

Quantum Advantage of Knowledge Distillation in the QAOA Setting

Abstract—In this work, we investigate the application of Knowledge Distillation (KD) to the Quantum Approximate Optimization Algorithm (QAOA) setting, introducing a framework to distill knowledge from a Multi-angle QAOA (ma-QAOA) teacher to a standard, single-layer QAOA student. We propose three distillation schemes: classical count-based KD, mean-value distillation using mean squared error, and mean-value distillation with Huber loss for enhanced robustness. Numerical experiments on diverse graph families—including complete graphs, Erdős–Rényi graphs, complete binary trees, and random 3-regular graphs—demonstrate that mean-value and Huber-based distillation reliably preserve or improve the student’s approximation performance, while count-based KD exhibits higher variance and limited generalization. Our results show that KD can effectively compress QAOA parameter spaces without sacrificing solution quality, providing a practical avenue for efficient quantum algorithm deployment and improved training stability.

I. INTRODUCTION

Knowledge Distillation (KD) is widely used in the field of machine learning to transfer knowledge from larger, more complex models (the ‘teacher’ models) to smaller, more efficient (‘student’) models, ideally with minimal performance loss [1] so as to reduce computational resources, or, as is feasible in a subset of cases, to enhance performance due to improved generalization from the simplified model via a reduction in overfitting. It has been applied widely across domains [2], including computer vision and natural language processing (NLP), including transformer-based large language models (LLMs) [3].

More recently, a small number of works [4]–[6] have sought to apply KD with respect to the problem of the transfer of learning from Classical to Quantum Neural Networks (QNNs) [7] in order to leverage the potentially advantageous optimization landscapes of QNNs (QNNs being implemented in terms of currently-feasible Noisy Intermediate-Scale Quantum (NISQ) hardware through Variational Quantum Circuits (VQCs) [8], which have been shown to attain a significantly greater effective dimension with faster training than parametrically-equivalent classical neural networks [9]).

To date, no work has sought to apply the principle of KD to the Quantum Approximate Optimization Algorithm (QAOA) setting of quantum machine learning, a lack that the current investigation seeks to address. QAOA has certain natural features that lend itself to this domain. In particular, QAOA represents parameter optimization of approximately solvable combinatorial optimization problems in the form of NISQ-era algorithm [10], such that a ground state *cost Hamiltonian* H_C encodes the solution to the combinatorial optimization problem and a *mixer Hamiltonian* H_M that encodes optimization

transitions (such as graph node operations). The two oracles $\exp(-i\alpha H_M)$ and $\exp(-i\alpha H_C)$ are applied alternately in an iterated manner to an initial state (consisting of a superposition in the computational basis) in order to adiabatically optimize the parameters γ, α in accordance with the Quantum Adiabatic Theorem, which guarantees that the solution remains in the ground state of final Hamiltonian. The unitary operators in standard QAOA utilize $2p$ parametric angles, where p can be considered the number of layers of the composite oracle. The Multi-angle Quantum Approximate Optimization Algorithm (ma-QAOA) [11] variant of QAOA introduces additional parameters such the new parameters are allocated to each summand of the cost and mixing operators instead of the application of a single angle applied to the cost operator and mixing operator. While the introduction of more parameters potentially leads to a more challenging optimization problem, the corresponding reduction in circuit depth p helps to reduce circuit noise such that good parameters can often be found in polynomial time.

In this context, QAOA present as an opportunity for evaluating the advantage of KD in the quantum setting: by investigation of the distillation of knowledge from ma-QAOA models into QAOA models we can assess the potential for further improvement of QAOA performance by taking advantage of so-called ‘dark knowledge’ contained with the more highly parameterized ma-QAOA quantum teacher model applied to a simpler student model specified in the standard QAOA format.

II. THEORETICAL BACKGROUND

A. Knowledge Distillation

In the standard classical setting of Knowledge Distillation [1] a student model directly learns to mimic the final output probabilities of a teacher model. The softened output of the teacher and the student are given as

$$p_m = \sigma\left(\frac{z_m}{T}\right) = \frac{\exp\left(\frac{z_m}{T}\right)}{\sum_{i=1}^N \exp\left(\frac{z_m^i}{T}\right)}, \quad (1)$$

where $m = \{t, s\}$ is an index that indicates teacher or student, z_m is the output vector of logits, N is the number of classes, and T is a temperature parameter to soften the probability. The overall loss function for response-based KD can be formulated as

