Supplementary Information for: "Quantum circuit architecture search for variational quantum algorithms"

Yuxuan Du,[1, 2] Tao Huang,[3] Shan You,[3] Min-Hsiu Hsieh,[4, 5] and Dacheng Tao[1, 2]
[1] *School of Computer Science, Faculty of Engineering,*
The University of Sydney, NSW 2008, Australia
[2] *JD Explore Academy, Beijing 101111, China*
[3] *SenseTime Research, Beijing 100080, China*
[4] *Hon Hai Quantum Computing Research Center, Taipei 114, Taiwan*
[5] *Centre for Quantum Software and Information,*
Faculty of Engineering and Information Technology,
University of Technology Sydney, NSW 2007, Australia
(Dated: April 8, 2022)

We organize the Supplementary as follows. In Supplementary I, we establish the connection between the bandit learning and the ansatz assignment task and discuss how to exploit bandit learning algorithms to further advance the ansatz assignment task. We then provide explanations and simulation results related to the fierce competition phenomenon and the classification task in Supplementary II. Afterwards, we present simulation and experiment details about the quantum chemistry tasks in Supplementary III. Subsequently, we exhibit how to introduce evolutionary algorithms into the ranking stage to boost the performance of QAS in Supplementary IV. Next, we empirically explore the trainability of QAS through the lens of barren plateaus in Supplementary V. Last, we demonstrate a variant of QAS to effectively accomplish large-scale problems in Supplementary VI.

I. The ansatz assignment task

In this section, we first connect the ansatz assignment task with the adversarial bandit learning problem. We then compare the method used in QAS with all bandit algorithms in terms of the regret measure. We last explain how to employ advanced bandit learning algorithms to reduce the runtime complexity of the ansatz assignment task.

A. The connection between the adversarial bandit learning and the ansatz assignment

Let us first introduce the adversarial bandit learning. In the adversarial bandit learning [23], a player has W possible arms to choose. Denote the total number of iterations as T. At the t-th iteration,

- The player chooses an arm $w^{(t)} \in [W]$ with a deterministic strategy or sampling from a certain distribution $\mathcal{P}_w$;
- The adversary chooses a cost $c^{(t)}(w^{(t)})$ for the chosen arm $w^{(t)}$;
- The cost of the selected arm $w^{(t)}$, i.e., $c^{(t)}(w^{(t)})$ with $w^{(t)} \in [W]$, is revealed to the player.

The goal of the adversarial bandit learning is minimizing the total cost over T iterations, where its performance is quantified by the regret r_T, i.e.,

$$r_T = \sum_{t=1}^{T} c^{(t)}(w^{(t)}) - \min_{w\in[W]} \sum_{t=1}^{T} c^{(t)}(w). \tag{1}$$

Intuitively, the regret r_T compares the cumulative cost of the selected arms $\{w^{(t)}\}_{t=1}^{T}$ with the best arm in hindsight. If $r_T = o(T)$, where the regret can be either negative or scales at most sublinearly with T, we say that the player is learning; otherwise, when $r_T = \Theta(T)$ such that the regret scales linearly with T, we say that the player is not learning, since the averaged cost per-iteration does not decrease with time.

We now utilize the language of the adversarial bandit learning to restate the ansatz assignment problem. In QAS, each arm refers to a supernet and the number of arms equals to the number of supernets. The cost $c^{(t)}(w^{(t)})$ is equivalent to the objection function $\mathcal{L}(\boldsymbol{\theta}^{(t,w)}, \boldsymbol{a}^{(t)})$, where $\boldsymbol{a}^{(t)}$ refers to the sampled ansatz $\boldsymbol{a}^{(t)} \in \mathcal{S}$, and $\boldsymbol{\theta}^{(t,w)}$ represents the trainable parameters of the w-th supernet $\mathcal{A}^{(w)}$. The aim of the ansatz assignment is to allocate

$\{\boldsymbol{a}^{(t)}\}_{t=1}^T$ to the best sequence of arms (supernets) to minimize the cumulative cost. Denote the selected sequence of arms (indices of supernets) of QAS as $\{I_w^{(t)}\}_{t=1}^T$. The regret in Eqn. (1) can be rewritten as

$$R_T = \sum_{t=1}^T \mathcal{L}(\boldsymbol{\theta}^{(t,I_w^{(t)})}, \boldsymbol{a}^{(t)}) - \min_{w\in[W]} \sum_{t=1}^T \mathcal{L}(\boldsymbol{\theta}^{(t,w)}, \boldsymbol{a}^{(t)}). \tag{2}$$

B. The comparison between the strategy used in QAS and all bandit algorithms

The following theorem shows that the strategy used in QAS outperforms all bandit algorithms in terms of the regret measure.

Theorem 1. *Let W and T be the number of supernets and iterations, respectively. Suppose that the ansatz $\boldsymbol{a}^{(t)}$ is assigned to the $I_w^{(t)}$-th supernet $\mathcal{A}^{(I_w^{(t)})}$ with $I_w^{(t)} \in [W]$ at the t-th iteration, where the corresponding objective function is $\mathcal{L}(\boldsymbol{\theta}^{(t,I_w^{(t)})}, \boldsymbol{a}^{(t)}) \in [0,1]$. Define the regret as*

$$R_T = \sum_{t=1}^T \mathcal{L}(\boldsymbol{\theta}^{(t,I_w^{(t)})}, \boldsymbol{a}^{(t)}) - \min_{w\in[W]} \sum_{t=1}^T \mathcal{L}(\boldsymbol{\theta}^{(t,w)}, \boldsymbol{a}^{(t)}), \tag{3}$$

where the randomness is over the selection of $I_w^{(t)}$. The method used in QAS to determine $\{I_w^{(t)}\}$ promises the regret $R_T \leq 0$, while the regret for the best bandit algorithms is lower bounded by $R_T = \Omega(T)$.

The proof of Theorem 1 exploits the following lemma.

Lemma 1 (Theorem 1, [24])**.** *Suppose $W \geq 2$ and $\delta \in (0, 1/4)$ and $T \geq 32(W-1)\log(2/\delta)$, then there exists a sequence of data $\{\boldsymbol{\theta}^{(t,I_w^{(t)})}, \boldsymbol{a}^{(t)}\}_{t=1}^T$, or equivalently, the objective values $\{\mathcal{L}(\boldsymbol{\theta}^{(t,I_w^{(t)})}, \boldsymbol{a}^{(t)})\}_{t=1}^T$ such that the regret in Eqn. (2) follows*

$$\mathcal{P}\left(R_T \geq \frac{1}{27}\sqrt{(W-1)T\log(1/(4\delta))}\right) \geq \delta/2. \tag{4}$$

The lower bound given in Lemma 1 indicates that under the adversarial setting, there does not exist an adversarial bandit algorithm can achieve the regret smaller than $\Omega(\sqrt{WT\log(1/\delta)})$ with probability at least $1-\delta$.

We are now ready to prove Theorem 1.

Proof of Theorem 1. Here we first prove the regret R_T in Eqn. (3) for the assignment strategy employed in QAS. We then quantity the lower bound of R_T for all adversarial bandit algorithms.

Recall the assignment strategy used in QAS. Given the sampled ansatz $\boldsymbol{a}^{(t)} \in \mathcal{S}$, QAS feeds this ansatz into W supernets and compares W values of objective functions, i.e., $\{\mathcal{L}(\boldsymbol{\theta}^{(t,w)}, \boldsymbol{a}^{(t)})\}_{w=1}^W$. Then, the ansatz $\boldsymbol{a}^{(t)}$ is assigned to the $I_w^{(t)}$-th supernet as

$$I_w^{(t)} = \arg\min_{w=1,\ldots,W} \mathcal{L}(\boldsymbol{\theta}^{(t,w)}, \boldsymbol{a}^{(t)}). \tag{5}$$

Denote the regret R_T in Eqn. (3) obtained by QAS as R_T^Q. By exploiting the explicit definition of $I_w^{(t)}$ in Eqn. (5), the regret R_T^Q yields

$$\begin{aligned} R_T^Q &= \sum_{t=1}^T \mathcal{L}(\boldsymbol{\theta}^{(t,I_w^{(t)})}, \boldsymbol{a}^{(t)}) - \min_{w\in[W]} \sum_{t=1}^T \mathcal{L}(\boldsymbol{\theta}^{(t,w)}, \boldsymbol{a}^{(t)}) \\ &= \sum_{t=1}^T \min_{w=1,\ldots,W} \mathcal{L}(\boldsymbol{\theta}^{(t,w)}, \boldsymbol{a}^{(t)}) - \min_{w\in[W]} \sum_{t=1}^T \mathcal{L}(\boldsymbol{\theta}^{(t,w)}, \boldsymbol{a}^{(t)}) \\ &\leq 0, \end{aligned} \tag{6}$$

where the last inequality employs the fact that the summation of minimum values of functions is less than the minimum value of summation of functions (i.e., $\sum_t \min_x f_t(x) \leq \min_x \sum_t f_t(x)$ and the equality is hold when the minimum of all functions $\{f_t(x)\}$ is identical).

