


Neural-network-encoded variational quantum algorithms

Jiaqi Miao¹, Chang-Yu Hsieh^{2,*} and Shi-Xin Zhang^{3,†}

¹*School of Physics, Zhejiang University, Hangzhou, Zhejiang 310000, China*

²*Innovation Institute for Artificial Intelligence in Medicine of Zhejiang University, College of Pharmaceutical Sciences, Zhejiang University, Hangzhou 310058, China*

³*Tencent Quantum Laboratory, Tencent, Shenzhen, Guangdong 518057, China*

 (Received 9 August 2023; revised 25 December 2023; accepted 10 January 2024; published 26 January 2024)

We introduce a general framework called neural-network- (NN) encoded variational quantum algorithms (VQAs), or NNVQA for short, to address the challenges of implementing VQAs on noisy intermediate-scale quantum (NISQ) computers. Specifically, NNVQA feeds input (such as parameters of a Hamiltonian) from a given problem to a neural network and uses its outputs to parameterize an ansatz circuit for the standard VQA. Combining the strengths of NN and parameterized quantum circuits, NNVQA can accelerate the training process of VQAs and handle a broad family of related problems with varying input parameters with the pretrained NN. To concretely illustrate the merits of NNVQA, we present results on a NN variational quantum eigensolver (VQE) for solving the ground state of parameterized XXZ spin models in one and two dimensions. Our results demonstrate that NNVQE is able to estimate the ground-state energies of parameterized Hamiltonians with high precision without fine tuning, and significantly reduce the overall training cost to estimate ground-state properties across the phases of the given Hamiltonian. We also employ an active learning strategy to further increase the training efficiency while maintaining prediction accuracy. These encouraging results demonstrate that NNVQAs offer an alternative hybrid quantum-classical paradigm to utilize NISQ resources for solving more realistic and challenging computational problems.

DOI: [10.1103/PhysRevApplied.21.014053](https://doi.org/10.1103/PhysRevApplied.21.014053)

I. INTRODUCTION

Today's noisy intermediate-scale quantum (NISQ) computers [1] are far from delivering an unambiguous quantum advantage. Variational quantum algorithms (VQAs), as one of the most representative algorithm primitives in the NISQ era [2–5], utilize a quantum-classical hybrid scheme, where the quantum processor prepares target quantum states and measurement is made to extract useful information for the classical computer to explore and optimize. VQAs have now been widely applied to solve quantum optimization, quantum simulation, and quantum machine-learning problems [6–16].

Among various VQAs, the variational quantum eigensolver (VQE) [6,7] is one of the most exemplary algorithms. The VQE employs the Rayleigh-Ritz variational principle to approximate the ground state of a given Hamiltonian \hat{H} with a parameterized quantum circuit (PQC). Many studies on the strengths and fundamental limitations of VQAs are first systematically investigated and revealed

by studying how the VQE performs in different contexts [17,18]. Despite some early hopes of VQAs' potential quantum advantages in addressing some realistic computational problems, this goal still remains elusive. In fact, it is now known that the current formulation of the standard VQAs faces many obstacles for them to deliver any practical advantages [19–25].

There is a pressing need to develop alternative hybrid quantum-classical approaches to better utilize the full power of quantum computational resources while avoiding as many shortcomings of the standard VQAs as possible. For instance, a core problem of the standard VQA is to identify the suitable circuit parameters for a given problem, i.e., the optimization or training procedure. From a practical perspective, the training procedure often takes many steps, which leads to a large budget for measurement shots [26,27]. Besides, the training procedure could be more sensitive to noise and decoherence compared to the inference procedure. Therefore, training of VQAs is expensive as it must be conducted on very high-quality quantum devices with a large budget of measurement shots.

In terms of theoretical perspective, the difficulties associated with the optimization of VQAs stem from at least

*kimhsieh@zju.edu.cn

†shixinzhang@tencent.com

two fundamental obstacles. One severe challenge is the phenomenon of vanishing gradients named barren plateaus (BPs) [19–23]. Though there are many attempts to mitigate BP issues [28–39], the occurrence of BP, in general, implies that exponential quantum resources are required to navigate through the exponentially flattened cost-function landscape $C(\theta)$, which could negate the potential quantum advantages of VQAs. Another related problem for VQAs' nonconvexity energy landscape is the occurrence of many local minima [24,25], which can easily trap the training trajectories.

In the plain VQA setups, application problems are optimized and solved instance by instance with the same circuit structure, namely, we need to retrain the model for each instance. This workflow renders the optimization issues discussed above more detrimental in the VQA context. Therefore, a general framework to solve the parameterized problem instances jointly and to separate the pretraining process from the inference process is highly desired. Such a framework would address the optimization bottlenecks from two angles. For the pretraining procedure, the joint training on multiple problem instances speeds up the optimization convergence by alleviating the BP and local minima issues. And for the inference procedure conducted by the end users, there is no need to retrain or fine tune the model so that the end users with limited quantum resources are free from the thorny training issues.

In this paper, we introduce a general framework—neural-network-encoded variational quantum algorithms (NNVQAs). Many works integrate the neural network with the quantum circuit from different angles such as quantum state tomography, quantum error mitigation, quantum architecture search, and expressive capacity enhancement [34,40–51]. Our framework further expands the possibility of such an interplay from another perspective. NNVQAs successfully address all the aforementioned challenges: (i) NNVQAs use the Hamiltonian parameters as the input to a neural network, which enables a parameterized model to be solved through only a single pretraining process and a single pretrained model; (ii) the pretrained NNVQAs can give a good estimation with test Hamiltonians beyond the training set with good generalization capability; (iii) active learning method can be adopted to further reduce the number of training samples and thus the number of total measurement shots; (iv) NNVQAs could significantly speedup the optimization convergence of VQAs. Therefore, by using a neural network as the encoding module, our approach provides a good ground-state approximation using only a small number of training points and greatly saves the required quantum resources. Moreover, our framework can enable the separation of training and inference and sketch a potential future interface to utilize VQAs for end users.

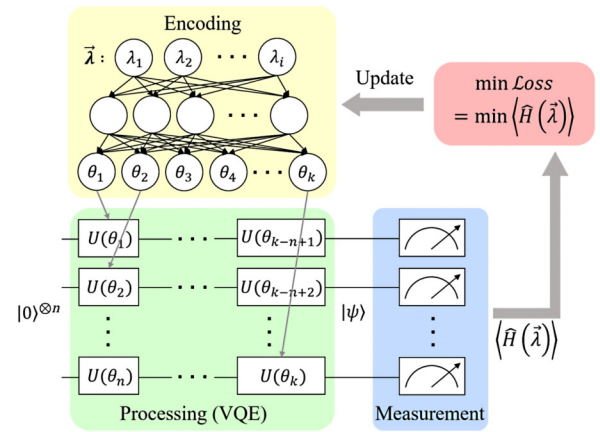


FIG. 1. Schematic workflow for NNVQE. The Hamiltonian parameters λ are the input of the encoder neural network, which produces the parameterized quantum circuit (PQC) parameters θ as output. The PQC, parameterized by θ , is then used in the processing module of the VQE to prepare an output state $|\psi\rangle = U(\theta)|\mathbf{0}\rangle$, where $|\mathbf{0}\rangle$ is the initial state. The cost function can be estimated according to Eq. (3), and the weights in the neural network are optimized using a gradient-based optimizer.

II. THEORETICAL FRAMEWORK

In this section, we introduce the framework of NNVQE for ground-state problems, and the framework can be similarly generalized to the VQE for excited states [52–54] or other VQA scenarios.

The schematic workflow for NNVQE is shown in Fig. 1. Given a parameterized Hamiltonian $\hat{H} = \hat{H}(\lambda)$, where λ consists of p different Hamiltonian parameters, our aim is to solve the ground state of the parameterized Hamiltonian. We choose a subset of λ as the training set $\tilde{\lambda} = \{\tilde{\lambda}_i\}$.

To train an NNVQE, we use $\tilde{\lambda}$ as the input of the encoding neural network, and get the output

$$\theta_i = f_\phi(\tilde{\lambda}_i), \quad (1)$$

where we denote the neural network as a general parameterized function f_ϕ with the training neural weights as ϕ . The number of the output of the neural network is the same as the number of the PQC parameters, and we load each neural-network output element to the corresponding circuit parameters. The PQC $U(\theta)$ for the VQE is initialized in the $|\mathbf{0}\rangle = |\mathbf{0}\rangle^{\otimes n}$ state. Therefore, the output target state for Hamiltonian $\hat{H}(\tilde{\lambda}_i)$ should be

$$|\psi_i\rangle = U(\theta)|\mathbf{0}\rangle = U(f_\phi(\tilde{\lambda}_i))|\mathbf{0}\rangle. \quad (2)$$

The cost function for ground-state VQE is the expectation of $\hat{H}(\lambda)$:

$$\begin{aligned} C(\phi) &= \sum_i \langle \hat{H}(\tilde{\lambda}_i) \rangle \\ &= \sum_i \langle \mathbf{0} | U^\dagger (f_\phi(\tilde{\lambda}_i)) \hat{H}(\tilde{\lambda}_i) U (f_\phi(\tilde{\lambda}_i)) | \mathbf{0} \rangle. \end{aligned} \quad (3)$$

Finally, we compute the gradients with respect to the neural network (backpropagation via the PQC parameters) and minimize the cost function $C(\phi)$ using gradient descent, obtaining the optimal weights ϕ^* for the neural network. Since such a training procedure happens only once and the trained model can be used to approximate the ground state of the family of Hamiltonians, we call this stage pretraining. Upon completion of pretraining, the efficacy of the NNVQE can be evaluated using a test set of different λ from the training set.

III. RESULTS

In this section, we demonstrate the effectiveness of our framework using numerical simulation with TensorCircuit [55]. The testbed model is the one-dimensional (1D) antiferromagnetic XXZ spin Hamiltonian with an external magnetic field subject to the periodic boundary conditions

$$\hat{H} = \sum_{i,i+1} (X_i X_{i+1} + Y_i Y_{i+1} + \Delta Z_i Z_{i+1}) + \lambda \sum_i Z_i, \quad (4)$$

where Δ is the anisotropy parameter and λ is the transverse-field strength.

We start from the one-parameter XXZ model with the transverse-field strength fixed to $\lambda = 0.75$. The training set of Δ is composed of 20 equispaced points in the interval of $[-3.0, 3.0]$. The performance of the NNVQE is evaluated on an expanded test set consisting of 201 equispaced values of Δ in the interval of $[-4.0, 4.0]$. The circuit ansatz we use in this section is inspired by MERA [56,57]. Specifically, we employ deep multiscale entanglement renormalization ansatz (DMERA) circuits [58,59], where D is the circuit depth in each block (see Appendix A for details). The neural network we use is a simple fully connected neural network with a dropout layer. The size of the input layer is 1 corresponding to the number of Hamiltonian parameters Δ , and the size of the output layer corresponds to the number of PQC parameters (see Appendix A for the detailed neural structure).