$$L_{KD} = \alpha L_{CE} + (1 - \alpha) L_{KL}, \quad (2)$$

where L_{CE} is the Cross Entropy (CE) used to minimize the difference between the student soft logits and the ground truth, L_{KL} is the Kullback-Leibler (KL) divergence loss used to minimize the difference between the student and teacher soft logits, and α is a hyperparameter that balances the contributions. The temperature T in the softmax function plays a crucial role in controlling the smoothness of the probability distribution produced by the teacher model. Higher temperatures produce softer probability distributions, revealing inter-class relationships that are otherwise obscured in standard hard labels. This softened output, often referred to as ‘dark knowledge’, provides richer information for the student model, facilitating better generalization.

While KL-divergence is the most commonly utilized loss in the classical KD setting, in practice any form of suitable distributional loss can be substituted.

B. QAOA and ma-QAOA

The QAOA algorithm as specified in [10] is implemented via the application of two operators,

$$U(C, \gamma) = e^{-iC\gamma} \quad \& \quad U(B, \beta) = e^{-iB\beta}$$

in iterative succession to the initial state (specified in the computational basis $|z\rangle$):

$$|s\rangle = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle.$$

Applied p times, this results in the state:

$$|\gamma, \beta\rangle = \prod_{i=1}^p U(B, \beta_i) U(C, \gamma_i) |s\rangle$$

where C encodes the cost function of the combinatorial optimization problem and B is responsible for ‘mixing’ transitions of the initial state, with both B and C typically summations (over clauses in the latter case):

$$B = \sum_{v \in V(G)} B_v \quad \& \quad C = \sum_a C_a$$

where $B_v = \sigma_v^x$ is the Pauli-x operator applied to the v^{th} qubit. Hence, the final Mixing and Cost Unitaries are:

$$U(C, \gamma) = e^{-i\gamma \sum_a C_a} \quad \& \quad U(B, \beta) = e^{-i\beta \sum_{v \in V(G)} B_v}$$

In ma-QAOA [11], QAOA is modified with additional parameters in the summand of the cost and mixing operators so that they each have their own angle:

$$U(C, \vec{\gamma}_l) = e^{-i \sum_a C_a \gamma_{l,a}} = \prod_a e^{-i \gamma_{l,a} C_a}$$

and

$$U(B, \vec{\beta}_l) = e^{-i \sum_{v \in V(G)} B_v \beta_{l,v}} = \prod_{v \in V(G)} e^{-i \beta_{l,v} B_v}$$

with $\vec{\gamma}_l = (\gamma_{l,a_1}, \gamma_{l,a_2}, \dots)$ and $\vec{\beta}_l = (\beta_{l,v_1}, \beta_{l,v_2}, \dots)$. The l parameter indexes over layers, a_i refers to edges in the

graph combinatorial optimization problem, and v_j denotes vertices. Typically, fewer l iterations are required for equivalent performance to QAOA.

C. The Max Cut Optimization Problem

The maximum cut of a graph is a cut whose size is no smaller than that of any other cut, with ‘cut’ being defined as a partitioning of the graph’s vertex set into two complementary sets S and T , such that the number of edges between S and T is a supremum. More formally, a undirected graph $G = (V, E)$ is defined as a set of vertices V and a set of undirected edges (E) connecting them. A cut is then some subset of edges $C \in E$ such that the terminals (defined as the vertex end points of an edge) are unconnected on the induced graph $G(C) = (V, E \setminus C)$. A cut of a graph is hence an edge-dependent bipartite subgraph, with size characterized as the number of edges contain within it. The problem of finding the maximal such cut is classically APX-hard, there being no polynomial-time approximation scheme with arbitrary closeness to the optimum unless it can be proved that $P = NP$ (i.e. the approximation ratio is strictly < 1).

III. PROPOSED METHODOLOGY

We perform knowledge distillation from a ma-QAOA to a standard QAOA, aiming to compress the parameter space while preserving optimization performance. Each distillation scheme jointly minimizes the MaxCut cost function and a teacher–student loss that encodes the knowledge transfer.

A. Distillation Schemes

The total loss for all schemes is defined as

$$\mathcal{L}_{\text{total}} = \langle C \rangle_S + \lambda \mathcal{L}_{\text{distill}}, \quad (3)$$

where $\langle C \rangle_S$ is the expected cost (energy) of the MaxCut Hamiltonian for the student QAOA, $\mathcal{L}_{\text{distill}}$ is the knowledge distillation term, and λ balances the two contributions.

