathbf{r}) - \mu + 2g(N_0 |\phi_0(\mathbf{r})|^2 + n_{\text{ex}}(\mathbf{r})) \right) \phi_{k'} d^3 \mathbf{r} \\ & + \frac{gN_0}{2} \sum_{k,k'} \hat{a}_k^\dagger \hat{a}_{k'}^\dagger \int \phi_k^* \phi_0^2 \phi_{k'}^* d^3 \mathbf{r} + \frac{gN_0}{2} \sum_{k,k'} \hat{a}_k \hat{a}_{k'} \int \phi_k (\phi_0^*)^2 \phi_{k'} d^3 \mathbf{r}. \end{aligned} \quad (4)$$

Here, Δ is the three-dimensional Laplace operator, m is a particle mass, $U(\mathbf{r})$ is an external potential, $g = 4\pi\hbar^2 a/m$ is an interaction constant, a is a s -scattering length, μ is a chemical potential, N_0 is the mean number of particles in the condensate, $n_{\text{ex}}(\mathbf{r})$ is the mean density of the excited particle fraction (the noncondensate), and $\phi_0(\mathbf{r})$ is a condensate wave function normalized to unity, $\int_V |\phi_0|^2 d^3 \mathbf{r} = 1$. The superscript $*$ means complex conjugation. The

Hamilton operator (4) can be equivalently rewritten in the matrix form via the (2×2) -block Hamiltonian matrix H as follows

$$\hat{H} = \begin{pmatrix} \hat{\mathbf{a}}^\dagger \\ \hat{\mathbf{a}} \end{pmatrix}^T H \begin{pmatrix} \hat{\mathbf{a}}^\dagger \\ \hat{\mathbf{a}} \end{pmatrix}, \quad H = \begin{bmatrix} \tilde{K} & K \\ K^* & \tilde{K}^* \end{bmatrix}. \tag{5}$$

The Bogoliubov transformation (2) diagonalizes the blocks K responsible for the co-rotating contributions $\hat{a}_k^\dagger \hat{a}_{k'}$ or $\hat{a}_k \hat{a}_{k'}^\dagger$ to the Hamiltonian and nullifies the blocks \tilde{K} responsible for the counter-rotating contributions $\hat{a}_k^\dagger \hat{a}_{k'}^\dagger$ or $\hat{a}_k \hat{a}_{k'}$ to the Hamiltonian,

$$R^T H R = \begin{bmatrix} 0 & E \\ E & 0 \end{bmatrix}, \quad E = \text{diag}\{E_j | j = 1, 2, \dots\}. \tag{6}$$

The point is that the quasiparticles are the eigenstates of the Bogoliubov Hamiltonian with the eigenenergies $\{E_j | j = 1, 2, \dots\}$.

The multimode squeezing matrix [1,68–73] $r = (r_{k,k'})$, which is a positive semi-definite Hermitian matrix, as well as the unitary matrices U and $e^{i\theta}$, are calculated in [1]. Any additional unitary transformation V to another complete orthonormal set of excited states $\{\phi'_k(\mathbf{r}) | k = 1, 2, \dots\}$ in the single-particle Hilbert space,

$$\phi_k = \sum_{k' \neq 0} V_{k',k} \phi'_{k'}, \tag{7}$$

results in the Bogoliubov transformation R' which differs from the R in Equation (2) just by replacement of the unitary U with the composite unitary transformation $U' = VU$.

Clearly, the Bogoliubov transformation (2) is a superposition of the squeezing and unitary transformations. In essence, their matrices r and U determine the complexity and variability of the covariance matrix G , since the classical thermal occupations of quasiparticles entering the matrix D are easy to compute. As a result, the ultimate reason for the $\#P$ -hardness (quantum advantage) of atomic boson sampling is an interplay between the squeezing (found in [74]) due to the interparticle interactions and interference due to the unitary mixing of bare-particle excited states. If either the squeezing or interference vanishes, then the joint occupation probabilities can be computed classically in polynomial time.

1.5. The Content of the Paper

Based on the main aspects of a truly hard for computing and innovative problem of atomic boson sampling in a BEC trap explained in the Introduction (Section 1), we formulate, in Section 2, general requirements for the BEC trap design facilitating testing the quantum advantage of atomic boson sampling in a full, controllable and clear way. A basic model of a multi-qubit BEC trap is devised in Section 3. In Sections 4 and 5, we present the analytical and numerical results for the single-particle energy spectrum and eigenstates in the one- and two-dimensional multi-qubit traps, respectively. The corresponding solutions to the Gross–Pitaevskii equation for the condensate wave function are presented in Section 6. In Section 7, they are employed in the analysis of the Bogoliubov transformation and couplings for estimates of the multimode squeezing parameters. We discuss also the multimode dimensionality, squeezing and interference, which determine an asymptotic parameter for computational $\#P$ -hardness of atomic boson sampling and require analysis of the Bogoliubov–De Gennes equations for the quasiparticle spectrum and eigenfunctions. The experimental aspects of atomic boson sampling are discussed in Section 8. Section 9 contains concluding remarks.

2. General Requirements for the BEC Trap Design Facilitating Atomic Boson Sampling

Although any general-case BEC trap can be employed for studying manifestations of its potential quantum advantage over classical computing of boson sampling [1], in order

to test this advantage in a controllable and unambiguous way, one should better use a specially designed trapping potential (see an example in Figure 1).

The challenge of the BEC trap design is twofold. On the one hand, it is desirable to have a trap with a finite (or even mesoscopic) number, M , of lower split-off excited states or groups of states, which are predominantly populated and strongly coupled to each other by means of Bogoliubov coupling. The atomic boson sampling could refer to the so-called marginal statistics—the quantum statistics of the occupations of these excited states irrespective to the occupations of all other states. It is especially informative if all of the higher excited states are separated from such a split-off miniband or sub-miniband of the selected M lower excited states or some groups of them by an energy gap ΔE wider than the temperature T and are not significantly coupled to the lower energy states. Then, these higher states are relatively poor populated, do not contribute to $\#P$ -hard complexity and can be skipped or accounted for as a kind of perturbation.

On the other hand, it is required to provide a way to simultaneously measure, that is to sample, the occupations of those M excited particle states or groups of them, say, by means of multi-detector optical imaging based on the light transmission through or scattering from the atomic cloud. Each detector should measure an appropriate occupation by projecting upon a prescribed state or group of states. Moreover, this subset of states or groups of states should be variable and controllable by means of tuning the detectors.

The geometry of a 2D multi-qubit BEC trap, such as the one shown in Figure 1, looks especially convenient for such boson sampling experiments since it allows one to implement multi-detector imaging by means of the laser light passing through the trap perpendicular to its plane. A controllable reconfiguration of the system of detectors aimed at varying the states or groups of states prescribed for occupation sampling also looks easier in the 2D geometry.

Suppose we design a multi-qubit BEC trap with a confining potential $U(\mathbf{r})$ supporting a finite number Q of single-qubit cells, which form a 1D, 2D or 3D lattice and have two split-off lower energy levels each. Those two levels appear when a twofold-degenerate ground level is split by a certain perturbation. Then, such a lattice of single-qubit cells should be placed on top of a slightly varying in space background potential with high walls at the trap borders, and the inter-cell potential walls should be adjusted to be relatively high but narrow enough to allow for a quantum tunneling of atoms between cells. All this is necessary for establishment of the common to all cells nonuniform condensate and significant interaction between atoms from different single-qubit cells. The last two conditions are required for the existence of significant Bogoliubov coupling between a large number of excited states without which the multimode squeezing as well as interference via dressed quasiparticles are not well pronounced in the Bogoliubov transformation matrix R in Equation (2) and, hence, the computational $\#P$ -hardness disappears.

In fact, building a confining potential in the form of a single-qubit cell and duplicating it into a lattice is a relatively straightforward enterprise since such potentials are reminiscent of a double-well potential and an optical lattice potential, in which the BEC as well as the Bogoliubov excitations had been studied a lot [35,43,44,75–81]. The size of the multi-qubit trap depends on its dimensionality. In the 2D case of Figure 1, an overall dimension of the BEC trap is about \sqrt{Q} μm , since each single-qubit cell has a scale of a de Broglie wavelength ~ 1 μm .

The starting point of our analysis is the limiting case of infinitely high inter-cell barriers and identical single-qubit cells, each with two single-particle eigenfunctions corresponding to the first and second energy levels, e_1 and e_2 . These eigenfunctions form a natural basis for constructing the single-particle excited states of the actual trap. There are 2^Q combinations of these single-qubit states which are the eigenfunctions of the whole multi-qubit trap. Their $Q + 1$ different energy levels $\{\varepsilon_q = (Q - q)e_1 + qe_2; q = 0, 1, \dots, Q\}$ constitute a lower energy miniband. Degeneracies of levels are given by binomial coefficients $g_q = \binom{Q}{q}$. Their sum coincides with the number of the single-qubit-state combinations: $\sum_{q=0}^Q \binom{Q}{q} = 2^Q$.

Those limiting-case eigenfunctions of the multi-qubit trap emerge adiabatically from the wavefunctions of an empty flat box trap when the inter-cell potential barriers are gradually introduced. This process can be easily understood within a 1D model of a flat background potential and almost equally spaced delta-function potential barriers. Its analysis shows that the limiting-case eigenfunctions of the finite lattice of independent qubit cells correspond to some superpositions of the $2Q$ lower-energy eigenfunctions of the whole trap with finite barriers. Hence, the $2Q$ lower-energy eigenfunctions $\psi_n \equiv \psi_{s,p}$ of order $n = p + sQ$, $p = 1, \dots, Q$, $s = 0, 1$, of the actual trap with finite barriers can be considered as a system of the “generating” eigenfunctions constituting two bands ($s = 0, 1$) and enumerated by the intra-band index $p = 1, \dots, Q$ and the band index $s = 0, 1, \dots$

Although one could consider all 2^Q energy levels associated with Q qubits, it is more convenient to operate with a smaller number of the single-particle energy levels $M + 1$ which constitute some miniband. In Sections 4 and 5, we show that it is possible to choose the trap parameters in such a way that $M + 1 = 2Q$ levels will form a lower miniband separated from the higher energy levels by an energy gap ΔE wider than the temperature T . Below, we mainly discuss the multi-qubit trap properties associated with such a miniband.

Analysis of the case when each cell has a larger number, $d > 2$, of the lower split-off energy levels is very similar. Then, a finite lattice of such cells forms a multi-qudit trap.

In any case, the eigenfunctions $\psi_{s,p}$ and energy levels of the actual trap with a finite trapping potential can be easily controlled and varied in a wide range by means of controlling the background and barrier potentials as well as the dimensions of the single-qubit cells. For instance, the relative occupations of cells, i.e., the relative wavefunction amplitudes in different cells, within an eigenfunction of a given order can be individually controlled by tuning the cell background potentials. The intra-cell qubit properties, including the energy splittings $\delta E_j, j = 1, \dots, Q$, can be addressed by adjusting the intra-cell barriers.

The ground-state properties also can be controlled in this way. Implementing also control of the interparticle interactions via the Feshbach resonance [65], one can adjust the condensate wave function as needed. Below, we consider a favorable for the atomic boson sampling regime of a common condensate which is macroscopically occupied and inhomogeneously spread over the entire trap at a low temperature $T \ll T_c$ and a relatively large number of trapped atoms $N \gg Q$. At certain conditions, a particular number M (for example, $M = 2Q - 1$) of lower-miniband excited states can be considered as being decoupled from the continuum of excited states of the total infinite-size Hilbert space and constituting a finite-size Hilbert subspace. The situation could become especially clean and favorable for atomic boson sampling experiments if, in addition, the Bogoliubov couplings are adjusted to be spread over the whole lower miniband but not above the energy gap.

Apparently, the multi-qubit trap is capable of providing a whole series of other BEC regimes [78], starting from a strongly correlated regime and a regime of anomalous fluctuations in the critical region at $T \approx T_c$ to the regimes of fragmented condensates of the individual single-qubit cells and a quasi-condensate. However, their discussion goes beyond the scope of the present paper.

3. A Basic Model of a Multi-Qubit BEC Trap

The first step in designing the multi-qubit trap is to find its single-particle energy spectrum $\{\varepsilon_n | n = 0, 1, \dots\}$ and eigenstates $\{\psi_n\}$, given by the linear Schrödinger equation

$$\left(-\frac{\hbar^2}{2m}\Delta + U(\mathbf{r}) \right) \psi_n(\mathbf{r}) = \varepsilon_n \psi_n(\mathbf{r}), \tag{8}$$

and adjust the trap parameters in order to fulfill the requirements on the split-off lower-energy miniband formulated above. The energies may be counted from the energy ε_0 of a nondegenerate ground state $n = 0$. An integer n orders all eigenstates in increasing energies $\varepsilon_0 < \varepsilon_1 \leq \varepsilon_2 \leq \dots$. Solutions to the single-particle Schrödinger Equation (8) provide a valuable starting point for the design of the multi-qubit trap—the zero-order

approximation for the energies and wave functions of the excited states ($n = 1, 2, \dots$) as well as the wave function of the ground state ($n = 0$).

The second step is to find how the repulsive interparticle interaction modifies the ground state, that is, to find the condensate wave function $\phi_0(\mathbf{r})$ which obeys the Gross–Pitaevskii equation (the nonlinear Schrödinger equation) [35,52,53]

$$\left(-\frac{\hbar^2 \Delta}{2m} + U(\mathbf{r}) + gN_0|\phi_0(\mathbf{r})|^2 + 2gn_{\text{ex}}(\mathbf{r}) - \mu \right) \phi_0 = 0, \quad g = \frac{4\pi\hbar^2 a}{m}. \quad (9)$$

The goal is to verify the presence of a non-fragmented condensate, which is common for the entire trap and spreading over all single-qubit cells. A non-uniformity of the condensate should be controllable by adjusting the trap parameters. Accurate knowledge of the condensate wave function is necessary for calculating the Bogoliubov couplings

$$\Delta_{kk'} = g N_0 \int \phi_k^*(\mathbf{r}) |\phi_0(\mathbf{r})|^2 \phi_{k'}(\mathbf{r}) d^3\mathbf{r}, \quad \tilde{\Delta}_{kk'} = g N_0 \int \phi_k^*(\mathbf{r}) \phi_0(\mathbf{r})^2 \phi_{k'}^*(\mathbf{r}) d^3\mathbf{r} \quad (10)$$

between the preselected bare-particle excited states $\{\phi_k | k = 1, 2, \dots\}$ and making sure that they are well pronounced for a large enough number of these states as per requirements stated in Section 2. If most of atoms are in the condensate, $N_0 \approx N$, then a characteristic length of a condensate inhomogeneity is equal to a so-called healing length

$$\xi = \frac{\hbar}{\sqrt{2mgN/V}} = \frac{1}{\sqrt{8\pi aN/V}}. \quad (11)$$

The Gross–Pitaevskii equation, as a mean field approximation, is valid if an average distance d between atoms is small compared to the healing length,

$$d \ll \xi. \quad (12)$$

The next step involves solving the Bogoliubov–De Gennes equations for the quasiparticle spectrum and eigenfunctions as well as calculation of the Bogoliubov transformation matrix, squeezing and other parameters describing the joint probability distribution of the excited atom occupations and atomic boson sampling. We just briefly comment on this step in Sections 7 and 8, since the analysis of quasiparticles goes beyond the scope of this article.

In the present paper, we limit ourselves to the first two steps and calculation of Bogoliubov couplings responsible for interparticle interactions in the Bogoliubov Hamiltonian.

For the sake of clarity and simplicity, we consider only a simple basic model of the multi-qubit BEC trap illustrated in Figure 2: namely, a one-dimensional (1D) or two-dimensional (2D) array of a finite number Q of the single-qubit cells. In the case of a 1D chain of the single-qubit cells, each q -th single-qubit cell includes two flat background potentials U_{2q-1}, U_{2q} and a delta-function potential $\beta_q \delta(x - x_q)$ located near its center, while the cells are separated by the delta-function potential walls $\{\alpha_q \delta(x - X_q) \geq 0 | q = 1, 2, \dots, Q - 1\}$ and ordered along the x axis so that $0 = X_0 \leq x_1 \leq X_1 \leq x_2 \leq X_2 \leq \dots \leq x_Q \leq X_Q = QL$. The corresponding 1D trapping potential is modeled as follows

$$U(x) = \sum_{q=1}^Q \{ U_{2q-1} [\theta(x - X_{q-1}) - \theta(x - x_q)] + U_{2q} [\theta(x - x_q) - \theta(x - X_q)] + \beta_q \delta(x - x_q) + \alpha_q \delta(x - X_q) \} \quad \text{if } x \in (0, QL);$$

$$U(x) = \infty \quad \text{if } x \leq 0 \quad \text{or } x \geq QL. \quad (13)$$

The amplitudes of the background potentials $\{U_j | j = 1, \dots, 2Q\}$ and all delta-function potentials $\{\alpha_q\}, \{\beta_q\}$ as well as their locations $\{x_q\}, \{X_q\}$ could be different for different single-qubit cells and constitute a set of controllable parameters of the multi-qubit BEC

trap; $\delta(x)$ is the Dirac delta function, and $\theta(x)$ is the unit step function: $\theta(x) = 0$ if $x < 0$, $\theta(x) = 1$ if $x \geq 0$.

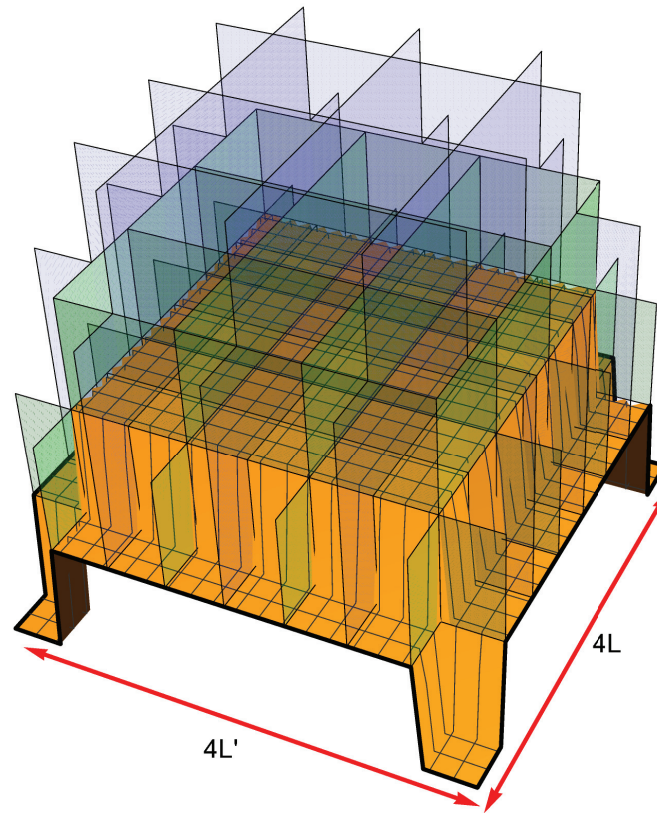


Figure 2. (4×4) -qubit BEC trap of a dimension $4L \times 4L'$ as per the 2D model (14) of trapping potential $U(x, y)$ consisting of the inter- and intra-cell walls atop a central pedestal. The infinite outer walls of the entire box trap are not shown.

In the case of a 2D square $Q_1 \times Q_1$ array of $Q = (Q_1)^2$ single-qubit cells, we adopt a model potential $U(x, y) = U(x) + U'(y)$ given by a sum of two 1D potentials along the axes x and y , each of which being similar to the 1D potential in Equation (13):

$$\begin{aligned}
 U(x) &= \sum_{q=1}^{Q_1} \{U_{2q-1}[\theta(x - X_{q-1}) - \theta(x - x_q)] + U_{2q}[\theta(x - x_q) - \theta(x - X_q)] \\
 &\quad + \beta_q \delta(x - x_q) + \alpha_q \delta(x - X_q)\} \quad \text{if } x \in (0, Q_1 L); \\
 U'(y) &= \sum_{q=1}^{Q_1} \{U'_{2q-1}[\theta(y - Y_{q-1}) - \theta(y - y_q)] + U'_{2q}[\theta(y - y_q) - \theta(y - Y_q)] \\
 &\quad + \beta'_q \delta(y - y_q) + \alpha'_q \delta(y - Y_q)\} \quad \text{if } y \in (0, Q_1 L').
 \end{aligned}
 \tag{14}$$

Again, we set the potential to be infinitely high beyond the outer borders of the entire multi-qubit trap: $U(x) = \infty$ if $x \leq 0$ or $x \geq Q_1 L$, $U'(y) = \infty$ if $y \leq 0$ or $y \geq Q_1 L'$. The amplitudes of the background potentials $\{U_j | j = 1, \dots, 2Q\}$, $\{U'_j | j = 1, \dots, 2Q\}$ and all delta-function potentials $\{\alpha_q\}$, $\{\beta_q\}$, $\{\alpha'_q\}$, $\{\beta'_q\}$ as well as their locations $\{x_q\}$, $\{X_q\}$, $\{y_q\}$, $\{Y_q\}$ constitute a set of controllable parameters of the 2D multi-qubit BEC trap.

Modeling the confining potential by piecewise flat and delta-function potentials is a well-justified textbook approach pertinent to the analysis of the effects of tunneling, reflection and trapping of particles by potential barriers and walls on the wave functions and energy spectrum in quantum mechanics (see, e.g., [82–85] and references therein). It is consistent with the well-known facts that (a) the Rayleigh–Ritz characterization of the

eigen energies involves only weighted averages of the potential and (b) the multiple-scale perturbation theory yields the correct leading-order asymptotics within the piecewise-flat-potentials approximation [86]. The main quantities in question for the analysis in the present paper are the condensate wave function and Bogoliubov couplings, which determine the ultimate result for the covariance matrix and statistics of atomic boson sampling. Their representativeness and robustness with respect to the adopted modeling by the piecewise flat and delta-function potentials are predetermined by the nature of the Bogoliubov couplings (10) as the overlapping integrals which do not depend significantly on a jump in the value of the first or second derivative of the wave function originated from the presence of the delta- or step-function, respectively, in the external potential. Furthermore, the actual potential in the Gross–Pitaevskii and Bogoliubov–de Gennes Equations (9) and (34) is always curved by the interparticle–interaction contribution $gN_0\phi_0^2(\mathbf{r})$ proportional to the continuous condensate occupation $|\phi_0^2(\mathbf{r})|$. Obviously, in an experimental setting, a non-flat background potential will lead to qualitatively the same results.

4. One-Dimensional Multi-Qubit Trap: Single-Particle Eigen Functions and Energies

Consider a 1D trap with the model potential (13). The basic model adopted above allows one to solve the 1D Schrödinger Equation (8),

$$\left(-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + U(x)\right) \psi_n(x) = \varepsilon_n \psi_n(x), \tag{15}$$

analytically and easily find the single-particle energy spectrum and wave eigenfunctions. In this section, we demonstrate the single-particle properties of the 1D multi-qubit traps in a series of generic examples.

4.1. Asymmetric 1D Single-Qubit Trap: Explicit Solution for a Double-Well Trap

The solution to Equation (15) for the eigen functions and energies of an asymmetric 1D single-qubit trap described by the model (13) of a double-well trap with the intra-cell delta-function potential of a magnitude β located at a position $x_1 = \eta L$, $\eta \in (0, 1)$, is elementary:

$$\psi_n(x) = A \sin(k_n x) \text{ if } 0 \leq x \leq \eta L, \quad \psi_n(x) = \frac{A \sin(\eta k_n L)}{\sin[(1 - \eta)k_n L]} \sin[k_n(L - x)] \text{ if } \eta L \leq x \leq L, \tag{16}$$

$$\sin(\eta k_n L) \sin[(1 - \eta)k_n L] = -\frac{\hbar^2 k_n}{2m\beta} \sin(k_n L), \quad \varepsilon_n = \frac{\hbar^2 k_n^2}{2m}. \tag{17}$$

Here, A is an appropriate normalization constant. The dependence of the first six eigen wave numbers k_n on the asymmetry parameter $\eta \in (0, 1)$ is illustrated in Figure 3. Note a very narrow energy splitting $\varepsilon_2 - \varepsilon_1 \ll \varepsilon_2$ between the two lower excited states $n = 1, 2$ and a very wide energy gap $\varepsilon_3 - \varepsilon_2 \approx 3\varepsilon_2$ separating them from the next two excited states $n = 3, 4$ in the case of the central, symmetric location of the intra-cell delta-function potential, $\eta = 1/2$. With an increasing asymmetry, the energy ladder experiences a significant restructuring. For example, if the asymmetry is $\eta \approx 1/3$ or $2/3$, then already, the three lower (the lowest first, ε_1 , and two very close second and third, $\varepsilon_2 \approx \varepsilon_3$) energy levels are separated from the higher energy levels by an energy gap $\varepsilon_4 - \varepsilon_3 \approx 1.3\varepsilon_3$.

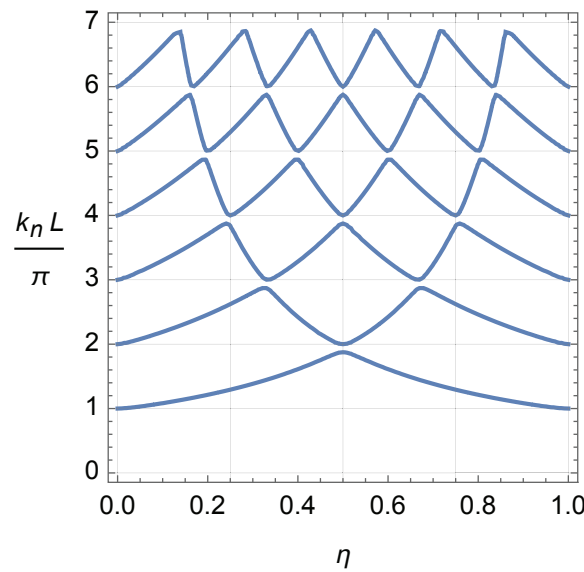


Figure 3. The first six eigen wave numbers k_n for the 1D asymmetric single-qubit trap of length L as the functions of the position ηL of the intra-cell delta-function potential of the dimensionless amplitude $\beta mL/\hbar^2 = 2.5$.

4.2. Symmetric 1D Two-Qubit Trap: Even versus Odd Eigenfunctions and Their Eigenenergies

Consider the symmetric two-qubit trap (13) with the central locations of the intra-cell delta-function potentials of equal magnitude $\beta_1 = \beta_2 \equiv \beta$ at $x_1 = L/2, x_2 = 3L/2$ and the inter-cell delta-function potential of the magnitude $\alpha_1 \equiv \alpha$ at $X_1 = L$ in the absence of the background potential, $U_1 = U_2 = 0$. The odd wave functions, which have the odd spatial symmetry relative to the center of the trap, equal zero at the trap center and are not affected by the inter-cell potential wall. Obviously, solutions for them are reduced to the single-qubit trap solution (16) in each of two single-qubit cells. For example, the odd wave function in the left single-qubit cell is

$$\psi_n(x) = A \sin(k_n x) \text{ if } 0 \leq x \leq L/2, \quad \psi_n(x) = A \sin[k_n(L - x)] \text{ if } L/2 \leq x \leq L. \quad (18)$$

Hence, the dimensionless energy spectrum, $\bar{\epsilon}_n = (2mL^2/\hbar^2)\epsilon_n$, of the odd eigenfunctions is given by Equation (17), that is

$$k_n L \cos(k_n L/2) + \bar{\beta} \sin(k_n L/2) = 0, \quad \bar{\epsilon}_n = (k_n L)^2; \quad \bar{\alpha} = \frac{\alpha mL}{\hbar^2}, \quad \bar{\beta} = \frac{\beta mL}{\hbar^2}. \quad (19)$$

A dimensionless parameter $\bar{\beta}$ describes the effect of the intra-cell delta-function potential. The solution to Equation (15) for the even eigenfunctions is more involved:

$$\begin{aligned} \psi_n(x) &= A_1 \sin(k_n x) & \text{if } 0 \leq x \leq L/2, \\ \psi_n(x) &= A_2 \sin[k_n(L - x) + \varphi] & \text{if } L/2 \leq x \leq L, \\ \psi_n(x) &= A_2 \sin[k_n(x - L) + \varphi] & \text{if } L \leq x \leq 3L/2, \\ \psi_n(x) &= A_1 \sin[k_n(2L - x)] & \text{if } 3L/2 \leq x \leq 2L. \end{aligned} \quad (20)$$

It includes two normalization constants A_1, A_2 , the phase shift φ (such that $\tan \varphi = k_n L/\bar{\alpha}$) and depends on the inter-cell delta-function potential via the dimensionless parameter $\bar{\alpha} = \alpha mL/\hbar^2$. The energy spectrum $\bar{\epsilon}_n = (k_n L)^2$ of the even eigenfunctions is determined by the eigen wave number k_n that can be found from the explicit transcendental equation:

$$k_n L \left[k_n L \cos(k_n L) + \bar{\beta} \sin(k_n L) \right] + 2\bar{\alpha} \sin\left(\frac{k_n L}{2}\right) \left[k_n L \cos\left(\frac{k_n L}{2}\right) + \bar{\beta} \sin\left(\frac{k_n L}{2}\right) \right] = 0. \quad (21)$$

Figure 4 shows clearly the full structure of energy spectrum of a two-qubit trap. Firstly, one can see the unperturbed energy level spread for an empty rectangular well when the inter-cell and intra-cell potentials are zero. This behavior can be modulated in two ways, by increasing either of the two potentials. There is almost a complete symmetry between how these two potentials effect the energy level structure, with the only difference coming in the even-numbered energy levels. While every fourth level is totally unperturbed by both potentials, the other even energy levels only see the intra-cell potentials, as these functions are zero at the center of the well. This leads to an asymmetry in the structure, which affects the orange-colored energy levels in the figure.

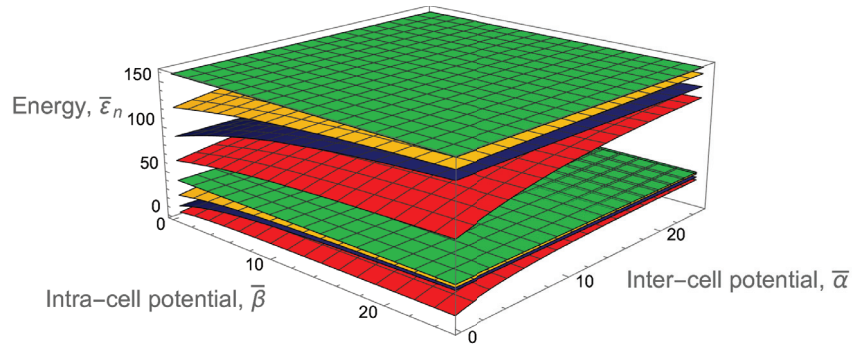


Figure 4. The first eight single-particle energy levels of the 1D symmetric two-qubit trap in the absence of a background potential as they depend on the inter-cell and intra-cell delta-function potential barriers as per Equations (19) and (21). On the far left, the unperturbed energy levels can be observed and compared to the energy levels on the far right, which show clearly the miniband behavior, with a larger gap between the strongly grouped first four and second four energy levels.

In addition to this asymmetry, the overall structure of the two-qubit-trap energy spectrum is largely determined by the formation of minibands. If either of the two potentials are individually raised to be large, four sub-minibands are formed, while the raising of both potentials leads to the formation of two minibands, with a large energy gap between the first four and second four energy levels. This can be quantitatively measured by taking the ratio of the energy separation between the first and second miniband with the energy width of the first miniband (see Figure 5). This will demonstrate how easy it will be to set the temperature such that the lower miniband is fully populated while the higher has little to no occupation. This figure of merit must be balanced against the necessity that atoms are still able to easily move between cells, requiring that the potential barriers not be too high.

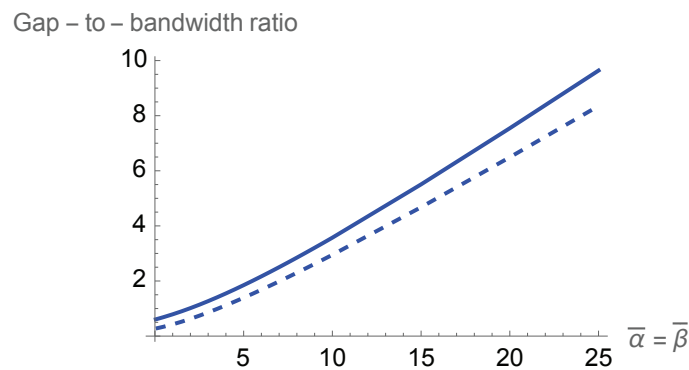


Figure 5. The ratio of the energy gap between the first and second minibands over the width of the first miniband for the symmetric two- (solid) and four- (dashed) qubit traps (Figures 4 and 6) as a function of the equal dimensionless amplitudes of the inter- and intra-cell delta-function potentials, $\bar{\alpha} = \bar{\beta}$.

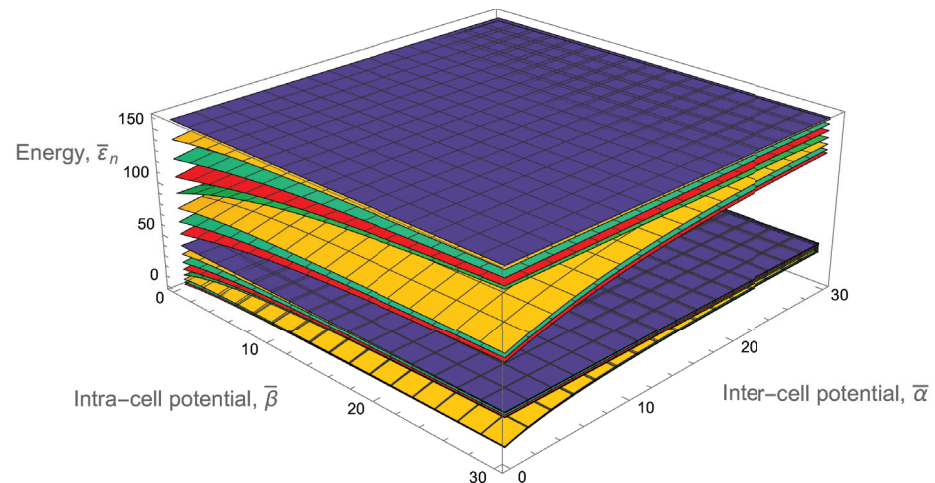


Figure 6. The first sixteen single-particle energy levels of the 1D four-qubit chain of identical symmetric single-qubit cells in the absence of a background potential ($U_j = 0 \forall j$) as they depend on the inter-cell and intra-cell delta-function potentials as per Equations (19), (21) and (24).

4.3. Four-Qubit Chain of Identical Symmetric Single-Qubit Cells: Hierarchy of Even/Odd Solutions

The analysis presented in Section 4.2 can be easily generalized to the case of the four-qubit trap (13) with similarly symmetric parameters $\alpha_q = \alpha, \beta_q = \beta, x_q = (q - 1/2)L, X_q = qL, U_{2q-1} = U_{2q} = 0 \forall q = 1, 2, 3, 4$ and total length $4L$. Again, the solutions for the wave functions with the odd spatial symmetry relative to the center of the trap and their energy spectrum are reduced to the solutions for the half trap, that is, for the two-qubit trap and, hence, are given (say, for the left half of the four-qubit trap) by Equations (18)–(21). The only novel element of the analysis is the solution for the even eigenfunctions. It has the following form in the left half of the four-qubit trap

$$\begin{aligned}
 \psi_n(x) &= A_1 \sin(k_n x) & \text{if } 0 \leq x \leq L/2, \\
 \psi_n(x) &= A_2 \sin[k_n(L - x) + \varphi_2] & \text{if } L/2 \leq x \leq L, \\
 \psi_n(x) &= A_3 \sin[k_n(x - L) + \varphi_3] & \text{if } L \leq x \leq 3L/2, \\
 \psi_n(x) &= A_4 \sin[k_n(2L - x) + \varphi_4] & \text{if } 3L/2 \leq x \leq 2L,
 \end{aligned}
 \tag{22}$$

with the same form of equations being found reflected across the center of the trap at $x = 2L$. Now, it includes four normalization constants A_1, A_2, A_3, A_4 , three phase shifts $\varphi_2, \varphi_3, \varphi_4$ and the eigen wave number k_n . The latter four quantities can be found from the following four equations expressing a discontinuity of the wave-function derivative across each delta-function potential barrier:

$$\begin{aligned}
 k_n L [\cot(\varphi_4) + \cot(\varphi_4)] &= 2\bar{\alpha}, \\
 k_n L [\cot(\varphi_3 + k_n L/2) + \cot(\varphi_4 + k_n L/2)] &= -2\bar{\beta}, \\
 k_n L [\cot(\varphi_2) + \cot(\varphi_3)] &= 2\bar{\alpha}, \\
 k_n L [\cot(k_n L/2) + \cot(k_n L/2 + \varphi_2)] &= -2\bar{\beta}.
 \end{aligned}
 \tag{23}$$

Excluding the phase shifts, we arrive to the explicit transcendental equation,

$$\begin{aligned}
 (k_n L/2)^3 [\bar{\beta} \sin(2k_n L) + k_n L \cos(2k_n L)] \\
 + (k_n L/2)^2 (2\bar{\alpha} + \bar{\beta}) \sin(k_n L) [\bar{\beta} \sin(k_n L) + k_n L \cos(k_n L)] \\
 + \bar{\alpha}^2 \sin^2(k_n L/2) [\bar{\beta} \sin(k_n L/2) + k_n L \sin(k_n L/2)]^2 \\
 + k_n L \bar{\alpha} \bar{\beta} \sin^2(k_n L/2) [\bar{\beta} \sin(k_n L) + k_n L \cos(k_n L)] = 0,
 \end{aligned}
 \tag{24}$$

for the eigen wave number k_n which determines the dimensionless energy spectrum $\bar{\epsilon}_n = (k_n L)^2$ of the even eigenfunctions.

The entire energy spectrum is illustrated in Figure 6 by dependence of the first sixteen lower energy levels on the inter-cell and intra-cell delta-function potentials. As expected, it is similar to the analogous dependence for the two-qubit trap shown in Figure 4. Again, on the far left, the unperturbed energy levels can be observed and compared to the energy levels on the far right, which show clearly the expected miniband behavior, with a larger gap between the strongly grouped first eight and second eight energy levels.

Let us look again at the ratio of the energy gap between the first and second minibands and the energy width of the first miniband in Figure 5. Although this ratio is smaller for similar delta-function potentials of the two-qubit trap, the doubling of the available energy levels is a strong advantage. The degradation is not significant, as to once again achieve a ratio of about 3.5, we only need to go from a dimensionless delta-function potential magnitude of 10 to about 12.

Apparently, the even/odd hierarchy of solutions revealed above is suggestive for an extension to any 1D chain of $Q = 2^p, p = 1, 2, 3, \dots$, identical symmetric single-qubit cells.

4.4. Multi-Qubit Chain of Q Identical Single-Qubit Cells: Asymptotics of Zeroes and Miniband of $2Q$ Energy Levels

Consider a 1D chain of Q identical symmetric single-qubit cells, each with a zero flat potential and of length L , separated by delta-function potential walls of the same amplitude $\alpha_q = \alpha$. The system is placed inside an infinitely high box potential well of length QL . Assume that each single-qubit cell contains a delta-function potential of the equal amplitude, $\beta_q = \beta$, placed at the center of the cell and perturbing its energy levels.

As was explained in Section 2, the eigenfunctions of the multi-qubit trap $\{\psi_{s,p}(x) | p = 1, \dots, Q; s = 0, 1, \dots\}$ can be considered as arising adiabatically with increasing delta-function potentials α and β from the sinusoidal wave eigenfunctions of a box trap with a zero flat potential and of length QL ,

$$\psi_n^{(0)}(x) = \sqrt{\frac{2}{QL}} \sin\left(\frac{n\pi x}{QL}\right), \quad \epsilon_n^{(0)} = \frac{(\hbar\pi n)^2}{2m(QL)^2}, \quad n = p + sQ; p = 1, \dots, Q; s = 0, 1, \dots \quad (25)$$

The index n is equal to the number of half-wavelength variations between the borders of the entire box trap and orders the wave functions in accord with the linearly growing numbers of zeroes, $n - 1$, and quadratically growing energies, $\epsilon_n^{(0)}$. The index $s = 0, 1, \dots$ enumerates the bands. Each band consists of Q eigenfunctions enumerated by the intra-band index $p = 1, \dots, Q$.

We find a general asymptotic rule: When $\alpha_q \rightarrow \infty$, all Q eigenfunctions $\psi_{s,p}$ within a given s -band have exactly the same (equal to the band order s) number of zeroes inside each single-qubit cell. The only exception constitutes such single-qubit cells and such eigenfunctions $\psi_{s,p}$ for which there is just one zero of the corresponding sinusoidal eigenfunction $\psi_{p+sQ}^{(0)}$ located exactly at the center of a single-qubit cell. This is an exceptional, degenerate case of a node frustration when neither of the two delta-function walls of the single-qubit cell are able to shift the location of this zero toward (underneath) its (wall's) location with increasing delta-potential $\alpha_q \rightarrow \infty$. The amplitude of this eigenfunction tends to zero everywhere inside such an exceptional single-qubit cell.

The remarkable asymptotic behavior stated above is a consequence of the fact that the eigenfunctions $\psi_{s,p}(x)$ tend to zero at the positions of the inter-cell delta-potential walls with increasing magnitude of the delta-function potential: $\psi_{s,p}(x = jL) \rightarrow 0$ at $\alpha_q \rightarrow \infty$ for $j = 1, \dots, Q - 1$. This occurs via two mechanisms. A delta-potential wall either (i) gradually digs a deep dip forcing the eigenfunctions to approach zero at the wall location, or (ii) gradually shifts the closest-to-the-wall zero of the sinusoidal wave eigenfunction $\psi_{p+sQ}^{(0)}$ to (underneath) the wall location. Accordingly, the eigenfunctions do not or do change their sign across the delta-potential wall. It is illustrated in Figure 7(left), where both

mechanisms of asymptotics formation are clearly represented. In particular, the inter-cell delta-potential walls at the dimensionless positions $x/L = 2$ and $x/L = 6$ implement the first mechanism on the eigenfunctions $\psi_{s,p=1}$ (blue) and $\psi_{s,p=2}$ (yellow), the second mechanism on the eigenfunction $\psi_{s,p=3}$ (green), and do not affect the eigenfunction $\psi_{s,p=4}$ (red) whose sinusoidal counterpart $\psi_{4+sQ}^{(0)}$ is already equal to zero at the wall locations. Another situation when the above two mechanisms clearly manifest themselves is discussed in the next Section 4.5 in regard to Figure 11.

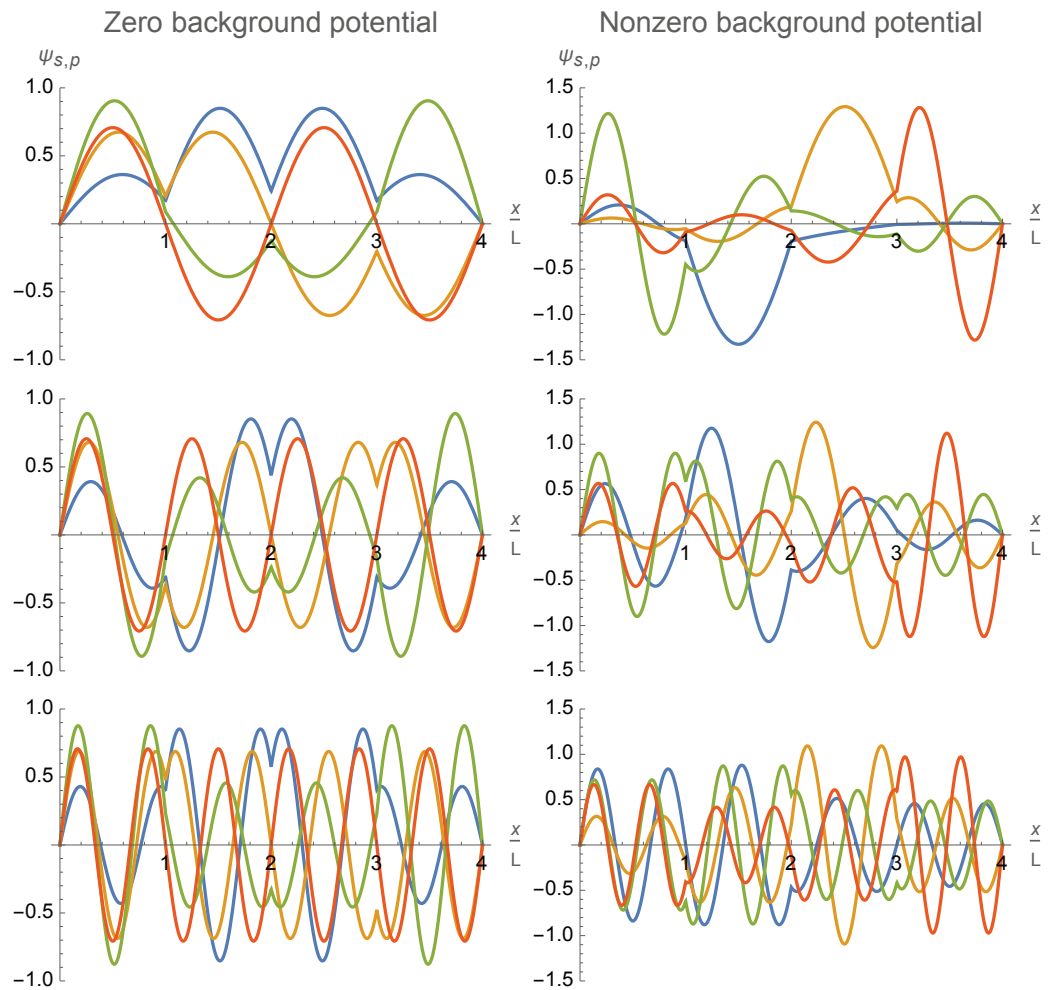


Figure 7. The first three bands, $s = 0$ (1st row), $s = 1$ (2nd row), and $s = 2$ (3rd row), of the eigenfunctions $\psi_{s,p}(x)$ ($p = 1$ in blue, $p = 2$ in yellow, $p = 3$ in green, $p = 4$ in red) for the 1D chain of $Q = 4$ single-qubit cells. The inter-cell walls are the delta-function potentials of the same magnitude, $\alpha_q = 9\hbar^2/mL$, located at the equally spaced dimensionless positions $x/L = 1, 2, 3$. There are no intra-cell potentials. The **left** column of graphs: Identical single-qubit cells with zero background potentials, $U_j = 0$. The **right** column of graphs: The single-qubit cells with different background flat potentials, $U_j mL^2/\hbar^2 = 0, 0, 10, 10, 20, 20, 0, 0$. In both cases, the two mechanisms of the general asymptotic rule (stated in Section 4.4) for a transition from the sinusoidal wave eigenfunctions (25) of a uniform box trap to the eigenfunctions of a multi-qubit trap are clearly observable.

