



# Article A NISQ Method to Simulate Hermitian Matrix Evolution

Keren Li<sup>1,\*</sup> and Pan Gao<sup>2,\*</sup>

- <sup>1</sup> Peng Cheng Laboratory, Shenzhen 518055, China
- <sup>2</sup> Beijing Academy of Quantum Information Sciences, Beijing 100193, China
- <sup>+</sup> Correspondence: likr@pcl.ac.cn (K.L.); gaopan@baqis.ac.cn (F.L.)

**Abstract:** As a universal quantum computer requires millions of error-corrected qubits, one of the current goals is to exploit the power of noisy intermediate-scale quantum (NISQ) devices. Based on a NISQ module–layered circuit, we propose a heuristic protocol to simulate Hermitian matrix evolution, which is widely applied as the core for many quantum algorithms. The two embedded methods, with their own advantages, only require shallow circuits and basic quantum gates. Capable to being deployed in near future quantum devices, we hope it provides an experiment-friendly way, contributing to the exploitation of power of current devices.

Keywords: matrix evolution; NISQ; quantum algorithm

## 1. Introduction

Building up a large-scale error-corrected quantum computer is to be one of the greatest scientific and engineering achievements [1–4]. However, stringent requirements such as millions of qubits with high accuracy are far to meet. Preskill coined "Noisy Intermediate-Scale Quantum" (NISQ) to describe this era, where NISQ devices represent the current state of the art in the fabrication of quantum devices [5]. The leading quantum computers contain up to a few hundred physical qubits, but provide rare practical applications as error correction is missing [6,7]. Therefore, while polishing the hardware-related techniques, one present-day goal is to exploit the power of current machines.

Matrix evolution is computationally hard in numerical mathematics, as  $O(d^3)$  operations are required for an unstructured  $d \times d$  matrix [8]. With advent of quantum algorithms, this can be solved to some extent by instinct priority of fault-tolerant quantum computation on matrix multiplications. For example, complexity of *t*-time analog Hamiltonian evolution is O(t). Furthermore, certain digital algorithms were proposed for hermitian matrices' evolution, which produce quantum speedups in many scenarios, such as simulation algorithms, quantum principal component analysis, quantum matrix inversion, and their generalizations [9–15]. However, deep quantum circuits and inaccessible oracles are required, which hinder their applications on near term devices. Accordingly, one question need be addressed: How to realize matrix evolution on current NISQ devices?

NISQ algorithms are a class of algorithms with no explicit requirements for error correction, promising to be deployed on NISQ hardware [5,16]. In regard to matrix evolution, we explore its near future application by introducing one typical NISQ module–layered circuit. Consequently, in Section 2, a heuristic protocol is proposed to employ a layered circuit to simulate hermitian matrix evolution, which is experimental friendly and can be employed in near-term applications of algorithms. To generate proper layered circuits, two methods are embedded in this protocol. The first is inspired by optimal control theory, which finds the simulating circuit directly, but with no scalability. The second, averting the scalability problem, generates the simulating circuit by a hybrid quantum-classical paradigm [17]. Both simulating circuits are with basic quantum gates and a pre-set depth, with the consumption of generating those circuits analyzed in Section 3. To support feasibility, simulating circuits are generated and validated numerically in Section 4, where



Citation: Li, K.; Gao, P. A NISQ Method to Simulate Hermitian Matrix Evolution. *Entropy* **2022**, 24, 899. https://doi.org/10.3390/ e24070899

Academic Editor: Fabio Benatti

Received: 23 May 2022 Accepted: 27 June 2022 Published: 29 June 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/). hermitian matrices are set as density matrix of Bell state, GHZ state, and Hamiltonian of Crotonic acid molecular [18]. Compared with the method such as product formula and density matrix evolution, simulating circuits by our methods are with shallower depth, and more friendly to the experimental realizations. Furthermore, as a generalization of linear combination unitaries and layered circuit [19–21], ancillary layered circuit is proposed in our protocol, which not only serves as a quantum compiler here, but also an essential subroutine for NISQ algorithms.

#### 2. Result

For hermitian matrices evolution, Lie–Trotter products and density matrix evolution provide solutions. We briefly review them here.

In the first case, for general Hamiltonian  $\rho = \sum_i \rho_i$  with  $\rho_i$  be local interactions,  $e^{-i\rho t}$  can be simulated by the lowest order Lie–Trotter product formula [22],

$$(\prod_{i} e^{-i\rho_i \Delta t})^n \to e^{-i\rho t} + \mathcal{O}(||\rho||^2 t^2/n),$$
(1)

with  $\Delta t = t/n$ , as shown in the schematic process in Figure 1a.  $n = O(t^2 e^{-1} ||\rho||^2)$  denotes the repetition times, and therefore the circuits size and depth for full  $e^{-i\rho t}$  with desired accuracy  $\epsilon$ . Though repeated applications of simulated circuit is feasible in theory, a low-depth quantum circuit is preferred by current NISQ devices as the limited coherence time. Therefore, under the circumstance of a tolerant accuracy, the implementation appears unfriendly to current or near term devices.

In the second case,  $\rho$  is not only Hermitian but also positive semi-definite and unit trace, that is,  $\rho$  is a physical quantum state,  $e^{-i\rho t}$  can be realized by multiple copies of  $\rho$  and infinitesimal swap operations [10]. Assuming that  $\sigma$  is another quantum state that  $e^{-i\rho t}$  act on, the infinitesimal swap operation has such effect,

$$\operatorname{tr}_{p}[e^{-iS\Delta t}\rho\otimes\sigma e^{iS\Delta t}] = \sigma - i\Delta t[\rho,\sigma] + \mathcal{O}(\Delta t^{2}), \tag{2}$$

where  $tr_p$  is the partial trace over  $\rho$  and *S* is the swap operator. Shown in Figure 1b, density matrix evolution with respect to  $\sigma$  can be constructed by repeated applications of (2) with  $n = t/\Delta t$  copies of  $\rho$ . Therefore, if the swap *S* and its infinitesimal exponential operation  $e^{-iS\Delta t}$  can be implemented in a single layer circuit, both size and depth of using Equation (2) are  $O(t^2\epsilon^{-1})$ . For accurate simulation, this depth and size of the circuit still dissatisfy the characteristics of current devices, and thus hinder its near term application.



**Figure 1.** Conventional method to realize  $e^{-i\rho t}$ , where  $\rho$  is a hermitian matrix. (a) is via decomposition from Lie–Trotter products and (b) is the density matrix evolution using infinitesimal swaps  $e^{-iS\Delta t}$ . Wherein, / denotes a register of multi qubits, and | means tracing out the corresponding register.

One typical NISQ module is the layered circuit, which concretely implement near term applications [23,24]. Specifically, a *m* layered circuit  $U(\theta) = \prod_{i=1}^{m} U_i(\theta_i)$  is presented in Figure 2a, which is parameterized by  $\theta$ . As NISQ devices are with characteristics such as, limited size, short coherence time, and basic quantum operations, a shallow layered circuit

seems perfect to undertake a NISQ applications. Remarkably, although shallow circuits and basic operations are utilized, the expressivity of layered circuits can be nontrivial and has been investigated in some recent papers [25,26]. Therefore, in our work, a layered circuit is employed to approach the simulation of matrix evolution, mathematically,

$$U_m(\theta_m)...U_i(\theta_1) \to e^{-it\rho},\tag{3}$$

where  $\rho$  is target hermitian matrix.