- 1) **Classical KD on Measurement Counts:** In this classical KD approach, the student QAOA is trained to replicate the full output probability distribution of the teacher ma-QAOA, effectively matching the likelihood of each possible state:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{KL}} = D_{\text{KL}}(p_T \parallel p_S), \quad (4)$$

where $D_{\text{KL}}(\cdot \parallel \cdot)$ denotes the Kullback–Leibler divergence.

- 2) **Mean-Value Distillation (MSE):** The second scheme distills the mean measurement values of each qubit through MSE:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^n (\langle Z_i \rangle_T - \langle Z_i \rangle_S)^2, \quad (5)$$

where n is the number of qubits.

- 3) **Mean-Value Distillation (Huber)**: The third variant replaces the MSE with the Huber Loss for robustness against outliers:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{Huber}} = \frac{1}{n} \sum_{i=1}^n \begin{cases} \frac{1}{2} \Delta_i^2, & |\Delta_i| \leq \delta, \\ \delta(|\Delta_i| - \frac{1}{2}\delta), & \text{otherwise,} \end{cases} \quad (6)$$

with $\Delta_i = \langle Z_i \rangle_T - \langle Z_i \rangle_S$ and tunable parameter δ .

Each scheme therefore combines the MaxCut objective with a KD term, providing compact QAOA parameterizations that maintain competitive performance while transferring knowledge from ma-QAOA.

IV. EXPERIMENTAL ASSESSMENT

A. Experimental Setting

To evaluate the performance of the proposed distillation schemes, we considered the MaxCut problem on 16 graph instances, grouped into four distinct classes. Each class includes four graphs with $n \in \{4, 6, 8, 10\}$ nodes:

- 1) The class K_n of complete graphs, in which every pair of nodes is connected.
- 2) The class $E_{0.5,n}$ of Erdős-Rényi graphs with edge probability $P = 0.5$, where each edge is independently assigned according to a Bernoulli distribution.
- 3) The class $T_{2,n}$ of complete binary trees with branching factor $b = 2$: trees in which every non-leaf node has exactly b children, and all levels are fully filled except possibly for the rightmost nodes at the lowest level.
- 4) The class $R_{3,n}$ of random 3-regular graphs with n nodes, in which each node has degree $\equiv 3$ with graphs randomly selected from the set of all such 3-regular graphs.

These graphs were employed both for training the ma-QAOA teacher and for evaluating the distilled QAOA student, providing a benchmark across differing topologies and sizes.

B. Hyperparameters

The ma-QAOA teacher circuits were constructed with $p = 10$ layers, while the student QAOA circuits used a single layer ($p = 1$). Optimization was performed using the `minimize` function from `scipy` with the COBYLA method. A fixed random seed was employed for reproducibility across all experiments.

The distillation loss balances the MaxCut cost and the KD term via a weight λ , for which we tested three values: $\lambda \in \{0.1, 0.25, 0.5\}$. For the Mean-Value Huber distillation, the robustness parameter was set to $\delta = 1$. In addition, student QAOA circuits were also trained without KD to assess the impact of distillation on performances.

C. Numerical Results

The performance of each distillation scheme was evaluated using the *Approximation ratio*, defined as the ratio between the MaxCut value obtained by the algorithm and the optimal classical solution:

$$r = \frac{C_{\text{alg}}}{C_{\text{opt}}}, \quad (7)$$

TABLE I
SUMMARY OF QAOA AND DISTILLATION PARAMETERS.

Parameter	Value
Teacher layers	10
Student layers	1
Distillation weight λ	0.1, 0.25, 0.5
Huber δ	1
Number of measurement shots	2200
Random seed	Fixed

where $C_{\text{alg}} = \langle C \rangle$ is the expectation value of the MaxCut cost Hamiltonian measured from the quantum circuit, and C_{opt} is the exact optimal cut value computed classically. Higher values of r therefore indicate better approximation quality.

Figures 1–3 report the average approximation ratios across all graph types and node counts for the three distillation paradigms.

Count-based Distillation: The count-based knowledge distillation (Fig. 1) shows relatively high variance across all graph types and graph sizes. While it occasionally improves upon the baseline QAOA for specific instances, it often struggles to generalize, leading to inconsistent performance. This sensitivity is especially evident for Erdős-Rényi graphs, where the output probability distributions can vary significantly depending on the particular $G(n, p)$ realization.