For the 1D XXZ spin chain consisting of eight qubits, we pretrain the model within the NNVQA framework and evaluate the performance with different circuit block depths D . The results for ground-state (GS) energy prediction are shown in Fig. 2(a). The simulation accuracy improves with larger D and dropout in the neural network. We also display the corresponding fidelities with the exact

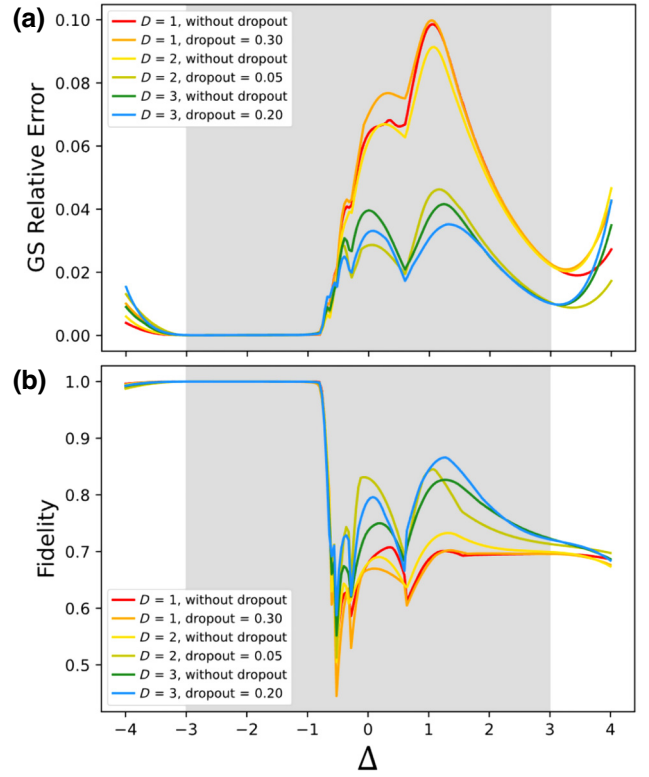


FIG. 2. Results on $n = 8$ qubit one-tunable-parameter 1D XXZ spin chain with a transverse-field strength $\lambda = 0.75$ within the NNVQE framework. (a) Relative errors of ground-state energies of different circuit block depth D with and without dropout from the pretrained model. (b) Fidelity between the output state of NNVQE and the exact ground state.

ground state in Fig. 2(b). The results underscore the ability of the NNVQE to effectively prepare the ground state as a function of the Hamiltonian parameters without fine tuning or retraining. We note that NNVQE demonstrates a favorable generalization capability. As shown in Fig. 2, in regions devoid of shadows on either side (regions of no training points), the NNVQE still provides highly reliable estimations to some extent.

Compared with previous work on meta-VQE [60], when the PQC structures are the same, NNVQE uses fewer quantum resources while yielding better ground-state energy estimation results (see Appendix F for details). Such advantages are mainly brought by the expressive power of general neural networks.

In the previous analysis, the training set is selected in an equispaced manner. Such a strategy can be improved by utilizing active learning techniques [61]. We can maintain the same level of ground-state energy accuracy while using a smaller number of training points.

Various active learning schemes can be easily incorporated into the NNVQE. For example, we begin by randomly selecting one point from the Δ pool P as the initial training set. We train the NNVQE based on the training

set and get the neural-network weights ϕ^* . Obviously, the training set P^* is a subset of the pool P . Subsequently, we calculate the acquisition function specifically designed in this scenario. The active learning acquisition function in our problem is defined as

$$C_{AL} = \langle \hat{H}^2(\Delta) \rangle_{\phi^*} - \langle \hat{H}(\Delta) \rangle_{\phi^*}^2 + \mu \min |\Delta - \Delta^*|, \quad (5)$$

where $\Delta \in P$, $\Delta^* \in P^*$, μ a preset hyperparameter. The first two terms are the variance of the Hamiltonian $\hat{H}(\Delta)$ with ϕ^* trained on the training set. In the last term, we first calculate the distance between Δ and all Δ^* in the training set and find the minimum distance. We employ the hyperparameter μ to find a large variance but prevent a close point from being chosen. The two terms reflect the exploitation and exploration trade-off of the active learning technique. We add the Δ with the largest $C_{AL}(\Delta)$ to the training set. Iteratively, we repeat this process of expanding the training set until the test relative error of the ground-state energy falls below a predetermined threshold.

By this method, we obtain a training set consisting of 11 points. The corresponding results are shown in Fig. 3. Remarkably, even with a training-set size that is only half of the previous set, the model still gives a reliable estimation of the ground-state energy. Moreover, when we visualize the training sets (see the orange dots in Fig. 3), we find them nearly equispaced except for the points near the phase transition point of the Hamiltonian (see the orange inverted triangles in Fig. 3). This observation roughly corresponds to an intuition that the ground-state wave function might experience a more dramatic change around the phase-transition point, which requires more training points to better capture. It is worth noting that the performance of NNVQE is worse in the XY phase (see Appendix J for the phase diagram of the XXZ model). We comment that the source of the degradation of performance is from the standard VQE, say the ansatz is not sufficiently expressive. This fact is reflected by the similar performance degradation for the standard VQE in the XY phase as the black line shown in Fig. 3.

Another remarkable advantage of NNVQE is the training efficiency. As shown in Figs. 4(a) and 4(c), NNVQE has a significant speedup in the optimization procedure compared with plain VQE especially at the early training stage. The energy cost function drops more rapidly in the NNVQE case, which offers great benefits for NISQ computers since fewer training epochs and thus fewer quantum resources are required. Such advantages benefit from the NN-PQC hybrid architecture. The neural network brings a more dramatic change in the PQC parameters at the beginning stage of the optimization process as shown in Figs. 4(b) and 4(d), which is relevant in mitigating the BP issue.

In order to demonstrate the effectiveness of the NNVQE in estimating a multiparameter Hamiltonian, we extend our

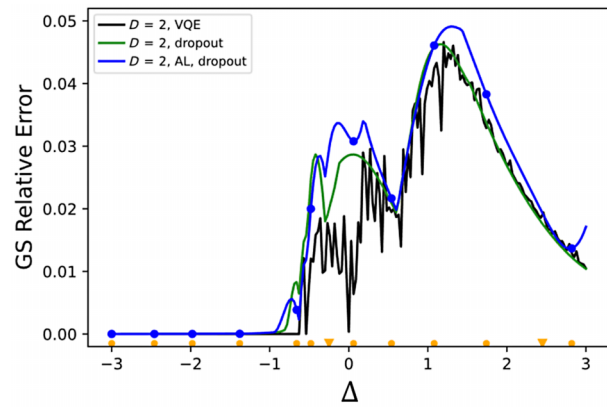


FIG. 3. Active learning for NNVQE. We use the MERA circuit with $D = 2$. The black line is the result of the VQE separately trained on each point. The green line is the NNVQE with dropout and has the same circuit structure as the black line. The blue line shares the same structure (NN encoder and circuit ansatz) as the green line but uses active learning to reduce sample size. The training set of the green line consists of a set of equispaced 20 Δ s in the interval of $[-3.0, 3.0]$ used in the previous analysis. However, by employing active learning, we use only 11 actively selected points to attain the blue line. The training set used for active learning is indicated by dots along the blue line. Remarkably, despite the reduced training set size, the blue line still exhibits a reliable estimation of the ground-state energy. The actively chosen dots are projected onto the x axis as the orange dots, and the orange inverted triangles are the phase-transition points.

study to the two-parameter one-dimensional XXZ model. In this model, both the anisotropy parameter Δ and the transverse-field strength λ are tunable in the Hamiltonian in Eq. (4). The training set for Δ consists of ten equispaced points in the interval of $[-1.0, 1.0]$, while λ consists of five equispaced points in the interval of $[0.0, 1.0]$. The ansatz used is the hardware-efficient ansatz [63] with two-qubit gates in the ladder layout of depth D (see Appendix C for details). The encoding neural network also shares a similar structure as the one-parameter case but now the input takes two values Δ and λ .

The numerical results are presented in Fig. 5. Remarkably, the NNVQE, using a neural network with two inputs, yields excellent performance in estimating the ground state across different phases. This result implies the robustness and versatility of the NNVQE in simulating complex quantum systems governed by a multiple-parameter Hamiltonian.

To further demonstrate the scalability and universality of the NNVQE framework, we apply the framework to a different model (two-dimensional XXZ spin model) with a larger system size ($4 \times 4 = 16$ qubits). The energy estimation precision ratios between NNVQE and the standard VQE are similar for different sizes and different models (see Appendix D for details), indicating that the proposed

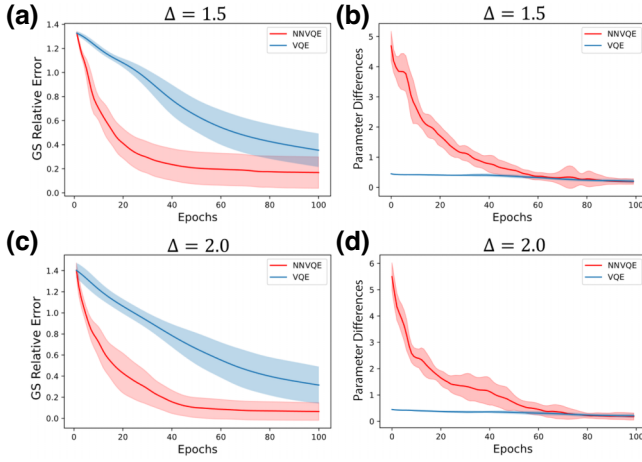


FIG. 4. Speedup in the optimization process of NNVQE and the corresponding PQC parameter changes. (a),(c) The ground-state energy relative errors for an $n = 12$ XXZ spin chain when $\Delta = 1.5, 2.0$ are shown with respect to epochs. Hardware-efficient ansatz with $D = 3$ is used. The red and blue lines correspond to the ground-state energy relative errors and standard deviation of NNVQE and the VQE, respectively. (b),(d) The parameter differences when training the VQE and NNVQE in corresponding Δ . The difference is the sum of the absolute value of parameter differences between epochs. NNVQE brings a more dramatic circuit parameter change at the beginning of the optimization process, which speeds up the optimization process.

framework scales well and is universally applicable to different parameterized systems. Furthermore, we have also conducted NNVQE workflow in the presence of quantum noises, and the results also imply the noise resilience of the NNVQE framework (see Appendix E for details).

