

Technical Appendices and Supplementary Material

To facilitate a thorough understanding of our work, this appendix is organized as follows. First, in Section A, we introduce the foundational concepts of quantum computing and variational quantum algorithms (VQAs), which form the basis of our work. Next, we review the related works focused on improving the optimization efficiency of VQAs in Section B. Then, we detail the implementation of the proposed PALQO, including its connection to the quantum neural tangent kernel (QNTK) and a breakdown of its design components in Section C. Subsequently, we present the theoretical analysis, covering both generalization error bounds and Lipschitz constant bounds for PALQO in Section D. In addition, we list the experimental details, including computational resources, variational ansätze used in VQE tasks, benchmark descriptions, and experimental setups in Section E. Finally, in Section F, we supplement the main results with additional numerical experiments, showcasing PALQO's performance on XXZ and LiH systems, a quantum machine learning task, and the robustness under noise. Besides, we also discussed that it can be complementary to existing approaches, such as measurement grouping, to further improve the optimization efficiency. Finally, we discuss the limitations of the proposed method in Section F.

A Quantum Computing and Variational Quantum Algorithms

A.1 Basic concepts of quantum computing

Quantum State In quantum computing, the quantum state that stores the information about the physical system is the essential element to be manipulated for computing. We usually describe it as a normalized complex vector in Hilbert space \mathcal{H} by Dirac notation, i.e. $|\psi\rangle \in \mathbb{C}^d$ ($\langle\psi|$ denotes the conjugate transpose of $|\psi\rangle$). For a single-qubit system, as the space $\mathcal{H} = \text{span}(|0\rangle, |1\rangle)$ where $|0\rangle = [1, 0]^\top$ and $|1\rangle = [0, 1]^\top$, the quantum state $|\psi\rangle$ can be expressed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, $|\alpha|^2 + |\beta|^2 = 1$. Similarly, since the Hilbert space \mathcal{H} of n -qubit system spanned by $\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_n$, an n -qubit quantum state $|\psi\rangle$ can be written as $|\psi\rangle = \sum_j \lambda_j |\psi_j\rangle$ where $\sum_{j=1}^2 |\lambda_j|^2 = 1$, $|\psi_j\rangle = \otimes_{k=1}^n |b_k\rangle$, $|b_k\rangle \in \{0, 1\}^{\otimes N}$.

Quantum Circuit Model To process data stored in a quantum state while preserving its normalization under the l_2 -norm, a unitary transformation U satisfies the requirement that $U^\dagger U = \mathbb{I}$. In quantum computing, the circuit model is a widely used language to describe how the quantum information flows through a network of unitary transformations. To process data stored in a quantum state while preserving its normalization under the l_2 -norm, the unitary transformation U satisfies the requirement such that $U^\dagger U = U U^\dagger = \mathbb{I}$.

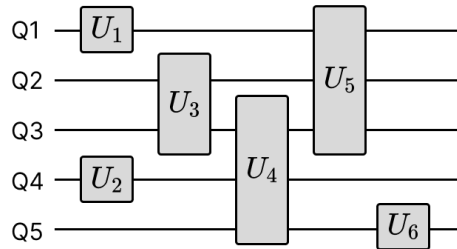


Figure 7: A diagram of a quantum circuit model. The solid block represents the quantum gate, and the horizontal lines stand for qubits. The running order of the quantum circuit is from left to right. The corresponding unitary matrix of this quantum circuit is $U = U_6 U_5 U_4 U_3 U_2 U_1$.

In quantum computing, the circuit model is a widely used language to describe how the quantum information flows through a network of unitary transformations. The diagram of the quantum circuit model is shown in Fig. 7. Like the classical circuit model, we name the unitary operation $U \in \mathbb{C}^{2^n \times 2^n}$ on n qubits as a quantum gate. A group of commonly used single-qubit gates is the Pauli gates, i.e.,

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (9)$$

Based on the Pauli gates, there are rotational gates around the X , Y , Z -axes of the Bloch sphere that can be parametrized with the rotation angle $\theta \in \mathbb{R}$, respectively, i.e.,

$$R_x = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, R_y = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, R_z = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}. \quad (10)$$

Besides, a widely used multi-qubit gate is a controlled gate which applies a specific operation on the target qubits according to the value of the control qubit, generally formed as $U_c = |0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes G$ where G is the operation applied on target qubits. The CNOT gate and CZ gate are two specific two-qubit controlled gates where G operation is X or Z gate, respectively. Their mathematical expressions are:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } \text{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \quad (11)$$

There is a specific collection of quantum gates, termed universal quantum gates, such that any unitary transformation can be represented as a finite sequence of the gates drawn from this set.

Measurement To extract the classical information from the quantum state, one needs to perform a quantum measurement, which causes the collapse of the superposition into one of its possible states. For instance, when we perform a projective measurement associated with measurement operator M_m where m refers to the measurement outcomes on $|\mathbf{u}\rangle$, then such an operation returns m with probability $\langle \mathbf{u} | M_m | \mathbf{u} \rangle$. Besides, through quantum measurement, we can estimate the expectation value of a given Hamiltonian H , which corresponds to the average energy of the system in the quantum state $|\psi\rangle$, i.e., $E = \langle \psi | H | \psi \rangle$.

These components together form the foundation of quantum computation, enabling the execution of quantum algorithms and the realization of quantum advantage.

A.2 Variational quantum algorithm

Variational Quantum Algorithms (VQAs) represent a promising class of hybrid quantum-classical algorithms tailored for the noisy intermediate-scale quantum (NISQ) era [5, 53]. These algorithms cleverly combine the power of quantum computation for preparing and measuring parameterized quantum states with classical optimization routines that iteratively adjust these parameters to minimize a cost function. Generally, the cost function can be expressed as

$$\mathcal{E}(f, U(\boldsymbol{\theta}), \{|\mathbf{u}\rangle\}, \{\mathbf{O}\}, \{\mathbf{s}\}) = \sum_{j,k,l} f(\langle \psi(\boldsymbol{\theta}, \mathbf{u}_j) | O_k | \psi(\boldsymbol{\theta}, \mathbf{u}_j) \rangle, s_l), \quad (12)$$

where $U(\boldsymbol{\theta})$ denotes parametrized quantum circuit with tunable parameters $\boldsymbol{\theta}$, \mathbf{s} refer to labels (optional), $\{|\mathbf{u}\rangle\}$ and $\{\mathbf{O}\}$ are a set of given states and observables, respectively, and $|\psi(\boldsymbol{\theta}, \mathbf{u}_j)\rangle = U(\boldsymbol{\theta})|\mathbf{u}_j\rangle$ refers to the parametrized quantum state. The following are two typical VQAs: variational quantum eigensolver (VQE) [53, 86, 90, 94] and quantum neural network (QNN) [41, 45, 46, 95].