Denote the regret R_T in Eqn. (3) obtained by a given bandit algorithm as R_T^B. Due to Lemma 1, we achieve

$$\mathcal{P}\left(R_T^B \geq \frac{1}{27}\sqrt{(W-1)T\log(1/(4\delta))}\right) \geq \delta/2. \tag{7}$$

In other words, for the ansatz assignment task, the regret for all adversarial bandit algorithms is lower bounded by $R_T^B \geq \Omega(\sqrt{WT\log(1/\delta)})$ with probability δ.

Based on Eqn. (6) and Eqn. (7), we conclude that with high probability, no bandit learning algorithm can achieve a lower regret than that of the strategy adopted in QAS. □

C. Applying bandit learning algorithms to the ansatz assignment task

Here we discuss how to apply bandit learning algorithms to improve the ansatz assignment task in terms of the runtime cost. Recall the ansatz assignment strategy used in QAS. At each iteration, the sampled ansatz should feed into W supernets separately and then compare the returned W objective values. In this way, the runtime complexity becomes expensive for a large W, as discussed in Method. The adversarial bandit learning algorithms are a promising solution to tackle the runtime issue. As explained in Supplementary I A, when adversarial bandit learning algorithms are employed, the ansatz is only required to feed into one supernet at each iteration, while the price is inducing a relatively large regret bound.

II. The synthetic dataset classification task

The outline of this section is as follows. In Supplementary II A, we first introduce some basic terminologies in machine learning to make our description self-consistent. In Supplementary II B, we explain how to construct the synthetic dataset $\mathcal{D}$. In Supplementary II C, we provide the simulation results omitted in the main text and elaborate on the fierce competition phenomenon. Last, in Supplementary II D, we compare the learning performance of the quantum classifier with the hardware-efficient ansatz and the ansatz searched by QAS under the noise model extracted from the real quantum device, i.e., an IBM's 5-qubit quantum machine nameds as 'Ibmq_lima'.

A. Basic terminologies in machine learning

When we apply QAS to accomplish the classification task, the terminology 'epoch', which is broadly used in the field of machine learning [25], is employed to replace 'iteration'. Intuitively, an epoch means that an entire dataset is passed forward through the quantum learning model. For the quantum kernel classifier used in the main text, each training example in $\mathcal{D}_{tr}$ is fed into the quantum circuit in sequence to acquire the predicted label. Since $\mathcal{D}_{tr}$ includes in total 100 examples, it will take 100 iterations to complete one epoch.

In the synthetic classification task, we split the datasets into three parts, i.e., the training, validation, and test datasets, following the convention of machine learning [25]. The training dataset $\mathcal{D}_{tr}$ is used to optimize the trainable parameters during the learning process. The function of the validation dataset $\mathcal{D}_{va}$ is estimating how well the classifier has been trained. During T epochs, the trainable parameters that achieve the highest validation accuracy are set as the output parameters. Mathematically, the output parameters satisfy

$$\hat{\boldsymbol{\theta}} = \max_{\{\boldsymbol{\theta}^{(t)}\}_{t=1}^{T}} \sum_i \mathbb{1}_{\tilde{y}^{(i)}(\boldsymbol{\theta}^{(t)}, \boldsymbol{x}^{(i)})=y^{(i)}}, \tag{8}$$

where $\{\boldsymbol{x}^{(i)}, y^{(i)}\} \in \mathcal{D}_{va}$, $\tilde{y}^{(i)}$ is the prediction of the classifier given $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{x}^{(i)}$, and $\mathbb{1}_z$ is the indicator function that takes the value 1 if the condition z is satisfied and zero otherwise. Finally, the output parameters $\hat{\boldsymbol{\theta}}$ are applied to the test dataset to benchmark the performance of the trained classifier.

B. Implementation of the synthetic dataset

Here we recap the method to construct the synthetic dataset proposed in [26]. Denote the encoding layer as

$$U_{\boldsymbol{x}} = \mathrm{R_Y}(\boldsymbol{x}_1) \otimes \mathrm{R_Y}(\boldsymbol{x}_2) \otimes \mathrm{R_Y}(\boldsymbol{x}_3). \tag{9}$$

To establish the synthetic dataset $\mathcal{D}$ used in the main text, we first generate a set of data points $\{\boldsymbol{x}^{(i)}\}$ with $\boldsymbol{x}^{(i)} \in \mathbb{R}^3$. We then define the optimal circuit as

$$U^*(\boldsymbol{\theta}^*) = \prod_{l=1}^{3} U_l^*(\boldsymbol{\theta}_l^*), \tag{10}$$

where $U_l^*(\boldsymbol{\theta}_l^*) = \otimes_{j=1}^3 \mathrm{R_Y}(\boldsymbol{\theta}_{l,j}^*)(\mathrm{CNOT} \otimes I_2)(I_2 \otimes \mathrm{CNOT})$ and the parameter $\boldsymbol{\theta}_{l,j}^*$ is uniformly sampled from $[0, 2\pi)$ for all $j \in [3]$ and $l \in [3]$. The strategy to label $\boldsymbol{x}^{(i)}$ is as follows. Let $\Pi = \mathbb{I}_4 \otimes |0\rangle\langle 0|$ be the measurement operator. The data point $\boldsymbol{x}^{(i)}$ is labeled as $y^{(i)} = 1$ if

$$\langle 000|U_{\boldsymbol{x}^{(i)}}^\dagger U^*(\boldsymbol{\theta}^*)\Pi U^*(\boldsymbol{\theta}^*) U_{\boldsymbol{x}^{(i)}}|000\rangle \geq 0.75. \tag{11}$$

The label of $\boldsymbol{x}^{(i)}$ is assigned as $y^{(i)} = 0$ if

$$\langle 000|U_{\boldsymbol{x}^{(i)}}^\dagger U^*(\boldsymbol{\theta}^*)\Pi U^*(\boldsymbol{\theta}^*) U_{\boldsymbol{x}^{(i)}}|000\rangle \leq 0.25. \tag{12}$$

Note that, if the measured result is in the range $(0.25, 0.75)$, we drop this data point and sample a new one. By repeating the above procedure, we can built the synthetic dataset $\mathcal{D}$.

C. Simulation results of the synthetic dataset classification and the fierce competition phenomenon

Here we first introduce how to use the quantum kernel classifier to conduct the prediction. Given the data point $\boldsymbol{x}^{(i)} \in \mathcal{D}$ at the t-th epoch, the quantum kernel classifier is composed of two unitraies, i.e., $U_{\boldsymbol{x}^{(i)}}$ and $U(\boldsymbol{\theta}^{(t)})$, where the sequence of quantum gates in $U(\boldsymbol{\theta}^{(t)})$ is fixed as shown in Figure 2 (b). The output of quantum kernel classifier yields

$$\tilde{y}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}) = \langle 000|U_{\boldsymbol{x}^{(i)}}^\dagger U(\boldsymbol{\theta}^{(t)})^\dagger \Pi U(\boldsymbol{\theta}^{(t)}) U_{\boldsymbol{x}^{(i)}}|000\rangle. \tag{13}$$

The predicted label of $\boldsymbol{x}^{(i)}$, i.e., $g(\tilde{y}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}))$, becomes

$$g(\tilde{y}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)})) = \begin{cases} 0, & \text{if } \tilde{y}(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}) < 0.5 \\ 1, & \text{otherwise} \end{cases}. \tag{14}$$

When QAS is employed to enhance the trainability and to mitigate error of the quantum kernel classifier, the arrangement of quantum gates in $U(\boldsymbol{\theta})$ is no longer fixed and depends on the sampled ansatz. In other words, at the t-th epoch, given the data point $\boldsymbol{x}^{(i)} \in \mathcal{D}$, the measured result $\tilde{y}(\mathcal{A}, \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)})$ is