The sinusoidal wave functions $\psi_n^{(0)}$ of the higher band orders $s \geq 2$, i.e., $n > 2Q$, have three or more zeroes at least within one, say j -th, single-qubit cell, that is, within the interval $x \in [qL, (q + 1)L]$. So, at $\alpha_q \rightarrow \infty \forall q$, they turn into the eigenfunctions $\psi_{s,p}, s \geq 2$, which has the s (two or more) zeroes inside each single-qubit cell and cannot be associated with the 2^Q eigenfunctions of the multi-qubit lowest miniband that have no more than one zero inside a single-qubit cell. Hence, only the first two bands of the eigenfunctions, $\psi_{0,p}$

and $\psi_{1,p}$, are relevant to the wave function superpositions that asymptotically yield the 2^Q eigenfunctions of the multi-qubit trap (i.e., combinations of the single-qubit states with the energies within the miniband).

Thus, we focus below on the analysis of the miniband of the first $2Q$ energy levels corresponding to the first two bands, $s = 0, 1$, of the eigenfunctions $\psi_{s,p}$. In principle, we can build a new system of Q qubits assigning an arbitrary pair of eigenfunctions $\psi_{0,p}$ and $\psi_{1,p'}$ to be the lower and upper energy states of any new qubit. The most natural system of Q qubits will be formed by the pairs of eigenfunctions $\{\psi_{0,p}, \psi_{1,p} | p = 1, \dots, Q\}$ with equal indices $p' = p$. Then, we again can consider their 2^Q multi-qubit combinations of the eigenfunctions of the miniband of the lowest $2Q$ energy levels in the multi-qubit trap with arbitrary finite (not necessary infinite) inter-cell potential walls. These qubits are not identical anymore, even if their lengths $L_q, q = 1, \dots, Q$ are the same.

4.5. Multi-Qubit Chain of Significantly Different Single-Qubit Cells: Control of Occupations, Energies

If the single-qubit cells in the multi-qubit chain are not identical, then the excited-state wave functions become less symmetric. However, by controlling the background flat potentials $\{U_j | j = 1, \dots, 2Q\}$ in Equation (13), one can make the ground state more uniform. A typical example of spatial profiles of the ground-state wave function and three lower-energy excited-state eigenfunctions is shown in Figure 8. In this figure and throughout the present paper as in Equation (19), a bar above a symbol of the potential or other energy quantity denotes its dimensionless value in terms of the energy unit $\hbar^2 / (2mL^2)$ where L is the length of a typical single-qubit cell, that is,

$$\bar{U}_j = (2mL^2 / \hbar^2)U_j, \quad \bar{\epsilon}_n = (2mL^2 / \hbar^2)\epsilon_n . \tag{26}$$

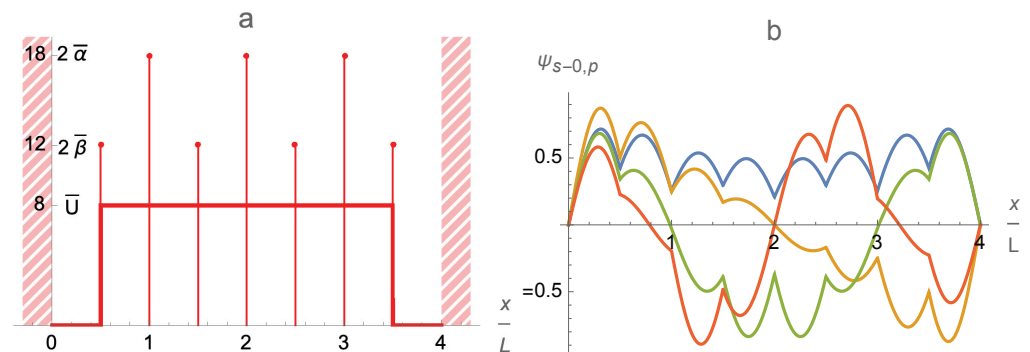


Figure 8. (a) An example of a four-qubit trap potential: Three inter-cell and four intra-cell delta-function potential barriers are separating eight flat potential segments in the form of a central pedestal. (b) The ground-state (blue) and first three excited-state eigenfunctions for the four-qubit trap (a).

Relatively large variations in the lengths, L_q , and background flat potentials, U_{2q-1}, U_{2q} , of different single-qubit cells allow one to control and vary the trap eigenfunctions $\psi_{s,p}$ and energy levels in a wide range. Adjusting separately the flat potentials U_j in different single-qubit cells allows one to control individually the relative amplitudes of eigenfunctions in different cells, that is, in particular, the relative occupations of different single-qubit cells as is illustrated in the right column of Figure 7.

The general structure and asymptotic behavior of the trap eigenfunctions described above for the chain of identical single-qubit cells is robust (remains qualitatively the same) with respect to small variations of the trap parameters. However, large variations change the picture. In particular, the single-qubit cells of significantly different lengths could acquire different numbers of eigenfunction zeroes per a single-qubit cell even within the same band of eigenfunctions as is illustrated in Figure 9a.

At last, tuning the intra-cell delta-function potentials $\beta_q, q = 1, \dots, Q$, provides one more tool for controlling and varying the profile and energy spectrum of the multi-qubit

trap eigenfunctions $\psi_{s,p}(x)$. As is illustrated in Figure 9b, it affects the eigenfunctions in the first, $s = 0$, band stronger than the eigenfunctions in the second, $s = 1$, band. Thus, it is an efficient tool for controlling the intra-qubit properties, in particular, the qubit energy splittings $\delta E_q, q = 1, \dots, Q$.

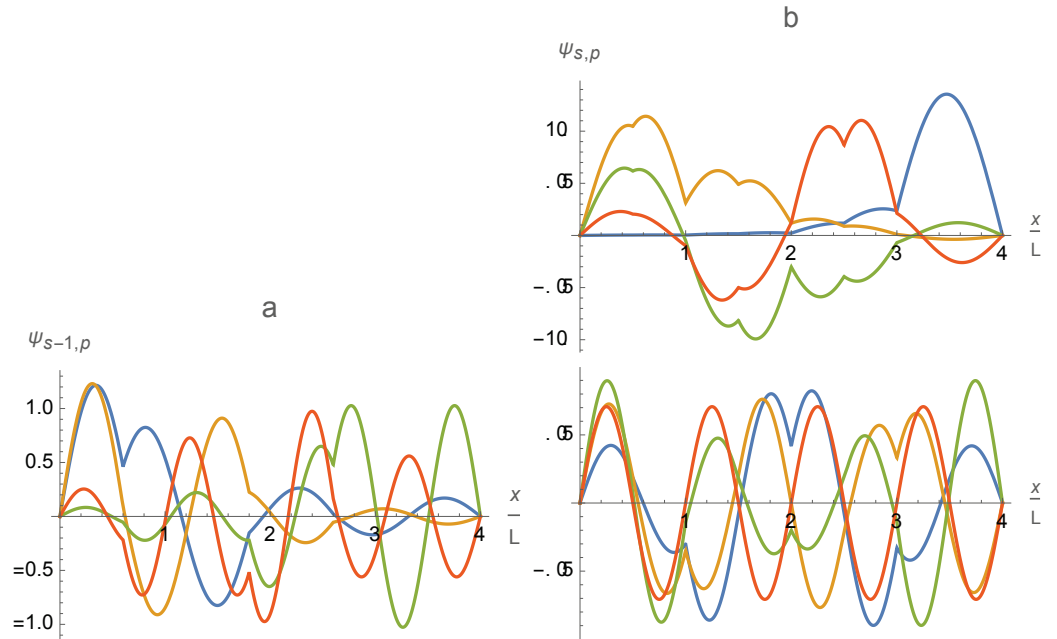


Figure 9. The eigenfunctions $\psi_{s,p}(x)$ ($p = 1$ in blue, $p = 2$ in yellow, $p = 3$ in green, $p = 4$ in red) for the chain of $Q = 4$ single-qubit cells with a zero background potential, $U_j = 0, j = 1, \dots, 8$. The inter-cell walls are the delta-function potentials of the same magnitude, $\alpha_q mL/\hbar^2 = 9, q = 1, 2, 3, 4$. (a) The eigenfunctions $\psi_{s=1,p}(x)$ of the second band in the case of single-qubit cells of different lengths $L_q/L = 0.6, 1.2, 0.8, 1.4$. Two mechanisms of the general asymptotic rule (stated in Section 4.4) for a transition from the sinusoidal wave eigenfunctions (25) of a uniform box trap to the eigenfunctions of a multi-qubit trap are clearly visible. Contrary to the case of identical single-qubit cells in Figure 7(left), now the numbers of zeroes per a single-qubit cell in the eigenfunctions of the second band $s = 1$ are not all equal to unity but could be also zero, two or even three and different in the different cells. (b) The eigenfunctions $\psi_{s,p}(x)$ of the first ($s = 0$) and second ($s = 1$) bands in the case of different delta-function potentials $\beta_q mL/\hbar^2 = 1, 2, 3, 0$ at the center of single-qubit cells of equal length L . A comparison with the case of identical single-qubit cells in Figure 7(left) shows that now, the eigenfunctions of the band $s = 0$ are notably modified while the eigenfunctions of the band $s = 1$ stay almost intact.

One other possibility when constructing these multi-qubit traps is to place the inter-cell and intra-cell delta potentials in such a way as to break the symmetry between each qubit cell. This can act as yet another knob by which to control the diversity of the system along with the heights of the delta potentials and the modulation of the background potential. Delta-function potentials can be moved individually or following some group pattern. The position of the potential can be represented by a number from 0 to 1, essentially what percentage of the cell is traveled starting from the center of the cell before dropping down the delta-function potential. Suppose the intra-cell barriers are shifted leftwards on the left side of the four-qubit trap and an equal distance rightward on the right side. The effect of this shifting on the energy levels of a four-qubit trap can be seen in Figure 10. It is not symmetric and can be understood by observing where the delta-function potentials are modulating the unperturbed wave functions for each energy level. For example, take the ninth energy level seen in cyan in Figure 10. This energy level is maximized around 0.18 and minimized around 0.36. When the positions of the delta-function potentials are overlaid on the unperturbed ninth eigenfunction, it becomes clear why this is the case. At

the maximum effect, the delta-function potentials are positioned near the maxima of the unperturbed function, while at the minimum effect, the delta potentials are placed instead near the zeros of the unperturbed function, as seen in Figure 11. The latter figure illustrates also the two mechanisms of a wave-function perturbation stated in Section 4.4. In the former case, the first mechanism, namely, digging a dip in a function profile underneath the delta-function potential, takes place. In the latter case, the second mechanism, namely, dragging the nearest node underneath the delta-function potential, takes place. The respective effects on the energy of the eigenfunctions are very different as is shown in Figure 11. Note that the digging effect of the central inter-cell delta-function potential is the same in both cases.

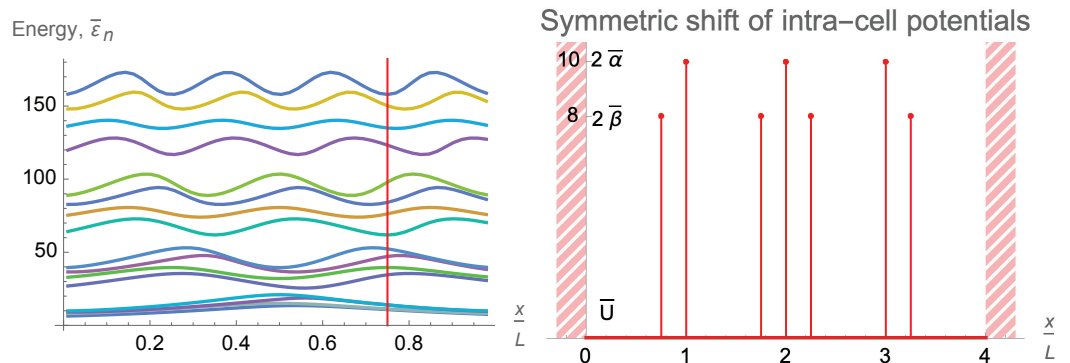


Figure 10. Left: The single-particle energy levels for a four-qubit trap with intra-cell potential barriers of strength $\bar{\beta} = 4$ and inter-cell barriers of strength $\bar{\alpha} = 5$ for various symmetrically shifted positions of the intra-cell potentials. Right: The specific trapping potential for the position marked by the red line. Note the symmetry about the center of the trap.

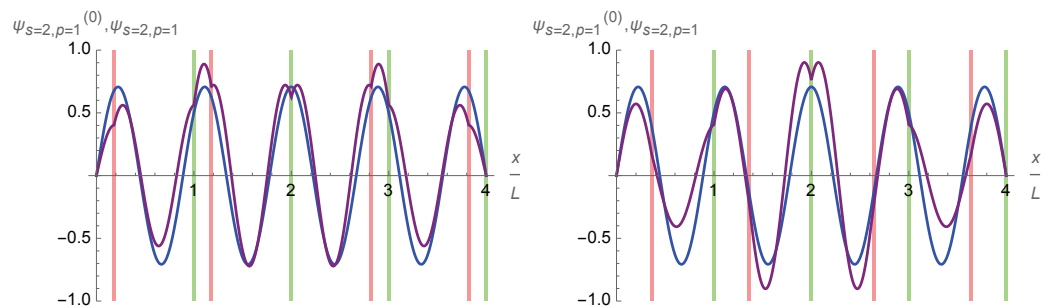


Figure 11. The unperturbed and perturbed ninth-energy-level eigenfunction, $\psi_{s=2,p=1}^{(0)}$ (in blue) and $\psi_{s=2,p=1}$ (in purple), in the case of the intra-cell delta-function potentials at position 0.18 (left) and 0.36 (right) marked by red vertical lines. Note how on the right, the intra-cell delta-function potentials act on the eigenfunction near zeros, reducing their effect on the eigenenergy, while on the left, the intra-cell delta-function potentials act on the eigenfunction near two peaks, having a very large impact on the eigenenergy $\bar{\epsilon}_{n=9}$ shown in Figure 10.

5. Two-Dimensional Multi-Qubit Trap: Single-Particle Eigen Functions and Energies

Here, we describe a simple 2D model of the multi-qubit trap formed by a potential, $U(x, y) = U_x(x) + U_y(y)$, which is the sum of the two 1D potentials considered above. In this case, the solution to the 2D single-particle Schrödinger Equation (8) is reduced, via a factorization, to the solutions to the 1D Schrödinger equation described above.

As a generic example, let us consider the 2D symmetric four-qubit trap, in which both the potential in the x and y plane are exactly the same. Both will share the same inter-cell and intra-cell delta-function potential strengths. As the energy levels are known for the 1D case (see, for example, Figure 6), constructing the energy levels for the 2D case is a simple task, requiring only for the individual energies to be added in every possible combination. A visualization of the energy levels created by combining the first nine energy levels in each

dimension is shown in Figure 12 for the symmetric 2D (4×4)-qubit trap. This fully covers the first miniband of each dimension plus the first energy level in the second miniband.

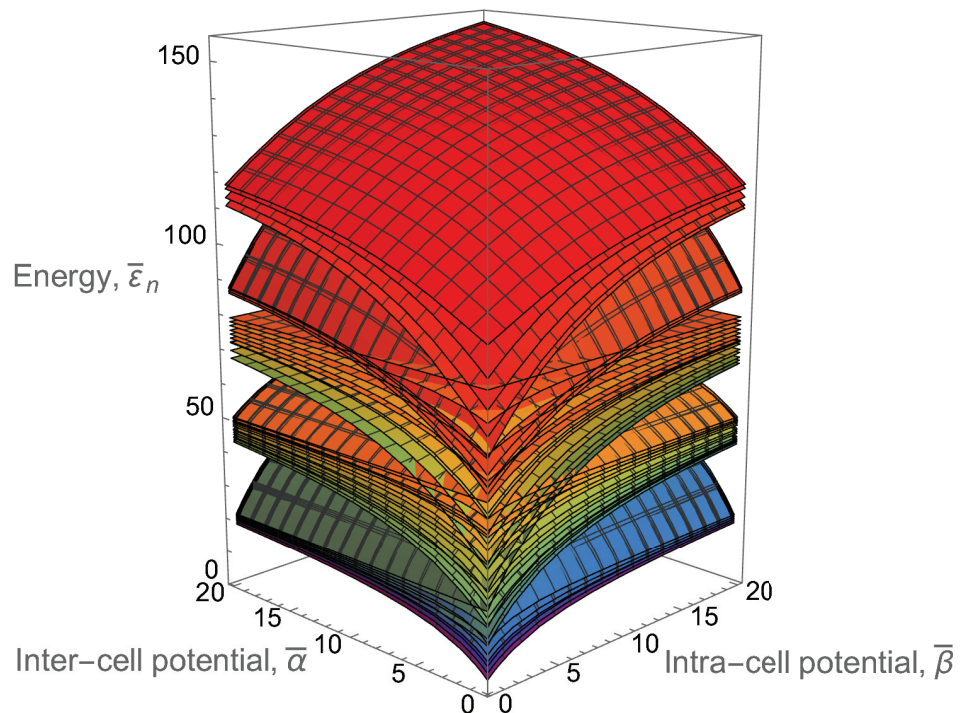


Figure 12. The first 80 energy level combinations comprising the eigenenergies $\{\bar{\epsilon}_n\}$ for the symmetric 2D (4×4)-qubit trap in the absence of a background potential ($U_j = 0 \forall j$). Compare this plot against its 1D 4-qubit counterpart in Figure 6 and note how there are energy crossings near the origin of the graph, and that there is a slight asymmetry in the inter-cell and intra-cell directions.

This structure is interesting. Firstly, let us understand why there is a noticeable asymmetry between the effects of the inter-cell and intra-cell potentials. We see that for the two different axes, the first set of energy levels that are bunched together contains either nine levels for intra-cell potentials or 16 levels for inter-cell potentials. This comes from the fact that for the 1D case of a large intra-cell potential with little inter-cell potential, there is a splitting of the energy levels in the first miniband into two parts, one containing the first three with the other containing the remaining five. These two sub-minibands have a significant enough energy gap between them so that when the 2D plot is created, we see the formation of three sub-minibands, one containing combinations consisting of only energy levels in the first half of the sub-miniband, one containing the crossover terms, and the final piece containing the combinations consisting of only energy levels in the second half of the sub-miniband. Because there are nine possible combinations for the first half, we see nine energy levels in the first band of the plot. However, for the case of the large inter-cell potential, we instead see a 4–4 split of the energy levels in the miniband rather than the 3–5 split in the intra-cell case. Thus, 16 possible combinations of the energy levels in the first half of the miniband.

To understand why there is a difference in splitting of the energy levels depending on if the inter-cell or intra-cell potentials dominate the trap, we need only to look at the fourth energy level and how it is affected by each of the two different types of traps. The easiest to understand is the inter-cell potential dominant traps. In this case, the fourth eigenfunction is almost totally unperturbed by the delta-function potentials, as its natural nodes are already placed at the locations of the inter-cell potential barriers, while the lower three energy levels are pulled up toward the fourth. Likewise, the fifth, sixth, and seventh energy levels are bought up to the totally unperturbed eighth, leading to the 4–4 split structure we observe. However, in the case of the intra-cell potentials dominating the trap,

the natural nodes of the fourth energy level wavefunction are placed directly between two of the intra-cell potential walls. This, as described in Section 4.4, leads to a situation where new nodes must be created to accommodate the large potentials. These new nodes drastically increase the average derivative, bringing the energy level much further above the third energy level below it, whose eigenfunction is able to shift its nodes to fall under the existing large potentials without much effect to its energy level. This effect is illustrated in Figure 13.

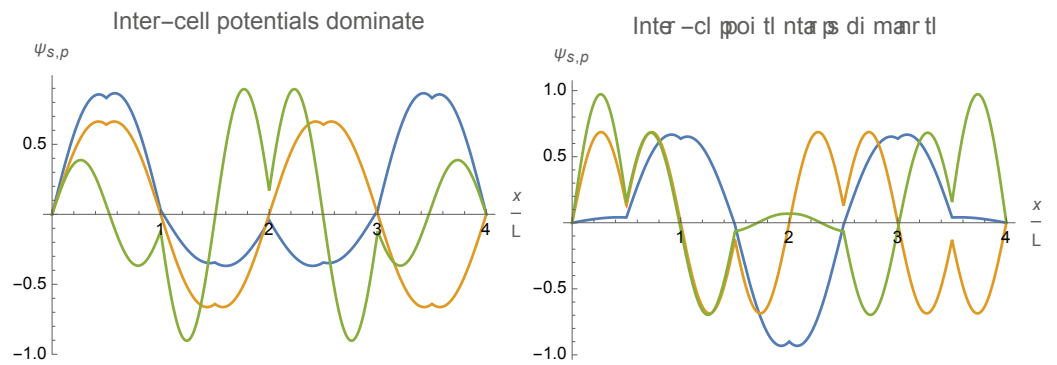


Figure 13. The eigenfunctions $\{\psi_n\}$ for the third (blue), fourth (yellow), and fifth (green) energy levels in the 1D multi-qubit trap of $Q = 4$ identical single-qubit cells for the inter-cell (**left**) and intra-cell (**right**) dominant delta-function potentials $\alpha_q = \alpha$ and $\beta_q = \beta \forall q$. The dominant delta-function potential has a magnitude of $31\hbar^2/(mL)$ while the non-dominant potential has a magnitude of $1\hbar^2/(mL)$. Note how in the inter-cell-potential dominant case, the fourth eigenfunction is most similar to the third, while in the intra-cell-potential dominant case, the fourth eigenfunction is most similar to the fifth.

The last notable aspect of Figure 12 is the crossing of energy levels that can be seen near this origin. This behavior arises from the fact that while initially, the energy levels are approximately evenly spread, once the potentials start ramping up, there are some significant gaps created in the miniband structure. Thus, for low potentials, the energy level created by combining the first and fifth energy levels may be lower than that created by the fourth energy level combined with itself. However, once the fifth energy level is drastically raised by the introduction of the potentials, the combination of the first and fifth levels will increase its total energy above that of the double fourth, at least for the inter-cell dominant case where the fourth energy level is mostly unperturbed. This sort of behavior is only seen near the origin where the energy levels change drastically with the introduction of the delta-function potentials, as once the overall structure begins to form, there is not a significant enough change to create more crossings.

One can also plot the occupation probability distribution for a specific single-particle state in a 2D multi-qubit trap. The spatial profile of the occupation distribution for the single-particle ground state is illustrated in Figure 14. This figure also demonstrates an important property of the multi-qubit trap: namely, that the parameters for the background potential and inter-/intra-cell walls can be tuned to achieve a desirable, in particular, relatively uniform distribution of occupation probability over the entire trap.

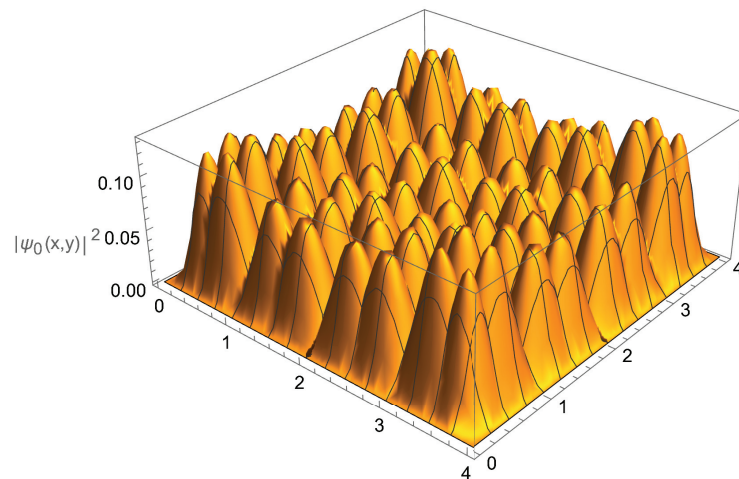


Figure 14. Occupation probability distribution in the single-particle ground state, $|\psi_0(x,y)|^2$, for the 2D (4×4) -qubit trap with delta-function potentials $\alpha_q = 4\hbar^2/(mL)$, $\beta_q = 2\hbar^2/(mL) \forall q$ and a central flat pedestal potential $U = 8\hbar^2/(mL^2)$ ranging from $0.5 < x < 3.5$ and $0.5 < y < 3.5$ as seen in Figure 2.

6. Controlling the Condensate in the Multi-Qubit Trap: The Gross–Pitaevskii Equation

In real interacting gases, the result for the macroscopic condensate wave function given by the Gross–Pitaevskii Equation (9) significantly deviates from the single-particle ground state of the linear Schrödinger Equation (8). A difference between the Schrödinger and Gross–Pitaevskii equations originates due to a collective effect of interparticle interactions described by the nonlinear self-interaction term $gN_0|\phi_0|^2\phi_0$ in Equation (9). The main features of the condensate are correctly described already in an approximation neglecting the interaction with the noncondensed fraction (the term $2gn_{ex}\phi_0$ in Equation (9)) and assuming $N_0 \approx N$. For simplicity’s sake, we adopt the above approximation and limit discussion to 1D and 2D models of the multi-qubit BEC trap.

A 1D model implies a situation when atoms are tightly confined in the transverse to the x axis directions, for example, in a cylinder of length L , (y,z) -cross-section area l_\perp^2 with a small transverse size $l_\perp \ll L$, and volume $V = Ll_\perp^2$. Then, in view of the normalization condition $\int_V |\phi_0^2| d^3\mathbf{r} = 1$ and averaging $\phi_0^2(x,y,z)$ over the small y,z cross-section, the 3D condensate wave function $\phi_0(\mathbf{r})$ can be efficiently replaced by a 1D function $\phi_0(x)/l_\perp$ that corresponds to a rescaled interaction parameter $g_1 = g/l_\perp^2$. Note that the mean-field condition (12) for validity of the 1D model of the Gross–Pitaevskii Equation (9) remains the same as in the usual 3D case,

$$8\pi a \ll \left(\frac{V}{N}\right)^{1/3}, \quad \frac{\xi}{d} = \frac{1}{\sqrt{8\pi a(N/V)^{1/3}}}, \quad g = \frac{4\pi\hbar^2 a}{m} \quad (3D \text{ mean – field regime}), \quad (27)$$

only if the average distance between atoms is $d = (Ll_\perp^2/N)^{1/3}$, i.e., the volume density of atoms in the trap is large enough: $N/(Ll_\perp^2) > 1/l_\perp^3$. Otherwise, the average distance between atoms becomes equal to $d = L/N$ and the mean-field validity condition (12) imposes a requirement on the scattering length

$$8\pi a \ll Nl_\perp^2/L, \quad \frac{\xi}{d} = \sqrt{\frac{Nl_\perp^2}{8\pi aL}}, \quad g_1 = \frac{g}{l_\perp^2} \quad \text{if } Nl_\perp < L \quad (1D \text{ mean – field regime}), \quad (28)$$

which is getting more stringent with a decreasing number of atoms. It is worth noting that when $aN \gg L$, the system locally retains the original 3D character despite its 1D geometrical appearance, $L \gg l_\perp$. Only in the opposite case, when $aN \ll L$, the system approaches the ground state in the transverse directions and enters the so-called 1D mean-field regime (see [35] and references therein). In the low-density limit $8\pi a \gg Nl_\perp^2/L$, which corresponds to the strong-coupling 1D limit $g_1 \equiv g/l_\perp^2 \rightarrow \infty$ and is opposite to (28), the

mean-field approach fails and the system becomes the so-called Tonks–Girardeau gas of impenetrable bosons.

Similarly, a 2D model implies a situation when atoms are tightly confined just in one, axial direction, say, along the z -axis within a small linear dimension l_z , while the cross-section area of the trap of volume $V = LL'l_z$ is relatively large, $LL' \gg l_z^2$. In this case, the 3D condensate wave function $\phi_0(\mathbf{r})$ can be efficiently replaced by a 2D function $\phi_0(x, y)/\sqrt{l_z}$ that corresponds to a rescaled interaction parameter $g_2 = g/l_z$. The mean-field condition (12) for validity of the 2D model of the Gross–Pitaevskii Equation (9) retains the usual 3D form (27) only if the average distance between atoms is $d = (LL'l_z/N)^{1/3}$, i.e., the volume density of atoms in the trap is large enough: $N/(LL'l_z) \gg 1/l_z^3$. Otherwise, that is when $Nl_z^2 < LL'$, the average distance between atoms becomes equal to $d = \sqrt{LL'/N}$ and the mean-field validity condition (12) is reduced to the requirement on the scattering length

$$8\pi a \ll l_z, \quad \frac{\xi}{d} = \sqrt{\frac{l_z}{8\pi a}}, \quad g_2 = \frac{g}{l_z} \quad \text{if } Nl_z^2 < LL' \quad (\text{2D mean - field regime}), \quad (29)$$

which is independent on the number of trapped atoms. Again, at extremely low densities, when $|\ln(Nl_z^2/LL')| > l_z/a$, the mean-field approach fails, the interaction constant $g_2 = g/l_z$ should be replaced by the density-dependent parameter $\tilde{g}_2 = 4\pi\hbar^2/[m|\ln(Nl_z^2/LL')|]$, and the system enters the regime analogous to the Tonks–Girardeau 1D regime [35].

6.1. Single-Qubit Trap: 1D Analytical and 2D Numerical Solutions to the Gross–Piraevskii Equation

The solution to the 1D nonlinear Schrödinger, that is Gross–Piraevskii, Equation (9) in the stated simple model can be found similar to the solution to the 1D linear Schrödinger Equation (15), described above, if one employs the elliptic Jacobi function $\text{sn}(x|p)$ instead of the exponential function $\exp(x)$. For simplicity’s sake, we adopt the Bogoliubov approximation at very low temperature $T \rightarrow 0$ assuming that practically all atoms are condensed, $N_0 \approx N$, and the effect of the noncondensed atoms on the condensate is negligible.

Then, the condensate wave function in a box trap with zero potential, $U(x) = 0$, and Dirichlet (zero) boundary conditions is given by the elliptic Jacobi function,

$$\phi_0(x) = \sqrt{\frac{pK(p)}{K(p) - E(p)}} \frac{\text{sn}(2K(p)\frac{x}{L}|p)}{\sqrt{L}}, \quad \frac{L}{\xi} = \sqrt{8K(p)(K(p) - E(p))}. \quad (30)$$

It varies from the half-period sine to an almost constant function (quickly decreasing to zero just in the narrow boundary regions) with the interaction g increasing from zero to the larger values. The characteristic scale of the condensate is determined by the healing length (11). The solution includes complete elliptic integrals of the first and second kinds:

$$K(p) = \int_0^{\pi/2} (1 - p \sin^2 \theta)^{-1/2} d\theta, \quad E(p) = \int_0^{\pi/2} (1 - p \sin^2 \theta)^{1/2} d\theta, \quad (31)$$

According to Equation (30), the range of the parameter p is from 0 to 1. The chemical potential is determined by the normalization condition $\int |\phi_0(x)|^2 dx = 1$ as follows

$$\mu = 4(1 + p)K^2(p)\hbar^2/(2mL^2). \quad (32)$$

The analytical solution in Equation (30) fully describes the effect of the interparticle interaction on the condensate profile in each single-qubit cell if the inter-cell potential walls are infinitely high, the background potentials are the same in both halves of each cell and the intra-cell delta-function potentials are absent. It is illustrated in Figure 15 for the single-qubit cell. When the healing length is much longer than the cell’s length, $\xi \gg L$, that is, the interaction is very weak and the gas is almost ideal, the parameter p is very close to zero. As a result, the condensate profile in each cell is very close to the ground-state solution to the single-particle Schrödinger Equation (15), that is, a half of the sine function,

$\phi_0(x) = \sqrt{2/L} \sin(\pi x/L)$, and $\mu \approx \pi^2 \hbar^2 / (2mL^2)$. In the opposite case of a very short healing length, $\zeta \ll L$, the parameter p approaches 1 and the strong interparticle interaction makes the condensate profile more flat and spread over the entire cell, except for narrow boundary layers of thickness ζ near the walls. Obviously, a similar situation takes place in each half of the single-qubit cells if the intra-cell potential walls are also infinitely high.

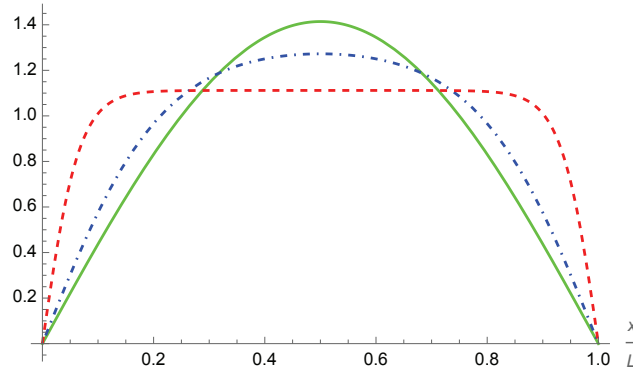


Figure 15. The interparticle interaction makes the condensate more uniform and spread over the entire single-qubit trap as is revealed by the analytical solution (30) to the Gross–Pitaevskii Equation (9) in the case of infinitely high inter-cell potential walls and zero background and intra-cell potentials: $L/\zeta = 0$ (an ideal gas—solid green curve, $p = 0$), $L/\zeta = 5$ (a moderate interaction—dot-dashed blue curve, $p \approx 0.86$), $L/\zeta = 20$ (a strong interaction—dashed red curve, $p \approx 0.999996$).

The effect of the repulsive interparticle interaction on the condensate profile in the single-qubit box cell in 2D is shown in Figure 16. In the case of an ideal gas, the atoms condense into the ground state $\psi_0(x, y) = \frac{2}{L} \sin \frac{\pi x}{L} \times \sin \frac{\pi y}{L}$ of the single-particle Schrödinger Equation (15), as shown in Figure 16a. In the case of an interacting gas, the condensate profile $\phi_0(x, y)$ is given by the numerical solution to the 2D Gross–Pitaevskii Equation (9), as shown in Figure 16b. Comparison of the two plots clearly shows that the particle repulsion flattens the peak of the ground-state wave function and forces the condensate to spread over the entire single-qubit cell. Just the boundary layers of a healing-length thickness remain unoccupied by the condensate.

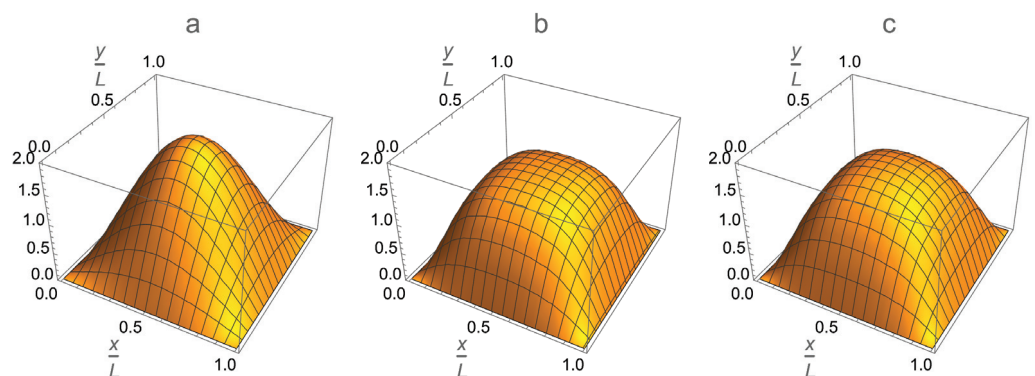


Figure 16. Two-dimensional (2D) single-qubit BEC trap with zero background and intra-cell potentials. The interparticle interaction makes the condensate more uniform and spread over the entire single-qubit cell as is revealed by comparing (a) the ground-state wave function $\psi_0(x, y) = \frac{2}{L} \sin \frac{\pi x}{L} \times \sin \frac{\pi y}{L}$ given by the single-particle Schrödinger Equation (15) in the absence of interaction against (b) the condensate wave function $\phi_0(x, y)$ in the presence of interaction, $L/\zeta = 5$, computed via an exact numerical solution to the Gross–Pitaevskii Equation (9). The plot (c) is an approximation of the latter condensate wave function $\phi_0(x, y)$ via a factorization (33) of the exact analytical solutions for the 1D box trap, Equation (30), along the x and y axes. All three plots present the dimensionless condensate wave function of the unity norm.

These effects can be approximately described analytically by a product of the exact analytical solution (30) to the Gross–Pitaevskii equation in the 1D box trap along the x -axis and the similar solution along the y -axis,

$$\phi_0(x, y) \approx \frac{pK(p)}{[K(p) - E(p)]L} \operatorname{sn}\left(2K(p)\frac{x}{L}\middle|p\right)\operatorname{sn}\left(2K(p)\frac{y}{L}\middle|p\right). \quad (33)$$

Such a 1D-factorization approximation is shown in Figure 16c. It takes into account the interparticle interaction only partially via its separate manifestations along the x and y dimensions. Comparing Figure 16b and Figure 16c, we conclude that the above approximation slightly underestimates the effect of 2D nonlinear diffusion of the condensate due to the self-interaction $gN_0|\phi_0(x, y)|^2\phi_0(x, y)$, which is opposite to the phenomenon of self-focusing of an intensive laser light beam in a nonlinear medium. Nevertheless, the 1D-factorization approximation represents the effect of the interparticle interaction on the condensate profile in a box trap in a qualitatively correct fashion.

6.2. Condensate Wave Function vs. Single-Particle Ground State in a Multi-Qubit Trap

For a nontrivial multi-qubit BEC trap, due to the presence of the trapping potential $U(x) \neq 0$, as shown in Equation (13), the Gross–Pitaevskii Equation (9), that is, the nonlinear Schrödinger equation, needs to be solved numerically, for instance, by the method of an imaginary-time evolution (see, e.g., [87]). It is illustrated in Figures 17 and 18 for the case of a four-qubit 1D and 2D trap, respectively.

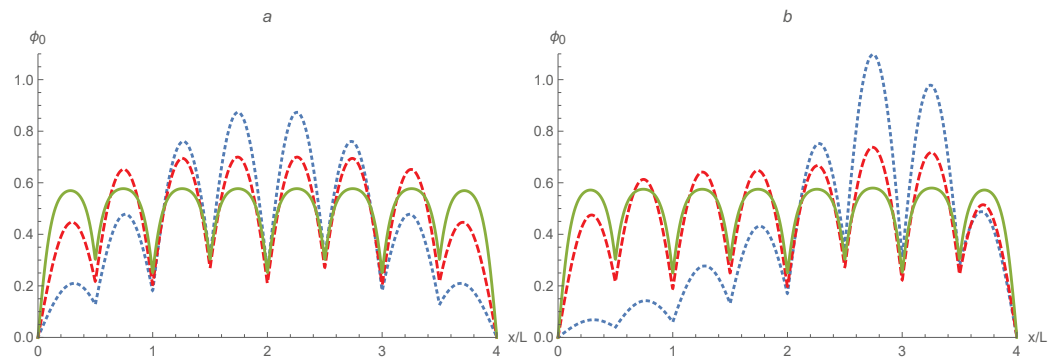


Figure 17. The ground-state wave function according to the single-particle Schrödinger Equation (15) (blue dotted curve) and the corresponding condensate wave function ϕ_0 according to the Gross–Pitaevskii Equation (9) in the presence of the moderate, $\frac{L}{\xi} = 2$, (red dashed curve) and strong, $\frac{L}{\xi} = 10$, (green solid curve) interaction in the case of (a) symmetric ($U(x) = 0$) and (b) asymmetric ($U(x) = (4\hbar^2/(mL^2))[\theta(x - 0.5L) - \theta(x - 2.5L)]$) 1D four-qubit trap; $\alpha_j = 1.5\beta_j = 16\hbar^2/(mL) \forall j$.

As a result of the interparticle repulsion, the condensate tends to spread more uniformly over all single-qubit cells. This tendency works against condensate fragmentation [78] and in favor of the formation of a common condensate occupying the entire BEC trap. Moreover, with increasing interaction, the regions of low condensate occupation near the inter- and intra-cell potential walls begin shrinking as well. Both of the above effects significantly increase the number of Bogoliubov-coupled excited states and magnitude of their Bogoliubov couplings in Equation (10) that favors manifestation of the #P-hardness of the atomic boson sampling as is explained in Section 2.

It is worth noting that such a considerable expansion of the condensate shown in Figures 17 and 18 is provided by means of the interparticle interaction alone, without employment of the background potential, which also allows one to control the condensate profile in a similar direction via restructuring the ground-state wave function as is shown in Figures 8 and 14.

Moreover, if the background potential makes the trap asymmetrical, the increasing repulsive interaction tends to restore the trap’s symmetry by converting an asymmetrical

single-particle ground state into a more symmetrical condensate wave function. Such evolution of the condensate in the asymmetrical trap is illustrated in Figure 17b and should be compared against the condensate evolution in the symmetrical trap shown in Figure 17a. Clearly, a strong interparticle interaction makes the condensate profiles in both traps almost indistinguishable, while the profiles of the ground state in these traps in the absence of interaction are very different.

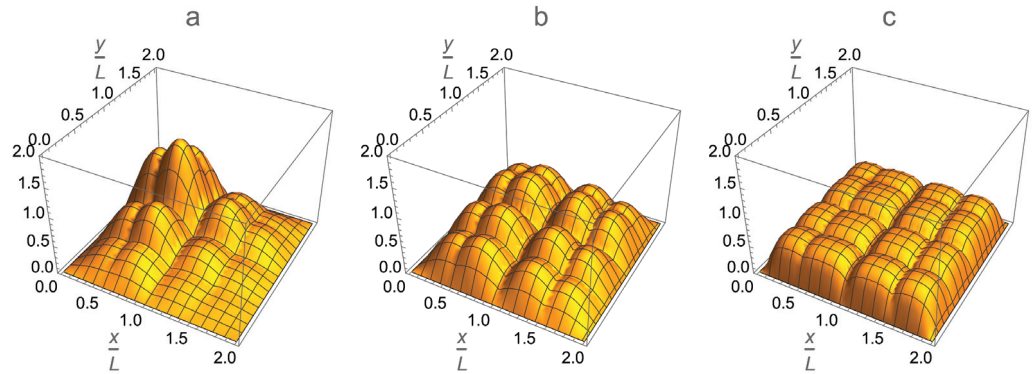


Figure 18. Two-dimensional (2D) (2×2) -qubit BEC trap: (a) The ground-state wave function ψ_0 according to the single-particle Schrödinger Equation (15) in the absence of interaction as well as the condensate wave function ϕ_0 according to the Gross–Pitaevskii Equation (9) in the presence of (b) moderate, $\frac{L}{\xi} = 5$, and (c) strong, $\frac{L}{\xi} = 20$, interaction; $\alpha_j = 8\hbar^2/(mL)$, $\alpha'_j = 6\hbar^2/(mL)$, $\beta_1 = \beta'_1 = 4\hbar^2/(mL)$, $\beta_2 = \beta'_2 = 2\hbar^2/(mL)$, $U_j = U'_j = 0 \forall j$ (see Equation (14)).

7. Controlling Multimode Squeezing of Bogoliubov Transform via Bogoliubov Couplings

In equilibrium, the statistics of the many-body system of atoms in the BEC trap is determined by the independent fluctuations of quasiparticles which form the eigenstates of the Bogoliubov Hamiltonian with the eigenenergies $\{E_j\}$ and have the Bose–Einstein occupation number statistics with an average occupation number $\bar{n}_j = (e^{(E_j - \mu)/T} - 1)^{-1}$. The two-component quasiparticle wave function $\{u_j, v_j\}$ determine the excited-particle field operator (3) and obeys the Bogoliubov–de Gennes equations:

$$\begin{aligned} \hat{L}u_j + gN_0\phi_0^2(\mathbf{r})v_j &= +E_ju_j, \\ \hat{L}v_j + gN_0(\phi_0^*)^2(\mathbf{r})u_j &= -E_jv_j, \end{aligned} \tag{34}$$

where

$$\hat{L} \equiv -\frac{\hbar^2\Delta}{2m} + U(\mathbf{r}) + 2g(N_0|\phi_0(\mathbf{r})|^2 + n_{\text{ex}}(\mathbf{r})) - \mu; \quad n_{\text{ex}} = \sum_j \left[|v_j(\mathbf{r})|^2 + \frac{|u_j(\mathbf{r})|^2 + |v_j(\mathbf{r})|^2}{\exp(E_j/T) - 1} \right].$$

In essence, the Bogoliubov–de Gennes equations express the fact of diagonalization of the Hamiltonian (4) by the Bogoliubov transformation to quasiparticle creation/annihilation operators, as stated in the matrix form in Equation (6), in the form of differential equations for the coefficients $\{u_j, v_j\}$ in the expansion of the field operator (3) via the quasiparticle operators. The wave functions are normalized to unity: $\int_V |\phi_0|^2 d^3\mathbf{r} = 1$, $\int_V (|u_j|^2 - |v_j|^2) d^3\mathbf{r} = 1$; $j = 1, 2, \dots$. For simplicity’s sake, hereinafter, we assume that all wave functions ϕ_0, u_j, v_j are real valued. Below, we again neglect by temperature-dependent, Popov’s corrections, that is, skip the contribution due to the noncondensate density n_{ex} and assume $N_0 \approx N$.

