

Article

Evolutionary Optimization of Neural Architectures for Holographic Phase Prediction in Optical Tweezer Arrays

Jaewon Choi and Minhyuk Kim * 

Department of Physics, Korea University, Seoul 02841, Republic of Korea

* Correspondence: minhyukkim@korea.ac.kr

Abstract

We present an evolutionary optimization framework for designing neural architectures for holographic phase prediction in optical tweezer arrays. The framework combines Fourier-transform-based network structures with genetic optimization of layer composition and connectivity, enabling optimization of prediction quality under an inference-time constraint. The optimized architecture achieves phase prediction quality comparable to several iterations of the Gerchberg–Saxton (GS) algorithm while significantly reducing the inference time. The resulting model generalizes to larger array sizes beyond the training set and can be directly implemented on a spatial light modulator. We present this as a proof-of-concept demonstration of the architecture-search methodology and of the physical realizability of the predicted holograms; a fully benchmarked model with large-scale statistical evaluation and real experimental validation is left for future work. Defect-free single-atom array assembly for neutral-atom quantum computation and quantum simulation provides the broader motivation for this line of work.

Keywords: optical tweezers; holographic phase prediction; neural architecture optimization; genetic algorithms; spatial light modulators; neutral-atom quantum computing

1. Introduction

Neutral atoms trapped in optical tweezer arrays have emerged as a versatile platform for quantum information processing and quantum simulation due to their scalability and flexible control of individual qubits [1–4]. Optical tweezers, realized by tightly focused laser beams, create localized optical dipole traps capable of confining single neutral atoms with high spatial resolution [5–7]. Recent experiments have demonstrated large-scale tweezer arrays containing hundreds to thousands of atoms, highlighting the potential of this platform for scalable quantum technologies [8–12].

A key challenge in optical tweezer platforms arises from the stochastic nature of the atom loading process. Due to collisional blockade, the initial loading typically results in a filling fraction of approximately 50% [6,7]. To overcome this limitation, atom rearrangement techniques have been developed, in which atoms are detected, assigned to target positions using optimization algorithms, and dynamically transported to form defect-free arrays [9,13–18]. Because atomic lifetimes and experimental duty cycles are finite, the speed of the rearrangement process plays an important role in determining the overall performance of the system.

A commonly used approach in neutral atom platforms is to generate a static optical tweezer array using a spatial light modulator (SLM) and rearrange individual atoms using movable tweezers controlled by an acousto-optic deflector (AOD) [9,17]. The acousto-optic



Received: 17 June 2026

Revised: 23 July 2026

Accepted: 29 July 2026

Published: 30 July 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

approach enables rapid atom shuttling along grid-line directions and has become a key technique toward fault-tolerant quantum computation [19–21]. In contrast, hologram-based approaches using SLMs offer a higher degree of freedom in tweezer manipulation, as all tweezers can be controlled simultaneously with greater flexibility [14,16,18]. Furthermore, quantum simulations and adiabatic quantum computations for optimization problems often require defect-free arrays with more complex geometries [2,4]. In such cases, SLM-based tweezer rearrangement remains an important approach.

One challenge in SLM-based tweezer rearrangement is the rapid generation of phase patterns corresponding to intermediate stages of the tweezer motion, since the lifetime of single atoms in optical tweezers is finite. Conventional approaches include iterative phase retrieval methods, such as the GS algorithm [16], as well as phase interpolation techniques [22]. Neural-network-based approaches provide a potential alternative by learning a direct mapping from target intensity distributions to phase patterns, thereby avoiding iterative optimization [18]. However, the performance of neural-network-based hologram generation depends strongly on the underlying network architecture. Standard convolutional neural networks (CNNs) rely on local operations with finite receptive fields, whereas holographic phase prediction is governed by global interference effects arising from Fourier-domain propagation. Although deeper architectures may improve prediction quality, they also increase inference time and computational complexity. Determining an appropriate architecture that balances prediction quality and inference speed therefore remains a nontrivial design problem [23]. In this context, evolutionary neural architecture search provides a natural strategy for discovering compact and efficient network architectures through population-based optimization [24,25].

In this work, we present a neural architecture search methodology for holographic phase prediction in optical tweezer arrays, in which Fourier-domain propagation is embedded directly into the searchable network. Prior neural-network approaches to hologram generation employ manually designed network topologies [18,26,27], in which the network structure is fixed by the designer and only its weights are trained. Our framework instead treats the network structure itself as the object of optimization. Fixed FT and IFT operations are embedded into the search space, and a genetic algorithm optimizes the layer composition and connectivity around them. The primary contribution is therefore methodological: a procedure for discovering compact architectures whose search space encodes the Fourier-domain structure of the problem, rather than a new high-performance model. We demonstrate the methodology as a proof of concept, and show through an ablation that embedding Fourier-domain operations into the search space is essential to the quality of the discovered architectures. Our aim is to establish the methodology and the physical realizability of its output as a proof of concept; a comprehensive quantitative benchmark against iterative methods at scale is beyond the present scope and is identified as future work.

2. Problem Formulation

Figure 1 illustrates a typical procedure for optical tweezer rearrangement to form a defect-free atomic array. Single atoms are initially loaded into individual tweezers in a stochastic manner, resulting in randomly occupied sites with approximately 50% filling. To generate a defect-free target array, an assignment algorithm, such as the Hungarian algorithm [15], determines the optimal mapping between initially occupied sites and target positions. During the rearrangement process, the holographic phase pattern on the SLM is continuously updated to move the tweezers. Because the entire tweezer configuration is determined by a single hologram, the atom transport process is divided into multiple intermediate frames, defined by n_f , each corresponding to a small displacement of the

tweezers. By sequentially applying these holograms, the atoms are transported to the target positions, resulting in a defect-free array.