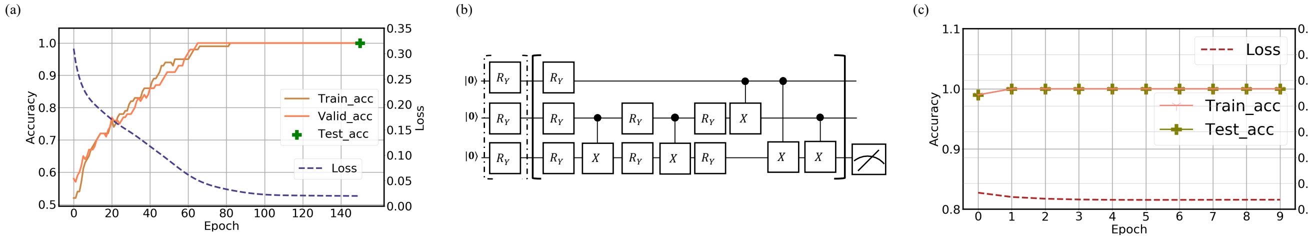
$$\tilde{y}(\mathcal{A}, \boldsymbol{x}^{(i)}, \boldsymbol{\theta}^{(t)}) = \langle 000|U_{\boldsymbol{x}^{(i)}}^\dagger U(\boldsymbol{\theta}^{(t)}, \boldsymbol{a})^\dagger \Pi U(\boldsymbol{\theta}^{(t)}, \boldsymbol{a}) U_{\boldsymbol{x}^{(i)}}|000\rangle, \tag{15}$$

where $U(\boldsymbol{\theta}^{(t)}, \boldsymbol{a})$ denotes that the trainable unitary amounts to the ansatz $\boldsymbol{a}$ and the corresponding trainable parameters $\boldsymbol{\theta}^{(t)}$ are controlled by the supernet $\mathcal{A}$.

We then provide the simulation results of the conventional quantum kernel classifier and QAS towards the synthetic dataset $\mathcal{D}$ under the noiseless setting. As exhibited in Supplementary Figure 1 (a), both the training and validation accuracies of the conventional quantum kernel classifier fast converge to 100% after 80 epochs. The test accuracy also reaches 100%, highlighted by the green marker. Meanwhile, the loss $\mathcal{L}$ decreases to 0.24. These results indicate that the conventional quantum kernel classifier with the protocol as depicted in Figure 2 (b) can well learn the synthetic dataset $\mathcal{D}$.

The hyper-parameters of QAS under the noiseless setting are identical to the noisy setting introduced in the main text. Specifically, we set $T = 400$ and $W = 1$ in the training stage (Step 2), $K = 500$ in the ranking stage (Step 3), and $T = 10$ in the retraining stage (Step 4). Supplementary Figure 1 (b) demonstrates the output ansatz in Step 3. Compared to the conventional quantum kernel classifier, the output ansatz includes fewer CNOT gates, which is more amiable for physical implementations. Supplementary Figure 1 (c) illustrates the learning performance of the output ansatz in the retraining stage. Concretely, both the training and test accuracies converge to 100% after one epoch. These results indicate that QAS can well learn the synthetic dataset $\mathcal{D}$ under the noiseless setting. Note that for all simulation results related to classification tasks, the Adam optimizer [25] is exploited to update the training parameters of the quantum kernel classifier and QAS. The learning rate is set as 0.05.

We end this subsection by explaining the fierce competition phenomenon encountered in the optimization of QAS. Namely, when the number of supernets is 1, some ansatze that can achieve high classification accuracies with



Supplementary Figure 1: **Simulation results for the synthetic data classification.** (a) The performance of the conventional quantum classifier under the noiseless setting. The label 'Train_acc', 'Valid_acc', and 'Test_acc' refers to the training, validation, and test accuracy, respectively. (b) The simulation results of QAS in the retraining stage under the noiseless setting. (c) The ansatz (i.e., the quantum circuit architecture) generated by QAS in Step 3.

independently training, will perform poorly in QAS. To exhibit that QAS indeed searches a set of ansatze (quantum circuit architectures) with high classification accuracies, we examine the correlation of the performance of the ansatz with independently optimization and training by QAS. In particular, we randomly sample 500 ansatze from all possible architectures and evaluate the widely-used Spearman and Kendall tau rank correlation coefficients [27, 28], which are in the range of $[0,1]$. In particular, larger correlation coefficients (or equivalently, stronger correlations) indicate that the ranking distribution achieved by QAS is consistent with the performance of different circuit architectures with independently training. Moreover, larger correlation coefficients also imply that the output ansatz of QAS can well estimate the target ansatz $\boldsymbol{a}^*$.

The Spearman rank correlation coefficient ρ_S quantifies the monotonic relationships between random variables r and s. Specifically, the spearman rank correlation coefficient between r and s is defined as

$$\rho_S = \frac{cov(r,s)}{\sigma_r \sigma_s}, \tag{16}$$

where $cov(\cdot,\cdot)$ is the covariance of two variables, and σ_r (σ_s) refers to the standard deviations of r (s). Suppose that $\boldsymbol{r}\in\mathbb{R}^n$ and $\boldsymbol{s}\in\mathbb{R}^n$ are two observation vectors of r and s, respectively, the explicit form ρ_S is

$$\rho_S = 1 - \frac{6\sum_{i=1}^{n}(\boldsymbol{r}_i - \boldsymbol{s}_i)^2}{n(n^2-1)}. \tag{17}$$

When the Spearman rank correlation is employed in QAS, the observation vector $\boldsymbol{r}$ $(\boldsymbol{s})$ corresponds to the achieved validation accuracy of the sampled 500 ansatze in the ranking stage, while the observation vector $\boldsymbol{s}$ corresponds to the achieved validation accuracy of the sampled 500 ansatze with independently training.

The Kendall tau rank correlation coefficient concerns the relative difference of concordant pairs and discordant pairs. Specifically, in QAS, denote $\boldsymbol{r}$ $(\boldsymbol{s})$ as the observation vector that refers to the achieved validation accuracy of the sampled 500 ansatze in the ranking stage (with independently training). Given any pair $(\boldsymbol{r}_i,\boldsymbol{r}_j)$ and $(\boldsymbol{s}_i,\boldsymbol{s}_j)$, it is said to be concordant if $(\boldsymbol{r}_i > \boldsymbol{r}_j)\wedge(\boldsymbol{s}_i > \boldsymbol{s}_j)$ or $(\boldsymbol{r}_i < \boldsymbol{r}_j)\wedge(\boldsymbol{s}_i < \boldsymbol{s}_j)$; otherwise, it is disconcordant. According to the above definition, the explicit form of the Kendall tau rank correlation coefficient is

$$\rho_K = \frac{2}{n(n-1)}\sum_{i<j}\operatorname{sign}(\boldsymbol{r}_i - \boldsymbol{r}_j)\operatorname{sign}(\boldsymbol{s}_i - \boldsymbol{s}_j), \tag{18}$$

where $\operatorname{sign}(\cdot)$ represents the sign function.

Supplementary Table I summarizes the correlation coefficients with $n = 500$. Specifically, when the number of supernets is 1, we have $\rho_K = 0.113$, which implies that the correlation between $\boldsymbol{r}$ and $\boldsymbol{s}$ is very low. By contrast, with increasing the number of supernets to 5 and 10, the correlation coefficients ρ_S and ρ_K are dramatically enhanced, which are 0.723 and 0.536, respectively. Moreover, when the number of supernets is $W = 10$ and the number of iterations is increased to $T = 1000$, the correlation coefficients ρ_S and ρ_K can be further improved, which are 0.774 and 0.591, respectively. These results indicate that the competition phenomenon in QAS can be alleviated by introducing more supernets and increasing the number of training iterations. In doing so, the performance of ansatze evaluated by QAS can well accord with their real performance with independently training.

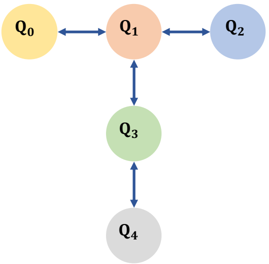
D. The performance of QAS towards the noise model extracted from the real quantum devices

We evaluate the classification accuracy of the quantum classifier equipped with the hardware-efficient ansatz and the ansatz searched by QAS under the noise model extracted from a real quantum device, i.e., an IBM's 5-qubit quantum

	$W = 1$ & $T = 500$	$W = 5$ & $T = 500$	$W = 10$ & $T = 500$	$W = 10$ & $T = 1000$
ρ_S	0.488	0.723	0.716	0.774
ρ_K	0.113	0.536	0.531	0.591

Supplementary Table I: **The correlation coefficients.** The label '$W = a$ & $T = b$' represents that the number of supernets and training iterations is a and b, respectively.

machine named 'Ibmq_lima'. The qubit connectivity of the deployed quantum machine is illustrated in Supplementary Figure 2 and its system parameters are summarized in Supplementary Figure 3.