Variational Quantum Eigensolver is a prominent variational quantum algorithm specifically designed to find the ground state energy of a quantum system. It utilizes a parameterized quantum circuit to prepare a trial wave function, and a classical optimizer iteratively adjusts the circuit's parameters to minimize the expectation value of the Hamiltonian of the system. Given a Hamiltonian $H = \sum_{k=1}^{N_H} \lambda_k H_k$, the cost function of VQE can be presented in the form of Eq. (12) by setting f as a identity function, $\{|\mathbf{u}\rangle\} = \{|0\rangle\}$, $\{\mathbf{s}\} = \emptyset$, and $\{\mathbf{O}\} = \{\lambda_k H_k\}_{k=1}^{N_H}$, i.e.

$$\mathcal{E}_{\text{VQE}} = \langle 0 | U(\boldsymbol{\theta})^\dagger H U(\boldsymbol{\theta}) | 0 \rangle. \quad (13)$$

Quantum Neural Network is a machine learning model that employs parameterized quantum circuits to learn from data, analogous to the role of layers in classical neural networks [46, 95]. Given training samples $\{\mathbf{x}_j, y_j\}_{j=1}^N$, the cost function of QNN can be expressed as

$$\mathcal{E}_{\text{QNN}} = \frac{1}{2N} \sum_{j=1}^N \left(\langle \mathbf{x}_j | U(\boldsymbol{\theta})^\dagger O U(\boldsymbol{\theta}) | \mathbf{x}_j \rangle - y_j \right)^2, \quad (14)$$

by setting $\{|u\rangle\} = \{|\mathbf{x}_j\rangle\}_{j=1}^N$, $\{O\} = \{O\}$, and $\{s\} = \{y_j\}_{j=1}^N$, where $f(\cdot, \cdot)$ can be the mean squared error between $\langle \mathbf{x}_j | U(\boldsymbol{\theta})^\dagger O U(\boldsymbol{\theta}) | \mathbf{x}_j \rangle$ and y_j .

B Related works on accelerating the optimization of VQAs

Reducing Measurement Costs . Since the number of terms in an electronic Hamiltonian generally scales with $\mathcal{O}(N^4)$, where N is the system size, many works explore ways of grouping compatible terms that can be simultaneously measured [27–29]. However, the reduction in measurements heavily relies on the interaction structure of the Hamiltonian, and finding the optimal groups could be computationally complicated.

Improving Convergence Efficiency Warm start is a common approach that generates superior initializations to improve efficiency in optimization and machine learning. The relevant studies naturally borrow ideas from warm start to enhance the convergence efficiency of VQAs [96]. One line utilizes problem-specific techniques like randomized rounding in QAOA [97], and imaginary time evolution in QUBO and learning quantum circuit [32, 98]. In a different vein, some studies focus on exploring generative-based approaches, such as Bayesian Learning [99], and diffusion model [99], to identify a promising region in parameter space. Nonetheless, the non-convex landscape of VQA loss appears to be filled with traps [100].

Predicting Dynamics of Parameter Updates Learning to optimize in VQAs aims to harness machine learning 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. Our method is developed from an alternative perspective, which explicitly approximates the training dynamics with a second-order nonlinear PDE, then utilizes a learning-based model to find the solution.

C Implementation Details of PALQO

In this section, we present a more detailed discussion about the PALQO, including the relation to QNTK, and details of the training and prediction process.

C.1 Relation to QNTK

The quantum neural tangent kernel (QNTK) is a tool used to analyze the behavior of VQAs, particularly variational quantum circuits [39, 41]. Inspired by the neural tangent kernel from classical deep learning, the QNTK allows for theoretical insights into the training dynamics and generalization properties of these quantum models.

Let us first present the explicit form of QNTK in QNN. Recall the definition of QNN in Eq. (14), where the loss function is \mathcal{E}_{QNN} , the number of trainable parameters is p . Let the residual of j -th sample be $\mathcal{E}_j = g(\mathbf{x}_j, \boldsymbol{\theta}) - y_j$ where $g(\mathbf{x}_j, \boldsymbol{\theta}) = \langle \mathbf{x}_j | U(\boldsymbol{\theta})^\dagger O U(\boldsymbol{\theta}) | \mathbf{x}_j \rangle$. The derivative of \mathcal{E}_j with respect to t can be expressed as

$$\frac{\partial \mathcal{E}_i}{\partial t} = -\frac{\eta}{2N} \sum_{j=1}^N \sum_{k=1}^p \frac{\partial g(\mathbf{x}_i, \boldsymbol{\theta}^{(t)})}{\partial \theta_k^{(t)}} \frac{\partial g(\mathbf{x}_j, \boldsymbol{\theta}^{(t)})}{\partial \theta_k^{(t)}} \mathcal{E}_j. \quad (15)$$

In this regard, the element of QNTK, $K_{i,j}$, is defined as

$$K_{i,j} \equiv \sum_{k=1}^p \frac{\partial g(\mathbf{x}_i, \boldsymbol{\theta}^{(t)})}{\partial \theta_k^{(t)}} \frac{\partial g(\mathbf{x}_j, \boldsymbol{\theta}^{(t)})}{\partial \theta_k^{(t)}}. \quad (16)$$

We next present QNTK in VQE. We consider the cost function of VQE in Eq. (13), the change between every two iterations can be expressed as

$$\Delta \mathcal{E}_{\text{VQE}} = \mathcal{E}_{\text{VQE}}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{E}_{\text{VQE}}(\boldsymbol{\theta}^{(t)}), \quad (17)$$

$$= \mathcal{E}_{\text{VQE}}(\boldsymbol{\theta}^{(t)} + \delta \boldsymbol{\theta}^{(t)}) - \mathcal{E}_{\text{VQE}}(\boldsymbol{\theta}^{(t)}). \quad (18)$$

Supported by Taylor expansion, we have

$$\mathcal{E}_{\text{VQE}}(\boldsymbol{\theta}^{(t)} + \delta \boldsymbol{\theta}^{(t)}) = \mathcal{E}_{\text{VQE}}(\boldsymbol{\theta}^{(t)}) + \sum_i \frac{\mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_i^{(t)}} \delta \boldsymbol{\theta}_i^{(t)} + \frac{1}{2} \sum_{j,k} \frac{\partial^2 \mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_j^{(t)} \partial \boldsymbol{\theta}_k^{(t)}} \partial \mathcal{E} \delta \boldsymbol{\theta}_j^{(t)} \delta \boldsymbol{\theta}_k^{(t)} + \mathcal{O}(\|\delta \boldsymbol{\theta}^{(t)}\|^3). \quad (19)$$

Since $\delta \boldsymbol{\theta}^{(t)} = -\eta \nabla_{\boldsymbol{\theta}} \mathcal{E}_{\text{VQE}}(\boldsymbol{\theta}^{(t)})$, suppose the learning rate η is infinitesimally small, we can write the dynamics of \mathcal{E}_{VQE} as

$$\frac{\partial \mathcal{E}_{\text{VQE}}}{\partial t} = - \sum_i \frac{\partial \mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_i^{(t)}} \frac{\partial \mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_i^{(t)}} + \frac{1}{2} \eta \sum_{j,k} \frac{\partial^2 \mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_j^{(t)} \partial \boldsymbol{\theta}_k^{(t)}} \frac{\partial \mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_j^{(t)}} \frac{\partial \mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_k^{(t)}} + \mathcal{O}(\eta^2). \quad (20)$$

In Eq. (20), the first contributing term can be regarded as a special case of QNTK in Eq. (16) that only has a single data point, denoted as

$$K' = \sum_i \frac{\partial \mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_i^{(t)}} \frac{\partial \mathcal{E}_{\text{VQE}}}{\partial \boldsymbol{\theta}_i^{(t)}}. \quad (21)$$

This suggests that due to the similarity in cost functions of various VQAs, PALQO can be naturally extended to other VQA models like QNNs.