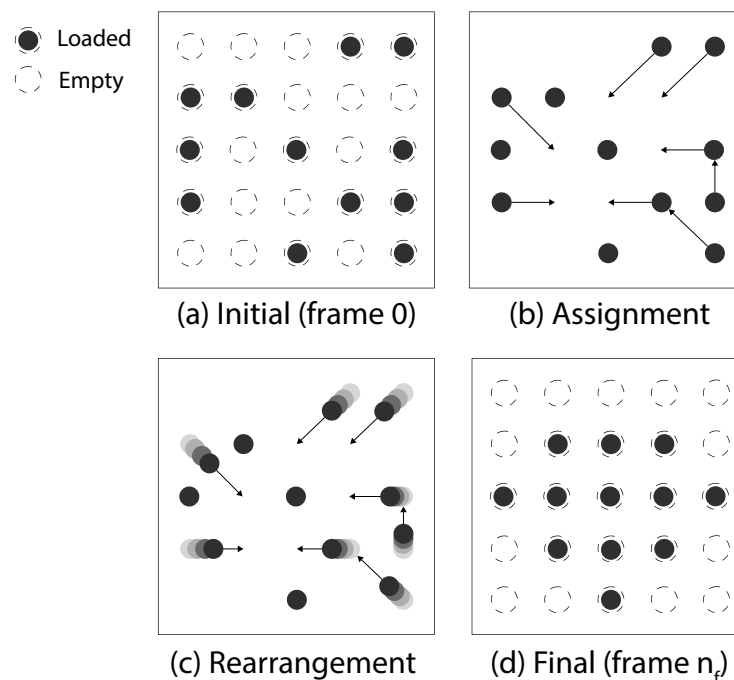


Figure 1. Atom rearrangement protocol. (a) Initial: atoms are stochastically loaded with approximately 50% filling. (b) Assignment: an algorithm determines the mapping from occupied sites to target sites. (c) Rearrangement: atoms are transported to the assigned positions by updating the holographic phase pattern. (d) Final: a defect-free target array is obtained.

Several deep-learning-based approaches have been investigated for computer-generated holography and holographic phase generation [26,27]. In addition, hybrid methods have been developed to combine iterative phase retrieval with neural network components. One example is GS-RNN (Gerchberg–Saxton recurrent neural network) [28], which improves upon purely iterative methods but retains the iterative structure of the GS algorithm, thereby limiting its inference speed. More recently, Lin et al. [18] demonstrated an AI-enabled rearrangement protocol achieving parallel assembly of defect-free atom arrays. However, their approach requires an additional inverse Fourier transform step after the network inference to obtain the SLM phase, rather than predicting it directly. Moreover, the systematic design of the neural architecture itself—specifically, how to select the network structure to balance phase prediction quality against inference speed—remains largely unexplored.

Two principal challenges arise when applying neural networks to holographic phase prediction. The first challenge concerns the locality of convolutional neural networks (CNNs). Convolutional layers are among the most widely used building blocks in modern neural networks due to their efficiency and strong performance across a variety of tasks. However, standard CNN architectures are built upon local operations with finite receptive fields, which inherently limit information flow to spatially neighboring regions. In contrast, holographic phase patterns arise from non-local interference effects governed by Fourier-domain propagation, where each pixel of the phase hologram can influence the entire intensity distribution in the focal domain [8,29]. This fundamental mismatch between the local nature of convolutional operations and the global structure of holographic phase formation limits the effectiveness of purely CNN-based architectures for accurate phase prediction.

The second challenge involves the trade-off between network depth and inference speed. Deeper networks with more layers can achieve higher quality of the tweezer array by capturing more complex input–output relationships. However, increased depth comes at the cost of longer inference times, which directly conflicts with the requirement for fast hologram generation in time-critical atom rearrangement. Conversely, shallow networks offer faster inference but may lack the representational capacity for accurate phase prediction. Finding the optimal balance between depth and quality is nontrivial and cannot be easily resolved through manual architecture design.

3. Method

We present an evolutionary optimization framework for neural architectures in holographic phase prediction. The framework consists of a Fourier transform (FT)/inverse Fourier transform (IFT)-embedded neural architecture and a genetic algorithm (GA) that optimizes the layer composition and connectivity of the network. The objective of the optimization is to identify compact architectures that minimize prediction error subject to an inference-time constraint for optical tweezer holography.

3.1. FT/IFT-Embedded Architecture Design

The neural architecture incorporates FT and IFT layers to explicitly represent Fourier-domain propagation within the network structure. As illustrated in Figure 2, the network accepts two inputs: the SLM phase from the previous rearrangement frame and the target amplitude. Three learnable gene modules are connected by fixed FT and IFT operations, each responsible for processing in either the SLM or focal domain.