Supplementary Figure 2: **The qubits connectivity of 'Ibmq_lima'.**

Qubit	T1(μs)	T2(μs)	Readout error	Single-qubit U2 error gate	CNOT error rate
Q0	94.36	170.91	1.90E-2	1.87E-4	cx0-1: 6.76E-3
Q1	108.32	140.36	2.38E-2	3.01E-4	cx1-0: 6.76E-3 cx1-2: 5.53E-3 cx1-3: 1.49E-2
Q2	146.07	154.00	1.75E-2	1.92E-4	cx2-1: 5.53E-3
Q3	106.5	89.98	3.44E-2	3.44E-4	cx3-1: 1.49E-2 cx3-4: 1.63E-2
Q4	22.34	19.02	5.48E-2	6.77E-4	cx4-3: 1.63E-2

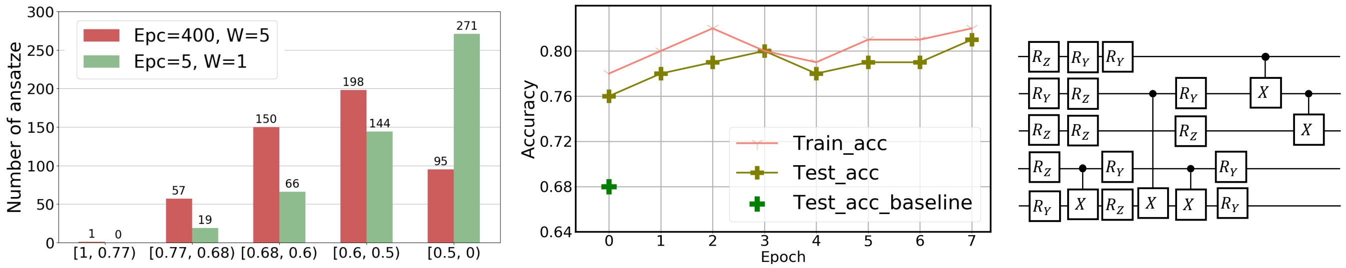
Supplementary Figure 3: **Performance of qubits for Ibmq_lima.** T1 and T2 refer to the energy relaxation time and dephasing time, respectively. U2 and CNOT gates error obtained via performing randomized benchmarking. The label 'cxa-b' represents that the CNOT gate is applied to the qubits a and b.

The implementation details are as follows. The construction of the synthetic dataset is identical to those introduced in Supplementary II B, except for setting the feature dimension as 5 instead of 3. For the quantum classifier with the hardware-efficient ansatz, the number of layers and the number of epochs are set as $L = 3$ and $T = 400$, respectively. The hardware-efficient ansatz used in the baseline experiment takes the form $U(\boldsymbol{\theta}) = \prod_{l=1}^{3} U_l(\boldsymbol{\theta}_l)$, where $U_l(\boldsymbol{\theta}_l) = \otimes_{j=1}^{5} \mathrm{R_Y}(\boldsymbol{\theta}_{l,j})(\mathrm{CNOT} \otimes \mathrm{CNOT} \otimes I_2)(I_2 \otimes \mathrm{CNOT}) \otimes \mathrm{CNOT})$. For QAS, the choices of single-qubit gates and two-qubit gates are $\{\mathrm{R_Y}, \mathrm{R_Z}\}$ and $\{\mathrm{CNOT}, \mathbb{I}_4\}$, respectively. The CNOT gates can conditionally interact with four qubit-pairs, i.e., {0-1, 1-2, 1-3, 3-4}, following the topology of Ibmq_lima. We apply two settings to evaluate the learning performance of QAS. In the first setting, we set the number of supernets as $W = 1$ and the number of epochs in the optimization stage as $T = 10$. In the second setting, we set the number of supernets as $W = 5$ and the number of epochs as $T = 400$. For both settings, the number of layers is set as $L = 3$ and the number of the sampled ansatze at the ranking stage is $K = 500$.

The simulation results are exhibited in Supplementary Figure 4. For the quantum classifier with the hardware-efficient ansatz, the achieved test accuracy is 68%. We utilize this test accuracy as the baseline to quantify the learning performance of QAS. As shown in the left panel of Supplementary Figure 4, for the first setting (i.e., $T = 5$ and $W = 1$), there are in total 19 ansatze out of $K = 500$ ansatze achieving a higher accuracy beyond the baseline. When we increase the number of epochs and the number of supernets to $T = 400$ and $W = 5$ (i.e., corresponding to the second setting), there are in total 58 ansatze out of $K = 500$ ansatze surpassing the baseline. Meanwhile, the average performance over the sampled $K = 500$ ansatze is better than the first setting. As shown in the middle panel of Supplementary Figure 4, when we retrain the searched ansatz in the second setting (depicted in the right panel of Supplementary Figure 4) with 8 epochs, the test accuracy improves to 81%. These observations validate the effectiveness of QAS of enhancing the learning performance of VQAs towards classification tasks. Moreover, increasing the number of supernets W and the number of epochs T contributes to improve the capability of QAS.

III. Experimental Details of the ground state energy estimation

In this section, we first briefly recap the ground state energy estimation task in Supplementary III A. In Supplementary III B, we compare the performance of QAS and conventional VQE towards the ground state energy estimation task when they are implemented on real quantum hardware.



Supplementary Figure 4: **Simulation results of the quantum classifiers under the noise model extracted from Ibmq_lima**. The left panel illustrates the performance of QAS with two different settings after the optimization process. The label 'Epc=a, W =b' represents that the number of epochs and supernets is $T = a$ and $W = b$, respectively. The x-axis means that the validation accuracy of the sampled ansatz is in the range of $[c, d)$. The middle panel exhibits the performance of the quantum classifiers with the hardware-efficient ansatz (labeled by 'Test_acc_baseline') and the ansatz searched by QAS (labeled by 'Train/Test_acc') at the fine tuning stage under the noisy setting. The right panel depicts the searched ansatz before quantum compiling.

A. The ground state energy estimation

A central application of VQAs is solving the electronic structure problem, i.e., finding the ground state energies of chemical systems described by Hamiltonians. Note that chemical Hamiltonians in the second quantized basis set approach can always be mapped to a linear combination of products of local Pauli operators [29]. In particular, the explicit form of the molecular hydrogen Hamiltonian H_h is

$$\begin{aligned} H_h = \ & -0.042 + 0.178(Z_0 + Z_1) - 0.243(Z_2 + Z_3) + 0.171 Z_0 Z_1 + 0.123(Z_0 Z_2 + Z_1 Z_3) + 0.168(Z_0 Z_3 + Z_1 Z_2) \\ & +0.176 Z_2 Z_3 + 0.045(Y_0 X_1 X_2 Y_3 - Y_0 Y_1 X_2 X_3 - X_0 X_1 Y_2 Y_3 + X_0 Y_1 Y_2 X_3). \end{aligned} \tag{19}$$

The goal of the variational Eigen-solver (VQE) is generating a parameterized wave-function $|\Psi(\boldsymbol{\theta})\rangle$ to achieve

$$\min_{\boldsymbol{\theta}} |\langle \Psi(\boldsymbol{\theta})|H_h|\Psi(\boldsymbol{\theta})\rangle - E_m|. \tag{20}$$

The linear property of H_h in Eqn. (19) implies that the value $|\langle \Psi(\boldsymbol{\theta})|H_h|\Psi(\boldsymbol{\theta})\rangle$ can be obtained by iteratively measuring $|\Psi(\boldsymbol{\theta})\rangle$ using Pauli operators in H_h, e.g., such as $|\langle \Psi(\boldsymbol{\theta})|\mathbb{I}_8 \otimes Z_0|\Psi(\boldsymbol{\theta})\rangle$ and $|\langle \Psi(\boldsymbol{\theta})|X_0 Y_1 Y_2 X_3|\Psi(\boldsymbol{\theta})\rangle$. The lowest energy of H_h equals to $E_m = -1.136\,\text{Ha}$, where 'Ha' is the abbreviation of Hartree, i.e., a unit of energy used in molecular orbital calculations with $1\,\text{Ha} = 627.5\text{kcal/mol}$. The exact value of E_m is acquired from a full configuration-interaction calculation [29].

