Quantum Circuit Architecture Search on a Superconducting Processor

Kehuan Linghu,† Yang Qian,‡ Ruixia Wang,∗ Meng-Jun Hu, Zhiyuan Li,† Xuegang Li,† Huikai Xu,† Jingning Zhang,† Teng Ma,†, Peng Zhao,† Dong E. Liu, Min-Hsiu Hsieh,§ Xingyao Wu,§,‡ Yuxuan Du,§,‡ Dacheng Tao,§,‡,∗, Yirong Jin,† and Haifeng Yu,†

1. Introduction

The successful exhibition of random quantum circuits sampling and Boson sampling over fifty qubits [1–4] evidences the potential of using current quantum hardware to address classically challenging problems. A leading strategy towards this goal is variational quantum algorithms (VQAs) [5,6], which leverage classical optimizers to train an ansatz that can be implemented on noisy intermediate-scale quantum (NISQ) devices [7]. In the past years, a growing number of theoretical studies have shown the computational superiority of VQAs in the regime of machine learning [8–16], quantum many-body physics [17–20], and quantum information processing [21–25]. On par with the achievements, recent studies have recognized some flaws of current VQAs through the lens of the trade-off between expressivity and learning performance [14,24]. That is, an ansatz with very high expressivity may encounter the barren plateau issues [25–28], while an ansatz with low expressivity could fail to fit the optimal solution [29]. With this regard, designing a problemspecific and hardware-oriented ansatz is of great importance to guarantee the good learning performance of VQAs and the precondition of pursuing quantum advantages.
Pioneered experimental explorations have validated the crucial role of ansatz when applying VQAs to accomplish tasks in different fields such as machine learning [30–33], quantum chemistry [19,34–38], and combinatorial optimization [39–42]. On the one side, envisioned by the no-free-lunch theorem [43,44], there does not exist a universal ansatz that can solve all learning tasks with optimal performance. To this end, myriad handcraft ansätze have been designed to address different learning problems [45–47]. For instance, the unitary coupled cluster ansatz and its variants attain superior performance in the task of estimating molecular energies [48–51]. Besides devising the problem-specific ansätze, another indispensable factor to enhance the performance of VQAs is the compatibility between the exploited ansatz and the employed quantum hardware, especially in the NISQ scenario [39]. Concretely, when the circuit layout of ansatz mismatches with the qubit connectivity, additional quantum resources, e.g., SWAP gates, are essential to complete the compilation. Nevertheless, these extra quantum resources may inhibit the performance of VQAs because of the limited coherence time and inevitable gate noise of NISQ machines. Considering that there are countless learning problems and diverse architectures of quantum devices [52–54], it is impractical to manually design problem-specific and hardware-oriented ansätze.

To enhance the capability of VQAs, initial studies have been carried out to seek feasible strategies of automatically designing a problem-specific and hardware-oriented ansatz with both good trainability and sufficient expressivity. Conceptually, the corresponding proposals exploit random search [55], evolutionary algorithms [56,57], deep learning techniques [58–65], and adaptive strategies [66–68] to tailor a hardware-efficient ansatz [19], i.e., inserting or removing gates, to decrease the cost function. In contrast with conventional VQAs that only adjust parameters, optimizing both parameters and circuit layouts enables the enhanced learning performance of VQAs. Meanwhile, the automatic nature endows the power of these approaches to address broad learning problems. Despite the prospects, little is known about the effectiveness of these approaches executed on real quantum devices.

In this study, we demonstrate the first proof-of-principle experiment of applying an efficient automatic ansatz design technique, i.e., quantum architecture search (QAS) scheme [69], to enhance VQAs on an 8-qubit superconducting quantum processor. In particular, we focus on data classification tasks and utilize QAS to pursue a better classification accuracy. To our knowledge, this is the first experimental study of multi-class learning. Moreover, to understand the noise-resilient property of QAS, we fabricate a controllable dephasing noisy channel and integrate it into our quantum processor. Assisted by this technique, we experimentally demonstrate that the ansatz designed by QAS is compatible with the topology of the employed quantum hardware and attains much better performance than the hardware-efficient ansatz [19] when the system noise becomes large. Experimental results indicate that under a certain level of noise, the ansatz designed by QAS achieves the highest test accuracy (95.6%), while other heuristic ansätze only reach 90% accuracy. Additional analyses of loss landscape further explain the advantage of the QAS-based ansatz in both optimization and effective parameter space. These gains in performance suggest the significance of developing QAS and other automatic ansatz design techniques to enhance the learning performance of VQAs.