A layered circuit can be fully determined by its structure and parameters. For most applications, structure is configured previously, which depends on the tasks at hand. The widely used structures include quantum alternating operator ansatz, variational Hamiltonian ansatz and unitary coupled clustered ansatz [27–30]. In this study, according to hardware efficiency, we employed an m = O(n) layered circuit for following *n*-qubit tasks. All two body interactions are involved and gates in one layer are commuted with each others, which aims at employing the two-body interactions of the devices with a limited depth circuit. This circuit is problem-agnostic, which is given in Appendix B For specific problem faced, the circuit should be re-designed, which exploits both expressibility and trainability.

Furthermore, before stepping into parameter determination, ancillary layered circuit is introduced as preliminary. It is schematically presented in Figure 2b. Besides the principal system which is denoted as p, the ancillary register a is added with the same size as Chio-Jamiolkowski isomorphism is employed. The entire circuit includes three parts: encoding circuit ( $U_e$ , colored blue), layered circuit part ( $U_l \otimes I$ , colored pink), and decoding circuit ( $U_d$ , colored green). The details of dynamics in Figure 2b are shown as follows.

For initialization, two registers are jointly prepared on  $\Omega_0 = |\psi_0\rangle \langle \psi_0|$ , where

$$\left|\psi_{0}\right\rangle = \left|0\right\rangle_{p}\left|0\right\rangle_{a}.\tag{4}$$

Encoding circuit  $U_e$  is supposed to evolve the system into  $\Omega_1 = |\psi_1\rangle \langle \psi_1|$ , where  $|\psi_1\rangle = \sum_i |i\rangle_p \otimes |i\rangle_a$  are pairs of Bell states. This step can be realized by a bunch of control-*z* gates,  $C_z$  and Hadamard gates,  $H_a$ ,

$$\Omega_1 = U_e \Omega_0 U_e^{\dagger}, 
U_e = I_p \otimes H_a \cdot C_z \cdot H_p \otimes H_a.$$
(5)

Then,  $U_l$  is applied on p, driving the system into  $\Omega_2$ , i.e., the Choi matrix of  $U_l$ ,

$$\Omega_2 = |\psi_2\rangle \langle \psi_2| = \sum_{i,j} U_l(|i\rangle \langle j|)_p U_l^{\dagger} \otimes |i\rangle \langle j|_a \,. \tag{6}$$

Noted that, arbitrary state  $\rho$  with the application of  $U_l$  has a state-channel duality,

$$U_l(\rho) = \operatorname{tr}_a[\Omega_2 \cdot I \otimes \rho^*]. \tag{7}$$

Thus, knowing  $\Omega_2$  is sufficient to completely determine  $U_l$ , i.e., converting quantum channel characterization to state characterization. Choi-Jamiolkowski isomorphism is exactly the correspondences in Equations (6) and (7).

Finally,  $\Omega_3$  is the result generated by steering  $\Omega_2$  through the decoding circuit  $U_d = U_e^{\dagger}$ , which disentangles the system and is essential in our methods.

For parameter determination, it can be solved by two optimization methods, which minimize the distance,

$$\|U_m(\theta_m)...U_i(\theta_1) - e^{-it\rho}\|,\tag{8}$$

where  $\rho$  is the target hermitian matrix. Specifically, we substitute  $U_i(\theta_i)$  with  $e^{-it\theta_i\rho_i}$ , where  $\rho_i$  are tensor products of Pauli operators. In this configuration, the problem states as approaching  $e^{-it\rho}$  by

$$e^{-it\theta_m\rho_m}\cdots e^{-it\theta_1\rho_1}.$$
(9)

The formalism of  $\rho_i$  determines the structure of layered circuit. It is an empirical task and is set in advance, which we have depicted before.



**Figure 2.** Layered circuit and ancillary layered circuit. (**a**) gives an example of an *m* layered circuit, which is employed in (**b**) as  $U_l(\theta)$ .  $U_i(\theta_i)$  are parametrized by tuneable  $\theta_i$ . (**b**) is ancillary layered circuit, which has three parts: encoding circuit (blue)  $U_e$ , layered circuit (pink)  $U_l \otimes I$  and decoding circuit (green)  $U_d$ .  $\Omega_0$ ,  $\Omega_1$ ,  $\Omega_2$ , and  $\Omega_3$  represent temporary states.

From the point of view of state, for a specified time *t*,  $\rho$  drives an ideal arbitrary system, which is labelled as  $\sigma$ , to

$$\sigma_0(t) = e^{-it\rho} \sigma e^{it\rho}.$$
(10)

Simultaneously, with the application of Equation (9), the real system is steered to  $\sigma(t)$ , where

$$\sigma(t) = \prod_{i=1}^{m} e^{-it\theta_i \rho_i} \sigma \prod_{i=1}^{m} e^{it\theta_i \rho_i}.$$
(11)

The objective function f is thus defined as the overlap which is measured by the standard inner product

$$f(\boldsymbol{\theta}) = \operatorname{tr}(\sigma_0(t)\sigma(t)) = \operatorname{tr}(\mu_k \cdot \nu_k), \tag{12}$$

where

$$\mu_{k} = \prod_{i=k+1}^{m} e^{it\theta_{i}\rho_{i}} e^{-it\rho} \sigma e^{it\rho} \prod_{i=m}^{k+1} e^{-it\theta_{i}\rho_{i}},$$
  

$$\nu_{k} = \prod_{i=k}^{1} e^{-it\theta_{i}\rho_{i}} \sigma \prod_{i=1}^{k} e^{it\theta_{i}\rho_{i}}.$$
(13)

Obviously,  $f(\theta) \leq 1$ , the maximum situation is satisfied when the outputs of  $\prod_{i=1}^{m} e^{-it\theta_i\rho_i}$ and  $e^{-it\rho}$  are the same. This is a state to state situation. To approach the dynamics, Choi matrix is employed, which can be produced by ancillary layered circuit. In this situation, the notations are redefined,

$$\sigma = U_e \Omega_0 U_e^{\dagger},$$
  

$$\mu_k = \prod_{i=k+1}^m (e^{it\theta_i \rho_i} \cdot e^{-it\rho}) \otimes I \cdot \sigma \cdot \prod_{i=m}^{k+1} (e^{it\rho} \cdot e^{-it\theta_i \rho_i}) \otimes I,$$
  

$$\nu_k = \prod_{i=k}^1 e^{-it\theta_i \rho_i} \otimes I \cdot \sigma \cdot \prod_{i=1}^k e^{it\theta_i \rho_i} \otimes I.$$
(14)

*Method 1* is a traditional method of gradient, where the partial derivative with respect to the parameter  $\theta_k$  are

$$\frac{\partial f}{\partial \theta_k} = -it \langle \mu_k | [\rho_k, \nu_k] \rangle.$$
(15)

 $\langle \cdot \rangle$  means trace here. Therefore, by updating  $\theta_k$  by  $\theta'_k$  with a learning rate  $\eta$ ,

$$\theta_k' = \theta_k + \eta \frac{\partial f}{\partial \theta_k},\tag{16}$$

the objective function would increase along the direction of gradient until converging into the local maximum. If  $f \approx 1$  is iteratively achieved, this layered circuit is said to approach the application of  $e^{-it\rho}$ .

However, one vital problem is at calculating the gradient. As obtaining  $v_k$  and  $\mu_k$  is in general inefficient for tremendous time-consuming. Therefore, *Method* 2 is reported.