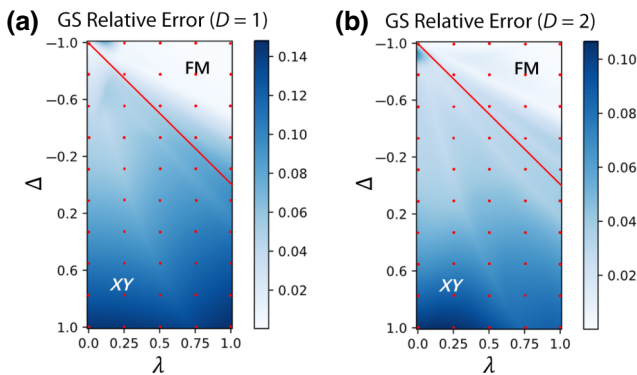


FIG. 5. Relative errors of ground-state energies for an $n = 12$ 1D XXZ spin chain with two tunable Hamiltonian parameters, using the hardware-efficient ansatz with circuit depth $D = 1, 2$. The red dots are the training set. The red lines are the exact phase transition line between the ferromagnetic phase (FM) and the XY phase [62].

IV. DISCUSSION

In this paper, we introduce the NNVQA framework. More specifically, we first use a neural network to transform the Hamiltonian parameters to the optimized parameters in the PQC for VQA. We show the validity and effectiveness of the framework in solving the XXZ Hamiltonian ground state with different parameters through only one pretraining procedure without any problem instance-specific fine tuning. In order to further reduce the pre-training overhead, we also employ an active learning heuristic where the progressively built training set can be greatly reduced. The effectiveness of the framework is also evaluated on different models and with quantum noises.

In terms of the neural-network part, we can introduce more physics-inspired neural-network structures for multiparameter Hamiltonian VQE problems. For example, considering the random Ising model where the couplings at each bond or site are different, we can abstract the Hamiltonian parameters as a graph where the node and edge weights describe the Hamiltonian form. In such cases, we believe a graph neural network (GNN) [64,65] is more suitable for the encoding task as the symmetry and geometry can also be properly addressed in a well-designed GNN. And the power of considering local geometry as in the GNN approach is proven to be exponentially sample efficient in learning quantum state properties [66–69].

Our framework envisions a future paradigm to utilize quantum computers. The encoding neural network can be pretrained on high-quality quantum devices with a large time and measurement budget. The pretrained model can be efficiently saved and shared on classical computers. Since the NNVQE can be targeted to a large family of quantum systems that are connected via lots of parameters, a large pretrained model could be of general interest for solving various problems. The users can utilize the large pretrained classical neural model and extract the trained circuit parameters given the specific problem they are interested in solving. In this paradigm, the users are free from training on quantum computers themselves. It is also worth noting that at the training stage, due to the nature of multiple training points, it is straightforward to utilize the data parallelism and pretrain the NNVQE with multiple quantum processors.

ACKNOWLEDGMENTS

We gratefully thank Gaoxiang Ye for the useful discussions. C.-Y.H. acknowledges support by the National Natural Science Foundation of China (Grant No. 22373085).

APPENDIX A: NOTATIONS FOR THE MODEL, THE VARIATIONAL CIRCUIT, AND THE NEURAL NETWORK

The Hamiltonian. The Hamiltonian used, unless mentioned specifically, is a 1D XXZ model with periodic

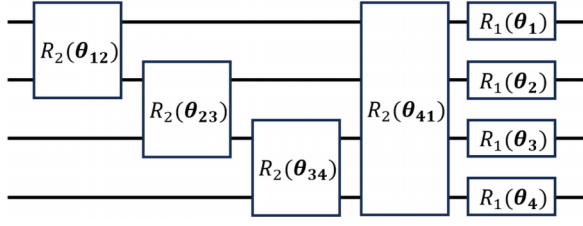


FIG. 6. A block of ladderwise hardware efficient ansatz circuit with four qubits ($D = 1$). Each one-qubit gate $R_1(\theta_i)$ corresponds to $R_z(\theta_{i1})R_x(\theta_{i2})$ and two-qubit gate $R_2(\theta_{ij})$ corresponds to $R_{zz}(\theta_{ij1})R_{xx}(\theta_{ij2})R_{yy}(\theta_{ij3})$.

boundary conditions, with the transverse-field strength λ and the anisotropy parameter Δ :

$$\hat{H} = \sum_{i,i+1} (X_i X_{i+1} + Y_i Y_{i+1} + \Delta Z_i Z_{i+1}) + \lambda \sum_i Z_i. \quad (\text{A1})$$

When we use the single-parameter 1D XXZ model, we fix the transverse-field strength to $\lambda = 0.75$, and the anisotropy parameter Δ is chosen as the varying parameter. When we use the two-parameter 1D XXZ model, both λ and Δ are varying parameters.

In Appendix D, we also evaluated the performance of NNVQE on a two-dimensional XXZ Hamiltonian with open boundary conditions on the square lattice.

The circuit ansatz. There are two main circuit ansatzes employed in this work: the ladderwise hardware-efficient ansatz (HEA) and the MERA ansatz. It is worth noting that both ansatzes work well in our model, and we utilize different ansatzes to demonstrate that the NNVQE framework is ansatz-agnostic and universally applicable.

The hardware-efficient ansatz [63] encompasses a range of ansatzes that are directly tailored to the given quantum hardware employed in the experiment, avoiding the circuit-depth overhead arising from transforming an arbitrary unitary into a sequence of local native gates. We employ ladderwise HEA (see Fig. 6) in this paper. The representation of the ladderwise HEA can be expressed as

$$|\psi_D\rangle = U|\psi_{D-1}\rangle, \quad (\text{A2})$$

where $U = \prod_{i,j} R_2(\theta_{ij}) \times \prod_i R_1(\theta_i)$ is a block of unitary (see Fig. 6 as a block of ladderwise HEA unitary), and D is the circuit depth representing the number of repetitions of the block. In each block, parameterized R_x and R_z gates are used as single-qubit gates $R_1(\theta_i) = R_z(\theta_{i1})R_x(\theta_{i2})$, and parameterized R_{xx} , R_{yy} , and R_{zz} gates are used as two-qubit gates $R_2(\theta_{ij}) = R_{yy}(\theta_{ij1})R_{xx}(\theta_{ij2})R_{zz}(\theta_{ij3})$ arranged in a ladderwise pattern. Before all the blocks, single-qubit rotation gates $R'_1(\theta_i) = R_x(\theta_{i1})R_z(\theta_{i2})R_x(\theta_{i3})$ are operated on all the qubits, transferring the initial state $|\mathbf{0}\rangle$ to $|\psi_0\rangle = R'_1(\theta_i)|\mathbf{0}\rangle$.

The multiscale entanglement renormalization ansatz (MERA) tensor network (see Fig. 7) can be adopted as a viable circuit ansatz for the VQE simulations. MERA starts from a single qubit in the $|\mathbf{0}\rangle$ state and progressively enlarges the Hilbert space by tensoring additional qubits in the $|\mathbf{0}\rangle$ state [56,57]. The treelike MERA tensor network constructed by this progressive enlargement corresponds to successively coarse-grained states and ultimately implements a fine-grained scaling transformation. The scale transformation is

$$|\psi_{l+1}\rangle = U_l(\theta) \left(|\psi_l\rangle \otimes |\mathbf{0}\rangle^{\otimes 2^l} \right), \quad (\text{A3})$$

where 2^l is the number of fresh qubits introduced in the l th layer and $U_l = \prod_{i=2}^{2^l} R_2(\theta_{i,i+1}) \times \prod_{j=1}^{2^l-1} R_2(\theta_{j,j+1}) \times \prod_{k=1}^{2^l} R_1(\theta_k)$ (all i are even and all j are odd). We denote D as the depth of the brickwork unitaries U_l . For example, we set $D = 2$ in Fig. 7. There are $D = 2$ blocks [yellow (1) and green (2)] in each U_l , and there are two brickwork unitaries U_1 and U_2 expanding the MERA network to four qubits. In each block, parameterized R_x and R_z gates are used as single-qubit gates $R_1(\theta_i) = R_z(\theta_{i1})R_x(\theta_{i2})$, and parameterized R_{xx} and R_{zz} gates are used as two-qubit gates $R_2(\theta_{ij}) = R_{zz}(\theta_{ij1})R_{xx}(\theta_{ij2})$. And before all the brickwork unitaries, single-qubit rotation gates $R'_1(\theta_i) = R_x(\theta_{i1})R_z(\theta_{i2})R_x(\theta_{i3})$ are operated on all the qubits.

The encoding neural network. The neural network of the NNVQE has one input layer, one hidden layer, and one output layer, which are all fully connected linear layers. The node number of the input layer corresponds to the number of Hamiltonian parameters, the node number of the hidden layer is a hyperparameter that varies with the circuit depth D , and the node number of the output layer corresponds to the number of PQC parameters. We also employ dropout layers after the hidden layer to avoid overfitting. To initialize the neural network, we choose the normal random initialization with the mean value = 0.0 and the standard deviation = 0.1. The optimizer we used in gradient descent is Adam and the learning schedule is the hyperparameter tuned for each case.

APPENDIX B: TECHNICAL DETAILS FOR THE SIMULATION

For results in Fig. 2, the Hamiltonian is the eight-qubit one-parameter 1D XXZ model. The training set of Δ is composed of 20 equispaced points in the interval of $[-3.0, 3.0]$; the test set is 201 equispaced values of Δ in the interval of $[-4.0, 4.0]$. The circuit ansatz is the eight-qubit MERA with depth $D = 1, 2, 3$. When $D = 1$, the node number of the hidden layer for the neural network is 20, and the node number of the output layer corresponds to the number of PQC parameters 74, without dropout and with dropout = 0.30; when $D = 2$, the node number of the hidden layer for the neural network is 20, and the node number

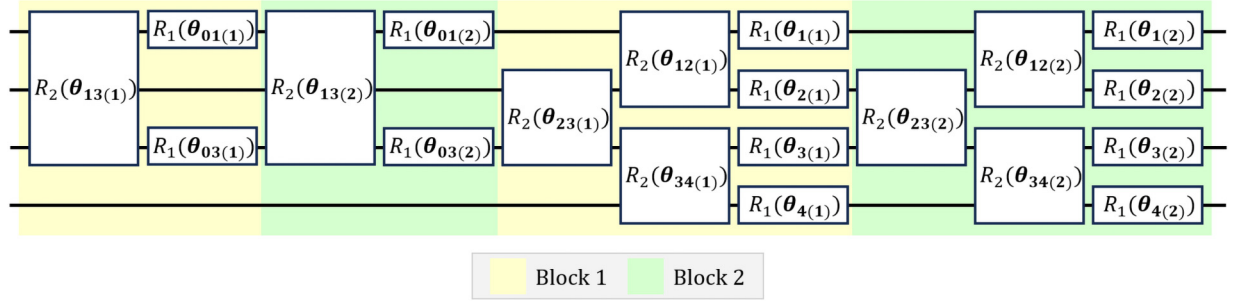


FIG. 7. Two blocks of MERA circuit with four qubits ($D = 2$). The yellow-shaded circuit is block 1 and the green-shaded circuit is block 2. In each block, there is a complete MERA structure. Each one-qubit gate $R_1(\theta_i)$ corresponds to $R_z(\theta_{i1})R_x(\theta_{i2})$ and the two-qubit gate $R_2(\theta_{ij})$ corresponds to $R_{zz}(\theta_{ij1})R_{xx}(\theta_{ij2})$. All parameters are independently optimized with no parameter sharing.

of the output layer corresponds to the number of PQC parameters 124, without dropout and with dropout = 0.05; when $D = 3$, the node number of the hidden layer is 30, and the node number of the output layer corresponds to the number of PQC parameters 174, without dropout and with dropout = 0.20. The starting learning rate for the optimizer is 0.009, and it decays to its 70% every 1000 steps. The maximum iteration for the optimization is 2500.

