PALQO: Physics-informed Model for Accelerating Large-scale Quantum Optimization

Yiming Huang[1*], **Yajie Hao**[2*], **Jing Zhou**[5], **Xiao Yuan**[1†], **Xiaoting Wang**[2†], **Yuxuan Du**[3,4†]
[1]Center on Frontiers of Computing Studies,
and School of Computer Science, Peking University, Beijing, China
[2]Institute of Fundamental and Frontier Sciences
University of Electronic Science and Technology of China, Chengdu, China
[3]College of Computing and Data Science
Nanyang Technological University, Singapore, Singapore
[4]School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore, Singapore
[5]Department of Physics, Fudan University, Shanghai, China
yiminghwang@gmail.com, yuxuan.du@ntu.edu.sg, xiaoyuan@pku.edu.cn
xiaoting@uestc.edu.cn

Abstract

Variational quantum algorithms (VQAs) are leading strategies to reach practical utilities of near-term quantum devices. However, the no-cloning theorem in quantum mechanics precludes standard backpropagation, leading to prohibitive quantum resource costs when applying VQAs to large-scale tasks. To address this challenge, we reformulate the training dynamics of VQAs as a nonlinear partial differential equation and propose a novel protocol that leverages physics-informed neural networks (PINNs) to model this dynamical system efficiently. Given a small amount of training trajectory data collected from quantum devices, our protocol predicts the parameter updates of VQAs over multiple iterations on the classical side, dramatically reducing quantum resource costs. Through systematic numerical experiments, we demonstrate that our method achieves up to a 30x speedup compared to conventional methods and reduces quantum resource costs by as much as 90% for tasks involving up to 40 qubits, including ground state preparation of different quantum systems, while maintaining competitive accuracy. Our approach complements existing techniques aimed at improving the efficiency of VQAs and further strengthens their potential for practical applications.

1 Introduction

Modern quantum computers, with a steadily increasing number of high-quality qubits, are approaching the threshold of practical utility [1–3]. In this pursuit, variational quantum algorithms (VQAs) [4–12] have emerged as a leading strategy, attributed to their flexibility in accommodating circuit depth and qubit connectivity among different platforms. In recent years, a wide range of theoretical and experimental studies have demonstrated the feasibility of VQAs across diverse applications, such as quantum chemistry [13–15], optimization [16–18], and machine learning [19–23]. Despite the progress, they face critical challenges when applied to large-scale problems. In particular, the no-cloning theorem and the unitary constraints in the quantum universe prohibit the use of backpropagation techniques common in deep learning [24], requiring VQAs to update sequentially to

*Equal contribution.
†Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

minimize a predefined cost function [25, 26]. This approach imposes substantial and even prohibitive quantum resource demands, especially in terms of the number of measurements required. Given the scarcity of quantum computers in the foreseeable future, enhancing the optimization efficiency of VQAs while minimizing resource consumption is crucial for enabling their practical deployment.

To advance VQAs for large-scale problems, substantial efforts have been devoted to improving the optimization efficiency. Prior literature in this field can be broadly categorized into three primary classes (see Sec. 2.1 for details). The first aims to reduce measurement costs in quantum many-body and chemistry problems by grouping terms in the Hamiltonian to enable simultaneous measurements [27–29]. The second harnesses classical simulators or learning models to identify well-initialized parameters that are close to local minima of the cost landscape for a given VQA, thereby improving convergence efficiency [30–32]. The third seeks to predict the dynamics of parameter updates by revising past optimization trajectories for the given task [33], as exemplified by methods such as recurrent neural networks [34, 35] and QuACK [36]. Despite significant advancements, no existing approach effectively balances optimization efficiency and accuracy at scale. In this regard, a critical question arises: *is it possible to achieve both for large-scale VQA systems?*

Towards this question, we observe that prior efforts have primarily focused on hardware improvements and heuristic optimizations, while the potential of approximating training dynamics to alleviate the quantum resource burden **remains underexplored**. In response, we introduce a fresh perspective by utilizing Taylor expansion to reformulate the parameter optimization process in VQAs as a nonlinear partial differential equation (PDE). In this way, the evolution of this nonlinear PDE corresponds to the trajectory of parameter updates during training. Building on this formulation, we devise a protocol, dubbed PALQO, that employs a physics-informed neural network (PINN) [37, 38] to approximate solutions to the PDE, where high-order terms in the Taylor expansion serve as boundary conditions. The proposed framework is generic, which encompasses the quantum neural tangent kernel (QNTK) as a special case [39–41]. Besides, we prove that a polynomial number of training samples is sufficient to ensure that PALQO attains a satisfactory generalization ability.

We then conduct extensive numerical simulations on different ground state preparation tasks, including the transverse-field Ising model, Heisenberg model, and multiple molecule systems, to investigate the effectiveness of PALQO. Simulation results up to 40 qubits indicate that by learning from a limited set of initial parameter updates obtained from quantum devices, PALQO, which is deployed on classical hardware, can accurately predict future parameter updates. These results indicate the effectiveness of reducing the number of quantum measurements required during optimization. Numerical experiments on ground state preparation tasks across large-scale systems, involving up to 40 qubits, validate the effectiveness of PALQO. Moreover, we show that the proposed PALQO is complementary to existing approaches for improving the optimization efficiency of VQAs. To support the community, we release our source code at [42]. These results open a new avenue for leveraging the power of PINNs to enhance the efficiency of VQAs and advance the frontier of practical quantum computing.

Contributions. For clarity, we summarize our main contributions below. **(1)** To our best knowledge, we establish the first general framework between the optimization trajectory of VQAs and PDE, thereby enabling the employment of various PINNs to advance the optimization efficiency of large-scale VQAs. **(2)** We propose PALQO, an effective PINN oriented to reduce the required number of measurements when training large-scale VQAs, and prove its generalization ability. **(3)** Unlike prior studies that mainly focus on small-scale tasks, we conduct extensive numerical studies to validate the advancements of PALQO up to 40 qubits, providing valuable insights to further improve the optimization efficiency of VQAs at scale.

2 Preliminaries

In this section, we provide a concise overview of the basic concepts of quantum computing, variational quantum algorithms and physics-informed neural networks to set the stage for their integration in accelerating the quantum optimization process. Please refer to Appendix A for more details.

Basics of Quantum Computing. Quantum state, quantum circuit, and quantum measurement are three key components in quantum computing [43, 44]. In particular, an n-qubit quantum state is mathematically represented as a unit vector $\mathbf{u} \in \mathbb{C}^N$ in Hilbert space, where $\sum_{j=0}^{N-1} |\mathbf{u}_j|^2 = 1, N = 2^n$. Here we follow conventions to use Dirac notation to represent $\mathbf{u}$ and its transpose conjugate $\mathbf{u}^\dagger$,

i.e., $|\mathbf{u}\rangle$ and $\langle\mathbf{u}|$. For a quantum circuit, it serves as a computational model consisting of a sequence of quantum gates that describes operations on the given input state. The most widely used quantum gates are Pauli gates, i.e., $\mathrm{X} = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $\mathrm{Z} = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$, $\mathrm{Y} = \left(\begin{smallmatrix} 0 & -i \\ i & 0 \end{smallmatrix}\right)$. According to the Solovay-Kitaev theorem [43], arbitrary operation can be approximated by a quantum circuit $U = \prod_j U_j$ where each gate U_j is drawn from a finite universal gate set, such as $\{\mathrm{CNOT}, \mathrm{H}, \mathrm{S}, \mathrm{R}_z(\theta), \mathrm{R}_x(\theta)\}$. Concretely, $\mathrm{CNOT} = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes \mathrm{X}$, $\mathrm{H} = 1/\sqrt{2}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$, $\mathrm{R}_x(\theta) = e^{-i\theta \mathrm{X}}$, $\mathrm{R}_z(\theta) = e^{-i\theta Z}$, $S = \sqrt{Z}$.

For quantum measurement, it is the process that collapses a quantum state into a definite classical outcome. In this study, we are interested in the expectation value of the measurement outcomes with a given observable O, a Hermitian operator, on quantum state $|\mathbf{u}\rangle$, i.e. $\langle\mathbf{u}|O|\mathbf{u}\rangle$. Suppose the observable presents as an n-qubit Hamiltonian that characterizes energy structure of the target quantum system in the form of $H = \sum_{j=1}^{N_H} c_j P_j$, where P_j is a tensor product of Pauli matrices, i.e. $P_j \in \{\mathbb{I}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\}^{\otimes n}$. To experimentally estimate $\langle\mathbf{u}|O|\mathbf{u}\rangle$ within error ϵ, we typically perform $M \sim \mathcal{O}(1/\epsilon^2)$ repeated measurements for each P_j on multiple copies of the state $|\mathbf{u}\rangle$ and get the outcomes $\{\hat{M}_{\mathbf{u}}^{j,k}\}_{k=1,..,M}$, then approximate the expectation value by statistical averaging $\langle\mathbf{u}|O|\mathbf{u}\rangle = 1/(MN_H)\sum_{j,k} c_j \hat{M}_{\mathbf{u}}^{j,k}$.

