COMPRESSING CLASSICAL NEURAL NETWORK WITH PHOTONIC QUANTUM-TRAIN

Chen-Yu Liu[1] **Kuan-Cheng Chen**[2]
[1] National Taiwan University, Taipei, Taiwan
[2] Imperial College London, London, UK

ABSTRACT

This study introduces the photonic Quantum-Train (photonic QT) framework, an extension of the Quantum-Train methodology tailored for photonic quantum computing. photonic QT utilizes parameterized photonic quantum gates and a tensor network mapping model to generate parameters for classical neural networks (NNs) efficiently. By leveraging distributed photonic quantum neural networks (QNNs), the framework eliminates the need for quantum hardware during inference, enhancing its practicality. Empirical results demonstrate that photonic QT achieves competitive testing accuracy with significantly fewer parameters compared to classical models. For example, photonic QT with a bond dimension of 10 achieves $95.502 \pm 0.84\%$ accuracy using 3292 parameters, while the classical model requires 6690 parameters for $96.890 \pm 0.31\%$. Even at a bond dimension of 4, photonic QT maintains strong performance ($93.292 \pm 0.62\%$) with only 688 parameters. The framework outperforms classical compression techniques such as weight sharing and pruning, showcasing its scalability and efficiency. photonic QT highlights the potential of photonic quantum computing for parameter-efficient neural network training and broader machine learning applications.

1 INTRODUCTION

Breaking down complex problems into components tailored for either classical or quantum computing enables a more efficient approach to problem-solving. Classical systems excel at tasks such as data processing, while quantum computing demonstrates significant promise in optimization and navigating expansive state spaces. The concept of quantum-centric supercomputing (Bravyi et al., 2022; Gambetta, 2022) leverages the strengths of both paradigms, providing scalable solutions to tackle challenging problems. This hybrid strategy is particularly promising in the realm of quantum machine learning (QML), where it could play a pivotal role in the training and fine-tuning of large-scale models.

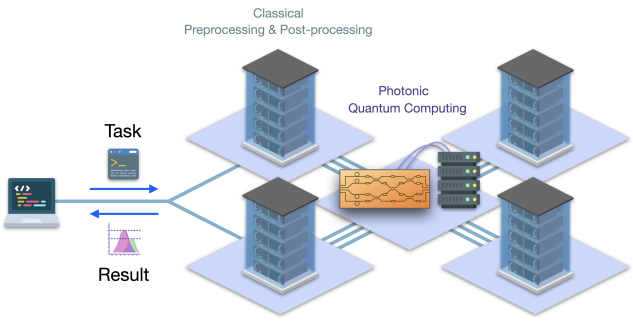

Figure 1: Quantum-centric supercomputing scheme.

Traditional QML approaches leverage parameterized quantum circuits (PQCs) as quantum neural networks (QNNs) (Chen et al., 2020), with data being introduced through specialized encoding techniques (Pérez-Salinas et al., 2020; Schuld et al., 2021). In this hybrid quantum-classical framework, QNN parameters are updated using computations performed on classical hardware (Mari et al., 2020). As this approach scales, quantum-centric supercomputing presents a transformative paradigm, especially relevant in the Noisy Intermediate-Scale Quantum (NISQ) era (Preskill, 2018). Furthermore, QNNs can be integrated with classical neural networks (NNs) as pre-processing or post-processing layers (Mari et al., 2020; Liu et al., 2021).

While theoretical and experimental findings highlight QML's potential to deliver advantages in specific domains (Cerezo et al., 2022; Huang et al., 2022; Biamonte et al., 2017; Caro et al., 2022; Huang et al., 2021),

key hurdles remain. One major issue is data encoding for large datasets, where methods like gate-angle encoding may produce quantum circuits that are either excessively deep or demand an impractical number of qubits. Another significant limitation is the dependence on quantum hardware during the inference phase, which poses challenges to the practicality of QML applications.