C.2 Implementation details of PALQO

PALQO is a hybrid quantum-classical algorithm designed to optimize VQA parameters by iteratively combining short VQA training runs on a quantum device with classical learning using a PINN. In each iteration, the algorithm performs a few VQA steps to gather data (i.e., $\boldsymbol{\theta}$ and \mathcal{E}), trains the PINN to model the local loss landscape, and then uses the trained PINN to predict a potentially better set of parameters. These predicted parameters are then used as the starting point for the next VQA training phase, repeating the cycle until the VQA loss converges, aiming to accelerate and improve the overall optimization process by leveraging the PINN as a surrogate model to guide the search in the parameter space. The whole process of PALQO is summarized in Algorithm 1.

Algorithm 1 PALQO

- 1: **Input:** a VQA with parameters $\boldsymbol{\theta}$, PINN-based model f_w with w constituting weights and biases.
 - 2: **Output:** Parameters $\hat{\boldsymbol{\theta}}^*$ to minimize the VQA loss.
 - 3: Randomly initialize the $\boldsymbol{\theta}$ and w .
 - 4: **repeat**
 - 5: Perform τ steps VQA training on quantum device to form $\mathcal{S} = \{\boldsymbol{\theta}^{(t)}, \mathcal{E}^{(t)}\}_{t=1}^{\tau}$.
 - 6: Train the model f_w over \mathcal{S} .
 - 7: $j \leftarrow 0, \boldsymbol{\theta}^{(j)} \leftarrow \boldsymbol{\theta}^{(\tau)}$
 - 8: **repeat**
 - 9: $\hat{\boldsymbol{\theta}}^{(j+1)} = f_w(\boldsymbol{\theta}^{(j)}), \boldsymbol{\theta}^{(j)} \leftarrow \hat{\boldsymbol{\theta}}^{(j+1)}$.
 - 10: **until** $\hat{\boldsymbol{\theta}}^{(j)}$ converge
 - 11: $\hat{\boldsymbol{\theta}}^* \leftarrow \hat{\boldsymbol{\theta}}^{(j)}, \boldsymbol{\theta} \leftarrow \hat{\boldsymbol{\theta}}^*$.
 - 12: **until** $\mathcal{E}_{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ converge
 - 13: **Return:** $\hat{\boldsymbol{\theta}}^*$
-

Instead of relying solely on the potentially noisy and gradient-limited information obtained directly from the quantum device in each step, PALQO uses the PINN to learn a smoother and more global picture of the loss landscape based on local explorations. This can potentially lead to faster convergence and help escape local minima in VQA optimization.

In the following, we elucidate the implementation of each step omitted in the main text.

Dataset Collection Here, we formally define the dataset required for each training session. The dataset consists of m sets, each corresponding to one step in the VQE iteration. Each training sample consists of an input-output pair $\{(t, \boldsymbol{\theta}^{(t)}), (\mathcal{E}^{(t)}, \boldsymbol{\theta}^{(t+1)})\}$, where $\boldsymbol{\theta}^{(t)}$ represents the variational parameters at step t , and $\mathcal{E}^{(t)}$ is the corresponding loss function value. The variable t is a custom-defined discrete sequence that maintains the temporal ordering of $\boldsymbol{\theta}^{(t)}$. To ensure consistency, here we specify the input-output pair as $\{(\hat{t}, \boldsymbol{\theta}^{(t)}), (\mathcal{E}^{(t)}, \boldsymbol{\theta}^{(t+1)})\}$ where \hat{t} is the time variable starting at 0.01 and increases by 0.01 for each step t . In other words, for a dataset with τ training samples, \hat{t} takes values from 0.01 to $0.01 \times \tau$.

Neural Network Structure The Neural Network is a fully connected feedforward neural network with two hidden layers. The total number of variational parameters is defined as p , making both the input and output dimensions $p + 1$. Each hidden layer consists of $50 \times p$ neurons, and the activation function for all layers is \tanh .

Iterative Prediction in PALQO As described in the main text, the prediction process involves feeding the input $(t + \tau, \boldsymbol{\theta}^{(t+\tau)})$ into the network to iteratively produce the m -step prediction, i.e. $\{\hat{\boldsymbol{\theta}}^{(t+\tau+j)}\}_{j=1}^m$. And the iterative prediction terminates once $\hat{\boldsymbol{\theta}}$ converges. Here, calculating the \mathcal{E} in each step is expensive, thereby the convergence is defined as satisfying the condition only on $\boldsymbol{\theta}$: $\Delta = \|\hat{\boldsymbol{\theta}}^{(t+\tau+m)} - \hat{\boldsymbol{\theta}}^{(t+\tau+m-1)}\|_2 < \epsilon$, where $\epsilon = 10^{-4}$. However, in the actual VQE optimization trajectory, Δ tends to decrease gradually as iterations progress. If the stopping condition is applied directly, it may lead to premature termination, resulting in suboptimal performance, or excessively delayed termination, leading to unnecessary computational overhead.

To address this issue, we incorporate an additional guarantee mechanism: the iterative prediction is executed for a fixed number of 2000 iterations. We separately calculate the loss \mathcal{E} using the $\boldsymbol{\theta}$ that minimizes Δ and $\hat{\boldsymbol{\theta}}^{(t+\tau+2000)}$, and then select the minimal one as the optimal variational parameter, $\hat{\boldsymbol{\theta}}^*$, which is subsequently used as the initialization $\boldsymbol{\theta}^{(0)}$ for the next VQE cycle.

D Theoretical Analysis

In this section, we provide a rigorous analysis of the performance of PALQO, which builds on a previous work, i.e., Corollary 1 of (De Ryck & Mishra (2022)) [76], to gain insights into the generalization ability of PALQO. Notably, while previous work offers a general bound, it cannot be directly applied to the nonlinear PDEs relevant to our problem. Therefore, we introduce Lemmas D.1, D.2 and D.4 and combine these with Corollary 1 in Ref. [76] to derive our Corollary D.5

Lemma D.1. *Given an L -layer \tanh neural network $f(\mathbf{x}, (\mathbf{W}, \mathbf{b}))$ constructed by bounded weights $\mathbf{W} = \{W^{(l)}, |W^{(l)}| \leq a, l \in [L]\}$, bias $\mathbf{b} = \{b^{(l)}, |b^{(l)}| \leq a, l \in [L]\}$ and activation function $\sigma = \tanh(x)$, the norm of Jacobian with respect to input vector \mathbf{x} is bounded by,*

$$|J_f| \leq a^L.$$

Proof. As the output of l -layer can be presented by $\mathbf{f}_l = \sigma(W^{(l)\top} \mathbf{f}_{l-1} + b^{(l)})$ and $\sigma'(x) = 1 - \sigma^2(x)$, the Jacobian with respect to the input vector is

$$J^{(l)} = \frac{\partial \mathbf{f}_l}{\partial \mathbf{f}_{l-1}} = \text{diag}[\sigma'(\mathbf{f}_{l-1})] \cdot W^{(l)\top}. \quad (22)$$

According to the chain rule, we can derive the Jacobian of $f(\mathbf{x}, (\mathbf{W}, \mathbf{b}))$ as

$$J_f = \prod_{l=0}^{L-1} J^{(L-l)} = \prod_{l=0}^{L-1} \text{diag}[\sigma'(\mathbf{f}_{L-l-1})] \cdot W^{(L-l)\top}. \quad (23)$$