*Method* 2 is outlined as follows, (i) implementing  $e^{i\rho\Delta t}$  with a layered circuit, where  $\Delta t$  is a sufficiently short period of time; (ii) implementing  $e^{i\rho t}$  with (i) as the starting point step by step, where ancillary layered circuit is used as a compiler and  $\mathcal{O}(\log(t/\Delta t))$  steps are required. This idea is extended from trotter decomposition, which induces a trotter error with  $\epsilon_t \sim \mathcal{O}(t\Delta t)$ . We illustrate the details as follows.

The first step is to realize an approximation,

$$\prod_{i} e^{-i\Delta t\theta_{i}\rho_{i}} \to e^{-i\Delta t\rho},\tag{17}$$

which requires us realizing following equation if only lowest order is considered,

$$\sum \theta_i \rho_i = \rho. \tag{18}$$

This is coincident with Lie–Trotter decomposition, where sufficiently small  $\Delta t$  is adopted. Related details for derivation can be found in Appendix A. Accordingly, assuming that  $\Delta t$  is small enough,  $e^{i\rho\Delta t}$  can be simulated efficiently by a layered circuit, where structure and parameters are given by Equation (18).

The second step is to simulate  $e^{i\rho t}$ . On basis of (i), this can be achieved by repeatedly applying ancillary layered circuit as compiler for  $\mathcal{O}(\log(t/\Delta t))$  times. We depict the *i*-th iteration in Figure 3, where  $U_l$  consists of  $U_{i+1}(\theta)$  and  $U_i^{-1}$  followed by  $n_c$  times. The ancillary layered circuit serves as a compiler, learning repeated  $U_i$  by  $U_{i+1}(\theta)$ .  $n_c$  represents the compression efficiency,  $U_i$  is known, and the structure of  $U_{i+1}$  is set previously. Specifically, the task for *i*-th iteration is to generate  $U_{i+1}(\theta)$  with an explicitly prior known  $U_i$ , where  $U_{i+1} \leftarrow U_i^{n_c}$  and  $\theta$  are the parameters to be optimized.

The objective function is defined as,

$$f = tr(\Omega_1 \Omega_2) = tr(\Omega_0 \Omega_3), \tag{19}$$

which reaches the maximum at  $U_{i+1}(\theta) = U_i^{n_c}$ . As  $\Omega_1$  are pairs of Bell states, this objective function cannot be measured with the local bases directly and efficiently. We investigate the second term in Equation (19): A decoding circuit  $U_d$  disentangles the target component into  $\Omega_3$ . In this situation, only with the local measurement on  $\Omega_0$ , the objective function can be estimated and optimized as conventional hybrid quantum algorithms [31]. Iteratively, parameters in  $U_{i+1}$ , which are obtained in the *i*-th iteration, could be applied in the (i + 1)-th step, where  $U_{i+2} = U_{i+1}^{n_c}$  is realized.

Accordingly, by  $\log(t/\delta)$  steps,  $e^{it\rho}$  can be implemented based on  $e^{i\Delta t\rho}$ . Compared to repeatedly applications of  $U_i$  which will cause the circuit depth affordable, this strategy trades the circuit depth with repeated applications of ancillary layered circuit(shown in Figure 3), which increases the training time while compresses the repeated circuit depth.



**Figure 3.** Ancillary layered circuit to implement *method* 2.  $U_{i+1}(\theta)$  is the circuit to be optimized with known  $U_i$ . Notations are the same as Figure 2a.  $\Omega_0$ ,  $\Omega_1$ ,  $\Omega_2$ , and  $\Omega_3$  are temporary states.

# 3. Analysis

To simulate hermitian matrix evolution with layered circuit, time complexity is bounded by the depth of used circuit, which is easily analyzed if circuits are determined. For errors introduced by simulation, it is the distance between target evolution and layered circuit, which is also generated by two embedded methods. Therefore, analysis for embedded methods are important: Learning complexities and learning errors are analyzed in this section.

We list procedures of two methods in the *Method 1* and *Method 2* and analyze the learning complexity first.

*Method* 1 relies on a classical optimization to generate parameter configurations. Therefore, complexities for time and memory is based on classical resources, such as classical logic gates and registers. During the procedure, approximating the gradient by Equation (15) costs the most. For each iteration, calculating  $\mu_k$  and  $\nu_k$  lies at the heart, which requires the implementation of matrix multiplications on a classical computer. Though  $e^{-it\rho_i}$  can be efficiently simulated individually for locality of  $\rho_i$ , simulating  $e^{-it\rho}$  is in general hard for most cases and there is no even universal efficient algorithms. To simulate an *n*-qubit problem,  $2^n \times 2^n$  matrix should be generated and stored, with  $\mathcal{O}(4^n)$  operations for evaluating the objective function and obtaining the gradient. Thus, the time complexity is  $\mathcal{O}(4^n) \times r$  for a total *r* iterations while memory complexity is  $\mathcal{O}(4^n)$ . As Choi states are employed, our expedition will be enlarged, qubits are doubled.

Accordingly, this method is in general inefficient. However, it would be applicable to specific circumstances where  $\nu_k$  and  $\mu_k$  can be efficiently simulated. In fact, some investigations have employed tensor network to work out certain structure simulation problem and shed the light on the middle-scale quantum systems [32].

*Method* 2 is a hybrid quantum classical paradigm, where ancillary layered circuits are executed on quantum computers, parameters are updated on classical computers. For tasks on quantum computers, ancillary layered circuit brings a 2*n*-qubit consumption on quantum register, where *n* is the size of simulation quantum system. Time complexity to execute ancillary layered circuit is additional O(n) Hadamard and control-z gates, with a depth of O(m) layered circuit consisting of at most  $O(m \times n)$  basic quantum gates. Luckily, our measurement is on  $\Omega_0$ , which is local measurement. Therefore, for single iteration, updating all parameters requires  $O(m \times n)$  times repeatedly applying and measure ancillary layered circuit. For *r* repetitions,  $O(m \times n) \times r$  is required. If  $e^{-it\rho}$  is to be realized by  $e^{-i\Delta t\rho}$ , an additional multiplied factor log  $(t/\Delta t)$  should be added. For tasks on classical computers, a storage of  $O(m \times n)$  parameters and their numerical gradient is required. Time complexity depends only on operations of fundamental arithmetic, which is within the reach of current machines.

Accordingly, time complexity of *Method* 2 is  $\mathcal{O}(m \times n \times r \times \log(t/\Delta t))$ . As in our configuration,  $m = \mathcal{O}(n)$ , which leads our method acceptable with respect to efficiency.

## Method 1

**Input:** layered circuit  $\prod_{i=1} e^{-it\theta_i\rho_i}$ .  $\theta$  is to be optimized,  $\epsilon_o$  is an optimized threshold, and  $\delta_1$  is the tolerance for improving.

**Output:**  $\theta$ , parameter configuration, which is optimized to approach  $e^{-it\rho}$  via  $\prod_{i=1} e^{-it\theta_i\rho_i}$ .

1: Evaluate the objective function Equation (12) with existed  $\theta$ , denoted as f. If  $f \le 1 - \epsilon_0$ , go to 2, otherwise, the algorithm terminates and  $\theta$  return.