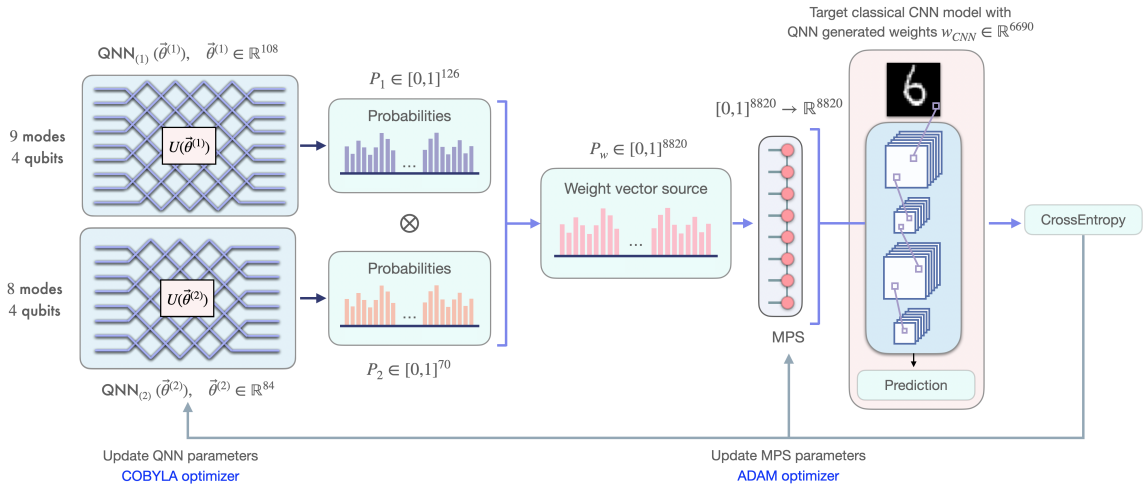

Figure 2: Overview of the photonic quantum-train scheme.

A promising approach to address both the data encoding challenge and the reliance on quantum hardware during inference is the Quantum-Train (QT) framework (Liu et al., 2024b). Instead of relying on a quantum neural network (QNN) to process data directly, this method employs QNNs to generate the weights of a target classical neural network (NN) during the training phase. By adopting this quantum-driven parameter generation technique, the data input process is kept entirely within the classical model, ensuring simplicity and efficiency. Furthermore, this approach eliminates the need for quantum computing hardware during inference, as the resulting trained model operates purely within a classical framework. Additionally, the number of training parameters is significantly reduced on a *polylogarithmic* scale by mapping the quantum state basis to the parameters of the target NN, provided that the QNN is constructed with a polynomial number of layers relative to the number of qubits. Further details of this methodology are presented in the following sections.

Recent studies (Liu et al., 2024d; Lin et al., 2024) on QT have primarily focused on gate-based quantum computing architectures, particularly those utilizing superconducting qubits, trapped ion qubits, or silicon-based qubits. These systems typically require extreme environmental conditions, such as near absolute zero temperatures or vacuum environments, to minimize noise and maintain stability. In contrast, photon-based quantum computers are significantly more resilient to environmental fluctuations and less susceptible to interference. They can operate at room temperature without the need for complex cooling mechanisms. Given these practical advantages, it is worth exploring whether the QT framework can be extended to the photonic quantum computing paradigm for broader and more accessible real-world applications.

In the Quandela Hybrid AI Quantum Challenge 2025, we propose the Photonic Quantum-Train (photonic QT), which extends the Quantum-Train (QT) framework to the photonic quantum computing paradigm. By leveraging parameterized photonic gates, the approach generates the weights for a target convolutional neural network (CNN) using two distributed photonic quantum neural networks (QNNs). A tensor network mapping model is employed to transform the resulting quantum probabilities from the range $[0, 1]$ to $\mathbb{R}$. An overview of the general scheme is presented in Fig. 2, with further details provided in the subsequent sections.

The main contributions of photonic QT in the Quandela Hybrid AI Quantum Challenge 2025 include:

- **Efficient classical model compression**: photonic QT utilizes quantum parameters from the Photonic QNN, which scale in proportion to the number of qubits N and modes M, to generate parameters that scale with the product Hilbert space size. When applied to generating parameters in

classical NN models, this property enables highly efficient compression by generating parameters through quantum methods.

- **Practical application of Photonic-QML during inference stage**: By incorporating quantum parameter generation, our approach to combining QML with LLMs is more practical compared to conventional QML methods (i.e., Quantum Transformer (Di Sipio et al., 2022)). The data-encoding issues associated with conventional QML (Yang et al., 2021) have been eliminated, and the inference stage no longer requires quantum hardware. This is particularly important for tasks that demand short response times, as reliance on remote quantum computers could introduce delays due to queuing and increased costs.
- **Flexibility, applicability, and efficiency in general machine learning**: While this study focuses solely on the MNIST classification task, the broader series of QT research has demonstrated that the QT framework can be applied to a wide range of machine learning domains. These include long short-term memory (LSTM) models (Lin et al., 2024), reinforcement learning (RL) (Liu et al., 2024d), federated learning (FL) (Liu & Chen, 2024), programming other parameterized quantum circuits (PQCs) (Liu et al., 2024a), and even fine-tuning large language models of up to 2 billion parameters (Liu et al., 2024e). This photonic QT study serves as a compelling example, illustrating how easily the photonic QT approach can be extended to various machine learning applications. This highlights the significant potential and practicality of QML in the photonic quantum computing paradigm.