The matrix R of the Bogoliubov transformation (2) can be found from the equation

$$\begin{pmatrix} \hat{\mathbf{a}}^\dagger \\ \hat{\mathbf{a}} \end{pmatrix} = R \begin{pmatrix} \hat{\mathbf{b}}^\dagger \\ \hat{\mathbf{b}} \end{pmatrix} = \begin{bmatrix} A & B^* \\ B & A^* \end{bmatrix} \begin{pmatrix} \hat{\mathbf{b}}^\dagger \\ \hat{\mathbf{b}} \end{pmatrix} \tag{35}$$

that relates creation and annihilation operators \hat{a}^\dagger, \hat{a} describing bare particles to the operators \hat{b}^\dagger, \hat{b} describing quasiparticles as per Equation (3). Since the Bogoliubov transformation (2) leaves the canonical Bose commutation relations invariant, it obeys the symplectic property

$$R\Omega R^T = \Omega, \quad \Omega = \begin{bmatrix} 0 & \mathbb{1} \\ -\mathbb{1} & 0 \end{bmatrix}. \tag{36}$$

Another, equivalent to (35), representation of the Bogoliubov transformation (2) can be written in terms of the wave functions, rather than the operators, determining the particle field operator in Equation (3):

$$\begin{pmatrix} \phi \\ 0 \end{pmatrix} = R \begin{pmatrix} \mathbf{u} \\ -\mathbf{v}^* \end{pmatrix} \equiv \begin{bmatrix} A & B^* \\ B & A^* \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ -\mathbf{v}^* \end{pmatrix}, \tag{37}$$

$$R^T \begin{pmatrix} 0 \\ \phi \end{pmatrix} \equiv \begin{bmatrix} A^T & B^T \\ B^\dagger & A^\dagger \end{bmatrix} \begin{pmatrix} 0 \\ \phi \end{pmatrix} = \begin{pmatrix} \mathbf{v}^* \\ \mathbf{u} \end{pmatrix}. \tag{38}$$

The column-vectors $\phi = (\phi_1, \phi_2, \dots)^T$ and $\mathbf{u} = (u_1, u_2, \dots)^T$, $\mathbf{v} = (v_1, v_2, \dots)^T$ are composed of the excited states $\{\phi_k\}$ and quasiparticle wave functions $\{u_j\}, \{v_j\}$, respectively.

Projecting Equation (38) onto a set of the orthonormal excited states $\{\phi_k | k = 1, 2, \dots\}$ which are also orthogonal to the condensate wave function ϕ_0 , we obtain the explicit formulae for the entries of the Bogoliubov block matrices $A = (A_{kj})$ and $B = (B_{kj})$,

$$A_{kj} = \int u_j^*(\mathbf{r})\phi_k(\mathbf{r})d^3\mathbf{r}, \quad B_{kj} = \int v_j^*(\mathbf{r})\phi_k^*(\mathbf{r})d^3\mathbf{r}, \tag{39}$$

as overlapping integrals between those bare-particle wave functions and the quasiparticle wave functions given by the solution to the Bogoliubov–De Gennes equations (34).

The Bogoliubov matrix R can be expressed explicitly also via the Bogoliubov couplings in Equation (10) by means of a pure algebraic diagonalization of the Bogoliubov Hamiltonian in the sense of the matrix Equation (6). Indeed, in any basis $\{\phi_k | k = 1, 2, \dots\}$ of excited states, orthogonal to the condensate wave function and constituting the excitation field operator $\hat{\psi}_{\text{ex}} = \sum_{k \neq 0} \phi_k \hat{a}_k$ as in Equation (3), the blocks of the Hamiltonian matrix in Equation (5) are explicitly given by the Bogoliubov couplings in Equation (10) as follows

$$K = (\varepsilon_{kk'} - \mu\delta_{k,k'} + 2\Delta_{kk'}), \quad \tilde{K} = \frac{1}{2}(\tilde{\Delta}_{kk'}), \quad H = \begin{bmatrix} \tilde{K} & K \\ K^* & \tilde{K}^* \end{bmatrix}. \tag{40}$$

Here, $\varepsilon_{kk'} = \langle \phi_k | \hat{\varepsilon} | \phi_{k'} \rangle$ is the matrix of the single-particle energy operator $\hat{\varepsilon} = -\hbar^2\Delta/(2m) + U(\mathbf{r})$ which constitutes the single-particle Schrödinger Equation (8). In particular, the basis $\{\phi_k | k = 1, 2, \dots\}$ can be constructed out of the excited-state eigenfunctions $\{\psi_n | n = 1, 2, \dots\}$ of the Schrödinger Equation (8) by means of the standard Gram–Schmidt orthonormalization starting from making these functions orthogonal to the condensate wave function ϕ_0 . Then, by means of the symplectic property (36), Equation (6) determining the Bogoliubov transformation can be rewritten as the following equation

$$\Omega HR = R \begin{bmatrix} E & 0 \\ 0 & -E \end{bmatrix}, \quad E = \text{diag}\{E_j | j = 1, 2, \dots\}. \tag{41}$$

It states that the j -th column of the Bogoliubov matrix, $\mathbf{R}_j = \{A_{1j}, A_{2j}, \dots, B_{1j}, B_{2j}, \dots\}^T$, is the eigenvector of the matrix ΩH corresponding to the quasiparticle eigenenergy E_j , that is

$$\Omega H \mathbf{R}_j = E_j \mathbf{R}_j, \quad \Omega H = \begin{bmatrix} K^* & \tilde{K}^* \\ -\tilde{K} & -K \end{bmatrix}. \tag{42}$$

(There is also the nonphysical eigenvector counterpart $\mathbf{R}_j^{(-)} = \{B_{1j}^*, B_{2j}^*, \dots, A_{1j}^*, A_{2j}^*, \dots\}^T$ corresponding to the negative eigenenergy $-E_j < 0$.)

After calculating the Bogoliubov transformation matrix R as per Equation (6), one can find the multimode squeezing parameters [1,68–73] from Equation (2). Quantum statistics of the many-body fluctuations in a BEC trap and, in particular, the computational complexity of the atomic boson sampling are determined by two fundamental sets of eigenvectors and eigenvalues associated with the diagonalization of (a) the squeezing matrix and (b) the Bogoliubov Hamiltonian, as is explained in Sections 1 and 2. Both of those sets of eigenvectors and eigenvalues are determined by the Bogoliubov couplings (10) via the Bogoliubov transformation matrix R . Thus, the key problem is to calculate the Bogoliubov couplings and understand how many of them can be essentially nonzero and controllable in a wide range within the multi-qubit BEC trap suggested and described in this paper.

Knowing the condensate wave function from the solution to the Gross–Pitaevskii equation outlined in Section 6 and choosing the bare-particle excited states, for example, as the purely harmonic, sine functions or the solutions to the single-particle Schrödinger Equation (8) (see Sections 4 and 5), made orthogonal to the condensate and each other via the standard Gram–Schmidt orthonormalization, it is straightforward to calculate the integrals constituting the Bogoliubov couplings (10) and analyze their set for the multi-qubit BEC trap. The related numerical results are illustrated for the cases of symmetrical and asymmetrical 1D four-qubit traps in the plots shown in Figures 19 and 20, respectively.

First, comparing Figure 20 against Figure 19 makes it clear that the trap asymmetry greatly enlarges the number of Bogoliubov-coupled bare-particle excited states. Indeed, in the asymmetrical trap, the essentially nonzero couplings spread much further from the main diagonal of the Bogoliubov-coupling matrix $\Delta_{k,k'}$ than in the symmetrical trap where only narrow lanes of entries around the main diagonal and anti diagonal are essentially nonzero. In addition, the degeneracy of zero coupling between the bare-particle excited states of exactly odd and even spatial parity in the symmetrical trap (Figure 19), that results in exactly zero values of all entries in each diagonal of an odd number parallel to the main diagonal, is essentially broken in the asymmetrical trap (Figure 20). It is restored only in the limit of a very strong interparticle interaction.

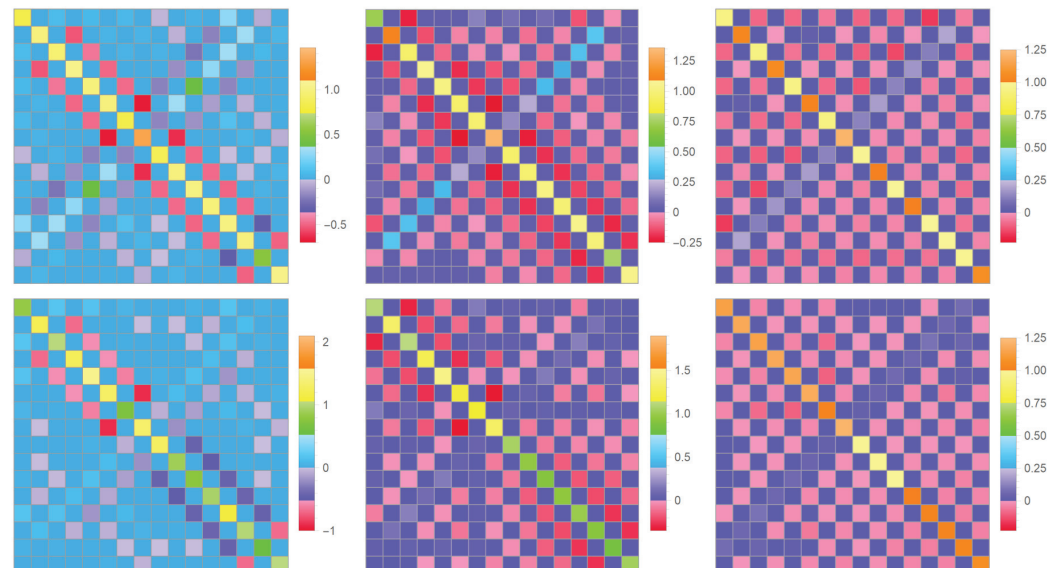


Figure 19. Matrix of Bogoliubov couplings (10) between the first sixteen excited states in the case of the symmetric 1D four-qubit trap shown in Figure 17a; $\alpha_j = 1.5\beta_j = 16\hbar^2/(mL)$, $U_j = 0 \forall j$. The excited states are obtained via the Gram–Schmidt orthogonalization from the condensate wave function ϕ_0 and (**upper row**) the sine functions $\sin(k\pi x/(4L))$, $k = 1, \dots, 16$, or (**lower row**) the first sixteen eigenfunctions of the single-particle Schrödinger Equation (15) in the presence of (the first column) vanishing, $\frac{L}{\xi} \rightarrow 0$, (the second column) moderate, $\frac{L}{\xi} = 2$, and (the third column) strong, $\frac{L}{\xi} = 10$, interaction.

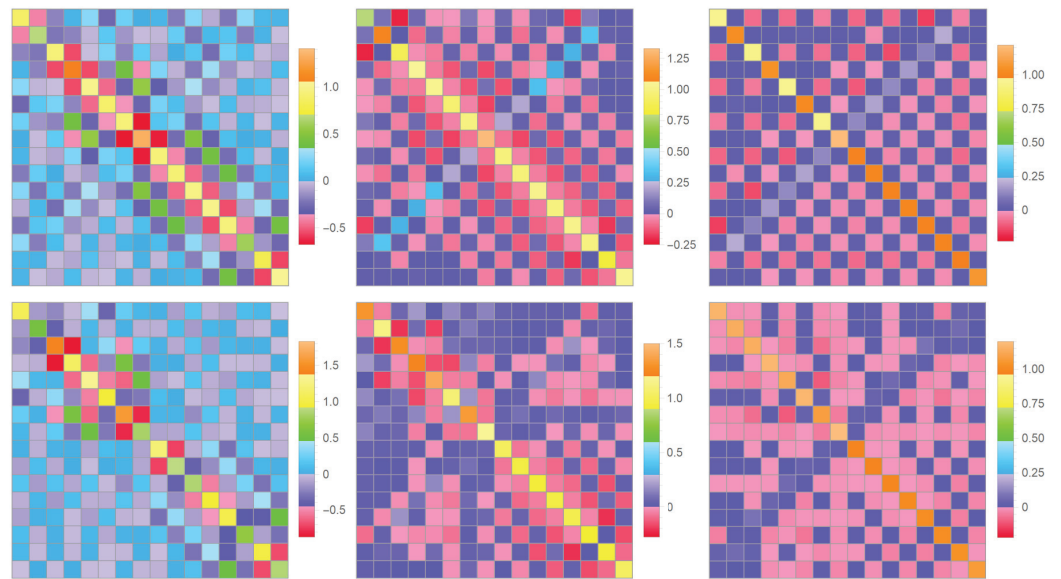


Figure 20. Matrix of Bogoliubov couplings (10) between the first sixteen excited states in the case of the asymmetric 1D four-qubit trap shown in Figure 17b; $\alpha_j = 1.5\beta_j = 16\hbar^2/(mL) \forall j, U(x) = (4\hbar^2/(mL^2))[\theta(x - 0.5L) - \theta(x - 2.5L)]$. The excited states are obtained via the Gram-Schmidt orthogonalization from the condensate wave function ϕ_0 and (**upper row**) the sine functions $\sin(k\pi x/(4L)), k = 1, \dots, 16$, or (**lower row**) the first sixteen eigenfunctions of the single-particle Schrödinger Equation (15) in the presence of (the first column) vanishing, $L/\xi \rightarrow 0$, (the second column) moderate, $L/\xi = 2$, and (the third column) strong, $L/\xi = 10$ interaction.

Second, the maximum spread of essentially nonzero couplings occurs at a moderate interparticle interaction $L/\xi \sim 1$ corresponding to the healing length ξ being on order of the single-qubit cell length L . Much stronger interaction, $L/\xi \gg 1$, tends to localize nonzero couplings just onto the main and anti diagonals. Both these effects are seen in each row of plots in Figures 19 and 20 where the interaction strength is increasing from left to right.

Third, changing the bare-particle excited states, chosen for the simultaneous atom-number detecting within the atomic boson sampling, from the set generated by the Gram-Schmidt orthonormalization out of the sine functions (the upper rows in Figures 19 and 20) to the set generated out of the solutions to the single-particle Schrödinger equation (the lower rows in Figures 19 and 20) greatly affects the structure of the Bogoliubov-coupling matrix both in the symmetrical and asymmetrical traps.

All of the above observations confirm that the inference the multi-qubit BEC trap provides is an excellent opportunity for controlling the Bogoliubov couplings and, hence, the multi-mode squeezing and interference of bare-atom excited modes in a very wide range. Obviously, the more chaotic, messy, dense and wide the distribution of the essentially nonzero elements over the Bogoliubov-coupling matrix (10), the more favorable the set of trap’s parameters and bare-atom excited states chosen for detection of atom numbers for testing manifestations of the computational #P-hardness of atomic boson sampling. Among patterns shown in Figures 19 and 20, the one in the center of the lower row in Figure 20 is the most representative picture of such a complexity.

The asymptotic parameter of this complexity is determined by the Bogoliubov transformation via a multimode dimensionality of the subspace of the excited-states involved in the squeezing-matrix eigenvectors with essentially nonzero squeezing parameters (see Equation (2)) and the Hamiltonian-matrix eigenvectors with low enough eigenenergies corresponding to quasiparticles with essentially nonzero populations (see Equation (42)). In general, this asymptotic parameter increases as the number of groups of excited states chosen for occupation sampling via multi-detector imaging is growing. However, for a given experimental setup with a BEC trap of a finite size, there is a maximum number M of

modes/channels started from which a further increase of the number of sampled/detected occupations would not essentially increase the complexity of boson sampling.

8. Toward Experiments on Atomic Boson Sampling in a BEC Trap

Suppose one has an appropriate BEC trap. (A possible model/example of such a trap is discussed in the previous sections.) Then, as is explained in Sections 1.1 and 1.2, the excited atoms, by themselves, naturally fluctuate and stay in the squeezed states inside the trap even at thermal equilibrium due to interactions with each other. This allows one to eliminate any nonequilibrium processes or dynamics, such as a precise time-dependent control of system parameters and gates or any other type of processing usually associated with quantum computers or simulators, as well as the sophisticated external sources of squeezed or single bosons (required for photonic sampling) from the atomic sampling experiments. It remains just to split the noncondensate into fractions based on the groups of excited states and to measure the distribution of atom numbers over the chosen groups of excited states by means of appropriate detectors.

For instance, one can divide the volume of the trap into a system of spatial cells. Another possibility is to separate atoms in accord with their velocities, that is, to deal with the cells in the momentum space. Anyway, the measurement of atom numbers could be completed by means of a multi-detector imaging. In a BEC destruction scheme, one switches off the confining trap and allows the cloud of trapped atoms to expand freely. In this case, following a standard time-of-flight measuring technique, it is required to take a few successive images of the expanding cloud and properly interpret them in terms of kinetic equations for expansion. In this way, different spatial or momentum subsets of atoms could be separated from each other, and sampling of their occupation numbers could be obtained.

The imaging technique implies an illumination of the atomic cloud with a laser pulse and measuring its transmitted or scattered components by multiple detectors. The transmitted signal carries information on the absorption, dispersion and polarization transformation of light caused by an atomic cloud [33,75,88–90]. The signal due to scattering and fluorescence [91] could be controlled and structured by employing special external cavities and laser sources that support light modes which mimic the excited states preselected for sampling. The optical imaging for atomic boson sampling has much in common with the experiments on the local atom-number fluctuations in BEC traps [47,48,88,90–94].

The spatial or momentum cells/modes represent groups of excited states selected for detection/sampling. Of course, the excited states can be described/composed via an arbitrary basis in the single-particle Hilbert space. Accordingly, the analytical formulae for their joint occupation probability distribution $\rho(\{n_k\})$ and characteristic function, derived in [1], have a universal form, i.e., are valid for any choice of such a basis. A transition from one basis to another one just adds an extra unitary transformation (7) to the Bogoliubov matrix R in Equation (2). Moreover, the universality of the general result for the characteristic function obtained in [1] extends to the so-called marginal or coarse-grained statistics of occupations of any groups of excited states, that is, to the occupation statistics evaluated irrespective to the occupations of all other excited states. The corresponding “incomplete” experiments on atomic boson sampling are the ones to be devised and implemented in reality. Obviously, the condensed atoms, which constitute the macroscopic condensate wave function $\phi_0(\mathbf{r})$ orthogonal to the excited states $\phi_k(\mathbf{r})$, should not be countered during the sampling procedure.

A computational complexity of atomic boson sampling depends on the number M of groups of excited atomic states which are resolved by the multi-detector imaging and are subject to interference due to mixing through the quasiparticles and to squeezing due to interparticle interaction. This number M plays a part of the number of channels in the optical interferometer. A mean occupation of the groups of excited states scales as $(N - N_0)/M$. Obviously, by increasing the total number N of atoms loaded in the trap and, therefore, the number $N - N_0$ of atoms in the noncondensate, one can make larger and,

hence, easier for detection the number of atoms in each of M groups of preselected excited states. Note, however, that the asymptotic parameter responsible for the \sharp P-hardness of atomic boson sampling is not proportional to any of the numbers N , $N - N_0$ or M .

Modern technology allows one to measure the number of atoms in a specified volume or subset of atoms with nearly single atom resolution [91,93–95]. Yet, achieving the single atom accuracy is not absolutely necessary. In particular, the \sharp P-hardness of boson sampling exists even in the case of the threshold detecting scheme in which an outcome of the measurement is just zero or non-zero occupation in each preselected group of excited states [16,24,25].

Finally, an experimental setup should provide the means to reconfigure detectors for projecting upon a vastly varying set of groups of excited states, i.e., to accumulate statistics of joint occupations for numerous different subsets of groups of excited states. Only in this way can the quantum advantage be demonstrated at the most challenging level of the average case.

9. Concluding Remarks

We introduce the multi-qubit BEC trap for studying manifestations of the quantum many-body statistical phenomena which are \sharp P-hard for computing. In particular, we describe the basic properties of the multi-qubit trap, including the single-particle excited states and their energy spectrum via the single-particle Schrödinger Equation (8), the condensate wave function versus the single-particle ground state via the Gross–Pitaevskii Equation (9), and the Bogoliubov couplings (10) between excited states responsible for the formation of quasiparticles and multimode squeezing via the Bogoliubov–de Gennes equations (34). It is completed within the 1D and 2D models, as shown in Equations (13) and (14).

We show that the multi-qubit BEC trap offers a convenient and thoroughgoing control of the many-body system parameters essential for the interplay between excited states' interference and squeezing. This interplay can be revealed via an appropriate decomposition of the Bogoliubov-transformation matrix in Equation (2) and is responsible for the computational \sharp P-hardness which is the basis for a potential quantum advantage of atomic boson sampling over classical computing [1].

It would be very interesting to study experimentally various phenomena associated with the atomic boson sampling. The BEC trap is a boson-sampling platform alternative to a photonic interferometer. Both systems provide the output multivariate statistics which shows computational \sharp P-hardness associated with the hafnian of complex-valued matrices. The proposed multi-qubit trap design discussed in the present paper allows one to vary those matrices and, hence, the output statistics over a wide range. Thus, the latter, major requirement for testing quantum advantage is fulfilled by the multi-qubit BEC trap. The remarkable fact is that classical computing of the hafnian of even relatively low-dimensional matrices corresponding to the number of sampled modes/channels of the order of $M = 8 \times 8 = 64$ is already inaccessible to modern supercomputers. Especially promising are boson-sampling experiments with the multi-qubit BEC trap containing a finite number M of lower-miniband split-off excited states or groups of them (see Figure 1).

The case of a few single-qubit cells with a relatively small number of sampled occupations $M = 2, 3, 4, \dots$ promises the discovery of new quantum effects similar and beyond a particle analog of the simple Hong-Ou-Mandel interference effect. It can be accomplished by means of the current magneto-optical trapping and detection technology. The value of such experiments for the comprehension of the fundamental aspects of the many-body quantum systems responsible for their computational \sharp P-hardness is difficult to overestimate.

The conclusive experiments with an asymptotically large numbers of single-qubit cells, $Q \ll 1$, and sampled excited states or groups of them, $M \gg 1$, addressing the computational \sharp P-hardness of quantum many-body processes are very challenging. Yet, they seem to be within reach and could hit convincing manifestations of quantum advantage.

Author Contributions: All coauthors, S.T., W.S., V.K. (Vadimir Kocharovsky) and V.K. (Vitaly Kocharovsky), contributed equally to deriving the results and writing the paper. All authors have read and agreed to the published version of the manuscript.

Funding: S.T. and V.K. (Vadimir Kocharovsky) acknowledge the support by the Center of Excellence “Center of Photonics” of The Ministry of Science and Higher Education of the Russian Federation, contract № 075-15-2022-316. W.S. acknowledges the support from the Herman F. Heep and Minnie Belle Heep Texas A&M University Endowed Fund via a fellowship in the Institute for Quantum Science and Engineering.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Kocharovsky, V.V.; Kocharovsky, V.V.; Tarasov, S.V. Quantum supremacy of the many-body fluctuations in the occupations of the excited particle states in a Bose–Einstein-condensed gas. *arXiv* **2022**, arXiv:2201.00427v2.
- Scheel, S. Permanents in linear optical networks. *arXiv* **2004**, arXiv:0406127v1.
- Aaronson, S.; Arkhipov, A. The computational complexity of linear optics. *Theory Comput.* **2013**, *9*, 143–252. [CrossRef]
- Lund, A.P.; Laing, A.; Rahimi-Keshari, S.; Rudolph, T.; O’Brien, J.L.; Ralph, T.C. Boson Sampling from a Gaussian State. *Phys. Rev. Lett.* **2014**, *113*, 100502. [CrossRef] [PubMed]
- Bentivegna, M.; Spagnolo, N.; Vitelli, C.; Flamini, F.; Viggianiello, N.; Latmiral, L.; Mataloni, P.; Brod, D.J.; Galvao, E.F.; Crespi, A.; et al. Experimental scattershot boson sampling. *Sci. Adv.* **2015**, *1*, e1400255. [CrossRef]
- Kalai G. The quantum computer puzzle (expanded version). *arXiv* **2016**, arXiv:1605.00992v1.
- Wu, J.; Liu, Y.; Zhang, B.; Jin, X.; Wang, Y.; Wang, H.; Yang, X. Computing Permanents for Boson Sampling on Tianhe-2 Supercomputer. *arXiv* **2016**, arXiv:1606.05836v1.
- Shchesnovich, V.S. Universality of Generalized Bunching and Efficient Assessment of Boson Sampling. *Phys. Rev. Lett.* **2016**, *116*, 123601. [CrossRef]
- Shchesnovich, V.S. Noise in boson sampling and the threshold of efficient classical simulatability. *Phys. Rev. A* **2019**, *100*, 012340. [CrossRef]
- Wang, H.; He, Y.; Li, Y.-H.; Su, Z.-E.; Li, B.; Huang, H.-L.; Ding, X.; Chen, M.-C.; Liu, C.; Qin, J.; et al. High-efficiency multiphoton boson sampling. *Nat. Photonics* **2017**, *11*, 361–365. [CrossRef]
- He, Y.; Ding, X.; Su, Z.-E.; Huang, H.-L.; Qin, J.; Wang, C.; Unsleber, S.; Chen, C.; Wang, H.; He, Y.-M.; et al. Time-Bin-Encoded Boson Sampling with a Single-Photon Device. *Phys. Rev. Lett.* **2017**, *118*, 190501. [CrossRef]
- Loredo, J.C.; Broome, M.A.; Hilaire, P.; Gazzano, O.; Sagnes, I.; Lemaitre, A.; Almeida, M.P.; Senellart, P.; White, A.G. Boson Sampling with Single-Photon Fock States from a Bright Solid-State Source. *Phys. Rev. Lett.* **2017**, *118*, 130503. [CrossRef]
- Hamilton, C.S.; Kruse, R.; Sansoni, L.; Barkhofen, S.; Silberhorn, C.; Jex, I. Gaussian Boson Sampling. *Phys. Rev. Lett.* **2017**, *119*, 170501. [CrossRef]
- Kruse, R.; Hamilton, C.S.; Sansoni, L.; Barkhofen, S.; Silberhorn, C.; Jex, I. Detailed study of Gaussian boson sampling. *Phys. Rev. A* **2019**, *100*, 032326. [CrossRef]
- Chin, S.; Huh, J. Generalized concurrence in boson sampling. *Sci. Rep.* **2018**, *8*, 6101. [CrossRef]
- Quesada, N.; Arrazola, J.M.; Killoran, N. Gaussian boson sampling using threshold detectors. *Phys. Rev. A* **2018**, *98*, 062322. [CrossRef]
- Zhong, H.-S.; Peng, L.-C.; Li, Y.; Hu, Y.; Li, W.; Qin, J.; Wu, D.; Zhang, W.; Li, H.; Zhang, L.; et al. Experimental Gaussian Boson sampling. *Sci. Bull.* **2019**, *64*, 511–515. [CrossRef]
- Paesani, S.; Ding, Y.; Santagati, R.; Chakhmakhchyan, L.; Vigiari, C.; Rottwitt, K.; Oxenløwe, L.K.; Wang, J.; Thompson, M.G.; Laing, A. Generation and sampling of quantum states of light in a silicon chip. *Nat. Phys.* **2019**, *15*, 925–929. [CrossRef]
- Brod, D.J.; Galvão, E.F.; Crespi, A.; Osellame, R.; Spagnolo, N.; Sciarrino, F. Photonic implementation of boson sampling: A review. *Adv. Photonics* **2019**, *1*, 034001.
- Yung, M.-H.; Gao, X.; Huh, J. Universal bound on sampling bosons in linear optics and its computational implications. *Natl. Sci. Rev.* **2019**, *6*, 719–729. [CrossRef]
- Kim, Y.; Hong, K.-H.; Kim, Y.-H.; Huh, J. Connection between BosonSampling with quantum and classical input states. *Opt. Express* **2020**, *28*, 6929–6936. [CrossRef] [PubMed]
- Wang, H.; Qin, J.; Ding, X.; Chen, M.-C.; Chen, S.; You, X.; He, Y.-M.; Jiang, X.; You, L.; Wang, Z.; et al. Boson Sampling with 20 input photons and a 60-mode interferometer in a 10^{14} -dimensional Hilbert space. *Phys. Rev. Lett.* **2019**, *123*, 250503. [CrossRef] [PubMed]
- Zhong, H.-S.; Deng, Y.-H.; Qin, J.; Wang, H.; Chen, M.-C.; Peng, L.-C.; Luo, Y.-H.; Wu, D.; Gong, S.-Q.; Su, H.; et al. Phase-Programmable Gaussian Boson Sampling Using Stimulated Squeezed Light. *Phys. Rev. Lett.* **2021**, *127*, 180502. [CrossRef] [PubMed]

24. Shi, J.; Byrnes, T. Gaussian boson sampling with partial distinguishability. *arXiv* **2021**, arXiv:2105.09583.
25. Villalonga, B.; Niu, M.Y.; Li, L.; Neven, H.; Platt, J.C.; Smelyanskiy, V.N.; Boixo, S. Efficient approximation of experimental Gaussian boson sampling. *arXiv* **2021**, arXiv:2109.11525v1.
26. Harrow, A.W.; Montanaro, A. Quantum computational supremacy. *Nature* **2017**, *549*, 203. [CrossRef]
27. Boixo, S.; Isakov, S.V.; Smelyanskiy, V.N.; Babbush, R.; Ding, N.; Jiang, Z.; Bremner, M.J.; Martinis, J.M.; Neven, H. Characterizing quantum supremacy in near-term devices. *Nat. Phys.* **2018**, *14*, 595–600. [CrossRef]
28. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef]
29. Zhong, H.-S.; Wang, H.; Deng, Y.-H.; Chen, M.-C.; Peng, L.-C.; Luo, Y.-H.; Qin, J.; Wu, D.; Ding, X.; Hu, Y.; Hu, P.; et al. Quantum computational advantage using photons. *Science* **2020**, *370*, 1460–1463. [CrossRef]
30. Dalzell, A.M.; Harrow, A.W.; Koh, D.E.; La Placa, R.L. How many qubits are needed for quantum computational supremacy? *Quantum* **2020**, *4*, 264. [CrossRef]
31. Nisbet-Jones, P.B.R.; Dilley, J.; Holleczek, A.; Barter, O.; Kuhn, A. Photonic qubits, qutrits and ququads accurately prepared and delivered on demand. *New J. Phys.* **2013**, *15*, 053007. [CrossRef]
32. Ringbauer, M.; Meth, M.; Postler, L.; Stricker, R.; Blatt, R.; Schindler, P.; Monz, T. A universal qudit quantum processor with trapped ions. *Nat. Phys.* **2022**, *18*, 1053–1057. [CrossRef]
33. Kristensen, M.; Christensen, M.; Gajdacz, M.; Igllicki, M.; Pawłowski, K.; Klempt, C.; Sherson, J.; Rzazewski, K.; Hilliard, A.; Arlt, J. Observation of atom number fluctuations in a Bose-Einstein condensate. *Phys. Rev. Lett.* **2019**, *122*, 163601. [CrossRef]
34. Tarasov, S.V.; Kocharovskiy, V.I.; Kocharovskiy, V.V. Bose-Einstein condensate fluctuations versus an interparticle interaction. *Phys. Rev. A* **2020**, *102*, 043315. [CrossRef]
35. Pitaevskii, L.; Stringary, S. *Bose-Einstein Condensation and Superfluidity*; Oxford University Press: Oxford, UK, 2016.
36. Steinhauer, J.; Ozeri, R.; Katz, N.; Davidson, N. Excitation spectrum of a Bose-Einstein condensate. *Phys. Rev. Lett.* **2002**, *88*, 120407. [CrossRef]
37. Niu, Q.; Carusotto, I.; Kuklov, A.B. Imaging of critical correlations in optical lattices and atomic traps. *Phys. Rev. A* **2006**, *73*, 053604. [CrossRef]
38. Makotyn, P.; Klauss, C.E.; Goldberger, D.L.; Cornell, E.A.; Jin, D.S. Universal dynamics of a degenerate unitary Bose gas. *Nat. Phys.* **2014**, *10*, 116–119. [CrossRef]
39. Chang, R.; Bouton, Q.; Cayla, H.; Qu, C.; Aspect, A.; Westbrook, C.I.; Clément, D. Momentum-resolved observation of thermal and quantum depletion in a Bose gas. *Phys. Rev. Lett.* **2016**, *117*, 235303. [CrossRef]
40. Lopes, R.; Eigen, C.; Navon, N.; Clement, D.; Smith, R.P.; Hadzibabic, Z. Quantum depletion of a homogeneous Bose-Einstein condensate. *Phys. Rev. Lett.* **2017**, *119*, 190404. [CrossRef]
41. Garratt, S.J.; Eigen, C.; Zhang, J.; Turzák, P.; Lopes, R.; Smith, R.P.; Hadzibabic, Z.; Navon, N. From single-particle excitations to sound waves in a box-trapped atomic Bose-Einstein condensate. *Phys. Rev. A* **2019**, *99*, 021601. [CrossRef]
42. Pieczarka, M.; Estrecho, E.; Boozarjmehr, M.; Bleu, O.; Steger, M.; West, K.; Pfeiffer, L.N.; Snoko, D.W.; Levinsen, J.; Parish, M.M.; et al. Observation of quantum depletion in a nonequilibrium exciton–polariton condensate. *Nat. Commun.* **2020**, *11*, 429. [CrossRef] [PubMed]
43. Shin, Y.; Saba, M.; Pasquini, T.A.; Ketterle, W.; Pritchard, D.E.; Leanhardt, A.E. Atom Interferometry with Bose-Einstein Condensation in a Double-Well Potential. *Phys. Rev. Lett.* **2004**, *92*, 050405-1.
44. Opanchuk, B.; Rosales-Zárate, L.; Teh, R.Y.; Dalton, B.J.; Sidorov, A.; Drummond, P.D.; Reid, M.D. Mesoscopic two-mode entangled and steerable states of 40 000 atoms in a Bose-Einstein-condensate interferometer. *Phys. Rev. A* **2019**, *100*, 060102(R). [CrossRef]
45. Egorov, M.; Anderson, R.P.; Ivannikov, V.; Opanchuk, B.; Drummond, P.; Hall, B.V.; Sidorov, A.I. Long-lived periodic revivals of coherence in an interacting Bose-Einstein condensate. *Phys. Rev. A* **2011**, *84*, 021605(R). [CrossRef]
46. Berrada, T.; van Frank, S.; Bucker, R.; Schumm, T.; Schaff, J.-F.; Schmiedmayer, J. Integrated Mach–Zehnder interferometer for Bose–Einstein condensates. *Nat. Commun.* **2013**, *4*, 2077. [CrossRef]
47. Sinatra, A.; Castin, Y.; Li, Y. Particle number fluctuations in a cloven trapped Bose gas at finite temperature. *Phys. Rev. A* **2010**, *81*, 053623. [CrossRef]
48. Klawunn, M.; Recati, A.; Pitaevskii, L.P.; Stringari, S. Local atom-number fluctuations in quantum gases at finite temperature. *Phys. Rev. A* **2011**, *84*, 033612. [CrossRef]
49. Calzetta, E.A.; Hu, B.L. Bose–Einstein condensate collapse and dynamical squeezing of vacuum fluctuations. *Phys. Rev. A* **2003**, *68*, 043625. [CrossRef]
50. Tarasov, S.V.; Kocharovskiy, V.I.; Kocharovskiy, V.V. Grand Canonical Versus Canonical Ensemble: Universal Structure of Statistics and Thermodynamics in a Critical Region of Bose–Einstein Condensation of an Ideal Gas in Arbitrary Trap. *J. Stat. Phys.* **2015**, *161*, 942–964. [CrossRef]
51. Popov, V.N. Green functions and thermodynamic functions of a non-ideal Bose gas. *Sov. Phys. JETP* **1965**, *20*, 1185–1188.
52. Griffin, A. Conserving and gapless approximations for an inhomogeneous Bose gas at finite temperatures. *Phys. Rev. B* **1996**, *53*, 9341–9347. [CrossRef]
53. Shi, H.; Griffin, A. Finite-temperature excitations in a dilute Bose-condensed gas. *Phys. Rep.* **1998**, *304*, 1–87. [CrossRef]
54. Kocharovskiy, V.V.; Kocharovskiy, V.I. Microscopic theory of a phase transition in a critical region: Bose–Einstein condensation in an interacting gas. *Phys. Lett. A* **2015**, *379*, 466–470. [CrossRef]

55. Kocharovskiy, V.V.; Kocharovskiy, V.I.V. Microscopic theory of phase transitions in a critical region. *Physica Scr.* **2015**, *90*, 108002. [CrossRef]
56. Kocharovskiy, V.V.; Kocharovskiy, V.I.V. Exact general solution to the three-dimensional Ising model and a self-consistency equation for the nearest-neighbors' correlations. *arXiv* **2016**, arXiv:1510.07327v3.
57. Kocharovskiy, V.V.; Kocharovskiy, V.I.V.; Tarasov, S.V. The Hafnian Master Theorem. *Linear Algebra Its Appl.* **2022**, *651*, 144–161. [CrossRef]
58. Valiant, L.G. The complexity of computing the permanent. *Theor. Comput. Sci.* **1979**, *8*, 189–201. [CrossRef]
59. Jerrum, M.; Sinclair, A.; Vigoda, E. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM* **2004**, *51*, 671–697. [CrossRef]
60. Bjorklund, A.; Gupt, B.; Quesada, N. A faster hafnian formula for complex matrices and its benchmarking on a supercomputer. *ACM J. Exp. Algorithmics* **2019**, *242019*, 1–17. [CrossRef]
61. Ryser, H.J. *Combinatorial Mathematics, The Carus Mathematical Monographs, No. 14*; The Mathematical Association of America: Washington, DC, USA, 1963.
62. Glynn, D.G. The permanent of a square matrix. *Eur. J. Comb.* **2010**, *31*, 1887–1891. [CrossRef]
63. Ikenmeyer, C.; Landsberg, J.M. On the complexity of the permanent in various computational models. *J. Pure Appl. Algebra* **2017**, *221*, 2911–2927. [CrossRef]
64. Kocharovskiy, V.V.; Kocharovskiy, V.I.V.; Tarasov, S.V. Unification of the nature's complexities via a matrix permanent—Critical phenomena, fractals, quantum computing, #P-complexity. *Entropy* **2020**, *22*, 322. [CrossRef] [PubMed]
65. Chin, C.; Grimm, R.; Julienne, P.; Tiesinga, E. Feshbach resonances in ultracold gases. *Rev. Mod. Phys.* **2010**, *82*, 1225–1286. [CrossRef]
66. Bloch, I.; Dalibard, J.; Zwerger, W. Many-body physics with ultracold gases. *Rev. Mod. Phys.* **2008**, *80*, 885. [CrossRef]
67. Kocharovskiy, V.V.; Kocharovskiy, V.V.; Martyanov, V.Y.; Tarasov, S.V. Exact recursive calculation of circulant permanents: A band of different diagonals inside a uniform matrix. *Entropy* **2021**, *23*, 1423. [CrossRef]
68. Ma, X.; Rhodes, W. Multimode squeeze operators and squeezed states. *Phys. Rev. A* **1990**, *41*, 4625–4631. [CrossRef]
69. Braunstein, S.L.; Squeezing as an irreducible resource. *Phys. Rev. A* **2005**, *71*, 055801. [CrossRef]
70. Cariolaro, G.; Pirobon, G. Reexamination of Bloch-Messiah reduction. *Phys. Rev. A* **2016**, *93*, 06211. [CrossRef]
71. Vogel, W.; Welsch, D.-G. *Quantum Optics*, 3rd ed.; Wiley-VCH Verlag GmbH: Berlin, Germany, 2006.
72. Huh, J.; Yung, M.-H. Vibronic Boson Sampling: Generalized Gaussian Boson Sampling for Molecular Vibronic Spectra at Finite Temperature. *Sci. Rep.* **2017**, *7*, 7462. [CrossRef]
73. Huh, J. Multimode Bogoliubov transformation and Husimi's Q-function. *J. Phys. Conf. Ser.* **2020**, *1612*, 012015. 10.1088/1742-6596/1612/1/012015. [CrossRef]
74. Kocharovskiy, V.V.; Kocharovskiy, V.V.; Scully, M.O. Condensation of N bosons. III. Analytical results for all higher moments of condensate fluctuations in interacting and ideal dilute Bose gases via the canonical ensemble quasiparticle formulation. *Phys. Rev. A* **2000**, *61*, 053606. [CrossRef]
75. Ilo-Okeke, E.O.; Sunami, S.; Foot, C.J.; Byrnes, T. Faraday imaging induced squeezing of a double-well Bose-Einstein condensate. *Phys. Rev. A* **2021**, *104*, 053324. [CrossRef]
76. Ho, T.-L.; Yip, S.K. Fragmented and Single Condensate Ground States of Spin-1 Bose Gas. *Phys. Rev. Lett.* **2000**, *84*, 4031–4034. [CrossRef]
77. Gati, R.; Hemmerling, B.; Fölling, J.; Albiez, M.; Oberthaler, M.K. Noise Thermometry with Two Weakly Coupled Bose-Einstein Condensates. *Phys. Rev. Lett.* **2006**, *96*, 130404. [CrossRef]
78. Mueller, E.J.; Ho, T.-L.; Ueda, M.; Baym, G. Fragmentation of Bose-Einstein condensates. *Phys. Rev. A* **2006**, *74*, 033612. [CrossRef]
79. Masiello, D.J.; Reinhardt, W.P. Symmetry-Broken Many-Body Excited States of the Gaseous Atomic Double-Well Bose-Einstein Condensate. *J. Phys. Chem. A* **2019**, *123*, 1962–1967. [CrossRef]
80. Borisenko, I.V.; Demidov, V.E.; Pokrovskiy, V.L.; Demokritov, S.O. Spatial separation of degenerate components of magnon Bose-Einstein condensate by using a local acceleration potential. *Sci. Rep.* **2020**, *10*, 14881. [CrossRef]
81. Salasnich, L.; Parola, A.; Reatto, L. Bose condensate in a double-well trap: Ground state and elementary excitations. *Phys. Rev. A* **1999**, *60*, 4171–4174. [CrossRef]
82. Griffiths, D. *Introduction to Quantum Mechanics*, 2nd ed.; Pearson Prentice Hall: New York, NY, USA, 2004.
83. Pade, J. One-Dimensional Piecewise-Constant Potentials. In *Quantum Mechanics for Pedestrians 2: Applications and Extensions*; Undergraduate Lecture Notes in Physics; Springer: Cham, Switzerland, 2014.
84. Baek, S. K.; Yi, S. D.; Kim, M. Particle in a box with a time-dependent δ -function potential. *Phys. Rev. A* **2016**, *94*, 052124. [CrossRef]
85. Sheils, N.E.; Deconinck, B. The time-dependent Schrödinger equation with piecewise constant potentials. *Eur. J. Appl. Math.* **2020**, *31*, 57–83. [CrossRef]
86. Janowicz, M. Method of multiple scales in quantum optics. *Phys. Rep.* **2003**, *375*, 327–410. [CrossRef]
87. Schmied, R. *Using Mathematica for Quantum Mechanics*; Springer: Singapore, 2020. [CrossRef]
88. Jacqmin, T.; Armijo, J.; Berrada, T.; Kheruntsyan, K.V.; Bouchoule, I. Sub-Poissonian fluctuations in a 1D Bose gas: From the quantum quasicondensate to the strongly interacting regime. *Phys. Rev. Lett.* **2010**, *105*, 230405. [CrossRef] [PubMed]
89. Kristensen, M.A.; Gajdacz, M.; Pedersen, P.L.; Klempt, C.; Sherson, J.F.; Arlt, J.J.; Hilliard, A.J. Sub-atom shot noise Faraday imaging of ultracold atom clouds. *J. Phys. B At. Mol. Opt. Phys.* **2017**, *50*, 034004. [CrossRef]

90. Esteve, J.; Trebbia, J.-B.; Schumm, T.; Aspect, A.; Westbrook, C.I.; Bouchoule, I. Observations of density fluctuations in an elongated Bose gas: Ideal gas and quasicondensate regimes. *Phys. Rev. Lett.* **2006**, *96*, 130403. [CrossRef]
91. Chuu, C.-S.; Schreck, F.; Meyrath, T.P.; Hanssen, J.L.; Price, G.N.; Raizen, M.G. Direct observation of sub-Poissonian number statistics in a degenerate Bose gas. *Phys. Rev. Lett.* **2005**, *95*, 260403. [CrossRef]
92. Armijo, J.; Jacqmin, T.; Kheruntsyan, K.V.; Bouchoule, I. Probing three-body correlations in a quantum gas using the measurement of the third moment of density fluctuations. *Phys. Rev. Lett.* **2010**, *105*, 230402. [CrossRef]
93. Dotsenko, I.; Alt, W.; Khudaverdyan, M.; Kuhr, S.; Meschede, D.; Miroshnychenko, Y.; Schrader, D.; Rauschenbeutel, A. Submicrometer Position Control of Single Trapped Neutral Atoms. *Phys. Rev. Lett.* **2005**, *95*, 033002. [CrossRef]
94. Schlosser, N.; Raymond, G.; Grangier, P. Collisional Blockade in Microscopic Optical Dipole Traps. *Phys. Rev. Lett.* **2002**, *89*, 023005. [CrossRef]
95. Pons, M.; del Campo, A.; Muga, J.G.; Raizen, M.G. Preparation of atomic Fock states by trap reduction. *Phys. Rev. A* **2009**, *79*, 033629. [CrossRef]

Article

Digital Quantum Simulation of the Spin-Boson Model under Markovian Open-System Dynamics

Andreas Burger^{1,2,3,*}, Leong Chuan Kwek^{3,4,5} and Dario Poletti^{2,3,4,6,7}

¹ Faculty of Physics, Ludwig-Maximilians-Universität Munich, Geschwister-Scholl-Platz 1, 80539 Munich, Germany

² Science, Mathematics and Technology Cluster, Singapore University of Technology and Design, Singapore 487372, Singapore

³ Centre for Quantum Technologies, National University of Singapore, Singapore 117543, Singapore

⁴ National Institute of Education and Institute of Advanced Studies, Nanyang Technological University, Singapore 637616, Singapore

⁵ MajuLab, CNRS-UNS-NUS-NTU International Joint Research Unit, Singapore 117543, Singapore

⁶ EPD Pillar, Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372, Singapore

⁷ The Abdus Salam International Centre for Theoretical Physics, Strada Costiera 11, 34151 Trieste, Italy

* Correspondence: andreas.burger@physik.uni-muenchen.de

Abstract: Digital quantum computers have the potential to simulate complex quantum systems. The spin-boson model is one of such systems, used in disparate physical domains. Importantly, in a number of setups, the spin-boson model is open, i.e., the system is in contact with an external environment which can, for instance, cause the decay of the spin state. Here, we study how to simulate such open quantum dynamics in a digital quantum computer, for which we use an IBM hardware. We consider in particular how accurate different implementations of the evolution result as a function of the level of noise in the hardware and of the parameters of the open dynamics. For the regimes studied, we show that the key aspect is to simulate the unitary portion of the dynamics, while the dissipative part can lead to a more noise-resistant simulation. We consider both a single spin coupled to a harmonic oscillator, and also two spins coupled to the oscillator. In the latter case, we show that it is possible to simulate the emergence of correlations between the spins via the oscillator.