Since $\sigma' = \text{sech}^2(x)$ and let $D = \text{diag}(\sigma')$, we have $|D_{i,i}| \leq 1$. Then, as $|W^{(l)}| \leq a$, we have

$$|J_f| \leq a^L. \quad (24)$$

□

Lemma D.2. For an L -layer tanh neural network $f(\mathbf{x}, (\mathbf{W}, \mathbf{b}))$ constructed by bounded weights $\mathbf{W} = \{W^{(l)}, |W^{(l)}| \leq a, l \in [L]\}$, bias $\mathbf{b} = \{b^{(l)}, |b^{(l)}| \leq a, l \in [L]\}$ and activation function $\sigma = \tanh(x)$, the norm of Hessian with respect to input vector \mathbf{x} is bounded by,

$$|H_f| \leq 2a^{2L}L. \quad (25)$$

Proof. Since $\sigma'(x) = 1 - \sigma^2(x)$ and $\sigma''(x) = -2\sigma(x)(1 - \sigma^2(x))$, the Hessian of $f(\mathbf{x}, (\mathbf{W}, \mathbf{b}))$ can be expressed as

$$H^{(l)} = \frac{\partial^2 \mathbf{f}_l}{\partial (\mathbf{f}_{l-1})^2} = \text{diag}[\sigma''(\mathbf{f}_{l-1})] \cdot W^{(l)} W^{(l)\top}. \quad (26)$$

According to the lemma of expression for Hessian H in terms of J [76], and $|\sigma''(x)| \leq 2$

$$H_f = \sum_{l=1}^L J^{(1)\top} \dots J^{(l-1)\top} \cdot \left(J^{(L)} \dots J^{(l+1)} H^{(l)} \right) \cdot J^{(l-1)} \dots J^{(1)}. \quad (27)$$

we can bound the H_f by

$$|H_f| \leq 2a^{2L}L. \quad (28)$$

□

Lemma D.3 (Lipschitz continuous of Jacobian and Hessian (Lemma 12, [76])). Let $a, b \in \mathbb{R}$, for an L -layer tanh neural network $f(\mathbf{x}, (\mathbf{W}, \mathbf{b}))$ constructed by bounded weights $\phi \in \{\mathbf{W}, \mathbf{b}\}$, $|\phi| \leq a$ and activation function $\sigma = \tanh(x)$, at most W width, it holds that for any $\mathbf{x} \in [-b, b]^p$,

$$\begin{aligned} |J_\phi - J_{\phi'}| &\leq b(p+7)La^{2L-1}W^{2L-2}2^L |\phi - \phi'|, \\ |H_\phi - H_{\phi'}| &\leq b(p+7)L^2a^{3L-1}W^{3L-3}2^{L+2} |\phi - \phi'|. \end{aligned}$$

Lemma D.4. Let $a, b, N \in \mathbb{R}$, suppose that the employed PINN is constructed by the tanh neural network with bounded weights and biases $\phi \in [-a, a]^m$, at most L layers and W width. Moreover, suppose it adopts a smooth activation function $\sigma = \tanh(x) = \frac{e^{-x} - e^x}{e^{-x} + e^x}$, and the input $\mathbf{x} = \{x_j\}_{j=1}^N$ where $x_j \in [-b, b]^p$. When applying such a PINN to approximate the solution of training dynamics of VQAs with a fixed learning rate η . The Lipschitz constant \mathcal{L} of training error \mathcal{E}_T or generalization error \mathcal{E}_G can be respectively bounded by

$$\mathcal{L} \leq \mathcal{O}(\text{poly}(b, p, L, \eta, a^L, W^L)). \quad (29)$$

Proof. Since the analysis of \mathcal{L} of \mathcal{E}_T and \mathcal{E}_G is similar, here we mainly focus on \mathcal{E}_T . As we select the square error as the loss function, i.e.

$$\mathcal{E}_T(\phi) = \frac{1}{N} \sum_{j=1}^N (\mathcal{R}[f_\phi(x_j)])^2 = \frac{1}{N} \sum_{j=1}^N (\partial_t f_\phi(x_j) - \mathcal{N}[f_\phi(x_j)])^2, \quad (30)$$

where \mathcal{R} is residual of PDE, and f_ϕ is the PINN approximation. As \mathcal{E}_T is differentiable, we have

$$|\mathcal{E}_T(\phi) - \mathcal{E}_T(\phi')| \leq 2 \max_{\phi} |\mathcal{R}[f_\phi]| |\mathcal{R}[f_\phi] - \mathcal{R}[f_{\phi'}]|. \quad (31)$$

For the $|\mathcal{R}[f_\phi] - \mathcal{R}[f_{\phi'}]|$ term, according to the chain rule for the derivative of a composite function, we have $J_\phi = \prod_{k=0}^{L-1} J_\phi^{L-k}$, $H_\phi = \sum_{k=0}^L (J_\phi^1)^\top \dots (J_\phi^{k-1})^\top \cdot (J_\phi^L \dots J_\phi^{k+1} H_\phi^k) \cdot J_\phi^{k-1} \dots J_\phi^1$, where J_ϕ^{L-k} is the Jacobian matrix at the $(L-k)$ -th layer, and H_ϕ^k is the Hessian matrix at the k -th layer. For the training dynamic of VQAs with a fixed learning rate η , we can formulate it as a PDE as shown in Eq. (5), i.e.

$$\mathcal{N}[f_\phi] = J_\phi^\top \cdot J_\phi - \frac{1}{2}\eta J_\phi^\top \cdot H_\phi \cdot J_\phi. \quad (32)$$

where \mathcal{N} is the differential operator. As $\partial_t f_\phi$ can also be regarded as the Jacobian only for the variable t . Thus, we have

$$|\mathcal{R}[f_\phi] - \mathcal{R}[f_{\phi'}]| \leq |J_\phi - J_{\phi'}| + \underbrace{|J_\phi^\top \cdot J_\phi - J_{\phi'}^\top \cdot J_{\phi'}|}_A + \frac{1}{2}\eta \underbrace{|J_\phi^\top \cdot H_{\phi'} \cdot J_{\phi'} - J_{\phi'}^\top \cdot H_\phi \cdot J_\phi|}_B. \quad (33)$$

Since the activation function $\sigma = \tanh(x)$, $|\sigma'|_\infty = 1$ and $|\sigma''|_\infty \leq 1$, and based on the Lemma of Lipschitz continuity of Jacobian and Hessian (Lemma D.3), we can bound the A term

$$\begin{aligned} A &= |J_\phi^\top \cdot J_\phi - J_{\phi'}^\top \cdot J_{\phi'}| \leq (|J_\phi^\top| + |J_{\phi'}^\top|) |J_\phi - J_{\phi'}| \\ &\leq b(p+7)La^{3L-1}W^{2L-2}2^{L+2}|\phi - \phi'|. \end{aligned} \quad (34)$$

Similarly, the term B can be bounded by

$$\begin{aligned} B &= |J_{\phi'}^\top \cdot H_{\phi'} \cdot J_{\phi'} - J_\phi^\top \cdot H_\phi \cdot J_\phi| \\ &\leq |J_{\phi'}^\top - J_\phi^\top| |H_{\phi'} - H_\phi| |J_{\phi'} - J_\phi| + |J_{\phi'}^\top| |H_{\phi'} - H_\phi| |J_\phi| \\ &\quad + |J_\phi^\top| |H_{\phi'}| |J_{\phi'} - J_\phi| + |J_{\phi'}^\top - J_\phi^\top| |H_\phi| |J_\phi| \\ &\leq b^5(p+7)^3L^3a^{5L-1}W^{5L-5}2^{2L+4}|\phi - \phi'|. \end{aligned}$$

Thus, we have

$$\begin{aligned} |\mathcal{R}[f_\phi] - \mathcal{R}[f_{\phi'}]| &\leq (b(p+7)^3La^{2L-1}W^{2L-2}2^L) \\ &\quad \times (1 + 4a^L + \eta b^4(p+7)^2La^{3L}W^{3L-3}2^{L+3})|\phi - \phi'|. \end{aligned} \quad (35)$$