2 Related Works

Parameter-Efficient Training Methods. Before presenting how the photonic QT approach demonstrates its effectiveness to reduce the number of training parameters, it is important to first acknowledge the classical methods for compressing NN models. On the classical side, a variety of techniques have been developed to compress NNs during training (Neill, 2020). These include network pruning (Blalock et al., 2020), weight sharing (Nowlan & Hinton, 2018), low-rank matrix and tensor decomposition (Kolda & Bader, 2009), knowledge distillation (Gou et al., 2021), and quantization (Lin et al., 2016).

To ensure a fair comparison with the proposed photonic QT method, we focused on techniques that reduce the number of trainable parameters during the training phase without relying on pre-trained models. Based on these criteria, weight sharing and pruning were selected, as they are both widely recognized and well-established approaches for model compression. Further details and comparative results are provided in the subsequent sections.

Training Classical Neural Networks via Quantum Computing. Leveraging quantum computing to train classical NNs presents a promising approach where the resulting models remain entirely classical, effectively overcoming challenges related to data encoding and dependence on quantum hardware during inference. In addition to QT, discussed in the previous and subsequent sections, other approaches have also been explored. For instance, one method utilizes quantum walks as a search process to optimize the parameters of classical NNs (de Souza et al., 2021). Another involves the use of a quantum hypernetwork to train a classical binary NN (Carrasquilla et al., 2023). However, the practical utility of the latter approach is limited, as it can only train binary NNs, restricting its applicability to more general neural network architectures.

3 Photonic Quantum-Train

At the core of QT, we hypothesize that utilizing a PQC-based QNN in conjunction with a mapping model as a parameter generator leverages the efficiency of quantum systems. Specifically, only a small number of QNN parameters are needed to control the measurement probabilities (or their product in a distributed scheme), which are determined by the dimensionality of the Hilbert space. In essence, we propose that the high-dimensional Hilbert space provides an efficient representation for constructing the weights of classical neural networks.

We propose a parameter generation process that takes a different approach from conventional QML. Consider a target NN model with parameters $w_{\text{CNN}} = (w_1, w_2, \ldots, w_m)$, where m is the number of parameters. We construct two photonic QNNs. The first QNN, denoted as $\text{QNN}_{(1)}(\vec{\theta}^{(1)})$, is configured with M_1 modes and N_1 qubits, such that the number of distinct outcome measurement probabilities is $C(M_1, N_1) = \frac{M_1!}{N_1!(M_1-N_1)!}$. Similarly, the second QNN, $\text{QNN}_{(2)}(\vec{\theta}^{(2)})$, is constructed with M_2 modes

and N_2 qubits, yielding a probability space of size $C(M_2, N_2)$. The values of M_1, N_1, M_2, and N_2 are chosen to satisfy the condition:

$$C(M_1, N_1) \times C(M_2, N_2) \geq m. \tag{1}$$

which means that the number of controlling parameters in the QNNs is significantly smaller than the number of parameters in the target NN.

Based on this design, the probabilities generated by the first QNN, denoted as $P_1 \in [0,1]^{C(M_1,N_1)}$, and those from the second QNN, $P_2 \in [0,1]^{C(M_2,N_2)}$, are combined to construct P_w as follows:

$$P_1 \otimes P_2 = P_w \in [0,1]^{C(M_1,N_1)\times C(M_2,N_2)}. \tag{2}$$

At this stage, controlling only $\frac{3M_1(M_1-1)}{2} + \frac{3M_2(M_2-1)}{2}$ parameters in the QNNs allows us to generate $C(M_1, N_1) \times C(M_2, N_2) \geq m$ parameters, which are then used to populate the target NN model. However, since the generated parameters are in the range $[0, 1]$, and the weights of the target NN typically lie in $\mathbb{R}$, a "mapping" model is required to perform the map $[0, 1] \to \mathbb{R}$. Following the method described in (Liu et al., 2024c), we employ a matrix product state (MPS) mapping model, $G_{\boldsymbol{b}}$, parameterized by $\boldsymbol{b}$, to perform this transformation. The target NN weights w_{CNN} are then derived as:

$$w_{\text{CNN}} \subset G_{\boldsymbol{b}}(P_w) \in \mathbb{R}^{C(M_1,N_1)\times C(M_2,N_2)}. \tag{3}$$

The inclusion relationship arises from the fact that $w_{\text{CNN}} \in \mathbb{R}^m$ and $C(M_1, N_1) \times C(M_2, N_2) \geq m$, meaning some of the generated parameters are discarded once m parameters are filled using $G_{\boldsymbol{b}}(P_w)$. By tuning $\vec{\theta}^{(1)}, \vec{\theta}^{(2)}$ and $\boldsymbol{b}$, we effectively influence the value of the loss function $\mathcal{L}$, which is evaluated by the target NN for a given task.