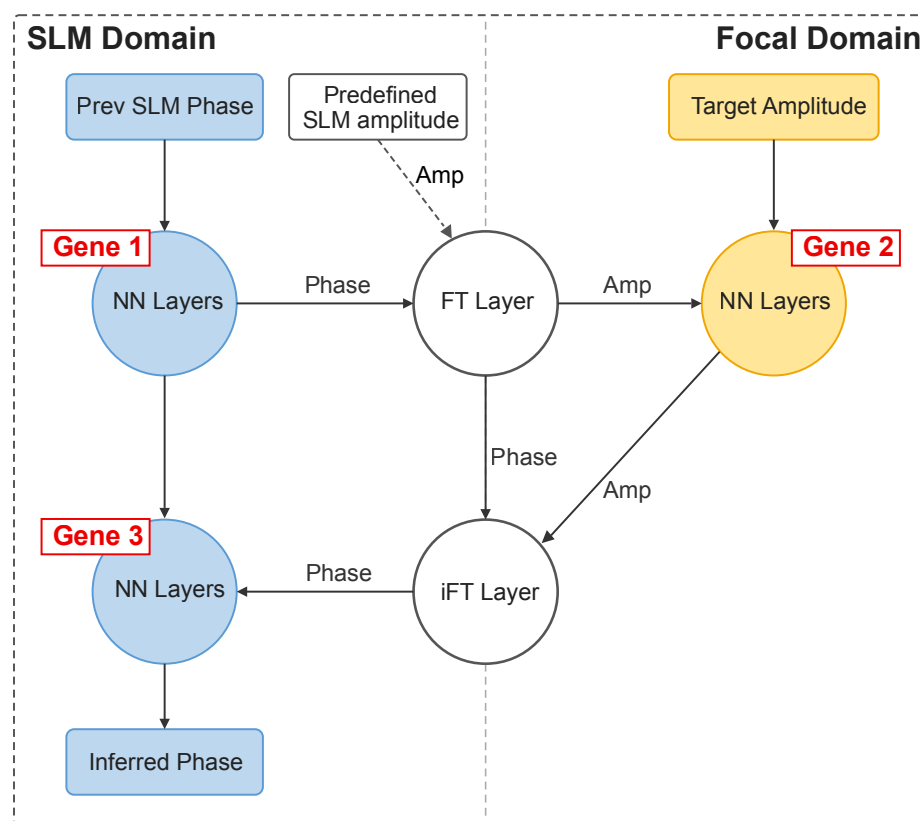


Figure 2. Architecture of the proposed FT/IFT-embedded network. Gene 1 processes the SLM phase and splits into two branches: one passes through FT to the focal domain where Gene 2 operates, followed by IFT back to the SLM domain; the other passes directly. Both branches are concatenated and processed by Gene 3 to produce the final SLM phase.

Gene 1 operates in the SLM domain and processes the input phase, with its output split into two parallel branches. In the first branch, the output of Gene 1 is combined with the predefined SLM amplitude and propagated to the focal domain via FT. The resulting focal-domain amplitude is concatenated with the target amplitude and processed by Gene 2, whose output is then recombined with the focal-domain phase and transformed back to the SLM domain via IFT. In the second branch, the output of Gene 1 is carried forward directly, bypassing the Fourier-domain processing. The outputs of both branches are concatenated and passed to Gene 3, which produces the final predicted SLM phase.

3.2. Data Generation

The training and validation data are generated by simulating the holographic phase-retrieval process on a computational grid of 1024×1024 pixels, matching the SLM resolution. The optical propagation from the SLM plane to the focal plane is modeled by a two-dimensional discrete Fourier transform: the focal-plane field is obtained as the Fourier transform of the complex SLM field $A \cdot \exp(i\phi)$, where A is the amplitude of the incident beam and ϕ is the SLM phase, with the standard fftshift/ifftshift centering convention. The incident beam is taken to have uniform amplitude across the SLM aperture. Each target tweezer configuration is represented in the focal plane as a sparse map that is nonzero only at the tweezer coordinates. A rearrangement sequence is discretized into 15 intermediate frames, and the network is applied autoregressively: the previous-frame SLM phase and the current-frame target amplitude are provided as inputs, and the predicted phase is used as the previous-frame input for the next frame. The initial-frame phase is drawn from a uniform random distribution.

3.3. Evolutionary Architecture Optimization

The NN layer composition within each gene module of the proposed architecture is determined through a genetic algorithm (GA), which searches over the discrete architecture space to minimize prediction error subject to an inference-time constraint. A GA is a gradient-free optimization strategy inspired by natural selection: a population of candidate solutions is randomly initialized, each individual is evaluated via a fitness criterion, and the best-performing individuals are selected and mutated to produce the next generation. This process is well suited for exploring discrete architecture spaces with non-differentiable objectives.

The search space is defined by three genes, each encoding a stack of neural network layers. The available layer types and their mutable parameters are: 2D convolution (number of output channels, kernel size, activation), residual blocks (kernel size, activation), pooling (mode: max or average, kernel size), and 2D recurrent layers (mode: ConvLSTM or ConvGRU, number of hidden channels, kernel size, number of layers, activation). Activation functions are selected from ReLU, GELU, SiLU, tanh, and identity.

Training data are generated from rearrangement simulations, and each architecture is trained to minimize the root-mean-square error (RMSE) between the predicted and target intensity distributions:

$$\mathcal{L}_{\text{RMSE}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (I_i^{\text{pred}} - I_i^{\text{target}})^2}, \quad (1)$$

where I_i^{pred} and I_i^{target} denote the predicted and target intensity at pixel i , respectively, and N is the total number of pixels. The predicted intensity is obtained by propagating the network's output phase to the focal plane using the Fourier propagation model described above, and the target intensity corresponds to the desired tweezer configuration. This

RMSE serves as the training loss for each candidate architecture; how it is combined with an inference-time constraint to drive architecture selection is described below.