Variational Quantum Algorithms. Variational quantum algorithms (VQAs) algorithms designed for machine learning tasks are called quantum neural networks (QNNs) [41, 45–52], while those applied to many-body physics and quantum chemistry are typically known as variational quantum eigensolvers (VQEs) [53–60]. The primary objective of VQE is to optimize a parameterized state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\phi\rangle$ to minimize the energy function $\mathcal{E}$ defined by a given Hamiltonian $H = \sum_{j=1}^{N_H} c_j P_j$. Mathematically, the energy function to be minimized in VQEs takes the form of

$$\min_{\boldsymbol{\theta}} \mathcal{E}(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle. \tag{1}$$

A common and widely adopted approach to complete this optimization problem is utilizing a gradient-based optimizer, like gradient descent, to iteratively adjust the parameters $\boldsymbol{\theta}$ according to the partial derivative $\partial_{\boldsymbol{\theta}}\mathcal{E}$. Because there is no-clone theorem and no backpropagation without exponential classical overhead in general VQEs [24], we need to perform the *parameter shift rule* without involving other quantum resource overhead, such as ancillary qubits to estimate the partial derivative [25]. Concretely, the calculation of the partial derivative with respect to $\boldsymbol{\theta}_i$ takes the form as

$$\frac{\partial\mathcal{E}}{\partial\boldsymbol{\theta}_i} = \frac{1}{2}\left[\mathcal{E}\left(\boldsymbol{\theta}_i + \frac{\pi}{2}\right) - \mathcal{E}\left(\boldsymbol{\theta}_i - \frac{\pi}{2}\right)\right] \approx \frac{1}{MN_H}\sum_{j,k} c_j\left[\hat{M}_{\psi_+}^{j,k} + \hat{M}_{\psi_-}^{j,k}\right], \tag{2}$$

where ψ_+, ψ_- correspond to state $|\psi(\boldsymbol{\theta}_i + \pi/2)\rangle$ and $|\psi(\boldsymbol{\theta}_i - \pi/2)\rangle$, respectively.

While such a method provides a closed-form expression for gradient estimation without requiring additional qubits and can be extended to general VQEs, it necessitates evaluating $\mathcal{E}$ twice with shifted parameter values at the same position to estimate the gradient of a single parameter. Hence, suppose the dimension of $\boldsymbol{\theta}$ is p, it requires to perform $\mathcal{O}(pN_H/\epsilon^2)$ measurements to estimate the partial derivative $\partial_{\boldsymbol{\theta}}\mathcal{E}$, which becomes **computationally prohibitive** in large-scale tasks, especially for large molecules as whose n-qubits second-quantized Hamiltonian has roughly $N_H \sim \mathcal{O}(n^4)$ terms [53]. Therefore, it requires substantial resources for estimating updated parameters $\boldsymbol{\theta}$ during the optimization, which is considered as one major limitation of large-scale VQEs. Similarly, such a scalability issue also arises in QNNs, where the number of measurements required per iteration scales linearly with p and the batch size.

Physical-Informed Neural Network. Physics-informed neural networks (PINNs) have become a promising learning-based tool in approximating the solution of partial differential equations (PDEs) [61–63]. With the advantages of computational efficiency for solving complex PDE, they have been widely employed in various practical scenarios such as fluid dynamics, battery degradation modeling, disease detection, and complex systems simulation [38, 64–67]. PINNs harness the core tool, automatic differentiation, of modern machine learning to efficiently enforce the physical constraints of the underlying PDE.

For a PDE problem, it can be generally written as $\mathcal{N}[u(\boldsymbol{x}, t)] = g(\boldsymbol{x}, t)$ where $\boldsymbol{x} \in \mathcal{D} \subset \mathbb{R}^d$ denotes variables, $\mathcal{N}$ represents the differential operators, $u(\boldsymbol{x}, t)$ stands for the solution, and $g(\boldsymbol{x}, t)$ refers to input or source function. The aim of PINNs is to build a neural network $f_{\boldsymbol{w}}$ with parameters $\boldsymbol{w}$ to approximate the true solution u. Hence, the loss function of PINN for solving a general PDE is based on *residuals*, including PDE residual and data residual. The PDE residual measures the difference

between the neural network solution and the true solution, expressed as

$$\mathcal{L}_P = \sum_j \left| \mathcal{N}\left[f_{\boldsymbol{w}}\left(\boldsymbol{x}_p^{(j)}, t_p^{(j)}\right)\right] - g\left(\boldsymbol{x}_p^{(j)}, t_p^{(j)}\right)\right|^2, \quad \mathcal{L}_D = \sum_j \left| f_{\boldsymbol{w}}\left(\boldsymbol{x}_d^{(j)}, t_d^{(j)}\right) - u_d^{(j)}\right|^2. \quad (3)$$

Here, $\{\boldsymbol{x}_p^{(j)}, t_p^{(j)}\}_{j=1}^{N_p}$ used in $\mathcal{L}_P$ are selected collocation points for enforcing PDE structure. For $\mathcal{L}_D$, the dataset $\{\boldsymbol{x}_d^{(j)}, t_d^{(j)}, u_d^{(j)}\}_{j=1}^{N_d}$ with $u_d^{(j)} = u(\boldsymbol{x}_d^{(j)}, t_d^{(j)})$ denote the training data on $u(\boldsymbol{x}, t)$ [37]. Thus, the total loss is putting all residuals together, i.e. $\mathcal{L} = \mathcal{L}_P + \mathcal{L}_D$. By embedding physical principles into the learning process, PINNs serve as a versatile tool that only requires a small amount of data to tackle the computationally complex problem.

2.1 Related Works

Prior literature related to improving the optimization efficiency of VQAs can be classified into three main classes, i.e., *measurement grouping*, *initializer design*, and *prediction of training dynamics*. Since the first two classes are complementary to PALQO, we defer the explanations to Appendix B.

The third class aims to harness learning models to approximate the training process. Some works inspired by meta-learning utilize the recurrent neural network to learn a sequential update rule in a heuristic manner [34, 35]. Nevertheless, the memory bottleneck and training instability of the recurrent neural network would lead to it being underwhelming [68]. Recent work proposed QuACK, involving linear dynamics approximation and nonlinear neural embedding, to accelerate the optimization [36]. However, the prediction phase requires estimating the energy loss of each step to find the optimal parameters, which is not friendly for large-scale problems. To overcome these limitations, the proposed PALQO uniquely approximates VQA training dynamics using a nonlinear PDE, embedding the dynamical laws directly into the learning process. In this way, it offers *deeper physical insight* and achieves *superior performance* through principled model-guided optimization.

3 PALQO: physics-informed model for accelerating quantum optimization

In this section, we first formally define the problem of learning the training dynamics of VQAs as nonlinear PDE problems in Sec. 3.1. Then, in Sec. 3.2, we introduce PALQO to solve this PDE via a tailored PINN-based model, where the optimized solutions correspond to the optimization trajectory of VQAs, followed by a generalization error analysis.