Gradient Estimation of Photonic Quantum Circuit Compressed Parameters. The target NN parameters w_{CNN} are generated through the use of QNNs coupled with a mapping model. The quantum-dependent parameters, denoted as $(\vec{\theta}^{(i)}, \boldsymbol{b})$, influence the target NN parameters through the quantum state preparation and measurement steps. The gradient of the loss function, which captures the effect of the quantum parameters, is expressed as:

$$\nabla_{\vec{\theta}^{(i)},\boldsymbol{b}}\mathcal{L} = \left(\frac{\partial w_{\text{CNN}}}{\partial(\vec{\theta}^{(i)}, \boldsymbol{b})}\right)^T \cdot \nabla_{w_{\text{CNN}}}\mathcal{L}. \tag{4}$$

Here, $\frac{\partial w_{\text{CNN}}}{\partial(\vec{\theta}^{(i)},\boldsymbol{b})}$ represents the Jacobian matrix, which quantifies the sensitivity of the classical parameters w_{CNN} to variations in the quantum parameters $(\vec{\theta}^{(i)}, \boldsymbol{b})$.

Parameter Update of Photonic Quantum Circuit Compressed Parameters. The learning rate η is a critical factor, particularly given the complex dynamics introduced by the quantum-classical interface. The update rule for the quantum parameters is defined as:

$$\vec{\theta}^{(i)}_{t+1}, \boldsymbol{b}_{t+1} = \vec{\theta}^{(i)}_t, \boldsymbol{b}_t + \eta\nabla_{\vec{\theta}^{(i)},\boldsymbol{b}}\mathcal{L}. \tag{5}$$

This update ensures that the quantum parameters are optimized to improve the performance of the target NN. The equation provides a high-level representation of the gradient update in an exact quantum state simulation. However, in practical applications using real quantum hardware or specific backend providers, the gradient calculation must incorporate the parameter shift rule and its variants (Mitarai et al., 2018; Schuld et al., 2019).

In our implementation, the mapping model parameters are updated using the ADAM optimizer, while the QNN parameters are updated using the COBYLA optimizer. An overview of the photonic QT framework is provided in Fig. 2, with detailed parameter settings described in the subsequent section.

Table 1: Configuration of the mapping model $G_{\boldsymbol{b}}$.

Hyperparameter	**Meaning**	**Value**
Input size	Input of the mapping model ($\|\phi_i\rangle, |\langle\phi_i\|\psi(\vec{\theta}^{(i)})\rangle|^2$) (Liu et al., 2024c)	$\lceil \log_2 m \rceil + 1$
Bond dimension	Main structure parameter of the MPS mapping model	$1 \sim 10$

4 Empirical Experiments

4.1 Decomposition

In our experiment, we adopt Clements' decomposition to our photonic QNN setup. We employ a multi-mode interferometer architecture comprising phase shifters (PSs) and balanced beam splitters (BSs) arranged in a rectangular mesh to systematically implement a desired unitary transformation. This strategy follows the prescription of Clements *et al.* (Clements et al., 2016), which provides an efficient, fully parameterized decomposition of an arbitrary $m \times m$ unitary matrix U. Formally, any such U can be factorized into a product of two-mode operations:

$$U = \prod_{l=1}^{L} B_{\theta_l, \phi_l}, \tag{6}$$

where B_{θ_l,ϕ_l} denotes a two-mode beam-splitter operation specified by the real parameters θ_l (transmittance or reflectance) and ϕ_l (phase shift). Each beam splitter is followed by an additional on-mode phase shifter to fine-tune the local phases. The total number of layers L scales on the order of $m(m-1)/2$, guaranteeing that *any* target transformation is realizable for a given device size.

Figure X (conceptual) illustrates our experimental workflow. We begin by initializing m photonic input modes in the quantum states required by the QNN algorithm (e.g., single photons in each mode for boson sampling). These inputs then propagate through a sequence of alternating BS and PS elements arranged in a checkerboard pattern. In each layer, we assign a set of parameters $\{\theta_l, \phi_l\}$ that control the beam-splitter transmission coefficients and the additional local phases. Mathematically, if $\hat{a}_i^\dagger$ and $\hat{a}_j^\dagger$ are the creation operators for two modes being mixed, a single BS operation at layer l transforms them as

$$\begin{pmatrix} \hat{a}_i^\dagger \\ \hat{a}_j^\dagger \end{pmatrix} \longrightarrow \begin{pmatrix} \cos\theta_l & -e^{-i\phi_l}\sin\theta_l \\ e^{i\phi_l}\sin\theta_l & \cos\theta_l \end{pmatrix} \begin{pmatrix} \hat{a}_i^\dagger \\ \hat{a}_j^\dagger \end{pmatrix}. \tag{7}$$

Subsequent phase shifts are applied independently to each output mode. Repeating this process across all layers in the rectangular mesh enables an *in situ* realization of the global unitary U, thereby permitting arbitrary multi-mode transformations essential for QNN training.