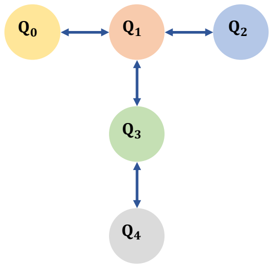
We note that the quantum natural gradient optimizer [30], which can accelerate the convergence rate, is employed to optimize the trainable parameters for both VQE and QAS, where the learning rate is set as 0.2.

B. The performance of QAS on real quantum devices

Here we carry out QAS and the conventional VQE on IBM's 5-qubit quantum machine, i.e., 'Ibmq_ourense', to accomplish the ground state energy estimation of H_h. The qubit connectivity of 'Ibmq_ourense' is illustrated in Supplementary Figure 5, and the system parameters of these five qubits are summarized in Supplementary Figure 6.

The implementation detail is as follows. The hyper-parameters of QAS are $L = 3$, $W = 10$, $K = 500$, and $T = 500$. To examine the compatibility of QAS, we restrict its searching spaces to be consistent with the qubit connectivity of 'IBM_ourense', i.e., the single-qubit gates are sampled from $\mathrm{R_Y}$ and $\mathrm{R_Z}$, and CNOT gates can conditionally apply to the qubits pair $(0, 1)$, $(1, 0)$, $(1, 2)$, $(2, 1)$, $(1, 3)$, and $(3, 1)$, based on Supplementary Figure 6. We call this setting as QAS with the real connectivity (QAS-RC). Under such a setting, the number of all possible circuit architectures for QAS-RC is 1024^3. The hyper-parameters setting for VQE are $L = 3$ and $T = 500$. The heuristic circuit architecture used in VQE is identical to the case introduced in the main text (Figure 3 (a)). In the training process, we optimize VQE and QAS on classical computers under a noisy environment provided by the Qiskit package, which can approximately simulate the quantum gates error and readout error in 'Ibmq_ourense'. The reason that we move the training stage on the classical numerical simulators is because training VQE and QAS on 'Ibmq_ourense' will take an unaffordable runtime, due to the fair share run mode [31].

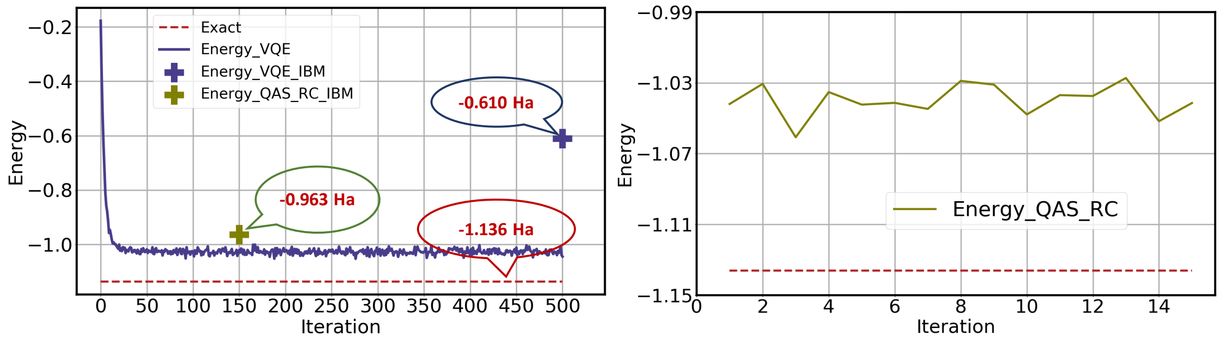
The training performance of VQE and QAS-RC is demonstrated in Supplementary Figure 7. In particular, as shown in the left panel, the estimated ground energy by VQE is around $-1.02\,\text{Ha}$ after 30 iterations, highlighted by the



Supplementary Figure 5: **The qubits connectivity of 'Ibmq_ourense'.**

Qubit	T1(μs)	T2(μs)	Readout error	Single-qubit U2 error gate	CNOT error rate
Q0	75.75	50.81	1.65E-2	5.22E-4	cx0-1: 9.55E-3
Q1	78.47	27.56	2.38E-2	4.14E-4	cx1-0: 9.55E-3 cx1-2: 9.44E-3 cx1-3: 1.25E-2
Q2	101.51	107.00	1.57E-2	1.83E-4	cx2-1: 9.44E-3
Q3	79.54	78.38	3.95E-2	4.30E-4	cx3-1: 1.25E-2 cx3-4: 8.34E-3
Q4	74.27	30.00	4.74E-2	4.20E-4	cx4-3: 8.34E-3

Supplementary Figure 6: **Performance of qubits for Ibmq_ourense.** T1 and T2 refer to the energy relaxation time and dephasing time, respectively. U2 and CNOT gates error obtained via performing randomized benchmarking. The label 'cxa-b' represents that the CNOT gate is applied to the qubits a and b.



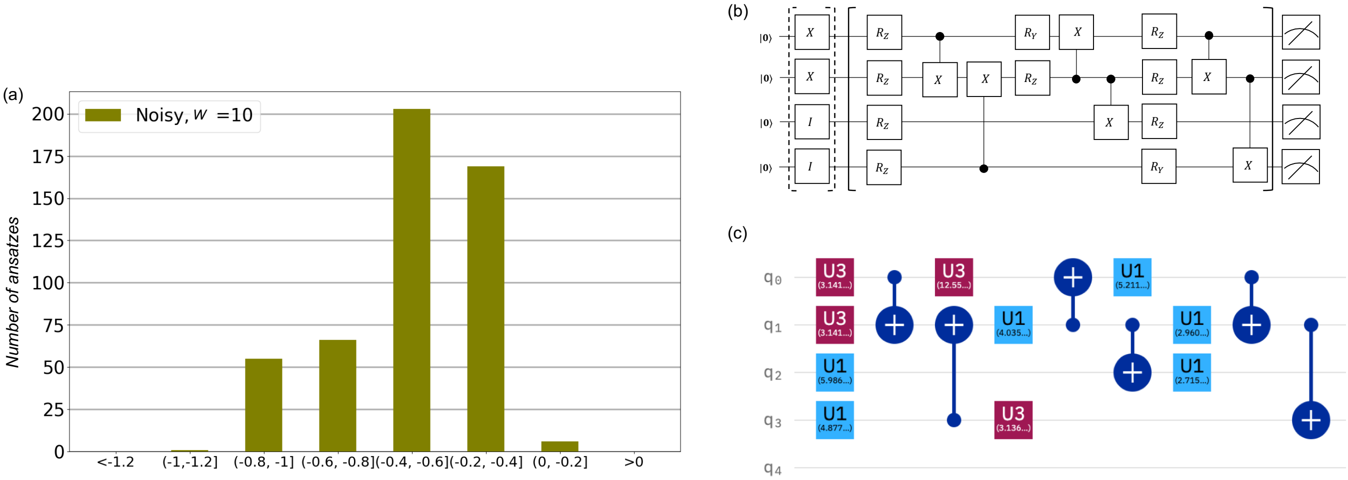
Supplementary Figure 7: **Experiment results of the ground state energy estimation of VQE and QAS.** In the left panel, the labels 'Exact' and 'Energy_VQE' correspond to the exact ground state energy and the estimated energy of VQE obtained in the training process (achieved by numerical simulators), respectively. In the left panel, the label 'Energy_QAS_RC' refer to the estimated energy of QAS-RC in the retraining phase. The estimated energy of VQE and QAS-RC when they are implemented on the real quantum processor 'Ibmq_ourense' are indicated by two cross markers, where the corresponding labels are 'Energy_VQE_IBM' and 'Energy_QAS_RC_IBM', respectively.

dark blue line. The performance of QAS-RC is shown in the right panel. Concretely, when we retrain the output ansatz with 15 iterations, its estimated energy slightly oscillates around -1.04 Ha, highlighted by the green solid line. When we implement the optimized VQE and the optimized output ansatz of QAS-RC on the real quantum device, i.e., 'Ibmq_ourense', their performances are varied. Specifically, as demonstrated in the left panel of Supplementary Figure 7, the estimated ground energy by VQE is -0.61 Ha (highlighted by the blue marker), while the estimated ground energy by QAS-RC is -0.963 Ha (highlighted by the green marker). Compared with VQE, the estimated result of QAS-RC is much closer to the exact result. We utilize the following formula to quantify the relative deviation between the simulation and experiment results. Denoted the estimated energy obtained by the numerical simulation as E_s and the test energy achieved by 'Ibmq_ourense' as E_t, the relative deviation follows

$$err = \frac{|E_s - E_t|}{E_m}, \tag{21}$$

where $E_m = -1.136$ is the exact result. Following this formula, the relative deviation for VQE and QAS-RC is 36.1% and 6.8%, respectively. Compared with the heuristic circuit architecture used in VQE, QAS that concerns the real qubits connectivities can dramatically reduce the relative deviation. The above results not only indicate the compatibility of QAS, but also demonstrate that QAS can well adapt to the weighted gates noise and achieve a high performance towards quantum chemistry tasks.