Besides, we can set $\phi' = 0$ to bound $2 \max_\phi |\mathcal{R}[f_\phi]|$ in Eq. (30) i.e.,

$$\begin{aligned} 2 \max_\phi |\mathcal{R}[f_\phi]| &\leq (b(p+7)^3La^{2L-1}W^{2L-2}2^L) \\ &\quad \times (a + 4a^{L+1} + \eta b^4(p+7)^2La^{3L+1}W^{3L-3}2^{L+3}). \end{aligned} \quad (36)$$

Combine Eq. (35) and Eq. (36), we have

$$\begin{aligned} \mathcal{L} &\leq (b^2(p+7)^6L^2a^{4L-1}W^{4L-4}2^{2L}) (1 + 4a^L + \eta b^4(p+7)^2La^{3L}W^{3L-3}2^{L+3})^2 \\ &= \mathcal{O}(\text{poly}(b, p, L, \eta, a^L, W^L)). \end{aligned} \quad (37)$$

□

D.1 Generalization error analysis

We now present the theoretical analysis of the generalization performance of the PINN model on learning the training dynamics of VQAs. We first start with the following general setting, let $D \subset \mathbb{R}^d$ be a compact space and $u : D \rightarrow \mathbb{R}$ be the true solution for the training dynamics and $u_\phi : D \rightarrow \mathbb{R}$ be the PINNs approximation with parameters $W \in \mathbb{R}^d$. Let $\mathcal{S} = \{x_i\}_{i=1}^N$ be the independently sampled training data-set with probability measure μ over D . Here, we define the empirical risk \mathcal{E}_T trained over \mathcal{S} and expected risk \mathcal{E}_E perspectively,

$$\begin{aligned} \mathcal{E}_T &= \frac{1}{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} |u(x_j) - u_\phi(x_j)|^2, \\ \mathcal{E}_E &= \int_D d\mu |u - u_\phi|^2. \end{aligned}$$

Here, we denote $\phi^* = \arg \min_{\phi \in \mathbb{R}^m} \mathcal{E}_T$ as the optimal parameters of PINN over training set \mathcal{S} , then the generalization error can be decomposed as follows [76],

$$\begin{aligned} \mathcal{E}_E(\phi^*) &\leq \sup_{\hat{\phi} \in \mathbb{R}^m} |\mathcal{E}_E(\phi^*) - \mathcal{E}_E(\hat{\phi})| + \sup_{\hat{\phi} \in \mathbb{R}^m} |\mathcal{E}_T(\hat{\phi}) - \mathcal{E}_T(\phi^*)| \\ &\quad + \sup_{\hat{\phi} \in \mathbb{R}^m} |\mathcal{E}_E(\hat{\phi}) - \mathcal{E}_T(\hat{\phi})| + \mathcal{E}_T(\phi^*). \end{aligned} \quad (38)$$

Based on this, we can utilize Hoeffding's inequality and the covering number to give an upper bound on the generalization error of PINN on learning VQAs' training dynamics.

Corollary D.5. *Let $L, W, p, m \in \mathbb{N}, c, k, \epsilon, \gamma, \eta > 0$, and $\phi \in [-a, a]^m$ be the parameters of a tanh neural network with most W width, L hidden layers and activation function σ . Let \mathcal{L} Lipschitz continuous of \mathcal{E}_E and \mathcal{E}_T . The generalization error of PINN, that is trained over*

$\mathcal{S} = \{(t_j, \boldsymbol{\theta}^{(j)}), (\mathcal{E}^{(j)}, \boldsymbol{\theta}^{(j)})\}_{j=1}^\tau$, where t_j and $\mathcal{E}^{(j)}$ are the time variable and loss value at step j , respectively, $\boldsymbol{\theta}^{(j)} \in [-b, b]^p$ for approximating the training dynamics of VQAs with a fixed η learning rate, with probability at least $1 - \gamma$,

$$\mathcal{E}_E(\phi^*) - \mathcal{E}_T(\phi^*) \leq \sqrt{\frac{4c^2}{N} p L W^2 \left(\ln \left(\frac{2a\mathcal{L}}{\epsilon} \right) + \ln \left(\frac{1}{\gamma} \right) \right)}. \quad (39)$$

where $\mathcal{L} = \mathcal{O}(\text{poly}(b, p, L, \eta, a^L, W^L))$,

According to the Corollary D.5 when we assume the training error \mathcal{E}_T is small, the generalization error \mathcal{E}_E for learning the training dynamics of VQAs can be bounded by a function which scales at $\mathcal{O}(\text{poly}(N, W, L, p))$. Besides, we also notice that the data size N polynomially depends on the dimension of data p to guarantee a small generalization error, which overcomes the curse of dimensionality and is also found in [76].

Proof. The main proof idea follows Corollary 1 of [76]. First, for arbitrary $\epsilon > 0$, assume $\mathcal{E}_E(\phi)$ and $\mathcal{E}_T(\phi)$ are \mathcal{L} -lipschitz, we have $\{\phi_i\}_{i=1}^{\mathcal{N}}$ to cover the parameter space Φ with balls of radius δ . Thus, we can bound the first two terms of Eq. (38),

$$\sup_{\hat{\phi} \in \mathbb{R}^m: |\phi - \hat{\phi}| \leq \delta} \left| \mathcal{E}_E(\hat{\phi}) - \mathcal{E}_E(\phi^*) \right| + \sup_{\hat{\phi} \in \mathbb{R}^m: |\phi - \hat{\phi}| \leq \delta} \left| \mathcal{E}_T(\hat{\phi}) - \mathcal{E}_T(\phi^*) \right| \quad (40)$$

$$\leq \sup_{\hat{\phi} \in \mathbb{R}^m} \left| \mathcal{E}_E(\hat{\phi}) - \mathcal{E}_E(\phi^*) \right| + \sup_{\hat{\phi} \in \mathbb{R}^m} \left| \mathcal{E}_T(\hat{\phi}) - \mathcal{E}_T(\phi^*) \right|, \quad (41)$$

where

$$\sup_{\hat{\phi} \in \mathbb{R}^m: |\phi - \hat{\phi}| \leq \delta} \left| \mathcal{E}_E(\phi^*) - \mathcal{E}_E(\hat{\phi}) \right| + \sup_{\hat{\phi} \in \mathbb{R}^m: |\phi - \hat{\phi}| \leq \delta} \left| \mathcal{E}_T(\hat{\phi}) - \mathcal{E}_T(\phi^*) \right| \leq \epsilon. \quad (42)$$

Besides, as parameter space Φ is compact and δ -covered by $\{\phi_i\}_{i=1}^{\mathcal{N}}$, thus for any $\phi_i, i \in [\mathcal{N}]$ we also have

$$\mathcal{E}_E(\phi^*) \leq |\mathcal{E}_E(\phi^*) - \mathcal{E}_E(\phi_i)| + |\mathcal{E}_T(\phi^*) - \mathcal{E}_T(\phi_i)| + |\mathcal{E}_E(\phi_i) - \mathcal{E}_T(\phi_i)| + \mathcal{E}_T(\phi^*). \quad (43)$$