Our experimental setup (adapted from the Perceval library (Heurtel et al., 2023)) programmatically assigns and updates these parameters. The user can either supply explicit lists of $\{\theta_l\}$ and $\{\phi_l\}$ or allow the code to initialize them randomly. In practice, phase settings are dynamically tuned using thermo-optic or electro-optic modulators integrated onto photonic chips. This adaptability supports real-time, closed-loop optimization of BS and PS parameters to minimize a specified cost function, thus facilitating gradient-based training of photonic QNNs.

This photonic platform can implement a wide range of quantum algorithms, including boson sampling, quantum-enhanced machine learning, and other protocols benefiting from low-noise multi-mode interference. By leveraging the Clements decomposition, we ensure a systematic and hardware-friendly architecture with minimal optical depth and robust reconfigurability for diverse QNN tasks in the following section.

4.2 Result

In the implementation to address the Quandela challenge of the classfication task of partial MNIST dataset, we consider a target classical CNN with $m = 6690$ parameters. Following the above scheme, the settings of modes and qubits are $M_1 = 9, N_1 = 4, M_2 = 8$, and $N_2 = 4$, such that using $108 + 84 = 192$ QNN parameters, we can construct $C(M_1, N_1) \times C(M_2, N_2) = 8820$ parameters and using 6690 of them to construct the weights of the classical CNN during training. In addition to these QNN parameters, the parameters of the MPS mapping model should also be accounted, controlled by the bond dimension of the MPS, where we investigate the bond dimension from 1 to 10 in the below experiments.

Fig. 3 illustrates the training loss and accuracy over 200 epochs for different bond dimensions (bond_dim) in the MPS mapping model. The left panel shows the training loss, which decreases consistently as the number of epochs increases for all bond dimensions. Models with higher bond dimensions achieve lower final loss values, indicating improved optimization and representation capacity. The right panel depicts the training accuracy, which rises with the number of epochs and stabilizes as training progresses. Higher bond dimensions result in higher final training accuracy, further emphasizing their advantage in capturing complex

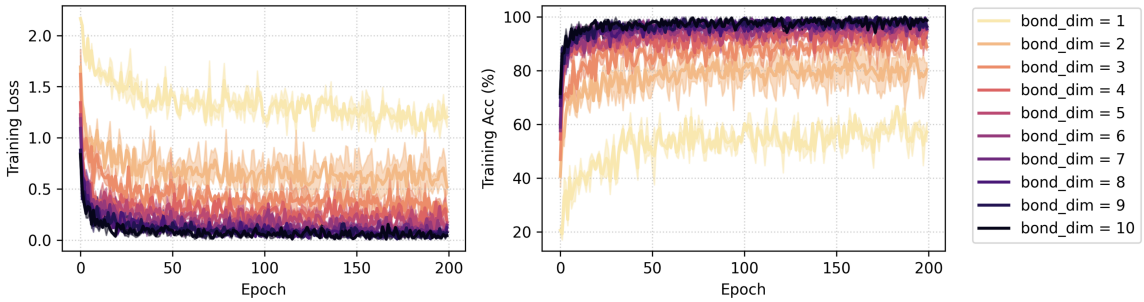

Figure 3: Training loss and accuracy.

patterns. These results demonstrate that increasing the bond dimension significantly enhances the model's performance by reducing training loss and improving accuracy.

Table 2 summarizes the performance of the original classical CNN. The model uses 6690 training parameters and achieves a training accuracy of $99.983 \pm 0.02\%$ and a testing accuracy of $96.890 \pm 0.31\%$, with a generalization error of 0.1690 ± 0.005. These results represent the baseline performance for comparison with the photonic QT approach. Table 3 presents the results of the photonic QT framework with varying bond dimensions in the MPS mapping model. As the bond dimension increases from 1 to 10, the number of training parameters grows from 223 to 3292. Correspondingly, the training and testing accuracies improve significantly. For example, with a bond dimension of 1, the training accuracy is $58.256 \pm 2.34\%$ and the testing accuracy is $55.775 \pm 3.27\%$. At a bond dimension of 10, these values increase to $98.990 \pm 0.34\%$ and $95.502 \pm 0.84\%$, respectively.

The generalization error also varies with the bond dimension. While smaller bond dimensions show lower generalization error (e.g., 0.0219 ± 0.007 for a bond dimension of 1), the error increases with larger bond dimensions, reaching 0.2552 ± 0.053 at a bond dimension of 10. This trade-off between model complexity and generalization highlights the role of bond dimension in balancing performance and overparameterization in the photonic QT framework.

Table 2: Result of the original classical CNN.

# of training parameters	**Training accuracy (%)**	**Testing accuracy (%)**	**Generalization error**
6690	99.983 ± 0.02	96.890 ± 0.31	0.1690 ± 0.005

Table 3: Results of photonic QT with different bond dimension settings of MPS mapping model.