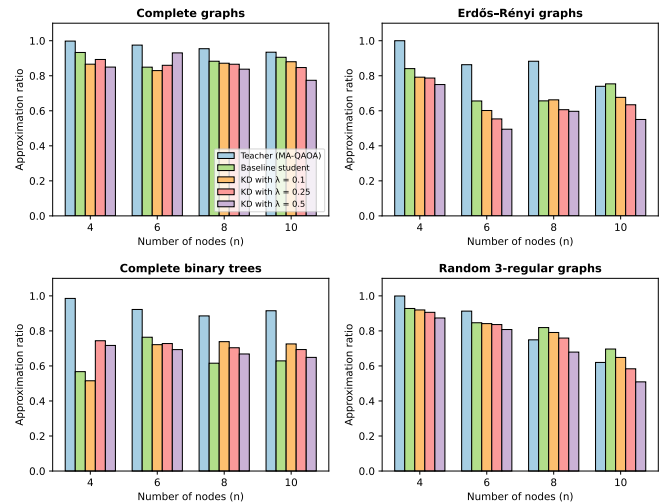


Fig. 1. Results for Count-based Distillation.

Mean-value Distillation ($\langle Z \rangle$): Distillation based on the mean expectation values of the qubits (Fig. 2) exhibits a more consistent and stable behavior. Although substantial improvements over the teacher are observed only in a few instances, the distilled student never underperforms the baseline QAOA. This indicates that the mean-value representation provides a smooth and reliable channel for knowledge transfer, enabling effective parameter reduction while maintaining the same level of performance and avoiding any degradation across graph families.

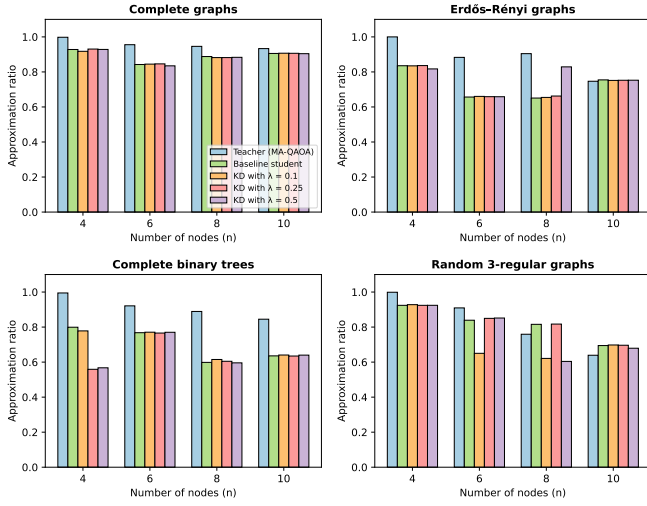


Fig. 2. Results for Mean-value Distillation ($\langle Z \rangle$).

Huber-based Distillation: Replacing the MSE loss with the Huber loss (Fig. 3) further stabilizes training by mitigating the impact of outliers in the expectation estimates. The improvement is modest but systematic, confirming that Huber loss provides a useful trade-off between sensitivity and robustness.

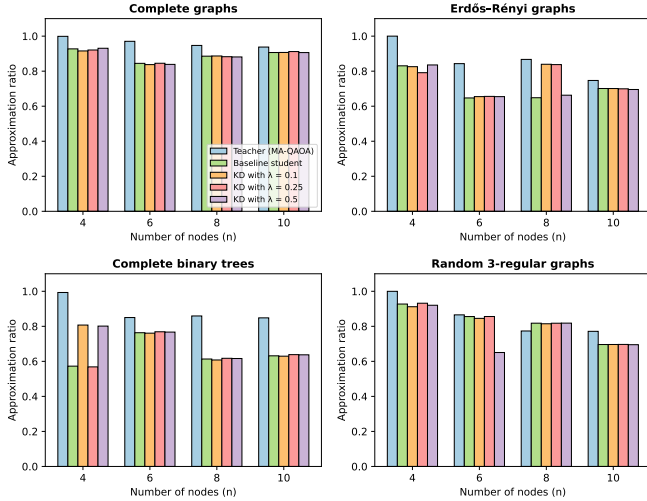


Fig. 3. Results for Mean-value Distillation (Huber).