2: for k = 1, ..., m do

- 3: Based on Equation (14),  $v_k$  and  $\mu_k$  are calculated.
- 4: Evaluate  $\partial f / \partial \theta_k$  by Equation (15).

```
5: end for
```

```
6: Update m-element \theta according to Equation (16)
```

- 7: Evaluate the objective function with the new  $\theta$
- 8: if  $f \ge 1 \epsilon_o$  then
- 9: the algorithm terminates and  $\theta$  return;
- 10: else if  $\Delta f \leq \delta_1$  then
- 11:  $\theta$  are re-initialized and go to 1.
- 12: else
- 13: go to 2.
- 14: end if
- 15:
- 16:
- 17: 18:

# Method 2

**Input:** layered circuit  $\prod_{i=1} e^{-i\theta_i\rho_i}$  with known  $U_1$ .  $\theta$  is to be optimized,  $\epsilon_o$  is the optimized threshold,  $\delta_2$  is the tolerance for improving, and  $n_c$  is the compressing factor.

**Output:**  $\theta$ , parameter configuration, which is optimized to approach  $e^{-it\rho}$  via  $\prod_{i=1} e^{-it\theta_i\rho_i}$ 

1: **for**  $i = 1, ..., \log(t/\Delta t)$  **do** 

```
2: if i = 1 then
```

- 3:  $U_1$  and its inversion are generated by the input.
- 4: **else**
- 5:  $U_i$  are generated as the output of the last iteration.

6: end if

- 7: Evaluate the objective function in Equation (19) with existed  $\theta$  and  $n_c$ , which is denoted as *f*. If  $f \le 1 \epsilon_o$ , go to 8, otherwise,  $\theta$  for  $U_{i+1}$  return and go to 1.
- 8:  $\theta$  for  $U_{i+1}$  are optimized as variant quantum algorithms,  $\theta$  are updated and f is evaluated.

```
9: if f \ge 1 - \epsilon_o then
```

10:  $\theta$  return and go to 1;

- 11: else if  $\Delta f \leq \delta_2$  then
- 12:  $\theta$  are re-initialized and go to 7.
- 13: **else**
- 14: go to 8.
- 15: end if

16: end for

To analyze errors of layered circuit by both methods, first, we define some notations:  $\epsilon_0$  is the optimized threshold which is supposed to terminate the training process;  $\epsilon_t$  is the deviations coming from lie-product decomposition.

If *Method* 1 is completed, the optimized threshold is targeted. The obtained layered circuit has an accuracy of  $1 - \epsilon_o$  as training process permits the error no more than  $\epsilon_o$ . For

*Method 2*, the error accumulates according to a chain rule when implementing  $U_{i+1}$  by  $U_i$ . It ends up with an error of  $\mathcal{O}(n_c^s \epsilon_o)$ , where the higher order terms are ignored and *s* is steps, which is in total of  $\mathcal{O}(\log(t/\Delta t))$  to realizing  $e^{-i\rho t}$  from  $e^{-i\rho\Delta t}$ . Thus, the discrepancy is  $\mathcal{O}((t/\Delta t)\epsilon_o)$ . Additionally, an error coming from Equation (1) is also considered as  $\epsilon_t$ , which is of  $\mathcal{O}(t\Delta t)$ . Accordingly, the total error for *Method 2* is  $\mathcal{O}((t^2/\epsilon_t)\epsilon_o) + \mathcal{O}(\epsilon_t)$ .

In the end of this section, we analyze the expressive power for typical layered circuit, which is given by our appendix method. In fact, it is problem-dependent and can be replaced with smarter ansatz with tasks at hand.

Expressibility is proposed recently as a distance, which measures the output states distribution of layered circuits and the Haar [26]. To explicitly present this value, Kullback-Leibler (KL) divergence(or relative entropy) is employed to estimate this distance, which is denoted as *Expr*. A highly expressible circuit would produce a small Kullback-Leibler value. In this part, besides numerically calculating *Expr* of circuit given in Appendix C, three other types of parametrized circuits are also studied as comparisons, which are also specified in Appendix C. Three compared circuits are repeated with up to 5 times to calculate *Expr*, while our typical circuit stays unchanged. Figure 4 shows the results of Expressibility values (or KL divergences), where circle, diamond and square represents three comparison circuits and cyan cross labels the circuit given by us. The layered circuit in this work has a similar performance as multi-applications of circuit 2 and 3 in Figure A1. Remarkably, in general, repeating a circuit layer would increase the expressive power. However, as no entanglement gate exists in circuit 1 in Figure A1, this argument does not hold for that circuit. More information on expressibility and its simulation can be found in Appendix C or related work [26].



**Figure 4.** Expressibility values computed for circuits specified in Appendix C and circuit used in our protocol. Circuit 1 (red circles), 2 (green diamonds) and 3 (blue squares) are repeatedly applied with up to 5, while the circuit employed in our strategy (cyan cross) keeps unchanged.

#### 4. Numerical Experiments

In this section, numerical experiments are investigated, including applications on Hamiltonian simulation and density matrix evolution. In our protocol,  $\rho$  is taken as (a) Bell state; (b) GHZ state and (c) Hamiltonian of a liquid NMR sample, Crotonic acid, which is specified in the Appendix E.

To generate the simulating layered circuit, two learning methods are employed, with

$$f(\boldsymbol{\theta}) = tr(\rho_t \Omega_3) \tag{20}$$

being the objective function.  $\theta = \alpha$ ,  $\beta$  are for *Method 1* and *Method 2*. The target state can be expressed as,

$$\rho_t = U_d \cdot e^{-\iota \rho t} \otimes I_a \cdot U_e(\Omega_0) \qquad (Method \ 1),$$
  

$$\rho_t = \Omega_0 \qquad (Method \ 2).$$

And additionally,  $\Omega_3 = U_d \cdot U \otimes I_a \cdot U_e(\Omega_0)$ , where

$$U = U_1(\boldsymbol{\alpha}) = \prod_{i=1}^m e^{-it\alpha_i\rho_i} \qquad (Method \ 1),$$
  
$$U = U_2(\boldsymbol{\beta}) = U_i^{-n_c} U_{i+1}(\boldsymbol{\beta}) \qquad (Method \ 2).$$

This is specified in Appendix D.

To find optimal parameter configuration, gradient-based method is employed. To begin with, random numbers by a single uniformly distribution are generated and assigned as the value of initial parameters. During the training process, the approximation of the partial derivatives can be estimated by following symmetric difference quotient

$$\partial_j f = \frac{f(\boldsymbol{\theta} + \Delta \theta_j) - f(\boldsymbol{\theta} - \Delta \theta_j)}{2\Delta \theta_j},\tag{21}$$

which can be realized by repeatedly running the training circuit with a small perturbation on the  $\theta$ , where  $\Delta \theta_j$  is naively chosen as 0.01 in our simulation. Therefore, the optimal configuration of the parameters can be obtained by repeating

$$\boldsymbol{\theta} = \boldsymbol{\theta} + \eta \boldsymbol{\nabla} f(\boldsymbol{\theta}), \tag{22}$$

where  $\nabla f(\mathbf{x}) = (\partial_1 f, \partial_2 f, ...)$  and the learning rate,  $\eta$ , is fixed at 0.02. For simulating *Method 1* and *Method 2*, the maximum numbers for repeating Equation (22), i.e., iterations are set as 300 and 350, respectively.

In this prototypical simulation, the program is conducted with assumptions that all one or two quantum gates are accurate. Three circuits with the depths of 3, 4 and 5 are employed for the simulation the evolution. The numbers of parameters to be optimized are thus 12, 24 and 40, which converge to  $\mathcal{O}(m \times n)$ , where *m* is the depth and *n* is the size of circuit. As the comparison, if  $\epsilon$  is tolerant error,  $\mathcal{O}(3/\epsilon)$ ,  $\mathcal{O}(4/\epsilon)$  and  $\mathcal{O}(5/\epsilon)$  depth circuits are required for simulation by lie-product decomposition. For density matrix evolution,  $\mathcal{O}(1/\epsilon)$  copies of density matrix are required as with  $\mathcal{O}(1/\epsilon)$ -depth circuit.