Bond dimension	**# of training parameters**	**Training accuracy (%)**	**Testing accuracy (%)**	**Generalization error**
1	223	58.256 ± 2.34	55.775 ± 3.27	0.0219 ± 0.007
2	316	83.340 ± 2.77	81.375 ± 2.28	0.0462 ± 0.032
3	471	88.693 ± 1.67	87.057 ± 2.66	0.0364 ± 0.016
4	688	93.916 ± 0.45	93.292 ± 0.62	0.0679 ± 0.002
5	967	95.450 ± 0.39	93.042 ± 0.77	0.0950 ± 0.010
6	1308	96.953 ± 0.02	94.917 ± 0.60	0.1135 ± 0.013
7	1711	97.773 ± 0.22	94.957 ± 0.82	0.1315 ± 0.031
8	2176	97.866 ± 0.78	94.707 ± 0.47	0.1399 ± 0.007
9	2703	98.373 ± 0.12	94.835 ± 0.48	0.1624 ± 0.021
10	3292	98.990 ± 0.34	95.502 ± 0.84	0.2552 ± 0.053

Fig. 4 compares the performance of the photonic QT framework with classical compression baselines, specifically weight sharing and pruning, as well as the original classical target model. The left panel illustrates the

testing accuracy (%) as a function of the number of trainable parameters, while the right panel focuses on the generalization error.

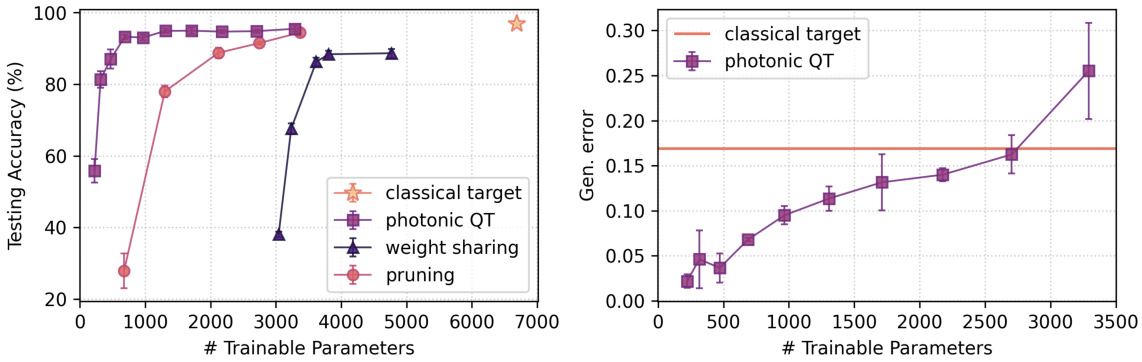

Figure 4: (Left) Testing accuracy of photonic QT and model compression baselines. (Right) Generalization error pf photonic QT and original classical CNN target model.

The left panel highlights the testing accuracy trends across different methods. The classical target model achieves the highest testing accuracy ($\sim 96\%$) but requires a significantly larger number of trainable parameters (6690), serving as a baseline for comparison. In contrast, the photonic QT framework demonstrates a clear improvement in testing accuracy as the number of trainable parameters increases. Remarkably, photonic QT achieves testing accuracy comparable to the classical target model while requiring far fewer parameters, showcasing its potential for parameter-efficient training.

The weight-sharing method, depicted as blue triangles, shows rapid gains in testing accuracy with increasing parameter counts. However, it plateaus at slightly lower accuracy levels compared to photonic QT, highlighting its limitations in fully capturing complex patterns. The pruning method, represented by red circles, starts with lower testing accuracy for smaller parameter counts but shows consistent improvement as the number of trainable parameters increases, eventually approaching the performance of photonic QT for larger models.

The right panel of Fig. 4 focuses on the generalization error. The classical target model with 6690 parameters shows a generalization error of 0.169. In comparison, the photonic QT framework exhibits a generalization error that increases as the number of trainable parameters grows. While photonic QT achieves competitive testing accuracy, the trade-off is an increasing generalization error for larger parameter counts, eventually surpassing that of the classical target model.

The Table 4 demonstrates that the photonic QT framework, particularly at lower bond dimensions, offers a highly parameter-efficient approach while maintaining strong testing accuracy, outperforming classical compression methods such as weight sharing and pruning. These results underscore the potential of the photonic QT framework to deliver high testing accuracy with a fraction of the parameters required by classical models. However, they also reveal a trade-off in terms of generalization, which warrants further investigation. Weight sharing and pruning serve as effective classical alternatives for parameter reduction, but their performance is limited compared to the photonic QT approach. This comparison highlights the unique advantages and challenges of photonic quantum approach for parameter-efficient NN training.

Table 4: # of trainable parameters with similar testing accuracy for comparison.