When the active learning strategy is explored as shown in Fig. 3, the Hamiltonian we used is the eight-qubit one-parameter 1D XXZ model. The training set of Δ is 11 actively learned points in the interval of $[-3.0, 3.0]$; the test set is 201 equispaced values of Δ in the interval of $[-3.0, 3.0]$. The ansatz is the eight-qubit MERA with $D = 2$. The node number of the hidden layer for the neural network is 25, and the node number of the output layer corresponds to the number of PQC parameters 124, without dropout and with dropout = 0.20. The hyperparameter μ in the cost function of active learning is 6.0. The starting learning rate is 0.009, and it decays to its 85% every 200 step. The maximum iteration is 2500.

For the training efficiency results presented in Fig. 4, the Hamiltonian is a 12-qubit one-parameter 1D XXZ model. The training set is $\Delta = 1.5$ or $\Delta = 2.0$ with only one point. The circuit ansatz is 12-qubit HEA with $D = 3$. The node number of the hidden layer is 36, and the node number of the output layer corresponds to the number of PQC parameters 216, with dropout = 0.20. The learning rate is 0.009. The maximum iteration is 100.

For the result in Fig. 5, the Hamiltonian is a 12-qubit two-parameter 1D XXZ model. The training set of Δ is composed of ten equispaced points in the interval of $[-1.0, 1.0]$, and λ is composed of five equispaced points in the interval of $[0.0, 1.0]$; the test set of Δ is composed of 101 equispaced points in the interval of $[-1.0, 1.0]$, and λ is composed of 51 equispaced points in the interval of $[0.0, 1.0]$. The ansatz is 12-qubit HEA with $D = 1, 2$; the node number of the hidden layer is chosen at 40, and the node number of the output layer corresponds to the number of PQC parameters (96 when $D = 1$ and 156 when $D = 2$),

with dropout = 0.2. The starting learning rate is 0.01, and it decays to its 70% every 800 step. The maximum iteration is 4000.

APPENDIX C: RESULT OF TWO-PARAMETER 1D XXZ MODEL WHEN $n = 8$

Here, we use a two-parameter eight-qubit 1D XXZ model as the Hamiltonian. The training set of Δ is composed of ten equispaced points in the interval of $[-1.0, 1.0]$, and λ is composed of five equispaced points in the interval of $[0.0, 1.0]$; the test set of Δ is composed of 101 equispaced points in the interval of $[-1.0, 1.0]$, and λ is composed of 51 equispaced points in the interval of $[0.0, 1.0]$; the ansatz is the eight-qubit HEA with $D = 1, 2$; the node number of the hidden layer is chosen at 25, and the node number of the output layer corresponds to the number of PQC parameters (64 when $D = 1$ and 104 when $D = 2$), dropout = 0.2. The starting learning rate is 0.01, and it decays to its 70% every 800 step. The maximum iteration is 4000.

The numerical results are presented in Fig. 9. Same as the $n = 12$ result in the main text, the NNVQE with a two-input neural network shows excellent performance in

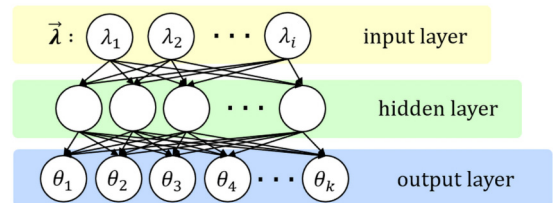


FIG. 8. Neural network of the NNQVE. The neural network has one input layer, one hidden layer, and one output layer, which are all fully connected linear layers. The number of nodes of the input layer corresponds to the number of Hamiltonian parameters. The node number of the hidden layer is a hyperparameter that varies with the circuit depth D . The node number of the output layer corresponds to the number of PQC parameters.

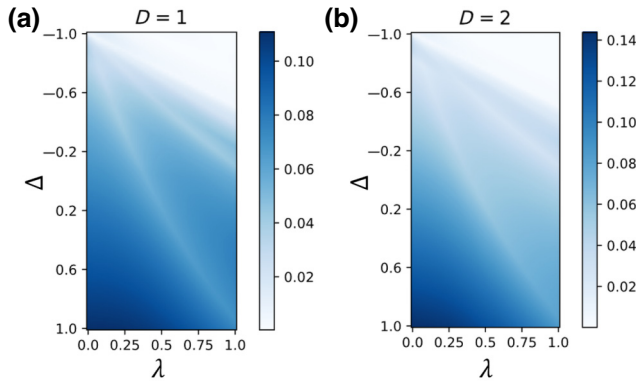


FIG. 9. Relative errors of ground-state energies for an $n = 8$ 1D XXZ spin chain with two tunable Hamiltonian parameters, using the hardware-efficient ansatz with circuit depth $D = 1, 2$.

estimating ground-state energy, which highlights the effectiveness of our approach in estimating multiple-parameter Hamiltonian of different system sizes.

APPENDIX D: RESULT OF SINGLE-PARAMETER 4×4 2D XXZ MODEL

In this section, we use a single-parameter 4×4 2D XXZ model as the testbed Hamiltonian. The open-boundary

2D XXZ model, similar to the 1D XXZ model, has the transverse-field strength λ and the anisotropy parameter Δ :

$$\hat{H}_{2D-XXZ} = \sum_{ij} (X_i X_j + Y_i Y_j + \Delta Z_i Z_j) + \lambda \sum_i Z_i, \quad (D1)$$

where i, j runs over nearest neighbor pairs on a square lattice. We fix the transverse field strength to $\lambda = 5.0$, and the anisotropy parameter Δ is chosen as the varying parameter.

The mapping from the qubits of the MERA to the 2D square lattice points is shown in Fig. 10.

The training set of Δ is composed of 12 equispaced points in the interval of $[0.0, 4.4]$; the test set is 148 equispaced values of Δ in the interval of $[0.0, 4.4]$. The circuit ansatz is 16-qubit MERA with depth $D = 3$. The node number of the hidden layer is 700 for the encoding neural network, and the node number of the output layer corresponds to the number of PQC parameters 384, with dropout = 0.20. The starting learning rate for the optimizer is 0.01, and it decays to its 85% every 300 step. The maximum iteration for the optimization is 5000.

The numerical results are presented in Fig. 11(a). We can see that, though we have changed to a more complicated model, the results of NNVQE are still similar to the standard VQE. To better compare the NNVQE effectiveness for different models and system sizes, we utilize the

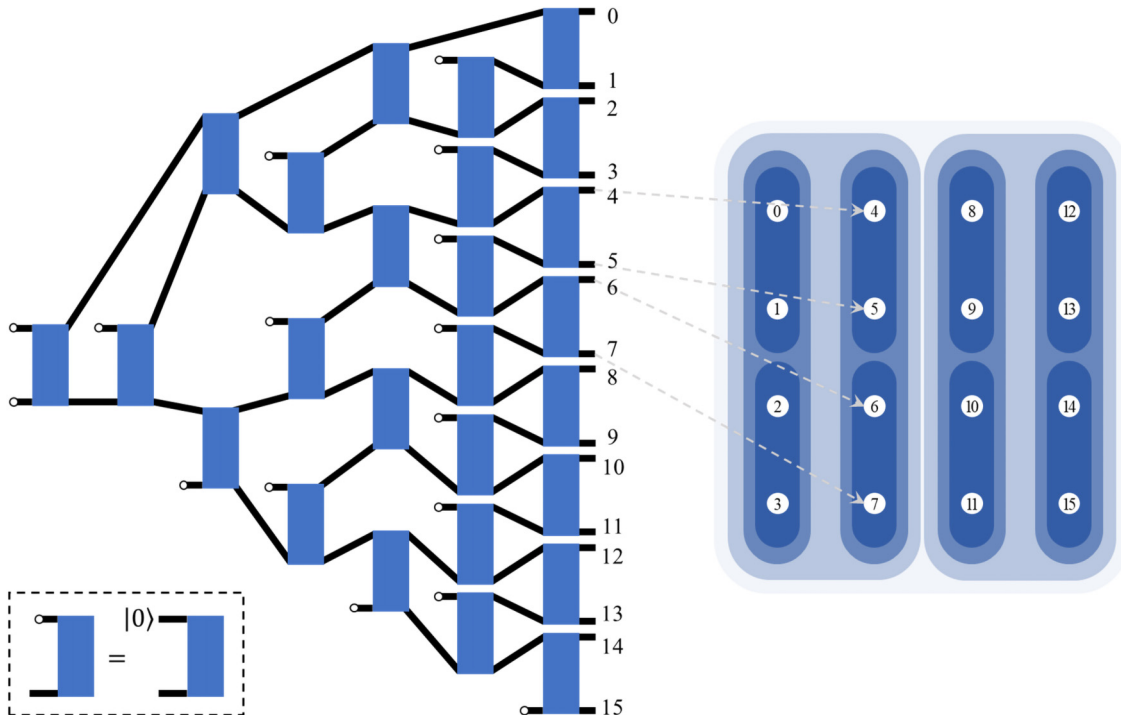


FIG. 10. The 16-qubit 1D MERA circuit diagram and its mapping to the 4×4 2D square lattice for the XXZ model. Each number of the qubits marked at the rear of the 1D MERA diagram corresponds to the number marked on the lattice point (white). Four qubit-lattice point mappings are chosen for demonstration. The darker the color on the lattice, the closer the relationship of the corresponding qubits on the MERA. Note that the fresh qubit $|0\rangle$ injected in each layer is divided into two sublayers due to the even-odd brickwall layout of the two-qubit gates.

measure as the ratio between the energy estimation from the NNVQE without fine tuning and the converged energy estimation of the standard VQE for the same parameterized Hamiltonian. The ratio gives the percent of energy that can be recovered by the pretrained model and avoids the impact of the standard VQE performance fluctuation. We compare the results here with the result of the eight-qubit single-parameter 1D XXZ model mentioned in the main text in Figs. 11(b) and 11(c), and find that the performance is similar, indicating that NNVQE not only has good scalability but can also work well universally for many different systems.