A candidate architecture is encoded as a chromosome consisting of the three genes described above. Within a gene, the layers form an ordered stack, and each layer carries its own type together with its mutable parameters. The fixed FT and IFT operations that connect the genes are not part of the chromosome and remain constant throughout the search; the GA therefore optimizes only the per-gene layer composition and the parameters within each layer.

New individuals are produced from the selected elites by mutation, as summarized in Algorithm 1. Each mutation first selects one of the three genes uniformly at random, and then applies one of three operations with equal probability: inserting a new randomly parameterized layer at a random position, removing an existing layer, or modifying a single parameter of an existing layer. These operations allow the search to explore both the depth of each gene and the internal configuration of its layers. Reproduction proceeds by mutation only; a crossover operator was not used in this study. Given the small population and the discrete, variable-length gene encoding, mutation alone provided sufficient exploration of the architecture space, and elitist selection preserved the best architectures across generations.

Algorithm 1 Evolutionary architecture search

```

1: Input: population size  $P$ , elite count  $E$ , generations  $G$ , inference-time threshold  $\tau$ 
2: Output: architecture with lowest validation RMSE
3: Initialize  $P$  random chromosomes, each with inference time  $< \tau$ 
4: Train each and evaluate its validation RMSE
5: for  $g \leftarrow 1$  to  $G$  do
6:   Select the  $E$  elites with lowest RMSE so far                                ▷ elitist selection
7:   Start a new empty offspring list
8:   for all elite chromosome  $e$  do
9:     repeat
10:       $c \leftarrow \text{MUTATE}(e)$ 
11:      if  $c$  has trainable layers and inference time of  $c < \tau$  then
12:        Add  $c$  to the offspring list
13:      end if
14:    until the offspring list contains  $P$  chromosomes
15:  end for
16:  Train each chromosome in the offspring list and evaluate its validation RMSE
17: end for
18: return chromosome with lowest RMSE across all generations
  
```

The RMSE in Equation (1) serves as the sole training loss and, together with an inference-time constraint, determines architecture selection. Inference time enters the search as a hard constraint rather than as a term in the selection objective: candidate architectures whose single-pass inference time exceeds a threshold $\tau \approx 0.033$ s (empirically chosen) are discarded before training and excluded from selection, while the surviving candidates are ranked solely by validation RMSE. This constraint reflects the practical requirement of SLM-based rearrangement, in which the transport path is typically divided into roughly $f_{\#} \approx 15$ intermediate frames [16], and the hologram for each frame must be generated faster than iterative phase retrieval.

4. Results

We evaluate the proposed architecture on 32 validation samples consisting of randomly sampled atom configurations in a 5×5 array with 50% filling probability, with 128 samples used for training. Since the total number of possible configurations is $2^{25} \approx 3.4 \times 10^7$,

the probability of overlap between the training and validation sets is expected to be negligible under uniform random sampling.

The best-performing architecture, identified at generation 15, consists of five learnable layers distributed across three genes, as summarized in Table 1. Gene 1, operating in the SLM domain, contains a pooling layer, a residual block, and a ConvGRU layer, while Gene 3 consists of two convolutional layers that generate the final phase prediction. Interestingly, in this particular solution, Gene 2 in the focal-domain branch contains no learnable layers, indicating that a competitive architecture was found without active layers in that branch. We caution against over-interpreting this as a general property: the outcome varies across optimization runs, and other runs yield architectures in which Gene 2 contains active layers. A systematic characterization of how often the focal-domain branch is retained, and under what conditions, would require a larger set of independent runs and is left for future work. Overall, the optimized architecture demonstrates that a relatively compact network is sufficient to achieve competitive phase prediction performance when combined with the FT/IFT-embedded structure.

Table 1. Layer composition of the best model. k : kernel size, c_{out} : output channels, h : hidden channels.

Gene	Layer	Parameters
1	MaxPool2d	$k = 7$
	Residual (Conv2d $\times 2$)	$k = 11, c_{\text{out}} = 1, \text{SiLU}$
	ConvGRU	$k = 15, h = 6, \text{GELU}$
2	(empty)	—
3	Conv2d	$k = 27, c_{\text{out}} = 13, \text{GELU}$
	Conv2d	$k = 23, c_{\text{out}} = 8, \text{ReLU}$

The evolutionary architecture search and model training were performed on an AWS `m1.p3.2xlarge` instance equipped with an NVIDIA Tesla V100 (16 GB) GPU (NVIDIA Corporation, Santa Clara, CA, USA), using PyTorch 2.6 with CUDA 12.4. Inference time measurements were conducted on a local workstation with an Intel Core Ultra 7 265K processor (Intel Corporation, Santa Clara, CA, USA), 128 GB RAM, and an NVIDIA RTX 5080 (16 GB) GPU (NVIDIA Corporation, Santa Clara, CA, USA).