As we can define a projection function f_P that maps ϕ to its nearest cover center ϕ_i , f_P partition the parameter space Φ into \mathcal{N} regions and $\forall \phi \in \Phi, \sum_i \mathcal{P}(f_P(\phi) = \phi_i) = 1$. As $\mathcal{E}_E(\phi) = \mathbb{E}[\mathcal{E}_T(\phi)]$, we first employ the Hoeffding's equation to get

$$\mathcal{P}(\mathcal{E}_E(\phi_j) - \mathcal{E}_T(\phi_j) \leq \epsilon | j \in [\mathcal{N}]) \geq 1 - \exp \left(\frac{-\epsilon^2 N}{2c^2} \right). \quad (44)$$

Then, let the radius be $\delta = \epsilon/2\mathcal{L}$, then the covering number \mathcal{N} can be bounded by $(2a\mathcal{L}/\epsilon)^m$. As such, we take a union bound over \mathcal{N} and achieve

$$\mathcal{P}(\exists \phi_j, \mathcal{E}_E(\phi_j) - \mathcal{E}_T(\phi_j) \leq \epsilon) \geq 1 - \left(\frac{2a\mathcal{L}}{\epsilon} \right)^m \exp \left(\frac{-\epsilon^2 N}{2c^2} \right). \quad (45)$$

and

$$\mathcal{P}(\mathcal{E}_E(\phi^*) - \mathcal{E}_T(\phi^*) \leq \epsilon) \geq \mathcal{P}(\exists \phi_j, f_P(\phi^*) = \phi_j, \mathcal{E}_E(\phi_j) - \mathcal{E}_T(\phi_j) \leq \epsilon). \quad (46)$$

Thus, by combining them, we have

$$\mathcal{P}(\mathcal{E}_E(\phi^*) - \mathcal{E}_T(\phi^*) \leq \epsilon) \geq 1 - \left(\frac{2a\mathcal{L}}{\epsilon} \right)^m \exp \left(\frac{-\epsilon^2 N}{2c^2} \right). \quad (47)$$

Therefore, we have a generalization error bound, with probability at least $1 - \gamma$ as follows

$$\mathcal{E}_E(\phi^*) - \mathcal{E}_T(\phi^*) \leq \sqrt{\frac{2c^2}{N} m \left(\ln \left(\frac{2a\mathcal{L}}{\epsilon} \right) + \ln \left(\frac{1}{\gamma} \right) \right)}. \quad (48)$$

If PINN is constructed using an L -layer tanh neural network with most W width of each layer, it has most $(L-2)W^2 + (p+1)W$ weights and $(L-1)W + 1$ biases. Consequently, $m \leq 2pLW^2$. Then, using the Lemma D.4, i.e. $\mathcal{L} \leftarrow \mathcal{O}(\text{poly}(b, p, L, a, W))$, we have

$$\mathcal{E}_E(\phi^*) - \mathcal{E}_T(\phi^*) \leq \sqrt{\frac{4c^2}{N} p L W^2 \left(\ln \left(\frac{2a\mathcal{O}(\text{poly}(b, p, L, \eta, a^L, W^L))}{\epsilon} \right) + \ln \left(\frac{1}{\gamma} \right) \right)}. \quad (49)$$

□

E Details of Experiments

E.1 Computational resources for all experiments

Most of the simulations were run on Dual NVIDIA GeForce RTX 4090 GPUs with a 96-core AMD EPYC 9654 Processor and 256 GiB of memory.

E.2 Variational quantum ansatz in VQE

Hardware-Efficient Ansatz (HEA) HEAs are a class of variational quantum circuits whose structure is primarily dictated by the connectivity and native gate operations available on a specific quantum computing hardware platform. The HEA typically consists of a repetitive structure of single-qubit rotation gates and fixed entangled gates that can be implemented directly and efficiently on the target hardware, often without requiring complex gate decompositions or extensive qubit routing [84]. Concretely, it can be expressed as

$$U_{\text{HEA}}(\theta) = \prod_{l=1}^L \left(\prod_{i=1}^n R_{i,l}(\theta_{i,l}) \prod_{(i,j) \in E} U_{\text{ent}}^{(i,j)} \right), \quad (50)$$

where $R_{i,l}(\theta_{i,l})$ refers to single-qubit rotation gates at l -th layer acting on i -th qubit, $U_{\text{ent}}^{(i,j)}$ is entanglement gate applied to pairs of qubits (i, j) that are connected according to a predefined graph E that typically reflects the physical connectivity of the qubits on the quantum hardware, ensuring that the entangling gates are applied only to directly connected qubits. In our experiment, we use R_y and R_z gates for single-qubit rotations and CZ gates for building the L -layer HEA with $2nL$ variational parameters.

Hamiltonian Variational Ansatz (HVA) HVA is a class of parameterized quantum circuits, the structure of which is inspired by the time evolution operator under the given Hamiltonian $H = \sum_k H_k$, often constructed as a sequence of exponential terms in the Hamiltonian [88]. By parameterizing the evolution time or related coefficients, the HVA explores the quantum state space in a way that is naturally aligned with the system dynamics, potentially leading to efficient encoding of low-energy states. Generally, it can be written as

$$U_{\text{HVA}}(\theta) = \prod_{l=1}^L \left(\prod_{k=1}^K e^{-i\theta_{k,l} H_k} \right), \quad (51)$$

if H_k is Pauli strings, each evolution operator $e^{-i\theta_{k,l} H_k}$ can be implemented using a sequence of $\{H, S, S^\dagger, \text{CNOT}, R_z\}$. For instance, if $H_k = XYZ$, the circuit implementation of $e^{-iX \otimes Y \otimes Z}$ as shown in Fig. 8. The number of layers L controls the expressivity of the ansatz. This form directly incorporates the structure of the problem's Hamiltonian into the design of the variational circuit.

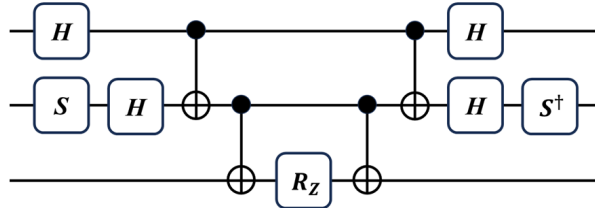


Figure 8: The circuit implementation of $e^{-iX \otimes Y \otimes Z}$.

Unitary Coupled-Cluster Singles and Doubles (UCCSD) Ansatz The UCCSD ansatz is a chemistry-inspired variational quantum circuit widely used in quantum computational chemistry [15, 91, 101]. The electronic structure Hamiltonian in quantum chemistry is expressed in second quantization as

$$H = \sum_{pq} h_{p,q} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s, \quad (52)$$

where \hat{a}_p^\dagger and \hat{a}_q are fermionic creation and annihilation operators, and h_{pq} , h_{pqrs} represent one- and two-electron integrals encoding kinetic energy, nuclear attraction, and electron-electron repulsion. The variational wavefunction is given by

$$|\Psi(\boldsymbol{\theta})\rangle = e^{T-T^\dagger} |\Phi_0\rangle, \quad (53)$$

where $|\Phi_0\rangle$ is the Hartree-Fock state, and $T = T_1 + T_2$ consists of single and double excitation operators:

$$T_1 = \sum_{i,m} \theta_i^m \hat{a}_m^\dagger \hat{a}_i, \quad T_2 = \sum_{i,j,m,n} \theta_{i,j}^{m,n} \hat{a}_n^\dagger \hat{a}_m^\dagger \hat{a}_j \hat{a}_i. \quad (54)$$

Here, i, j index occupied orbitals, m, n index virtual orbitals, and $\boldsymbol{\theta}$ denotes variational parameters. The Jordan-Wigner transformation maps fermionic operators \hat{a} and \hat{a}^\dagger onto qubit operators, ensuring preservation of anticommutation relations and enabling implementation on quantum hardware. In our experiment, we use the BeH_2 molecule as an example. The mapped Hamiltonian requires 14 qubits, and the UCCSD ansatz involves 90 variational parameters.