APPENDIX E: SIMULATION RESULTS IN THE PRESENCE OF QUANTUM NOISES

We use the eight-qubit single-parameter 1D XXZ model as the Hamiltonian. The training set of Δ is composed of 20 equispaced points in the interval of $[-3.0, 3.0]$. The circuit ansatz is the eight-qubit MERA with depth $D = 2$. The node number of the hidden layer is 700, and the node number of the output layer corresponds to the number of PQC parameters 124, with dropout = 0.20. The starting learning rate for the optimizer is 0.01, and it decays to its 85% every 300 step. The maximum iteration for the optimization is 2500.

The noise model we utilized for the noisy emulator is as follows: the depolarizing quantum channels are applied right after each two-qubit gate. The depolarizing error rates in the x , y , and z directions p_x, p_y, p_z are all set to 1×10^{-3} . The Kraus representation of the channel is given as $\rho \rightarrow (1 - p_x - p_y - p_z)\rho + p_x X \rho X + p_y Y \rho Y + p_z Z \rho Z$.

The numerical results are presented in Fig. 12. We find that the consideration of noise does not render NNVQE less effective overall. It is an interesting future direction to explore the noise resilience in the NNVQA for different models, phases, and with different noise profiles.

APPENDIX F: A COMPARISON BETWEEN NNVQE AND META-VQE

We use the eight-qubit one-parameter 1D XXZ model as the Hamiltonian, which is the same Hamiltonian as in the meta-VQE work. The training set of Δ is composed of 20 equispaced points in the interval of $[-3.0, 3.0]$, and the test set is 201 equispaced values of Δ in the interval of $[-4.0, 4.0]$.

The NNVQE circuit ansatz and the meta-VQE processing layer are set the same as the MERA circuit. The encoding layer of meta-VQE is the same as Ref. [60] (see Fig. 13). The encoding function it uses is $f(\Delta, \phi) = w\Delta + \phi$, where Δ is the Hamiltonian parameter and w and ϕ are encoding parameters to optimize.

As for the neural network of the NNVQE (see Fig. 8), we choose the hidden-layer node number = 20 when $D = 1, 2$, and the hidden-layer node number = 30 when $D = 3$.

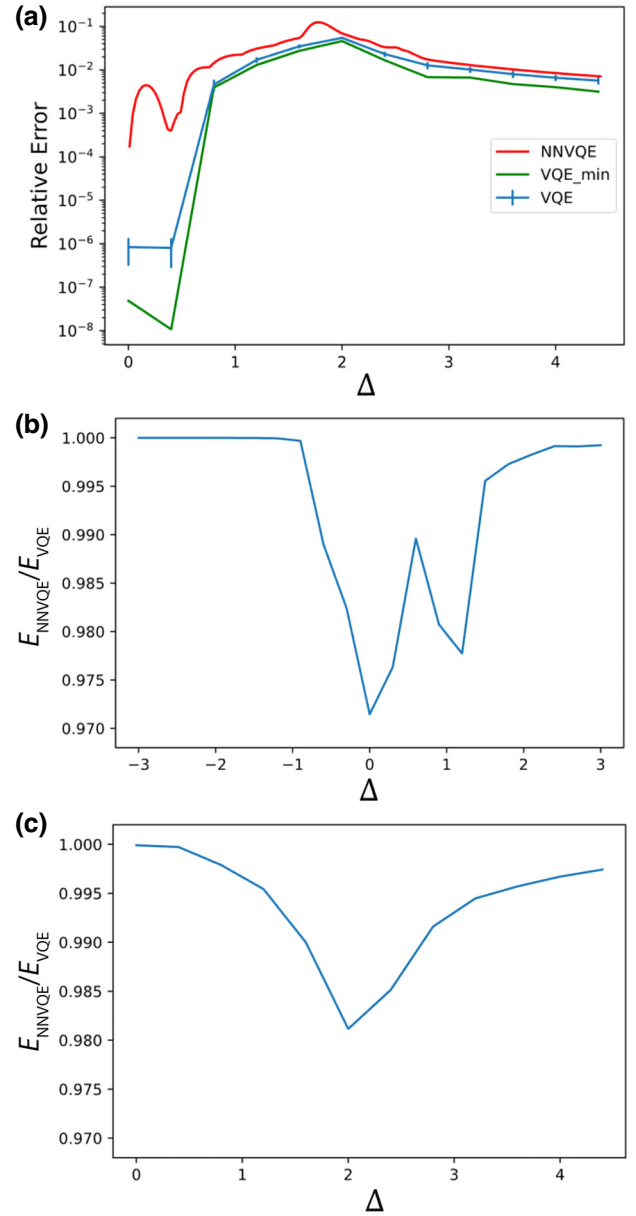


FIG. 11. Results on single-parameter 4×4 2D XXZ model with a transverse-field strength $\lambda = 5.0$ within NNVQE framework. (a) Relative errors of ground-state energies of standard VQE and NNVQE. Standard VQE and NNVQE share the same circuit with circuit depth $D = 3$. The red line is the NNVQE with dropout. The blue line is the average result of 100 batches of standard VQE with different random initializations, while the green line corresponds to the best result among the 100 trials. (b), (c) are the ratio of the ground-state energy obtained by NNVQE to standard VQE. (b) is the result of the eight-qubit single-parameter 1D XXZ model mentioned in the main paper; while (c) is the result of 4×4 2D XXZ model.

Dropout was also employed to avoid overfitting. The starting learning rate is 0.009, and it decays to its 70% every 1000 step. The max iteration is 2500.

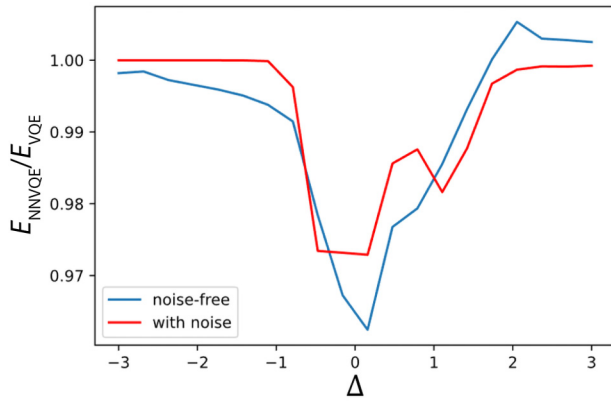


FIG. 12. Simulation results in the presence of quantum noises. The model used is the eight-qubit single-parameter 1D XXZ model, and the circuits used are identical to the main text. We calculate the ratio of the ground-state energy obtained by NNVQE to standard VQE. The blue line corresponds to the results from the noise-free simulation, while the red line is from the noisy simulation.

The result of meta-VQE and NNVQE (with and without dropout) is shown in Fig. 14 with different D s. In the region of the training set, NNVQE performs far better than meta-VQE, especially when the circuit gets deeper. When it comes to the training points not in the test region, the error of our NNVQE is significantly lower than that of the meta-VQE. The result shows that our NNVQE can give more precise ground-state energy estimation than meta-VQE and has better generalization ability without any fine tuning.

APPENDIX G: CIRCUIT PARAMETERS CHANGE VISUALIZATION

To gain deeper insights, we examine the variations of gate parameters by plotting their cosine values with respect to the Hamiltonian parameter Δ , shown in Fig. 16. The

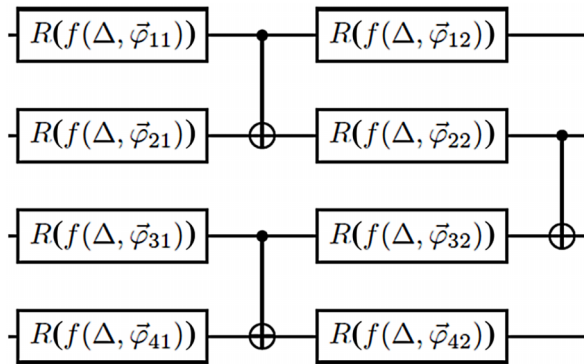


FIG. 13. The four-qubit example encoding layer of meta-VQE from Ref. [60]. Each $R(\theta)$ gate corresponds to $R_z(\theta_1)R_x(\theta_2)$. The function used for the encoding layer is $f(\Delta, \phi) = w\Delta + \phi$.

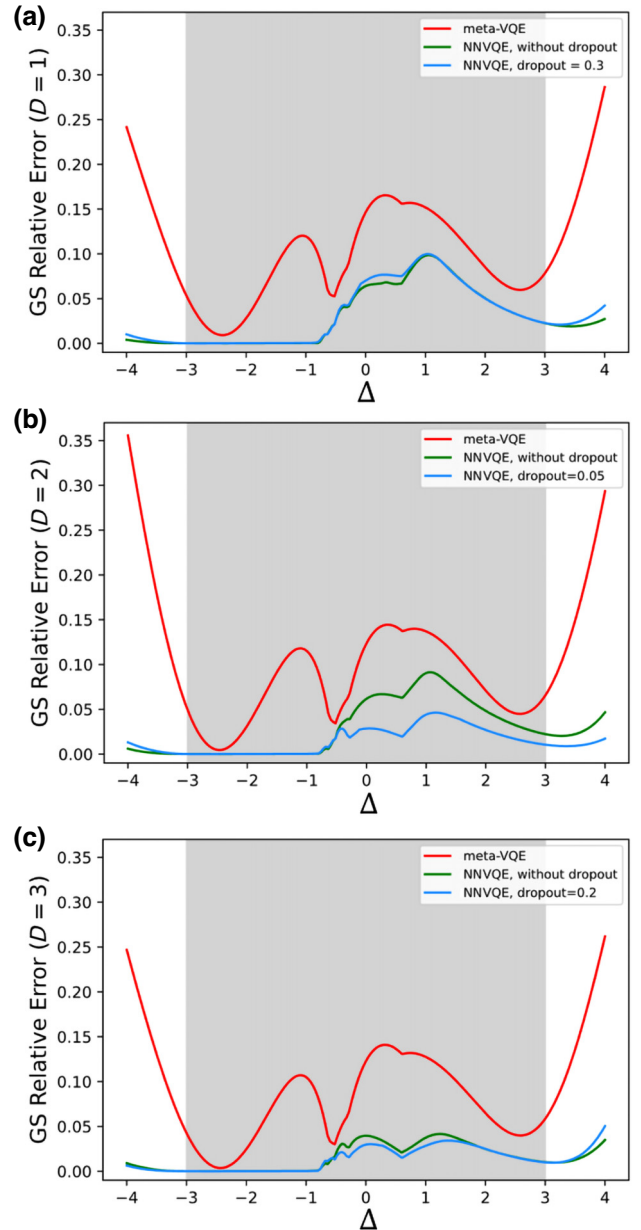


FIG. 14. A comparison between NNVQE and meta-VQE. The parameterized Hamiltonian is an eight-qubit one-parameter 1D XXZ spin chain with a transverse field strength fixed at $\lambda = 0.75$. The ansatz is MERA with (a) $D = 1$, (b) $D = 2$, and (c) $D = 3$. The training set interval is on the gray background and the whole line is in the interval of the test set.

sequence number of the subfigures corresponds to the gate number in Fig. 15. As shown in the figures, the relationship between Δ and the gate parameters (or the cosine of the gates' parameters) is neither linear nor low-order polynomial, which could explain why our NNVQE performs far better than meta-VQE that utilizes simple analytical formula [60]. The abrupt change of the circuit parameters is often related to the region near criticality (see Fig. 18).