For data collection, strategies are different for *Method 1* and *Method 2*. For *Method 1*, once the optimization completed, parameters is supposed to be optimal for simulating a *t*-time interval evolution. Otherwise, we need to re-run the simulation and repeat Equation (22) until the objective function satisfying the optimized threshold. For *Method 2*, optimizing ancillary layered circuit is not once for all. After the *i*-th repetition of training ancillary layered circuit with Equation (22) for 350 iterations, only  $U_i^{-n_c}U_{i+1}(\beta)$  is learned. Therefore,  $\mathcal{O}(\log(t/\Delta t))$  repetitions are required. In our simulation,  $\Delta t/t = 1/2^{10}$ , i.e., 10 steps are required and the base number, 2, determines the compression efficiency of the circuit.

Figure 5 shows the results of our simulation, which provide supports on our protocol by converging to the target matrix evolutions. (a) (b) and (c) simulate the evolutions of Bell state, GHZ state and Hamiltonian of Crotonic acid, where the evolution timescales are set as 0.05, 0.1 and 0.2, respectively. Fidelities are shown by vertical axis and calculated by tracing the inner product of two operators. The results show that, with the accurate local quantum gate assumption, the simulating circuit gets more and more like the evolution of certain matrix evolution as with the number of iterations.



**Figure 5.** Results of *Method 1* and *Method 2* for simulating the evolutions by  $\rho$  for a period of t = 0.05, 0.1 and 0.2. The fidelities (vertical axis) vary with the iterations(horizon axis).  $\rho$  is chosen as (**a**) bell state; (**b**) GHZ state and (**c**) Hamiltonian of a molecular Crotonic acid system.

#### 5. Limitation

In the end, defects in our protocol should be listed. First, only Hermitian operations can be dealt with. Evolution of an open quantum system is not considered in this article. With formalism for the description of open quantum systems, this target is potentially solved in future work [33]. Second, similar to the widely-used parameterized circuits architecture such as QAOA and VQE, expressive ability of ansatz in our methods determinate the upper-bound of accuracy [25,28]. Therefore, a proper ansatz will have a good performance. If the termination threshold cannot be reached, an alternative ansatz or iterative training algorithms should be resorted to. Third, optimization should be clarified, which is also the most important one. Many optimization problems are in fact NP-hard problems. The methods embedded are essentially optimization-based, and cannot get over them, too. Taking an example of gradient-based methods, they cannot avoid the local optimum problems, especially when the feasible region is complicated. Therefore, the proposed methods would fail in finding a global optimum with a bad initial guess, just as the classical cases. Although numerical results are good without considering the initialization, we have to admit a good initial guess is crucial to decrease the possibility of failure, especially for dealing with a larger problem. Investigations on the optimization method itself should attract more attention, which benefits not only the exploitation of near-future quantum devices but also for most modern technologies [34,35].

# 6. Conclusions

NISQ era has come in and would last for decades. Thus, finding a near-future algorithm which exploits the power of NISQ devices becomes more and more important. Simulating matrix evolution, which is an essential module for many quantum algorithms, such as density matrix evolution for quantum principal component analysis, and static or dynamic simulation for quantum systems. The existed implementation still relies on a deep quantum circuit, which is intractable with current NISQ devices.

In this paper, a heuristic layered circuit protocol is proposed for simulating  $e^{-i\rho t}$  with Hermitian  $\rho$ . To construct these circuits, two methods are given. *Method* 1 is with the classical optimal control theory and *Method* 2 is with the hybrid quantum-classical paradigm. For *Method* 1, learning a layered circuit requires  $O(4^n \times r)$  operations, where *n* is the size of quantum circuit and *r* is the number of total iterations. As computational resource may be put into its limits, we provide *Method* 2, which is hybridized with a

quantum agent. As a comparison, time complexity is  $\mathcal{O}(m \times n \times r \times \log(t/\Delta t))$ , with 2n being the size of circuit and m being circuit depth. From the point of view of error, although *Method 1* is in general inefficient, the error can be bounded to  $\mathcal{O}(\epsilon_o)$ . *Method 2* introduces a larger one,  $\mathcal{O}((t^2/\epsilon_t)\epsilon_o) + \mathcal{O}(\epsilon_t)$ . Only when  $\epsilon_o$  is sufficiently small, *Method 2* can approach an accuracy as Trotter method. Accordingly, circuits by both methods would be shallow, which is easily deployed on current devices.

Simulating matrix evolution is important for many quantum information tasks. Our protocol, which realizes hermitian matrix evolutions, is supposed to contribute to the field of NISQ algorithm. For layered circuit in our both methods, only basic quantum gates are required, which means at most 2-qubit interactions are needed. With respect to experimental technology, it is already mature for most physical platforms, such as optical lattice, spin-based, and superconducting qubit [7,36,37]. In despite of some limitations, the protocol is with an affordable computational consumption, which paves a way for possibly applications on NISQ devices. In addition, ancillary layered circuit, which serves as a compiler in this article, is promising to be an important subroutine in near future and provides a new way to exploit the power of NISQ devices.

**Author Contributions:** K.L. and P.G. contribute equally to this work. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Major Key Project OF PCL and National Natural Science Foundation of China under grant No. 11905111.

**Data Availability Statement:** All data for the figure and table are available on request. All other data about experiments are available upon reasonable request.

Conflicts of Interest: The authors declare no competing interests.

#### Abbreviations

The following abbreviations are used in this manuscript:

NISQ Noisy Intermediate-Scale Quantum

# Appendix A. Derivations on $e^{-i\Delta t\rho}$ in *Method* 2

First, if one wants to simulate  $e^{-i\Delta t\rho}$  with  $\prod_i e^{-i\Delta t\beta_i\rho_i}$ , the idea of optimizations can always be resorted to. An objective function can thus be defined as,

$$f(\boldsymbol{\beta}) = \operatorname{tr}(e^{-i\Delta t\rho}\sigma e^{i\Delta t\rho}\prod_{i}e^{-i\Delta t\beta_{i}\rho_{i}}\sigma\prod_{i}e^{i\Delta t\beta_{i}\rho_{i}}),\tag{A1}$$

where  $\sigma$  is an arbitrary initial density matrix, and operators in  $\langle \cdot \rangle$  satisfy the permutation equality. We expand matrix exponentiation in Equation (A1) with Taylor series on  $\Delta t$ ,

$$e^{-i\Delta t\rho} = \sum_{k=0}^{\infty} (-i\Delta t\rho)^k / k!,$$
  
$$\prod_{i=1}^{N} e^{-i\Delta t\beta_i\rho_i} = \prod_{i=1}^{N} \sum_{k=0}^{\infty} (-i\Delta t\beta_i\rho_i)^k / k!.$$
 (A2)