2. Materials and Methods

2.1. The Mechanism of QAS

The underlying principle of QAS is optimizing the quantum circuit architecture and the trainable parameters simultaneously to minimize an objective function. For elucidating, in the following, we elaborate on how to apply QAS to tailor the hardware-efficient ansatz (HEA). Mathematically, an N-qubit HEA \( U(\theta) = \prod_{l=1}^{L} U_l(\theta) \in SU(2^N) \) yields a multi-layer structure, where the circuit layout of all blocks is identical, the \( l \)-th block \( U_l(\theta) \) consists of a sequence of parameterized single-qubit and two-qubits gates, and \( L \) denotes the block number. Note that our method can be generalized to prune other ansätz, such as the unitary coupled cluster ansatz [49] and the quantum approximate optimization ansatz [70].
QAS is composed of four steps to tailor HEA and output a problem-dependent and hardware-oriented ansatz as shown in Figure 1. The first step is specifying the ansätze pool $\mathcal{S}$, collecting all candidate ansätze. Suppose that $U_l(\theta)$ for $\forall l \in [L]$ can be formed by three types of parameterized single-qubit gates, i.e., rotational gates along three axes, and one type of two-qubits gates, i.e., CNOT gates. When the layout of different blocks can be varied by replacing single-qubit gates or removing two-qubit gates, the ansätze pool $\mathcal{S}$ includes in total $O((3^N + 2^N)^L)$ ansatz. Denote the input data as $D$ and an objective function as $\mathcal{L}$. The goal of QAS is finding the best candidate ansatz $a \in \mathcal{S}$ and its corresponding optimal parameters $\theta^*_a$, i.e.,

$$ (\theta^*_a, a^*) = \arg \min_{\theta_a \in \mathcal{C}, a \in \mathcal{S}} \mathcal{L}(\theta_a, a, D, \mathcal{E}_a), $$

where the quantum channel $\mathcal{E}_a$ simulates the quantum system noise induced by $a$.

Figure 1. Experimental implementation of QAS. The first step is to construct the ansätze pool $\mathcal{S}$ which contains all candidate ansätze satisfying the hardware and physical constraints, such as hardware topology and maximal decoherence time. In the meantime, the parameter pool for all candidate ansätze is initialized in a layer-by-layer manner. The gate arrangement together with corresponding parameters constitutes the supernet. The second step is to sample ansatz from the supernet, measure the observable, calculate the loss, optimize the corresponding parameters based on the objective function, and update the parameters in the supernet. Repeat the above process until reaching the maximal number of iterations. Once the ansatz pool is well trained, the following steps are searching in $\mathcal{S}$, ranking according to performance, and selecting the optimal ansatz for fine-tuning.

The second step is optimizing Equation (1) with, in total, $T$ iterations. As discussed in our technical companion paper [69], seeking the optimal solution $(\theta^*_a, a^*)$ is computationally hard since the optimization of $a$ is discrete and the size of $\mathcal{S}$ and $\mathcal{C}$ exponentially scales with respect to $N$ and $L$. To conquer this difficulty, QAS exploits the supernet and weight-sharing strategy to ensure a good estimation of $(\theta^*_a, a^*)$ within a reasonable computational cost. Concisely, the weight-sharing strategy correlates parameters among different ansätze in $\mathcal{S}$ to reduce the parameter space $\mathcal{C}$. As for supernet, it plays two significant roles, i.e., configuring the ansätze pool $\mathcal{S}$ and parameterizing ansatz $a \in \mathcal{S}$ via the specified weight-sharing strategy. In doing so, at each iteration $t$, QAS randomly samples an ansatz $a^{(t)} \in \mathcal{S}$ and updates its parameters, with $\theta^{(t+1)}_a = \theta^{(t)}_a - \eta \nabla \mathcal{L}(\theta^{(t)}_a, a^{(t)}, D, \mathcal{E}_{a^{(t)}})$, and $\eta$ being the learning rate. Due to the weight-sharing strategy, the parameters of the unsampled ansätze are also updated.

The last two steps are ranking and fine-tuning. Specifically, once the training is completed, QAS ranks a portion of the trained ansätze and chooses the one with the best performance. The ranking strategies are diverse, including random searching and evolutionary searching. Finally, QAS utilizes the selected ansatz to fine-tune the optimized parameters with a few iterations. Refer to Ref. [69] for the omitted technical details of QAS.
2.2. Experimental Implementation

We implement QAS on a quantum superconducting processor to accomplish the classification tasks for the Iris dataset. Namely, the Iris dataset \( D = \{ x_i, y_i \}_{i=1}^{150} \) consists of three categories of flowers (i.e., \( y_i \in \{ 0, 1, 2 \} \)), and each category includes 50 examples characterized by 4 features (i.e., \( x_i \in \mathbb{R}^4 \)). In our experiments, we split the Iris dataset into three parts, i.e., the training dataset \( D_T = \{ x, y \} \), the validating dataset \( D_V \), and the test dataset \( D_E \) with \( D = D_T \cup D_V \cup D_E \). The functionality of \( D_T \), \( D_V \), and \( D_E \) is estimating the optimal classifier, preventing the classifier from being over-fitted, and evaluating the generalization property of the trained classifier, respectively.

