QIANets: Quantum-Integrated Adaptive Networks for Reduced Latency and Improved Inference Times in CNN Models

Zhumazhan Balapanov **Vanessa Matvei** **Olivia Holmberg** **Edward Magongo**
Jonathan Pei **Kevin Zhu**
Algoverse AI Research
kevin@algoverse.us

Abstract

Convolutional neural networks (CNNs) have made significant advances in computer vision tasks, yet their high inference times and latency often limit real-world applicability. While model compression techniques have gained popularity as solutions, they often overlook the *critical balance between low latency and uncompromised accuracy*. By harnessing **quantum-inspired pruning**, **tensor decomposition** and **annealing-based matrix factorization** – three quantum-inspired concepts – we introduce QIANets: a novel approach of redesigning the traditional GoogLeNet, DenseNet, and ResNet-18 model architectures to process more parameters and computations whilst maintaining low inference time. Code: https://github.com/quantum-inspired-model-compression

1 Introduction

The field of computer vision (CV) has recently experienced a substantial rise in interest [1]. This surge has created transformative advancements, driving the development of deep learning models, particularly within convolutional architecture, such as DenseNet [2], GoogLeNet [3], and ResNet-18 [4]. These methods have significantly optimized neural networks for image processing tasks, achieving state-of-the-art performance across multiple benchmarks [5]. However, the increasing computational complexity, memory consumption, and model size (millions to billions of parameters) pose substantial challenges for deployment, especially in time-sensitive and computationally-limited scenarios. The demand for *low-latency processing* in real-time applications, such as image processing and automated CV systems, is critical; compact models are needed for faster responses [6].

To address these issues, researchers have explored various optimization techniques to reduce inference times and latency while maintaining high accuracy. Model compression techniques such as pruning, quantization, and knowledge distillation have shown promise in enhancing model efficiency [7]. Yet, these methods often come with trade-offs that can impact model performance, necessitating a careful balance between energy efficiency and accuracy.

In recent years, the principles of quantum computing have emerged as an avenue for accelerating inference in machine learning [8]. Quantum-inspired methods, which leverage phenomena such as quantum optimization algorithms, strive to maintain model performance by reducing computational requirements, thereby offering significant speedups for certain tasks [9]. Meanwhile, traditional model compression techniques reduce the size of neural networks by removing less important weights, *sacrificing accuracy for lower latency* [10]. By integrating concepts from quantum mechanics into convolutional neural network (CNN) models, our approach seeks to address these limitations. We explore the potential of designing CNNs to balance improved inference times with minimal accuracy loss, creating a novel solution.

Preprint. Under review.

Within this context, we employ three key quantum-inspired principles: 1. quantum-inspired pruning: reducing model size by removing unnecessary parameters, guided by quantum approximation algorithms; 2. tensor decomposition: breaking down high-dimensional tensors into smaller components to reduce computational complexity; and 3. annealing-based matrix factorization: optimizing matrix factorization by using annealing techniques to find efficient representations of the data.

Our work addresses the following research question: **How can principles from quantum computing be used to design and optimize CNNs to reduce latency and improve inference times, while still maintaining stable accuracies across various models?**

In this paper, we propose a Quantum-Integrated Adaptive Networks (QIANets) – a comprehensive framework that *integrates* these quantum computing techniques into the DenseNet, GoogLeNet, and ResNet-18 architectures. To the best of our knowledge, this is the first attempt made to: 1) apply quantum computing-inspired algorithms into the models' architectures to reduce computational requirements and achieve efficient performance improvements, and 2) specifically target these models.

The contributions of this work include:

- QIANets: a comprehensive framework that integrates QAOA-inspired pruning, tensor decomposition and quantum annealing-inspired matrix factorization into three CNNs.
- An exploration of the trade-offs between latency, inference time, and accuracy, highlighting the effects of applying quantum principles to CNN models for real-time optimization.

2 Related Works

Our proposed method builds upon the ideas of model compression & quantum-inspired techniques to improve the inference times of CNNs.