Keywords: quantum computing; NISQ; open system

Citation: Burger, A.; Kwek, L.C.; Poletti, D. Digital Quantum Simulation of the Spin-Boson Model under Markovian Open-System Dynamics. *Entropy* **2022**, *24*, 1766. <https://doi.org/10.3390/e24121766>

Academic Editor: Rosario Lo Franco

Received: 7 November 2022

Accepted: 29 November 2022

Published: 2 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A natural application of quantum computers is the simulation of quantum systems [1,2]. Furthermore, most hardware realizations of quantum computers implement the qubit. A prevalent qubit-based quantum system is the spin system. Existing quantum computers are based on unitary quantum circuits. Consequently, there has been a plethora of research on closed quantum systems [3–6]. Amongst the spin models, an important class is the spin-boson problem, where one or more spins are coupled to several bosonic degrees of freedom. These models possess rich many-body physics and they can model realistic coupling between electron transfer and protein motion or a solvent [7–11].

In recent years, NISQ computers [12,13] have offered a new perspective on the implementations on digital devices, leading to an explosion of activities. Not all computing tasks are amenable to quantum processing. Classical optimization can often perform better than quantum algorithms. The challenges of device-induced noise have led to the popularity of hybrid quantum-classical variational algorithms (VQA) that split the workload between a quantum and a classical processor. These techniques are ideally suited for the evaluation of different quantities such as eigenstates [14], general quantum approximate optimization algorithms [15], off-diagonal elements of matrices [16] and more [13]. Importantly, new error mitigation approaches have also been proposed [17–19]. VQA has been applied to boson-spin systems or its equivalents [20–22]. Regarding open systems, different VQA approaches

have been tested. They include approaches based on imaginary time evolution [23,24], stochastic Schrödinger Equation [25], variational quantum eigensolvers to reach steady states [26,27], and the quantum-assisted simulator without a classical-quantum feedback loop [28]. Mapping bosonic problems to quantum circuits has been laid out in [29–31], while a recent implementation of spin-boson models can be found in [6].

Simulating open quantum systems entirely on digital quantum computers has primarily focused on two-level systems. The amplitude damping channel has been implemented with a unitary dilation of the Kraus operators [32], using uniformly controlled gates [33,34], and with the amplitude damping circuit [2,35]. Larger systems have been realized using a linear combination of unitary matrices [36,37] and modified stochastic Schrödinger equation methods [38]. In [25], the authors proposed a hybrid classical-quantum variational approach to simulate generic Markovian open quantum systems.

Our aim is to simulate the open dynamics of a spin-boson model coupled to a dissipative channel on a digital quantum computer. We do this by mapping the bosonic modes to qubits, Trotterizing the unitary evolution, and modeling the dissipative portion via repeated collisions with a reset auxiliary qubit [35,39,40]. In doing so, we focus on using different noise levels in the quantum computer, from the value in current hardware, to 1% of it. With this in mind, we study how different implementations of the simulation perform in presence of different noise levels.

The paper is organized as follows. In Section 2.1, we introduce the open spin-boson model and lay out the circuit implementation. In Section 2.2, we describe the circuit implementation of the unitary and dissipative evolutions. We then detail our use of quantum hardware and noise-related limitations of the devices in Section 2.3. Our results are presented in Section 3. We quantify the error stemming from approximations in the model, and for different magnitudes of noise in the device. We study the optimal time-step-sizes and dissipative rates in terms of fidelity. Finally, we increase the system size to two spins and investigate whether it is possible to observe rising correlations amongst the spins.

2. Method

2.1. Model

We consider N_S non-interacting spins coupled to a single harmonic oscillator, as well as to a bath, as can be seen in Figure 1. The closed system is governed by the quantum Rabi Hamiltonian [41–43], which describes the ultra-strong coupling regime, where the usual rotating wave approximation breaks down and the counter-rotating term can no longer be neglected [44–46].

$$\hat{H}_{SB} = \hbar\omega\hat{a}^\dagger\hat{a} + \sum_{i=1}^{N_S} \frac{1}{2}(h\hat{\sigma}_k^z + \epsilon\hat{\sigma}_k^x) + \lambda\hat{\sigma}_k^x(\hat{a}^\dagger + \hat{a}), \tag{1}$$

Experimentally, the ultra-strong coupling regime has been investigated in circuit QED [47–52], trapped ions [53], photonic systems [54], and semiconductors [55,56].

In Equation (1), \hat{a}^\dagger and \hat{a} , respectively, create and destroy one excitation in the harmonic oscillator while $\hat{\sigma}_k^x = \hat{\sigma}_k^+ + \hat{\sigma}_k^-$ and $\hat{\sigma}_k^z$ are Pauli operators acting on the spin(s). h is the local magnetic field in the z direction while ϵ is a field in the x direction. λ is the magnitude of the coupling between the spins and the harmonic oscillator, with frequency ω . In the following, we will work in units such that $h = \hbar = 1$.

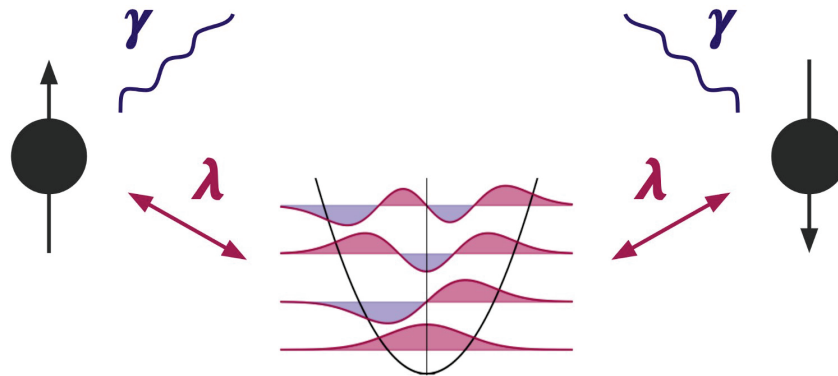


Figure 1. Depiction of the model described by Equations (1) and (2) for a number of spins $N_S = 2$. The two spin sites are coupled to one harmonic oscillator of frequency ω via coupling parameter λ . Each of the spins dissipates independently into the environment at a rate of γ .

The dissipative part of the dynamics is here described by a Markovian master equation in Gorini–Kossakovski–Sudarshan–Lindblad form [57,58]

$$\frac{d\hat{\rho}}{dt} = -\frac{i}{\hbar}[\hat{H}_{SB}, \hat{\rho}] + \gamma \sum_k (2\hat{L}_k \hat{\rho} \hat{L}_k^\dagger - \{\hat{L}_k^\dagger \hat{L}_k, \hat{\rho}\}) \tag{2}$$

with the amplitude damping channel $\hat{L}_k = |\downarrow\rangle_k \langle\uparrow|$ acting on the k -th spin and γ being the decay rate. $|\downarrow\rangle$ represents the vacuum state, whereas $|\uparrow\rangle$ represents the excited state of the spin.

Equation (2) describes a setup where loss from imperfections in the cavity is negligible compared to the spins emissions. In these systems, undesired decay transitions can include the emission of frequencies which are suppressed in the cavity and are thus effectively lost [59–61].

2.2. Circuit Implementation

In this section, we describe how we implement the evolution governed by Equations (1) and (2) in a quantum circuit.

2.2.1. Encoding of the Hamiltonian

We map the spin and bosonic operators in \hat{H}_{SB} to Pauli operators, and Trotterize the unitary $e^{-i\hat{H}_{SB}t}$. The spin part is trivially mapped to qubits. For the bosonic subspace and operators, we use a d-level-to-qubit mapping with Gray Code as the integer-to-bit encoding, as described in [31,62]. We give more details on the mapping to Q_B qubits in Appendix A.

2.2.2. Trotterization of Unitary

To implement the unitary evolution operator $U = e^{-i\hat{H}_{SB}t}$, we consider the first-order U_1 and second-order U_2 Suzuki–Trotter product formulas [63,64]

$$U_1 = (e^{-ih_1\Delta t} e^{-ih_2\Delta t} \dots e^{-ih_N\Delta t})^{\frac{t}{\Delta t}} \tag{3}$$

$$U_2 = (e^{-ih_1\frac{\Delta t}{2}} \dots e^{-ih_N\frac{\Delta t}{2}} e^{-ih_N\frac{\Delta t}{2}} \dots e^{-ih_1\frac{\Delta t}{2}})^{\frac{t}{\Delta t}} \tag{4}$$

where h_k are N different, non-commuting, terms of the Hamiltonian after encoding and $\Delta t = t/N$. The individual exponentials of Pauli strings $e^{-ih_k\Delta t}$ are then implemented via the CNOT-staircase [2,3], which is taken care of by Qiskit [65]. See Equations (A2) and (A3) in Appendix A for more details on h_k .

2.2.3. Collisional Model

We model the local master equation, namely Equation (2), via repeated collisions [39,66]. Figure 2 gives a depiction of a single collision. We consider the spin qubit s , and auxiliary qubit a and where a controlled- $R_Y(\theta)$ (rotation around y axis) is followed by a controlled-NOT and a reset of the auxiliary qubit, see Appendices C and D for more details. To reproduce Equation (2) we use $\theta = \arcsin(\sqrt{1 - e^{-\gamma t}})$ [2].

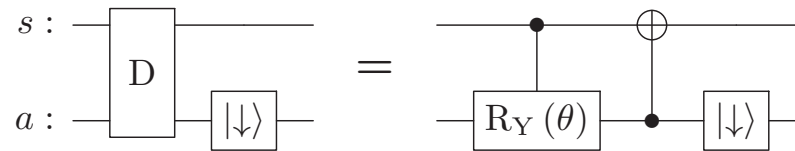


Figure 2. Circuit implementation of the dissipative part of the circuit, D , which represent a single collision to model Equation (2). s is the qubit representing the spin while a represents the auxiliary qubit.

2.2.4. Integration of Dissipative and Unitary Part

To integrate the step of Figure 2 in the main circuit, we employ a first-order Suzuki–Trotter decomposition which alternates between the unitary and the dissipative parts. In Figure 3a, we depict three steps of the evolution of a single spin coupled to a harmonic oscillator mapped to two qubits, while in Figure 3b, we show our implementation of three-step evolution of the case with two spins and one harmonic oscillator. For considerations of connectivity, the auxiliary qubits needed for the dissipative channel are placed at the edges of the circuit, next to the spins. After all time-steps are finished, the qubits representing the spin(s) s_k and the bosons b_k are measured, while the state of the auxiliary qubit is ignored.

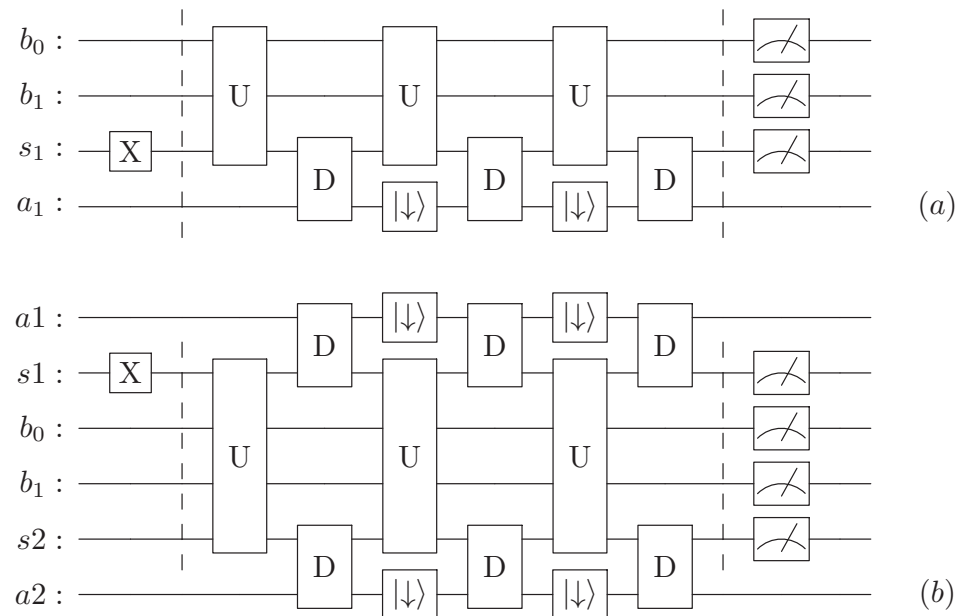


Figure 3. Circuit structure, alternating between a unitary evolution and collisions with auxiliary qubits and resets. (a) For a single spin and a (b) two spin system. Here, the spins are represented by s_k , the harmonic oscillator modes (4 levels) are encoded in the b_k qubits and the auxiliary qubits are represented by a_k . X-Gate represents the initial state preparation, $|\downarrow\rangle$ represents resets, the final gates represent measurements, while D is described in Figure 2.

2.3. Quantum Hardware Simulation

To perform our quantum circuit simulations and run it on actual quantum hardware, we use IBM’s Qiskit software [65]. The Quantum Computer we use is the 7-qubit

ibmq_jakarta device with a native gate set {CNOT, ID, RZ, SX, X}. Each circuit is run with $2^{13} = 8192$ shots (repetitions).

We quantify the error at each point in time as the infidelity \mathcal{I} [67]

$$\mathcal{I}(\hat{\rho}, \hat{\rho}') = 1 - \left(\text{Tr} \left[\sqrt{\sqrt{\hat{\rho}} \hat{\rho}' \sqrt{\hat{\rho}}} \right] \right)^2 \tag{5}$$

where we obtain the density matrix $\hat{\rho}'$ of the circuit via quantum state tomography. We also consider a time-averaged version of the infidelity $\bar{\mathcal{I}}$, which is obtained by averaging the infidelity over time, except for the time $t = 0$, which consists of just the state preparation. The exact density matrix $\hat{\rho}$ for the benchmark is obtained from the exact evolution of the master Equation (Equation (2)), for which we use QuTiP [68]. To mitigate the measurement error on noisy hardware, we classically post-process the results with Qiskit’s error mitigation, which approximates the inverse of the noise matrix of the readout [69].

Reduced-Noise Models

While it is important to study how current quantum processors can evaluate the model we study, we also aim to explore what could be the performance of future, less noisy, hardware. To model these scenarios, we use the same error channels that IBM uses to describe their current devices.

The noise models include error sources in the gates, as thermal relaxation (relaxation and dephasing) and depolarizing errors, and also readout errors [70].

For our reduced-noise models we scale down the average gate infidelity \mathcal{I}_{Gate} , the gate times t_{Gate} , and the false-readout probabilities, probability of measuring 1 when the state is 0 $P(1|0)$ or vice versa $P(0|1)$, by the same noise-factor ζ , or more precisely

$$\mathcal{I}_{Gate} \rightarrow \zeta \cdot \mathcal{I}_{Gate} \tag{6}$$

$$t_{Gate} \rightarrow \zeta \cdot t_{Gate} \tag{7}$$

$$P(1|0), P(0|1) \rightarrow \zeta \cdot P(1|0), \zeta \cdot P(0|1) \tag{8}$$

where ζ ranges from 0 to 1.

Indeed, realistically some of these parameters will not see equal improvement in the coming years, but a more detailed analysis of differentiated improvements of different aspects is beyond the scope of this work. Details on the error channels can be found in Appendix B.

3. Results

Inaccuracies in the implementation of the model on a quantum computer can stem from different causes of completely different nature. We will first consider errors that rise from the Trotterization of the evolution in Section 3.1. We will then consider errors due to the noisy nature of the quantum computer in Section 3.2. In Section 3.3, we will then study the case of two spins coupled to the harmonic oscillator.

In the following, for the Hamiltonian, we choose the parameters $\epsilon = 0.5, \omega = 4, \lambda = 2$ for one spin and $\epsilon = 0.5, \omega = 6, \lambda = 2$ for two spins. For the open dissipative rate, we choose $\gamma = 1$. With these parameters, an accurate evolution of the system up to a time $t = 2$ can be obtained considering simply four levels for the harmonic oscillator, which can then be encoded with two qubits. For the initial state, we consider a pure product state between spins and bosons, with one spin in the excited state and zero excitations in the harmonic oscillator. This choice of initial conditions allows observing oscillatory, non-trivial dynamics from early times, while not requiring too many levels for the harmonic oscillator.

3.1. Error from the Circuit Implementation

As explained earlier, to implement the open dynamics, we Trotterize the unitary and dissipative parts of the master equation. However, for the implementation of the

unitary evolution, we need to rely on another layer of Trotterization. In Figure 4, we consider a unitary evolution with Hamiltonian \hat{H}_{SB} from Equation (1) for a time-step Δt and the possible implementation error, but considering no noise from the machine (blue lines). Implementing the various non-commuting terms of \hat{H}_{SB} in Qiskit [65] requires 48 single-qubit- and 19 CX-Gates or 79 single-qubit- and 28 CX-Gates, when using first or second-order Trotter, respectively, (Table A1).

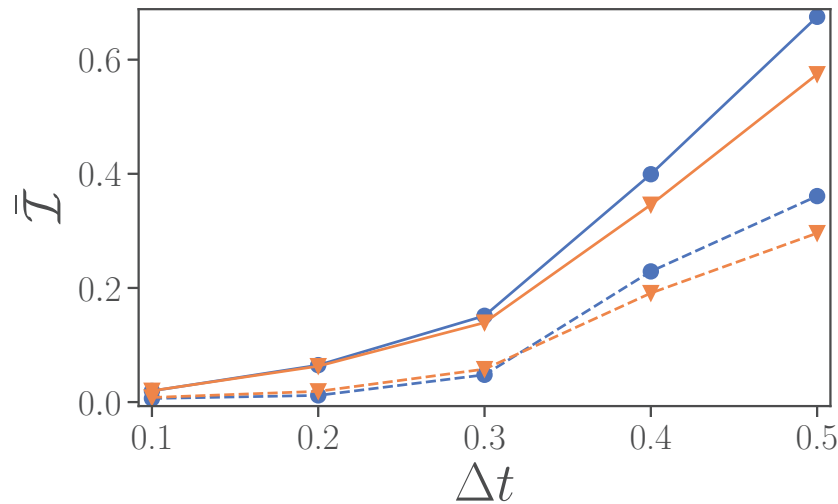


Figure 4. Time-averaged infidelity for the evolution from $t = 0$ to $t = 2$. Noiseless simulations of the Hamiltonian $\gamma = 0$ (blue line, dots) and the open system $\gamma = 1$ (orange line, triangle). The solid and dashed lines are used, respectively, for first-order and second-order Trotter implementations. The common parameters are $\epsilon = 0.5$, $\omega = 4$, $\lambda = 2$. The number of time-steps for $\Delta t = t/N$ $N = 0.1, 0.2, 0.3, 0.4, 0.5$ are $N = 20, 10, 7, 5, 4$, respectively, not counting $t = 0$, which just consists of the initial state preparation.

In Figure 4, we evaluate the infidelity for both unitary and dissipative evolutions, i.e., following Equation (2) for $\gamma = 0$ (blue lines with circles) or $\gamma = 1$ (orange lines with triangles), versus Δt . We observe that the second-order Trotterization, dashed lines, has significantly smaller infidelity than a first-order implementation, continuous lines. Interestingly, beyond $\Delta t \approx 0.3$, the infidelity in just the Hamiltonian simulation is larger than the infidelity when including the dissipation. Furthermore, independently of whether one considers first-order or second-order Trotterization, the dissipative dynamics has either smaller infidelity or it is very close to the unitary case. This implies that the unitary step implementing the Hamiltonian is the main contribution to the infidelity compared to the implementation of the dissipation.

3.2. Error in the Presence of Noise

We now turn to more realistic, and thus noisy, devices. In Figure 4, for noiseless simulations, we observed that the infidelity increases monotonously with the time-step size Δt , and that a second-order Trotterization is always preferred. In the presence of noise, however, an increased number of gates can lead to stronger noise effects, and thus instead of improving the quality of the simulations, it may result in worse fidelity. In Figure 5a, we thus consider the evolution of the full model, unitary and dissipative part, up to a time $t = 2$ for different magnitudes of noise $\zeta = 0.01, 0.1, 1$ (from lighter to darker colors), for either a first-order Trotter step (continuous lines) or a second-order Trotter step (dashed lines). In particular, we depict the infidelity versus the length of the time-step Δt . We observe that for intermediate values of noise $\zeta = 0.1, 1$ there is an optimal time interval Δt that corresponds to the lowest infidelity, and that first-order Trotterization can perform better at smaller Δt .

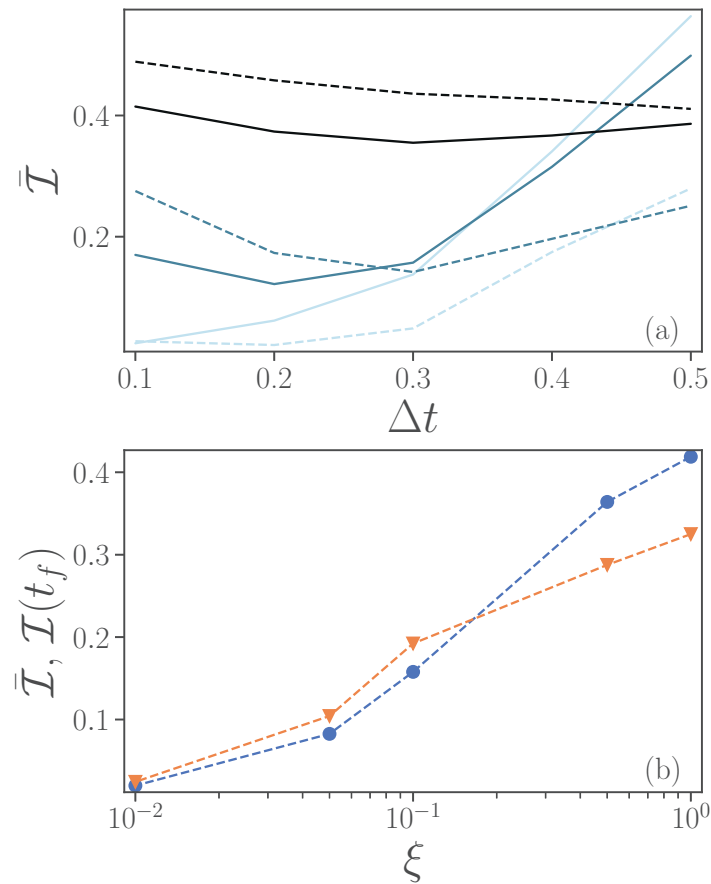


Figure 5. (a) Infidelity averaged over time as a function of time-step size Δt for an evolution from $t = 0$ to a final time $t_f = 2$. Different noise levels $\zeta = 0.01, 0.1, 1$ are represented by lighter to darker colors. (b) Time-averaged (blue circles) and final (orange triangles) infidelity as a function of noise levels. Here, the final time is taken as $t_f = 2$ and we choose $\Delta t = 0.2$. In both panels, results from first-order Trotter implementations are represented by continuous lines, while those from second-order are represented by dashed lines. Parameters $\epsilon = 0.5, \omega = 4, \lambda = 2, \gamma = 1$.

We now consider the open-system dynamics case. The impact of noise on fidelity is depicted in Figure 5b. Here, we show both the average infidelity over the time interval from $t = 0$ to $t = 2$ (blue line with circles), and the infidelity at the final time (orange line with triangles). We consider exclusively a second-order Trotter decomposition and a time-step $\Delta t = 0.2$. Figure 5b indicates a monotonous growth of infidelity with the noise-factor ζ , for the parameters explored.

In Figure 6, we show the infidelity versus time for first-order (solid lines) and second-order (dashed line) Trotterizations, while $\Delta t = 0.2$. We observe that, only for small values of ζ , one would prefer a second-order Trotterization to improve on the fidelity of the states. We note, not shown here, that for $\zeta = 0.01$, the dynamics is almost identical to the noiseless case.

To better understand the role of dissipation, we aim to verify its effect on the accuracy of the simulation. To focus specifically on the role of γ , we consider only a second-order Trotter evolution, a fixed value of $\Delta t = 0.2$ and $\zeta = 0.01$, where the simulation of the quantum computer shows generally better performance compared to levels of higher magnitudes of noise $\zeta = 0.1, 1$. In Figure 7a, we plot the time-averaged infidelity at different values of γ , with (an orange line with circles) and without noise (blue line with triangles). In noiseless simulations, the infidelity increases with γ , while in noisy simulations the infidelity initially reduces to a minimum at $\gamma = 1$. Our understanding is that the dissipation in the exact calculations acts in a similar way as the intrinsic noise on the device, by drawing the system to its ground state and reducing coherence. It can thus be easier

for a lossy quantum hardware to simulate a lossy system compared to a closed system ($\gamma = 0$). However, a system with larger γ also implies further difficulties in the simulations stemming, for example, from Trotterization. It thus occurs that the intrinsic dissipative dynamics can, in some regimes, be better represented on a noisy device.

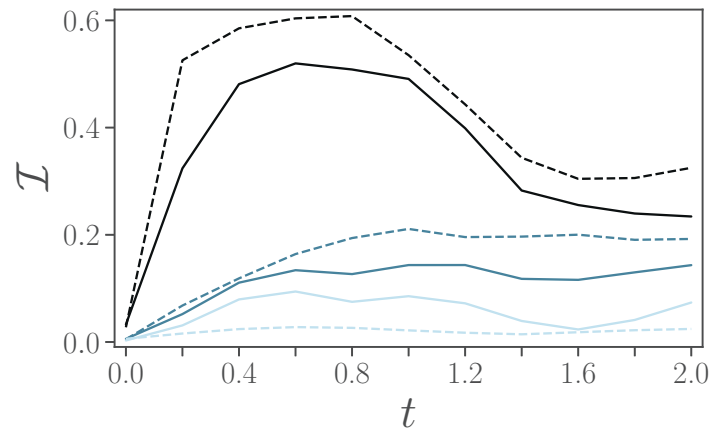


Figure 6. Infidelity as a function of time in open-system simulation in presence of noise. Using first-order Trotter (solid) and second-order Trotter (dashed) at $\Delta t = 0.2$. At noise-factor $\xi = 0.01, 0.1, 1$ (from lighter to darker colors). Other parameters are $\epsilon = 0.5, \omega = 4, \lambda = 2, \gamma = 1$.

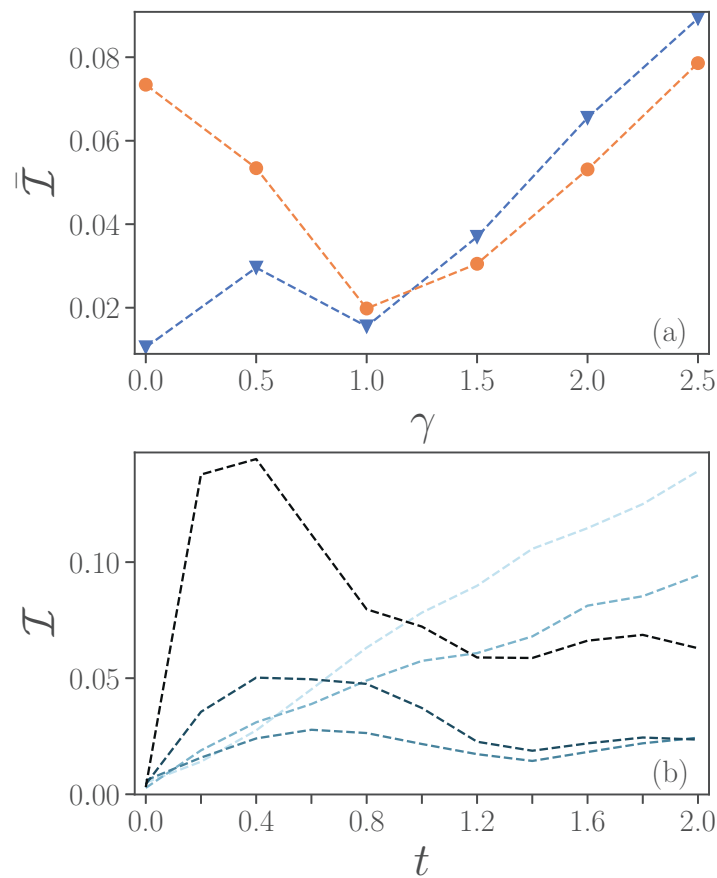


Figure 7. (a) Infidelity averaged over time versus dissipative rate γ with noise $\xi = 0.01$ (orange line with circles) and without noise (blue line with triangles). (b) Infidelity as a function of time $\gamma = 0, 0.5, 1, 1.5, 2, 2.5$ (from lighter to darker colors). Second-order Trotter at $\Delta t = 0.2$ and the other parameters are $\epsilon = 0.5, \omega = 4, \lambda = 2$.

In Figure 7b, we plot the infidelity versus time for different values of the dissipative rate γ . We observe that, for $\gamma \leq 1$, the infidelity tends to increase with time, while for larger values of $\gamma \geq 1.5$, the infidelity can decrease after a maximum at an earlier time $t \approx 0.4$.

Figure 8 shows the average occupation in the harmonic oscillator, panel (a), and the expectation values of $\hat{\sigma}^z$ of the spin, panel (b), versus time. In both panels, the dotted line corresponds to the exact values, whilst the solid and dashed lines to $\zeta = 0.01, 0.1, 1$, respectively, from lighter to darker shades, and solid lines are used for first-order Trotterizations, while dashed lines are used for second-order. For each noise level ζ , we used the Trotterization order which corresponds to the lower fidelity.

The oscillatory evolution of the occupation of the harmonic oscillator is captured, only partially, with the smaller non-zero noise parameter considered $\zeta = 0.01$, panel (a), while the occupation of the harmonic oscillator at $\zeta = 1$ quickly stagnates around a value of 1. Instead, the simpler evolution of $\hat{\sigma}^z$ is also captured fairly well for the different values of ζ , as the simulated dissipation of the spin is closer to the relaxation of the spin-qubit under noise.

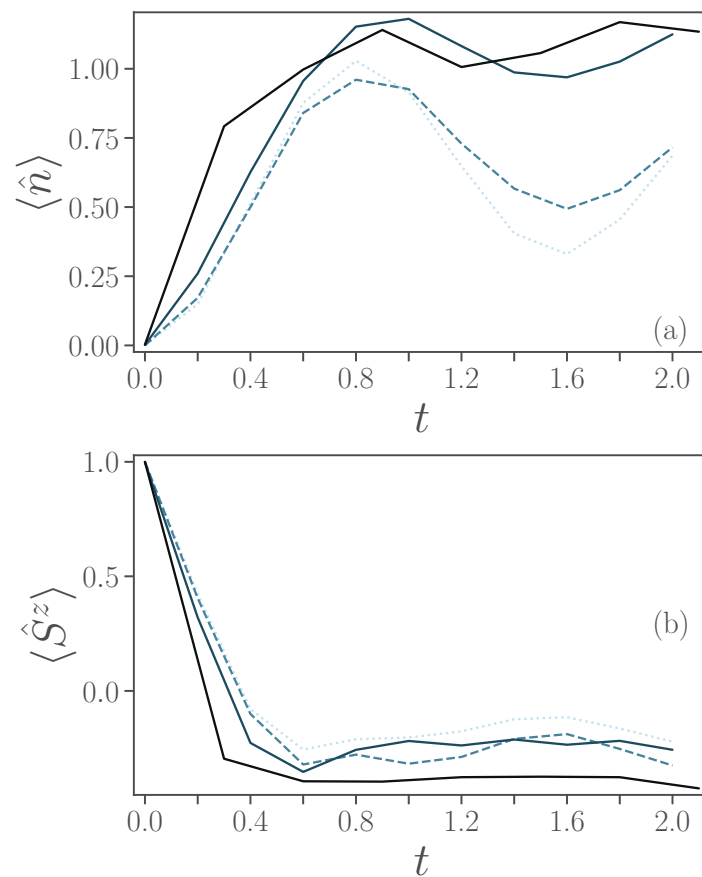


Figure 8. (a) Average bosonic occupation $\langle \hat{n} \rangle$ and (b) $\langle \hat{\sigma}^z \rangle$ as a function of time. Different noise levels $\zeta = 0.01, 0.1, 1$ are presented, respectively, by lighter to darker colors. As a reference, exact simulations are depicted by dotted lines. Results obtained using first-order Trotterization are with solid lines, while second-order with dashed lines. Other parameters are $\epsilon = 0.5, \omega = 4, \lambda = 2$ and $\gamma = 1$.

3.3. Two-Spin System

We here extend the system to two spins to see whether it is possible to the study correlation developing between them through a mediated interaction via the harmonic oscillator, as the two spins do not directly interact with each other. We use the parameters $\epsilon = 0.5, \omega = 6, \lambda = 2$ and $\gamma = 1$. We prepare the initial state in a product state of one spin in the excited state, one in the ground state, and the harmonic oscillator as completely empty. This allows us to observe non-trivial dynamics while still requiring just a few occupied levels of the harmonic oscillator.

As for the single-spin simulations, we first evaluate infidelity in the presence of noise. Simulating two spins requires roughly twice the number of gates as simulating one spin. A single Δt evolution with a first-order Trotter requires 113 single-qubits and 36 CX-Gates, while the second-order Trotter requires 177 single-qubits and 70 CX gates, as can be seen in Figure A1 and Table A1 in Appendix A. Furthermore, in the case of two spins, we find the optimal Trotter time-step Δt to be the same as for the single spin case (not shown).

To study the emerging correlations between the spins mediated by interaction with the photons, we consider the spin–spin correlators

$$\begin{aligned} C^{ZZ} &= \langle \hat{\sigma}_1^z \hat{\sigma}_2^z \rangle - \langle \hat{\sigma}_1^z \rangle \langle \hat{\sigma}_2^z \rangle \\ C^{XX} &= \langle \hat{\sigma}_1^x \hat{\sigma}_2^x \rangle - \langle \hat{\sigma}_1^x \rangle \langle \hat{\sigma}_2^x \rangle. \end{aligned} \tag{9}$$

These connected correlation functions (also called second-order Ursell functions or cumulants) correspond to the covariance in statistics and vanish if and only if $\hat{\sigma}_1^{(\cdot)}$ and $\hat{\sigma}_2^{(\cdot)}$ are statistically independent [71–73].

In Figure 9, we show C^{ZZ} and C^{XX} for, again, $\xi = 0.01, 0.1, 1$ from lighter to darker lines. The solid lines correspond to first-order Trotter and dashed lines to second-order Trotter and these Trotterization orders have been chosen as they result, for the respective amount of noise, in the lowest infidelity. In both panels, the dotted lines correspond to the exact values. The exact case simulations show a build-up in anti-correlation in z direction at $t = 0.4$, before reducing to 0 which can already be observed for $\xi = 0.1$. A correlation in x direction builds up monotonously over time and one would need $\xi = 0.01$ for a clearer signal.

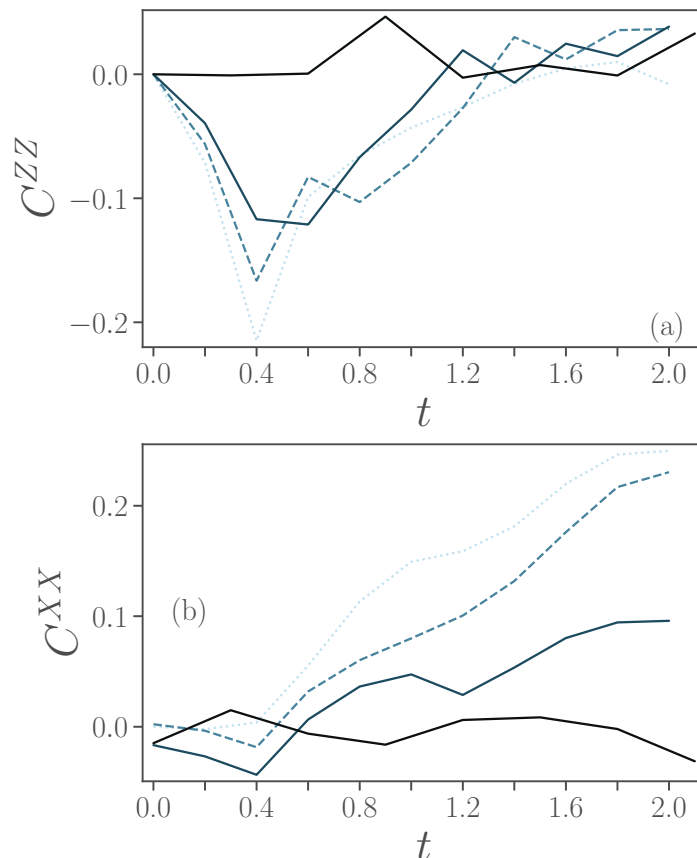


Figure 9. Correlations for the case of two spins: (a) spin- z connected correlation C^{ZZ} (b) spin- x connected correlation C^{XX} as a function of time. Different noise levels $\xi = 0.01, 0.1, 1$ are presented, respectively, by lighter and darker colors. As a reference, exact simulations are depicted by the dotted lines. Results obtained using first-order Trotterization are represented by solid lines, while those using second-order Trotterization are represented by dashed lines. Other parameters are $\epsilon = 0.5$, $\omega = 6$, $\lambda = 2$ and $\gamma = 1$.

In principle, correlations could be observed for a higher number of spins. In practice, the larger number of qubits needed, and their connectivity, would result in an increased number of gates which would limit the fidelity in NISQ devices. We also note that, going from one to two spins, we had to increase ω to keep the higher levels of the harmonic oscillator sparsely populated. If one does not want to increase the number of levels studied for the harmonic oscillator, a similar adjustment, such as decreasing the coupling between the harmonic oscillator and the spins, would be necessary when increasing the number of spins.

4. Conclusions

In this paper, we have studied the feasibility of simulating open spin-boson dynamics on a quantum computer. We used a second-quantization mapping of the bosonic degrees of freedom and Trotterization of the unitary to implement the Hamiltonian. To implement the dissipative dynamics, we used collisions and resets with auxiliary qubits.

We found that, in our parameter regime, the Hamiltonian simulation is the limiting factor to the fidelity. We surveyed optimal Trotterization formulas and time-step sizes depending on the level of noise in the system. We selected the open dissipative rate with the highest fidelity in noisy circuits, and we found that current noise levels in the machine we considered would make such simulations particularly challenging.

Anticipating future improved devices, we ran our simulations on 10% and 1% of current noise levels, and we were able to show that it would be possible to attain much higher fidelities. Furthermore, certain observables could be well represented with larger amounts of noise. Importantly, the simulation of an open system can be more accurate than unitary evolution as the open system dynamics could be closer to how a noisy computer is already affecting a state.

Future developments in noise reduction in the hardware, in post-processing error mitigation, as well as in reducing the number of gates for unitary evolutions can lead to a significant increase in simulation power.

In our system, we have limited the dissipation to the spins. An interesting avenue for future work could be the inclusion of loss in the bosonic degrees of freedom of the cavity, for which additional auxiliary qubits, gates, and connectivity requirements could prove challenging.

Author Contributions: Conceptualization, L.C.K. and D.P.; methodology, D.P.; software, A.B.; validation, A.B., L.C.K. and D.P.; formal analysis, A.B.; investigation, A.B.; resources, L.C.K.; data curation, A.B.; writing—original draft preparation, A.B.; writing—review and editing, D.P.; visualization, A.B.; supervision, D.P.; project administration, L.C.K.; funding acquisition, D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: A.B. acknowledges support from the Ministry of Education of Singapore AcRF MOE Tier-II (Project No. T2MOE2002). L.C.K. and D.P. acknowledge support from the National Research Foundation, Singapore under its QEP2.0 program (NRF2021-QEP2-02-P03).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Encoding of Bosonic Operators Onto Qubits

We will quickly review the d-level-to-qubit mapping we used to encode the bosonic operators as strings of Pauli matrices. The method and different binary encodings are discussed in [31]. The steps can be summarized as:

1. Truncate the infinite-dimensional harmonic oscillator at some level d_{HO}
2. Rewrite each bosonic operator \hat{A} as a sum of level transitions

$$\hat{A} = \sum_{l,l'=0}^{d_{HO}-1} a_{l,l'} |l\rangle \langle l'|, \hat{A} = \{\hat{a}, \hat{a}^\dagger, \hat{a}^\dagger \hat{a}\}$$

3. Assign each level an integer

$$|l\rangle \xrightarrow{\text{integer}} |i\rangle, i \in \mathbb{N}$$

4. Write each integer in binary

$$|i\rangle \xrightarrow{\text{integer-to-bit}} \bigotimes_{m=1}^{Q_B} |b_m\rangle, b_m \in \{0,1\}$$

5. Map each bit pair $|b_m\rangle \langle b'_m|$ to Pauli matrices using

$$\begin{aligned} |0\rangle \langle 0| &= \frac{1}{2}(\mathbb{1} + \hat{\sigma}^z) \\ |1\rangle \langle 1| &= \frac{1}{2}(\mathbb{1} - \hat{\sigma}^z) \\ |0\rangle \langle 1| &= \frac{1}{2}(\hat{\sigma}^x + i\hat{\sigma}^y) = \hat{\sigma}^+ \\ |1\rangle \langle 0| &= \frac{1}{2}(\hat{\sigma}^x - i\hat{\sigma}^y) = \hat{\sigma}^- \end{aligned}$$

The result is that each level transition is written as a string of Pauli operators and each bosonic operator \hat{A} as a sum of N_P Pauli strings

$$\hat{A} = \sum_{k=1}^{N_P} c_k \bigotimes_{j=1}^{Q_B} \hat{\sigma}_{kj}, \hat{\sigma}_{kj} \in \{\mathbb{1}, \hat{\sigma}^x, \hat{\sigma}^y, \hat{\sigma}^z\} \tag{A1}$$

where $Q_B = \lceil \sqrt{d_{HO}} \rceil$ is the number of qubits which encode the bosonic levels ($\lceil \cdot \rceil$ is the ceiling function).

Appendix A.1. Gate Requirements

When writing the integers in binary in step 4, different integer-to-bit encodings result in different Pauli strings and ultimately in a different representation of the Hamiltonian. While the representations of the Hamiltonian are theoretically equivalent, they come with different gate counts and thus result in different performances on noise devices.

As integer-to-bit encodings, we considered standard binary and gray code, since both of these are compact, i.e., they require the minimum amount of qubits. Table A1 shows the gates required to evolve one time-step of the Trotterized unitary $e^{-i\hat{H}_{SB}\Delta t}$ and dissipation on the ibmq_jakarta device. This includes additional CX-Gates to implement any necessary SWAP-Gates due to limited qubit connectivity (Figure A1). For our Hamiltonian \hat{H}_{SB} , gray code yielded less gates than standard binary in all cases, which is why we used gray code throughout the main text.

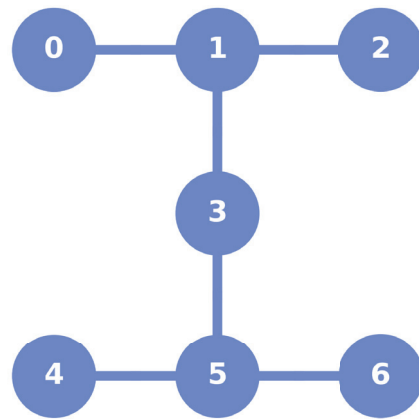


Figure A1. Qubit connectivity of the used ibmq_jakarta device.

Table A1. Gate counts, both for CX-Gate and single-qubit gates, to evolve one time-step of master Equation (2) on the Jakarta device.

N_S	d_{HO}	Trotter order	Standard Binary		Gray Code	
			Single	CX	Single	CX
1	4	First	53	21	94	43
1	4	Second	94	34	75	28
1	8	First	156	66	122	60
1	8	Second	282	124	191	107
2	4	First	106	37	122	36
2	4	Second	191	65	168	74
2	8	First	270	139	200	156
2	8	Second	496	272	409	255

Appendix A.2. Mapped Hamiltonian

After the mapping of the harmonic oscillator to qubits, the Hamiltonian (Equation (1)) is written as a sum of Pauli strings h_k . The unitary $e^{\sum_k h_k}$ is then Trotterized (Equation (3)). The mapped Hamiltonian $\hat{H}_{SB} = \sum_k h_k$ we implemented for the main text reads explicitly

$$\begin{aligned} \hat{H}_{SB} = & -\sqrt{2}\sigma^x_0\sigma^x_1\sigma^z_2 + \sqrt{2}\sigma^x_0\sigma^x_1 \\ & + (1 - \sqrt{3})\sigma^x_0\sigma^x_2\sigma^z_1 + (1 + \sqrt{3})\sigma^x_0\sigma^x_2 \\ & + \frac{1}{4}\sigma^x_0 - \frac{1}{2}\sigma^z_0 - 2\sigma^z_1\sigma^z_2 - 4\sigma^z_1 \end{aligned} \tag{A2}$$

for the single spin case, and

$$\begin{aligned} \hat{H}_{SB} = & -\sqrt{2}\sigma^x_0\sigma^x_1\sigma^z_2 + \sqrt{2}\sigma^x_0\sigma^x_1 \\ & + (1 - \sqrt{3})\sigma^x_0\sigma^x_2\sigma^z_1 + (1 + \sqrt{3})\sigma^x_0\sigma^x_2 \\ & + \frac{1}{4}\sigma^x_0 - \sqrt{2}\sigma^x_1\sigma^x_3\sigma^z_2 + \sqrt{2}\sigma^x_1\sigma^x_3 \\ & + (1 - \sqrt{3})\sigma^x_2\sigma^x_3\sigma^z_1 + (1 + \sqrt{3})\sigma^x_2\sigma^x_3 \\ & + \frac{1}{4}\sigma^x_3 - \frac{1}{2}\sigma^z_0 - 3\sigma^z_1\sigma^z_2 - 6\sigma^z_1 \\ & - \frac{1}{2}\sigma^z_3 \end{aligned} \tag{A3}$$

for two spins case. Each term constitutes one of the h_k in Equations (3) and (4).