Finally, we compare the output ansatz of QAS-RC with the heuristic circuit architecture used in VQE. The simulation results of QAS-RC in the ranking stage are summarized in Supplementary Figure 8. In particular, the left panel (a) exhibits the ranking distributions of QAS-RC, where the estimated ground energy of most ansatze concentrates on $[-0.6\,\text{Ha}, -0.4\,\text{Ha}]$. Supplementary Figure 8 (b) shows the output ansatz of QAS-RC, where the corresponding circuit implementation on 'IBM_ourense' is exhibited in Supplementary Figure 8 (c). Compared with the heuristic circuit architecture used in VQE (Figure 3 (a) in the main text), the output ansatz of QAS-RC contains fewer CNOT gates.



Supplementary Figure 8: **The simulation results of QAS-RC in the ranking stage.** The panel (a) demonstrates the ansatze ranking distributions with $K = 500$ of QAS-RC. The x-axis refers to the estimated energy of the given ansatz is in the range of $(a\,\text{Ha}, b\,\text{Ha}]$ with $a, b \in \mathbb{R}$. For most ansatze, their estimated energies are above $-0.2\,\text{Ha}$. The middle panel (b) exhibits the output ansatz of QAS-RC. The right panel (c) shows the implementation of the output ansatz of QAS-RC on 'Ibmq_ourense'.

This implies that QAS-RC has the ability to appropriately reduce the number of two-qubit gates to avoid introducing too much error, while the expressive power of the trainable circuit $U(\boldsymbol{\theta})$ can be well preserved. In other words, QAS can adapt to the weighted gate noise to seek the best circuit architecture.

IV. Improving the ranking stage of QAS

Recall the ranking stage of QAS, i.e., Step 3 of Figure 1, is uniformly sampling K ansatze from the supernet $\mathcal{A}$. The aim of this step is sampling the one, among the sampled ansatze, with the best performance. However, the uniformly sampling method implies that the sampled ansatze maybe come from $\mathcal{S}_{\text{bad}}$ with a high probability when $|\mathcal{S}_{\text{bad}}| > |\mathcal{S}_{\text{good}}|$. It is highly desired to devise more effective sampling methods.

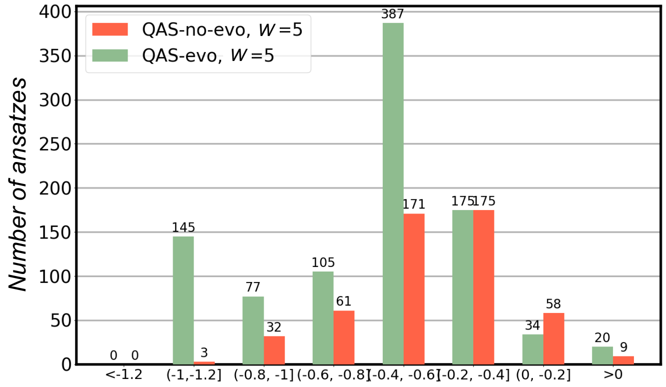
Here we utilize an evolutionary algorithm, i..e, nondominated sorting genetic algorithm II (NSGA-II) [32], to facilitate the ansatze ranking problem. The intuition behind employing NSGA-II is actively searching potential ansatze with good performance instead of uniformly sampling ansatze from all possible circuit architectures. Note that several recent studies, e.g., Refs [33, 34], have directly utilized the evolutionary and multi-objective genetic algorithms to complete ansatz design.

We apply QAS with the evolutionary algorithm to tackle the ground state energy estimation problem described in the main text. Note that all hyper-parameters settings are identical to the uniformly sampling case, except for the settings related to the evolutionary algorithm. Particularly, we set the population size as $N_{pop} = 50$ and the number of generations as $G_T = 20$. The simulation results under the noiseless setting are shown in Supplementary Figure 9. In particular, QAS assisted by NSGA-II searches in total 943 ansatze, and the estimated energy of 143 ansatze (15.2%) lies in the range from $-1\,\text{Ha}$ to $-1.2\,\text{Ha}$. By contrast, QAS with uniformly sampling strategy only finds 3 ansatze among in total 500 ansatze (0.6%) in the same range. This result empirically confirms that evolutionary algorithms can advance the performance of QAS.

We remark that other advanced machine learning techniques such as reinforcement learning [35] can also be exploited to benefit the performance of QAS.

V. An empirical exploration for the trainability of QAS

Here we empirically investigate the trainability of QAS through the lens of barren plateaus [36–39]. Recall the main conclusion of the barren plateaus is that the gradient vanishes exponentially in the qubits count N. Mathematically, the expectation of the gradient norm of the objective function tends to be zero and the corresponding variance will fast converge to zero with respect to N, i.e., $\text{Var}_{\boldsymbol{\theta}}(\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\|) \sim O(e^{-LN})$. With this regard, barren plateaus can be



Supplementary Figure 9: **Simulation result of QAS assisted by NSGA-II.** The label 'QAS-no-evo, W=5' and 'QAS-evo, W=5' refer to the QAS introduced in the main text and QAS assisted by evolutionary algorithm with the number of supernets being $W = 5$, respectively. The x-axis refers that the estimated energy of the given ansatz is in the range of $(a\,\mathrm{Ha}, b\,\mathrm{Ha}]$ with $a, b \in \mathbb{R}$.

utilized as a measure to quantify the trainability of quantum algorithms. That is, when an algorithm experiences a less impact of barren plateaus, it could possess a better trainability.

Following the above explanations, we conduct the following numerical simulations to demonstrate that the alleviation of barren plateaus in QAS. In particular, we compare the variance of the gradient norm, i.e., $\mathrm{Var}_{\boldsymbol{\theta}}(\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\|)$, with respect to the hardware-efficient ansatz and the ansatze pool implied by QAS. The mathematical expression of the objective function is

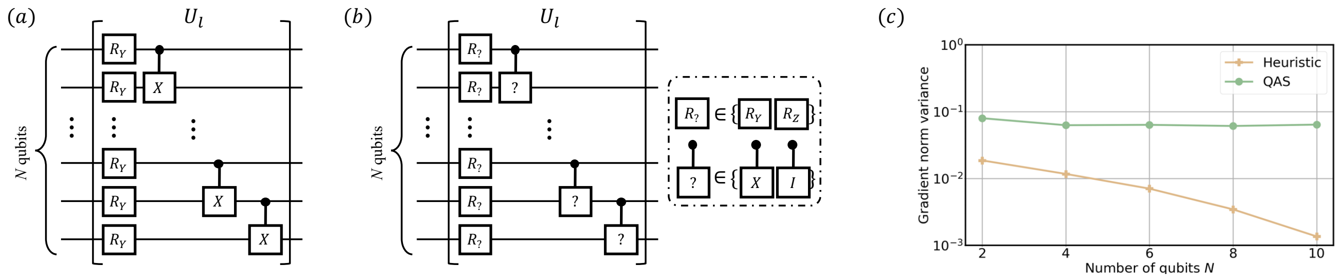
$$\mathcal{L} = \mathrm{Tr}(HU(\boldsymbol{\theta})\rho U(\boldsymbol{\theta})^{\dagger}) \ , \tag{22}$$

where the observable H equals to $\mathbb{I}_{2^{N-1}} \otimes |0\rangle\langle 0|$, the input state is $\rho = (|0\rangle\langle 0|)^{\otimes N}$, and $U(\boldsymbol{\theta})$ corresponds to the hardware-efficient ansatz or the ansatz explored in QAS. For the hardware-efficient ansatz, we set the layer number as $L = 3$, i.e., $U(\boldsymbol{\theta}) = \prod_{l=1}^{L} U_l(\boldsymbol{\theta})$ and the implementation of $U_l(\boldsymbol{\theta})$ is shown in Supplementary Figure 10 (a). The calculation of $\mathrm{Var}_{\boldsymbol{\theta}}(\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\|)$ is completed by randomly sampling $\boldsymbol{\theta}$ from a uniform distribution with 2000 times. For QAS, the ansatze pool $\mathcal{S}$ is constructed by tailoring the hardware-efficient ansatz $U(\boldsymbol{\theta})$ introduced above. As shown in Supplementary Figure 10 (b), for each U_l with $l \in [L]$, there are two choices of the single-qubit gates (i.e., $\mathrm{R_Y}$ and $\mathrm{R_Z}$) and two choices of the two-qubit gates (i.e., CNOT and an identity operation). The calculation of $\mathrm{Var}_{\boldsymbol{\theta}}(\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})\|)$ is completed by sampling 2000 different ansatzes and sampling one random $\boldsymbol{\theta}$ from a uniform distribution for each ansatz. The number of qubits N ranges from 2 to 10.