To evaluate the optimized architecture, we compare the phase prediction performance of the proposed method with the conventional GS algorithm. The GS algorithm was implemented under conditions identical to those of the proposed method, without additional hardware-specific optimization. Figure 3a shows the trap-uniformity RMSE of the inferred 5×5 tweezer arrays with respect to the number of iterations of the GS algorithm. The reference point for the comparison is 5 iterations, which is the typical number used in realistic atom-rearrangement experiments [16]. At this point, the quality of the inferred tweezer arrays is comparable for the two methods. Figure 3b shows the inference time as a function of the number of iterations of GS. At five iterations, the proposed method achieves an approximately 30-fold reduction in inference time. Although optimized implementations of the GS algorithm may reduce this gap, the proposed approach retains the advantage of non-iterative phase prediction. We further compare against the weighted GS (WGS) algorithm, which is commonly used for generating uniform intensity distributions. The per-iteration computation time of WGS is comparable to that of GS, differing by less than roughly 15% in our implementation, so the speed advantage of the proposed non-iterative method holds against both baselines.

We note that, in the context of atom rearrangement, the intermediate holograms correspond to transient stages through which the tweezers pass as the atoms are transported. Because these frames are displayed only momentarily, a modest reduction in trap quality

at intermediate frames has limited impact on the transport of atoms, while the reduced generation time directly benefits the time-critical rearrangement. This makes the favorable inference speed of the proposed method particularly relevant for intermediate-frame hologram generation, even when its trap quality does not exceed that of iterative methods at convergence.

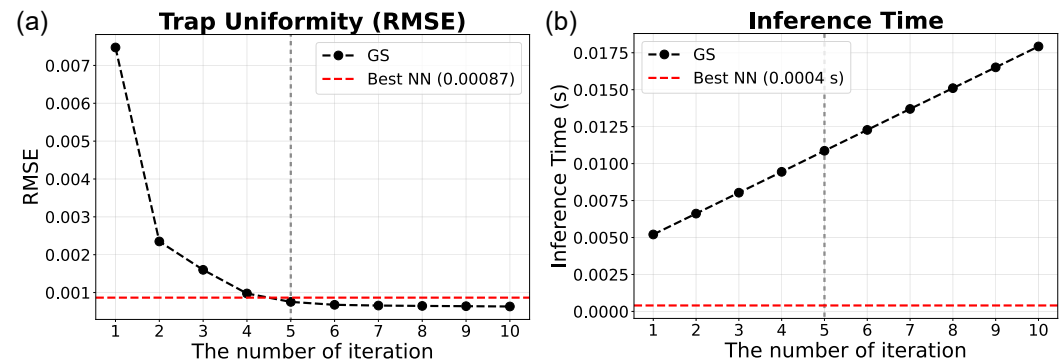


Figure 3. Comparison between the proposed method and the GS algorithm as a function of the number of GS iterations. **(a)** Trap-uniformity RMSE. The optimized architecture achieves competitive RMSE in a single forward pass, comparable to about five GS iterations (best NN: 0.00087). **(b)** Inference time. The GA-optimized architecture achieves substantially reduced inference time while maintaining comparable phase prediction quality (best NN: 0.0004 s). The vertical gray dashed lines at 5 iterations mark the reference point for the comparison between GS and the best NN.

Using the model trained with 5×5 arrays, we found that larger tweezer arrays can also be inferred with reasonable performance. Figure 4 shows the RMSE of the inferred tweezer array depths for 5×5 , 10×10 , 15×15 , and 20×20 . Up to the 15×15 array, the RMSE obtained by the proposed model remains comparable to that of the GS algorithm with five iterations, even without additional retraining. For even larger arrays, the evolutionary search can be warm-started from the previously optimized architecture rather than restarting from scratch, enabling efficient adaptation to new experimental requirements. Because the input is represented on a fixed 1024×1024 grid regardless of the array size and the phase is produced in a single forward pass, the inference time is essentially independent of the number of tweezers, in contrast to iterative methods whose cost grows with array size. We confirmed this empirically: relative to the 5×5 case, the measured inference time increased by less than about 2% for 10×10 , 15×15 , and 20×20 arrays.

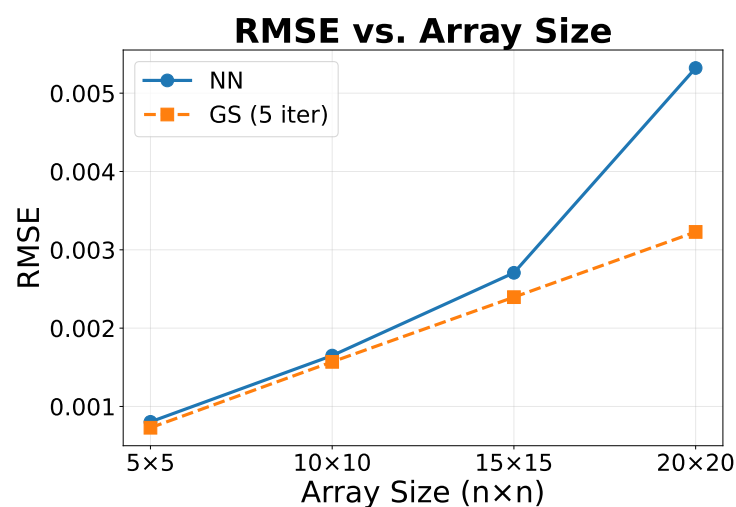


Figure 4. Phase prediction results for varying array sizes: 5×5 , 10×10 , 15×15 , and 20×20 . Although the model is trained on 5×5 arrays, the RMSE of the inferred tweezer array depths remains comparable to that of the GS algorithm up to 15×15 arrays.