3.1 Reformulating the optimization of VQA as a PDE problem

Recall the optimization of VQEs in Sec. 2. As it is costly in querying $\boldsymbol{\theta}^{(t)}$ of each step t, it is demanded to develop a protocol that only learns from a few trajectory data $\{\boldsymbol{\theta}^{(t)}\}_{t=1}^{\tau}$ to classically predict future steps, thereby avoiding prohibitive resource costs without compromising accuracy. To achieve this goal, we start by revisiting the gradient descent dynamics of VQEs. The updating rules of parameters $\boldsymbol{\theta}$ with learning rate η at step t is given by $\delta\boldsymbol{\theta} = \boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)} = -\eta\partial\mathcal{E}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$, where $\partial\mathcal{E}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ is estimated through phase shift rule shown in Eq. (2). Suppose η is infinitesimally small, the following ordinary differential equation, a.k.a., gradient flow, characterizes how parameters change in continuous time, i.e.,

$$\frac{\partial\boldsymbol{\theta}}{\partial t} = -\frac{\partial\mathcal{E}}{\partial\boldsymbol{\theta}}. \quad (4)$$

Besides, we can similarly define the dynamics of $\mathcal{E}$ in a general form under Taylor expansion as

$$\frac{\partial\mathcal{E}}{\partial t} = -\sum_i \left(\frac{\partial\mathcal{E}}{\partial\boldsymbol{\theta}_i}\right)^2 + \frac{\eta}{2}\sum_{j,k}\frac{\partial^2\mathcal{E}}{\partial\boldsymbol{\theta}_j\partial\boldsymbol{\theta}_k}\frac{\partial\mathcal{E}}{\partial\boldsymbol{\theta}_j}\frac{\partial\mathcal{E}}{\partial\boldsymbol{\theta}_k} + \mathcal{O}(\eta^2), \quad (5)$$

where the first term of RHS of Eq. (5) is termed as quantum neural tangent Kernel (QNTK) [39, 40, 69], which captures the sensitivity of outputs to parameter changes, shaping how the gradient flow evolves in parameter space. The second term involves the Hessian matrix of $\mathcal{E}$ that reflects the local curvature of the cost function in the optimization landscape. Thus, tackling the problem of learning optimization trajectory can be recast to solve PDEs presented in Eqs. (4) and (5). This reformulation

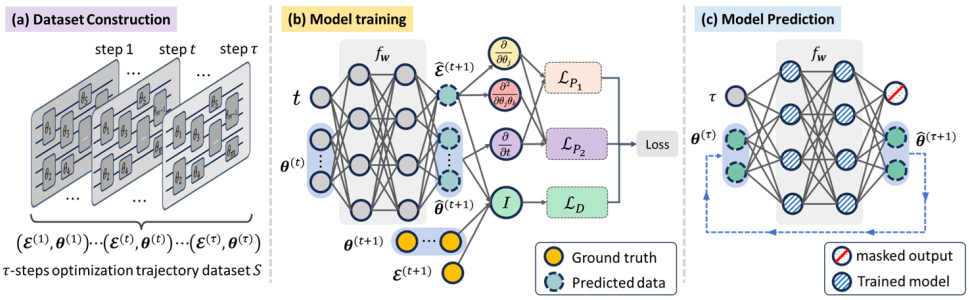

Figure 1: **Framework of PALQO**. (a) Dataset construction: the tunable angles $\boldsymbol{\theta}$ and corresponding loss $\mathcal{E}$ from a VQE over τ optimization steps are collected, forming a sequential training dataset that captures the optimization trajectory over time. (b) Model training: A PINN $f_{\boldsymbol{w}}$ is trained by minimizing the loss $\mathcal{L} = \lambda_{P_1}\mathcal{L}_{P_1} + \lambda_{P_2}\mathcal{L}_{P_2} + \lambda_D\mathcal{L}_D$. (c) Prediction: starting with the last step $\boldsymbol{\theta}^{(\tau)}$, the trained $f_{\boldsymbol{w}}$ is used to recurrently predict parameters, mimicking the optimization process without access to the quantum device.

provides a powerful framework and allows us to leverage a rich set of tools developed for PDEs to understand the underlying behavior of the optimization process in large-scale VQEs.

Remark. In Appendix C.1, we elucidate the derivation of Eq. (5) and its relation to QNTK. Besides, the above reformulation can be effectively extended to broader VQAs such as QNNs.

3.2 Implementation of PALQO

In light of the above reformulation, we propose PALQO based on PINN introduced in Eq. (3) to learn the optimization trajectory of large-scale VQEs. Conceptually, once PALQO attains a low training error, it can predict the optimization path, which substantially reduces the measurement cost as considered in large-scale VQEs. As such, it enhances scalability and resource efficiency by minimizing the need for extensive quantum circuit evaluations while maintaining high fidelity in modeling complex quantum optimization.

An overview of our protocol is depicted in Fig. 1, which consists of consists of three stages: *Dataset construction, Model training, and Model prediction*. In Fig. 1 (a), it starts by generating a dataset $\mathcal{S} = \{\boldsymbol{\theta}^{(t)}, \mathcal{E}^{(t)}\}_{t=1}^{\tau}$ corresponding to trajectory data consisting of $\boldsymbol{\theta}$ and energy loss $\mathcal{E}$ over τ optimization steps collected from a quantum device. Then, as shown in Fig. 1 (b), PALQO utilizes the collected $\mathcal{S}$ to train a PINN-based model $f_{\boldsymbol{w}}$ to capture the underlying optimization dynamics. Given the trained $f_{\boldsymbol{w}}$, as shown in Fig. 1 (c), it can recursively predict the parameters $\boldsymbol{\theta}$ of the future steps until it convergence, i.e. $|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t+1)}| + |\mathcal{E}^{(t)} - \mathcal{E}^{(t+1)}| \leq \varsigma$ with ς being a small constant.

Dataset construction. To mimic the dynamic of VQE optimization, we need first perform τ steps optimization via gradient descent to collect a sequential trajectory as the training data, including loss $\{\mathcal{E}^{(1)}, \mathcal{E}^{(2)}, \ldots, \mathcal{E}^{(\tau)}\}$ and $\{\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots, \boldsymbol{\theta}^{(\tau)}\}$ where $\boldsymbol{\theta}^{(j)} = (\boldsymbol{\theta}_1^{(j)}, \cdots, \boldsymbol{\theta}_p^{(j)}) \in \mathbb{R}^p$ is the parameters at j-th epoch. Notably, assume a VQE with a total T optimization steps, PALQO only needs $\tau \ll T$ steps to construct the training dataset $\mathcal{S}$. This is because the PINN-based model leverages strong inductive biases from the dynamical laws, such that it does not need to infer fundamental principles from scratch, allowing the model to generalize well from limited data.

Model training. Once the dataset is collected, PALQO employs a deep neural network $f_{\boldsymbol{w}}$ that takes $\{(t, \boldsymbol{\theta}^{(t)})\}_{t=1}^{\tau}$ comprising t-th time step and trajectory data $\boldsymbol{\theta}^{(t)}$ as the input, and predict the loss and parameters for the $(t+1)$-th step, i.e.,$(\mathcal{E}^{(t+1)}, \boldsymbol{\theta}^{(t+1)})$. Refer to Appendix C for the details about the neural architecture of $f_{\boldsymbol{w}}$. Denote the prediction of $f_{\boldsymbol{w}}$ for the t-th step as $(\hat{\boldsymbol{\theta}}^{(t)}, \hat{\mathcal{E}}^{(t)})$. Through leveraging the automatic differentiation capabilities of neural networks, the derivatives of outputs with respect to inputs, i.e., $\partial\hat{\boldsymbol{\theta}}/\partial t$ and $\partial\hat{\mathcal{E}}/\partial\boldsymbol{\theta}$, can be efficiently computed on classical devices. This efficiency enables the direct incorporation of dynamical law constraints, as described in Eqs. (4) and (5), into the loss function. In this regard, we devise the loss function of PALQO as

$$\mathcal{L} = \lambda_D\mathcal{L}_D + \lambda_{P_1}\mathcal{L}_{P_1} + \lambda_{P_2}\mathcal{L}_{P_2}, \tag{6}$$

where $\{\lambda_D, \lambda_{P_1}, \lambda_{P_2}\}$ denote hyperparameters to balance the data-driven loss $\mathcal{L}_D$ and two PDE residual losses $\mathcal{L}_{P_1}$ and $\mathcal{L}_{P_2}$. In particular, the data residual loss is defined as $\mathcal{L}_D = \sum_{t=1}^{\tau}(\mathcal{E}^{(t)} -$

$\hat{\mathcal{E}}^{(t)})^2 + \sum_{t=1}^{\tau}(\boldsymbol{\theta}^{(t)} - \hat{\boldsymbol{\theta}}^{(t)})^2$, aiming to capture the temporal changes of $\mathcal{E}$ and $\boldsymbol{\theta}$ among τ steps. Meanwhile, the two PDE residual losses aim to enforce the PINN to capture the evolution of VQE optimization dynamics through the underlying derivative structure of the loss landscape. Following Eq. (4), the explicit form of the first PDE loss is $\mathcal{L}_{P_1} = \sum_{t=1}^{\tau}(\sum_{j=1}^{p}(\partial\hat{\boldsymbol{\theta}}_j^{(t)}/\partial t + \partial\hat{\mathcal{E}}^{(t)}/\partial\boldsymbol{\theta}_j^{(t)}))^2$. By focusing on the first two orders of the derivatives in Eq. (5), the second PDE loss yields

$$\mathcal{L}_{P_2} = \sum_{t=1}^{\tau}\Big(\frac{\partial\hat{\mathcal{E}}^{(t)}}{\partial t} + \sum_{j=1}^{p}\Big(\frac{\partial\hat{\mathcal{E}}^{(t)}}{\partial\boldsymbol{\theta}_j^{(t)}}\Big)^2 - \frac{\eta}{2}\sum_{j,k=1}^{p}\frac{\partial^2\hat{\mathcal{E}}^{(t)}}{\partial\boldsymbol{\theta}_j^{(t)}\partial\boldsymbol{\theta}_k^{(t)}}\frac{\partial\hat{\mathcal{E}}^{(t)}}{\partial\boldsymbol{\theta}_j^{(t)}}\frac{\partial\hat{\mathcal{E}}^{(t)}}{\partial\boldsymbol{\theta}_k^{(t)}}\Big)^2. \tag{7}$$

The model $f_{\boldsymbol{w}}$ is optimized by minimizing $\mathcal{L}$ in Eq. (6) via a gradient-based optimizer Adam [70].