The simulation results under the noiseless setting are shown in Supplementary Figure 10 (c). For the hardware-efficient ansatz, the variance of the gradient norm is continuously decreased with respect to the increased N. This result can be treated as an evidence of barren plateaus. By contrast, for the ansatze pool explored by QAS, the variance of the gradient norm for $N = 4, 6, 8, 10$ is almost the same with each other. Meanwhile, for the same N, the variance of the gradient norm corresponding to the ansatze pool explored by QAS is always higher than that of the hardware-efficient ansatz. Recall that Ref. [36] states that the variance of gradients is continuously deceased with respect to the increased N and L, which induces the barren plateau phenomena for the sufficiently large N and L. Nevertheless, according to the simulation results in Supplementary Figure 10 (c), QAS does not obey such a tendency. These observations imply the potential of QAS to alleviate the influence of the barren plateaus.

VI. Progressive QAS for solving large-scale problems

The proposed QAS introduced in the main text is only a prototype towards automatically seeking a good ansatz instead of the handcraft design. Namely, even though QAS utilizes weight sharing strategy to reduce the parameter space to $O(dLQ^N)$, there still exists an exponential dependence with N. This exponential relation implies that in certain cases, the searched ansatz by QAS may not well estimate the optimal ansatz $\boldsymbol{a}^*$ when N becomes large within a reasonable runtime complexity. In this section, we devise a variant of QAS, termed as progressive QAS (Pro-QAS), to dramatically improve the learning performance of QAS for large-scale problems.



Supplementary Figure 10: **Numerical simulations related to barren plateaus**. (a) The circuit implementation of the hardware-efficient ansatz. (b) The circuit implementation of the ansatze pool used in QAS. (c) The variance of the gradient norm versus the number of qubits N. The label ‘heuristic’ and ‘QAS’ refers to the hardware-efficient ansatz and the ansatze pool used in QAS, respectively.

A. Algorithmic implementation of Pro-QAS

The key concept behind Pro-QAS is narrowing the size of ansatze pool to ensure its performance. Different from QAS that directly samples an ansatz from $\mathcal{S}$ to conduct optimization, Pro-QAS seeks the targeted ansatz in a progressive way. Namely, given a hardware-efficient ansatz $U(\boldsymbol{\theta})$, Pro-QAS first freezes the gate arrangement of $U_l(\boldsymbol{\theta})$ with $l \neq 1$ and search the best gate arrangement of $U_{l=1}(\boldsymbol{\theta})$. Such a searching process is the same with Step 2 (optimization) and Step 3 (Ranking K ansatze) in the original QAS. Once the search is completed, Pro-QAS begins to optimize the gate arrangement at the second layer $U_{l=2}(\boldsymbol{\theta})$ and freezes the rest $L-1$ layers. After progressively optimizing the gate arrangement of L layers, the established ansatz $\boldsymbol{a}^{(T)}$ and its corresponding parameters $\boldsymbol{\theta}^{(T)}$ are used to approximate the optimal result $(\boldsymbol{a}^*, \boldsymbol{\theta}^*)$. Notably, similar ideas of progressively constructing and optimizing ansatz have been exploited in Refs. [40–43]. Note that Ref. [44] observed that an abrupt transition phenomenon for the progressive strategy. That is, when the cost function has the identity extrema and the number of layers is less than a critical value, the layer-wise training strategy could lead to an unfavorable performance. These results can be employed as guidance to improve the learning performance of Pro-QAS. For instance, the cost function adopted in Pro-QAS should be carefully designed to avoid the identity extrema.

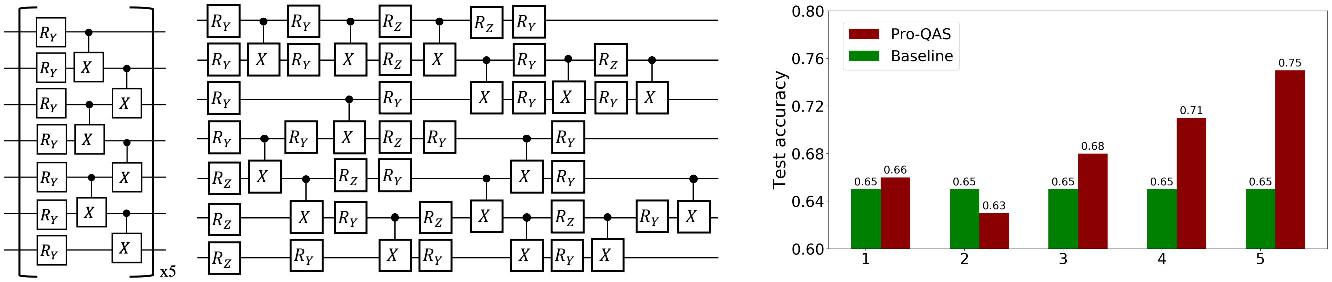
We then analyze the required runtime complexity and the reduced search space of Pro-QAS. Compared with the original QAS, the only difference of Pro-QAS is involving an extra outer loop to progressively optimize L layers. Hence, in conjunction with the runtime complexity cost of QAS derived in Method, we conclude that the execution of Pro-QAS takes at most $O(dQNL)$ memory and $O(QNL^2)$ runtime. As for the size of search space, the progressively searching strategy decreases the size of the ansatze pool to $O(LQ^N)$, which is exponentially less than that of QAS in terms of L. Remarkably, such space can be further reduced when we progressively search the gate arrangement of each layer along the index of qubits. In this way, the search space of possible ansatze scales with $O(QNL)$, while the price to pay is linearly increasing the runtime cost with respect to N.

B. Numerical simulation results of Pro-QAS

We conduct numerical simulations to demonstrate the capability of the proposed Pro-QAS towards large-scale problems. In particular, we apply Pro-QAS to achieve a binary classification task. The construction rule of the dataset $\mathcal{D} = \{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i=1}^{300}$ mainly follows Supplementary II B, where the only difference is enhancing the feature dimension of the input example from 3 to 7 and 10, respectively. In other words, the number of qubits to load the input example $\boldsymbol{x}^{(i)}$ is $N = 7$ (or $N = 10$), which is remarkably larger than the classification task discussed in the main text with $N = 3$.

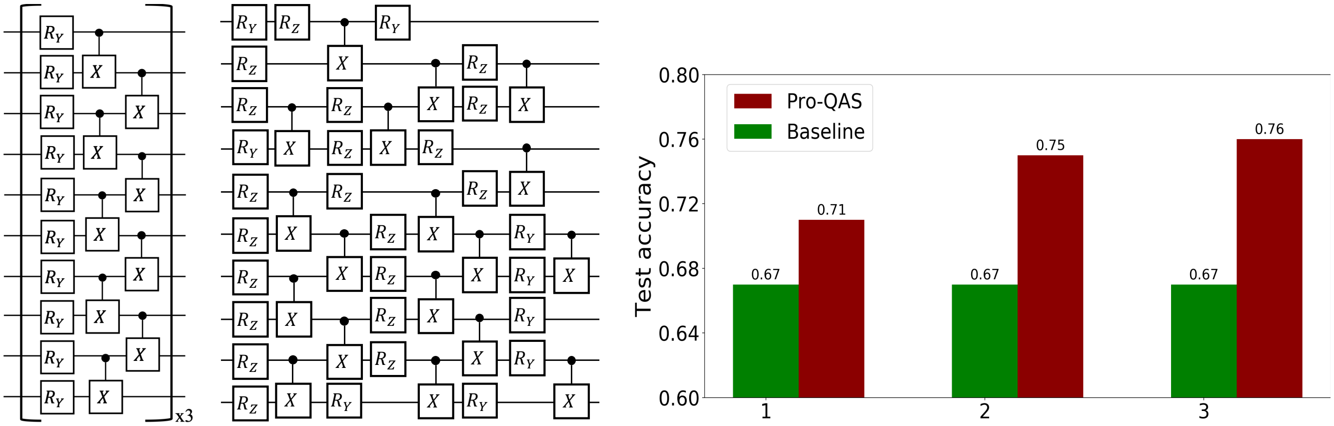
The hyper-parameters setting are as follows. For the case of $N = 7$, the number of supernet is set as $W = 1$ and the layer number is $L = 5$. The number of epochs in Step 2 is set as $T = 200$. The allowed types of quantum gates are $\{\mathrm{R_Y}, \mathrm{R_Z}, \mathrm{CNOT}\}$ with $Q = 3$ and the qubits connectivity follows the chain structure. This setting implies that the total number of ansatze without any operation is $|\mathcal{S}| = 2^{40}$. The depolarization channel is employed to simulate the quantum system noise. The depolarization rates for the single-qubit and two-qubit gates are set as $p = 0.05$ and $p = 0.1$, respectively. The number of sampled ansatze at the ranking stage is $K = 128$. For the case of $N = 10$, all settings are the same with the above one, except for setting $L = 3$. To this end, the total number of ansatze is $|\mathcal{S}| = 2^{33}$.