With substitutions such as,  $H \rightarrow -i\Delta t\rho$  and  $M_i \rightarrow -i\Delta t\beta_i\rho_i$ , ingredients in Equation (A1) can be rewritten as,

$$\begin{split} e^{i\Delta t\rho} & \times \prod_{i=N}^{1} e^{-i\Delta t\beta_{i}\rho_{i}} \\ &= \left(\frac{(-H)^{0}}{0!} + \frac{(-H)^{2}}{1!} + \frac{(-H)^{3}}{3!} + \cdots\right) \times \left(\frac{M_{N}^{0}}{0!} + \frac{M_{N}^{1}}{1!} + \frac{M_{N}^{2}}{2!} + \cdots\right) \dots \left(\frac{M_{1}^{0}}{0!} + \frac{M_{1}^{1}}{1!} + \frac{M_{1}^{2}}{2!} + \cdots\right) \\ &= \sum_{k=0}^{\infty} \sum_{\substack{j_{i}, j_{\rho} \geq 0, \\ \sum_{i} j_{i} + j_{\rho} = k}} \frac{(-H)^{j_{\rho}}}{j_{\rho}!} \prod_{i=N}^{1} \frac{M_{i}^{j_{i}}}{j_{i}!}, \\ &\prod_{i=1}^{N} e^{i\Delta t\beta_{i}\rho_{i}} \times e^{-i\Delta t\rho} \\ &= \left(\frac{(-M_{1})^{0}}{0!} + \frac{(-M_{1})^{1}}{1!} + \frac{(-M_{1})^{2}}{2!} + \cdots\right) \cdots \left(\frac{(-M_{N})^{0}}{0!} + \frac{(-M_{N})^{1}}{1!} + \frac{(-M_{N})^{2}}{2!} + \cdots\right) \times \left(\frac{H^{0}}{0!} + \frac{H^{2}}{1!} + \frac{H^{3}}{3!} + \cdots\right) \\ &= \sum_{k=0}^{\infty} \sum_{\substack{j_{i}, j_{\rho} \geq 0, \\ \sum_{i} j_{i} + j_{\rho} = k}} \prod_{i=1}^{N} \frac{(-M_{i})^{j_{i}}}{j_{\rho}!} \frac{H^{j_{\rho}}}{j_{\rho}!}. \end{split}$$

Under the circumstances that no approximation is adopted, the objective function is

$$f = \langle \psi | \sum_{k=0}^{\infty} \sum_{\substack{j_i, j_\rho \ge 0, \\ \sum_i j_i + j_\rho = k}} \prod_{i=1}^{N} \frac{(-M_i)^{j_i}}{j_i!} \frac{H^{j_\rho}}{j_\rho!} |\psi\rangle \langle \psi | \sum_{k=0}^{\infty} \sum_{\substack{j_i, j_\rho \ge 0, \\ \sum_i j_i + j_\rho = k}} \frac{(-H)^{j_\rho}}{j_\rho!} \prod_{i=N}^{1} \frac{M_i^{j_i}}{j_i!} |\psi\rangle,$$
(A3)

where  $\sigma = |\psi\rangle \langle \psi|$  and  $\langle \cdot \rangle$  represents the expectation on  $|\psi\rangle$ . Obviously, it is a taylor-like polynomial with respect to  $\Delta t$ . As  $\Delta t$  can be a sufficiently small quantity when we simulate  $e^{-it\rho}$ , only lowest order terms on  $\Delta t$  are considered. For the objective function *f*, all terms which is linear with or higher than  $\mathcal{O}(\Delta t^2)$  are ignored. Therefore, *f* can be simplified as,

$$(1 + \langle \frac{H^2}{2!} \rangle - \langle H \sum_{i} M_i \rangle + \langle \sum_{i} \frac{M_i^2}{2!} \rangle + \langle \sum_{i>j} M_i M_j \rangle)^2 - (\langle H \rangle - \langle \sum M_i \rangle)^2,$$
(A4)

where  $\sum M_i = H$  is the requirement for eliminating above lowest order error, which is also independent of  $|\psi\rangle$ . In the original representation,  $\beta_i$  satisfy following correspondence

$$\sum \beta_i \rho_i = \rho. \tag{A5}$$

which is also the correspondence by Lie–Trotter products. Therefore, a parameterized quantum circuit for  $e^{-i\Delta t\rho}$  can be obtained by Equation (A5), where  $\rho_i$  is comparably easy-access to realize and benchmark in near-future quantum devices.

### Appendix B. Method to Generate an N-Layer Circuit

In this section, a method to generate an *N*-layer circuit structure with all two-body interactions is introduced, where *N* is the size of the system.  $P_2(i, j)$  are assumed as the products of Pauli matrices, where  $P_2(i, j) = \sigma_k \otimes \sigma_l$  are on both *i*-th and *j*-th qubits and i, j = 1, ..., N and k, l = x, y, z. All of them can be presented in an  $N \times N$  triangle parameter matrix *P*,

$$P = \begin{bmatrix} 1 & P_2(1,2) & P_2(1,3) & P_2(1,4) & P_2(1,5) & \cdots, & P_2(1,N) \\ 2 & P_2(2,3) & P_2(2,4) & P_2(2,5) & \cdots, & P_2(2,N) \\ 3 & P_2(3,4) & P_2(3,5) & \cdots, & P_2(3,N) \\ 4 & P_2(4,5) & \cdots, & P_2(3,N) \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & &$$

where  $P_{ii} = i$ ,  $P_{ij} = P_2(i, j)$  for i < j and  $P_{ij} = 0$  for i > j. Thus, the non-zeros elements in *P* can be divided into *N* group with the following Method A1, where *N* is supposed to be odd.

#### Method A1 Algorithm on grouping the Pauli words

Input: Parameter matrix *P*, *N* null sets, and a  $N \times N$  label matrix *L*, where  $L_{mn} = 1$  for arbitrary *m*, *n*. Output: *N* sets of Pauli words:  $S_1, S_2, ..., S_N$ for i = 1, ..., N do for j = i, ..., N do if  $i = j \cap L_{ij} = 1$  then  $P_{ij} \in S_i, L_{ij} = 0$ . else for m = 1, ..., N do if  $L_{mi} = 1 \cap L_{mj} = 1$  then  $P_{ij} \in S_m, L_{mi} = 0, L_{mj} = 0$ end if end for end if end for end for

With the complexity of  $\mathcal{O}(N^3)$  operations, Those Pauli words are divided into *N* group, where the Pauli words in each group commute with each other. Via matrix exponentiation formation multiplied by tunable parameters, they can be arranged into a parameterized circuit, with a depth of *N*, being linear with the size of the system. If different type Pauli combinations are considered,  $\mathcal{O}(N)$  is the depth. Therefore, this algorithm provided a depth of  $\mathcal{O}(N)$  parameterized circuit with all two-body interactions considered. Remarkably, this is one type of problem-agnostic circuit, which exploit the power of current quantum devices with two-body interactions available. For different problems, problem-inspired circuit can be designed, employing information about the problem, which bring better expressibility and trainability. Surely, we will study the expressibility of our proposed circuit in next section.

#### Appendix C. Expressibility of the Layered Circuit

A layered circuit with excellent expression is more likely to represent a target state. To measure the expressive power of a circuit, expressibility is defined in recent work as a distance of two state distributions, characterized with  $|| \cdot ||$ , the Hilbert-Schmidt norm [26].

$$A = || \int_{Haar} (|\psi\rangle \langle \psi|)^{\otimes t} d\psi - \int_{\theta} (|\psi_{\theta}\rangle \langle \psi_{\theta}|)^{\otimes t} d\psi_{\theta} ||,$$
(A7)

where  $|\psi\rangle_{\theta}$  are outputs of a layered circuit with randomized configuration  $\theta$  and  $|\psi\rangle$  are from a distribution according to the Haar measure. It is a generalization from the investigation of pseudo-random circuit. Thus, a highly expressible circuit would produce a small *A*, with *A* = 0 corresponding to being maximally expressive, i.e., generating a state distribution to the Haar measure.