We further perform an experimental validation of the holograms predicted by the model using the optical setup shown in Figure 5. A home-built external-cavity diode laser operating at a wavelength of 815 nm is incident on a spatial light modulator (SLM, HSP1K-500-1200-PC8, Meadowlark Optics, Frederick, CO, USA), which displays the predicted hologram. The setup is designed for our near-term experiments on the trapping and rearranging of single ^{87}Rb atoms in optical tweezer arrays. After the SLM, the modulated beam passes through a telescope consisting of lenses L1 ($f = 750$ mm) and L2 ($f = 200$ mm), which reduces the beam waist to match the pupil of the objective lens L3 (G Plan Apo 50X, Mitutoyo, Kawasaki, Japan). The optical tweezer focal plane is formed inside a vacuum chamber after the objective lens. For trap characterization, an insertable mirror redirects the beam to a CCD camera, allowing direct imaging of the generated tweezer arrays.

The inset of Figure 5 shows a representative CCD image obtained using a hologram predicted by the optimized architecture for an intermediate stage of the rearrangement process. The displayed hologram corresponds to the seventh frame of a 15-frame trajectory for a 5×5 tweezer array and was directly uploaded to the SLM without additional optimization. The resulting tweezer pattern is clearly observed in the experimental setup, confirming that the predicted holograms can be implemented on a physical optical system and generate the intended trap geometry. This experimental validation is intended as a proof of principle that the predicted holograms can be realized on a physical optical system and produce the intended trap geometry. A full evaluation under rearrangement conditions, including quantitative trap-uniformity measurements across a frame sequence, frame-to-frame intensity flicker and sequence stability, and atom-level trapping and transport fidelity, is left for future work on the target atomic platform.

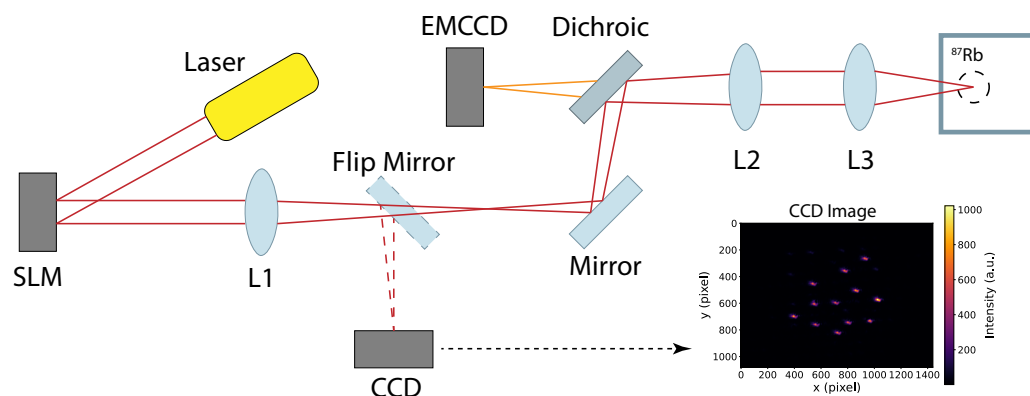


Figure 5. Experimental implementation of the predicted holograms. (a) Optical tweezer setup used for validation. An 815 nm laser beam is modulated by an SLM and focused by the objective lens to generate optical tweezers. An insertable mirror redirects the beam to a CCD camera for imaging. (b) Representative CCD image obtained using a hologram predicted by the optimized architecture for an intermediate frame of a rearrangement sequence. (Inset: CCD camera image captured during atom rearrangement (frame 7). The trap positions generated by the predicted phase hologram are clearly resolved.)

5. Discussions

To investigate the role of the FT and IFT layers in the proposed architecture, we compare the optimized model with a stacked architecture that does not include any Fourier-domain operations. As illustrated in Figure 6a, the stacked model consists of neural network layers connected sequentially, while retaining the same set of candidate layer types used in the evolutionary search. Both architectures were searched under the same procedure, using identical candidate layer types, the same RMSE-based selection, and the same inference-time constraint. The only difference between them is the presence of the embedded FT and IFT operations in the search space.

Figure 6b compares the evolution of the RMSE over successive generations for the two search spaces. Although the stacked model exhibits a gradual reduction in RMSE, its final RMSE remains significantly higher than that of the proposed FT/IFT-embedded architecture. Because the two searches differ only in the presence of the Fourier-domain operations, the gap is attributable to the embedded FT and IFT structure itself: architectures without these operations must learn the non-local mapping implicitly, and converge to worse solutions under otherwise identical conditions. These results indicate that embedding Fourier-domain structure into the searched architecture, rather than requiring the network to learn it from data, is important for holographic phase prediction.