2.1 Model Compression Techniques:

Pruning is one of the most effective ways to accelerate CNNs. Cheng et al. (2018) [11] reviewed model compression techniques for deep neural networks (DNNs), focusing on parameter pruning: removing individual weights based on importance to reduce model size while generally preserving performance.

Despite the advancements in parameter pruning, overall conventional pruning techniques have limitations: 1) high cost when applied *during* training and 2) the risk of prematurely removing important data. Hou et al. (2022) [12] introduced CHEX, a *training-based channel pruning* and regrowing method that reallocates channels across layers using a column subset selection (CSS) formulation, achieving significant compression without a fully pre-trained model.

2.2 Quantum-Inspired Techniques for CNNs:

Quantum computing is currently recognized as a potential game-changer for various fields, including NLP, due to its ability to process complex data more efficiently than classical computers.

Shi et al. (2021) [13] proposed a quantum-inspired architecture (QICNNs) with complex-valued weights to enhance CNN representational capacity, achieving higher accuracy and faster convergence on datasets than standard CNNs. In contrast, our methodology prioritizes structural optimization for greater computational efficiency, reducing latency and improving inference times through quantum techniques.

Hu et al. (2022) [14] set a high standard in the field by addressing quantum neural networks (QNNs) compression. Their CompVQC framework leverages an alternating direction method of multipliers (ADMM) approach, achieving a remarkable reduction in circuit depth by over 2.5× with less than 1% accuracy loss. While their results in QNN compression are impressive, our research introduces a novel first-attempt technique that applies QAOA-inspired pruning, tensor decomposition and quantum annealing-inspired matrix factorization to *classical CNNs*, potentially complementing their approach and enhancing model efficiency.

3 Methodology

3.1 Quantum-Inspired Pruning

We build upon the established technique of *pruning* to reduce the complexities of CNNs, as demonstrated in early studies ([15]; [16]; [17]). However, we introduce a new optimization way, utilizing the *Quantum Approximate Optimization Algorithm* (QAOA) [18] to frame pruning as a probabilistic optimization problem. For a neural network layer represented by weights as a tensor: $W \in \mathbb{R}^{C_{out} \times C_{in} \times H \times W}$, we define the importance of each weight using its absolute value:

$$I_{i,j} = |W_{i,j}| \tag{1}$$

To facilitate decision-making regarding weight retention, we normalize these importance scores with the softmax function:

$$P_{i,j} = \frac{e^{I_{i,j}}}{\sum_{k,l} e^{I_{k,l}}} \tag{2}$$

These probabilities are then used in a quantum-inspired decision-making process. Weights are pruned based on a threshold λ, influenced by a hyperparameter known as layer sparsity α:

$$R_{i,j} = \begin{cases} 1 & \text{if } P_{i,j} \geq \lambda \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here, $R_{i,j}$ serves as a binary retain mask indicating whether a weight is pruned (set to 0) or retained. The threshold λ is calibrated to ensure that approximately $100\alpha\%$ of the weights are pruned.

When implemented, we adopt an iterative approach across multiple stages, recalculating the retain mask based on updated probabilities. To enhance this process, we introduce a neighboring entanglement mechanism: when a weight is pruned, adjacent weights in the tensor may also be pruned with the probability P_{entangle}, simulating quantum entanglement and reflecting correlated behavior among nearby weights. For convolutional layers, this sequential pruning strategy is executed over several iterations, progressively reducing the number of parameters.

3.2 Tensor Decomposition

Tensor decomposition further reduces the dimensionality of the weight tensor while preserving essential information for accurate predictions. Inspired by Quantum Circuit Learning (QCL), high-dimensional tensors are decomposed into lower-dimensional forms for efficient training of quantum circuits [19].

For a weight tensor $W \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times H \times W}$, we use Singular Value Decomposition (SVD) [20] to its flattened representation $W_f \in \mathbb{R}^{C_{\text{out}} \times (C_{\text{in}} \cdot H \cdot W)}$:

$$W_f = U \Sigma V^T \tag{4}$$

Here, U and V are orthogonal matrices, and Σ is a diagonal matrix of singular values. The rank r, chosen as a hyperparameter, controls the compression by retaining only the top r singular values:

$$W_f \approx U_r \Sigma_r V_r^T \tag{5}$$

After decomposition, we reconstruct the original weight tensor using truncated matrices, significantly decreasing parameters without greatly affecting model performance.