Appendix B. Noise Model

Qiskit supplies noise models based on device properties measured during calibration. In order to simulate an improved future device, we engineer our noise from an identical model, but from lower noise levels.

The noise model contains three error sources [70]: (i) thermal relaxation (relaxation and dephasing); (ii) depolarizing (Pauli) error; and (iii) readout (measurement) error. At every gate, first the thermal relaxation and then the depolarizing error is applied. The strength of the depolarizing error is calculated backwards, to reach a target ‘gate error’ when combined with the thermal relaxation. Details can be found at [74].

Appendix B.1. Error Sources

Appendix B.1.1. Thermal Relaxation Error

Thermal relaxation is defined by the qubit-specific parameters T_1 time, T_2 time, qubit frequency f_{Qubit} and qubit temperature \mathcal{T}_{Qubit} . The thermal error channel is then given time to act according to a gate-dependent gate time. For two-qubit-gates, the error is simply the tensor product between two single-qubit channels.

T_1 is qubit-specific time until relaxation, i.e., to decay from the excited state to the ground state. T_2 qubit-specific coherence time, or time until dephasing. The qubit frequency f_{Qubit} is the difference in energy between the ground and excited states. The qubit temperature \mathcal{T}_{Qubit} is assumed to be 0 in Qiskit’s and our noise models.

The qubit frequency and temperature enter only via the excited state population. If $f_{Qubit} \rightarrow \infty$ or $\mathcal{T}_{Qubit} = 0$, the excited state population is 0. Since $\mathcal{T}_{Qubit} = 0$ in our models, both the frequency and temperature can effectively be ignored as parameters.

For $T_2 < T_1$, thermal relaxation is most straightforwardly described by (assuming the device to be at 0 temperature)

$$K_{T_0} = \sqrt{\mathcal{P}_1} \mathbb{1}, \tag{A4}$$

$$K_{T_1} = \sqrt{\mathcal{P}_Z} \hat{\sigma}^z, K_{T_2} = \sqrt{\mathcal{P}_{reset}} |\downarrow\rangle \langle \downarrow| \tag{A5}$$

$$\mathcal{E}_T(\hat{\rho}) = \sum_{i=10}^2 K_{T_k} \hat{\rho} K_{T_k}^\dagger \tag{A6}$$

It is composed of the probabilities of a phase-flip \mathcal{P}_Z , a reset to the ground state \mathcal{P}_{reset} , or for nothing to happen \mathcal{P}_1 . The probabilities $\mathcal{P}_Z, \mathcal{P}_{reset}$ are calculated from T_1, T_2 and the gate time t_{Gate} .

$$\mathcal{P}_{reset} = 1 - \mathcal{P}_{T_1} = 1 - e^{-t_{Gate} \cdot \frac{1}{T_1}} \tag{A7}$$

$$\mathcal{P}_Z = (1 - \mathcal{P}_{reset}) \left(1 - \frac{\mathcal{P}_{T_2}}{\mathcal{P}_{T_1}} \right) / 2 \tag{A8}$$

$$= (1 - \mathcal{P}_{reset}) \left(1 - e^{-t_{Gate} \cdot \left(\frac{1}{T_2} - \frac{1}{T_1} \right)} \right) / 2 \tag{A9}$$

$$\mathcal{P}_1 = 1 - \mathcal{P}_Z - \mathcal{P}_{reset}. \tag{A10}$$

If $2T_1 \geq T_2 > T_1$ thermal relaxation has to be described by its Choi matrix

$$\hat{\rho} \rightarrow \mathcal{E}_T(\hat{\rho}) = tr_1[C(\hat{\rho}^T \otimes I)] \tag{A11}$$

$$C_{\mathcal{E}_T} = \begin{pmatrix} 1 & 0 & 0 & \mathcal{P}_{T_2} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{P}_{reset} & 0 \\ \mathcal{P}_{T_2} & 0 & 0 & 1 - \mathcal{P}_{reset} \end{pmatrix} \tag{A12}$$

which can also be used if $T_2 < T_1$ to compute the process fidelity in Equation (A21).

At the time of writing, all qubits on the Jakarta hardware satisfied $T_2 < T_1$. This is not necessarily the case for all devices provided by IBM or in general.

Appendix B.1.2. Depolarizing Error

The depolarizing noise (or Pauli) channel is composed of either a bit-flip ($\hat{\sigma}^x$), a phase-flip ($\hat{\sigma}^z$) or both at the same time ($\hat{\sigma}^y$), all with equal probability [70].

$$\hat{\rho} \rightarrow \mathcal{E}_D(\hat{\rho}) = \sum_{i=1}^3 K_{\mathcal{P}_i} \hat{\rho} K_{\mathcal{P}_i}^\dagger \tag{A13}$$

$$K_{\mathcal{P}_0} = \sqrt{1 - \mathcal{P}_D} \mathbb{1}, K_{\mathcal{P}_1} = \sqrt{\frac{\mathcal{P}_D}{3}} \hat{\sigma}^x \tag{A14}$$

$$K_{\mathcal{P}_2} = \sqrt{\frac{\mathcal{P}_D}{3}} \hat{\sigma}^y, K_{\mathcal{P}_3} = \sqrt{\frac{\mathcal{P}_D}{3}} \hat{\sigma}^z \tag{A15}$$

Gate Infidelity

The probability of a depolarizing error is calculated from the target gate infidelity \mathcal{I}_{Gate} , and the infidelity due to thermal relaxation \mathcal{I}_T .

$$\mathcal{I}_D = \mathcal{I}_{Gate} - \mathcal{I}_T \tag{A16}$$

The target gate infidelity is given as a parameter, while \mathcal{I}_T has to be calculated as

$$\mathcal{F}_T = 1 - \mathcal{I}_T \tag{A17}$$

$$= \mathcal{F}_{avg}(\mathcal{E}_T, U) \tag{A18}$$

$$= \int d\psi \langle \psi | U^\dagger \mathcal{E}_T(|\psi\rangle\langle\psi|) U | \psi \rangle \tag{A19}$$

$$= \frac{d\mathcal{I}_{pro}(\mathcal{E}_T, U) + 1}{d + 1} \tag{A20}$$

where $\mathcal{I}_{pro}(\mathcal{E}_T, U)$ is the process fidelity of the input quantum channel \mathcal{E}_T with a target unitary U , and d is the dimension of the channel.

$$\mathcal{I}_{pro}(\mathcal{E}_T, \mathcal{F}) = F(C_{\mathcal{E}_T}/d, \rho_{\mathcal{F}}) \tag{A21}$$

where \mathcal{F} is the state fidelity as defined in the main text

$$\mathcal{F}(\rho_1, \rho_2) = \left(\text{Tr} \left[\sqrt{\sqrt{\rho_1} \rho_2 \sqrt{\rho_1}} \right] \right)^2 \tag{A22}$$

$C_{\mathcal{E}_T}/d$ is the normalized Choi matrix for the channel \mathcal{E}_T , and d is the input dimension of \mathcal{E}_T .

Importantly, for our reduced-noise models, the infidelity from thermal relaxation \mathcal{I}_T is linear in the gate time t_{Gate} . Thus, when we rescale $\mathcal{I}_{Gate} \rightarrow \xi \cdot \mathcal{I}_{Gate}$, $t_{Gate} \rightarrow \xi \cdot t_{Gate}$, we indirectly scale $\mathcal{I}_D \rightarrow \xi \cdot \mathcal{I}_D$, $\mathcal{I}_T \rightarrow \xi \cdot \mathcal{I}_T$. This way, the relative contribution of the error channels $\mathcal{I}_D/\mathcal{I}_T$ to the infidelity remains unchanged.

Depolarizing Error Probability

If we write the depolarizing error in terms of the identity and the complete depolarizing channel D , we can rewrite the gate fidelity

$$\mathcal{E}_D = (1 - \mathcal{P}_D) \cdot \mathbb{1} + \mathcal{P}_D \cdot D \tag{A23}$$

$$\mathcal{F}_{gate} = 1 - \mathcal{I}_{Gate} \tag{A24}$$

$$= \mathcal{F}(\mathcal{E}_D \cdot \mathcal{E}_T) \tag{A25}$$

$$= (1 - \mathcal{P}_D)\mathcal{F}_T + \mathcal{P}_D \cdot \mathcal{F}_D \tag{A26}$$

$$= \mathcal{F}_T - \mathcal{P}_D \cdot (d \cdot \mathcal{F}_T - 1)/d \tag{A27}$$

where $d = 2^{qubits}$ is the dimensionality of the gate. From this, the solution to the depolarizing error probability is

$$\mathcal{P}_D = d(\mathcal{F}_T - \mathcal{F}_{gate}) / (d \cdot \mathcal{F}_T - 1) \tag{A28}$$

$$= d(\mathcal{I}_{Gate} - \mathcal{I}_T) / (d \cdot \mathcal{F}_T - 1) \tag{A29}$$

More details can be found at [74].

Appendix B.1.3. Measurement Error

A measurement error is equivalent to a bit-flip $\hat{\sigma}^x$ followed by a noiseless readout [70]. The probability of the readout error \mathcal{P}_R is given by the probability $P(n|m)$ of recording a noisy measurement outcome as n , given that the true measurement outcome is m .

$$K_{R_0} = \sqrt{1 - \mathcal{P}_R}\mathbb{1}, K_{R_1} = \sqrt{\mathcal{P}_R}\hat{\sigma}^x \tag{A30}$$

$$\mathcal{P}_R = \sum_{n \neq m} P(n|m) \tag{A31}$$

where n, m run over all qubits, in the case of two qubits $n, m \in \{00, 01, 10, 11\}$. See [74] for further details.

Appendix B.1.4. Error Sources in the Reference Device

Given the three error sources, one can ask which error source causes the dominant contribution to the noise in our results. As we use measurement error mitigation and it is independent of the circuit depth, we will ignore the measurement error. Instead, we focus on the ratio of thermal and depolarizing errors in contributing to the infidelity, $\mathcal{I}_T/\mathcal{I}_D$. To give a rough estimation, we assume all gates $g \in \{CNOT, RZ, SX, X\}$ and all qubits q are used equally often, and average over both.

$$\mathcal{I}_T/\mathcal{I}_D = \frac{1}{N_q} \sum_{q=1}^{N_q=7} \left(\frac{1}{N_g} \sum_g^{N_g=4} \left(\frac{\mathcal{I}_T(q, g)}{\mathcal{I}_D(q, g)} \right) \right) \tag{A32}$$

We calculate $\mathcal{I}_T(q, g)$ and $\mathcal{I}_D(q, g)$ using Equations (A17) and (A16), respectively, and obtain the current calibration data from IBM. At the time of writing, the result for the Jakarta device is $\mathcal{I}_T/\mathcal{I}_D = 15.4$. We conclude that thermal relaxation is the main source of infidelity in our simulations, by one order of magnitude compared to depolarization.

Appendix B.1.5. Calibration Data

We base our reduced-noise models on the same hardware that we run our full-noise circuits on, the 7 qubit IBMQ Jakarta device.

At the time of writing, the calibration data are:

Processor: Falcon r5.11H, V1.1.0

Avg. CX-Gate error: $1.109e^{-2}$

Avg. readout error: $3.349e^{-2}$

Avg. T_1 : 139.01 us
 Avg. T_2 : 44.82 us
 Avg. gate time: 454.095 ns
 Avg. qubit frequency: 5.08 GHz
 Avg. qubit anharmonicity -0.329 GHz
 For more details, see [75].

Appendix C. Gate Definition

Some of the gates used are defined here. A controlled operation CO is defined as

$$CO(\theta) = I \otimes |\downarrow\rangle\langle\downarrow| + O(\theta) \otimes |\uparrow\rangle\langle\uparrow|, \tag{A33}$$

where the operation O is a X gate in case of the CX -Gate, or a rotation around the y -axis R_Y or z axis R_Z . R_Y and R_Z are, respectively, defined as

$$R_Y(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \tag{A34}$$

$$R_Z(\lambda) = \exp\left(-i\frac{\lambda}{2}Z\right) = \begin{pmatrix} e^{-i\frac{\lambda}{2}} & 0 \\ 0 & e^{i\frac{\lambda}{2}} \end{pmatrix}. \tag{A35}$$

Furthermore, the \sqrt{X} gate is given by

$$\sqrt{X} = \frac{1}{2} \begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}. \tag{A36}$$

Appendix D. Transpiled Circuits

The amplitude damping circuit as in Figure 2 uses gates which are not available on the quantum computer we were using. Instead, the IBM Jakarta device uses the gate set {CNOT, ID, RZ, SX, X}. The amplitude damping circuit, in terms of these gates and as it was implemented on the hardware, is in Figure A2.

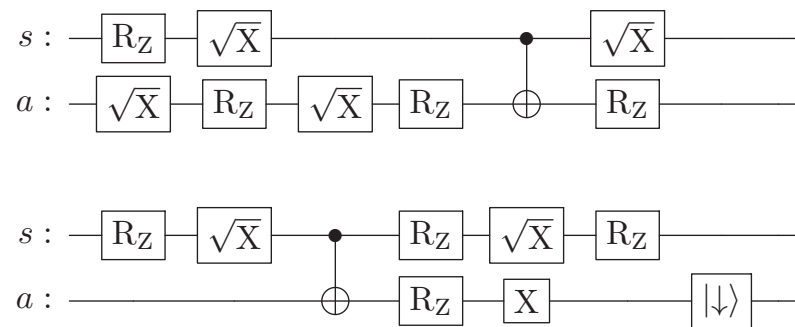


Figure A2. The dissipation circuit represented in Figure 2 in terms of the gates available on the IBM Jakarta device. Both lines for qubits s, a continue from the first row to the second.

References

1. Lloyd, S. Universal Quantum Simulators. *Science* **1996**, *273*, 1073–1078. [CrossRef] [PubMed]
2. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*; Cambridge University Press: Cambridge, UK, 2010. [CrossRef]
3. Whitfield, J.D.; Biamonte, J.; Aspuru-Guzik, A. Simulation of Electronic Structure Hamiltonians Using Quantum Computers. *Mol. Phys.* **2011**, *109*, 735–750. [CrossRef]
4. Wiebe, N.; Berry, D.W.; Høyer, P.; Sanders, B.C. Simulating quantum dynamics on a quantum computer. *J. Phys. A Math. Theor.* **2011**, *44*, 445308. [CrossRef]
5. Tacchino, F.; Chiesa, A.; Carretta, S.; Gerace, D. Quantum Computers as Universal Quantum Simulators: State-of-the-Art and Perspectives. *Adv. Quantum Technol.* **2019**, *3*, 1900052. [CrossRef]

6. Jaderberg, B.; Eisfeld, A.; Jaksch, D.; Mostame, S. Recompilation-enhanced simulation of electron–phonon dynamics on IBM quantum computers. *New J. Phys.* **2022**, *24*, 093017. [CrossRef]
7. Leggett, A.J.; Chakravarty, S.; Dorsey, A.T.; Fisher, M.P.A.; Garg, A.; Zwerger, W. Dynamics of the dissipative two-state system. *Rev. Mod. Phys.* **1987**, *59*, 1–85. [CrossRef]
8. Weiss, U. *Quantum Dissipative Systems*; World Scientific: Singapore, 2011. [CrossRef]
9. Xu, D.; Schulten, K. Coupling of protein motion to electron transfer in a photosynthetic reaction center: Investigating the low temperature behavior in the framework of the spin–Boson model. *Chem. Phys.* **1994**, *182*, 91–117. [CrossRef]
10. Renger, T.; Marcus, R.A. On the relation of protein dynamics and exciton relaxation in pigment–protein complexes: An estimation of the spectral density and a theory for the calculation of optical spectra. *J. Chem. Phys.* **2002**, *116*, 9997–10019. [CrossRef]
11. Fleming, G.R.; Cho, M. Chromophore solvent dynamics. *Annu. Rev. Phys. Chem.* **1996**, *47*, 109–134. [CrossRef]
12. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
13. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J.S.; Menke, T.; et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004. [CrossRef]
14. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.H.; Zhou, X.Q.; Love, P.J.; Aspuru-Guzik, A.; O’Brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213. [CrossRef] [PubMed]
15. Farhi, E.; Goldstone, J.; Gutmann, S. A Quantum Approximate Optimization Algorithm. *arXiv* **2014**, arXiv:1411.4028. [CrossRef]
16. Erbanni, R.; Bharti, K.; Kwek, L.C.; Poletti, D. NISQ algorithm for the matrix elements of a generic observable. *arXiv* **2022**, arXiv:2205.10058. [CrossRef]
17. Li, Y.; Benjamin, S.C. Efficient Variational Quantum Simulator Incorporating Active Error Minimization. *Phys. Rev. X* **2017**, *7*, 021050. [CrossRef]
18. Temme, K.; Bravyi, S.; Gambetta, J.M. Error Mitigation for Short-Depth Quantum Circuits. *Phys. Rev. Lett.* **2017**, *119*, 180509. [CrossRef]
19. Endo, S.; Cai, Z.; Benjamin, S.C.; Yuan, X. Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation. *J. Phys. Soc. Jpn.* **2021**, *90*, 032001. [CrossRef]
20. Di Paolo, A.; Barkoutsos, P.K.; Tavernelli, I.; Blais, A. Variational Quantum Simulation of Ultrastrong Light-Matter Coupling. *Phys. Rev. Res.* **2020**, *2*, 033364. [CrossRef]
21. Miessen, A.; Ollitrault, P.J.; Tavernelli, I. Quantum algorithms for quantum dynamics: A performance study on the spin-boson model. *Phys. Rev. Res.* **2021**, *3*, 043212. [CrossRef]
22. Fitzpatrick, N.; Apel, H.; Ramo, D.M. Evaluating low-depth quantum algorithms for time evolution on fermion-boson systems. *arXiv* **2021**, arXiv:2106.03985. [CrossRef]
23. Kamakari, H.; Sun, S.N.; Motta, M.; Minnich, A.J. Digital quantum simulation of open quantum systems using quantum imaginary time evolution. *PRX Quantum* **2022**, *3*, 010320. [CrossRef]
24. McArdle, S.; Jones, T.; Endo, S.; Li, Y.; Benjamin, S.C.; Yuan, X. Variational ansatz-based quantum simulation of imaginary time evolution. *Npj Quantum Inf.* **2019**, *5*, 75. [CrossRef]
25. Endo, S.; Sun, J.; Li, Y.; Benjamin, S.; Yuan, X. Variational quantum simulation of general processes. *Phys. Rev. Lett.* **2020**, *125*, 010501. [CrossRef] [PubMed]
26. Yoshioka, N.; Nakagawa, Y.O.; Mitarai, K.; Fujii, K. Variational quantum algorithm for nonequilibrium steady states. *Phys. Rev. Res.* **2020**, *2*, 043289. [CrossRef]
27. Liu, H.Y.; Sun, T.P.; Wu, Y.C.; Guo, G.P. Variational Quantum Algorithms for the Steady States of Open Quantum Systems. *Chin. Phys. Lett.* **2021**, *38*, 080301. [CrossRef]
28. Bharti, K.; Haug, T. Quantum-assisted simulator. *Phys. Rev. A* **2021**, *104*, 042418. [CrossRef]
29. Macridin, A.; Spentzouris, P.; Amundson, J.; Harnik, R. Digital quantum computation of fermion-boson interacting systems. *Phys. Rev. A* **2018**, *98*, 042312. [CrossRef]
30. Somma, R.D.; Ortiz, G.; Knill, E.H.; Gubernatis, J. Quantum simulations of physics problems. In Proceedings of the Quantum Information and Computation, SPIE, Orlando, FL, USA, 21–25 April 2003; Volume 5105, pp. 96–103. [CrossRef]
31. Sawaya, N.P.D.; Menke, T.; Kyaw, T.H.; Johri, S.; Aspuru-Guzik, A.; Guerreschi, G.G. Resource-efficient digital quantum simulation of d-level systems for photonic, vibrational, and spin-s Hamiltonians. *NPJ Quantum Inf.* **2020**, *6*, 49. [CrossRef]
32. Hu, Z.; Xia, R.; Kais, S. A quantum algorithm for evolving open quantum dynamics on quantum computing devices. *Sci. Rep.* **2020**, *10*, 3301. [CrossRef]
33. Schlimgen, A.W.; Head-Marsden, K.; Sager, L.M.; Narang, P.; Mazziotti, D.A. Quantum Simulation of Open Quantum Systems Using a Unitary Decomposition of Operators. *Phys. Rev. Lett.* **2021**, *127*, 270503. [CrossRef]
34. Udayakumar, P.; Kumar-Eslami, P. Kraus operator formalism for quantum multiplexer operations for arbitrary two-qubit mixed states. *Quantum Inf. Process.* **2019**, *18*, 361. [CrossRef]
35. García-Pérez, G.; Rossi, M.A.C.; Maniscalco, S. IBM Q Experience as a versatile experimental testbed for simulating open quantum systems. *Npj Quantum Inf.* **2020**, *6*, 1–10. [CrossRef]
36. Wei, S.J.; Ruan, D.; Long, G.L. Duality quantum algorithm efficiently simulates open quantum systems. *Sci. Rep.* **2016**, *6*, 30727. [CrossRef] [PubMed]
37. Cleve, R.; Wang, C. Efficient Quantum Algorithms for Simulating Lindblad Evolution. *arXiv* **2016**, arXiv:1612.09512.

38. Jo, M.; Kim, M. Simulating open quantum many-body systems using optimised circuits in digital quantum simulation. *arXiv* **2022**, arXiv:2203.14295.
39. Ciccarello, F.; Lorenzo, S.; Giovannetti, V.; Palma, G.M. Quantum collision models: Open system dynamics from repeated interactions. *Phys. Rep.* **2022**, *954*, 1–70. [CrossRef]
40. Algaba, M.G.; Ponce-Martinez, M.; Munuera-Javaloy, C.; Pina-Canelles, V.; Thapa, M.; Taketani, B.G.; Leib, M.; de Vega, I.; Casanova, J.; Heimonen, H. Co-Design quantum simulation of nanoscale NMR. *arXiv* **2022**, arXiv:2202.05792. [CrossRef]
41. Rabi, I.I. On the Process of Space Quantization. *Phys. Rev.* **1936**, *49*, 324–328. [CrossRef]
42. Rabi, I.I. Space Quantization in a Gyating Magnetic Field. *Phys. Rev.* **1937**, *51*, 652–654. [CrossRef]
43. Bloch, F.; Siegert, A. Magnetic Resonance for Nonrotating Fields. *Phys. Rev.* **1940**, *57*, 522–527. [CrossRef]
44. Jaynes, E.; Cummings, F. Comparison of quantum and semiclassical radiation theories with application to the beam maser. *Proc. IEEE* **1963**, *51*, 89–109. [CrossRef]
45. Cummings, F.W. Reminiscing about thesis work with ET Jaynes at Stanford in the 1950s. *J. Phys. B At. Mol. Opt. Phys.* **2013**, *46*, 220202. [CrossRef]
46. Xie, Q.; Zhong, H.; Batchelor, M.T.; Lee, C. The quantum Rabi model: Solution and dynamics. *J. Phys. A Math. Theor.* **2017**, *50*, 113001. [CrossRef]
47. Forn-Díaz, P.; Lisenfeld, J.; Marcos, D.; García-Ripoll, J.J.; Solano, E.; Harmans, C.J.P.M.; Mooij, J.E. Observation of the Bloch-Siegert Shift in a Qubit-Oscillator System in the Ultrastrong Coupling Regime. *Phys. Rev. Lett.* **2010**, *105*, 237001. [CrossRef] [PubMed]
48. Niemczyk, T.; Deppe, F.; Huebl, H.; Menzel, E.P.; Hocke, F.; Schwarz, M.J.; Garcia-Ripoll, J.J.; Zueco, D.; Hümmer, T.; Solano, E.; et al. Circuit quantum electrodynamics in the ultrastrong-coupling regime. *Nat. Phys.* **2010**, *6*, 772–776. [CrossRef]
49. Braumüller, J.; Marthaler, M.; Schneider, A.; Stehli, A.; Rotzinger, H.; Weides, M.; Ustinov, A.V. Analog quantum simulation of the Rabi model in the ultra-strong coupling regime. *Nat. Commun.* **2017**, *8*, 779. [CrossRef] [PubMed]
50. Forn-Díaz, P.; García-Ripoll, J.J.; Peropadre, B.; Orgiazzi, J.L.; Yurtalan, M.A.; Belyansky, R.; Wilson, C.M.; Lupascu, A. Ultrastrong coupling of a single artificial atom to an electromagnetic continuum in the nonperturbative regime. *Nat. Phys.* **2016**, *13*, 39–43. [CrossRef]
51. Yoshihara, F.; Fuse, T.; Ashhab, S.; Kakuyanagi, K.; Saito, S.; Semba, K. Superconducting qubit-oscillator circuit beyond the ultrastrong-coupling regime. *Nat. Phys.* **2016**, *13*, 44–47. [CrossRef]
52. Langford, N.K.; Sagastizabal, R.; Kounalakis, M.; Dickel, C.; Bruno, A.; Luthi, F.; Thoen, D.J.; Endo, A.; DiCarlo, L. Experimentally simulating the dynamics of quantum light and matter at deep-strong coupling. *Nat. Commun.* **2017**, *8*, 1715. [CrossRef]
53. Lv, D.; An, S.; Liu, Z.; Zhang, J.N.; Pedernales, J.S.; Lamata, L.; Solano, E.; Kim, K. Quantum Simulation of the Quantum Rabi Model in a Trapped Ion. *Phys. Rev. X* **2018**, *8*, 021027. [CrossRef]
54. Crespi, A.; Longhi, S.; Osellame, R. Photonic Realization of the Quantum Rabi Model. *Phys. Rev. Lett.* **2012**, *108*, 163601. [CrossRef] [PubMed]
55. Todorov, Y.; Andrews, A.M.; Sagnes, I.; Colombelli, R.; Klang, P.; Strasser, G.; Sirtori, C. Strong Light-Matter Coupling in Subwavelength Metal-Dielectric Microcavities at Terahertz Frequencies. *Phys. Rev. Lett.* **2009**, *102*, 186402. [CrossRef] [PubMed]
56. Günter, G.; Anappara, A.A.; Hees, J.; Sell, A.; Biasiol, G.; Sorba, L.; Liberato, S.D.; Ciuti, C.; Tredicucci, A.; Leitenstorfer, A.; et al. Sub-cycle switch-on of ultrastrong light-matter interaction. *Nature* **2009**, *458*, 178–181. [CrossRef] [PubMed]
57. Lindblad, G. On the generators of quantum dynamical semigroups. *Commun. Math. Phys.* **1976**, *48*, 119–130. [CrossRef]
58. Gorini, V. Completely positive dynamical semigroups of N-level systems. *J. Math. Phys.* **1976**, *17*, 821. [CrossRef]
59. Ritsch, H.; Domokos, P.; Brennecke, F.; Esslinger, T. Cold atoms in cavity-generated dynamical optical potentials. *Rev. Mod. Phys.* **2013**, *85*, 553–601. [CrossRef]
60. Reiserer, A.; Rempe, G. Cavity-based quantum networks with single atoms and optical photons. *Rev. Mod. Phys.* **2015**, *87*, 1379–1418. [CrossRef]
61. Fabre, C.; Sandoghdar, V.; Treps, N.; Cugliandolo, L.F. (Eds.) *Quantum Optics and Nanophotonics*; Oxford University Press: Oxford, UK, 2017. [CrossRef]
62. Di Matteo, O.; McCoy, A.; Gysbers, P.; Miyagi, T.; Woloshyn, R.M.; Navrátil, P. Improving Hamiltonian encodings with the Gray code. *Phys. Rev. A* **2021**, *103*, 042405. [CrossRef]
63. Hatano, N.; Suzuki, M. Finding Exponential Product Formulas of Higher Orders. In *Quantum Annealing and Other Optimization Methods*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 37–68. [CrossRef]
64. Berry, D.W.; Ahokas, G.; Cleve, R.; Sanders, B.C. Efficient Quantum Algorithms for Simulating Sparse Hamiltonians. *Commun. Math. Phys.* **2006**, *270*, 359–371. [CrossRef]
65. Aleksandrowicz, G.; Alexander, T.; Barkoutsos, P.; Bello, L.; Ben-Haim, Y.; Bucher, D.; Cabrera-Hernández, F.J.; Carballo-Franquis, J.; Chen, A.; Chen, C.F.; et al. *Qiskit: An Open-Source Framework for Quantum Computing*; CERN: Geneva, Switzerland, 2021. [CrossRef]
66. Karevski, D.; Platini, T. Quantum Nonequilibrium Steady States Induced by Repeated Interactions. *Phys. Rev. Lett.* **2009**, *102*, 207207. [CrossRef] [PubMed]
67. Jozsa, R. Fidelity for Mixed Quantum States. *J. Mod. Opt.* **1994**, *41*, 2315–2323. [CrossRef]
68. Johansson, J.; Nation, P.; Nori, F. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **2013**, *184*, 1234–1240. [CrossRef]

69. Bravyi, S.; Sheldon, S.; Kandala, A.; Mckay, D.C.; Gambetta, J.M. Mitigating measurement errors in multiqubit experiments. *Phys. Rev. A* **2021**, *103*, 042605. [CrossRef]
70. Georgopoulos, K.; Emary, C.; Zuliani, P. Modeling and simulating the noisy behavior of near-term quantum computers. *Phys. Rev. A* **2021**, *104*, 062432. [CrossRef]
71. Ursell, H.D. The evaluation of Gibbs' phase-integral for imperfect gases. *Math. Proc. Camb. Philos. Soc.* **1927**, *23*, 685–697. [CrossRef]
72. Percus, J.K. Correlation inequalities for Ising spin lattices. *Commun. Math. Phys.* **1975**, *40*, 283–308. [CrossRef]
73. Shlosman, S.B. Signs of the Ising model Ursell functions. *Commun. Math. Phys.* **1986**, *102*, 679–686. [CrossRef]
74. IBM. *Qiskit Dokumentation*; IBM: Armonk, NY, USA, 2022.
75. IBM. *IBMQ Devices*; IBM: Armonk, NY, USA, 2022.

Quantum Algorithm for Variant Maximum Satisfiability [†]

Abdirahman Alasow *, Peter Jin and Marek Perkowski *

Department of Electrical & Computer Engineering, Portland State University, Portland, OR 97207, USA

* Correspondence: alasow@pdx.edu (A.A.); mperkows@ee.pdx.edu (M.P.)

[†] This paper is an extended version of our paper published in the IEEE 52nd International Symposium on Multiple-Valued Logic (ISMVL 2022) 18–20 May 2022.

Abstract: In this paper, we proposed a novel quantum algorithm for the maximum satisfiability problem. Satisfiability (SAT) is to find the set of assignment values of input variables for the given Boolean function that evaluates this function as TRUE or prove that such satisfying values do not exist. For a POS SAT problem, we proposed a novel quantum algorithm for the maximum satisfiability (MAX-SAT), which returns the maximum number of OR terms that are satisfied for the SAT-unsatisfiable function, providing us with information on how far the given Boolean function is from the SAT satisfaction. We used Grover's algorithm with a new block called quantum counter in the oracle circuit. The proposed circuit can be adapted for various forms of satisfiability expressions and several satisfiability-like problems. Using the quantum counter and mirrors for SAT terms reduces the need for ancilla qubits and realizes a large Toffoli gate that is then not needed. Our circuit reduces the number of ancilla qubits for the terms T of the Boolean function from T of ancilla qubits to $\approx \log_2 T + 1$. We analyzed and compared the quantum cost of the traditional oracle design with our design which gives a low quantum cost.

Keywords: satisfiability; maximum satisfiability; quantum counter; Grover search algorithm; quantum circuit

Citation: Alasow, A.; Jin, P.; Perkowski, M. Quantum Algorithm for Variant Maximum Satisfiability. *Entropy* **2022**, *24*, 1615. <https://doi.org/10.3390/e24111615>

Academic Editors: Brian R. La Cour and Giuliano Benenti

Received: 18 October 2022

Accepted: 2 November 2022

Published: 5 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Satisfiability

The satisfiability (SAT) problem for a given Boolean function is the problem of determining if there exists a set of assignment values of input variables for the given Boolean function that evaluates this function to TRUE. Boolean or propositional-logic expressions are formed using operators AND, OR, EXOR, and NOT from input variables. Satisfiability expression (circuit) is often expressed as a product-of-sum (POS) form. POS is a logical ANDs of OR terms, where each OR term is an inclusive sum of literals. For instance, the POS SAT function $f(a, b, c) = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c)$ is satisfiable because when $c = 1$ and either a or b is equal to 1, then $f(a, b, c)$ evaluates to 1. Another example, $f(a, b) = (a + b)(\bar{a} + \bar{b})(\bar{a} + b)(a + \bar{b})$ is not satisfiable because no binary assignment of values for variables a and b , $f(a, b)$ would evaluate to 1.

Satisfiability problems have a wide range of applications, such as model checking in electronic design automation (EDA) [1], automatic test pattern generation (ATPG) [2], software and hardware verification [3], and circuit design [4]. Satisfiability problems also have many applications in Artificial Intelligence [5], robotics, and electronic design. Based on Cook's theorem [6], satisfiability is an NP-complete problem. Solving a satisfiability problem involving many variables and terms using traditional algorithms is computationally expensive.

1.2. Maximum Satisfiability

Maximum satisfiability (MAX-SAT) is an optimization version of the SAT problem. MAX-SAT finds the maximum number of constraints of a given Boolean function that are satisfied.

Suppose a Boolean function in the POS form contains thousands of sum (OR) terms (also called clauses). The MAX-SAT problem is to examine the maximum number of terms that are satisfied. For example, $f(a, b, c, \dots, N) = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c) \dots (\dots, \dots) = 1$. The function f is true for a binary assignment of values to variables a, b, c, \dots, N for which all terms are true. This is the SAT satisfiability. In contrast, the goal of MAX-SAT is not only to find the decision satisfied/unsatisfied (yes/no) but also to provide the maximum number of terms (clauses) that are satisfied with the actual satisfying assignment values for the variables in case the formula is not SAT satisfiable. The MAX-SAT is considered to be an NP-hard problem [7].

There are several extensions and modifications to the MAX-SAT problem formulated as above. For instance, sometimes not all constraints of a problem can be satisfied, but some of them must be satisfied. In such a case, MAX-SAT constraints can be divided into two sets of clauses:

- Hard clauses: The constraints that must be satisfied.
- Soft clauses: The constraints that may or may not be satisfied, but we want to satisfy as many as possible.

There are three main variants of MAX-SATs [8,9]:

1. Weighted MAX-SAT: Each clause has an associated weight cost, and the objective is to maximize the sum of the weights of the satisfied clauses.
2. Partial MAX-SAT: Finds the assignment values for the variables that must be satisfied for all hard clauses and must be maximized on the soft clauses.
3. Weighted partial MAX-SAT is a combination of the partial and weighted MAX-SAT.

The applications of these different variants will be discussed in the next section.

2. Related Work

2.1. Maximum Satisfiability Applications

There are many optimization problems and real-world applications that can be encoded to MAX-SAT. Some of the successful applications used for MAX-SAT are data analysis and machine learning, planning and scheduling, verification and security, bioinformatics, and combinatorial optimization [8]. We will briefly discuss some of these applications.

2.1.1. Data Analysis and Machine Learning

MAX-SAT has been used in many problems in Data Analysis, Artificial Intelligence (AI) and Machine Learning [10]. Correlation clustering is a well-studied problem in data analysis and AI in which data are divided into subgroups in a meaningful way. Discovering an optimal way of making such a division is a computational challenge. There are many approaches to find the optimal clustering, including a greedy local-search and approximation algorithms, which cannot find optimal clusterings. Solving exact formulations of the correlation clustering as MAX-SAT based approach leads to cost-optimal correlation clustering [11]. Bayesian Network Structure Learning (BNSL) is a computationally hard problem of finding a directed acyclic graph structure that optimally describes a given data structure. These problems use learning that can be based on probabilistic or exact inference methods. Using MAX-SAT as exact inference has been shown to yield a competitive approach to learning optimal bounded tree-width Bayesian network structures (BTW-BNSL) [12]. There are many other AI applications and data analysis approaches formulated as MAX-SAT, including causal structure discovery [13], and deriving interpretable classification rules [14].

2.1.2. Planning and Scheduling

MAX-SAT can be applied in linear temporal logic (LTL) specifications for robotic motion planning and control of autonomous systems. Suppose that we want to design a controller for a robotic museum guide; the robot has to give a tour of the exhibitions in a specific order, which constitutes the hard specification. Preferably, it also avoids certain locations, such as the staff's office, the library, or the passage when it is occupied. These

preferences are encoded in the soft specifications [15]. This is an example of a partial MAX-SAT formulation. There are other planning problems that can be encoded as MAX-SAT for cost-optimal planning [16,17].

Scheduling problems are well-known problems that appear in various contexts, including health care, airlines, transportation services, and various financial and money transfer problems in organizations. These scheduling problems can be encoded as a weighted partial MAX-SAT problem [18].

2.1.3. Verification and Security

Functional verification tasks dominate the effort of contemporary VLSI and SoC design cycles. A major step of functional verification is design debugging, which determines the root cause of failed verification tasks such as simulation or equivalence checking. The MAX-SAT formulation is used as a pre-processing step to construct a highly optimized debugging framework [19–21]. One of the techniques for debugging both hardware and software is fault localization, where the goal is to pinpoint the localization of bugs. Fault localization is performed using the MAX-SAT approach to reduce and improve automation for error localization, which can speed up the debugging process [22,23].

MAX-SAT has many applications in security. Starting with solving the user authorization query problem [24], reconstructing AES key schedule images [25], detecting hardware Trojans [26], and malware detection [27].

2.1.4. Bioinformatics

MAX-SAT has many applications in the bioinformatics field, such as cancer therapy, finding the optimal set of drugs to fix or rectify the fault areas of the gene regulatory network [28], modeling biological networks and checking their consistency [29], finding the maximum similarity between RNA sequences [30] and finding the minimum-cardinality set of haplotypes that explains a given set of genotypes [31].

2.1.5. Combinatorial Optimization Problems

Combinatorial optimization problems are widely studied in fundamental academic research and in solving real-life problems. Many of these problems are NP-hard, where an exhaustive search is not tractable. For instance, MAX-SAT has been used to encode and solve such problems as the Max-Clique problem [32–34], given a group of vertices. The maximal clique is the largest subset of vertices in which each point is directly connected to every other vertex in the subset.

Other applications within this domain that have been encoded into MAX-SAT consist of determining the Treewidth of a graph [35] and finding solutions for the maximum quartet consistency problem [36].

2.2. Classical Algorithm for Maximum Satisfiability Problem

There are many classical algorithms for solving MAX-SAT problems: exact algorithms, stochastic local search algorithms [37–39], evolutionary algorithms [40,41], and hybrids of local search and evolutionary algorithms [42,43]. Exact algorithms are often used for small or medium size problems that can be easily verified as satisfied or unsatisfied. The exact algorithms are based on the Davis–Putnam–Logemann–Loveland algorithm (DPLL) [44], an example being the Branch-and-Bound algorithm [45,46] which represents the search space of all possible value assignments to variables as a search tree. Branch-and-Bound explores the branch of the tree and creates new formulas with partial assignments in the internal nodes until the solution is found. The solution is stored in the leaf nodes, which are bound to prevent unnecessary branches. Large size problems use stochastic local search algorithms and evolutionary algorithms which can potentially provide a high-quality solution [42,47].

2.3. Quantum Algorithms for Maximum Satisfiability Problem

MAX-SAT is an NP-hard problem and is one of the most widely studied optimization problems in classical algorithms. These NP-hard problems can be potentially solved by quantum algorithms which would offer significant improvements over the classical algorithms, assuming the existence of quantum computers with sufficiently many qubits.

There is some active research to solve the SAT and MAX-SAT problems using the currently available quantum computers, especially the D-wave quantum annealer (QA) systems [48]. The SAT and MAX-SAT are encoded into Quadratic Unconstrained Binary Optimization (QUBO) compatible with the quantum annealer architecture. QUBO is a mathematical class of problems expressed in binary variables as linear or pairwise quadratic terms, which may include constraints.

Practical MAX-SAT problems contain hundreds of variables and terms/clauses which cannot be handled by the currently available quantum computers. Thus, due to the limited number of qubits available, some algorithms suggested reducing the number of qubits. For instance, the quantum cooperative search algorithm for 3-SAT [49] proposed Grover’s search algorithm combined with a classical algorithm that decreases the total number of variables by replacing some qubits with classical bits. However, still, the number of needed ancilla qubits is equal to the number of terms when applied to POS 3-SAT problems.

We propose a new quantum circuit using Grover’s search algorithm, which can be applied to both SAT and MAX-SAT problems with a reduced quantum cost. The main idea is to avoid large Toffoli gates that have high quantum costs and lead to decoherence. Our novel quantum oracle circuit design requires fewer logical qubits to implement the maximum satisfiability problem. This is based on replacing large AND gate collecting results from clauses by a quantum counter that counts the number of satisfied clauses inside the SAT oracle upgraded MAX-SAT oracle. Because modern quantum computers and simulators have a limited total number of qubits, our quantum algorithm allows us to solve larger MAX-SAT problems. However, because of a limited number of qubits, it is not competing with modern software MAX-SAT solvers.

3. Definitions and Preliminaries

In this section, we will define some basic concepts related to quantum gates and quantum cost. A few useful gates are shown in Figure 1.

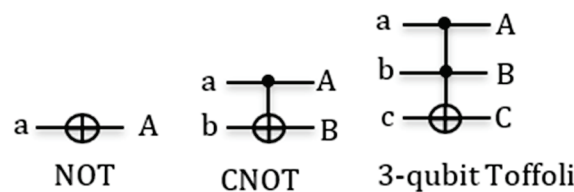


Figure 1. Gate symbol: NOT, CNOT, 3-qubit Toffoli gates.

Definition 1: Reversible gate is $n \times n$ quantum gate that has n input variables and n output variables. A quantum gate is reversible if it maps an n -input binary vector into a unique n -output binary vector. In addition, it is a one-to-one mapping or a permutation of vectors. For example, the NOT gate is reversible because if the output is 0, then you know the input must be 1, and vice versa.

Definition 2: Controlled-NOT (CNOT) is a 2-qubit gate, where the first qubit is called control and the second qubit is called target. CNOT applies the NOT gate on the target qubit when the control qubit is one. The value of the control qubit is not affected. Thus $A = a$, $B = a \oplus b$. The CNOT gate is also called the Feynman gate. Using Definition 1, the reader can check that this function is reversible.

Definition 3: *n*-control Toffoli gate consists of *n*-control qubits and one target qubit. The target qubit is inverted if all control qubits are 1. Otherwise, the target qubit is unchanged: $C = ab \oplus c$. The values of all control qubits are not changed, thus $A = a$, $B = b$, etc. This is the universal reversible gate; it realizes AND with $c = 0$ and NAND with $c = 1$.

Definition 4: Ancilla qubits are extra qubits to allow extra working space during the computation. They are necessary to convert arbitrary Boolean functions to reversible Boolean functions.

For instance, the Boolean function $X = a \cdot b$ is not reversible, but function $X = a \cdot b \oplus c$ is a reversible gate with $c = 0$.

Although the iterative quantum counter can be built from NOT, CNOT, and multi-qubit Toffoli gates, our design uses Peres gates because the design with Peres gates leads in many cases to substantial circuit cost reduction. Peres gates are built from truly quantum gates CV and CV+ and other Controlled-Nth Root of NOT gates, which requires explaining these gates first.

3.1. Nth Root of Not Gate

Mathematically, a quantum gate with *n* qubit input can be represented as a $2^n \times 2^n$ unitary matrix. *N*-th root of NOT gate can be constructed from matrix representation as follows:

$$\sqrt[n]{NOT} = \frac{1}{2} \begin{vmatrix} 1 + e^{\frac{i\pi}{n}} & 1 - e^{\frac{i\pi}{n}} \\ 1 - e^{\frac{i\pi}{n}} & 1 + e^{\frac{i\pi}{n}} \end{vmatrix}.$$

Below given are notations and properties that will be used in the paper to design larger Peres gates:

V gate = \sqrt{NOT} gate

V^\dagger gate is inverse of V gate. Where V^\dagger is called V dagger or conjugate of V .

$W = \sqrt{V} = \sqrt[4]{NOT}$

$G = \sqrt{W} = \sqrt[8]{NOT}$

$VV = NOT$

$VV^\dagger = I$

$WW = V$

$GG = W$

3.2. Controlled-Nth Root of NOT Gate

The controlled-*N*th root of NOT gate is a 2-qubit gate, where the first qubit is the control, and the second qubit is the target. When the control is one ($|1\rangle$) then the target qubit calculates the *N*-th root of NOT gate applied to its input value. Otherwise, with control $|0\rangle$ the target qubit is not changed. The matrix representation of controlled-*N*th root of NOT gate is:

$$\text{Controlled-}\sqrt[n]{NOT} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1+e^{\frac{i\pi}{n}}}{2} & \frac{1-e^{\frac{i\pi}{n}}}{2} \\ 0 & 0 & \frac{1-e^{\frac{i\pi}{n}}}{2} & \frac{1+e^{\frac{i\pi}{n}}}{2} \end{vmatrix}$$

The inverse of *N*-th root of NOT gate and controlled-*N*th root of NOT gate are constructed from a matrix where the plus and minus signs are reversed.