Our experiments are carried out on a quantum processor, including 8 Xmon superconducting qubits with a one-dimensional chain structure. As shown in Figure 2b, the employed quantum device is fabricated by sputtering an aluminum thin film onto a sapphire substrate. The single-qubit rotation gate \( R_X (R_Y) \) along X-axis (Y-axis) is implemented with a microwave pulse, and the Z rotation gate \( R_z \) is realized by virtual Z gate \([71]\). The construction of the CZ gate is completed by applying the avoided level crossing between the high level states \( |11 \rangle \) and \( |02 \rangle \) or \( |11 \rangle \) and \( |20 \rangle \). The calibrated readout matrix is shown in Figure 2d and the device parameters is summarized in Table A1 of Appendix A.

![Figure 2. Experimental setups.](image)

(a) The construction of quantum classifiers with the exploited three different ansätze, i.e., hardware-agnostic ansatz (HAA), hardware-efficient ansatz (HEA), and the ansatz searched by QAS, towards the Iris dataset. For all classifiers, the gate encoding method is adopted to embed the classical feature vector \( x_i \) into the quantum state \( \rho_i \). After the interaction of \( \rho_i \) with the ansatz \( U(\theta) \), the generated state is measured by a fixed operator \( \Pi \) to obtain the prediction \( \text{Tr}(\Pi U(\theta) \rho_i U(\theta)^\dagger) \). (b) All three quantum classifiers are deployed on an 8-qubit superconducting processor with the chain topology. The activated qubits are highlighted by the purple color. (c,d) To suppress the system noise, error mitigation techniques of measurements are used in our quantum hardware. Namely, the collected measurement results are operated with a correction matrix to estimate the ideal results. Refer to the Method section for details. (e) A classical optimizer continuously updates the parameters in \( U(\theta) \) to minimize the discrepancy between the predictions of quantum classifiers and ground-truth labels indicated by the objective function.

We fabricate the controllable dephasing noise as a measurable disturbance to the quantum evolution. The operators for the noise channel can be written as \( E_0 = \sqrt{1 - \alpha} p [1, 0; 0, 1] \) and \( E_1 = \sqrt{\alpha} p [1, 0; 0, -1] \). \( \alpha \) is a constant, and the value of \( p \) can be tuned in our experiment by changing the average number of the coherent photons on the readout cavity’s steady state. The intensity of coherent photons is represented by the amplitude \( p \) of the curve shown on the AWGs.
The experimental implementation of the quantum classifiers is as follows. As illustrated in Figure 2a, the gate encoding method is exploited to load classical data into quantum states. The encoding circuit yields \( U_F(x) = \otimes_{j=1}^4 R_Y(x_{ij}) \). To evaluate the effectiveness of QAS, three types of ansätze \( U(\theta) \) are used to construct the quantum classifier. The first two types are heuristic ansätze, which are the hardware-agnostic ansatz (HAA) and hardware-efficient ansatz (HEA). As depicted in Figure 2a, HAA \( U_{\text{HAA}}(\theta) \) is designed for a general paradigm and ignores the topology of specific quantum hardware platforms; HEA \( U_{\text{HEA}}(\theta) \) adapts to the quantum hardware constraints, where all inefficient two-qubit operators that connect two physically nonadjacent qubits are forbidden. The third type of ansatz refers to the output of QAS, denoted as \( U_{\text{QAS}}(\theta) \). The mean square error between the prediction and real labels is employed as the objective function for all quantum classifiers. The noise rate of the dephasing channel \( p \) is set as 0, 0.01, and 0.015. We benchmark the test accuracy of these three ansätze HAA, HEA, and QAS, and explore whether QAS attains the highest test accuracy. Refer to Appendix B for more implementation details.