3.3 Quantum Annealing-Inspired Matrix Factorization

Quantum annealing optimizes systems toward their lowest energy state [21]. We apply this concept to factorize weight tensors, treating it as an optimization problem aimed at minimizing the difference between the original and factorized weights.

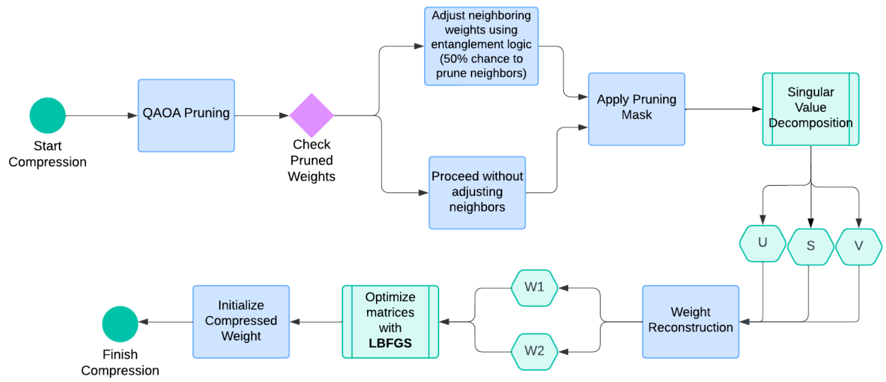

Figure 1: An illustrative diagram showcasing the QIANets framework

Given a weight matrix $W \in \mathbb{R}^{m \times n}$, we factor it into two lower-dimensional matrices, $W_1 \in \mathbb{R}^{m \times r}$ and $W_2 \in \mathbb{R}^{r \times n}$, where r is a hyperparameter that controls the rank:

$$W \approx W_1 W_2 \tag{6}$$

The objective is to minimize the reconstruction error:

$$L(W_1, W_2) = \|W - W_1 W_2\|_F^2 \tag{7}$$

Here, $\| \cdot \|_F$ denotes the Frobenius norm. We use an iterative optimization procedure based on quantum annealing to minimize this loss.

The factorization employs gradient-based optimization, initializing W_1 and W_2 randomly. Using an optimizer like LBFGS, suitable for small parameter sets and non-convex landscapes, we simulate quantum annealing by gradually reducing step size to ensure convergence to a local minimum. Once complete, the compressed weight matrix is defined as:

$$W_c = W_1 W_2 \tag{8}$$

This compressed matrix replaces the original matrix, reducing model complexity.

4 Experiments and Results

We applied our method to compress three CNNs: DenseNet, GoogLeNet, and ResNet-18, all on the CIFAR-10 dataset. These networks were selected for their different design structures and computational demands, such as parameter count, depth, and layer types, providing a comprehensive assessment of our method's effectiveness across different models. We evaluated the models for image classification performance, focusing on metrics such as inference time, speedup ratio, and accuracy.

Each experiment involved 1) applying the QIANets framework to the respective model architecture and 2) evaluating the models and comparing them to their baseline counterparts. *The results, including networks' changes before and after compression, are shown in Table 1.*

Table 1: Model Performance Comparison Before and After Compression with Rounded Figures

Model	**Base Accuracy**	**Base Latency**	**New Accuracy**	**New Latency**	**Compression Ratio**
GoogleNet	94%	0.00096 T/I	86%	0.00083 T/I	x1.9
ResNet	93%	0.00011 T/I	87%	0.00007 T/I	x1.6
DenseNet	94%	0.000050 T/I	88%	0.000042 T/I	x1.8

4.1 Experimental Setup

The experiments were conducted within the PyTorch framework, utilizing the CIFAR-10 dataset [22]. The CIFAR-10 dataset, which consists of 60,000 32x32 RGB color images in 10 classes, with 6,000 images per class, was split into training and validation sets with an 80/20 ratio. Data preprocessing included resizing images to 224x224 pixels and normalizing pixel values to the range [-1, 1]. Moreover, to generate data variability, data augmentation strategies (random horizontal flipping and cropping) were applied, improving performance on unseen data.