Method	**# of training parameters**	**Testing accuracy (%)**
Original	6690	96.890 ± 0.31
Weight sharing	4770	88.666 ± 1.207
Pruning	3370	94.443 ± 0.923
Photonic QT (bond dimension $= 10$)	3292	95.502 ± 0.84
Photonic QT (bond dimension $= 4$)	688	93.292 ± 0.62

5 CONCLUSION

The results of this study demonstrate the significant potential of the photonic QT framework in addressing the challenges of compressing classical NNs while maintaining competitive performance. By leveraging parameterized photonic quantum gates and a tensor network mapping model, the photonic QT framework effectively generates parameters for classical NNs in a highly efficient manner. This quantum-driven approach not only reduces the reliance on training models, as required in many classical compression techniques, but also eliminates the dependency on quantum hardware during the inference phase, making it more practical for real-world applications. Empirical evaluations reveal that photonic QT achieves testing accuracy comparable to the original classical target model with significantly fewer trainable parameters. For example, photonic QT with a bond dimension of 10 achieves $95.502 \pm 0.84\%$ testing accuracy with only 3292 parameters, compared to $96.890 \pm 0.31\%$ accuracy for the original model with 6690 parameters. Furthermore, even at a bond dimension of 4, photonic QT maintains a strong performance with $93.292 \pm 0.62\%$ accuracy using only 688 parameters, demonstrating its efficiency in parameter reduction.

Comparisons with classical compression baselines, such as weight sharing and pruning, further highlight the advantages of photonic QT. While classical methods are effective in reducing model size, they often fall short in terms of testing accuracy and flexibility. photonic QT, on the other hand, achieves a better balance between parameter efficiency and model performance, underscoring the benefits of integrating quantum-inspired techniques into machine learning workflows. Despite its promising results, the study also reveals a trade-off between model complexity and generalization in photonic QT. As the bond dimension increases, the generalization error grows, indicating a need for further optimization to balance overparameterization and generalization. Future research could explore enhancements to the tensor network mapping model or alternative quantum-classical integration strategies to mitigate this trade-off.

This study establishes the photonic QT framework as a viable and efficient approach for neural network compression, with broad applicability in various machine learning domains. Its ability to combine the strengths of quantum computing with the practicality of classical systems paves the way for new advancements in photonic QML and beyond.

REFERENCES

Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

Sergey Bravyi, Oliver Dial, Jay M Gambetta, Darío Gil, and Zaira Nazario. The future of quantum computing with superconducting qubits. *Journal of Applied Physics*, 132(16), 2022.

Matthias C Caro, Hsin-Yuan Huang, Marco Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J Coles. Generalization in quantum machine learning from few training data. *Nature communications*, 13(1):4919, 2022.

Juan Carrasquilla, Mohamed Hibat-Allah, Estelle Inack, Alireza Makhzani, Kirill Neklyudov, Graham W Taylor, and Giacomo Torlai. Quantum hypernetworks: Training binary neural networks in quantum superposition. *arXiv preprint arXiv:2301.08292*, 2023.

Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022.

Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan. Variational quantum circuits for deep reinforcement learning. *IEEE access*, 8:141007–141024, 2020.

William R Clements, Peter C Humphreys, Benjamin J Metcalf, W Steven Kolthammer, and Ian A Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 2016.

Luciano S de Souza, Jonathan HA de Carvalho, and Tiago AE Ferreira. Classical artificial neural network training using quantum walks as a search procedure. *IEEE Transactions on Computers*, 71(2):378–389, 2021.

Riccardo Di Sipio, Jia-Hong Huang, Samuel Yen-Chi Chen, Stefano Mangini, and Marcel Worring. The dawn of quantum natural language processing. In *Proc. ICASSP*, pp. 8612–8616. IEEE, 2022.

Jay Gambetta. Quantum-centric supercomputing: The next wave of computing. *IBM Research Blog*, 2022.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Nicolas Heurtel, Andreas Fyrillas, Grégoire De Gliniasty, Raphaël Le Bihan, Sébastien Malherbe, Marceau Pailhas, Eric Bertasi, Boris Bourdoncle, Pierre-Emmanuel Emeriau, Rawad Mezher, et al. Perceval: A software platform for discrete variable photonic quantum computing. *Quantum*, 7:931, 2023.

Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):2631, 2021.

Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, et al. Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186, 2022.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Chu-Hsuan Abraham Lin, Chen-Yu Liu, and Kuan-Cheng Chen. Quantum-train long short-term memory: Application on flood prediction problem. *arXiv preprint arXiv:2407.08617*, 2024.

Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy. Fixed point quantization of deep convolutional networks. In *International conference on machine learning*, pp. 2849–2858. PMLR, 2016.

Chen-Yu Liu and Samuel Yen-Chi Chen. Federated quantum-train with batched parameter generation. *arXiv preprint arXiv:2409.02763*, 2024.