Model prediction. As the trained $f_{\boldsymbol{w}}$ can not only approximate the solution of the underlying PDE but also capture temporal dependencies of the trajectory data, we are able to recurrently predict the upcoming updates of the $\boldsymbol{\theta}$. As shown in Fig. 1 (c), by passing $(\tau, \boldsymbol{\theta}^{(\tau)})$ through the trained $f_{\boldsymbol{w}}$ and masking the $\hat{\mathcal{E}}^{(\tau)}$ node, we can obtain the predicted data $\hat{\boldsymbol{\theta}}^{(\tau+1)}$, and then fed $(\tau+1, \hat{\boldsymbol{\theta}}^{(\tau+1)})$ as the input back to the network to make the followed prediction. It is worth noting that directly making long-term forecasts to reach the optimal solution of the target problem is exceedingly challenging. To that end, we employ the non-overlapping sliding windows to enhance the network for long-term prediction. More details on the prediction process can be found in Appendix C.

Remark. The second PDE residual loss $\mathcal{L}_{P_2}$ can be extended to arbitrary higher orders. Empirical results indicate that a second-order approximation offers a sufficient balance between accuracy and computational cost for modeling the optimization dynamics of the VQEs studied herein. Moreover, while we mainly focus on learning the training process of VQE, our model can be efficiently extended to more general tasks such as quantum machine learning [19, 20, 22], quantum simulation [71–73], and quantum optimization [74, 75] by slightly modifying the Eqs. (4) and (5). See Appendix F for details.

Building upon prior work on the error analysis of PINNs [76], we conduct the analysis of the Lipschitz constant bound for PALQO and derive a corollary to establish a generalization bound for PALQO applied to VQEs. An informal statement of the derived generalization bound is provided below, where the formal statement and the related proof are deferred to Appendix D.

Corollary 3.1. *(Informal) When utilizing PALQO, whose PINN is constructed by a L layer tanh neural network with most W width of each layer and trained over τ data samples, to approximate the solution of PDE that describes training dynamics of a VQE with p tunable parameters $\boldsymbol{\theta}$, with probability at least $1-\gamma$, its the generalization error is upper bounded by*

$$\tilde{\mathcal{O}}\left(\sqrt{\frac{pL^2W^2}{\tau}\left(\ln\left(\frac{p}{\epsilon}\right) + \ln\left(\frac{1}{\gamma}\right)\right)}\right). \tag{8}$$

The achieved results indicate that for any $\epsilon > 0$, the number of training examples scales at most polynomially in p, L, and W is sufficient to guarantee a well-bounded generalization error. This warrants the applicability of PALQO in large-scale scenarios.

4 Experiments

To evaluate the practical performance of the proposed PALQO framework, we apply it to two representative quantum applications: finding ground state energy of many-body quantum system and molecules in quantum chemistry, which have broad applicability in understanding many-body physical phase transitions [77–79] and simulation of complex electronic structures of molecular systems in drug design and discovery [15, 80, 81]. The concrete settings are elucidated below.

Many-body quantum system. A many-body quantum system consists of interacting quantum particles whose collective behavior and correlations lead to complex phenomena beyond single-particle descriptions. Here, we consider three typical many-body quantum systems. 1) **Transverse-field Ising model (TFIM)** describes spin particles on a lattice interacting via nearest-neighbor coupling and subject to a transverse magnetic field, whose Hamiltonian is typically in a form of $H_{\text{TFIM}} = -J\sum_j Z_j \otimes Z_{j+1} - h\sum_j X_j$, where Z_j and X_j refer to the Pauli matrices Z and X applied on the j-th qubit, respectively. 2) **Quantum Heisenberg model** also describes the spin

particles on a lattice, but spin-spin interactions occur along all spatial directions. Its Hamiltonian can be represented as $H_{\text{QH}} = -1/2 \sum_j (J_x X_j X_{j+1} + J_y Y_j Y_{j+1} + J_z Z_j Z_{j+1} + h Z_j)$. 3) **Bond-alternating XXZ model** is an anisotropic variant of the Heisenberg model with unequal coupling strengths in the transverse and longitudinal directions. The Hamiltonian is given by $H_{\text{XXZ}} = \sum_{j=odd} J(X_j X_{j+1} + Y_j Y_{j+1} + \delta Z_j Z_{j+1}) + \sum_{j=even} J'(X_j X_{j+1} + Y_j Y_{j+1} + \delta Z_j Z_{j+1})$. The coefficients $J, h, J_x, J_y, J_z, J', \delta$ within the explored Hamiltonians represent the coupling strength that determines the ground state properties and phase transitions.

Quantum chemistry. The Hamiltonian of a molecule describes the total energy of its electrons and nuclei and serves as the fundamental operator for determining the molecule's electronic structure in quantum chemistry. The general form of the molecule Hamiltonian can be presented as $H = \sum_j h_j P_j$ where P_j represents tensor products of Pauli matrices, and h_j are the associated real coefficients. Here, we select a widely studied molecule—LiH, and a relatively large and challenging BeH_2 molecule as the target molecules [8, 53, 82]. We generate these molecule Hamiltonians with Openfermion [83]. Refer to Appendix E for more details about the molecule experiments.

4.1 Experimental Setup

We employ three standard ansatzes, i.e., hardware-efficient ansatz (HEA) [84–86], Hamiltonian variable ansatz (HVA) [87–89], unitary coupled cluster with single and double excitations ansatz (UCCSD) [90–92], to implement VQE for the different Hamiltonians mentioned above. These ansatzes adopt a layered architecture. Refer to Appendix E.2 for the implementation of these ansatzes. For all ansatzes, their initial parameters $\boldsymbol{\theta}^{(0)}$ are uniformly sampled from $[0, 1]$, following the strategy adopted in QuACK [36]. The gradient descent is set as the default optimizer.

For implementing PALQO, we randomly initialize the parameters $\boldsymbol{w}$ of the neural network $f_{\boldsymbol{w}}$ from $[-1, 1]$ and employ the Adam as the optimizer, where the architectures of $f_{\boldsymbol{w}}$ are listed in Appendix C.2. To improve the training stability and convergence, we utilize a linear decay strategy to adaptively adjust the learning rate during training. Besides, the weight hyperparameters in loss function $\lambda_{P_1}, \lambda_{P_2}, \lambda_D$, are set as $1.0, 1.0$ and 10^{-4}, respectively.

Benchmark models. To show the outperformance of PALQO against the state-of-the-art methods, we introduce the following baseline and benchmark. First, we use a *vanilla VQE* as the baseline since it provides a well-established reference point to evaluate improvements in accuracy, convergence, and efficiency. Second, we select a *LSTM-based model* [34] as a benchmark since it provides a strong reference for evaluating methods in modeling the temporal dependencies and iterative dynamics of optimization trajectories. Third, we pick *QuACK* [36] as another benchmark as it represents an advanced approach that learns surrogate dynamics of VQAs by embedding Koopman operator-based linear representations into nonlinear neural networks. The implementation of these reference models is deferred to Appendix E.3.

Evaluation metrics. To quantify the performance of PALQO in accuracy and efficiency, we consider the following metrics, 1) **Accuracy.** we define the accuracy as how close the estimated energy $\hat{E}$ is to a given target energy E of a quantum system, i.e. $\Delta E = |\hat{E} - E|$. 2) **Efficiency.** we define the speedup ratio as $\alpha = \mathcal{I}_P / \mathcal{I}_V$, where $\mathcal{I}_P$ and $\mathcal{I}_V$ refer to the number of iterations required by the baseline method (vanilla VQE) and PALQO or other benchmark models, respectively, to achieve an acceptable accuracy a. Specifically, we set $a \leq 10^{-3}$.