E.3 Details of benchmarks

Long-short Term Memory (LSTM) The LSTM model employed in our study adopts a standard recurrent architecture, specifically tailored for sequence modeling and parameter optimization tasks. It consists of an LSTM layer with one hidden layer, where the input size corresponds to the number of variational parameters p , and the hidden size is set to $50 \times p$ to enhance its representational capacity. The model takes as input a sequence of past optimization states with a predefined sequence length τ_{LSTM} , allowing it to learn temporal dependencies in parameter evolution. It processes input sequences in a batch-first manner to ensure efficient training. The final hidden state of the LSTM, corresponding to the last time step, is passed through a fully connected linear layer to produce the output, which has the same dimensionality as the input parameters. This structure enables the model to leverage past optimization information effectively to enhance parameter updates.

QuACK. For the QuACK model, we adopt the specific implementation of Dynamic Mode Decomposition (DMD) as proposed in [36]. This approach leverages the Koopman operator learning algorithm to find an appropriate embedding space where the system dynamics can be approximated as linear. By mapping the variational parameter updates into this learned representation, QuACK enables more efficient optimization within the VQA framework. In our implementation, we define the number of samples used per training iteration as τ_{QuACK} , which determines the number of past optimization steps considered for learning the underlying dynamical structure. This parameter plays a crucial role in capturing the temporal evolution of variational parameters while ensuring the stability and generalization ability of the learned model.

E.4 Details of experimental setup

Estimation of Shot Numbers for Measurement We now estimate the measurement on a real quantum computer. The estimation strategy follows the approach outlined in [7], where the number of the Pauli strings in a Hamiltonian is denoted by M , and the target accuracy for the expected value of the measurement is ε . The required number of shots for measuring the expected value of the Hamiltonian is $\mathcal{O}(M/\varepsilon^2)$. Therefore, the required number of shots for one VQE iteration, given p as the number of variational parameters, can be estimated as $2 \times p \times M/\varepsilon^2$. We use $\varepsilon = 1 \times 10^{-3}$ in our specific calculation.

Performance on Different Ansatz In the experimental setup of the 12-qubit TFIM, the 3-layer HEA has a total of $2 \times 12 \times 3$ variational parameters. For the 14 qubits BeH_2 system, the USCCSD ansatz involves 90 variational parameters. In both experiments, the network architecture and training procedure of PALQO follow the standard settings described in Appendix C with a maximum training epoch of $T_{\text{epoch}} = 3400$ and $\tau = 2$ training samples per cycle. The maximum number of LSTM training iterations is $T_{\text{epoch}} = 2000$, with $\tau_{\text{LSTM}} = 3$ training samples per cycle. Additionally, the number of samples $\tau_{\text{QuACK}} = 3$ is used by QuACK.

Scalability In this experiment, the number of variational parameters in an n -qubit TFIM with an L -layer HEA is given by $p = 2 \times n \times L$, where $L = \{2, 3, 4, 5, 6, 7, 8\}$. In experiments conducted with $n = 4$ to $n = 40$ qubits using a fixed 3-layer HEA, the network architecture in PALQO follows

the settings in Appendix C with the maximum number of training epochs T_{epoch} set to $\{3000, 3000, 3000, 3500, 3500, 3500, 3500, 4000, 4000, 4000\}$. Additionally, the number of samples used in the first cycle is set to $\tau = 1$ for the 4-qubit system. For systems with sizes between 4 and 12, $\tau = 2$ samples are employed in subsequent cycles, while for larger systems with sizes ranging from 16 to 40, $\tau = 3$ samples are used. In experiments with a fixed 12-qubit system and varying HEA layers from 2 to 8, the maximum number of training epochs T_{epoch} follows $\{3000, 3000, 3500, 3500, 3500, 3500, 4000\}$. In this setting, except for the first cycle, the number of training samples used per cycle remains $\tau = 2$.

F Additional Numerical Experiments

In this section, we present additional numerical experiments to further validate the superior performance of PALQO. Specifically, we evaluate its effectiveness in three representative tasks: the XXZ model, the LiH molecule, and a quantum machine learning (QML) classification problem. We also examine the robustness of PALQO in the presence of quantum noise. Moreover, our results indicate that PALQO can be effectively integrated with resource-saving techniques, such as measurement grouping, to further reduce quantum resource consumption during the VQA optimization process.

F.1 XXZ and LiH

We present the additional numerical experiments of performance comparisons of PALQO on 12 qubits XXZ with HVA, and 14 qubits LiH with UCCSD ansatz for varying structural parameters are presented in Fig. 9. The results demonstrate that PALQO achieves lower ΔE and higher speedup ratio in most cases. In the case of $J = 1, \delta = 0.5$, PALQO exhibits a comparable speedup ratio to the reference methods, primarily due to the smaller energy gap in this setting, which makes the optimization landscape more challenging and hinders PALQO’s convergence efficiency.

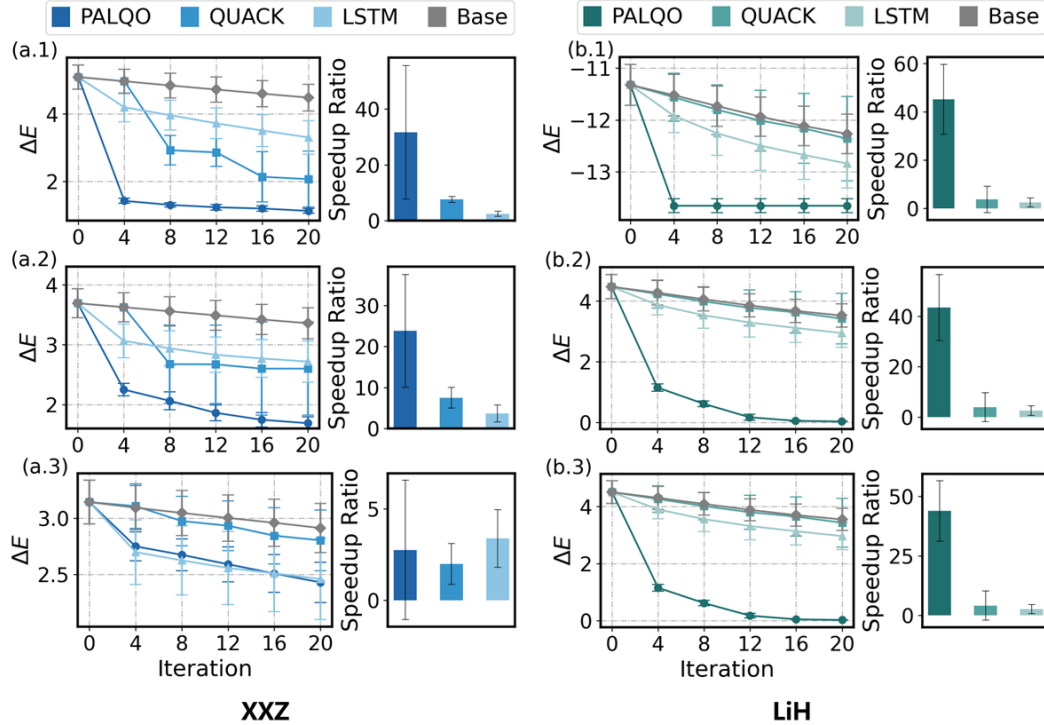


Figure 9: Performance comparison between PALQO and the reference models in XXZ with HVA and 12-qubit LiH with UCCSD ansatz. Each subplot comprises a ΔE curve over iterations performed on a quantum device, along with a bar chart depicting the speedup ratios achieved by PALQO and competing models. The left column illustrates results for XXZ with $J = J' = 1, \delta = \{2, 1, 0.5\}$. The right column displays the model performance on LiH2 with the bond length $b = \{1.4, 1.5, 1.6\}$.