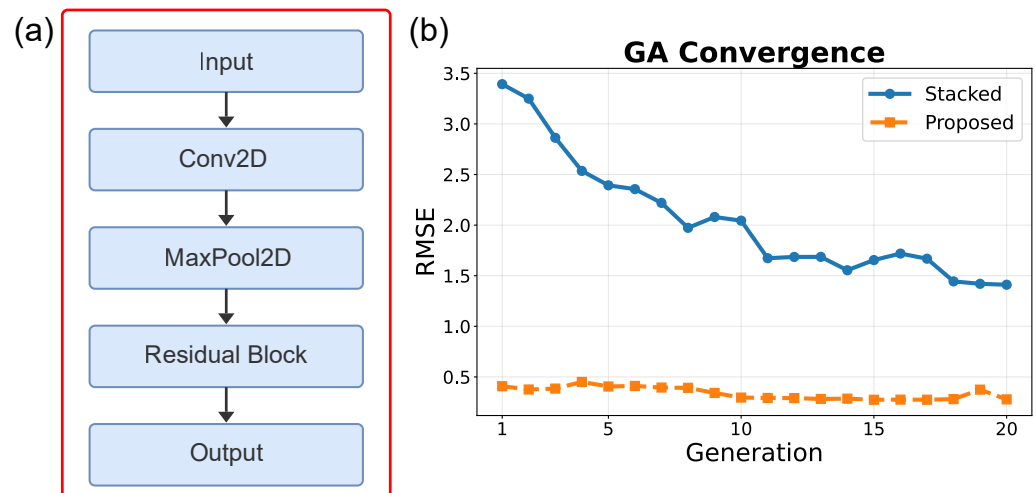


Figure 6. (a) Example architecture of the stacked model used for comparison. A single gene forms one individual, with layers connected sequentially without FT or IFT operations. (b) RMSE over generations for the proposed FT/IFT-embedded architecture (Proposed) and a purely stacked model (Stacked). Both were searched under identical conditions; the stacked model converges to a significantly higher RMSE, reflecting the benefit of embedding Fourier-domain structure into the searched architecture.

Several directions remain for future investigation. First, the current study employs a relatively small training dataset consisting of 128 samples. Expanding the dataset may further improve the phase prediction quality and enable the optimization of more expressive architectures. Second, the present optimization framework considers only the reconstructed intensity distribution and inference time. During practical atom rearrangement, however, intermediate holograms may induce intensity flickering due to transient phase variations. Incorporating flicker suppression as an additional optimization objective could further enhance the robustness of holographic tweezer transport.

6. Conclusions

We have presented an evolutionary optimization framework for neural architectures in holographic phase prediction for optical tweezer arrays. The framework combines FT/IFT-embedded network structures with a genetic algorithm that optimizes the layer composition and connectivity of the architecture to balance phase prediction quality and inference speed. The optimized architecture achieves phase prediction quality comparable to several iterations of the GS algorithm while substantially reducing the inference time. Experimental implementation demonstrates that the predicted holograms can be directly displayed on an SLM to generate the intended optical tweezer patterns.

The proposed framework provides a systematic approach for designing neural architectures tailored to holographic phase prediction in optical tweezer systems. By combining Fourier-domain operations with evolutionary architecture optimization, the framework en-

ables efficient hologram generation while maintaining competitive phase prediction quality. As optical tweezer arrays continue to scale in size and complexity, such learning-based approaches may become valuable for real-time hologram generation and atom rearrangement. Taken together, these results constitute a proof of concept for Fourier-embedded architecture search in holographic phase prediction: they establish the methodology and the physical realizability of its output, while a full quantitative and atom-level evaluation remains for future work. Rapid, defect-free single-atom array assembly for neutral-atom quantum computation and quantum simulation remains the longer-term motivation for developing such methods, though bridging the present proof-of-concept to that setting will require further work on larger arrays and experimental atom-level validation.

Author Contributions: Conceptualization, J.C. and M.K.; methodology, J.C.; software, J.C.; validation, J.C. and M.K.; formal analysis, J.C. and M.K.; investigation, J.C. and M.K.; resources, J.C. and M.K.; data curation, J.C.; writing—original draft preparation, J.C. and M.K.; writing—review and editing, M.K.; visualization, J.C. and M.K.; supervision, M.K.; project administration, M.K.; funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Research Foundation of Korea (NRF) under Grant No. RS-2024-00337653, funded by the Korea government (MSIT).

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

GS	Gerchberg–Saxton
SLM	Spatial light modulator
AOD	Acousto-optic deflector
CNNs	Convolutional neural networks
GS-RNN	Gerchberg–Saxton recurrent neural network
FT	Fourier transform
IFT	Inverse Fourier transform
GA	Genetic algorithm
RMSE	Root-mean-square error (RMSE)
WGS	Weighted GS

References

1. Saffman, M. Quantum computing with atomic qubits and Rydberg interactions: Progress and challenges. *J. Phys. B At. Mol. Opt. Phys.* **2016**, *49*, 202001.
2. Browaeys, A.; Lahaye, T. Many-body physics with individually controlled Rydberg atoms. *Nat. Phys.* **2020**, *16*, 132–142.
3. Morgado, M.; Whitlock, S. Quantum simulation and computing with Rydberg-interacting qubits. *AVS Quantum Sci.* **2021**, *3*, 023501.
4. Kim, M.; Ahn, J.; Song, Y.; Moon, J.; Jeong, H. Quantum computing with Rydberg atom graphs. *J. Korean Phys. Soc.* **2023**, *82*, 827–840.
5. Grimm, R.; Weidemüller, M.; Ovchinnikov, Y.B. Optical dipole traps for neutral atoms. *Adv. At. Mol. Opt. Phys.* **2000**, *42*, 95–170.
6. Schlosser, N.; Reymond, G.; Protsenko, I.; Grangier, P. Sub-poissonian loading of single atoms in a microscopic dipole trap. *Nature* **2001**, *411*, 1024–1027.
7. Schlosser, N.; Reymond, G.; Grangier, P. Collisional blockade in microscopic optical dipole traps. *Phys. Rev. Lett.* **2002**, *89*, 023005.