4.2 Experimental Results

We next evaluate the performance of PALQO and other reference models when applied to the aforementioned Hamiltonians under different settings.

PALQO significantly reduces the measurement overhead. Here, we utilize the number of measurements incurred during the optimization as a quantum resource measure to explore the performance of PALQO and the other benchmark models when applied to 20 qubits TFIM with HEA, 20 qubits Heisenberg model with HVA, and 14 qubits BeH_2 with UCCSD ansatz, The number of parameters for each case are $120, 180, 90$, respectively. As shown in Tab. 1, PALQO achieves significant quantum resource efficiency in aforementioned tasks, with around 90% average reduction in measurement overhead while preserving ΔE around 10^{-3}.

These substantial savings stem from two key factors: 1) compared to the vanilla VQE, PALQO leverages PINN to predict parameter updates, thereby reducing reliance on frequent quantum measurements; 2) the rapid convergence on the classical side enables further reduction in quantum resource expenditure.

Table 1: The number of quantum measurement shots ($\times 10^8$) required for TFIM, Heisenberg model, and BeH_2.

SYSTEM SIZE	$H_{\text{TFIM}} = 20$	$H_{\text{HQ}} = 20$	$H_{\text{BEH}_2} = 14$
VANILLA VQE	10.97	21.66	464.3
LSTM	3.126	14.49	312.4
QUACK	5.217	14.15	461.8
PALQO	**1.535**	**5.749**	**28.01**

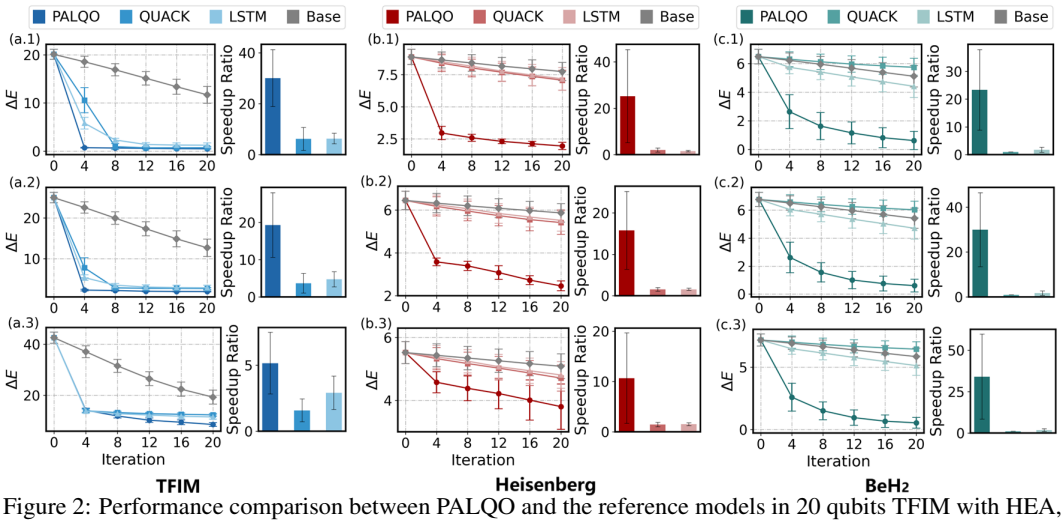

Figure 2: Performance comparison between PALQO and the reference models in 20 qubits TFIM with HEA, 20 qubits Heisenberg model with HVA, and 14 qubits BeH2 with UCCSD ansatz. Each subplot comprises a ΔE curve over iterations performed on a quantum device, along with a bar chart depicting the speedup ratios achieved by PALQO and competing models. The left column illustrates results for TFIM with $J/h = \{2, 1, 0.5\}$. The central column shows results for Heisenberg model with $J_x = J_y = h = 1, J_z = \{0.5, 1, 2\}$. The right column displays the model performance on BeH2 with the bond length $b = \{1.3, 1.4, 1.5\}$.

PALQO outperforms benchmark models in accuracy and efficiency. The performance comparisons of PALQO on 20 qubits TFIM with HEA, 20 qubits Heisenberg model with HVA, and 14 qubits BeH2 with UCCSD ansatz for varying structural parameters are presented in Fig. 2. In particular, we observed that PALQO consistently outperformed, up to 30x speedup and lower $\Delta E = |\hat{E} - E|$ around 10^{-3}, like the case of TFIM with $J/h = 2$ and HEA ansatz, compared to the other evaluated approaches. Furthermore, as the PALOQ predicts the future optimization steps on classical hardware, it exhibited a faster rate of convergence, achieving a substantial reduction in ΔE within fewer iterations performed on a quantum device, compared to the baseline methods. Although the speedup ratio of PALQO has a relatively large variance, its minimum value remains comparable to the average performance of the other approaches. Refer to Appendix F for the results of XXZ model and LiH.

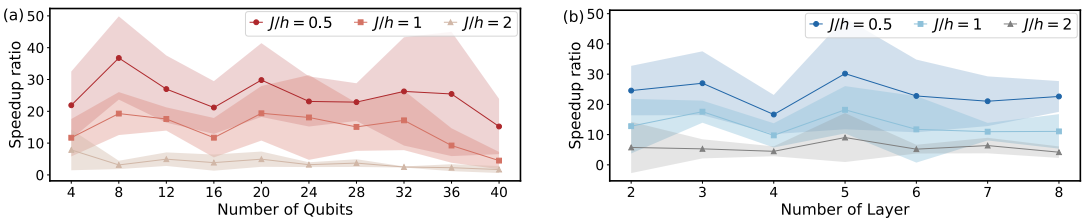

Figure 3: Scalability analysis of PALQO on TFIMs. (a) The speedup ratio achieved by PALQO in modeling VQE training dynamics with a fixed HEA, ranging from 4 to 40 qubits. (b) The speedup ratio of PALQO with a fixed system size of 12 qubits, assessed under increasing HEA ansatz layers from 2 to 8.

Scalability of PALQO. We next investigate the scalability of PALQO on the TFIM with HEA, examining its performance in increasing system sizes (from $n = 4$ to $n = 40$) and the number of ansatz layers (from 2 to 8). In Fig. 3, the results reveal that the speedup of PALQO is contingent upon the specific system configuration. Nevertheless, as shown in Fig. 3 (a), while the speedup ratio fluctuates with the number of qubits varying, it still achieves up to 30x speedup when $J/h = 0.5$. The lower speedup at $J/h = 2$ is due to a smaller energy gap between the ground and first excited states, making the optimization more challenging. Similar behavior also appears in Fig. 3 (b). This suggests that the performance benefits of PALOQ are maintained as the computational demands grow, indicating its potential for large-scale quantum optimization.

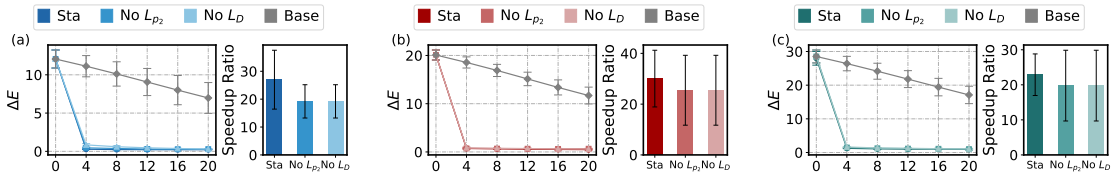

Figure 4: Ablation study on the loss function configuration in PALQO. The three panels show the performance on TFIM with $n = 12, 20, 28$ qubits, evaluating the impact of the components $\mathcal{L}_D$ and $\mathcal{L}_{P_2}$ in $\mathcal{L}$.

Ablation studies on loss function. We further evaluate the performance of PALOQ against variants where specific loss terms in Eq. (6) are removed in the task of TFIM with HEA ansatz. Specifically, we carried out separate ablation studies on the PDE residual and data residual components of the loss function shown in Fig. 4. We noticed that while both the PDE and data residual positively influence model performance, their contributions are not essential. These findings suggest that adopting low-order approximations during the construction of PALQO enables the retention of satisfactory speedup while simultaneously reducing the complexity of downstream model training and preventing the degradation of higher-order derivative information.