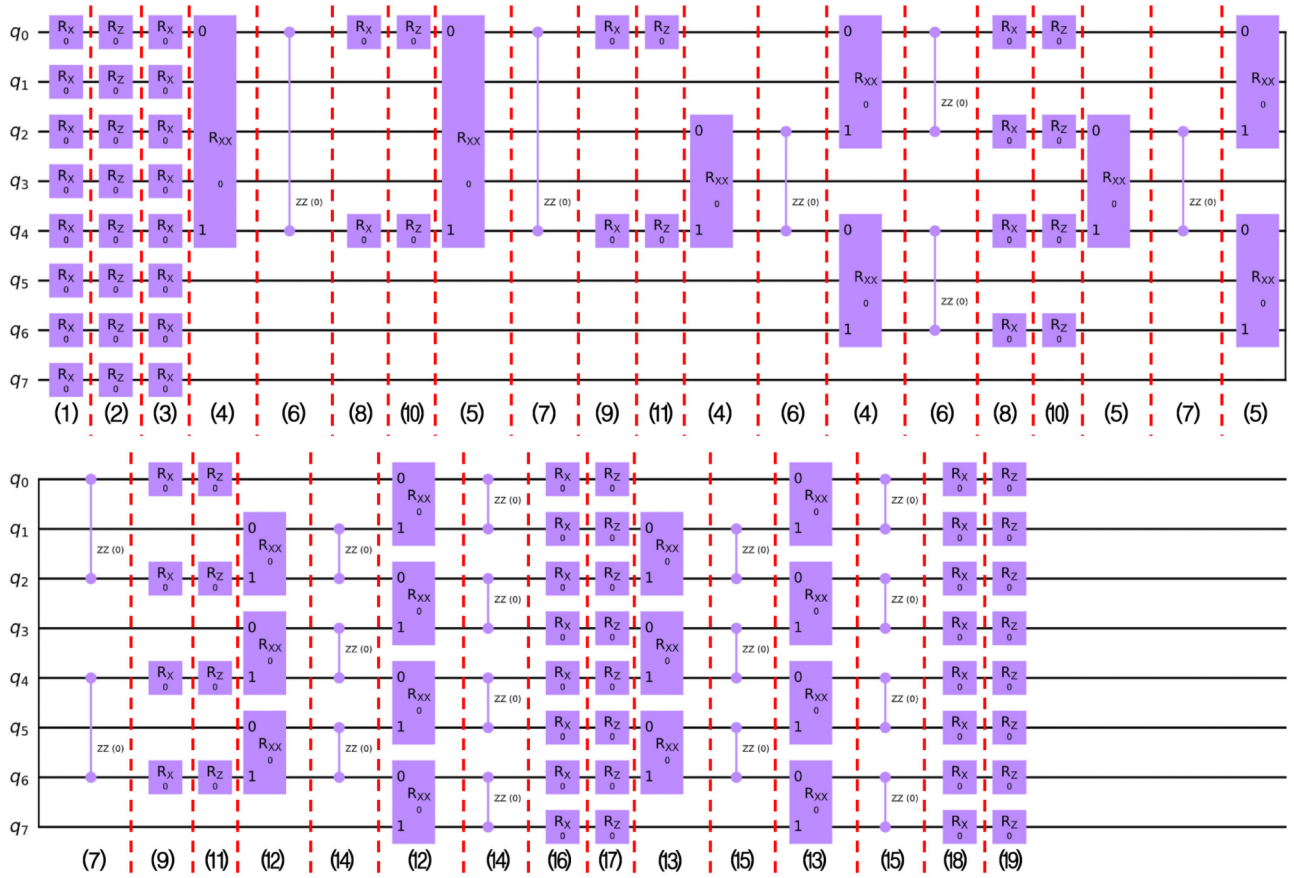


FIG. 15. $n = 8$ MERA circuit with $D = 2$.

Instead, smooth variations are observed on the ferromagnetic (FM) phase and the antiferromagnetic (AFM) phase. Consequently, this nontrivial relationship indicates that employing a neural network as the encoding module is of great necessity to capture various quantum phases with the same setup.

The model investigated is the MERA circuit with $D = 2$ and dropout = 0.05.

APPENDIX H: ADDITIONAL RESULTS ON THE OPTIMIZATION PROCESS SPEEDUP BY NNVQE

It is mentioned that NNVQE could speed up the training process, allowing rapid convergence. Here we use the same ansatz circuit, Hamiltonian, and training schedule as in the main text but report further details.

To avoid the impact of parameter initialization, the initial PQC parameters of the standard VQE are set the same as the initial PQC parameters of the NNVQE, which are generated by a randomly initialized neural network. The results of $\Delta = 1.5, 2.0$ are shown in the main text and we show the supplemental result for $\Delta = 1.0$ in Fig. 17. We can see that NNVQE converges much faster than the standard VQE when training. Also, NN-PQC hybrid architecture brings a more dramatic change in the PQC parameters

at the beginning stage of the optimization process, which might help escape barren plateaus.

Furthermore, we investigate the results of different Δ s and system sizes ns . The result is summarized in Table I. We can see that the convergent rate of the NNVQE is higher than the VQE. The number of epoch budgets is 100.

The setup hyperparameters for Table I is as follows. When $n = 8$, the ansatz is the eight-qubit MERA with $D = 3$. The node number of the hidden layer is 25, and the node number of the output layer corresponds to the number of PQC parameters 144, with dropout = 0.20. The learning rate is 0.009. The max iteration is 100. When $n = 10$, the ansatz is the eight-qubit MERA with $D = 3$. The node number of the hidden layer is 32, and the node number of the output layer corresponds to the number of PQC parameters 180, with dropout = 0.20. The learning rate is 0.009. The max iteration is 100.

APPENDIX I: QUANTUM SOFTWARE FRAMEWORK

All the high-performance numerical simulations in this work are conducted with TensorCircuit [55]: an open-source, high-performance, full-featured quantum software framework for the NISQ era. The software simulates the

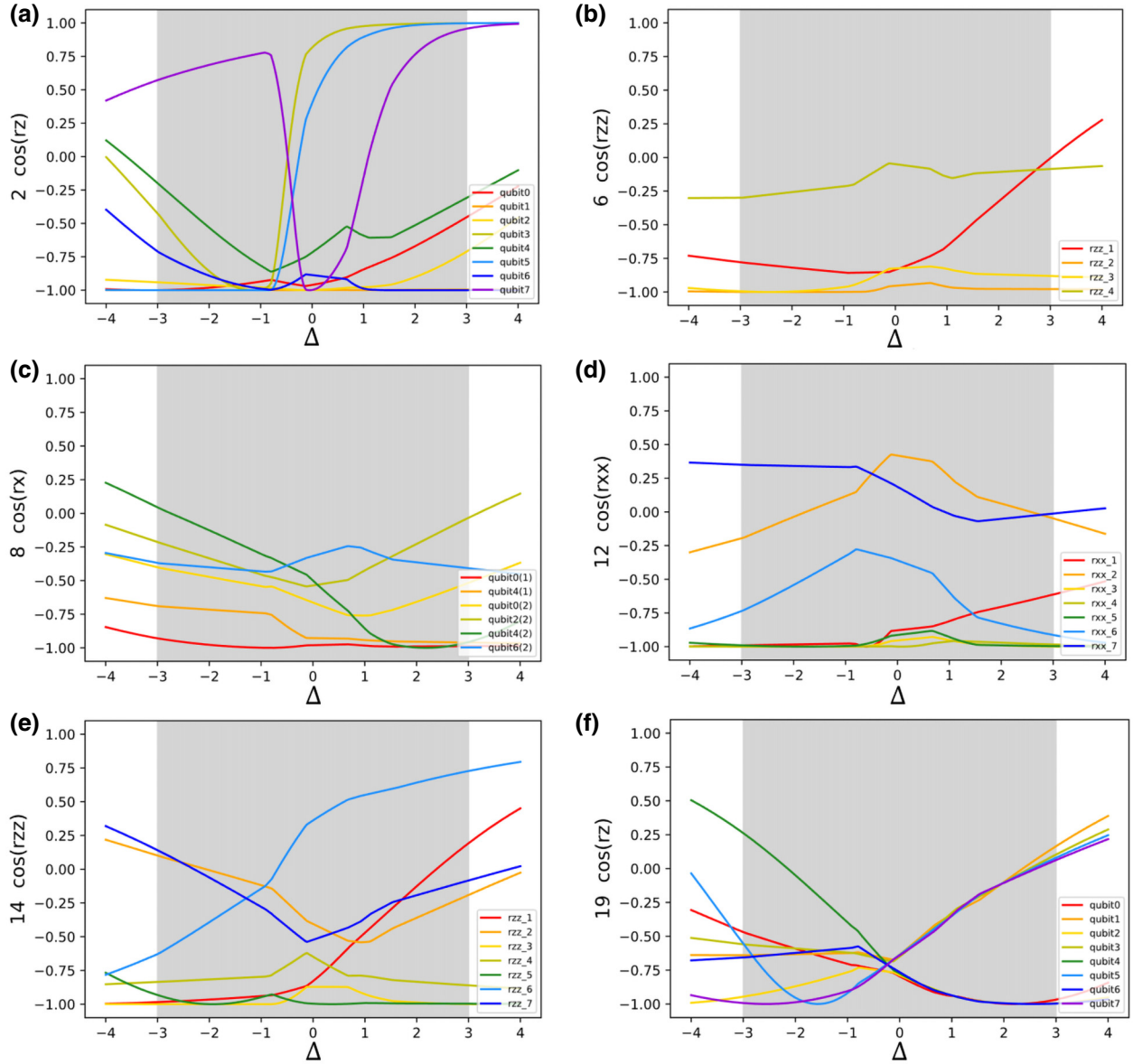


FIG. 16. Parameters of gates with respect to Δ . The ansatz used is MERA with $D = 2$. Numbers in the y axis of each subplot correspond to a set of gate numbers (below the circuit) in Fig. 15. Qubit numbers in the labels of the subplots associated with the qubits (do not consider qubits without gates) in order, arranged from left to right and from top to bottom.

quantum circuit with an advanced tensor-network contraction engine and supports modern machine-learning engineering paradigms: automatic differentiation, vectorized parallelism, just-in-time compilation, and GPU acceleration. It is specifically suitable to simulate the hybrid system with both neural networks and quantum circuits.