In order to explicitly estimate the expressibility with a discrete simulated result, the Kullback-Leibler (KL) divergence, i.e., relative entropy is employed, which measures the difference between one probability distribution and a reference probability distribution. It is denoted as *Expr*,

$$Expr = D_{KL}(P_l(f|\boldsymbol{\theta}, \boldsymbol{\theta'})|P_{Haar}(f)), \tag{A8}$$

where  $P_1(f|\theta, \theta')$  is the probability distribution of  $f = |\langle \psi_{\theta} | \psi_{\theta'} \rangle|^2$ , outputs of a layered circuit with random  $\theta$  and  $\theta'$ . For the reference probability distribution, i.e., to Haar measure,  $P_{Haar}(f) = (N-1)(1-f)^{N-2}$  is analytical, with N being size of dimension. *Expr* scores the similarity to the distribution of Haar as a layered circuit with a lower value of KL divergence is a more expressible circuit.



**Figure A1.** A set of circuit templates considered in the study, each labeled with a circuit ID. The dashed box indicates a single circuit layer, denoted by *L* in the text, that can be repeated.

Accordingly, numerical experiments to compute and compare the expressibility in the strategy with three other typical layered circuits are studied, which are shown in Figure A1. They are all 4 qubit circuit. As with the original work, to construct a distribution of  $P_l$  with histogram, a bin size is set as 75, and 5000 f are measured. For each type of layered circuits except for the one in our strategy, the instances where the number of layers are investigated with up to 5. Compared with 10,000 times repeatedly running the circuits and state tomography, f can be measured directly with the setup proposed in Figure A2 for 5000 time and without tomography. Only 1-qubit is added as the ancillary system A, at the end of circuit, the system is evolved as

$$\rho = |0\rangle \langle 0| \otimes |\psi_{\theta}\rangle \langle \psi_{\theta}| + |1\rangle \langle 0| \otimes |\psi_{\theta'}\rangle \langle \psi_{\theta}| + |0\rangle \langle 1| \otimes |\psi_{\theta}\rangle \langle \psi_{\theta'}| + |1\rangle \langle 1| \otimes |\psi_{\theta'}\rangle \langle \psi_{\theta'}| \,. \tag{A9}$$



**Figure A2.** Circuit for measurement of the expressivity of the layered circuit. It costs one extra qubit as ancillary.  $\theta$  and  $\theta'$  are two random parameter configurations for layered circuits  $U_{\theta}$  and  $U_{\theta'}$ , *H* is a one-qubit Hadamard gate.

Via the measurements on ancillary system, *f* can be obtained with the real and imaginary part of  $\langle \langle \psi_{\theta} | \psi_{\theta'} \rangle \rangle$ , which are extracted as

$$Re = tr(\sigma_x \otimes I \cdot \rho) = \frac{1}{2} (\langle \psi_{\theta} | \psi_{\theta'} \rangle + \langle \psi_{\theta'} | \psi_{\theta} \rangle)$$
  

$$Im = tr(\sigma_y \otimes I \cdot \rho) = \frac{-i}{2} (\langle \psi_{\theta} | \psi_{\theta'} \rangle - \langle \psi_{\theta'} | \psi_{\theta} \rangle).$$
(A10)

Results are shown in the Figure 4 in main text. For three types of layered circuits which are denoted in circle, diamond and square, the instances are considered where the circuit, which is as a unit layer, are repeated with up to 5. For the circuit proposed by Method A1, the repetition is always 1.

In general, repeating a circuit layer in Figure A1 would increase the expressibility. As there is no entanglement gate in circuit 1, this argument does not hold. For circuit 2 and 3, even if a single layer for circuit 3 have pretty good expression, this argument holds. From the result, there are convergences for the metric of expressibility. For the argument whether more numbers of layers can help the expression, at least, we can not obtain more information from this metric. Additionally, the layered circuit generated in our strategy, which is labeled as cyan cross, keeps a same structure. It has a similar performance with respect to the repeated circuits 2 and 3. As a small value of expressibility imply an excellent expression, the generated circuits in the strategy performs no worse than other typical circuit.

# Appendix D. Objective Functions

In this section, the explicit expression of the objective functions is derived, as well as the step-by-step illustrations of the circuit to implement the Numerical simulation. The objective function can be concluded as

$$f(\boldsymbol{\theta}) = tr(\rho_t \Omega_3),\tag{A11}$$

where  $\theta = \alpha$ ,  $\beta$  are for *Method 1* and *Method 2*, respectively. As for intermediate states and quantum operations of the circuit shown in original manuscript, they are depicted as,

$$\Omega_{0} = |\psi_{0}\rangle \langle \psi_{0}|, \quad |\psi_{0}\rangle = |0\rangle_{p} |0\rangle_{a}, 
\Omega_{1} = |\psi_{1}\rangle \langle \psi_{1}|, \quad |\psi_{1}\rangle = U_{e} |\psi_{0}\rangle, 
\Omega_{2} = |\psi_{2}\rangle \langle \psi_{2}|, \quad |\psi_{2}\rangle = U_{l} \otimes I_{a} |\psi_{1}\rangle, 
\Omega_{3} = |\psi_{3}\rangle \langle \psi_{3}|, \quad |\psi_{3}\rangle = U_{d} |\psi_{2}\rangle.$$
(A12)

where  $U_e = I_p \otimes H_a \cdot C_z \cdot H_p \otimes H_a$ ,  $U_d = H_p \otimes H_a \cdot C_z \cdot I_p \otimes H_a$ . In addition,  $|\psi_1\rangle = \sum_{i=1}^d |i\rangle_p \otimes |i\rangle_a$  is the bell state pairs, where the idea of Choi matrix for assisting tomography is employed.

With respect to the  $U_l$  in *Method* 1, it aims at finding optimal parameters  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ...)$  to generate a layered circuit. While, for the *i*-th repetition of *Method* 2, it aims at finding appropriate parameters  $\boldsymbol{\beta} = (\beta_1, \beta_2, ...)$ , generating a layered circuit which realizes repeatedly applied  $U_i$ . Thus, the different  $U_l$  are to be optimized

$$U_1(\boldsymbol{\alpha}) = \prod_{i=1}^m e^{-it\alpha_i\rho_i} \qquad (Method \ 1),$$
  
$$U_2(\boldsymbol{\beta}) = U_i^{-2}U_{i+1}(\boldsymbol{\beta}) \qquad (Method \ 2).$$

As for target states, they are also different. *Method* 1 realizes an evolution by a target Hermitian matrix on system *p* while *Method* 2 realizes  $I_p$ . Therefore, the density matrix  $\rho_t$  for both strategies are

$$\rho_t = U_d \cdot e^{-i\rho t} \otimes I_a \cdot U_e(\Omega_0) \qquad (Method 1),$$
$$\rho_t = \Omega_0 \qquad (Method 2),$$

where  $\rho$  can be density matrix or Hamiltonian, for simulating hermitian matrix exponentiation.

#### **Appendix E. Matrices for Numerical Simulation**

In this section, the detailed information for  $\rho$  in numerical simulation are given. Three cases are studied. Two of them are density matrices and one is a Hamiltonian from previous experiments. The specified form are as follows.

a. Bell State

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad \rho = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 1 \end{pmatrix}.$$
 (A13)

b. GHZ State

c. Hamiltonian of Crotonic Acid

The Hamiltonian simulated in our numerical experiments is from <sup>13</sup>C-labeled Crotonic acid dissolved in d6-acetone, which is usually employed as a four-qubit quantum system for NMR-based quantum information processing [18].