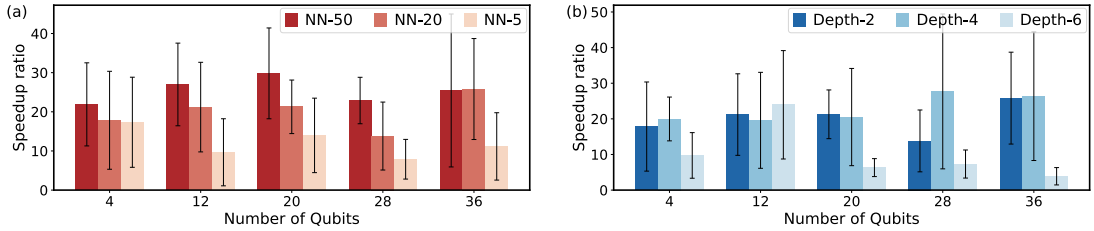

Figure 5: Performance of PALQO evaluated on the TFIM under different neural network architecture configurations. (a) Results obtained using 2-layer neural networks with varying hidden layer widths of $W = \{50p, 20p, 5p\}$, where p denotes the number of parameters $\boldsymbol{\theta}$. (b) Results with a fixed hidden layer width of $20p$, varying the number of hidden layers from 2 to 6.

Performance on varying the size of PALQO. To investigate the influence of varying neural network sizes (width W and depth L) within PALQO on its performance, we conducted tests on the TFIM with HEA ansatz and $p = 6n$ parameters, where n is the system size varying from 4 to 36. In Fig. 5 (a), we observed that an increase in the width of the hidden layers leads to a corresponding improvement in speedup ratio. However, an inverse phenomenon occurs in Fig. 5 (b), further increasing the neural network depth does not effectively enhance PALQO's performance, which may be related to the vanishing gradient phenomenon, where higher-order derivative information tends to diminish as the neural network becomes deeper [93]. These observations provide guidance for the neural network design in PALQO, indicating that increasing hidden layer width should be prioritized.

The impact of the number of training samples on model prediction. We performed experiments on 12-qubit TFIMs with HEA ansatz using various sample sizes to explore how the number of training samples affects the performance.In Fig. 6, the results validate that PALQO can achieve satisfactory performance even with a limited number of

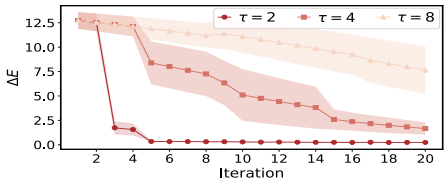

Figure 6: Performance comparison of PALQO when trained with 2, 4, and 8 data samples.

training samples. Such data efficiency arises from the direct integration of the physical constraints imposed by the governing PDE into the loss function of the neural network, a core characteristic of PINNs [63].

5 Conclusion

In this study, we devised PALQO towards optimizing large-scale VQAs given restrictive quantum resources. In contrast to previous studies, we derive PALQO from reformulating the training dynamics as a nonlinear PDE and using PINN to approximate the solution, and also provide a generalization analysis. Extensive numerical experiments up to 40 qubits validate the effectiveness of PALQO. Although it is still uncertain whether PALQO can scale to the regime where quantum hardware decisively outperforms classical methods, its results at the currently accessible scale are highly encouraging.

Limitations and future works PALQO reduces the need for repeated quantum gradient evaluations by learning the optimization path classically. While this lowers the number of quantum queries compared to vanilla VQE, it may diminish some quantum advantages. Besides, mitigating the high variance in speedup ratios is crucial for achieving more stable and reliable performance. One future research direction is to incorporate adaptive strategies and variance reduction techniques to achieve this goal and further unlock its potential.

6 Acknowledge

We thank Yusen Wu for the helpful discussions and the anonymous reviewers for their constructive feedback and valuable suggestions. This work was supported by Innovation Program for Quantum Science and Technology (Grant No. 2023ZD0300200), the National Natural Science Foundation of China NSAF (Grant No. U2330201 and No. 92265208), Sichuan Science and Technology Program (Grant No. 2025YFHZ0336). Y. D. acknowledges the support from the SUG grant of NTU. The numerical experiments in this work were supported by the High-Performance Computing Platform of Peking University.

References

[1] Google Quantum AI et al. Quantum error correction below the surface code threshold. *Nature*, 638(8052):920, 2024.

[2] John Preskill. Beyond nisq: The megaquop machine. *ACM Transactions on Quantum Computing*, 6(3):1–7, 2025.

[3] Trond I Andersen, Nikita Astrakhantsev, Amir H Karamlou, Julia Berndtsson, Johannes Motruk, Aaron Szasz, Jonathan A Gross, Alexander Schuckert, Tom Westerhout, Yaxing Zhang, et al. Thermalization and criticality on an analogue–digital quantum simulator. *Nature*, 638(8049):79–85, 2025.

[4] Dave Wecker, Matthew B Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, 2015.

[5] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[6] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.

[7] A Peruzzo, J Mcclean, P Shadbolt, MH Yung, X Zhou, P Love, A Aspuru-Guzik, and J O'Brien. A variational eigenvalue solver on a quantum processor. *Nature communications*, 5.

[8] Quoc Hoan Tran, Shinji Kikuchi, and Hirotaka Oshima. Variational denoising for variational quantum eigensolver. *Physical Review Research*, 6(2):023181, 2024.

[9] Harper R Grimsley, George S Barron, Edwin Barnes, Sophia E Economou, and Nicholas J Mayhall. Adaptive, problem-tailored variational quantum eigensolver mitigates rough parameter landscapes and barren plateaus. *npj Quantum Information*, 9(1):19, 2023.

[10] Jonas Jäger and Roman V Krems. Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines. *Nature Communications*, 14(1):576, 2023.

[11] Xinyu Ye, Ge Yan, and Junchi Yan. Towards quantum machine learning for constrained combinatorial optimization: a quantum qap solver. In *International Conference on Machine Learning*, pages 39903–39912. PMLR, 2023.

[12] Jinkai Tian, Xiaoyu Sun, Yuxuan Du, Shanshan Zhao, Qing Liu, Kaining Zhang, Wei Yi, Wanrong Huang, Chaoyue Wang, Xingyao Wu, et al. Recent advances for quantum neural networks in generative learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12321–12340, 2023.

[13] Google AI Quantum, Collaborators*†, Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Sergio Boixo, Michael Broughton, Bob B Buckley, et al. Hartree-fock on a superconducting qubit quantum computer. *Science*, 369(6507):1084–1089, 2020.

[14] Ashutosh Kumar, Ayush Asthana, Vibin Abraham, T Daniel Crawford, Nicholas J Mayhall, Yu Zhang, Lukasz Cincio, Sergei Tretiak, and Pavel A Dub. Quantum simulation of molecular response properties in the nisq era. *Journal of Chemical Theory and Computation*, 19(24): 9136–9150, 2023.

[15] Shaojun Guo, Jinzhao Sun, Haoran Qian, Ming Gong, Yukun Zhang, Fusheng Chen, Yangsen Ye, Yulin Wu, Sirui Cao, Kun Liu, et al. Experimental quantum computational chemistry with optimized unitary coupled cluster ansatz. *Nature Physics*, pages 1–7, 2024.

[16] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[17] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuleti, and M. D. Lukin. Quantum optimization of maximum independent set using rydberg atom arrays. *Science*, 376(6598): 1209–1215, 2022. doi: 10.1126/science.abo6587. URL https://www.science.org/doi/abs/10.1126/science.abo6587.

[18] Amira Abbas, Andris Ambainis, Brandon Augustino, Andreas Bärtschi, Harry Buhrman, Carleton Coffrin, Giorgio Cortiana, Vedran Dunjko, Daniel J Egger, Bruce G Elmegreen, et al. Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, pages 1–18, 2024.

[19] Vojtoch Havlcek, Antonio D. Corcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, Mar 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-0980-2. URL https://doi.org/10.1038/s41586-019-0980-2.

[20] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, Sep 2021. ISSN 1745-2481. doi: 10.1038/s41567-021-01287-z. URL https://doi.org/10.1038/s41567-021-01287-z.

[21] Yuxuan Du, Yibo Yang, Dacheng Tao, and Min-Hsiu Hsieh. Problem-dependent power of quantum neural networks on multiclass classification. *Physical Review Letters*, 131(14): 140601, 2023.

[22] Yaswitha Gujju, Atsushi Matsuo, and Rudy Raymond. Quantum machine learning on near-term quantum devices: Current state of supervised and unsupervised techniques for real-world applications. *Physical Review Applied*, 21(6):067001, 2024.