APPENDIX J: THE PHASE DIAGRAM OF THE 1D AND 2D XXZ MODEL

The phase diagram of the 1D XXZ model is shown in Fig. 18. The three phases are separated by two curves hs

and hc

$$hs = d(1 + \Delta) \tag{J1}$$

$$hc = \frac{\pi \sinh \lambda}{\lambda} \sum_{n=-\infty}^{\infty} \operatorname{sech} \frac{\pi^2}{2\lambda} (1 + 2n), \tag{J2}$$

where $d = 1$ is the dimension of the model and $\lambda = \operatorname{arccosh} \Delta$.

The phase diagram of the 2D XXZ model is shown in Fig. 19.

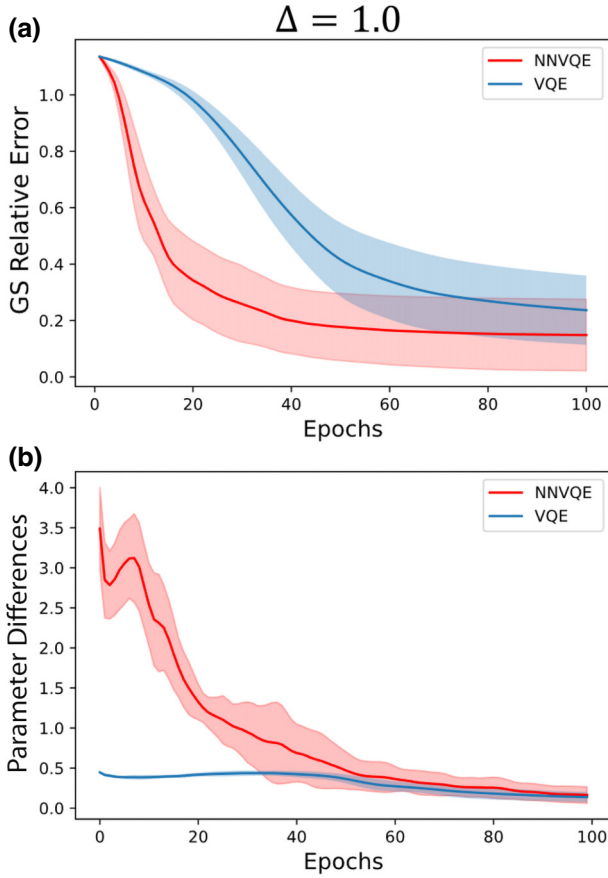


FIG. 17. Speedup in the optimization process of NNVQE and the corresponding PQC parameter changes. (a) The ground-state energy relative errors for an $n = 12$ XXZ spin chain when $\Delta = 1.0$ are shown with respect to epochs. Hardware-efficient ansatz with $D = 3$ is used. The red and blue lines correspond to the ground-state energy relative errors and standard deviation of NNVQE and VQE, respectively. (b) The summed parameter updates when training the VQE and NNVQE in corresponding Δ .

APPENDIX K: RELATED WORKS

Meta-VQE [60] aims to solve the problem of parameterized Hamiltonian. They divide the PQC into two parts:

TABLE I. Convergent rate. The convergent threshold of ground-state energy relative error is set to 0.1, we call the model converged when the energy error is below such a value and within the optimization epoch budget. We call the number of convergent models divided by total trials as the convergent rate. The convergence rates of an XXZ spin chain with $\Delta = 1.0, 1.5, 2.0$ and $n = 8, 10, 12$ within the 100 epochs are reported. The PQC ansatz is the hardware-efficient ansatz with $D = 3$, and the optimizer is Adam.

Δ	1.0			1.5			2.0		
	8	10	12	8	10	12	8	10	12
NNVQE / %	60	60	70	80	70	60	90	85	85
VQE / %	5	0	20	5	5	0	20	30	10

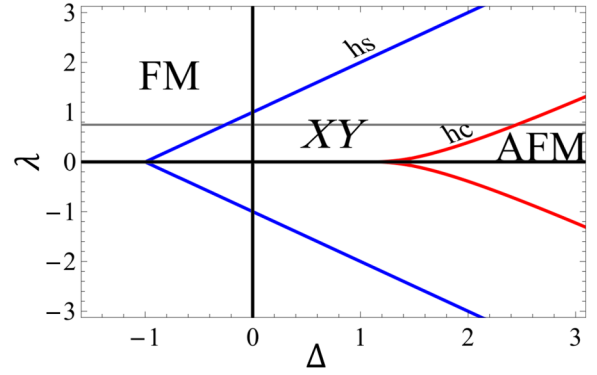


FIG. 18. Phase diagram of the 1D XXZ model [62]. The line hs separates the XY phase from the ferromagnetic (FM) phase. The curve hc separates the antiferromagnetic (AFM) phase from the XY phase. The gray line corresponds to $\lambda = 0.75$.

the encoding layer and the processing layer. In the encoding layer, Hamiltonian parameters are mapped to some simple formula. And the processing layer is a standard VQE requiring fine tuning for each Hamiltonian parameter. The method has some drawbacks though. Firstly, most of the circuit parameters still need to be trained or fine tuned on separate Hamiltonian instances. Secondly, the expressive power of their parameter prediction model is very weak to achieve the given accuracy. In contrast, our NNVQE encodes all parameters in the circuit and utilizes a more complicated neural network for the prediction. These improvements greatly improve the performance and accuracy of the method due to the high expressiveness of neural networks. Besides, our method totally avoids retraining

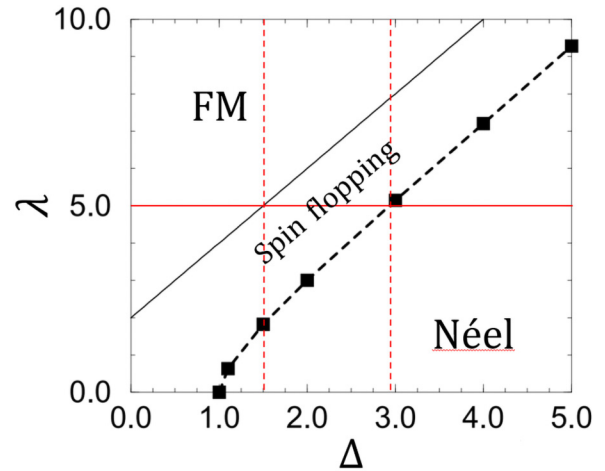


FIG. 19. Phase diagram of the 2D XXZ model [70]. There exist three phases: (i) Néel ordered phase, (ii) spin-flopping phase, and (iii) fully saturated ferromagnetic phase. The solid line is $2(1 + \Delta)$. The horizontal red line corresponds to $\lambda = 5.0$; the vertical dashed red lines are indicators for the intersection of the red horizontal line and the phase transition line.

or fine tuning on each separate Hamiltonian instance (see Fig. 14) since all circuit parameters are encoded.

In this work, we implement the VQE as a representative example of VQAs to demonstrate the efficacy of our NNVQA framework, but other NNVQAs for different applications can also show their effectiveness. Some works employed similar ideas in this work for optimization algorithm (QAOA). Since identifying the optimal parameters is a difficult task in QAOA, researchers employ neural networks or other machine-learning techniques to find better initialization parameters. A spectrum of neural-network architectures, including deep neural networks (DNN) [71], convolutional neural networks (CNN) [72], and graph neural networks (GNN) [73] have been explored. By employing these neural-network-encoding methods, significant progress has been made in enhancing the performance and efficiency of QAOA. However, in the QAOA cases, the neural-network-encoded parameters are only utilized as a good initialization point and further fine tuning on QAOA is still required. In contrast, in the NNVQE case, we directly use the neural-network-encoded parameters as the final parameter choice, which has already given satisfying performance in real applications.

Our work also shares some similarities with Ref. [74]. In their work, with the Hamiltonian parameters as input to the GNN, they extract some intermediate representation as the input for another generative network to generate the classical shadows [75] of the ground state, which is an efficient but approximate representation of the quantum state. In our framework, we also extract the intermediate representation via encoding neural network and Hamiltonian as input. However, we aim to directly generate the ground state in the quantum form on a quantum computer instead of only recovering a classical shadow representation with many known limitations.

-
- [1] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [2] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, *Rev. Mod. Phys.* **94**, 015004 (2022).
- [3] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021). <https://www.nature.com/articles/s42254-021-00348-9>.
- [4] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, *J. Phys. Soc. Jpn* **90**, 032001 (2021).
- [5] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, The variational quantum eigensolver: A review of methods and best practices, *Phys. Rep.* **986**, 1 (2022).
- [6] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* **5**, 4213 (2014).
- [7] J. McClean, J. Romero, R. Babbush, and A.-G. Alán, The theory of variational hybrid quantum-classical algorithms, *New J. Phys.* **18**, 023023 (2016).
- [8] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, *ArXiv:1411.4028* (2014).
- [9] S. C. Kak, Quantum neural computing, *Adv. Imaging Electron Phys.* **94**, 259 (1995). <https://www.sciencedirect.com/science/article/pii/S1076567008701472>.
- [10] P. Pylkkänen and P. Pylkkö, in *Proceedings of the International Symposium, Publications of the Finnish Artificial Intelligence Society: International conferences* (Finnish Artificial Intelligence Society, Lapland, Finland, 1995). <https://books.google.com.hk/books?id=SyUmAAAACAAJ>.
- [11] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, Training deep quantum neural networks, *Nat. Commun.* **11**, 808 (2020).
- [12] I. Cong, S. Choi, and M. Lukin, Quantum convolutional neural networks, *Nat. Phys.* **15**, 1273 (2019).
- [13] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, *Phys. Rev. A* **101**, 032308 (2020).
- [14] J. Romero, J. Olson, and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, *Quantum Sci. Technol.* **2**, 045001 (2017).
- [15] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, A generative modeling approach for benchmarking and training shallow quantum circuits, *npj Quantum Inf.* **5**, 45 (2019).
- [16] L. Cheng, Y.-Q. Chen, S.-X. Zhang, and S. Zhang, Error-mitigated quantum approximate optimization via learning-based adaptive optimization, *ArXiv:2303.14877* (2023).
- [17] H. L. Tang, V. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou, Qubit-adapt-VGE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor, *PRX Quantum* **2**, 020310 (2021).
- [18] S. Kanno and T. Tada, Many-body calculations for periodic materials via restricted Boltzmann machine-based VGE, *Quantum Sci. Technol.* **6**, 025015 (2021).
- [19] Jarrod R. McClean, S. Boixo, Vadim N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [20] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, *Nat. Commun.* **12**, 1791 (2021).
- [21] M. Cerezo and P. Coles, Higher order derivatives of quantum neural networks with barren plateaus, *Quantum Sci. Technol.* **6**, 035006 (2021).
- [22] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, Noise-induced barren plateaus in variational quantum algorithms, *Nat. Commun.* **12**, 6961 (2021).