Figure 2 shows examples of various controlled-*N*th root of NOT gates that we will use in our design of large Peres gates used in counters.

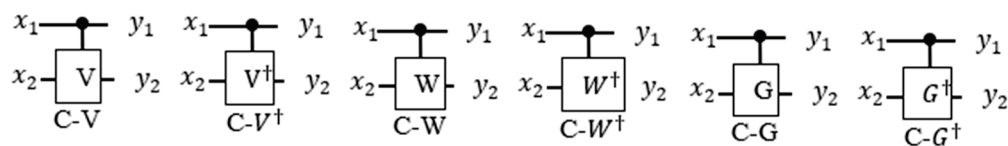


Figure 2. Some symbols for quantum gates of Controlled- n th root of NOT gate and their inverse (\dagger) dagger or conjugate.

3.3. Quantum Cost

Quantum cost of a quantum circuit is the number of elementary quantum gates used to build the circuit. The elementary quantum gates are primitive gates which are 1×1 and 2×2 reversible gates. The cost of the primitive gates is equal to 1; therefore, the quantum cost is just the number of primitive gates. For illustration, these are three elementary quantum gates that are used to calculate the quantum cost: NOT, controlled- n th root of NOT, and CNOT gates where cost of each gate is equal to 1. (There are some more accurate characterizations of costs of primitive quantum gates [50] but for this paper we use the approximate costs defined as above.)

Toffoli gate could be built using controlled- n th root of NOT gate [51]. A 3-bit Toffoli gate from Figure 3 has two control qubits and one target qubit and is built from controlled V/V^\dagger gates and CNOT gates. The quantum cost of the 3-bit Toffoli gate is 5. The generalized formula for quantum cost of m -control Toffoli gate [52] is equal to $2^{m+1} - 3$.

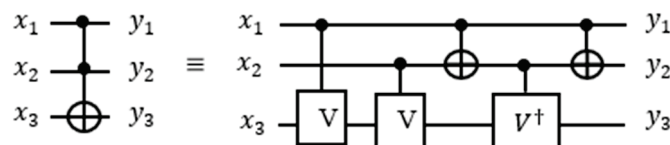


Figure 3. 3-bit Toffoli gate represented as controlled- V/V^\dagger and CNOT gates.

3.4. Peres Gate

The Peres gate [53] can be characterized as a sequence of n -Toffoli followed by Feynman (CNOT) gates. For instance, a 3-bit Peres gate consists of a 3-bit Toffoli and a CNOT gates (Figure 4I). When the 3-bit Toffoli and CNOT gates are implemented separately, the cost would be six (Figure 4II). However, the 3-bit Peres gate costs four because the adjacent CNOT gates cancel each other. Thus, the Peres gates are used for quantum cost reduction of quantum circuits and for blocks of the iterative counter in this paper specifically.

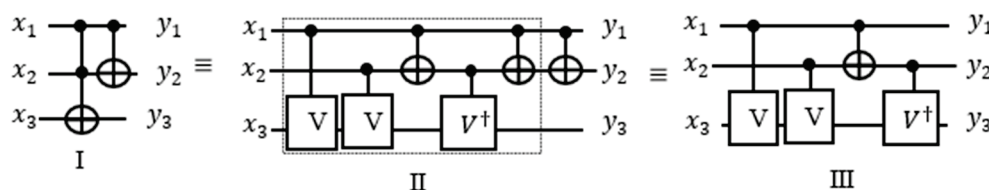


Figure 4. (I) 3-bit Peres gate (II) decomposed Toffoli gate with CNOT. (III) 3-bit Peres gate and its representation using controlled- V/V^\dagger and CNOT gates.

The Figure 4III:

- If x_1 is 1 and x_2 is equal to 0 or vice versa, then the transformation applied to x_3 and one of the V -gate will become active and the other one will be inactive which behaves as the identity. Also, CNOT will become active which produces 1 that will activate V^\dagger -gate, thus $VV^\dagger = I$.
- If both x_1 and x_2 are equal to 1, then the transformation applied to x_3 and two of the V -gate will become active. Also, CNOT will become inactive which produces 0 that will inactivate V^\dagger -gate, thus $VV = NOT$.
- If both x_1 and x_2 are equal to 0, then no transformation is applied on the gates.

4. Proposed Quantum Algorithm for Maximum Satisfiability

In traditional Grover's algorithm, oracles are composed of Toffoli and NOT gates; one needs to keep the results of all OR terms for the final AND gate being the decision output of the oracle. The answer to each OR term is stored in a separate ancilla qubit; thus, we need the number of ancilla qubits equal to the number of terms in the function. In Boolean functions involving thousands of terms, this would mean Grover's oracle needs thousands of ancilla qubits. If there are T terms in a function, we would need T ancilla qubits. For large T , the number of required ancilla qubits becomes unrealistically large, even for future large quantum computers with thousands of logical qubits. Therefore, we present here a novel quantum oracle circuit design that requires $\lceil \log_2 T \rceil + 1$ ancilla qubits when T is not a power of 2 or $\lceil \log_2 T \rceil + 2$ ancilla qubits when T is a power of 2 in order to keep the circuit from growing too large. Our design also improves the overall runtime. For example, in traditional oracles if there are 1,000,000 terms, then we need the same number as 1,000,000 ancilla qubits, but for our design, we need only 21 ancilla qubits. To eliminate the need for ancilla qubits, we make use of the concept of an iterative quantum counter built from blocks, with each block built from controlled Peres gates. We connect one block of the iterative quantum counter after each Toffoli gate representing the OR term of the function POS formula. The satisfiability value of this term controls the block of the counter by activating this block or not. It then increments the count by 1 or 0, depending on the truth value of the OR term. Thus, our quantum counter counts the number of satisfied OR terms in the Boolean function implemented as a POS.

We assign a counter block for each OR term, where the result of the term is used as one of the control qubits of the counter. When the term evaluates to 0, nothing is registered in the counter. When it evaluates to 1, the counter outputs the binary number $value + 1$ to the previously accumulated count value. The use of a quantum counter allows us to send the result from the Toffoli gate representing one OR term to the counter circuit, hence eliminating the need for an ancilla qubit. We can set the function qubit back to 1 by mirroring the Toffoli gate used to compute the result and set the input qubits back to the original by applying NOT gates when appropriate. Our design drastically reduces the number of qubits needed for a function at the cost of replicating Toffoli gates in the POS expression and the costs of the iterative counter.

4.1. Grover's Search Algorithm

Grover's Algorithm [55] searches an unordered array of N elements to find a particular element with a given property. Grover's algorithm is often used as a subroutine in other quantum algorithms [56–58]. In classical computations, in the worst case, this search takes N queries (tests, evaluations of the classical oracle). In the average case, the particular element will be found in $N/2$ queries. Grover's algorithm can find the element in \sqrt{N} queries. Thus, Grover's algorithm can be used to solve the decision maximum satisfiability k -SAT for every value of k . Grover's algorithm is a quantum search algorithm, which speeds up a classical search algorithm of complexity $O(N)$ to $O(\sqrt{N})$ in the space of N objects, hence Grover gives a quadratic speed up. To solve the optimization problem of finding MAX-SAT with maximum value of k Grover's Algorithm has to be repeated.

The MAX-SAT contains n variables from the given Boolean function which is used to represent the search space of $N = 2^n$ elements. To apply the MAX-SAT in Grover's algorithm, these N elements are applied in a superposition state which is the input to the oracle. If the oracle recognizes an element as the solution, then the phase of the desired state is inverted. This is called the Phase inversion of the marked element. The marked element is a true minterm of function f from the oracle. The true minterm is a product of all variables of function f that evaluates to $f = 1$. Grover's search algorithm uses another trick called inversion about the mean (average), which is also known as diffusion operation or amplitude amplification. Inversion about the mean amplifies the amplitude of the marked states and shrinks the amplitudes of other items. The amplitude amplification increases

the probability of marked states, so that measuring the final states will return the target solution with a high probability near 1.

As shown in Figure 7a, the n qubits in the superposition state result from applying a vector of Hadamard gates to initial state $|0\rangle^n$. Next applied is repeated operator G which is called the Grover Loop. After the iteration of the Grover Loop operator $O(\sqrt{N})$ times the output is measured for all input qubits. Oracle can use an arbitrary number of ancilla qubits, but all these qubits must be returned to value $|0\rangle$ inside the oracle. The number of required iterations for Grover operator is: $R \leq \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil$ where N is number of all search space elements and M is number of solutions. The Grover Loop G is a quantum subroutine which can be broken into four steps as shown in Figure 7b:

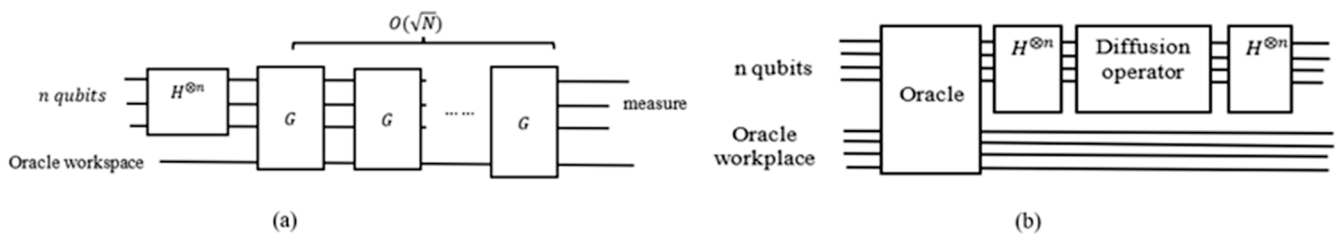


Figure 7. (a) Schematic circuit for Grover’s algorithm [55]. (b) Grover operator G .

1. Phase inversion: apply the oracle. If the oracle recognizes the solution, then the phase of the desired state is inverted
2. Apply the Hadamard transform $H^{\oplus n}$ ($H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$)
3. Zero state phase shift: Perform the condition phase shift, in which all states receive a phase shift of -1 except for the zero state $|0\rangle$.
4. Apply the Hadamard transform $H^{\oplus n}$

4.2. Quantum Counter

As described in Section 3.3, the quantum counter block should be constructed from multiple-controlled Peres gates, where the first qubit of the Peres gate is applied a constant 1 with other variables combined, and the Peres gate is then turned into a quantum counter. (This qubit will be next taken from the OR term of the satisfiability formula to activate the counter block realized from Peres gates). For simplicity of explanation, we assume that the counter block is built from Toffoli and CNOT gates, as shown in Figure 8.

Here z is the least significant qubit and x the most significant. The outputs of CNOT and two of the Toffoli gates are $1 \oplus z$, $1 \oplus z \oplus y$, and $1 \cdot z \cdot y \oplus x$, respectively. When $xyz = 000$, the first Toffoli gate outputs $1 \cdot z \cdot y \oplus x = 1 \cdot 0 \cdot 0 \oplus 0 = 0 \oplus 0 = 0$ and the second $1 \cdot z \oplus y = 1 \cdot 0 \oplus 0 = 0 \oplus 0 = 0$. The outputs of the qubits y and x are both zeros. The output of the qubit z is $1 \oplus z = 1 \oplus 0 = 1$. Hence the circuit incremented 000 by 1 to 001 . Quantum counter circuit indeed outputs the value $\text{input}+1$.

If we connect the first control input of the quantum counter block to a circuit, then the output of the connected circuit (a term of the POS) will either activate or deactivate the counter. When the output of the connected circuit is equal to 1 , the output of the counter block is incremented by 1 . When the output of the circuit is equal to 0 , the output of the counter block is unchanged.

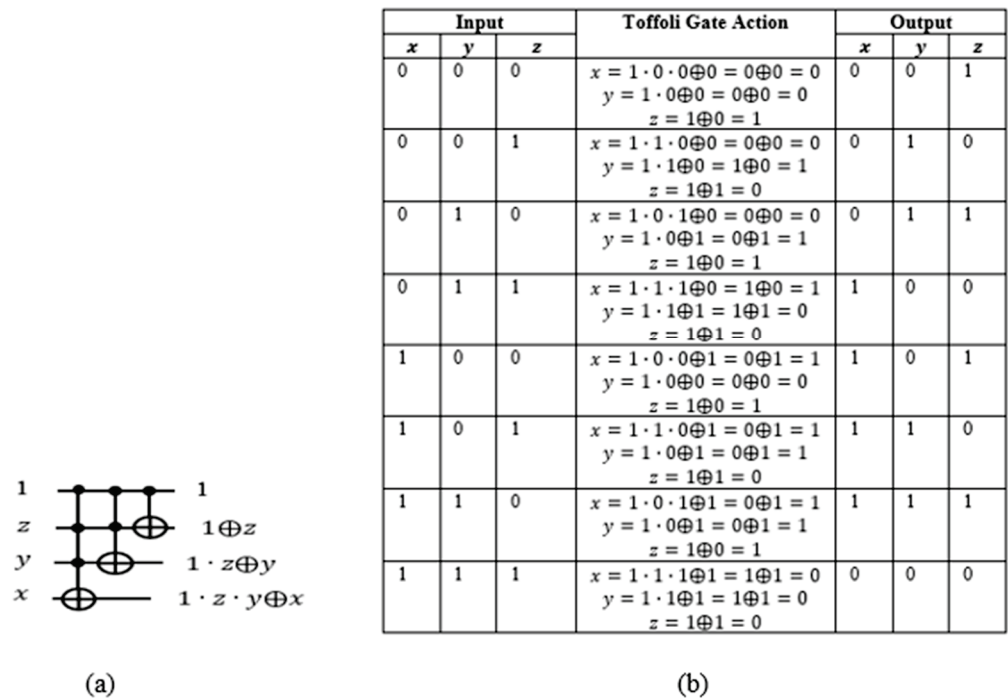


Figure 8. (a) Three-qubit quantum counter. (b) Analysis of 3-qbit quantum counter block from (a).

4.3. Traditional Oracle for Satisfiability Boolean Function

To build an OR term using a Toffoli gate, we use De Morgan’s Law to convert the term into a product of the same variables $a + b + c = \overline{\overline{a + b + c}} = \overline{\overline{a} \cdot \overline{b} \cdot \overline{c}}$. With the XOR operation, $1 \oplus a = \overline{a}$. Hence $a + b + c = \overline{\overline{a} \cdot \overline{b} \cdot \overline{c}} = 1 \oplus \overline{\overline{a} \cdot \overline{b} \cdot \overline{c}}$. The corresponding quantum circuit using a Toffoli gate is shown in Figure 9.

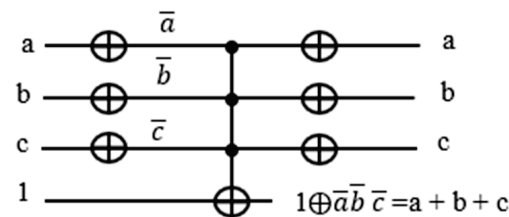


Figure 9. Convert sum term to product term using De Morgan’s law.

Suppose we have a Boolean function $f(a, b, c) = (a + b + \overline{c})(\overline{a} + \overline{b} + c)(b + c)$ from Karnaugh map in Table 1. As one can see in Table 1, there are four which means the solution of the Boolean variables in binaries are $(abc = 010, 011, 111, 101)$, which are satisfied for the Boolean function. Every true minterm in the Karnaugh map from Table 1 is a marked element and potential solution to the Grover Algorithm. However, in one run of Grover’s search algorithm, only one solution is found.

Table 1. Karnaugh map of POS for the Boolean function $f(a, b, c) = (a + b + \overline{c})(\overline{a} + \overline{b} + c)(b + c)$.

ab \ c	0	1
00	0	0
01	1	1
11	0	1
10	0	1

We build a quantum oracle for the Grover’s Loop using Toffoli gates, in which the XOR gate is controlled by the product of variables. We need to first convert the Sum expressions into Products using De Morgan’s Law.

$$a + b + \bar{c} = \overline{\overline{a + b + \bar{c}}} = \overline{\bar{a}\bar{b}c} = \overline{\bar{a}\bar{b}}\bar{c}$$

$$\bar{a} + \bar{b} + c = \overline{\overline{\bar{a} + \bar{b} + c}} = \overline{\overline{\bar{a}\bar{b}}c} = \overline{\bar{a}\bar{b}}\bar{c}$$

$$b + c = \overline{\overline{b + c}} = \overline{\bar{b}\bar{c}}.$$

After building each term with the corresponding product expression, each with an assigned ancilla qubit for the output, we need to put the terms together as the product of the OR terms for the entire function $f(a, b, c) = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c)$. Since $xyz \oplus 0 = xyz$, we use another Toffoli gate controlled by the product of the OR terms XORed with 0. The schematic of the entire circuit for $f(a, b, c) = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c)$ is shown in Figure 10:

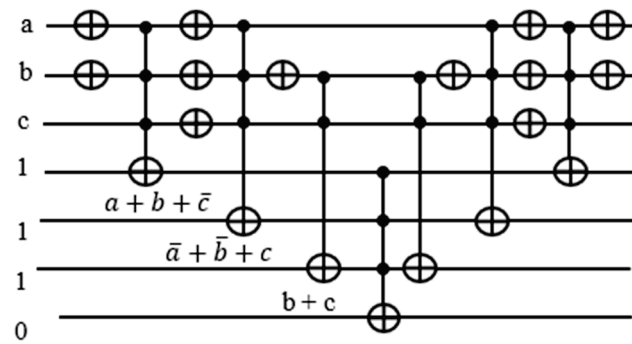


Figure 10. Traditional oracle for Multiple input Toffoli gate used as global AND gate $f = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c)$.

To set the input qubits and ancilla qubits back to their original states, we mirror all the circuits up to the $f(a, b, c)$ on the right-hand side of the function gate.

Let us define n number for variables and t number for terms then the number of qubits q needed for the oracle is: $q = n + t + 1$. Where 1 is for the OR terms XORed with 0. Notice that we need three ancilla qubits, which is equal to the number of terms. For a function involving thousands of terms, we would need an equal number of ancilla qubits.

4.4. Proposed Construction of a Quantum Oracle for MAX-SAT

Our proposed circuit does not require keeping the OR terms for the later calculation of the function. All we need to know is whether each term is satisfied or not, and we pass the result to the counter block assigned to it. Thereafter, we put the ancilla qubit back to the original state 1 by mirroring. Depending on neighboring expressions, there are opportunities to cancel double NOT gates, yet saving the number of gates needed.

The target output of each Toffoli gate realizing an OR term is used to activate the counter block corresponding to it. In Figure 11, notice that there are two NOT gates adjacent to each other, canceling each other out. Hence, we can remove those gates from our circuit.

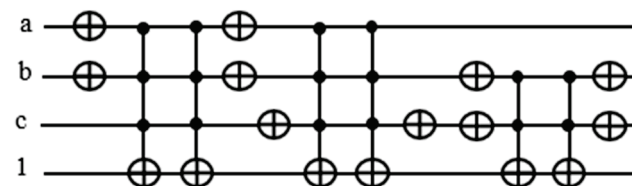


Figure 11. Improved version of the part the oracle $f = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c)$.

There are eight NOT and six Toffoli gates in this design in Figure 12 as opposed to 12 NOT and 7 Toffoli gates in the traditional design in Figure 10. The reason we need ancilla qubits in the traditional design is that we need the outputs from the Toffoli gates recorded in the ancilla qubits for counting the number of satisfied terms. By sending the satisfaction result for each term to the quantum counter, we are able to reset the output line back to 1.

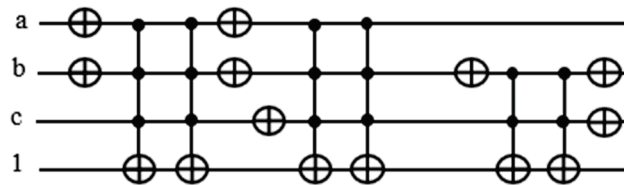


Figure 12. Improved and optimized version of the part the oracle $f = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c)$.

The count for the number of satisfied terms is output on the xy qubits. In this case, we have three satisfied terms and want to have three as the output expressed as 11 which are expressed as $xy \oplus out_0 = xy \oplus 0$ on a Toffoli gate. If the Boolean function f is satisfied, then the outcome out_0 should be 1. The entire oracle with the function and the iterative counter is shown in Figure 13. We applied this oracle in the Grover search algorithms for $R = 2$ iterations from this formula: $R \leq \left\lceil \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rceil$ where $M = 4$ is the number of solutions in our problem from Table 1, and $N = 8$ is the number of all search space elements (cells of the Karnaugh map from Table 1). In general, the value of M is calculated using Quantum Counting algorithm [55], but an unsolved problem, the value of M , is taken as 1 to run the Grover iterations R .

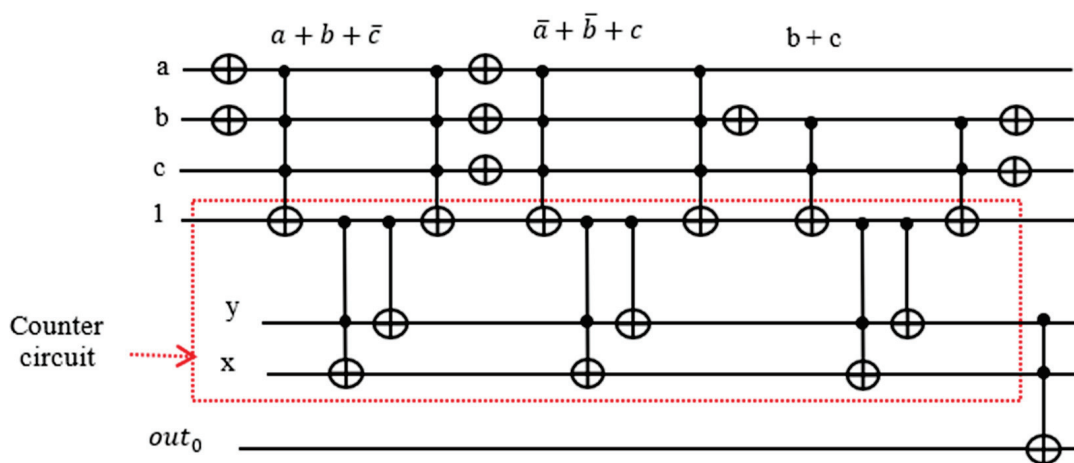


Figure 13. Improved complete oracle using quantum counter.

In Figure 14, we run the circuit on the ‘qasm_simulator’ from QISKIT for 1024 shots (independent runs to obtain high precision probability) for which the circuit produces the correct answers. We measured a_0, a_1, a_2 and out_0 in Figure 14 where a_0, a_1, a_2 correspond to the Boolean variables, a, b, c , respectively in Figure 13. As can be seen in Figure 15, it illustrates the QISKIT [59] output graphics for the simulated circuit. The measured values with high probability are 1010, 1101, 1110, and 1111, where the most significant qubit is out_0 which is 1, and the least three significant qubits 010, 101, 110, 111 are all satisfied values for the Boolean function. These solutions correspond to the true minterms from Table 1. For the unsatisfied, the measured values with low probability are 0000, 0001, 0011, and 0100, where the most significant qubit is out_0 which is 0, and the least three significant qubits 000, 001, 011, 100 are all unsatisfied values for the Boolean function.

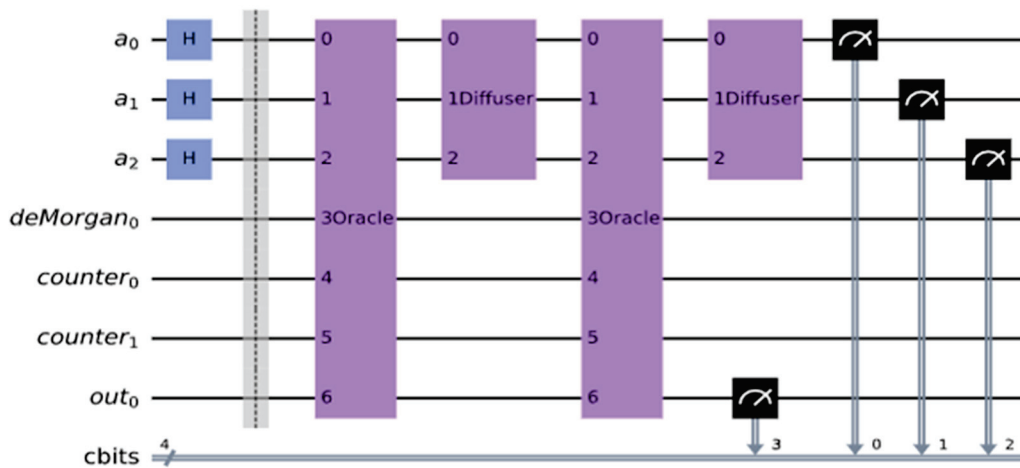


Figure 14. MAX-SAT applied Grover’s search algorithm. $f(a, b, c) = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c)$.

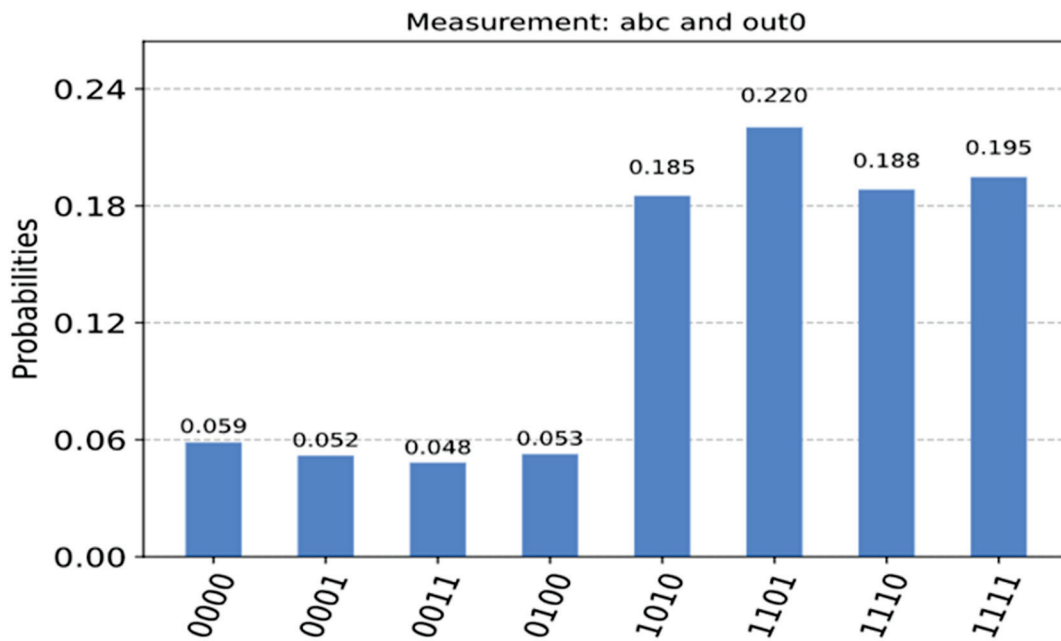


Figure 15. Measurement of the Boolean variables and the outcome of function $f(a, b, c) = (a + b + \bar{c})(\bar{a} + \bar{b} + c)(b + c)$.

As can be seen in Figure 15, the four values 000, 001, 011, and 100 have some value with less probability because of noise created by the simulator. However, we verified the solutions by applying the number of iterations R , and the output from the simulation with high probability 010, 101, 110, and 111 matches the theoretical values, which can be verified manually. We also applied different shots to test, and the results were closely similar, with a high probability for all satisfying values.

4.5. Verifying an Unsatisfiable Function

Suppose a function with four OR terms $f(a, b) = (a + b)(\bar{a} + \bar{b})(a + \bar{b})(\bar{a} + \bar{b})$ which no assignment of values a and b evaluates the function to 1. We need to first convert the OR terms into Products using De Morgan’s Law and then build the oracle for the given Boolean function.

$$a + b = \overline{\overline{a + b}} = \overline{\bar{a} \cdot \bar{b}}$$

$$\bar{a} + \bar{b} = \overline{\overline{\bar{a} + \bar{b}}} = \overline{\overline{\bar{a}} \cdot \overline{\bar{b}}} = \overline{a \cdot b}$$

$$a + \bar{b} = \overline{\overline{a + \bar{b}}} = \overline{\bar{a} \cdot b} = \bar{a} \cdot \bar{b}$$

$$\bar{a} + \bar{b} = \overline{\overline{\bar{a} + \bar{b}}} = \overline{\bar{a} \cdot b} = \bar{a} \cdot \bar{b}$$

The four qubits (1, z, y, x) in block (A) realize the counter, which can count from 0 to 7. We need the last qubit with out_0 ancilla bit to produce 1 when all terms are satisfied for Grover’s algorithm. Since this function has four terms, to check satisfiability which is the last qubit should be 1, we need to add two NOT gates in the block (B) which makes the last qubit to produce 1 if the Boolean function is satisfied. The function $f(a, b)$ from Figure 16 is not satisfiable, so comparing to a value of 4 in the last gate would not generate any correct solution. Grover’s algorithm will give a few random values that can be verified on the satisfiability formula outside Grover’s Algorithm using function $f(a, b)$. Therefore, we remove the two NOT gates in block (B) to get the maximum satisfied terms of the function.

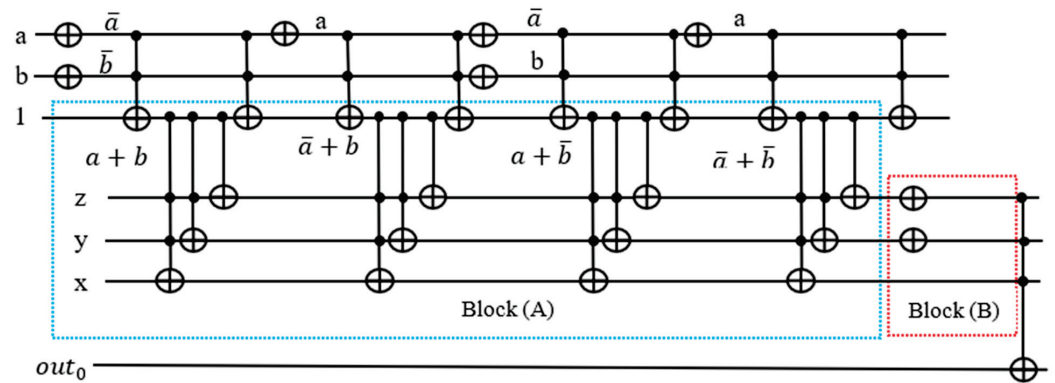


Figure 16. Oracle with counter $f(a, b) = (a + b)(\bar{a} + b)(a + \bar{b})(\bar{a} + \bar{b})$.

In a more general case in Figure 17, we repeat the Grover Algorithm with tuning values of thresholds until equal to counter value xyz . The comparator $G = H$ compares the output from the counter with the threshold value given as constant values n_1, n_1 , and n_3 . For instance, $f(a, b) = (a + b)(\bar{a} + b)(a + \bar{b})(\bar{a} + \bar{b})$ has 4 terms, we tune the threshold value from 4, 3, 2, and 1 until the condition is met. The value of the counter where the condition is met is the MAX-SAT value. If the condition is met, the ancilla qubit out_0 will be flipped. It changes the quantum phase of the solution so that the elements that satisfy all constraints are marked. This method of the threshold with comparator is useful to check when the exact number of terms (constraints) are known, which can be checked whether the threshold is equal to the counter value. For instance, if there are 10 constraints in a given function, but it should satisfy a minimum seven constraints, then set the threshold to seven and check if the counter equals to seven. There are applications based on the method of the threshold with a comparator, such as finding the minimum set of support [60].

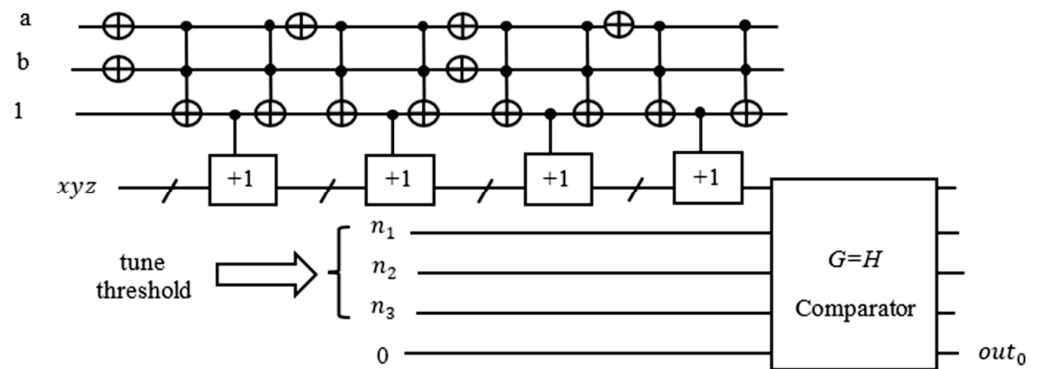


Figure 17. Oracle with counter circuit and threshold with comparator.

Every binary vector $|a, b\rangle$ of a solution can be verified by running outside of the Grover Algorithm, as can be seen in Figure 18 in which the maximum number of satisfied terms is 3 out of 4. We applied one Grover’s Loop iteration for this oracle to get the MAX-SAT. In Figure 19, we run the circuit on the ‘qasm_simulator’ from QISKIT for 1024 shots.

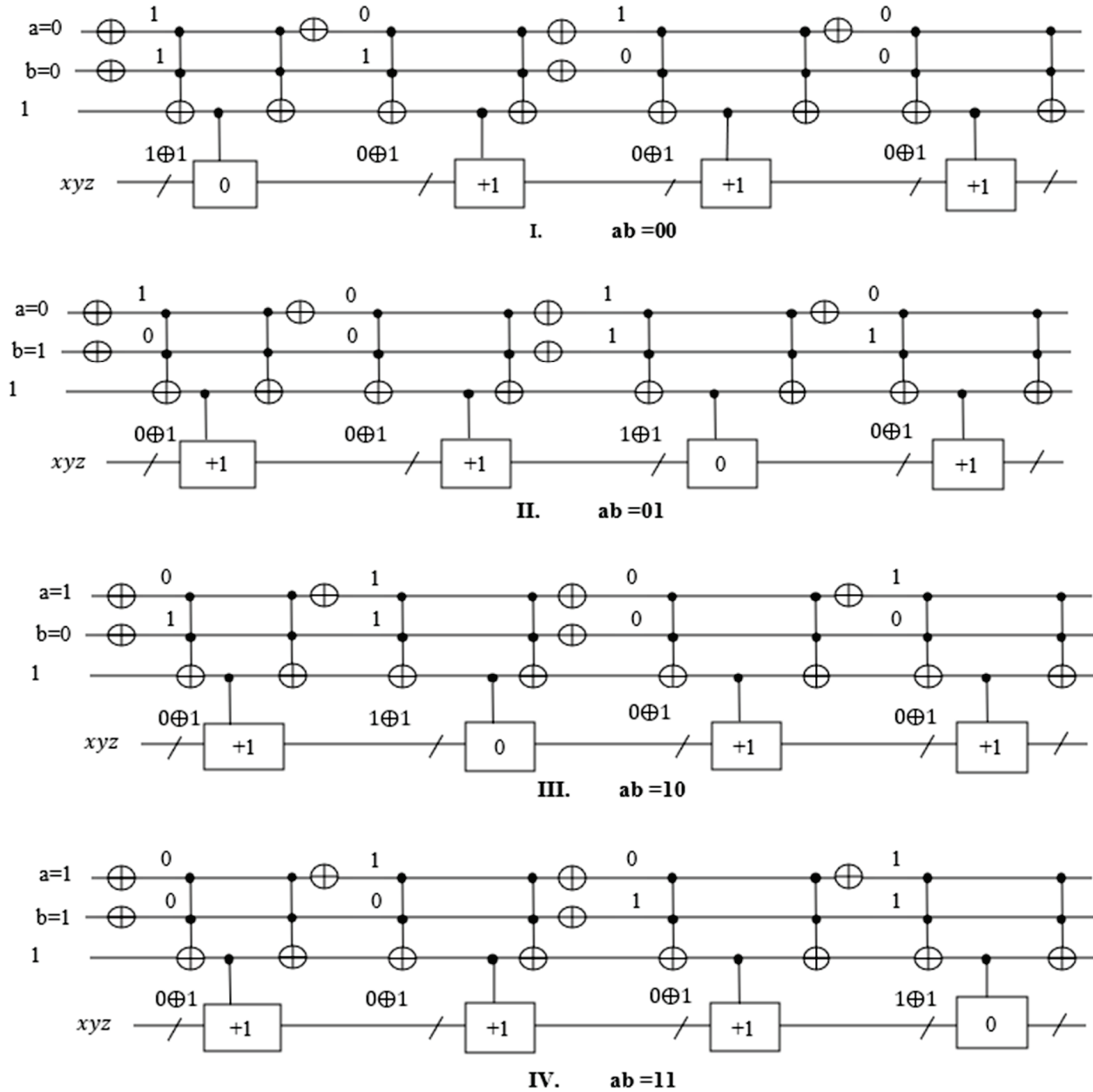


Figure 18. MAX-SAT verification.

In Figure 19, we measured the Boolean variables, counter, and output. In Figure 20, the most significant qubit out_0 always is 0, which means the Boolean function is not satisfied because there are no such binary values for the least two significant qubits 00, 01, 10, and 11, which would satisfy the Boolean function. However, the novelty of our design is that the counter qubits give the maximum numbers of satisfied terms in the Boolean function. The counter qubits are the second, third, and fourth qubits from the most significant qubit, which in this case is 011.

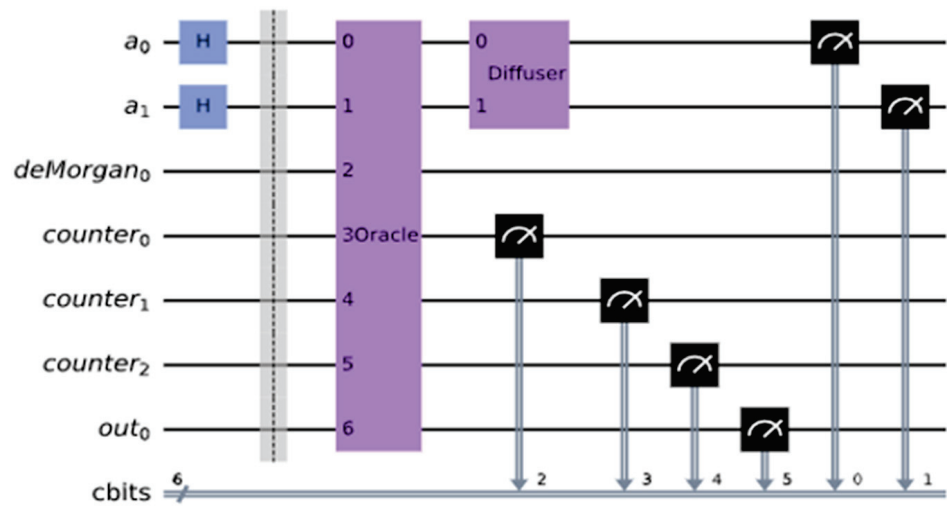


Figure 19. $f(a, b) = (a + b)(\bar{a} + b)(a + \bar{b})(\bar{a} + \bar{b})$ applied Grover's algorithm.

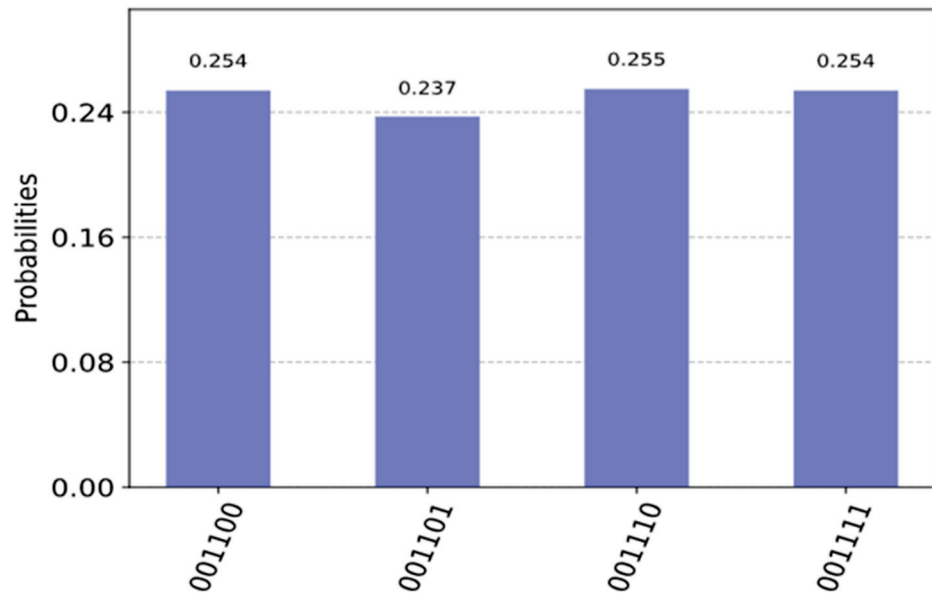


Figure 20. Measurement of $f = ((a, b, c) a + b)(\bar{a} + b)(a + \bar{b})(\bar{a} + \bar{b})$.

5. Calculation of Quantum Cost

5.1. Calculation of Quantum Counter Size

In the Table 2 shows the required number of qubits for the quantum counter which each term is not required for one ancilla qubit, but many terms require a few ancilla qubits.

In general, if there are T terms in given Boolean function then the total number of qubits that need for quantum counter is:

- $\lceil \log_2 T \rceil + 1$ ancilla qubits when T is not a power of 2
- $\log_2 T + 2$ ancilla qubits when T is power of 2

As shown in Figure 21, for instance, if there are 100,000 terms, then the number of required ancilla qubits in traditional oracle is 100,000, but in our design, the quantum counter requires only $\lceil \log_2 T \rceil + 1 = 18$ ancilla qubits. Using the quantum counter, each term is not required for one ancilla qubit, but many terms are required for a few ancilla qubits.

Table 2. Quantum counter size; total qubits for counter.

Number of Terms (Clauses)	Total Qubits for Quantum Counter
2	$\lceil \log_2 T \rceil + 2 = 3$
3	$\lceil \log_2 T \rceil + 1 = 3$
4	$\lceil \log_2 T \rceil + 2 = 4$
5 ... 7	$\lceil \log_2 T \rceil + 1 = 4$
8	$\lceil \log_2 T \rceil + 2 = 5$
9 ... 15	$\lceil \log_2 T \rceil + 1 = 5$
16	$\lceil \log_2 T \rceil + 2 = 6$
17 ... 31	$\lceil \log_2 T \rceil + 1 = 6$
32	$\lceil \log_2 T \rceil + 1 = 7$
33 ... 63	$\lceil \log_2 T \rceil + 1 = 7$
64	$\lceil \log_2 T \rceil + 2 = 8$
65 ... 127	$\lceil \log_2 T \rceil + 1 = 8$
128	$\lceil \log_2 T \rceil + 2 = 9$
129 ... 255	$\lceil \log_2 T \rceil + 1 = 9$
256	$\lceil \log_2 T \rceil + 2 = 10$
257 ... 511	$\lceil \log_2 T \rceil + 1 = 10$
...	...
...	...
T	$\begin{cases} \lceil \log_2 T \rceil + 1, & \text{if } T \text{ is not power of } 2 \\ \lceil \log_2 T \rceil + 2, & \text{if } T \text{ is power of } 2 \end{cases}$

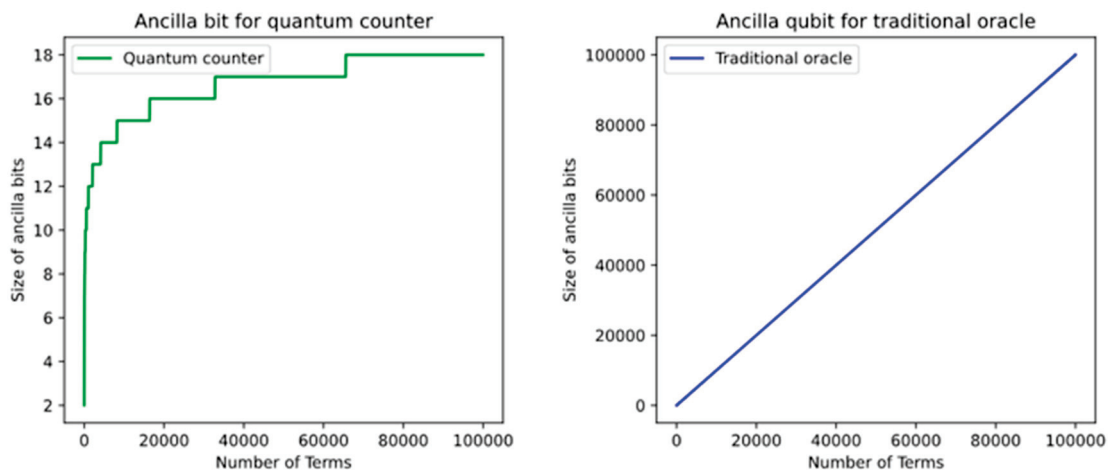


Figure 21. Comparison of required numbers of ancilla qubits for our oracle and the traditional oracle.

5.2. Quantum Cost Calculation for Quantum Counter

Each term in the Boolean function is represented as n -bit Toffoli gate, and the satisfiability result is passed down to the counter. We need as many counter blocks as there are terms in the given POS Boolean function. The counter can be built from Toffoli gates or Peres gates. It is important to have low cost quantum circuits for this high demand for n -bit Toffoli gate. Since the Peres gate is a low-cost quantum circuit, we replaced the Toffoli gates with Peres gates for cost reduction [52]. The formula of quantum cost for m -controlled bits of Peres gate is m^2 and for Toffoli gate is $2^{m+1} - 3$.