[23] Yuxuan Du, Xinbiao Wang, Naixu Guo, Zhan Yu, Yang Qian, Kaining Zhang, Min-Hsiu Hsieh, Patrick Rebentrost, and Dacheng Tao. Quantum machine learning: A hands-on tutorial for machine learning practitioners and researchers. *arXiv preprint arXiv:2502.01146*, 2025.

[24] Amira Abbas, Robbie King, Hsin-Yuan Huang, William J Huggins, Ramis Movassagh, Dar Gilboa, and Jarrod McClean. On quantum backpropagation, information reuse, and cheating measurement collapse. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[26] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, 2022.

[27] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F Izmaylov. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *The Journal of chemical physics*, 152(12), 2020.

[28] Tzu-Ching Yen, Aadithya Ganeshram, and Artur F Izmaylov. Deterministic improvements of quantum measurements with grouping of compatible operators, non-local transformations, and covariance estimates. *npj Quantum Information*, 9(1):14, 2023.

[29] Bujiao Wu, Jinzhao Sun, Qi Huang, and Xiao Yuan. Overlapped grouping measurement: A unified framework for measuring quantum states. *Quantum*, 7:896, 2023.

[30] Alexey Galda, Xiaoyuan Liu, Danylo Lykov, Yuri Alexeev, and Ilya Safro. Transferability of optimal qaoa parameters between random graphs. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 171–180. IEEE, 2021.

[31] Shikun Zhang, Zheng Qin, Yongyou Zhang, Yang Zhou, Rui Li, Chunxiao Du, and Zhisong Xiao. Diffusion-enhanced optimization of variational quantum eigensolver for general hamiltonians. *arXiv preprint arXiv:2501.05666*, 2025.

[32] Ricard Puig, Marc Drudis, Supanut Thanasilp, and Zoë Holmes. Variational quantum simulation: A case study for understanding warm starts. *PRX Quantum*, 6:010317, Jan 2025. doi: 10.1103/PRXQuantum.6.010317. URL https://link.aps.org/doi/10.1103/PRXQuantum.6.010317.

[33] Yuxuan Du, Yan Zhu, Yuan-Hang Zhang, Min-Hsiu Hsieh, Patrick Rebentrost, Weibo Gao, Ya-Dong Wu, Jens Eisert, Giulio Chiribella, Dacheng Tao, et al. Artificial intelligence for representing and characterizing quantum systems. *arXiv preprint arXiv:2509.04923*, 2025.

[34] Guillaume Verdon, Michael Broughton, Jarrod R McClean, Kevin J Sung, Ryan Babbush, Zhang Jiang, Hartmut Neven, and Masoud Mohseni. Learning to learn with quantum neural networks via classical neural networks. *arXiv preprint arXiv:1907.05415*, 2019.

[35] Ankit Kulshrestha, Xiaoyuan Liu, Hayato Ushijima-Mwesigwa, and Ilya Safro. Learning to optimize quantum neural networks without gradients. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 1, pages 263–271. IEEE, 2023.

[36] Di Luo, Jiayu Shen, Rumen Dangovski, and Marin Soljacic. Quack: accelerating gradient-based quantum optimization with koopman operator learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[37] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[38] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[39] Junyu Liu, Francesco Tacchino, Jennifer R Glick, Liang Jiang, and Antonio Mezzacapo. Representation learning via quantum neural tangent kernels. *PRX Quantum*, 3(3):030323, 2022.

[40] Yehui Tang and Junchi Yan. Graphqntk: Quantum neural tangent kernel for graph data. *Advances in neural information processing systems*, 35:6104–6118, 2022.

[41] Xuchen You, Shouvanik Chakrabarti, Boyang Chen, and Xiaodi Wu. Analyzing convergence in quantum neural networks: deviations from neural tangent kernels. In *International Conference on Machine Learning*, pages 40199–40224. PMLR, 2023.

[42] Implementatio of physics-informed model for accelerating large-scale quantum optimization. https://github.com/Yajie-Hao/PALQO.

[43] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[44] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An introduction to quantum computing*. OUP Oxford, 2006.

[45] Fabio Valerio Massoli, Lucia Vadicamo, Giuseppe Amato, and Fabrizio Falchi. A leap among quantum computing and quantum neural networks: A survey. *ACM Computing Surveys*, 55(5): 1–37, 2022.

[46] Weikang Li, Zhi-de Lu, and Dong-Ling Deng. Quantum neural network classifiers: A tutorial. *SciPost Physics Lecture Notes*, page 061, 2022.

[47] 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.

[48] Saverio Monaco, Oriel Kiss, Antonio Mandarino, Sofia Vallecorsa, and Michele Grossi. Quantum phase detection generalization from marginal quantum neural network models. *Physical Review B*, 107(8):L081105, 2023.

[49] Jiaming Zhao, Wenbo Qiao, Peng Zhang, and Hui Gao. Quantum implicit neural representations. In *Proceedings of the 41st International Conference on Machine Learning*, pages 60940–60956, 2024.

[50] Yiming Huang, Xiao Yuan, Huiyuan Wang, and Yuxuan Du. Coreset selection can accelerate quantum machine learning models with provable generalization. *Physical Review Applied*, 22 (1):014074, 2024.

[51] Quynh T Nguyen, Louis Schatzki, Paolo Braccia, Michael Ragone, Patrick J Coles, Frederic Sauvage, Martin Larocca, and Marco Cerezo. Theory for equivariant quantum neural networks. *PRX Quantum*, 5(2):020328, 2024.

[52] Yifeng Peng, Xinyi Li, Samuel Yen-Chi Chen, Kaining Zhang, Zhiding Liang, Ying Wang, and Yuxuan Du. TITAN: A Trajectory-Informed Technique for Adaptive Parameter Freezing in Large-Scale VQE, 2025. arXiv:2509.15193v1.

[53] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 2022.

[54] M Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J Coles. Variational quantum state eigensolver. *npj Quantum Information*, 8(1):113, 2022.

[55] Stuart M Harwood, Dimitar Trenev, Spencer T Stober, Panagiotis Barkoutsos, Tanvi P Gujarati, Sarah Mostame, and Donny Greenberg. Improving the variational quantum eigensolver using variational adiabatic quantum computing. *ACM Transactions on Quantum Computing*, 3(1): 1–20, 2022.

[56] Alexis Ralli, Tim Weaving, Andrew Tranter, William M Kirby, Peter J Love, and Peter V Coveney. Unitary partitioning and the contextual subspace variational quantum eigensolver. *Physical Review Research*, 5(1):013095, 2023.

[57] Yuki Sato, Hiroshi C Watanabe, Rudy Raymond, Ruho Kondo, Kaito Wada, Katsuhiro Endo, Michihiko Sugawara, and Naoki Yamamoto. Variational quantum algorithm for generalized eigenvalue problems and its application to the finite-element method. *Physical Review A*, 108 (2):022429, 2023.

[58] Byungjoo Kim, Kang-Min Hu, Myung-Hyun Sohn, Yosep Kim, Yong-Su Kim, Seung-Woo Lee, and Hyang-Tag Lim. Qudit-based variational quantum eigensolver using photonic orbital angular momentum states. *Science Advances*, 10(43):eado3472, 2024.

[59] Xinbiao Wang, Junyu Liu, Tongliang Liu, Yong Luo, Yuxuan Du, and Dacheng Tao. Symmetric pruning in quantum neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=K96AogLDT2K.

[60] Albie Chan, Zheng Shi, Luca Dellantonio, Wolfgang Dür, and Christine A Muschik. Measurement-based infused circuits for variational quantum eigensolvers. *Physical Review Letters*, 132(24):240601, 2024.

[61] MWM Gamini Dissanayake and Nhan Phan-Thien. Neural-network-based approximations for solving partial differential equations. *communications in Numerical Methods in Engineering*, 10(3):195–201, 1994.

[62] Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9 (5):987–1000, 1998.

[63] Jiequn Han, Arnulf Jentzen, et al. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in mathematics and statistics*, 5(4):349–380, 2017.

[64] Tomohisa Okazaki, Takeo Ito, Kazuro Hirahara, and Naonori Ueda. Physics-informed deep learning approach for modeling crustal deformation. *Nature Communications*, 13(1):7092, 2022.

[65] Liheng Bian, Haoze Song, Lintao Peng, Xuyang Chang, Xi Yang, Roarke Horstmeyer, Lin Ye, Chunli Zhu, Tong Qin, Dezhi Zheng, et al. High-resolution single-photon imaging with physics-informed deep learning. *Nature Communications*, 14(1):5902, 2023.