All computations were accelerated using CUDA on an NVIDIA A40 GPU via Runpod. Each model underwent training for 50 epochs utilizing the Adam optimizer (learning rate = 0.001, weight decay = 1e-4), using approximately 1.6 million TFLOPS-seconds of compute. To ensure consistency across models, batch sizes of 128 were used for training, while batch sizes of 256 were employed for both evaluation and testing within the dataset. The training process involved 10 trials of 10 epochs each, followed by a final trial of 50 epochs.

4.2 Hyperparameter Tuning

Hyperparameter tuning is performed using Optuna, a framework that implements various techniques to optimize certain parameters. Optuna tests various combinations of hyperparameters (batch size, learning rate and ECA Kernel Size) and dynamically adjusts them based on each trial. Following each trial, validation accuracy is calculated to test the effectiveness of current parameters. This information refines the subsequent parameters, ultimately approaching optimized parameter configurations.

4.3 Model Specific Analysis

To effectively accommodate the unique architectures of each model, we made minimal targeted adjustments to the QIANets method but ensured that all models were trained and evaluated under consistent and fair conditions throughout the experiments. *See Appendix A for additional details*

4.3.1 GoogLeNet

GoogLeNet is a convolutional network with nine multi-scale processing Inception modules. The QIANets framework targets these modules, reducing the weight in their convolutions: layer sparsity of 0.1417, while employing a rank of 41 to efficiently decompose and factorize these weights. *See Table 2.*

Table 2: Training and validation metrics for GoogLeNet

Metric	Quantum-Inspired GoogLeNet	Baseline GoogLeNet
Training Loss	0.7786	1.7066 (Epoch 1)
Validation Accuracy	80.19%	38.52% (Epoch 1)
Test Loss	0.5732	0.2557
Test Accuracy	86.65%	94.29%
Average Inference Time/Image	0.000835 seconds (13.65% Faster)	0.000967 seconds

4.3.2 DenseNet

We experimented on DenseNet – a CNN structured with 12 dense blocks, with layer-by-layer connections. This intricate connectivity requires careful application of QAOA pruning to ensure that weight removal does not disrupt the model's residual stream. *See Table 3.*

Table 3: Training and validation metrics for DenseNet

Metric	Quantum-Inspired DenseNet	Baseline DenseNet
Training Loss	0.5351	2.3027
Validation Accuracy	81.33%	10.34%
Test Loss	0.4712	0.2462
Test Accuracy	88.52%	94.05%
Average Inference Time/Image	0.000042 seconds (15.20% faster)	0.000050 seconds

4.3.3 ResNet-18

Lastly, we tested on ResNet-18, a CNN characterized by its unique residual learning framework and shortcut connections. The QIANets framework targets the residual blocks in the model, reducing less significant weights and channels, detected by ECA's straightforward 1D convolution. *See Table 4.*

Table 4: Training and validation metrics for ResNet-18

Metric	Quantum-Inspired ResNet-18	Baseline ResNet-18
Training Loss	0.4078	0.6501
Validation Accuracy	90.25%	91.30%
Test Loss	0.6447	0.3195
Test Accuracy	87.11%	93.11%
Average Inference Time/Image	0.00007 seconds (36.4% faster)	0.00011 seconds

4.4 Analysis of the QIANets Framework

The QIANets framework achieves compression ratios of 1.9× for GoogleNet, 1.8× for DenseNet, and 1.6× for ResNet-18, demonstrating effective latency reductions. Each model showed consistent loss reduction, approached baseline accuracy post-fine-tuning, and achieved faster inference. Although results are slightly below some CNN compression methods [23], QIANets shows promise for quantum-inspired compression.

5 Conclusion

In this paper, we introduced the QIANets framework, applying it to DenseNet, GoogLeNet, and ResNet-18 to reduce latency and improve inference time while preserving accuracy. Our results highlight the potential of quantum-inspired techniques for CNN compression, yielding valuable insights into the trade-offs between latency and accuracy across various architectures.