In Figure 22 a three-qubit counter (3-control qubits) consists of three Toffoli gates which are 3-control, 2-control, and 1-control (CNOT) gates. For each of these Toffoli gates, the quantum cost is calculated separately: $(2^{3+1} - 3) + (2^{2+1} - 3) + (2^{1+1} - 3) = 28 - 9 = 19$. Four-qubit counter consists of four Toffoli gates, and the quantum cost is also calculated separately: $(2^{4+1} - 3) + (2^{3+1} - 3) + (2^{2+1} - 3) + (2^{1+1} - 3) = 60 - 12 = 48$.

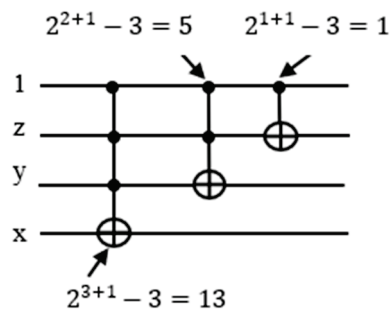


Figure 22. Quantum cost for 3-bit counter.

Thus, we can derive a general formula for the quantum cost of m -bit quantum counter using the Toffoli gate:

$$2^{m+2} - 4 - 3m.$$

The total quantum cost of the quantum counter for each term T is:

$$\text{Peres cost} = T * m^2. \tag{1}$$

$$\text{Toffoli cost} = T * (2^{m+2} - 4 - 3m). \tag{2}$$

Based on these two Formulas (1) and (2), the Toffoli gate has a higher quantum cost than Peres gate. Thus, we used in our design the Peres gates. As we mentioned before, our final counter uses Peres gates, so we built our oracle using the Peres gate, and it is mapping to the n th root of NOT gates which leads to low quantum cost. The recursive design method from Peres gate was used.

6. Variants of SAT Oracles Using Quantum Counter

Following our preliminary work [61], in this section, we discuss some other applications of the quantum counter in variants of satisfiability, such as the product of SOPs SAT.

6.1. Oracle for SOPs

MAX-SAT can be solved for a Product of any function. In particular, this can be a Product of SOPs. The SOP functions can be implemented with a counter by summing the digits of the counter at the end, using De Morgan’s rule. Each product term is simply a Toffoli gate, and the counter can be checked in a similar way to a regular sum term. Figure 23 presents an example circuit for the function $ab + b\bar{c} + \bar{a}c$.

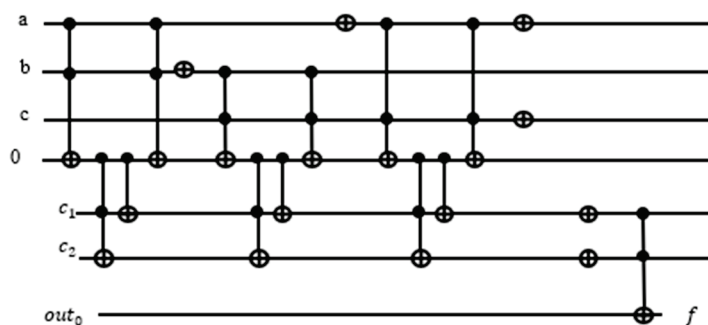


Figure 23. Part of the product of SOP oracle that realizes SOP function $f = ab + b\bar{c} + \bar{a}c$.

6.2. Oracle for Product of SOPs (POSOP SAT)

POSOP functions consist of products of SOP functions. We were not able to find any references to this form of SAT. However, we can take advantage of the fact that every term must be true in a product for the product to be true, and thus we can check against a counter value of the number of terms in order to construct the oracle for POSOP. For example, Figure 24 presents the circuit for function $(\bar{a}b + \bar{a}c)(abc + \bar{b}\bar{c})$.

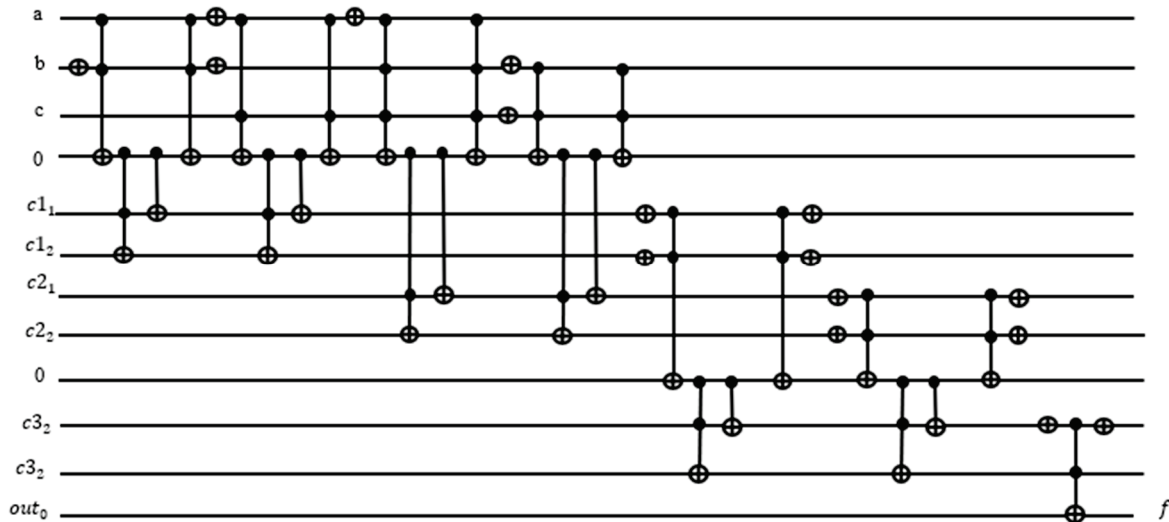


Figure 24. Realization of the Oracle for $f = (\bar{a}b + \bar{a}c)(abc + \bar{b}\bar{c})$, with POSOP SAT.

POSOP circuits are much larger than traditional SOP circuits since an additional counter is required for each SOP term. As such, it may be more advantageous to convert POSOP to a more standard form, such as SOP or POS to be implemented in reversible logic. This depends on a particular problem instance.

6.3. Oracle for Exclusive-or-Sum-of-Products (ESOP)

An Exclusive-or-Sum-of-Products (ESOP) form is an exclusive sum (using the ‘ \oplus ’) operator of product terms. There is not much published on ESOP SAT except for [62], although this is an interesting subject. Grover’s Oracle can be trivially applied to ESOP SAT, a problem that has also not been discussed yet. The advantage of ESOP SAT over OR SAT presented in the previous section is that ESOP SAT can be realized without the need for a large AND gate or a counter. Since every product in the EXOR sum can be implemented as a Toffoli gate, SAT with ESOP can be formulated with just the input qubits and one output qubit. For example, given a function such as $ab \oplus b\bar{c} \oplus \bar{a}\bar{c}$, we can implement Grover’s Oracle, as shown in Figure 25.

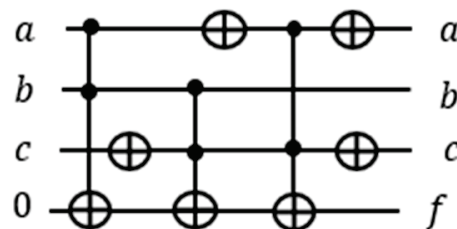


Figure 25. Realization of Oracle $f = ab \oplus b\bar{c} \oplus \bar{a}\bar{c}$ for ESOP SAT realized in Grover’s Algorithm.

7. OR Satisfiability Problems for Electronic Design Automation

In this section, we will show that many EDA (Electronic Design Automation) problems can be reduced to SAT and MAX-SAT. In the most general case, the Satisfiability Decision Function problem is formulated as an arbitrary binary-valued-input, binary-output, and

single-output function. For instance, a product of sums of literals, (the literals are variables negated or not), EXOR of products of literals, and product of sums of products of literals. These functions are created by transforming some natural language or mathematical decision problems, such as, for instance, cryptographic puzzles. The question is to find out for which values of variables the formula for SAT or MAX-SAT is satisfied. In some problems, one has to find all solutions; in some other problems we look for just one solution or only some solutions. For all these variants, we have some freedom to modify Grover's Algorithm, and/or call it several times with modified oracles [60].

Below we will systematically formulate several satisfiability types of problems, starting from the simplest ones. We concentrate on problems that have applications in EDA. Each of these basic problems below can have in addition several variants related to specific applications. Given is a product of terms, each term being a Boolean sum of literals, and each literal being a Boolean variable or its negation. We are interested in the following problems.

Problem 1 (Satisfiability): *Answer Yes if there exists a product of literals that satisfies all terms or No if such a product does not exist. Give the solution as a set of literals.*

Problem 2 (Optimization of the Generalized Petrick function): *Find a product with the minimum number of literals that satisfies all terms or prove that such a product does not exist.*

Problem 3 (Optimization of the Generalized Petrick function-nonnegated literal variant): *Find such a product of literals that satisfies all terms and in which a minimum number of literals is not negated or prove that no such product exists. (The not negated literals will also be called positive literals). In particular, the Petrick Function is positive unate, which it means has only positive literals.*

Problem 4 (MAX-SAT): *Find such set of literals that satisfies the maximum number of terms.*

Problem 5 (Tautology Checking): *Verify whether a function is a Sum of Product Form is a Boolean tautology. Function F is a tautology (all input combinations are 1) when its negation \bar{F} is not satisfiable (all combinations are 0).*

In some variants of these problems, depending on a particular application, we can look for all solutions, all optimal solutions, some optimal solutions, or for a single optimal solution. The central role of the Problem 1 is well-established in computer science. All NP-complete combinational decision problems are equivalent to the Satisfiability Problem [63]. Many reductions of practically important problems to other above problems were shown, including problems from VLSI Design Automation, especially in logic design and state machine design. SAT and MAX-SAT also have many applications in logistics, scheduling, AI, and robotics. Ashenhurst/Curtis Decomposition of Boolean functions can be done in an algorithm that repeatedly applies Satisfiability [64]. Generalized Ashenhurst/Curtis Decomposition was also realized by building a complex oracle for Grover's Algorithm based on the mathematics of Partition Calculus [65]. These SAT-like problem formulations are also of fundamental importance in many algorithms for Boolean minimization, factorization, and multi-level design. The set covering problem is reduced to the minimization of Petrick Function. The reductions of many practically important NP-hard combinatorial optimization problems can also be found in the literature. For instance, the minimization of the Sum of Products Boolean functions can be reduced to the Covering Problem [66] and Covering Problem can be further reduced to the Petrick Function Optimization Problem (PFOP) [67]. Many other problems, like test minimization, can also be reduced to the Covering Problem [66,68]. The problems of Partial Satisfiability and its applications are discussed by K. Lieberherr [69]. Many other reductions to the formulated above problems are discussed in [63,70]. Paper [71] discusses the reduction of three-level NAND circuits, TANT, to the covering-closure problem solved similarly to SAT. A similar problem of the synthesis of networks from negative gates uses the same reduction [72]. A design

automation system [73] was created, in which many problems were first reduced to the few selected “generic” combinatorial optimization problems. These problems include some of the problems listed above.

The problem of minimization of Finite State Machines includes: (1) the Maximum Clique Problem and (2) the problem of finding the minimum closed and a complete subgraph of a graph (Closure/Covering Problem) [71]. The first of these problems, (1), can be reduced to the Petrick Function Optimization Problem (PFOP). The problem of optimum output phase optimization of PLA [74] can be reduced to PFOP. The second problem, (2), can be reduced to the Generalized Petrick Function Optimization Problem (GPFOP), introduced above and illustrated below. Many other problems, like AND/OR graph searching [75], were reduced to the Closure/Covering Problem.

A number of problems (including Boolean minimization [76], layout compaction, and minimization of the number of registers in hardware compilation can be reduced to the Minimal Graph Coloring Problem. Regular layout problems can be reduced to SAT [77]. The Minimal Graph Coloring can be reduced to the Problem of Finding the Maximum Independent Sets, and next the Covering Problem (Maghout algorithm). The Problem of Finding the Maximum Independent Sets can be reduced to PFOP. The PFOP is a particular case of the GPFOP. The role and importance of Tautology and conversion methods from SOP to POS and vice versa in logic design are well known. These problems can also be solved using SAT.

Concluding on OR SAT. In theory, every NP problem can be polynomially reduced to SAT and also to OR 3-SAT. But this is not practical. Many problems can be reduced to graph coloring or maximum clique problems that can be in turn reduced to satisfiability problems.

As we see now, many problems can be solved with quadratic speedup using future quantum computers. A hybrid classical/quantum computer based on Grover tuned to solve variants of SAT problems of various types would be a tremendous asset to all these problems [60].

8. Conclusions

We have designed a novel quantum oracle circuit that requires a logarithmically reduced number of qubits for solving SAT and MAX-SAT problems. The oracle circuit uses the iterative quantum counter circuit, which replaces the ancilla qubits of a global large AND gate for traditional oracle design. Our design showed a significant reduction overall in the number of qubits in Grover’s search algorithm for MAX-SAT. Also, our design calculates the quantum measurable number of the maximum satisfiable OR terms for unsatisfiable Boolean functions. We also compared using Peres and Toffoli gates in terms of quantum cost, where the Peres gates built from truly quantum primitives provide lower quantum costs. Finally, we tested and showed two examples on the IBM QISKIT simulator [23] that provided the expected results. We presented other Variants of SAT oracles that can be designed for the oracle circuit using the quantum counter. Also, we discussed many other potential problems in the area of EDA that can be reduced to SAT and MAX-SAT such that the oracle can be constructed the quantum counter idea.

Suppose one wants to calculate the number of satisfied true minterms for a SAT or MAX-SAT problems. This corresponds to the number of ones in certain Boolean functions. This type of problem is solved using the Quantum Counting Algorithm [14], which in turn is based on Quantum Phase Estimation. Also, many other quantum algorithms use oracles with large AND gate at the output. We plan to work on finding solutions to these problems. The obvious improvement and generalization will be that the yes/no solutions will be extended to solutions for non-solvable problems where the answer will be given to tell how far we are from the solution by creating the “MAX versions” of the problems instead of the current “YES/NO” versions.

Author Contributions: Conceptualization, A.A. and P.J.; Data curation, A.A.; Formal analysis, A.A.; Investigation, A.A.; Methodology, A.A., P.J. and M.P.; Project administration, M.P.; Resources, A.A.; Software, A.A.; Supervision, M.P.; Validation, A.A. and M.P.; Visualization, A.A.; Writing—original

draft, A.A. and P.J.; Writing—review & editing, A.A. and M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This article’s publication was funded by the Portland State University Open Access Article Processing Charge Fund, grant number: PO 23-24362, administered by the Portland State University Library.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marques-Silva, J.; Glass, T. Combinational equivalence checking using satisfiability and recursive learning. In Proceedings of the Conference on Design, Automation and Test in Europe, Munich, Germany, 1 January 1999; p. 33.
2. Konuk, H.; Larrabee, T. Explorations of sequential ATPG using Boolean satisfiability. In Proceedings of the Digest of Papers Eleventh Annual 1993 IEEE VLSI Test Symposium, Atlantic City, NJ, USA, 6–8 April 1993; IEEE: Manhattan, NY, USA, 1993; pp. 85–90.
3. Biere, A.; Cimatti, A.; Clarke, E.M.; Fujita, M.; Zhu, Y. Symbolic model checking using SAT procedures instead of BDDs. In Proceedings of the 36th Annual ACM/IEEE Design Automation Conference, New Orleans, LA, USA, 21–25 June 1999; pp. 317–320.
4. Hong, T.; Li, Y.; Park, S.B.; Mui, D.; Lin, D.; Kaleq, Z.A.; Hakim, N.; Naeimi, H.; Gardner, D.S.; Mitra, S. QED: Quick error detection tests for effective post-silicon validation. In Proceedings of the 2010 IEEE International Test Conference, Austin, TX, USA, 2–4 November 2010; IEEE: Manhattan, NY, USA, 2010; pp. 1–10.
5. Wang, P.W.; Donti, P.; Wilder, B.; Kolter, Z. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; PMLR: New York, NY, USA, 2019; pp. 6545–6554.
6. Cook, S.A. The complexity of theorem-proving procedures. In Proceedings of the Third Annual ACM Symposium on Theory of Computing, Shaker Heights, OH, USA, 3–5 May 1971; pp. 151–158.
7. Kohli, R.; Krishnamurti, R.; Mirchandani, P. The minimum satisfiability problem. *SIAM J. Discret. Math.* **1994**, *7*, 275–283. [CrossRef]
8. Biere, A.; Heule, M.; van Maaren, H. (Eds.) *Handbook of Satisfiability*; IOS Press: Washington, DC, USA, 2009; Volume 185.
9. Fu, Z.; Malik, S. On solving the partial MAX-SAT problem. In Proceedings of the International Conference on Theory and Applications of Satisfiability Testing, Seattle, WA, USA, 12–15 August 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 252–265.
10. Berg, O.J.; Hyttinen, A.J.; Järvisalo, M.J. Applications of MaxSAT in data analysis. In Proceedings of the Pragmatics of SAT 2015 and 2018, Oxford, UK, 7 July 2018.
11. Berg, J.; Järvisalo, M. Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability. *Artif. Intell.* **2017**, *244*, 110–142. [CrossRef]
12. Berg, J.; Järvisalo, M.; Malone, B. Learning optimal bounded treewidth Bayesian networks via maximum satisfiability. In Proceedings of the Artificial Intelligence and Statistics, Reykjavik, Iceland, 22–25 April 2014; PMLR: New York, NY, USA, 2014; pp. 86–95.
13. Hyttinen, A.; Saikko, P.; Järvisalo, M. A core-guided approach to learning optimal causal graphs. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, Australia, 19–25 August 2017.
14. Malioutov, D.; Meel, K.S. MLIC: A MaxSAT-based framework for learning interpretable classification rules. In Proceedings of the International Conference on Principles and Practice of Constraint Programming, Lille, France, 27–31 August 2018; Springer: Cham, Switzerland, 2018; pp. 312–327.
15. Dimitrova, R.; Ghasemi, M.; Topcu, U. Maximum realizability for linear temporal logic specifications. In Proceedings of the International Symposium on Automated Technology for Verification and Analysis, Los Angeles, CA, USA, 7–10 October 2018; Springer: Cham, Switzerland, 2018; pp. 458–475.
16. Zhang, L.; Bacchus, F. MAXSAT heuristics for cost optimal planning. In Proceedings of the AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; Volume 26, pp. 1846–1852.
17. Muise, C.; Beck, J.C.; McIlraith, S.A. Optimal partial-order plan relaxation via MaxSAT. *J. Artif. Intell. Res.* **2016**, *57*, 113–149. [CrossRef]
18. Demirović, E.; Musliu, N.; Winter, F. Modeling and solving staff scheduling with partial weighted maxSAT. *Ann. Oper. Res.* **2019**, *275*, 79–99. [CrossRef]
19. Safarpour, S.; Mangassarian, H.; Veneris, A.; Liffiton, M.H.; Sakallah, K.A. November. Improved design debugging using maximum satisfiability. In *Formal Methods in Computer Aided Design (FMCAD’07)*; IEEE: Cham, Switzerland, 2007; pp. 13–19.
20. Chen, Y.; Safarpour, S.; Veneris, A.; Marques-Silva, J. Spatial and temporal design debug using partial MaxSAT. In Proceedings of the 19th ACM Great Lakes symposium on VLSI, Boston Area, MA, USA, 10–12 May 2009; pp. 345–350.
21. Chen, Y.; Safarpour, S.; Marques-Silva, J.; Veneris, A. Automated design debugging with maximum satisfiability. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2010**, *29*, 1804–1817. [CrossRef]

22. Jose, M.; Majumdar, R. Cause clue clauses: Error localization using maximum satisfiability. *ACM SIGPLAN Not.* **2011**, *46*, 437–446. [CrossRef]
23. Zhu, C.S.; Weissenbacher, G.; Malik, S. Post-silicon fault localisation using maximum satisfiability and backbones. In Proceedings of the 2011 Formal Methods in Computer-Aided Design (FMCAD), Austin, TX, USA, 30 October–2 November 2011; IEEE: Cham, Switzerland, 2011; pp. 63–66.
24. Wickramarachchi, G.T.; Qardaji, W.H.; Li, N. An efficient framework for user authorization queries in RBAC systems. In Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, Stresa, Italy, 3–5 June 2009; pp. 23–32.
25. Liao, X.; Zhang, H.; Koshimura, M. Reconstructing AES key schedule images with SAT and MaxSAT. *IEICE Trans. Inf. Syst.* **2016**, *99*, 141–150. [CrossRef]
26. Shabani, A.; Alizadeh, B. PMTP: A MAX-SAT-based approach to detect hardware trojan using propagation of maximum transition probability. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2018**, *39*, 25–33. [CrossRef]
27. Feng, Y.; Bastani, O.; Martins, R.; Dillig, I.; Anand, S. Automated synthesis of semantic malware signatures using maximum satisfiability. *arXiv* **2016**, arXiv:1608.06254.
28. Lin, P.C.K.; Khatri, S.P. Application of Max-SAT-based ATPG to optimal cancer therapy design. *BMC Genom.* **2012**, *13*, 1–10. [CrossRef]
29. Guerra, J.; Lynce, I. Reasoning over biological networks using maximum satisfiability. In Proceedings of the International Conference on Principles and Practice of Constraint Programming, Québec City, QC, Canada, 8–12 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 941–956.
30. Martins, R. Solving RNA alignment with MaxSAT. In *MaxSAT Evaluation*; University of Helsinki: Helsinki, Finland, 2017; p. 29.
31. Graça, A.; Marques-Silva, J.; Lynce, I.; Oliveira, A.L. Haplotype inference with pseudo-Boolean optimization. *Ann. Oper. Res.* **2011**, *184*, 137–162. [CrossRef]
32. Li, C.M.; Quan, Z. An efficient branch-and-bound algorithm based on maxsat for the maximum clique problem. In Proceedings of the Twenty-fourth AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010.
33. Li, C.M.; Jiang, H.; Xu, R.C. Incremental MaxSAT reasoning to reduce branches in a branch-and-bound algorithm for MaxClique. In Proceedings of the International Conference on Learning and Intelligent Optimization, Lille, France, 12–15 January 2015; Springer: Cham, Switzerland, 2015; pp. 268–274.
34. Fang, Z.; Li, C.M.; Qiao, K.; Feng, X.; Xu, K. Solving Maximum Weight Clique Using Maximum Satisfiability Reasoning. In Proceedings of the ECAI, Prague, Czech, 18–22 August 2014; pp. 303–308.
35. Berg, J.; Järvisalo, M. SAT-based approaches to treewidth computation: An evaluation. In Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, Limassol, Cyprus, 10–12 November 2014; IEEE: Cham, Switzerland, 2014; pp. 328–335.
36. Morgado, A.; Marques-Silva, J. Combinatorial optimization solutions for the maximum quartet consistency problem. *Fundam. Inform.* **2010**, *102*, 363–389. [CrossRef]
37. Smyth, K.; Hoos, H.H.; Stützle, T. Iterated robust tabu search for MAX-SAT. In Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence, Halifax, NS, Canada, 11–13 June 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 129–144.
38. Mastrolilli, M.; Gambardella, L.M. Maximum satisfiability: How good are tabu search and plateau moves in the worst-case? *Eur. J. Oper. Res.* **2005**, *166*, 63–76. [CrossRef]
39. Cai, S.; Lei, Z. Old techniques in new ways: Clause weighting, unit propagation and hybridization for maximum satisfiability. *Artif. Intell.* **2020**, *287*, 103354. [CrossRef]
40. Marchiori, E.; Rossi, C. A flipping genetic algorithm for hard 3-SAT problems. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), Orlando, FL, USA, 13–17 July 1999; pp. 393–400.
41. Layeb, A.; Deneche, A.H.; Meshoul, S. A new artificial immune system for solving the maximum satisfiability problem. In Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Cordoba, Spain, 1–4 June 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 136–142.
42. Munawar, A.; Wahib, M.; Munetomo, M.; Akama, K. Hybrid of genetic algorithm and local search to solve max-sat problem using NVIDIA CUDA framework. *Genet. Program. Evolvable Mach.* **2009**, *10*, 391–415. [CrossRef]
43. Lardeux, F.; Saubion, F.; Hao, J.K. GASAT: A genetic local search algorithm for the satisfiability problem. *Evol. Comput.* **2006**, *14*, 223–253. [CrossRef]
44. Davis, M.; Logemann, G.; Loveland, D. A machine program for theorem-proving. *Commun. ACM* **1962**, *5*, 394–397. [CrossRef]
45. Li, C.M.; Manyà, F.; Planes, J. Exploiting unit propagation to compute lower bounds in branch and bound Max-SAT solvers. In Proceedings of the International Conference on Principles and Practice of Constraint Programming, Sitges, Spain, 1–5 October 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 403–414.
46. Li, C.M.; Xu, Z.; Coll, J.; Manyà, F.; Habet, D.; He, K. Combining clause learning and branch and bound for MaxSAT. In Proceedings of the 27th International Conference on Principles and Practice of Constraint Programming (CP 2021), Montpellier, France, 25–29 October 2021; Schloss Dagstuhl-Leibniz-Zentrum für Informatik: Wadern, Germany, 2021.
47. Gu, J. Efficient local search for very large-scale satisfiability problems. *ACM SIGART Bull.* **1992**, *3*, 8–12. [CrossRef]

48. Bian, Z.; Chudak, F.; Macready, W.; Roy, A.; Sebastiani, R.; Varotti, S. Solving sat and maxsat with a quantum annealer: Foundations and a preliminary report. In Proceedings of the International Symposium on Frontiers of Combining Systems, Brasília, Brazil, 27–29 September 2017; Springer: Cham, Switzerland, 2017; pp. 153–171.
49. Cheng, S.T.; Tao, M.H. Quantum cooperative search algorithm for 3-SAT. *J. Comput. Syst. Sci.* **2007**, *73*, 123–136. [CrossRef]
50. Lee, S.; Lee, S.-J.; Kim, T.; Biamonte, J.; Perkowski, M. The cost of Quantum Gate Primitives. *J. Mult. Valued Log. Soft Comput.* **2006**, *12*, 561–573.
51. Barenco, A.; Bennett, C.H.; Cleve, R.; DiVincenzo, D.P.; Margolus, N.; Shor, P.; Sleator, T.; Smolin, J.A.; Weinfurter, H. Elementary gates for quantum computation. *Phys. Rev. A* **1999**, *52*, 3457. [CrossRef] [PubMed]
52. Maslov, D.; Dueck, G.W. Improved quantum cost for n-bit Toffoli gates. *Electron. Lett.* **2003**, *39*, 1790–1791. [CrossRef]
53. Peres, A. Reversible logic and quantum computers. *Phys. Rev. A* **1985**, *32*, 3266. [CrossRef] [PubMed]
54. Szyppowski, M.; Kerntopf, P. Low quantum cost realization of generalized peres and toffoli gates with multiple-control signals. In Proceedings of the 2013 13th IEEE International Conference on Nanotechnology (IEEE-NANO 2013), Beijing, China, 5–8 August 2013; IEEE: Manhattan, NY, USA, 2013; pp. 802–807.
55. Nielsen, M.A.; Chuang, I. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2002.
56. Wong, R.; Chang, W.L. Quantum speedup for protein structure prediction. *IEEE Trans. NanoBiosci.* **2021**, *20*, 323–330. [CrossRef]
57. Chang, W.L.; Chen, J.C.; Chung, W.Y.; Hsiao, C.Y.; Wong, R.; Vasilakos, A.V. Quantum speedup and mathematical solutions of implementing bio-molecular solutions for the independent set problem on IBM quantum computers. *IEEE Trans. NanoBiosci.* **2021**, *20*, 354–376. [CrossRef]
58. Wong, R.; Chang, W.L. Fast quantum algorithm for protein structure prediction in hydrophobic-hydrophilic model. *J. Parallel Distrib. Comput.* **2022**, *164*, 178–190. [CrossRef]
59. Aleksandrowicz, G.; Alexander, T.; Barkoutsos, P.; Bello, L.; Ben-Haim, Y.; Bucher, D.; Cabrera-Hernández, F.J.; Carballo-Franquis, J.; Chen, A.; Chen, C.F.; et al. *Qiskit: An Open-Source Framework for Quantum Computing*; Zenodo: Honolulu, HI, USA, 2019.
60. Perkowski, M. Inverse Problems, Constraint Satisfaction, Reversible Logic, Invertible Logic and Grover Quantum Oracles for Practical Problems. In Proceedings of the International Conference on Reversible Computation, Oslo, Norway, 9–10 July 2020; Springer: Cham, Switzerland, 2020; pp. 3–32.
61. Alasow, A.; Perkowski, M. Quantum Algorithm for Maximum Satisfiability. In Proceedings of the 2022 IEEE 52nd International Symposium on Multiple-Valued Logic (ISMVL), Dallas, TX, USA, 18–20 May 2022; pp. 27–34.
62. Csanky, L. On the Generalized Reed-Muller Canonical Form of Boolean Functions: Research Project. Ph.D. Thesis, University of California, Berkeley, CA, USA, 1972.
63. Garey, M.R.; Johnson, D.S. *A Guide to the Theory of NP-Completeness. Computers and Intractability*; W. H. Freeman & Co: New York, NY, USA, 1979.
64. Lin, H.P.; Jiang, J.H.R.; Lee, R.R. To SAT or not to SAT: Ashenurst decomposition in a large scale. In Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 10–13 November 2008; IEEE: Manhattan, NY, USA, 2008; pp. 32–37.
65. Li, Y.; Tsai, E.; Perkowski, M.; Song, X. Grover-based Ashenurst-Curtis decomposition using quantum language quipper. *Quantum Inf. Comput.* **2019**, *19*, 35–66. [CrossRef]
66. Breuer, M.A.; Preiss, R.J. *Design Automation of Digital Systems*; Prentice Hall: Hoboken, NJ, USA, 1972; Volume 1.
67. Slagle, J.R. *Artificial Intelligence: The Heuristic Programming Approach*; McGraw-Hill: New York, NY, USA, 1971.
68. Kohavi, Z.; Jha, N.K. *Switching and Finite Automata Theory*; Cambridge University Press: Cambridge, UK, 2009.
69. Lieberherr, K.; Specker, E. Complexity of partial satisfaction. In Proceedings of the 20th Annual Symposium on Foundations of Computer Science (sfcs 1979), Washington, DC, USA, 29–31 October 1979; IEEE: Manhattan, NY, USA, 1979; pp. 132–139.
70. Perkowski, M. *State-Space Approach to the Design of a Multipurpose Problem-Solver for Logic Design*; KU Leuven: North Holland, Amsterdam, 1978.
71. Perkowski, M. Synthesis of multioutput three level NAND networks. In Proceedings of the Seminar on Computer Aided Design, Budapest, Hungary, 3–5 November 1976; pp. 238–265.
72. Perkowski, M. Minimization of two-level networks from negative gates. In Proceedings of the Midwest, Lincoln, Nebraska, 10–12 August 1986; Volume 86, pp. 756–761.
73. Perkowski, M.; Liu, J.; Brown, J. A System for Fast Prototyping of Logic Design Programs. In Proceedings of the 1987 Midwest Symposium on Circuits and Systems, Syracuse, NY, USA, 17–18 August 1987.
74. Sasao, T. Input variable assignment and output phase optimization of PLA's. *IEEE Trans. Comput.* **1984**, *33*, 879–894. [CrossRef]
75. Nilsson, J. *Problem-Solving Methods in Artificial Intelligence*; McGraw-Hill: New York, NY, USA, 1971.
76. Nguyen, L.B.; Perkowdki, M.A.; Goldstein, N.B. Palmimi—Fast Boolean minimizer for personal computers. In Proceedings of the 24th ACM/IEEE Design Automation Conference, Miami Beach, FL, USA, 28 June–1 July 1987; pp. 615–621.
77. Perkowski, M.; Mishchenko, A. Logic synthesis for regular layout using satisfiability. In Proceedings of the International Symposium on Boolean Problems, Freiberg, Germany, 19–20 September 2002.

Article

Using Variational Quantum Algorithm to Solve the LWE Problem

Lihui Lv ^{1,2}, Bao Yan ^{1,2}, Hong Wang ^{1,2,*}, Zhi Ma ^{1,2,*}, Yangyang Fei ^{1,2}, Xiangdong Meng ^{1,2} and Qianheng Duan ^{1,2}

¹ State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

² Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China

* Correspondence: redwang@meac-skl.cn (H.W.); ma.zhi@meac-skl.cn (Z.M.)

Abstract: The variational quantum algorithm (VQA) is a hybrid classical–quantum algorithm. It can actually run in an intermediate-scale quantum device where the number of available qubits is too limited to perform quantum error correction, so it is one of the most promising quantum algorithms in the noisy intermediate-scale quantum era. In this paper, two ideas for solving the learning with errors problem (LWE) using VQA are proposed. First, after reducing the LWE problem into the bounded distance decoding problem, the quantum approximation optimization algorithm (QAOA) is introduced to improve classical methods. Second, after the LWE problem is reduced into the unique shortest vector problem, the variational quantum eigensolver (VQE) is used to solve it, and the number of qubits required is calculated in detail. Small-scale experiments are carried out for the two LWE variational quantum algorithms, and the experiments show that VQA improves the quality of the classical solutions.

Keywords: quantum; LWE; QAOA; VQE; KYBER

Citation: Lv, L.; Yan, B.; Wang, H.; Ma, Z.; Fei, Y.; Meng, X.; Duan, Q. Using Variational Quantum Algorithm to Solve the LWE Problem. *Entropy* **2022**, *24*, 1428. <https://doi.org/10.3390/e24101428>

Academic Editors: Brian R. La Cour and Giuliano Benenti

Received: 30 August 2022

Accepted: 2 October 2022

Published: 8 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Lattice theory is a classic subject in mathematical research, and it has critical applications in many fields such as the optimization problem and information coding. In 1996, Ajtai [1] proved that the worst-case hardness of the shortest vector problem (SVP) can be reduced to the hardness of SVP in a class of random lattices, thus providing provable security of lattice-based cryptosystems. Since then, various lattice-based cryptosystems are proposed, such as Ajtai-Dwork [2] and the Number Theory Research Unit [3].

In 2005, Regev proposed an encryption algorithm based on LWE [4]. Compared with previous lattice-based cryptosystems, the ciphertext size and key size of LWE-based cryptosystems are greatly reduced. Therefore, LWE began to be applied to many cryptographic primitives, such as Key-Dependent Message [5], Fully Homomorphic Encryption [6] and so forth. In July 2022, The National Institute of Standards and Technology completed the third round of the Post-Quantum Cryptography standardization process, and four candidate algorithms have been announced. Among them, the public-key encryption algorithm CRYSTALS-KYBER [7] and the digital signature algorithm CRYSTALS-Dilithium [8] are constructed based on the module-LWE problem. Therefore, analyzing LWE algorithms is important to the security of post-quantum cryptography.

The analysis methods of LWE can be classified into combinatorial methods, algebraic methods, lattice methods and the exhaustive search. The combinatorial method mainly refers to an extended application of the Gaussian elimination [9], but it requires a large number of samples. The algebraic method refers to the Arora-Ge algorithm [10], and the complexity is also exponential in the number of LWE dimensions. There are three main lattice methods: the dual method is used to attack decision-LWE instances by solving the short integer solution problem on the dual lattice [1]; the decoding method is used to directly solve the bounded distance decoding problem (BDD) on the original lattice [11,12];

the primary method is used to further reduce the BDD problem to the Unique-SVP problem [13–15]. The exhaustive search is not suitable for practical applications because of its high time complexity.

At the same time, VQA, such as QAOA [16], VQE [17], and FQE [18], has become the most suitable technology to achieve quantum advantage using noisy intermediate-scale quantum (NISQ) devices. Some works have studied how to solve hard lattice problems by VQA. Paper [19] analyzed the energy gaps between the first three excited states of the Hamiltonian when solving SVP with low dimension by quantum adiabatic computation. The conclusion in [19] inspired the use of QAOA to find the ground state. Ref. [20] calculated the number of qubits for special lattices and concluded that $1.5n \log n + n + \log(\det(\mathcal{L}))$ qubits sufficed to obtain the shortest vector of n -dimensional lattice \mathcal{L} . Ref. [21] proposed to solve SVP by VQE and also pointed out that their algorithm was not limited to special lattices.

The work in this paper consists of two aspects. Firstly, we use QAOA to optimize the Nearest Plane algorithm and solve LWE. Secondly, inspired by Ref. [21], we propose a hybrid algorithm using VQE to attack LWE and calculate the number of qubits required to attack specific LWE cryptosystems. For the two LWE algorithm ideas, we conduct small-scale experimental simulations. The experiments show that QAOA improves the quality of classical solutions, and the quality of solutions obtained by VQE is at least equal to that of classical solutions when the memory is big enough.

2. Preliminary

2.1. Lattice Theory

Let $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n \in \mathbb{R}^m$ be a set of linearly independent vectors, and the lattice generated by $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$ is

$$\Lambda = \mathcal{L}(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n) = \{\alpha_1 \mathbf{b}_1 + \alpha_2 \mathbf{b}_2 + \dots + \alpha_n \mathbf{b}_n \mid \alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{Z}\}.$$

In cryptography applications, the lattice dimension is n . Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the q -ary lattice refers to

$$\Lambda_q(\mathbf{A}^T) = \{\mathbf{x} \in \mathbb{Z}^m \mid \exists \mathbf{y} \in \mathbb{Z}^n, s.t. \mathbf{x} \equiv \mathbf{y} \mathbf{A}^T \pmod{q}\}.$$

For a lattice \mathcal{L} and its basis matrix $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$, the volume of the lattice is $vol(\mathcal{L}) = \sqrt{\det(\mathbf{B}^T \mathbf{B})}$ and the fundamental domain is $\mathcal{P}_{1/2}(\mathbf{B}) = \{\sum_{i=1}^n \alpha_i \mathbf{b}_i \mid \alpha_i \in [-\frac{1}{2}, \frac{1}{2}]\}$. The distance between \mathcal{L} and vector $\mathbf{v} \in \mathbb{R}^m$ is $dist(\mathbf{v}, \mathcal{L}) = \min\{\|\mathbf{v} - \mathbf{y}\| \mid \mathbf{y} \in \mathcal{L}\}$. The i -th successive minima $\lambda_i(\mathcal{L})$ is the minimum radius of the ball centered at the origin, which contains i linearly independent vectors in the lattice. Let \mathcal{L} be an n -dimensional lattice; then, the Gaussian heuristic states that $\lambda_1(\mathcal{L}) \approx \sqrt{\frac{n}{2\pi e}} vol(\mathcal{L})^{1/n}$.

Definition 1. (Shortest vector problem, SVP) For a lattice \mathcal{L} , the SVP problem asks to find a nonzero lattice vector \mathbf{v} that minimizes the Euclidean nonzero norm $\|\mathbf{v}\|$.

Definition 2. (Closest vector problem, CVP) For a lattice \mathcal{L} , given a target vector $\mathbf{t} \in \mathbb{R}^m$ that is not in \mathcal{L} , the CVP problem asks to find a lattice vector \mathbf{v} that minimizes the Euclidean norm $\|\mathbf{v} - \mathbf{t}\|$.

Definition 3. (Unique shortest vector problem, Unique-SVP) For a lattice \mathcal{L} satisfying $\lambda_2(\mathcal{L}) > \gamma \lambda_1(\mathcal{L})$, where $\gamma \gg 1$, the uSVP problem asks to find the shortest nonzero lattice vector.

Definition 4. (Bounded distance decoding, BDD) For a target vector $\mathbf{t} \in \mathbb{R}^m$ that is not in the given lattice \mathcal{L} , which satisfies $dist(\mathbf{t}, \mathcal{L}) < \gamma \lambda_1(\mathcal{L})$, where $\gamma < 1/2$, the BDD problem asks to find a nonzero lattice vector \mathbf{v} that minimizes the Euclidean norm $\|\mathbf{v} - \mathbf{t}\|$.

Algorithms for hard problems on lattices usually perform lattice basis reduction as a preprocessing module, because a sufficiently good basis improves the algorithms' success

probability. The LLL (Lenstra–Lenstra–Lovász) algorithm [22] and the BKZ (block–Korkin–Zolotarev) algorithm [23] are two famous basis reduction algorithms.

Before introducing the LLL reduction algorithm, we first explain the Gram–Schmidt orthogonalization. With a lattice basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$, one can calculate its Gram–Schmidt orthogonalization $\mathbf{B}^* = [\mathbf{b}_1^*, \mathbf{b}_2^*, \dots, \mathbf{b}_n^*]$ by the recursion $\mathbf{b}_1^* = \mathbf{b}_1, \mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*$ for $i = 2, 3, \dots, n$, where the Gram–Schmidt coefficients $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle$. The LLL algorithm was proposed in 1982, and the formal description of LLL reduction is detailed as shown in Algorithm 1.

Algorithm 1 LLL algorithm.

Input: lattice basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$, a reduction parameter δ .

Output: a δ -LLL reduced basis

- 1: Calculate the Gram–Schmidt orthogonalization $\mathbf{B}^* = [\mathbf{b}_1^*, \mathbf{b}_2^*, \dots, \mathbf{b}_n^*]$.
 - 2: **for** $i = 2, 3, \dots, n$ **do**
 - 3: **for** $j = i - 1, i - 2, \dots, 1$ **do**
 - 4: $\mathbf{b}_i = \mathbf{b}_i - c_{i,j} \mathbf{b}_j$, where $c_{i,j} = \lceil \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle \rceil$;
 - 5: **end for**
 - 6: **end for**
 - 7: **if** $\exists i$, s.t. $\delta \|\mathbf{b}_{i-1}^*\|^2 > \|\mu_{i,i-1} \mathbf{b}_{i-1}^* + \mathbf{b}_i^*\|^2$ **then**
 - 8: Swap \mathbf{b}_{i-1} and \mathbf{b}_i ;
 - 9: Go to Step 1.
 - 10: **end if**
 - 11: **return** \mathbf{B} .
-

The BKZ algorithm is derived from the KZ (Korkine–Zolotarev) reduction. BKZ uses the block reduction to improve the LLL algorithm and outputs an (δ, β) -BKZ reduced basis. To be specific, the BKZ algorithm runs the enumeration algorithm on the sub-lattice with block size β and obtains its shortest vector. After inserting the shortest vector into the original basis, LLL reduction with parameter δ is applied on the entire basis to remove the linear dependency. BKZ performs the above steps iteratively until the basis is no longer updated.

2.2. The LWE Problem

Definition 5. (Learning with errors distribution) Let $n, q > 0$ be integers, and $\alpha \in \{0, 1\}$. Let $\mathbf{s} \in \mathbb{Z}_q^n$ be a secret vector. The LWE distribution $\chi_{\mathbf{s}, \alpha}$ refers to $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{a} \in \mathbb{Z}_q^n$ is uniformly selected randomly and e is a discrete Gaussian error with standard deviation αq .

Definition 6. (Learning with errors problem) Let $n, m, q > 0$ be integers, $\alpha > 0$. Given m samples $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i), i = 1, 2, \dots, m$, the search-LWE problem asks to recover the secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, and the decision-LWE problem asks to determine whether the samples are sampled according to $\chi_{\mathbf{s}, \alpha}$ or the uniform distribution.

Now, we review some lattice-based methods for analyzing the LWE problem. In general, the decision-LWE can be solved by the short integer solution strategy, and the search-LWE can be attacked by the BDD strategy or the inhomogeneous short integer solution strategy. Now, we mainly describe the decoding method and the primal method in the BDD strategy.

The LWE problem can be written in a matrix form $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$. Given $q \in \mathbb{Z}$, $\mathbf{c} \in \mathbb{Z}_q^m, \mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_m]^T \in \mathbb{Z}_q^{m \times n}$, the problem recovers \mathbf{s} . The basic idea of the decoding method is to regard \mathbf{c} as the target vector and then use the Nearest Plane algorithm to find the closest vector in $\Lambda_q(\mathbf{A})$. Assuming the basis of $\Lambda_q(\mathbf{A})$ is \mathbf{B} , before applying the Babai’s Nearest Plane algorithm, \mathbf{B} should be preprocessed to a Gram–Schmidt basis \mathbf{B}^* . The strategy outputs \mathbf{s} if and only if \mathbf{e} lies in $\mathbf{s} + \mathbb{P}_{1/2}(\mathbf{B}^*)$, which is determined by the quality of the basis. Lindner and Peikert improved Babai’s algorithm by admitting a time/success

trade-off. To be specific, in each iteration, the Lindner–Peikert Nearest Plane algorithm chooses several close hyperplanes instead of only the closest hyperplane. The idea stretches $\mathbb{P}_{1/2}(\mathbf{B}^*)$ to a cube-like shape and amplifies the success probability.

The primal method is to solve LWE by reducing BDD to the Unique-SVP problem using an embedding technique. The embedding method is to construct a $(m + 1)$ -dimensional lattice $\mathbf{B}' = \begin{bmatrix} \mathbf{B} & \mathbf{c} \\ \mathbf{0} & t \end{bmatrix}$. Obviously, the short vector $[-\mathbf{e}, t] \in \mathbb{Z}_q^{m+1}$ is in \mathbf{B}' . Therefore, solving the Unique-SVP instance recovers the error vector and the secret vector in passing.

2.3. Variational Quantum Algorithm

VQA is a quantum–classical hybrid algorithm that is considered to be implemented on NISQ devices. Therefore, VQA is expected to demonstrate quantum advantages over classical computers when solving some specific problems. The workflow of VQA is shown in Algorithm 2.

Algorithm 2 VQA algorithm.

Input: An optimization problem.

Output: Parameters in the parameterized quantum circuit.

- 1: Construct the objective function.
 - 2: Construct the parameterized quantum circuit.
 - 3: Prepare the quantum state and measure the expectation value.
 - 4: Use a classical optimizer to determine new parameters.
 - 5: Iterate the procedure in step 3 and 4 until the convergence of the value.
 - 6: **return** the final parameters.
-

There are four important modules in VQA [24,25]: the objective function refers to the cost function that needs to be minimized; the parameterized quantum circuit refers to a set of unitary operators that manipulate parameters in the optimization process; the measurement scheme calculates the expectation value; the classical optimizer outputs the parameters that minimize the objective function.