The structure of this molecule is shown in Figure A3, where  $C_1$  to  $C_4$  are four carbon atoms, M is one of three hydrogen atoms in a methyl group,  $H_1$  and  $H_2$  are two hydrogen atoms. When it is used as a four qubit quantum system, the hydrogen atoms are decoupled from the carbon atoms with a shaped radio frequency pulse. Under the circumstance of weak coupling, the internal Hamiltonian of this molecule can be expressed as,

$$H_{int} = \sum_{j=1}^{4} \frac{1}{2} \nu_j \sigma_z^j + \sum_{j< k}^{4} \frac{\pi}{2} J_{jk} \sigma_z^j \sigma_z^k.$$
 (A15)

 $v_j$  and  $J_{jk}$  are inside Hamiltonian parameters whose values are listed in the table besides the molecule.  $v_j$ , the chemical shift, are the diagonal elements and  $J_{jk}$ , the J-coupling, are the off-diagonal elements. As for  $T_2$ , the transverse relaxation time, is not involved in our simulation as the unitary process is assumed. All parameters can be found in related paper which is under a magnetic field of 9.4 T at room temperature (296.5 K).



**Figure A3.** Molecular structure of <sup>13</sup>C-labeled Crotonic acid.  $\nu_j$  and  $J_{jk}$  are the chemical shifts and J-couplings, respectively, which are listed by the diagonal and off-diagonal elements. T<sub>2</sub> (in Seconds) are the relaxation time which are shown at bottom.

# References

- 1. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, 41, 303–332. [CrossRef]
- 2. Grover, L.K. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.* 1997, 79, 325. [CrossRef]
- 3. Boixo, S.; Rønnow, T.F.; Isakov, S.V.; Wang, Z.; Wecker, D.; Lidar, D.A.; Martinis, J.M.; Troyer, M. Evidence for quantum annealing with more than one hundred qubits. *Nat. Phys.* **2014**, *10*, 218–224. [CrossRef]
- 4. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, 549, 195–202. doi: 10.1038/nature23474. [CrossRef] [PubMed]
- 5. Preskill, J. Quantum Computing in the NISQ era and beyond. Quantum 2018, 2, 79. [CrossRef]
- 6. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef]
- 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]
- 8. Kressner, D.; Luce, R. Fast computation of the matrix exponential for a Toeplitz matrix. *SIAM J. Matrix Anal. Appl.* **2018**, *39*, 23–47. [CrossRef]
- 9. Low, G.H.; Chuang, I.L. Hamiltonian simulation by qubitization. *Quantum* 2019, 3, 163. [CrossRef]
- 10. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. Nat. Phys. 2014, 10, 631-633. [CrossRef]
- 11. Rebentrost, P.; Schuld, M.; Wossnig, L.; Petruccione, F.; Lloyd, S. Quantum gradient descent and Newton's method for constrained polynomial optimization. *New J. Phys.* **2019**, *21*, 073023. [CrossRef]
- 12. Gao, P.; Li, K.; Wei, S.; Gao, J.; Long, G. Quantum gradient algorithm for general polynomials. *Phys. Rev. A* 2021, 103, 042403. [CrossRef]
- 13. Gao, P.; Li, K.; Wei, S.; Long, G.L. Quantum second-order optimization algorithm for general polynomials. *Sci. China Phys. Mech. Astron.* **2021**, *64*, 100311. [CrossRef]
- 14. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, 103, 150502. [CrossRef] [PubMed]
- 15. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef]
- 16. 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 (NISQ) algorithms. *arXiv* **2021**, arXiv:2101.08448.
- 17. 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]
- 18. Li, K.; Wei, S.; Gao, P.; Zhang, F.; Zhou, Z.; Xin, T.; Wang, X.; Rebentrost, P.; Long, G. Optimizing a polynomial function on a quantum processor. *NPJ Quant. Inf.* **2021**, *7*, 16. [CrossRef]
- 19. Gui-Lu, L. General quantum interference principle and duality computer. Commun. Theor. Phys. 2006, 45, 825. [CrossRef]
- 20. Long, G.; Liu, Y. Duality quantum computing. Front. Comput. Sci. China 2008, 2, 167. [CrossRef]
- 21. Childs, A.M.; Wiebe, N. Hamiltonian simulation using linear combinations of unitary operations. arXiv 2012, arXiv:1202.5822.
- 22. Childs, A.M.; Su, Y.; Tran, M.C.; Wiebe, N.; Zhu, S. Theory of trotter error with commutator scaling. *Phys. Rev. X* 2021, *11*, 011020. [CrossRef]
- 23. Farhi, E.; Goldstone, J.; Gutmann, S. A quantum approximate optimization algorithm. arXiv 2014, arXiv:1411.4028.
- 24. 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]
- 25. Nakaji, K.; Yamamoto, N. Expressibility of the alternating layered ansatz for quantum computation. arXiv 2020, arXiv:2005.12537.
- 26. Sim, S.; Johnson, P.D.; Aspuru-Guzik, A. Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Adv. Quant. Technol.* **2019**, *2*, 1900070. [CrossRef]
- 27. Romero, J.; Babbush, R.; McClean, J.R.; Hempel, C.; Love, P.J.; Aspuru-Guzik, A. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quant. Sci. Technol.* **2018**, *4*, 014008. [CrossRef]
- 28. 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]
- 29. Wiersema, R.; Zhou, C.; de Sereville, Y.; Carrasquilla, J.F.; Kim, Y.B.; Yuen, H. Exploring entanglement and optimization within the hamiltonian variational ansatz. *PRX Quant.* **2020**, *1*, 020319. [CrossRef]
- 30. Anand, A.; Schleich, P.; Alperin-Lea, S.; Jensen, P.W.; Sim, S.; Díaz-Tinoco, M.; Kottmann, J.S.; Degroote, M.; Izmaylov, A.F.; Aspuru-Guzik, A. A quantum computing view on unitary coupled cluster theory. *Chem. Soc. Rev.* **2022**. [CrossRef]
- Lavrijsen, W.; Tudor, A.; Müller, J.; Iancu, C.; de Jong, W. Classical optimizers for noisy intermediate-scale quantum devices. In Proceedings of the 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), Denver, CO, USA, 12–16 October 2020; pp. 267–277.
- 32. Pan, F.; Zhang, P. Simulating the Sycamore quantum supremacy circuits. arXiv 2021, arXiv:2103.03074.
- 33. Eleuch, H.; Rotter, I. Nearby states in non-Hermitian quantum systems I: Two states. Eur. Phys. J. D 2015, 69, 229. [CrossRef]
- 34. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* 2016, arXiv:1609.04747.
- 35. Gill, P.E.; Murray, W.; Wright, M.H. Practical Optimization; SIAM: Philadelphia, PA, USA, 2019.

- 36. Homid, A.; Abdel-Aty, M.; Qasymeh, M.; Eleuch, H. Efficient quantum gates and algorithms in an engineered optical lattice. *Sci. Rep.* **2021**, *11*, 15402. [CrossRef]
- 37. Vandersypen, L.M.; Chuang, I.L. NMR techniques for quantum control and computation. *Rev. Mod. Phys.* 2005, *76*, 1037. [CrossRef]