Discussion: Overall, the results demonstrate that knowledge distillation from ma-QAOA to QAOA is feasible and can lead to tangible performance improvements. Distillation based on qubit-level expectation values ($\langle Z \rangle$) provides a particularly stable and reliable transfer mechanism, enabling a significant reduction in the number of parameters without compromising solution quality. Huber loss further enhances this stability by mitigating the influence of outliers in the expectation estimates, offering a balanced trade-off between sensitivity and robustness. While the improvements over the baseline QAOA are modest, they are systematic, and in all tested scenarios, the student model maintains comparable or slightly better performance.

These findings confirm that both mean-value and Huber-based distillation approaches can effectively compress QAOA circuits while preserving training stability and approximation quality.

V. CONCLUSIONS

KD is widespread in classical machine learning for compressing large models into smaller, efficient ones while preserving—sometimes even enhancing—performance [1], [3]. KD in the context of quantum machine learning is much less explored, and focussed only on transferring knowledge from classical NN to QNNs [4]–[6], in order to exploit optimization landscapes of VQCs implemented on NISQ hardware [8], [9]. To date, no study has addressed KD in relation the QAOA family, despite the natural fit of KD to the ma-QAOA to QAOA KD problem.

In this work, we systematically evaluated knowledge distillation schemes for transferring information from a ma-QAOA teacher to a standard single-layer QAOA student, proposing 3 distillation strategies: classical count-based KD, mean-value KD via MSE, and mean-value KD with Huber loss. Numerical experiments on diverse graph families—including complete graphs, Erdős-Rényi graphs, complete binary trees & random 3-regular graphs—demonstrate that:

- Count-based distillation exhibits high variance and limited generalization, particularly for random graph instances.
- Mean-value distillation (MSE) provides a stable and consistent transfer of knowledge, reliably preserving or slightly improving the student’s approximation ratio across all tested graphs.
- Incorporating the Huber loss further enhances robustness by mitigating the impact of outliers in qubit expectation estimates, leading to systematic, modest performance gains without compromising stability.

Overall, the results confirm that knowledge distillation can significantly reduce the QAOA parameter space while maintaining competitive performance, providing a viable mechanism for efficient quantum algorithm deployment.

Future directions include a more granular analysis of small graph instances to better understand which aspects of the teacher’s knowledge are successfully transferred and under what conditions. Insights from such analyses could pave the way toward generalizable techniques for reliable parameter and knowledge transfer across QAOA variants, potentially enabling systematic strategies for accelerating quantum optimization and improving generalization in variational quantum algorithms.

Acknowledgment: We thank Leonardo Lavagna for providing the QAOA¹ & ma-QAOA² codebases, and for valuable discussions.

¹<https://github.com/leonardoLavagna/qaoa>

²https://github.com/leonardoLavagna/ma_qaoa

REFERENCES

- [1] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015.
- [2] C. Hu, X. Li, D. Liu, H. Wu, X. Chen, J. Wang, and X. Liu, “Teacher-student architecture for knowledge distillation: A survey,” 2023.
- [3] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou, “A survey on knowledge distillation of large language models,” 2024.
- [4] M. J. Hasan and M. Mahdy, “Bridging classical and quantum machine learning: Knowledge transfer from classical to quantum neural networks using knowledge distillation,” *arXiv preprint arXiv:2311.13810*, 2023.
- [5] S. Piperno, G. Vittori, D. Windridge, A. Rosato, and M. Panella, “Classical to quantum knowledge distillation: a study on the impact of hybridization,” in *Proceedings of International Joint Conference on Neural Networks (IJCNN 2025)*, 30 Jun - 05 Jul 2025, Rome, Italy, 2025.
- [6] M. Li, L. Fan, A. Cummings, X. Zhang, M. Pan, and Z. Han, “Hybrid quantum classical machine learning with knowledge distillation,” in *ICC 2024-IEEE International Conference on Communications*. IEEE, 2024, pp. 1139–1144.
- [7] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, “An artificial neuron implemented on an actual quantum processor,” *npj Quantum Information*, vol. 5, no. 1, pp. 5–10, 2019.
- [8] M. Cerezo *et al.*, “Variational quantum algorithms,” *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.
- [9] A. Abbas *et al.*, “The power of quantum neural networks,” *Nature Computational Science*, vol. 1, no. 6, pp. 403–409, 2021.
- [10] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” 2014. [Online]. Available: <https://arxiv.org/abs/1411.4028>
- [11] R. Herrman, P. C. Lotshaw, J. Ostrowski, T. S. Humble, and G. Siopsis, “Multi-angle quantum approximate optimization algorithm,” 2021. [Online]. Available: <https://arxiv.org/abs/2109.11455>