6 Limitations

While our results demonstrate the potential of QIANets and quantum-inspired principles in model compression, they also highlight several factors influencing performance. Future work should address these limitations through more in-depth experiments assessing the scalability and practical relevance of quantum-inspired techniques.

1. **Data Constraints**: The evaluation was limited to the relatively simple CIFAR-10 dataset, which may not fully capture the diversity, complexity, or scalability challenges present in larger real-world datasets. Additionally, due to the computational expense, the approach was tested on a restricted number of trials.
2. **Model Adaptation**: The lack of adaptation across different architectures may hinder the QIANets framework's ability to balance latency and accuracy. Performance in certain scenarios does not guarantee similar results across architectures without model-specific adjustments, complicating future adaptations.
3. **Hardware Limitations**: This study does not address hardware-specific limitations. Our techniques have yet to be optimized for specialized hardware, such as custom FPGAs or GPUs, which could further reduce latency and improve data throughput.

Acknowledgments and Disclosure of Funding

This work was completed through the Algoverse program, and we acknowledge the team for their support. We also thank Sean O'Brien and the anonymous reviewers for their valuable feedback.

References

[1] Norman Makoto Su and David J Crandall. The affective growth of computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9291–9300, 2021.

[2] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] CS Anumol. Advancements in cnn architectures for computer vision: A comprehensive review. In *2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS)*, pages 1–7. IEEE, 2023.

[6] Dominik Honegger, Helen Oleynikova, and Marc Pollefeys. Real-time and low latency embedded computer vision hardware based on a combination of fpga and mobile cpu. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4930–4935. IEEE, 2014.

[7] Zhuo Li, Hengyi Li, and Lin Meng. Model compression for deep neural networks: A survey. *Computers*, 12(3):60, 2023.

[8] R Divya and J Dinesh Peter. Quantum machine learning: A comprehensive review on optimization of machine learning algorithms. In *2021 Fourth International Conference on Microelectronics, Signals & Systems (ICMSS)*, pages 1–6. IEEE, 2021.

[9] Shyambabu Pandey, Nihar Jyoti Basisth, Tushar Sachan, Neha Kumari, and Partha Pakray. Quantum machine learning for natural language processing application. *Physica A: Statistical Mechanics and its Applications*, 627:129123, 2023.

[10] Samer Francy and Raghubir Singh. Edge ai: Evaluation of model compression techniques for convolutional neural networks. *arXiv preprint arXiv:2409.02134*, 2024.

[11] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*, 2017.

[12] Zejiang Hou, Minghai Qin, Fei Sun, Xiaolong Ma, Kun Yuan, Yi Xu, Yen-Kuang Chen, Rong Jin, Yuan Xie, and Sun-Yuan Kung. Chex: Channel exploration for cnn model compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12287–12298, 2022.

[13] Shangshang Shi, Zhimin Wang, Guolong Cui, Shengbin Wang, Ruimin Shang, Wendong Li, Zhiqiang Wei, and Yongjian Gu. Quantum-inspired complex convolutional neural networks. *Applied Intelligence*, 52(15):17912–17921, 2022.

[14] Zhirui Hu, Peiyan Dong, Zhepeng Wang, Youzuo Lin, Yanzhi Wang, and Weiwen Jiang. Quantum neural network compression. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9, 2022.

[15] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[16] Stephen Hanson and Lorien Pratt. Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems*, 1, 1988.

[17] Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pages 293–299. IEEE, 1993.

[18] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou. The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size. *Quantum*, 6:759, 2022.

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

[20] Xin Wang, Zhixin Song, and Youle Wang. Variational quantum singular value decomposition. *Quantum*, 5:483, 2021.