8. Nogrette, F.; Labuhn, H.; Ravets, S.; Barredo, D.; Béguin, L.; Vernier, A.; Lahaye, T.; Browaeys, A. Single-atom trapping in holographic 2D arrays of microtraps with arbitrary geometries. *Phys. Rev. X* **2014**, *4*, 021034.
9. Barredo, D.; de Léséleuc, S.; Lienhard, V.; Lahaye, T.; Browaeys, A. An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays. *Science* **2016**, *354*, 1021.
10. Barredo, D.; Lienhard, V.; de Léséleuc, S.; Lahaye, T.; Browaeys, A. Synthetic three-dimensional atomic structures assembled atom by atom. *Nature* **2018**, *561*, 79–82.
11. Pause, L.; Sturm, L.; Mittenbühler, M.; Amann, S.; Preuschoff, T.; Schäffner, D.; Schlosser, M.; Birkel, G. Supercharged two-dimensional tweezer array with more than 1000 atomic qubits. *Optica* **2024**, *11*, 222–226.
12. Manetsch, H.J.; Nomura, G.; Bataille, E.; Lv, X.; Leung, K.H.; Endres, M. A tweezer array with 6100 highly coherent atomic qubits. *Nature* **2025**, *647*, 60–67.
13. Endres, M.; Bernien, H.; Keesling, A.; Levine, H.; Anschuetz, E.R.; Krajenbrink, A.; Senko, C.; Vuletić, V.; Greiner, M.; Lukin, M.D. Atom-by-atom assembly of defect-free one-dimensional cold atom arrays. *Science* **2016**, *354*, 1024–1027.
14. Kim, H.; Lee, W.; Lee, H.-g.; Jo, H.; Song, Y.; Ahn, J. In situ single-atom array synthesis using dynamic holographic optical tweezers. *Nat. Commun.* **2016**, *7*, 13317.
15. Lee, W.; Kim, H.; Ahn, J. Defect-free atomic array formation using the Hungarian matching algorithm. *Phys. Rev. A* **2017**, *95*, 053424.
16. Kim, H.; Kim, M.; Lee, W.; Ahn, J. Gerchberg-Saxton algorithm for fast and efficient atom rearrangement in optical tweezer traps. *Opt. Express* **2019**, *27*, 2184–2196.
17. Schymik, K.-N.; Lienhard, V.; Barredo, D.; Scholl, P.; Williams, H.; Browaeys, A.; Lahaye, T. Enhanced atom-by-atom assembly of arbitrary tweezer arrays. *Phys. Rev. A* **2020**, *102*, 063107.
18. Lin, R.; Zhong, H.-S.; Li, Y.; Zhao, Z.-R.; Zheng, L.-T.; Hu, T.-R.; Wu, H.-M.; Wu, Z.; Ma, W.-J.; Gao, Y.; et al. AI-enabled parallel assembly of thousands of defect-free neutral atom arrays. *Phys. Rev. Lett.* **2025**, *135*, 060602.
19. Bluvstein, D.; Levine, H.; Semeghini, G.; Wang, T.T.; Ebadi, S.; Kalinowski, M.; Keesling, A.; Maskara, N.; Pichler, H.; Greiner, M.; et al. A quantum processor based on coherent transport of entangled atom arrays. *Nature* **2022**, *604*, 451–456.
20. Bluvstein, D.; Evered, S.J.; Geim, A.A.; Li, S.H.; Zhou, H.; Manovitz, T.; Ebadi, S.; Cain, M.; Kalinowski, M.; Hangleiter, D.; et al. Logical quantum processor based on reconfigurable atom arrays. *Nature* **2024**, *626*, 58–65.
21. Bluvstein, D.; Geim, A.A.; Li, S.H.; Evered, S.J.; Bonilla, Ataide, J.P.; Baranes, G.; Gu, A.; Manovitz, T.; Xu, M.; Kalinowski, M.; et al. A fault-tolerant neutral-atom architecture for universal quantum computation. *Nature* **2026**, *649*, 39–46.
22. Knottnerus, I.H.A.; Tseng, Y.C.; Urech, A.; Spreeuw, R.J.C.; Schreck, F. Parallel assembly of neutral atom arrays with an SLM using linear phase interpolation. *SciPost Phys.* **2025**, *19*, 118.
23. Elsken, T.; Metzen, J. H.; Hutter, F. Neural architecture search: A survey. *J. Mach. Learn. Res.* **2019**, *20*, 1–21.
24. Xie, L.; Yuille, A. Genetic CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1379–1388.
25. Real, E.; Aggarwal, A.; Huang, Y.; Le Q.V. Regularized evolution for image classifier architecture search. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 4780–4789.
26. Horisaki, R.; Takagi, R.; Tanida, J. Deep-learning-generated holography. *Appl. Opt.* **2018**, *57*, 3859–3863.
27. Eybposh, M.H.; Cairra, N.W.; Atisa, M.; Chakravarthula, P.; Pégard, N.C. DeepCGH: 3D computer-generated holography using deep learning. *Opt. Express* **2020**, *28*, 26636–26650.
28. Hu, X.; Gui, J.; Hu, L.; Tong, Z.; Li, K. GS-RNN: A phase-only hologram generation model based on the fusion of the Gerchberg-Saxton algorithm and a neural network. *Opt. Express* **2025**, *33*, 19464–19478.
29. Di Leonardo, R.; Ianni, F.; Ruocco, G. Computer generation of optimal holograms for optical trap arrays. *Opt. Express* **2007**, *15*, 1913–1922.

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.