The simulation results for the case of $N = 7$ are exhibited in Supplementary Figure 11. As a reference, we employ the hardware-efficient ansatz with the identical layer number $L = 5$, as shown in the left subplot, to learn the same



Supplementary Figure 11: **Simulation result for the 7-qubit case.** The left subplot illustrates the implementation of the employed hardware-efficient ansatz. The middle subplot exhibits the ansatz searched by Pro-QAS. The right subplot compares the test accuracy between the hardware-efficient ansatz and the ansatz searched by Pro-QAS. The x-axis refers that the optimization of the l-th layer for Pro-QAS is accomplished.

dataset $\mathcal{D}$ [26]. After optimizing 200 epochs, the test accuracy of the hardware-efficient ansatz converges to 65%, which is exploited as the baseline. The center subplot illustrates the ansatz searched by Pro-QAS. Compared with the hardware-efficient ansatz, the number of CNOT gates reduces from 30 to 14, which suppresses noise by removing the unnecessary two-qubit gates. The achieved test accuracy is demonstrated in the right subplot. Specifically, when the optimization of the third layer is finished, Pro-QAS outperforms the baseline. When all L layers are optimized, the ansatz searched by Pro-QAS attains 75% test accuracy. The simulation results for $N = 10$ are exhibited in Supplementary Figure 12.



Supplementary Figure 12: **Simulation result for the 10-qubit case.** The left subplot illustrates the implementation of the employed hardware-efficient ansatz. The middle subplot exhibits the ansatz searched by Pro-QAS. The right subplot compares the test accuracy between the hardware-efficient ansatz and the ansatz searched by Pro-QAS. The x-axis has the same meaning with the one introduced in Supplementary Figure 11.

The simulation results for the case of $N = 10$ are exhibited in Supplementary Figure 12. Analogous to the case of $N = 7$, the hardware-efficient ansatz with $L = 3$ achieves 67% test accuracy, which is employed as the baseline. The results shown in the right subplot evidence that the ansatz searched by Pro-QAS is superior to the hardware-efficient ansatz. Concretely, after searching, the test accuracy improves to 76%. The center subplot illustrates the ansatz searched by Pro-QAS. To alleviate system noise and improve learning performance, the number of CNOT decreases from 27 to 18.

[1] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

[2] Sébastien Gerchinovitz and Tor Lattimore. Refined lower bounds for adversarial bandits. In *Advances in Neural Information Processing Systems*, pages 1198–1206, 2016.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

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

[5] W Pirie. Spearman rank correlation coefficient. *Encyclopedia of statistical sciences*, 12, 2004.

[6] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[7] Sam McArdle, Suguru Endo, Alan Aspuru-Guzik, Simon C Benjamin, and Xiao Yuan. Quantum computational chemistry. *Rev. Mod. Phys.*, 92(1):015003, 2020.

[8] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[9] Qiskit: An open-source framework for quantum computing, 2019.

[10] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[11] D Chivilikhin, A Samarin, V Ulyantsev, I Iorsh, AR Oganov, and O Kyriienko. Mog-vqe: Multiobjective genetic variational quantum eigensolver. *arXiv preprint arXiv:2007.04424*, 2020.

[12] Arthur G Rattew, Shaohan Hu, Marco Pistoia, Richard Chen, and Steve Wood. A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver. *arXiv preprint arXiv:1910.09694*, 2019.

[13] Richard S Sutton et al. *Introduction to reinforcement learning*, volume 135. MIT Press, 1998.

[14] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018.

[15] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement-induced barren plateaus. *PRX Quantum*, 2(4):040316, 2021.

[16] Taylor L Patti, Khadijeh Najafi, Xun Gao, and Susanne F Yelin. Entanglement devised barren plateau mitigation. *arXiv preprint arXiv:2012.12658*, 2020.

[17] Tyler Volkoff and Patrick J Coles. Large gradients via correlation in random parameterized quantum circuits. *Quantum Sci. Technol.*, 6(2):025008, 2021.

[18] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nat Commun*, 10(1):1–9, 2019.

[19] Andrea Skolik, Jarrod R McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. Layerwise learning for quantum neural networks. *Quantum Mach. Intell.*, 3(1):1–11, 2021.

[20] Ho Lun Tang, VO Shkolnikov, George S Barron, Harper R Grimsley, Nicholas J Mayhall, Edwin Barnes, et al. qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum*, 2(2):020310, 2021.

[21] Linghua Zhu, Ho Lun Tang, George S Barron, FA Calderon-Vargas, Nicholas J Mayhall, Edwin Barnes, et al. An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. *arXiv preprint arXiv:2005.10258*, 2020.

[22] Ernesto Campos, Aly Nasrallah, and Jacob Biamonte. Abrupt transitions in variational quantum circuit training. *Phys. Rev. A*, 103(3):032607, 2021.

[23] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

[24] Sébastien Gerchinovitz and Tor Lattimore. Refined lower bounds for adversarial bandits. In *Advances in Neural Information Processing Systems*, pages 1198–1206, 2016.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

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

[27] W Pirie. Spearman rank correlation coefficient. *Encyclopedia of statistical sciences*, 12, 2004.

[28] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[29] Sam McArdle, Suguru Endo, Alan Aspuru-Guzik, Simon C Benjamin, and Xiao Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 92(1):015003, 2020.

[30] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[31] Qiskit: An open-source framework for quantum computing, 2019.

[32] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[33] D Chivilikhin, A Samarin, V Ulyantsev, I Iorsh, AR Oganov, and O Kyriienko. Mog-vqe: Multiobjective genetic variational quantum eigensolver. *arXiv preprint arXiv:2007.04424*, 2020.

[34] Arthur G Rattew, Shaohan Hu, Marco Pistoia, Richard Chen, and Steve Wood. A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver. *arXiv preprint arXiv:1910.09694*, 2019.

[35] Richard S Sutton et al. *Introduction to reinforcement learning*, volume 135. MIT Press, 1998.

[36] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018.

[37] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement-induced barren plateaus. *PRX Quantum*, 2(4):040316, 2021.

[38] Taylor L Patti, Khadijeh Najafi, Xun Gao, and Susanne F Yelin. Entanglement devised barren plateau mitigation. *arXiv preprint arXiv:2012.12658*, 2020.

[39] Tyler Volkoff and Patrick J Coles. Large gradients via correlation in random parameterized quantum circuits. *Quantum Science and Technology*, 6(2):025008, 2021.

[40] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. An adaptive variational algorithm for

exact molecular simulations on a quantum computer. *Nature communications*, 10(1):1–9, 2019.

[41] Andrea Skolik, Jarrod R McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. Layerwise learning for quantum neural networks. *Quantum Machine Intelligence*, 3(1):1–11, 2021.

[42] Ho Lun Tang, VO Shkolnikov, George S Barron, Harper R Grimsley, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum*, 2(2):020310, 2021.

[43] Linghua Zhu, Ho Lun Tang, George S Barron, FA Calderon-Vargas, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. *arXiv preprint arXiv:2005.10258*, 2020.

[44] Ernesto Campos, Aly Nasrallah, and Jacob Biamonte. Abrupt transitions in variational quantum circuit training. *Physical Review A*, 103(3):032607, 2021.