F.2 Quantum machine learning

To further assess the applicability of PALQO in other VQAs like quantum neural network (QNN), we conduct experiments on a classification task. Based on Eq. (15), we rebuild PALQO for QNN with the reformulated cost function. We construct the 4-qubit QNN with 3-layer HEA and measurement observable $O = I \otimes I \otimes Z \otimes Z$ as the baseline model, and employ the quantum circuit shown in Fig. 10 as the feature encoder to map classical input data into quantum states. The performance comparison between PALQO and the baseline model on a classification task over the Iris dataset [102] is shown in Fig. 11. In Fig. 11(a), it shows that PALQO achieves significantly lower loss values than the baseline throughout the iterations. During the initial optimization phase, PALQO is capable of more rapidly reaching the points with lower loss, which in turn reduces the optimization steps. In Fig. 11(b), as PALQO can more swiftly attain lower loss values, it reaches an average accuracy over 90% by the 120 steps, significantly outperforming the baseline model, which only achieves 75%. Therefore, it indicates that the robust applicability of PALQO while exhibiting favorable performance.

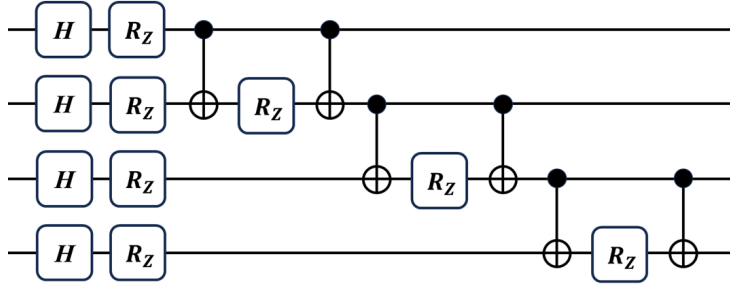


Figure 10: The illustration of a quantum encoder circuit. It employs an instantaneous quantum polynomial (IQP) encoding strategy for QNN [103], in which data features are embedded into the rotation angles of parameterized quantum gates such as R_x, R_z . In our implementation, the Iris dataset features $\mathbf{x} = (x_0, \dots, x_7)$ are individually encoded into the rotation angles of 7 corresponding parameterized gates.

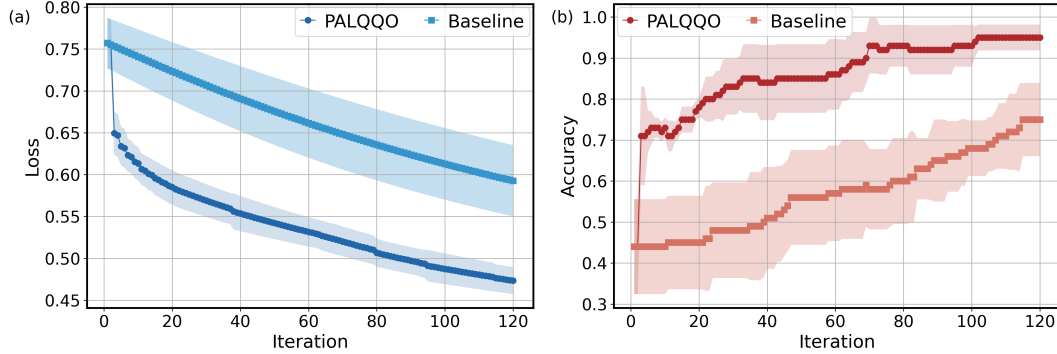


Figure 11: Performance comparison between PALQO and the baseline method on a quantum machine learning classification task using the Iris dataset. (a) The loss curve between PALQO and the baseline model. (b) The accuracy curve of PALQO versus the baseline model over the iterations. Shaded regions refer to the range of the loss and accuracy over multiple runs.

F.3 Performance under noise

We further assess the robustness of PALQO in the presence of noise, specifically evaluating its performance on a 12-qubit TFIM with a 3-layer HEA. In this experiment, we test ten randomly initialized sets of variational parameters for each noise scenario. The results are presented in Fig. 12. Despite the presence of noise, PALQO consistently demonstrates strong performance, highlighting its robustness and practical applicability in realistic quantum environments.

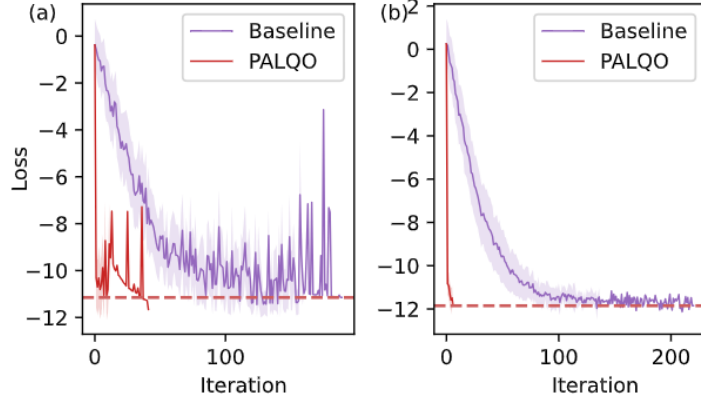


Figure 12: Performance of PALQO under noise conditions. (a) Optimization results with 5% depolarizing noise. (b) Performance under shot noise with a shot count of 100.

F.4 Complement to measurement optimization

Here, we provide the results of the complementary experiments of PALQO and measurement grouping on 20-qubit TFIM, 12-qubit LiH, and 14-qubit BeH₂, which demonstrate that PALQO offers a valuable complement to existing strategies for further enhancing the optimization efficiency of VQAs. Measurement grouping strategically reduces the number of distinct measurements by exploiting the commutativity of Hamiltonian terms, thereby enabling the simultaneous measurement of multiple observables. Thus, PALQO can seamlessly incorporate measurement grouping into the overall framework. As shown in Fig. 13, rather than replacing grouping strategies, our method works in tandem with them, offering a multi-faceted approach to further reduce the quantum resource burden.

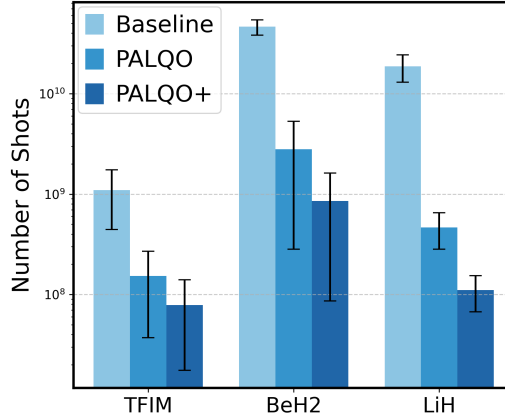


Figure 13: The measurement shots of PALQO, combined with measurement grouping, are evaluated on tasks including the TFIM, LiH, and BeH₂. PALQO+ refers to the PALQO enhanced by measurement grouping.