- [23] A. Arrasmith, Z. Holmes, M. Cerezo, and P. J. Coles, Equivalence of quantum barren plateaus to cost concentration and narrow gorges, *Quantum Sci. Technol.* **7**, 045015 (2022).
- [24] L. Bittel and M. Kliesch, Training variational quantum algorithms is NP-hard, *Phys. Rev. Lett.* **127**, 120502 (2021).
- [25] E. R. Anschuetz and B. T. Kiani, Quantum variational algorithms are swamped with traps, *Nat. Commun.* **13**, 7760 (2022). <https://www.nature.com/articles/s41467-022-35364-5>.
- [26] W. J. Huggins, J. R. McClean, N. C. Rubin, Z. Jiang, N. Wiebe, K. B. Whaley, and R. Babbush, Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers, *npj Quantum Inf.* **7**, 23 (2021).
- [27] B. Wu, J. Sun, Q. Huang, and X. Yuan, Overlapped grouping measurement: A unified framework for measuring quantum states, *Quantum* **7**, 896 (2023).
- [28] M. Kashif and S. Al-kuwari, Resqnets: A residual approach for mitigating barren plateaus in quantum neural networks, *EPJ Quantum Technol.* **11**, 4 (2024).
- [29] Q. Miao and T. Barthel, Isometric tensor network optimization for extensive hamiltonians is free of barren plateaus, *ArXiv:2304.14320* (2023).
- [30] C.-Y. Park and N. Killoran, Hamiltonian variational ansatz without barren plateaus, *ArXiv:2302.08529* (2023).
- [31] N. F. Robertson, A. Akhriev, J. Vala, and S. Zhuk, Escaping barren plateaus in approximate quantum compiling, *ArXiv:2210.09191* (2022).
- [32] A. A. Mele, G. B. Mbeng, G. E. Santoro, M. Collura, and P. Torta, Avoiding barren plateaus via transferability of smooth solutions in a hamiltonian variational ansatz, *Phys. Rev. A* **106**, L060401 (2022).
- [33] X. Liu, G. Liu, J. Huang, H.-K. Zhang, and X. Wang, Mitigating barren plateaus of variational quantum eigensolvers, *ArXiv:2205.13539* (2022).
- [34] L. Friedrich and J. Maziero, Avoiding barren plateaus with classical deep neural networks, *Phys. Rev. A* **106**, 042433 (2022).
- [35] A. Kulshrestha and I. Safro, in *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE Computer Society, Los Alamitos, CA, USA, 2022), p. 197.
- [36] L. Broers and L. Mathey, Reducing barren plateaus in quantum algorithm protocols, *ArXiv:2111.08085* (2022).
- [37] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, Absence of barren plateaus in quantum convolutional neural networks, *Phys. Rev. X* **11**, 041011 (2021).
- [38] S. Liu, S.-X. Zhang, S.-K. Jian, and H. Yao, Training variational quantum algorithms with random gate activation, *Phys. Rev. Res.* **5**, L032040 (2023).
- [39] H.-K. Zhang, S. Liu, and S.-X. Zhang, Absence of barren plateaus in finite local-depth circuits with long-range entanglement, *ArXiv:2311.01393* (2023).
- [40] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nat. Phys.* **14**, 447 (2018). <https://www.nature.com/articles/s41567-018-0048-5>.
- [41] G. Verdon, J. Marks, S. Nanda, S. Leichenauer, and J. Hidary, Quantum hamiltonian-based models and the variational quantum thermalizer algorithm, *ArXiv:1910.02071* (2019).
- [42] J.-G. Liu, L. Mao, P. Zhang, and L. Wang, Solving quantum statistical mechanics with variational autoregressive networks and quantum circuits, *Mach. Learn.: Sci. Technol.* **2**, 025011 (2021).
- [43] C.-Y. Hsieh, Q. Sun, S. Zhang, and C. K. Lee, Unitary-coupled restricted Boltzmann machine ansatz for quantum simulations, *npj Quantum Inf.* **7**, 2056 (2021).
- [44] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Differentiable quantum architecture search, *Quantum Sci. Technol.* **7**, 045023 (2022).
- [45] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Neural predictor based quantum architecture search, *Mach. Learn.: Sci. Technol.* **2**, 045027 (2021).
- [46] M. Benedetti, B. Coyle, M. Fiorentini, M. Lubasch, and M. Rosenkranz, Variational inference with a quantum computer, *Phys. Rev. Appl.* **16**, 044057 (2021).
- [47] S.-X. Zhang, Z.-Q. Wan, C.-K. Lee, C.-Y. Hsieh, S. Zhang, and H. Yao, Variational quantum-neural hybrid eigensolver, *Phys. Rev. Lett.* **128**, 120502 (2022).
- [48] S. Zhang, Z. Wan, C. Hsieh, H. Yao, and S. Zhang, Variational quantum-neural hybrid error mitigation, *Adv. Quantum Technol.* **6**, 2300147 (2023).
- [49] E. R. Bennewitz, F. Hopfmueller, B. Kulchytskyy, J. Carrasquilla, and P. Ronagh, Neural error mitigation of near-term quantum simulations, *Nat. Mach. Intell.* **4**, 618 (2022).
- [50] P. Huembeli, G. Carleo, and A. Mezzacapo, Entanglement forging with generative neural network models, *ArXiv:2205.00933* (2022).
- [51] P. de Schouepnikoff, O. Kiss, S. Vallecorsa, G. Carleo, and M. Grossi, Hybrid ground-state quantum algorithms based on neural Schrödinger forging, *ArXiv:2307.02633* (2023).
- [52] D.-B. Zhang, B.-L. Chen, Z.-H. Yuan, and T. Yin, Variational quantum eigensolvers by variance minimization, *Chin. Phys. B* **31**, 120301 (2022).
- [53] R. Santagati, J. Wang, A. A. Gentile, S. Paesani, N. Wiebe, J. R. McClean, S. Morley-Short, P. J. Shadbolt, D. Bonneau, J. W. Silverstone, D. P. Tew, X. Zhou, J. L. O'Brien, and M. G. Thompson, Witnessing eigenstates for quantum simulation of Hamiltonian spectra, *Sci. Adv.* **4**, 1 (2018).
- [54] S. Liu, S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Probing many-body localization by excited-state variational quantum eigensolver, *Phys. Rev. B* **107**, 024204 (2023).
- [55] S.-X. Zhang, J. Allcock, Z.-Q. Wan, S. Liu, J. Sun, H. Yu, X.-H. Yang, J. Qiu, Z. Ye, Y.-Q. Chen, C.-K. Lee, Y.-C. Zheng, S.-K. Jian, H. Yao, C.-Y. Hsieh, and S. Zhang, TensorCircuit: A quantum software framework for the NISQ era, *Quantum* **7**, 912 (2023).
- [56] G. Vidal, Class of quantum many-body states that can be efficiently simulated, *Phys. Rev. Lett.* **101**, 110501 (2008).
- [57] G. Evenbly and G. Vidal, Algorithms for entanglement renormalization, *Phys. Rev. B* **79**, 144108 (2009).
- [58] I. H. Kim and B. Swingle, Robust entanglement renormalization on a noisy quantum computer, *ArXiv:1711.07500* (2017).
- [59] T. J. Sewell, N. Bao, and S. P. Jordan, Variational quantum simulation of the critical Ising model with symmetry averaging, *Phys. Rev. A* **107**, 042620 (2023).

- [60] A. Cervera-Lierta, J. S. Kottmann, and A. Aspuru-Guzik, Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation, *PRX Quantum* **2**, 020329 (2021).
- [61] V. Fedorov, *Theory of Optimal Experiments, Cellular Neurobiology* (Academic Press, Cambridge, MA, USA, 1972).
- [62] B. Braiorn-Orrs, M. Weyrauch, and M. V. Rakov, Phase diagrams of one-, two-, and three-dimensional quantum spin systems derived from entanglement properties, *Quantum Inf. Comput.* **16**, 885 (2016).
- [63] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, and C. J. M. J. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature* **549**, 242 (2017).
- [64] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, The graph neural network model, *IEEE Trans. Neural Netw.* **20**, 61 (2009). <https://ieeexplore.ieee.org/document/4700287>.
- [65] A. Micheli, Neural network for graphs: A contextual constructive approach, *IEEE Trans. Neural Netw.* **20**, 498 (2009). <https://ieeexplore.ieee.org/document/4773279>.
- [66] H.-Y. Huang, R. Kueng, G. Torlai, V. V. Albert, and J. Preskill, Provably efficient machine learning for quantum many-body problems, *Science* **377**, eabk3333 (2022).
- [67] E. Onorati, C. Rouzé, D. S. Franca, and J. D. Watson, Efficient learning of ground & thermal states within phases of matter, *ArXiv:2301.12946* (2023).
- [68] L. Lewis, H.-Y. Huang, V. T. Tran, S. Lehner, R. Kueng, and J. Preskill, Improved machine learning algorithm for predicting ground state properties, *ArXiv:2301.13169* (2023).
- [69] Y. Che, C. Gneiting, and F. Nori, Exponentially improved efficient machine learning for quantum many-body states with provable guarantees, *ArXiv:2304.04353* (2023).
- [70] S. Yunoki, Numerical study of the spin-flop transition in anisotropic spin- $\frac{1}{2}$ antiferromagnets, *Phys. Rev. B* **65**, 092402 (2002).
- [71] O. Amosy, T. Danzig, E. Porat, G. Chechik, and A. Makmal, Iterative-free quantum approximate optimization algorithm using neural networks, *ArXiv:2208.09888* (2022).
- [72] N. Xie, X. Lee, D. Cai, Y. Saito, and N. Asai, Quantum approximate optimization algorithm parameter prediction using a convolutional neural network, *ArXiv:2211.09513* (2023).
- [73] N. Jain, B. Coyle, E. Kashefi, and N. Kumar, Graph neural network initialisation of quantum approximate optimisation, *Quantum* **6**, 861 (2022).
- [74] H. Wang, M. Weber, J. Izaac, and C. Y.-Y. Lin, Predicting properties of quantum systems with conditional generative models, *ArXiv:2211.16943* (2022).
- [75] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, *Nat. Phys.* **16**, 1050 (2020).