Chen-Yu Liu, Samuel Yen-Chi Chen, Kuan-Cheng Chen, Wei-Jia Huang, and Yen-Jui Chang. Programming variational quantum circuits with quantum-train agent. *arXiv preprint arXiv:2412.01173*, 2024a.

Chen-Yu Liu, En-Jui Kuo, Chu-Hsuan Abraham Lin, Jason Gemsun Young, Yeong-Jar Chang, Min-Hsiu Hsieh, and Hsi-Sheng Goan. Quantum-train: Rethinking hybrid quantum-classical machine learning in the model compression perspective. *arXiv preprint arXiv:2405.11304*, 2024b.

Chen-Yu Liu, Chu-Hsuan Abraham Lin, and Kuan-Cheng Chen. Quantum-train with tensor network mapping model and distributed circuit ansatz. *arXiv preprint arXiv:2409.06992*, 2024c.

Chen-Yu Liu, Chu-Hsuan Abraham Lin, Chao-Han Huck Yang, Kuan-Cheng Chen, and Min-Hsiu Hsieh. Qtrl: Toward practical quantum reinforcement learning via quantum-train. *arXiv preprint arXiv:2407.06103*, 2024d.

Chen-Yu Liu, Chao-Han Huck Yang, Min-Hsiu Hsieh, and Hsi-Sheng Goan. A quantum circuit-based compression perspective for parameter-efficient learning. *arXiv preprint arXiv:2410.09846*, 2024e.

Junhua Liu, Kwan Hui Lim, Kristin L Wood, Wei Huang, Chu Guo, and He-Liang Huang. Hybrid quantum-classical convolutional neural networks. *Science China Physics, Mechanics & Astronomy*, 64(9):290311, 2021.

Andrea Mari, Thomas R Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. Transfer learning in hybrid classical-quantum neural networks. *Quantum*, 4:340, 2020.

Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

James O' Neill. An overview of neural network compression, 2020. URL https://arxiv.org/abs/2006.03669.

Steven J Nowlan and Geoffrey E Hinton. Simplifying neural networks by soft weight sharing. In *The mathematics of generalization*, pp. 373–394. CRC Press, 2018.

Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

Chao-Han Huck Yang, Jun Qi, Samuel Yen-Chi Chen, Pin-Yu Chen, Sabato Marco Siniscalchi, Xiaoli Ma, and Chin-Hui Lee. Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. In *Proc. ICASSP*, pp. 6523–6527. IEEE, 2021.

A Details of classical compression baselines

A.1 Weight sharing

Weight sharing is one of the simplest methods for network reduction, involving the reuse of weights either across layers or within structures of a layer. However, it is not always straightforward to determine which and how many weights can be shared before performance degradation becomes unacceptable for a specific network architecture and task (Neill, 2020). Some approaches address this by clustering similar weights and sharing their centroid among the members of each cluster or by using a weight penalty term in the objective function to encourage weights to group in ways that facilitate sharing. However, these techniques often require a pre-trained model to identify "similar" weights for clustering. In this study, we employ a simpler weight-sharing method by representing the weight matrix using repetitive rows, which avoids the need for pre-trained models.

Let $W \in \mathbb{R}^{m \times n}$ be the original weight matrix, where m is the number of input features, and n is the number of output features. Instead of learning an independent $m \times n$ matrix, we define a smaller set of shared vectors $\mathbf{v}_k \in \mathbb{R}^n$; where $k < m$, which are repeated across multiple rows of the weight matrix. Specifically, each row W_i, is assigned to one of the shared vectors by sequence. This allows us to control the number of trainable parameters by adjusting the number of distinct shared vectors $\mathbf{v}_k$.

By selecting an appropriate number of shared vectors, we precisely regulate the number of parameters, offering a controlled trade-off between model size and performance. As shown in Fig. 4, the testing accuracy results of the weight-sharing method are plotted as blue triangles. Each data point in the figure represents the average of 3 runs, demonstrating the performance consistency of the weight-sharing approach.

A.2 Pruning

The second method benchmarked against photonic QT is pruning, one of the most widely used techniques for reducing the number of parameters in neural networks. Traditionally, pruning is applied to a pre-trained model, where less important weights are removed to achieve a smaller, more efficient network. This approach not only reduces storage requirements but also speeds up model runtime, with performance often maintained through retraining of the pruned network.

However, to ensure a fair comparison with photonic QT, we do not rely on pre-trained models in this study. Instead, we apply random pruning by setting a predefined ratio of weights to be pruned within the target model. A portion of the model's weights is randomly removed based on this ratio. To account for variability introduced by the random selection process, we run each experiment three times and report the average performance for each data point. As illustrated in Fig. 4, the testing accuracy results of the pruning method are depicted as red circles. The plot highlights how the performance of the pruned models changes with varying amounts of pruning.