First, VQA encodes the problem into an objective function O . Let the probability of measuring qubit q in state $|0\rangle$ be p^q ; then, the objective function of VQA can be expressed as $\min_{\theta} O(\theta, \{p(\theta)\})$.

Because it is inconvenient to obtain the function value directly by the measurement probability, the expectation value of a Hamiltonian is introduced, and constructing the objective function is equivalent to constructing its corresponding Hamiltonian. The Hamiltonian is a quantum operator that encodes the information of a physical system. Its expectation value corresponds to the energy of a quantum state. The ground state of the Hamiltonian is often used as the minimization target of a VQA problem. In practice, the expectation value of Hamiltonian H

$$\langle H \rangle_{U(\theta)} = \langle 0 | U^\dagger(\theta) H U(\theta) | 0 \rangle$$

is used to describe the measurement results of the quantum state produced by $U(\theta)$. Therefore, the objective function is

$$\min_{\theta} O(\theta, \langle H \rangle_{U(\theta)}).$$

If the objective function is defined more compactly, it can be described as $\min_{\theta} \langle H \rangle_{U(\theta)}$. The objective functions or cost functions constructed in this paper are all in the compact form.

Second, parameterized quantum circuits are a set of unitary operations that depend on parameters. The parameterized quantum circuit acting on quantum state $|\psi_0\rangle$ can be expressed as

$$|\psi(\theta)\rangle = U(\theta)|\psi_0\rangle,$$

where θ are variational parameters.

Most ansatz U can be classified as problem-inspired or hardware-efficient. The construction of problem-inspired ansatz requires the information of specific problems. For example, the united coupled cluster ansatz in quantum chemistry is constructed by a parameterized cluster operator $T(\theta)$ and acts on the ground state $|\psi_{HF}\rangle$ in the way of $|\psi(\theta)\rangle = e^{T(\theta)-T^\dagger(\theta)}|\psi_{HF}\rangle$. Ansatz in the QAOA algorithm is also problem-inspired, and its construction is shown in Section 3. Hardware-efficient ansatz is usually expressed as $\prod_{k=1}^D U_k(\theta_k)W_k$, where $\theta = (\theta_1, \dots, \theta_D)$, $U_k(\theta_k) = e^{-i\theta_k V_k}$ is a unitary operator derived from Hamiltonian V_k , and W_k is an unparametrized unitary operator.

Third, in order to obtain the information of quantum state, we need to measure it in the computational basis and calculate the expectation value of the objective function. The expectation value of the operator σ^z can be obtained by $\langle \sigma^z \rangle = \langle \psi | \sigma^z | \psi \rangle = |\alpha|^2 - |\beta|^2$, where $|\alpha|^2$ and $|\beta|^2$ are the probabilities to measure $|\psi\rangle$ in state $|0\rangle$ and $|1\rangle$. The measurement defined by σ^x and σ^y is first transformed into the basis of σ^z by $\sigma^x = R_y^\dagger(\pi/2)\sigma^z R_y(\pi/2)$, $\sigma^y = R_x^\dagger(\pi/2)\sigma^z R_x(\pi/2)$ and then measured on a σ^z basis. Any Pauli string is measured in the same way, except that it is measured on each qubit separately.

QAOA and VQE are two quantum variational algorithms, so they can be used to solve optimization problems. Since a quantum circuit is equivalent to a tensor product, it can be represented on a classical computer, and the expectation value of the cost function can be calculated, but the memory it consumes grows exponentially with the size of the problem. For a quantum computer, repeating the preparation of ansatz state and the quantum measurements, the expectation can be obtained. The quantum resources it consumes increase polynomially with the scale of the problem, thus showing its superiority over classical algorithms.

3. The Decoding Method for Solving LWE

This section applies the decoding method to solve LWE. When solving BDD, we use QAOA to improve Babai's Nearest Plane algorithm.

First, construct a q -ary lattice $\Lambda_q(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}_q^m | \exists \mathbf{x} \in \mathbb{Z}^n, s.t. \mathbf{v} \equiv \mathbf{A}\mathbf{x} \text{ mod } q\}$, whose lattice basis is equivalent to $\mathbf{B} = [\mathbf{A} | q\mathbf{I}_m]^T \in \mathbb{Z}^{(m+n) \times m}$. Second, perform elementary row transformations on \mathbf{B} and obtain a basis matrix $[\mathbf{b}'_1, \dots, \mathbf{b}'_m]^T \in \mathbb{Z}^{m \times m}$. Third, solve CVP with the target vector \mathbf{c} , and finally output the closest vector \mathbf{w} . The last step is to use the Gaussian elimination to recover $\mathbf{s} = \mathbf{A}^{-1}\mathbf{w}$.

Now, introduce the application of QAOA when improving Babai's Nearest Plane algorithm. Babai's Nearest Plane algorithm consists of two steps: first, perform the LLL reduction on the input lattice basis, and then find the linear combination in the reduced basis so that it forms the closest lattice vector to the given target vector. The formal description is detailed as Algorithm 3.

In the loop, $u_j = \lceil \langle \mathbf{b}, \mathbf{b}_j^* \rangle / \langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle \rceil$ only takes one value by the "round to the nearest integer" function. Through experiments, it is found that when the value range is expanded to $\{u_j + x | x = 0, 1, -1\}$, a better solution is often obtained. In a classical algorithm, the process requires an exponential increase in computation with respect to the lattice dimension n . In quantum computing, due to quantum properties, the computing complexity can be greatly reduced. Therefore, we now introduce the method of encoding the random floating in u_j in two qubits and solving the optimization problem by QAOA.

Algorithm 3 Babai’s Nearest Plane algorithm.

Input: lattice basis $\mathbf{B}' = [\mathbf{b}'_1, \mathbf{b}'_2, \dots, \mathbf{b}'_m] \in \mathbb{R}^{m \times m}$, target vector $\mathbf{t} \in \mathbb{Z}^m$
Output: vector $\mathbf{x} \in \mathcal{L}(\mathbf{B}')$, which satisfies $\|\mathbf{x} - \mathbf{t}\| \leq 2^{m/2} \text{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}'))$

- 1: Perform the LLL reduction on \mathbf{B}' with parameter $\delta = 3/4$.
- 2: Use the Gram–Schmidt orthogonalization on the reduced basis and obtain $\mathbf{B}^* = [\mathbf{b}^*_1, \mathbf{b}^*_2, \dots, \mathbf{b}^*_m]$.
- 3: $\mathbf{b} = \mathbf{t}$.
- 4: **for** $j = m, m - 1, \dots, 1$ **do**
- 5: $\mathbf{b} = \mathbf{b} - u_j \mathbf{b}^*_j$, where $u_j = \lceil \langle \mathbf{b}, \mathbf{b}^*_j \rangle / \langle \mathbf{b}^*_j, \mathbf{b}^*_j \rangle \rceil$;
- 6: **end for**
- 7: **return** $\mathbf{t} - \mathbf{b}$

First, apply Babai’s Nearest Plane algorithm to calculate the classical optimal solution, that is, the shortest distance vector $\mathbf{b}_{op} = (b_{op}^1, b_{op}^2, \dots, b_{op}^m)$. Then, the result is improved by QAOA. Let the LLL-reduced basis in Babai’s algorithm be $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m]$, and construct the optimization function

$$F(x_1, x_2, \dots, x_m) = \left\| \sum_{i=1}^m x_i \mathbf{d}_i - \mathbf{b}_{op} \right\|^2,$$

where $x_i \in \{-1, 0, 1\}, i = 1, 2, \dots, m$. It is easy to verify that $F(x_1, x_2, \dots, x_m)$ is a non-negative function. Let $\hat{x}_i = \frac{\sigma_{2i-1}^z + \sigma_{2i}^z}{2}$, which is a quantum operator encoded in the Pauli-Z basis. The eigenvalues of operator \hat{x}_i are $-1, 0, 1$, which exactly encodes the value of the variable x_i . Therefore, the corresponding problem Hamiltonian is

$$H_C = \sum_{j=1}^m \left| \sum_{i=1}^m d_{i,j} \hat{x}_i - b_{op}^j \right|^2.$$

Obviously, for an m -dimensional lattice, the number of qubits required to optimize Babai’s algorithm is $2m$.

To solve the problem, it is necessary to introduce a mixing Hamiltonian $H_M = \sum_{i=1}^{2m} \sigma_i^x$, where σ_i^x is the Pauli-X operator acting on the i th bit. The quantum circuit of QAOA is defined by the problem Hamiltonian H_C , the mixing Hamiltonian H_M and parameters (γ, β) . For D -layer QAOA circuits, there are usually $2D$ variational parameters. The process of using QAOA to solve the optimization problem is shown in Figure 1, and the algorithm description is shown in Algorithm 4.

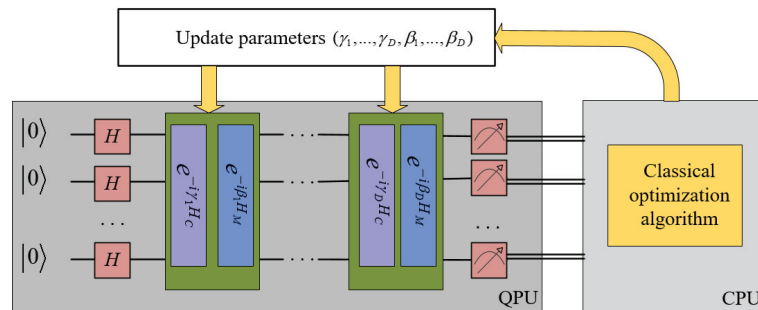


Figure 1. A schematic description of the VQE.

Algorithm 4 QAOA solving optimization.

Input: the problem Hamiltonian H_C , the mixing Hamiltonian H_M .

Output: the ground state $|\Psi_C\rangle$ of H_C .

- 1: Prepare the quantum register into $|\Psi_0\rangle = |+\rangle^{\otimes m}$.
 - 2: Choose the initial parameters γ, β . Perform H_C and H_M alternately and obtain $|\Psi(\gamma, \beta)\rangle$.
 - 3: Measure the quantum registers and calculate the cost function.
 - 4: Repeat Step 2 and Step 3 several times and calculate the expectation value of the cost function.
 - 5: Pass the expectation value and parameters (γ, β) to a classical optimizer. Update the parameters (γ, β) .
 - 6: Repeat Steps 2–5 until the result meets a fixed threshold and the parameters are updated to (γ^*, β^*) .
 - 7: **return** $|\Psi_C\rangle = |\Psi(\gamma^*, \beta^*)\rangle$
-

Now, we explain the steps in Algorithm 4. Step 1 performs $H^{\otimes m}$ on $|0\rangle^{\otimes m}$, and we obtain $|+\rangle^{\otimes m}$, which is an eigenvector of the Pauli-X operator.

Step 2 applies operators $e^{-i\gamma_k H_C}$ and $e^{-i\beta_k H_M}$, $k = 1, 2, \dots, D$, alternately. So, we generate a variational wave function

$$|\phi(\gamma, \beta)\rangle = e^{-i\gamma_D H_C} e^{-i\beta_D H_M} \dots e^{-i\gamma_1 H_C} e^{-i\beta_1 H_M} |+\rangle^{\otimes m}. \tag{1}$$

The wave function has $2D$ parameters $\{\gamma_1, \dots, \gamma_D, \beta_1, \dots, \beta_D\}$.

The expectation value means

$$\langle \Psi(\gamma, \beta) | H_C | \Psi(\gamma, \beta) \rangle, \tag{2}$$

which can be obtained by repeatedly preparing $|\Psi(\gamma, \beta)\rangle$ on the quantum processor and measuring it on a computational basis. Then, the classical computer performs classical optimization algorithms to find the optimal parameter. For example, the optimizers use the gradient descent algorithm to minimize the cost function in an iterative manner. The method calculates the first-order derivative of the function to compute the gradient. Then, it moves in the negative direction of the gradient. The termination condition of the gradient descent method is that the slope of the gradient is below a very small threshold. In the actual experiment, the algorithm is terminated by setting the empirical number of iterations.

In fact, classical optimization problems are often mapped to a simple Hamiltonian, which is diagonal in the computational basis. However, it does not mean that the problem is easy to solve or does not require a quantum solver. First, for example, MaxCut is a classical NP-hard problem, and the design of MaxCut problem Hamiltonian is $H = \sum_{ij} \frac{1}{2} (I - \sigma_i^z \sigma_j^z)$ [16]. In computational complexity theory, P is a set of relatively easy problems, and NP indicates hard problems. If MaxCut can be solved by classical computers easily, then $P = NP$, which completely overturns the theoretical basis of a range of fields. Second, processing classical optimization by QAOA usually requires a mixing Hamiltonian consisting of σ^x or σ^y , so quantum computers still work when solving classical optimization problems.

4. The Primal Method for Solving LWE

In this section, we propose a quantum primal method for solving LWE, where the Unique-SVP problem is solved by VQE. Although the quantum advantage of solving classical optimization by VQE is not as obvious as it is in quantum chemistry, understanding the evolution of the algorithm process is still crucial for improving algorithms running on classical hardware. We detail the number of qubits required and estimate the quantum resources when attacking the KYBER cryptosystem. With the development of quantum computers, resource estimation can also be used as a direction for comparison with pure classical algorithms.

4.1. LWE Algorithm

Algorithm 5 shows the procedure of the LWE algorithm.

Algorithm 5 The LWE algorithm.

Input: LWE samples $(\mathbf{A}, \mathbf{c} = \mathbf{A}\mathbf{s} + e) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$

Output: secret vector $\mathbf{s} \in \mathbb{Z}_q^n$

1: Construct a q-ary lattice $\Lambda_q(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}_q^m | \exists \mathbf{x} \in \mathbb{Z}^n, s.t. \mathbf{v} \equiv \mathbf{A}\mathbf{x} \text{ mod } q\}$, whose lattice basis is equivalent to $\mathbf{B} = [\mathbf{A} | q\mathbf{I}_m]^T \in \mathbb{Z}^{(m+n) \times m}$.

2: Perform elementary row transformations on \mathbf{B} and obtain the lattice basis

$$\mathbf{B}_1 = \begin{bmatrix} \mathbf{I}_n & \mathbf{A}'_{n \times (m-n)} \\ \mathbf{0} & q\mathbf{I}_{m-n} \end{bmatrix} \in \mathbb{Z}^{m \times m}.$$

3: Using Kannan's embedding technique, reduce BDD to Unique-SVP and obtain

$$\mathbf{B}_2 = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} \\ \mathbf{c} & M \end{bmatrix} \in \mathbb{Z}^{(m+1) \times (m+1)}.$$

4: Process \mathbf{B}_2 with VQE and derive a short vector \mathbf{e} .

5: **return** $\mathbf{s} = \mathbf{A}^{-1}(\mathbf{c} - \mathbf{e})$

Step 3 expands the q-ary basis by one dimension and embeds the target vector \mathbf{c} and the embedding factor M into matrix \mathbf{B}_2 . When $M = \|\mathbf{e}\|$, there exists $(\mathbf{e}, -M) \in \mathcal{L}(\mathbf{B}_2)$ [26]. In this case, proposing the first m bits of the vector recovers \mathbf{e} . In the experiment, we generally take $M = 1$.

Unique-SVP can be seen as a special case of SVP, and step 4 in Algorithm 5 solves SVP by VQE. The detailed description is shown in Algorithm 6.

Algorithm 6 VQE solving SVP.

Input: the lattice basis $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_m]^T \in \mathbb{Z}^{(m+1) \times (m+1)}$.

Output: short vector \mathbf{x} .

1: Perform BKZ-reduction on \mathbf{B} .

2: The SVP problem is encoded to the ground state of the Hamiltonian operator H .

3: Construct parameterized quantum circuits.

4: Repeat preparing an ansatz state $|\Psi(\theta)\rangle$ from the parameterized quantum circuit and measuring it in Pauli-Z basis. Calculate the expectation value $C(\theta)$.

5: Pass $C(\theta)$ and parameters to a classical optimizer. Update the parameter θ and go to step 4 until the expectation value converges.

The VQE procedure is visualized in Figure 2. Now, we explain the steps in Algorithm 6 in detail. In step 1, the larger the lattice size, the more quantum resources it occupies. In order to reduce the required qubits, a new basis matrix is first obtained by performing the BKZ reduction.

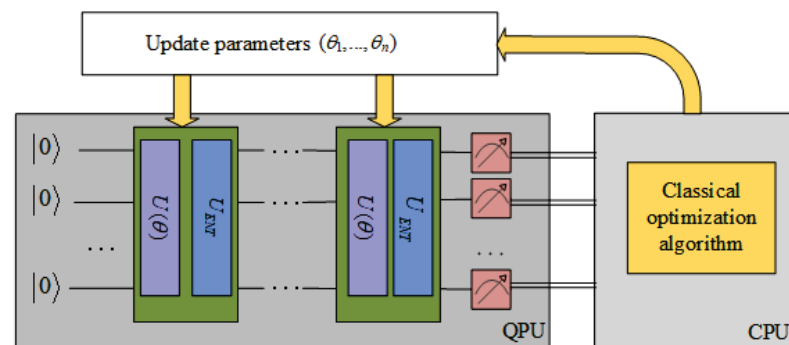


Figure 2. A schematic description of the VQE.

Step 2 constructs the problem Hamiltonian. For Lattice \mathbf{B} , SVP is to find a nonzero vector \mathbf{x} satisfying $\min_{\mathbf{x} \in \mathcal{L}(\mathbf{B})} \|\mathbf{x}\|$. Let the row vector of coefficients be \mathbf{z} and $\mathbf{z} \neq \mathbf{0}$; then, we have $\mathbf{x} = \mathbf{z}\mathbf{B}$. Let $\mathbf{G} = \mathbf{B}\mathbf{B}^T$; then, we have $\|\mathbf{x}\|^2 = \mathbf{z}\mathbf{B}\mathbf{B}^T\mathbf{z}^T = \mathbf{z}\mathbf{G}\mathbf{z}^T$. According to Algorithm 5, the dimension of the lattice is $m' = m + 1$. So, the SVP problem is equivalent to

$$\min_{\mathbf{x} \in \mathcal{L}(\mathbf{B})} \|\mathbf{x}\|^2 = \min_{\mathbf{z} \in \mathbb{Z}^{m'}} \left(\sum_{i=1}^{m'} z_i^2 \mathbf{G}_{ii} + 2 \sum_{0 \leq i < j \leq m'} z_i z_j \mathbf{G}_{ij} \right). \tag{3}$$

Before mapping the SVP problem into a Hamiltonian, we first introduce the method of reducing numbers in the integer interval $[-d, d]$ to a Boolean variable polynomial. Let $t = \lfloor \log d \rfloor$, introducing $t + 1$ Boolean variables $\beta_0, \beta_1, \beta_2, \dots, \beta_t$; the number in the interval can be expressed as $\sum_{i=0}^{t-1} 2^i \beta_i + (2d + 1 - 2^t) \beta_t - d$. Therefore, for the coefficient vector \mathbf{z} , if each entry satisfies $|z_i| \leq d_i, i = 1, 2, \dots, m'$, it can be expressed by Boolean variables $\beta_{i0}, \dots, \beta_{it_i}$. Substituting the Boolean variable polynomials into (3), we have

$$\min_{\beta_{10}, \dots, \beta_{1t_1}, \dots, \beta_{m'0}, \dots, \beta_{m't_{m'}}} \left(h + \sum_{ij} h_{ij} \beta_{ij}^2 + \sum_{ij \neq kl} l_{ij,kl} \beta_{ij} \beta_{kl} \right),$$

where $h, h_{ij}, l_{ij,kl}$ are calculated constants. Because β_{ij} are Boolean variables, the above equation is equivalent to

$$\min_{\beta_{10}, \dots, \beta_{1t_1}, \dots, \beta_{m'0}, \dots, \beta_{m't_{m'}}} \left(h + \sum_{ij} h_{ij} \beta_{ij} + \sum_{ij \neq kl} l_{ij,kl} \beta_{ij} \beta_{kl} \right). \tag{4}$$

In the above formula, it is required to find the parameter vector

$$\boldsymbol{\beta} = (\beta_{10}, \dots, \beta_{1t_1}, \dots, \beta_{m'0}, \dots, \beta_{m't_{m'}}) \tag{5}$$

to minimize the function

$$\sum_{ij} h_{ij} \beta_{ij} + \sum_{ij \neq kl} l_{ij,kl} \beta_{ij} \beta_{kl}.$$

Encoding the cost function into a Hamiltonian requires a mapping $\beta_{ij} \rightarrow (1 - \gamma_{ij})/2$, where $\gamma_{ij} \in \{-1, 1\}$. Then, substitute $\gamma_{ij} \rightarrow \sigma_{ij}^z$ and $1 \rightarrow I_{ij}$ to obtain the problem Hamiltonian

$$H = \sum_{ij} h_{ij} \frac{I_{ij} - \sigma_{ij}^z}{2} + \sum_{ij \neq kl} l_{ij,kl} \frac{I_{ij} - \sigma_{ij}^z}{2} \otimes \frac{I_{kl} - \sigma_{kl}^z}{2},$$

where $ij, kl \in \{10, \dots, 1t_1, m'0, \dots, m't_{m'}\}$ and σ_i^z is the Pauli-Z operator acting on the i th bit. The Hamiltonian acts on a Hilbert space spanned by $QNum$ qubits, and it can also be written as a sum over many local interactions.

To find the ground state of H , step 3 generates a hardware-efficient trial wavefunction, which is more suitable for available quantum devices [27]. Let $|\Psi(\boldsymbol{\theta})\rangle = (U(\boldsymbol{\theta})U_{ENT})^D |\Psi_0\rangle$ and the reference state is set to $|00\dots 0\rangle$. $U(\boldsymbol{\theta})$ are a group of single-qubit rotations determined by rotation angles $\boldsymbol{\theta}$. U_{ENT} are entangling drift operations generating sufficient entanglement. D defines the level of the quantum circuit. Obviously, with the increase of D , the convergence speed increases, but the fidelity decreases.

Step 4 calculates $C(\boldsymbol{\theta}) = \langle \Psi(\boldsymbol{\theta}) | H | \Psi(\boldsymbol{\theta}) \rangle$. Each iteration requires measuring N times and the cost obtained for the i -th time is C_i . Then, the expectation value is

$$C(\boldsymbol{\theta}) = \langle \Psi(\boldsymbol{\theta}) | H | \Psi(\boldsymbol{\theta}) \rangle = \frac{1}{N} \sum_{i=1}^N C_i. \tag{6}$$

If the Hilbert space is too large, because the interaction is local, the Hamiltonian can be split into a summation over many terms. The expectation calculations for one term are relatively simple, and we can speed up the computation by parallelizing the quantum

expectation-value estimation algorithm [28]. After calculating the expectation of each item on the quantum processor, multiply it by the weight and sum on the classical processor to obtain the final expectation value.

However, the shortest vector is $\mathbf{0}$ in this algorithm, so the restriction $\mathbf{x} \neq \mathbf{0}$ needs to be added. The idea is to increase C when appearing as $\mathbf{0}$. We assume that among the N measurements, there are N_0 results that are not $\mathbf{0}$, $C = \frac{1}{N_0} \sum_{i=1}^N C_i$. Obviously, the larger the N_0 , the smaller the C .

Step 5 uses the classical optimization algorithm to update θ until the expectation value converges and the process is similar to QAOA.

We give a toy example to illustrate the process on the quantum processor. For a more convenient description, the LWE dimension is further limited, and the example also supports simple experiments on the IBM quantum system. Let $q = 3, n = 1, m = 2$. The samples are $s + e_1 = 1 \pmod 3, 2s + e_2 = 2 \pmod 3$. The LLL-reduced matrix after Kannan’s embedding is

$$\begin{bmatrix} 0 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 2 & 0 \end{bmatrix}.$$

To simplify the model, suppose $z_i, i = 1, 2, 3$, are already Boolean variables. Then, the SVP problem can be reduced into finding the minimum value of $C = z_1 + 2z_2 + 5z_3 + 2z_2z_3$. The problem Hamiltonian is

$$H = 4.5I_1 \otimes I_2 \otimes I_3 - 0.5Z_1 \otimes I_2 \otimes I_3 - 1.5I_1 \otimes Z_2 \otimes I_3 - 3I_1 \otimes I_2 \otimes Z_3 + 0.5I_1 \otimes Z_2 \otimes Z_3. \tag{7}$$

Now, construct a hardware-efficient Ansatz consisting of several parameterized single-qubit rotation operations and controlled-NOT gates. Using the parameterized circuit shown in Figure 3, any 3-qubit quantum state $|\Psi(\theta)\rangle$ can be prepared, and different quantum states can be output by adjusting the six parameters.

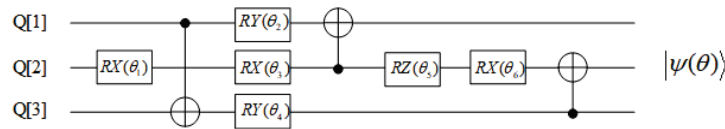


Figure 3. Quantum circuit for 3 qubits.

After preparing the ansatz state and measuring it repeatedly, we calculate the expectation value. Then, we perform the optimization process on the classical processor. Iterate the above process, and finally, the parameters corresponding to the optimal result are $(0, \pi, 0, 0, 0, \pi)$ and $[z_1, z_2, z_3] = [1, 0, 0]$. So, $[e_1, e_2] = [0, 0], s = 1$.

4.2. Algorithm Analysis

First, we analyze the range of d_i in the restriction condition $|z_i| < d_i, i = 1, 2, \dots, m'$. Let $\tilde{\mathbf{B}} = (\mathbf{B}^{-1})^T = [\tilde{\mathbf{b}}_1, \dots, \tilde{\mathbf{b}}_{m'}]^T$; then, there exists $\langle \mathbf{b}_i, \tilde{\mathbf{b}}_i \rangle = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$. Let the shortest vector $\mathbf{v} = \sum_{i=1}^{m'} t_i \mathbf{b}_i$; then, $|\langle \mathbf{v}, \tilde{\mathbf{b}}_i \rangle| = |t_i| \leq \|\mathbf{v}\| \|\tilde{\mathbf{b}}_i\|$. Due to the Gaussian heuristic, $\|\mathbf{v}\| = \sqrt{\frac{m'}{2\pi e}} \text{vol}(\mathcal{L})^{1/m'}$, we have $|t_i| \leq \sqrt{\frac{m'}{2\pi e}} \text{vol}(\mathcal{L})^{1/m'} \|\tilde{\mathbf{b}}_i\|$.

For an m' -dimensional matrix \mathbf{B} , its orthogonality defect $\delta(\mathbf{B}) = \frac{\prod_{i=1}^{m'} \|\mathbf{b}_i\|}{|\det(\mathbf{B})|}$. Obviously, for \mathbf{B} , there exists $\delta(\mathbf{B}) \geq 1$ and $\delta(\mathbf{B}) = 1$ if and only if \mathbf{B} is an orthogonal matrix. Therefore, the total number of qubits can be expressed as

$$QNum = \sum_{i=1}^{m'} (\lfloor \log d_i \rfloor + 1) \leq m' + \log(d_1 d_2 \dots d_{m'}), \tag{8}$$

where $\log(d_1 d_2 \dots d_{m'}) \leq 0.5m' \log(\frac{m'}{2\pi e}) + \log(\text{vol}(\mathcal{L}) \prod_{i=1}^{m'} \|\tilde{\mathbf{b}}_i\|) = 0.5m' \log(\frac{m'}{2\pi e}) + \log(\delta(\tilde{\mathbf{B}}))$. For a KZ-reduced matrix \mathbf{B} , its orthogonality defect satisfies [29]

$$\delta(\mathbf{B}) \leq (\frac{1}{8}m' + \frac{6}{5})^{m'/2} (\prod_{i=1}^{m'} \frac{\sqrt{i+3}}{2}) \leq (\frac{1}{8}m' + \frac{6}{5})^{m'/2} (m'+3)^{m'/2} (\frac{1}{2})^{m'}$$

So,

$$\log(\delta(\tilde{\mathbf{B}})) \leq \frac{m'}{2} \log(\frac{1}{8}m' + \frac{6}{5}) + \frac{m'}{2} \log(m'+3) - m' \leq m' \log(m'+3) - m'$$

Substituting into Equation (8), we have

$$\begin{aligned} QNum &\leq m' + (\frac{m'}{2} \log(m') - \frac{m'}{2} \log(2\pi e) + m' \log(m'+3) - m') \\ &= \frac{m'}{2} \log(m') - \frac{m'}{2} \log(2\pi e) + m' \log(m'+3) \end{aligned} \tag{9}$$

Therefore, the maximum number of qubits is $O(m' \log m')$. Now, we review the value of $d_i, i = 1, 2, \dots, m'$ when using VQE for enumeration. In practice, each z_i is represented by $QNum/m'$ qubits and the range of d_i is $[2^{(QNum/m')-1}, 2^{(QNum/m')} - 1]$, where $d_i \in \mathbb{Z}$.

In Kannan’s embedding, the lattice dimension is $m + 1$, where m is the sample number. In most cases, an LWE-based scheme produces only $m = poly(n)$ LWE samples (and the polynomial bound can be as small as $m = \Theta(n)$). In the LWE-based cryptosystem proposed in paper [12], $m = \sqrt{nlg(q)/lg(\delta)}$ and δ here means the root-Hermite factor. The theoretical worst-case reduction for LWE requires $\alpha q \geq 2\sqrt{n}$ [4], so we set $\alpha q = 2\sqrt{n}$. Now, we analyze the average number of qubits required, and its LWE parameters are shown in Table 1.

Table 1. LWE parameters.

	n	10	20	30	40
meirent	q	2053	2053	2053	2053
	αq	6.3246	8.9444	10.954	12.649
	m	34	65	91	127
	δ	1.069	1.0365	1.0280	1.0191

There are 4 groups of parameters in the table. For each group, 10 experiments are performed, and the average value of the cost function C is obtained. Finally, we calculate the average number of qubits required, and the result is illustrated in Figure 4. The four curves with different colors represent that the preprocessing method for the lattice basis is LLL, BKZ-20, BKZ-40 and BKZ-80, respectively. By the regression analysis, taking BKZ-20 as an example, we have

$$QNum = 92.54n \log n - 612.27n + 1343.8 \log n - 1234.37.$$

For example, for a 40-dimensional LWE problem, the maximum number of qubits required is 1126, which is a scale that is considered achievable in the near future. With the further development of quantum computers, LWE with larger dimensions can also be solved successively.

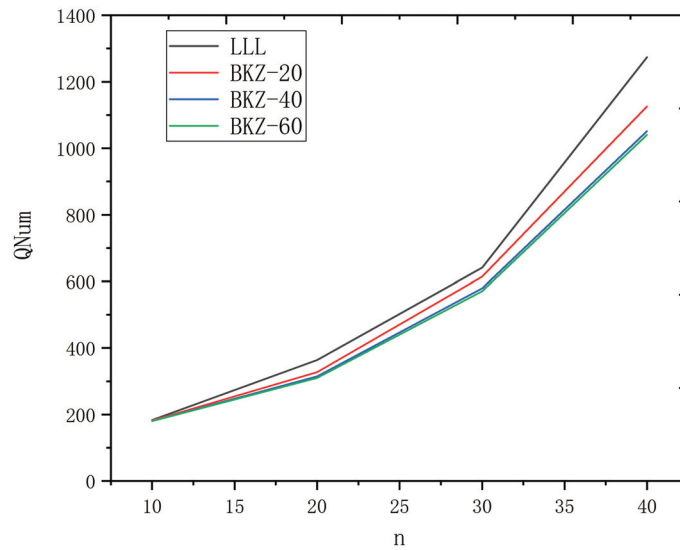


Figure 4. Average number of qubits required for different LWE dimensions.

4.3. Attacks on Existing Cryptosystems

In this section, we calculate the number of qubits required for a VQE attack on the KYBER cryptosystem. KYBER is a key encapsulation mechanism based on the module-LWE problem, which means it is based on Ring $R = \mathbb{Z}[X]/(X^{256} + 1)$. KYBER has three modes to satisfy 128/192/256-bit security, respectively. The parameters are listed in Table 2.

Table 2. KYBER parameters.

	<i>n</i>	<i>k</i>	<i>q</i>
KYBER512	256	2	3329
KYBER768	256	3	3329
KYBER1024	256	4	3329

In the table, *n, k, q* represents the maximum degree of polynomial, the number of polynomials in each vector and the modulus. The most famous attack on the MLWE problem does not utilize the special structure of a lattice, so we still analyze it as an LWE problem. Paper [7] mentioned that the number of samples is between 0 and $(k + 1)n$. To analyze the worst case, let $m = (k + 1)n$. Therefore, in the primal attack, the lattice dimension $d = m + 1 = (k + 1)n + 1$. Using the conclusion in Section 4.2, for the above three parameter settings, the required maximum qubits are 13,768, 19,538, and 25,482, respectively.

Although the quantum computers made at this stage are all NISQ devices, after IBM launched the 127-QubitEagle processor in 2021, it plans to launch the 1121-QubitCondor processor in 2023. At the same time, the IBM team also fully considered the future million-qubit system when designing the world’s largest dilution refrigerator “Goldeneye”, which is an important part of the IBM’s roadmap for scaling quantum technology.

5. Algorithm Implementation and Experimental Results

5.1. Using QAOA Algorithm to Improve the Decoding Method

In this section, we discuss the quantum advantage of the algorithm introduced in Section 3. Since it is difficult to estimate the computing complexity of QAOA, the QAOA process is regarded as a black box; that is, it is assumed that QAOA returns the solution to the optimization problem in a limited time. Now, without considering the actual complexity of QAOA, we only analyze the results of the algorithm through small-scale experiments.

The LWE instance is $(\mathbf{A}, \mathbf{c} = \mathbf{As} + \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$. Thus, after reducing to the BDD problem, the target vector is *c*. Algorithm 3 outputs a classical closest vector *w*, and the error vector can be obtained by $\mathbf{e} = \mathbf{c} - \mathbf{w}$. Then, Algorithm 4 updates vectors *w* and *e* by

QAOA. The result quality $r = \|\mathbf{e}\|$, which means the norm of the error vector. It is obvious that the smaller the r , the higher the quality.

Taking the dimension of the secret vector as $n = 3$ and $n = 5$, the experiment generates 50 groups of random LWE samples, respectively. Each group forms an LWE instance. For each instance, after obtaining the closest vector by Babai’s algorithm and calculating the result quality r , we use QAOA for optimization to obtain a new approximate closest vector and calculate the quality. Figure 5 shows the comparison of r between classical solutions and solutions after quantum optimization when $n = 3$, and Figure 6 illustrates the comparison when $n = 5$.

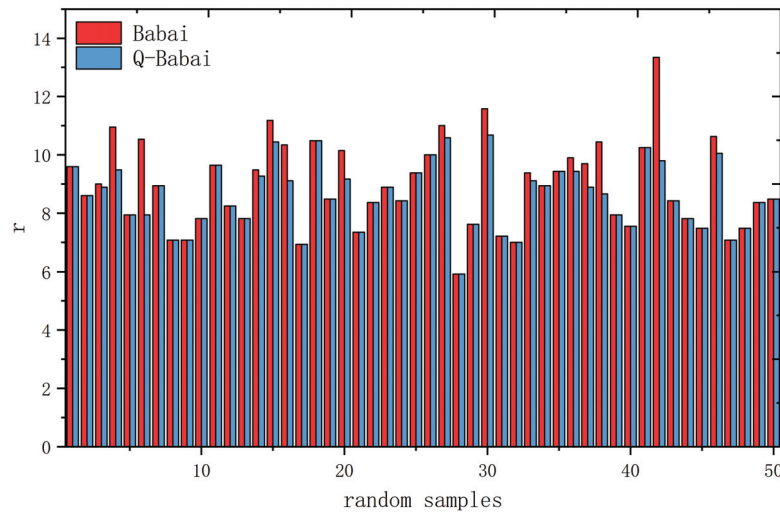


Figure 5. Quantum advantage demonstration of 50 random lattice samples when $n = 3$.

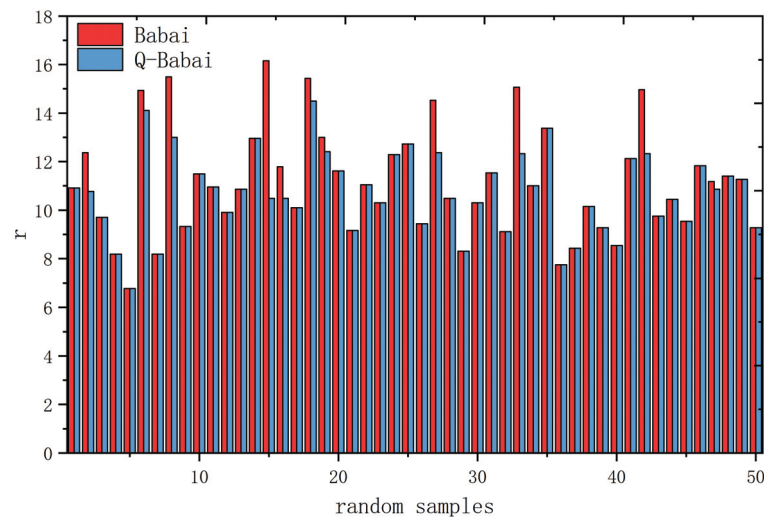


Figure 6. Quantum advantage demonstration of 50 random lattice samples when $n = 5$.

In Figures 5 and 6, the horizontal axis represents 50 groups of random samples, and the vertical axis represents the result quality r . The red columns represent the results of the classical Babai’s algorithm, and the blue columns represent the results after quantum optimization. According to the definition $r = \|\mathbf{e}\|$, a smaller r indicates a closer vector and higher quality. As evident in the figures, quantum results have higher quality than classical results in many cases, while in other cases, the results are the same. Therefore, the conclusion that can be drawn from the experiment is that quantum results obtained by QAOA are no worse than their classical counterparts.

5.2. Using VQE Algorithm to Realize the Primal Method

In this section, we present the experiments of solving LWE by the primal method. When quantum simulation is performed in a classical computer, the underlying quantum simulation uses QuSET [30], and the front-end interface to implement the algorithm uses C++. In the experiment, better results can be obtained by using the Conditional Value at Risk (CVaR) method [31]. Specifically, assuming C_1, C_2, \dots, C_n are sorted in non-decreasing order and in each loop, $C = \frac{1}{\lceil pN \rceil} \sum_{i=1}^{\lceil pN \rceil} C_i$, where $0 < p < 1$. Paper [21] proposes that $p = 0.175$ gives better results.

On the simulation platform, due to memory constraints, the maximum lattice dimension does not exceed 30, which means the LWE dimension n is much smaller than 30. If the input lattice matrix already contains the shortest vector, since the initial parameters of VQE are random and the algorithm still outputs the shortest vector after several iterations, it verifies the correctness of the algorithm. Therefore, when the VQE input is the reduced basis or the shortest vector can be obtained by simple vector addition or subtraction of the input matrix, the solution obtained by VQE is the same as that of the classical algorithm.

When the input is an arbitrary basis, the actual experimental results of the VQE are of poorer quality. The reason is that the simulation platform occupies classical memory, and the qubits for representing entries of the coefficient vector are limited. So, the correct coefficient vector cannot be accurately obtained. As the number of available qubits increases in the future, its coefficient representation will become more and more accurate, and the solution quality of the VQE algorithm will be better.

6. Discussion and Conclusions

VQA uses a classical optimizer to train parameterized quantum circuits, and it is one of the most promising quantum algorithms to achieve quantum supremacy. When researchers envision applications for quantum computers, it is almost impossible to bypass VQA algorithms. In this paper, we first present two LWE attacking tools, using QAOA to improve Babai's algorithm when solving BDD and utilizing VQE to solve Unique-SVP. The two algorithms combine classical optimization techniques and variational quantum techniques, providing ideas for solving LWE when the quantum resources are limited. Second, we estimate the number of qubits required for both algorithms. Third, for the two algorithms, experimental simulations are carried out, respectively. The experimental results show that for the first algorithm, QAOA improves the result quality of classical algorithms, and for the second algorithm, when the memory is large enough, the quality of quantum solutions is at least comparable to that of the classical solutions. How to further reduce the number of qubits by using the structure of the modular lattice is the direction that needs to be studied in the future.

Author Contributions: Formal analysis, L.L., B.Y. and H.W.; supervision, Z.M.; implement, Y.F. and X.M.; data analysis, Q.D.; writing—original draft preparation, L.L. and B.Y.; writing—review and editing, H.W. and Z.M.; funding acquisition, Z.M. All authors were involved in refining the ideas and writing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grants No. 61972413, 61901525, 62002385).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ajtai, M. Generating hard instances of lattice problems (extended abstract). In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC '96), New York, NY, USA, 22–24 May 1996; pp. 99–108.
2. Ajtai, M.; Dwork, C. A public-key cryptosystem with worst-case/average-case equivalence. In Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing (STOC '97), New York, NY, USA, 4–6 May 1997; pp. 284–293.
3. Hoffstein, J.; Pipher, J.; Silverman, J.H. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory*; Buhler, J.P., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1423.
4. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing (STOC '05), New York, NY, USA, 22–24 May 2005; pp. 84–93.
5. Applebaum, B.; Cash, D.; Peikert, C.; Sahai, A. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In *Advances in Cryptology-CRYPTO 2009*; Halevi, S., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5677.
6. Brakerski, Z.; Vaikuntanathan, V. Efficient Fully Homomorphic Encryption from (Standard) LWE. In Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA, USA, 22–25 October 2011.
7. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-KYBER: Algorithm Specifications and Supporting Documentation. 2021. Available online: <https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf> (accessed on 15 February 2022).
8. Bai, S.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium: Algorithm Specifications and Supporting Documentation (Version 3.1). Available online: <https://pq-crystals.org/dilithium/data/dilithiumspecification-round3-20210208.pdf> (accessed on 30 January 2022).
9. Blum, A.; Kalai, A.; Wasserman, H. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM* **2003**, *50*, 506–519. [CrossRef]
10. Arora, S.; Ge, R. New Algorithms for Learning in Presence of Errors. In *Automata, Languages and Programming. ICALP 2011*; Aceto, L., Henzinger, M., Sgall, J., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6755.
11. Babai, L. On Lovász' lattice reduction and the nearest lattice point problem. In *STACS 1985*; Mehlhorn, K., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1984; Volume 182.
12. Lindner, R.; Peikert, C. Better Key Sizes (and Attacks) for LWE-Based Encryption. In *Topics in Cryptology—CT-RSA 2011*; Kiayias, A., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6558.
13. Albrecht, M.R.; Fitzpatrick, R.; Göpfert, F. On the Efficacy of Solving LWE by Reduction to Unique-SVP. In *Information Security and Cryptology—ICISC 2013*; Lee, H.S., Han, D.G., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8565, pp. 293–310.
14. Kannan, R. Minkowski's Convex Body Theorem and Integer Programming. *Math. Oper. Res.* **1987**, *12*, 415–440. [CrossRef]
15. Bai, S.; Galbraith, S.D. An Improved Compression Technique for Signatures Based on Learning with Errors. In *Topics in Cryptology—CT-RSA 2014*; Benaloh, J., Ed.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014; Volume 8366.
16. Farhi, E.; Goldstone, J.; Gutmann, S. A quantum approximate optimization algorithm. *arXiv* **2014**, arXiv:1411.4028.
17. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.H.; Zhou, X.Q.; Love, P.J.; Aspuru-Guzik, A.; O'Brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213. [CrossRef]
18. Wei, S.; Li, H.; Long, G. A Full Quantum Eigensolver for Quantum Chemistry Simulations. *Research* **2020**, *2020*, 1486935. [CrossRef] [PubMed]
19. Joseph, D.; Callison, A.; Ling, C.; Mintert, F. Two quantum Ising algorithms for the shortest-vector problem. *Phy. Rev. A* **2021**, *103*, 032433. [CrossRef]
20. Joseph, D.; Ghionis, A.; Ling, C.; Mintert, F. Not-so-adiabatic quantum computation for the shortest vector problem. *Phys. Rev. Res.* **2020**, *2*, 013361. [CrossRef]
21. Albrecht, M.R.; Prokop, M.; Shen, Y.; Wallden, P. Variational quantum solutions to the Shortest Vector Problem. *IACR Cryptol. ePrint Arch.* **2022**, *2022*, 233.
22. Lenstra, A.K.; Lenstra, H.W.; Lovász, L. Factoring polynomials with rational coefficients. *Math. Ann.* **1982**, *261*, 515–534. [CrossRef]
23. Schnorr, C.-P.; Euchner, M. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.* **1994**, *66*, 181–199. [CrossRef]
24. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
25. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J.S.; Menke, T.; et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004. [CrossRef]
26. Lyubashevsky, V.; Micciancio, D. On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In *Advances in Cryptology-CRYPTO 2009*; Halevi, S., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; pp. 577–594.
27. Nikolaj, M.; Panagiotis, B.; Bishop, L.S.; Chow, J.M.; Cross, A.; Egger, D.J.; Filipp, S.; Fuhrer, A.; Gambetta, J.M.; Ganzhorn, M. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Sci. Technol.* **2018**, *3*, 030503.

28. McClean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **2016**, *18*, 023023. [CrossRef]
29. Wen, J.; Chang, X.-W. On the KZ Reduction. In Proceedings of the IEEE International Symposium on Information Theory (ISIT 2015), Hong Kong, China, 14–19 June 2015; Volume 65, pp. 1921–1935
30. Jones, T.; Brown, A.; Bush, I.; Benjamin, S.C. QuEST and High Performance Simulation of Quantum Computers. *Sci. Rep.* **2019**, *9*, 10736. [CrossRef]
31. Barkoutsos, P.K.; Nannicini, G.; Robert, A.; Tavernelli, I.; Woerner, S. Improving Variational Quantum Optimization using CVaR. *Quantum* **2020**, *4*, 256. [CrossRef]

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
www.mdpi.com

Entropy Editorial Office
E-mail: entropy@mdpi.com
www.mdpi.com/journal/entropy



Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.



Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-0019-3