[21] Alessandro Gherardi and Alberto Leporati. An analysis of quantum annealing algorithms for solving the maximum clique problem. *arXiv preprint arXiv:2406.07587*, 2024.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

A Extended Results and Model Analysis

A.1 GoogLeNet:

Performance Progression Across Epochs: During Trial 0 of optimization, the model began with a validation accuracy of 21.08% at Epoch 1 and quickly progressed to 70.18% by Epoch 10, indicating *rapid learning during the first stages of training*. The highest validation accuracy was 80.19% throughout Trial 5. The final quantum-inspired GoogLeNet's test accuracy was 86.65%, a notable improvement from the earlier 80.19%, approaching the baseline accuracy of 94.29% after fine-tuning.

Loss Reduction: Over the 50 epochs, the model's loss steadily decreased, showing consistent improvement. It began at 2.5205 in Epoch 1 trial 0 and reduced to 0.8256 by Epoch 50, effectively minimizing error throughout training. A key outcome of this experiment was the final **13.65%** reduction in inference time, minimizing to 0.000835 seconds per image, which underscores the efficiency of our approach compared to the baseline GoogLeNet's 0.000967 seconds. *See Table 2.*

A.2 DenseNet:

Performance Progression Across Epochs: The model began with a validation accuracy of 9.66% at Epoch 1 (Trial 0) but exhibited minimal improvement by Epoch 10, reaching 10.34%. However, Trial 1 demonstrated significant progress, starting at 27.53% and achieving a remarkable 81.33% by Epoch 10. The model achieved its highest validation accuracy of 86.65% in Trial 1, with a layer sparsity of 0.3779 (62% of weights pruned). After fine-tuning, the quantum-inspired DenseNet achieved a test accuracy of 88.52%, an improvement from the 86.65%, approaching the baseline accuracy (94.05%).

Loss Reduction: The model demonstrated steady loss reduction throughout the training process, beginning at 2.3028 during Epoch 1 in Trial 0 and decreasing to 0.5606 by Epoch 10 in Trial 1, indicating effective error minimization. This consistent decline reflects the model's ability to optimize its parameters and improve performance across trials. One of the standout results of this experiment was the final reduction in inference time by **15.20%**, dropping to 0.001043 seconds/image, marking a considerable improvement compared to the baseline DenseNet's 0.00123 seconds. *See Table 3.*

A.3 ResNet-18:

Performance Progression: Throughout the trials, there were notable fluctuations in performance. The highest validation accuracy across all trials peaked at 91.42% in Trial 4, where the model achieved a layer sparsity of approximately 0.3779 (meaning nearly 62% of the weights were pruned while maintaining performance). After extensive fine-tuning, the final quantum-inspired ResNet-18 reached a test accuracy of 87.11%, a significant improvement from the earlier 84.56% (the highest accuracy before the final fine-tuning) and approaching the baseline accuracy of 93.11%.

Loss Reduction: The loss reduction across trials also followed a clear downward trend. In Trial 3, with 0.4805 layer sparsity and a rank of 10, the validation loss dropped from 1.9847 to 0.6321 (first to tenth epoch), indicating better model convergence. Notably, inference time dropped by 36.4% to 0.00007 seconds/image, improving from the baseline of 0.00011 seconds/image *See Table 4.*

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The introduction clearly introduces QIANets and how the method aims to integrate the quantum-inspired strategies, accurately reflecting the paper's scope. The claims made in the abstract/introduction are supported by the results obtained.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in section 6, explaining the data and hardware constraints and limited number of runs. We state the lack of adaptation and provide avenues for future work to experiment on a more expansive range of models and datasets.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include any theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The code and instructions on how to reproduce the experiments are included in the GitHub link. Within the paper, the model is described in detail with code snippets and is reproducible using the information provided.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code is provided with instructions through the GitHub link.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper covers the experimental set up, including training and test details in section 4.1. Full details are included in the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Statistical significance tests are not included as we could not conduct multiple trails of our finalized method due to the computational expense. As a result, error bars and confidence intervals are not included in the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies our use of an NVIDIA A40 GPU and the approximate amount of compute used in section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted complies fully with all of the ethical guidelines listed.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The research has no potential societal impacts besides increasing the speed of responses from preexisting models.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not introduce/include data or models with such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly cites the CIFAR-10 dataset that the models were tested on.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All the code is provided as supplemental material, and there are no new assets besides the proposed method.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.