[66] Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. Generative learning for forecasting the dynamics of high-dimensional complex systems. *Nature Communications*, 15(1):8904, 2024.

[67] Fujin Wang, Zhi Zhai, Zhibin Zhao, Yi Di, and Xuefeng Chen. Physics-informed neural network for lithium-ion battery degradation stable modeling and prognosis. *Nature Communications*, 15(1):4332, 2024.

[68] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.

[69] Li-Wei Yu, Weikang Li, Qi Ye, Zhide Lu, Zizhao Han, and Dong-Ling Deng. Expressibility-induced concentration of quantum neural tangent kernels. *Reports on Progress in Physics*, 87 (11):110501, 2024.

[70] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[71] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, 2019.

[72] Suguru Endo, Jinzhao Sun, Ying Li, Simon C Benjamin, and Xiao Yuan. Variational quantum simulation of general processes. *Physical Review Letters*, 125(1):010501, 2020.

[73] Christian Kokail, Christine Maier, Rick van Bijnen, Tiff Brydges, Manoj K Joshi, Petar Jurcevic, Christine A Muschik, Pietro Silvi, Rainer Blatt, Christian F Roos, et al. Self-verifying variational quantum simulation of lattice models. *Nature*, 569(7756):355–360, 2019.

[74] David Amaro, Carlo Modica, Matthias Rosenkranz, Mattia Fiorentini, Marcello Benedetti, and Michael Lubasch. Filtering variational quantum algorithms for combinatorial optimization. *Quantum Science and Technology*, 7(1):015021, 2022.

[75] Pablo Díez-Valle, Jorge Luis-Hita, Senaida Hernández-Santana, Fernando Martínez-García, Álvaro Díaz-Fernández, Eva Andrés, Juan José García-Ripoll, Escolástico Sánchez-Martínez, and Diego Porras. Multiobjective variational quantum optimization for constrained problems: an application to cash handling. *Quantum Science and Technology*, 8(4):045009, 2023.

[76] Tim De Ryck and Siddhartha Mishra. Error analysis for physics-informed neural networks (pinns) approximating kolmogorov pdes. *Advances in Computational Mathematics*, 48(6):79, 2022.

[77] Jiehang Zhang, Guido Pagano, Paul W Hess, Antonis Kyprianidis, Patrick Becker, Harvey Kaplan, Alexey V Gorshkov, Z-X Gong, and Christopher Monroe. Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. *Nature*, 551(7682):601–604, 2017.

[78] Shuo Liu, Ming-Rui Li, Shi-Xin Zhang, Shao-Kai Jian, and Hong Yao. Noise-induced phase transitions in hybrid quantum circuits. *Physical Review B*, 110(6):064323, 2024.

[79] Hirsh Kamakari, Jiace Sun, Yaodong Li, Jonathan J Thio, Tanvi P Gujarati, Matthew PA Fisher, Mario Motta, and Austin J Minnich. Experimental demonstration of scalable cross-entropy benchmarking to detect measurement-induced phase transitions on a superconducting quantum processor. *Physical Review Letters*, 134(12):120401, 2025.

[80] Huanjin Wu, Xinyu Ye, and Junchi Yan. Qvae-mole: The quantum vae with spherical latent variable learning for 3-d molecule generation. *Advances in Neural Information Processing Systems*, 37:22745–22771, 2024.

[81] Raul Conchello Vendrell, Akshay Ajagekar, Michael T Bergman, Carol K Hall, and Fengqi You. Designing microplastic-binding peptides with a variational quantum circuit–based hybrid quantum-classical approach. *Science Advances*, 10(51):eadq8492, 2024.

[82] Lila Cadi Tazi and Alex JW Thom. Folded spectrum vqe: A quantum computing method for the calculation of molecular excited states. *Journal of Chemical Theory and Computation*, 20 (6):2491–2504, 2024.

[83] Jarrod R McClean, Nicholas C Rubin, Kevin J Sung, Ian D Kivlichan, Xavier Bonet-Monroig, Yudong Cao, Chengyu Dai, E Schuyler Fried, Craig Gidney, Brendan Gimby, et al. OpenFermion: the electronic structure package for quantum computers. `https://github.com/quantumlib/OpenFermion`, 2025-02-12.

[84] Lorenzo Leone, Salvatore FE Oliviero, Lukasz Cincio, and Marco Cerezo. On the practical usefulness of the hardware efficient ansatz. *Quantum*, 8:1395, 2024.

[85] Xin Wang, Bo Qi, Yabo Wang, and Daoyi Dong. Entanglement-variational hardware-efficient ansatz for eigensolvers. *Physical Review Applied*, 21(3):034059, 2024.

[86] J-Z Zhuang, Y-K Wu, and L-M Duan. Hardware-efficient variational quantum algorithm in a trapped-ion quantum computer. *Physical Review A*, 110(6):062414, 2024.

[87] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. Exploring entanglement and optimization within the hamiltonian variational ansatz. *PRX quantum*, 1(2):020319, 2020.

[88] Chae-Yeun Park and Nathan Killoran. Hamiltonian variational ansatz without barren plateaus. *Quantum*, 8:1239, 2024.

[89] Xiaoyang Wang, Yahui Chai, Xu Feng, Yibin Guo, Karl Jansen, and Cenk Tüysüz. Imaginary hamiltonian variational ansatz for combinatorial optimization problems. *Physical Review A*, 111(3):032612, 2025.

[90] Rongxin Xia and Sabre Kais. Qubit coupled cluster singles and doubles variational quantum eigensolver ansatz for electronic structure calculations. *Quantum Science and Technology*, 6 (1):015001, 2020.

[91] Choy Boy, Maria-Andreea Filip, and David J Wales. Energy landscapes for the unitary coupled cluster ansatz. *Journal of Chemical Theory and Computation*, 21(4):1739–1751, 2025.

[92] Jiaqi Hu, Qingchun Wang, and Shuhua Li. Unitary block-correlated coupled cluster ansatz based on the generalized valence bond wave function for quantum simulation. *Journal of Chemical Theory and Computation*, 2025.

[93] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[94] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

[95] Martin Larocca, Nathan Ju, Diego García-Martín, Patrick J Coles, and Marco Cerezo. Theory of overparametrization in quantum neural networks. *Nature Computational Science*, 3(6): 542–551, 2023.

[96] Felix Truger, Johanna Barzen, Marvin Bechtold, Martin Beisel, Frank Leymann, Alexander Mandl, and Vladimir Yussupov. Warm-starting and quantum computing: A systematic mapping study. *ACM Computing Surveys*, 56(9):1–31, 2024.

[97] Daniel J Egger, Jakub Mareček, and Stefan Woerner. Warm-starting quantum optimization. *Quantum*, 5:479, 2021.

[98] Yahui Chai, Karl Jansen, Stefan Kühn, Tim Schwägerl, and Tobias Stollenwerk. Structure-inspired ansatz and warm start of variational quantum algorithms for quadratic unconstrained binary optimization problems. *arXiv preprint arXiv:2407.02569*, 2024.

[99] Ali Rad, Alireza Seif, and Norbert M Linke. Surviving the barren plateau in variational quantum circuits with bayesian learning initialization. *arXiv preprint arXiv:2203.02464*, 2022.

[100] Eric R Anschuetz and Bobak T Kiani. Quantum variational algorithms are swamped with traps. *Nature Communications*, 13(1):7760, 2022.

[101] Panagiotis Kl Barkoutsos, Jerome F Gonthier, Igor Sokolov, Nikolaj Moll, Gian Salis, Andreas Fuhrer, Marc Ganzhorn, Daniel J Egger, Matthias Troyer, Antonio Mezzacapo, et al. Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions. *Physical Review A*, 98(2):022322, 2018.

[102] R. A. Fisher. Iris. UCI Machine Learning Repository, 1936. DOI: https://doi.org/10.24432/C56C76.

[103] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

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 main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   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: we have discussed the limitations of our work in Conclusion section.

   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: [Yes]

Justification: we provide the full set of assumptions and a complete proof.

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: we have disclosed 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

   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: we provide the open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results.

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: we specify all the training and test details to help understand the results.

   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: [Yes]

   Justification: we report error bars that provide the information about the statistical significance of the experiments

   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: we provide the related computer resources information in Appendix.

   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: our work conducted in the paper conform with the NeurIPS Code of Ethics.

   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: there is no societal impact of the work performed.

    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: our work poses no 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: CC-BY 4.0

    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: [NA]

    Justification: the paper does not release new assets.

    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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.