3. Results

To comprehend the importance of the compatibility between quantum hardware and ansätze, we first examine the learning performance of the quantum classifiers with HAA and HEA under different noise rates. The achieved experimental results are demonstrated in Figure 3a. In particular, in the measure of training loss (i.e., the lower the better), the quantum classifier with the HEA significantly outperforms HAA for all noise settings. At the 10-th epoch, the training loss of the quantum classifier with HAA and HEA is 0.06 and 0.017 (0.049 and 0.012; 0.049 and 0.012) when \( p = 0.015 (p = 0; p = 0.01) \), respectively. In addition, the optimization of the quantum classifier with HAA seems to be divergent when \( p = 0.015 \). We further evaluate the test accuracy to compare their learning performance. As shown in Figure 3b, there exists a manifest gap between the two ansätze, highlighted by the blue and yellow colors. For all noise settings, the test accuracy corresponding to HAA is only 31.1%, whereas the test accuracy corresponding to HEA is at least 95.6%. These observations signify the significance of reconciling the topology between the employed quantum hardware and ansatz as the key motivation of this study. Specifically, the performance gap between HAA and HEA drives us to introduce QAS, which focuses on searching for the optimal quantum circuit architecture rather than iteratively updating gate parameters within a single, fixed architecture.

![Figure 3. The performance of quantum classifiers. (a) The training loss of quantum classifiers with the HAA and HEA ansätze under different noise settings. (b) The test accuracy achieved by HAA, HEA, and the ansatz searched by QAS under different noise settings.](image)

We next experiment on QAS to quantify how it is a problem-specific and hardware-oriented design to enhance the learning performance of quantum classifiers. Concretely, as shown in Figure 3b, for all noise settings, the quantum classifier with the ansatz searched by QAS attains the best test accuracy compared to those of HAA and HEA. That is, when \( p = 0 (p = 0.01 \text{ and } p = 0.015) \), the test accuracy achieved by QAS is 97.8% (97.8% and 95.6%), which is higher than HEA with 96.7% (92.2% and 90.0%). Notably, although the test accuracy is slightly decreased for the increased system noise, the strength of QAS
becomes evident over the other two ansätze. In other words, QAS shows the advantages of simultaneously alleviating the effect of quantum noise and searching the optimal ansatz to achieve high accuracy. The superior performance validates the effectiveness of QAS in classification tasks.

We last investigate the potential factors of ensuring the good performance of QAS from two perspectives, i.e., the circuit architecture and the corresponding loss landscape. The searched ansätze under three noise settings, HAA, and HEA are pictured at the top of Figure 4. Compared with HEA and HAA, QAS reduces the number of CZ gates with respect to the increased level of noise. When \( p = 0.015 \), QAS chooses the ansatz containing only one CZ gate. This behavior indicates that QAS can adaptively control the number of quantum gates to balance the expressivity and learning performance. We plot the loss landscape of HAA, HEA, and the ansatz searched by QAS in the middle row of Figure 4.

To visualize the high-dimension loss landscape in a 2D plane, the dimension reduction technique, i.e., principal component analysis (PCA) [72] is applied to compress the parameter trajectory corresponding to each optimization step. After dimension reduction, we choose the obtained first two principal components that explain most of the variance as the landscape spanning vector. Refer to [73] and Appendix C for details. For HAA and HEA, the objective function is governed by both the 0-th component (98.75% of variance for HAA, 98.55% of variance for HEA) and 1-th component (1.24% of variance for HAA, 1.45% of variance for HEA). By contrast, for the ansätze searched by QAS, their loss landscapes totally depend on the 0-th component. Furthermore, the optimization path for QAS is exactly linear, while the optimization of HAA and HEA experiences a nonlinear curve. This difference reveals that QAS enables a more efficient optimization trajectory. As indicated in the bottom row of Figure 4, there is a major parameter that contributes the most to the 0-th component in the three ansätze searched by QAS, while HAA and HEA have to consider multiple parameters to determine the 0-th component. This phenomenon reflects that ansätze searched by QAS are prone to having a smaller effective parameter space, which leads to less noise accumulation and further stronger noise robustness. These observations can be treated as empirical evidence to explain the superiority of QAS.

Figure 4. The circuit architecture and corresponding loss landscape. The top row demonstrates the structure of HAA, HEA, and the ansatz searched by QAS under different noise settings. The color and angles refer to the gate type and corresponding parameter. The middle row visualizes the loss landscape of each ansatz with respect to the trained parameters based on the technique developed in [73]. The red line tracks the optimization path of loss during the 50 epochs. The linear path indicates that the loss landscape enjoys a simple structure and optimization is easy to converge. The bottom row shows the absolute value of the first row vector of the PCA transform matrix, which reflects the contribution of each parameter to the first component. Numbers 0–7 denote the parameter index.
4. Discussion

Our experimental results provide the following insights. First, we experimentally verify the feasibility of applying automatically designing a problem-specific and hardware-oriented ansatz to improve the power of quantum classifiers. Second, the analysis related to the loss landscape and the circuit architectures exhibits the potential of applying QAS and other variable ansatz construction techniques to compensate for the caveats incurred by executing variational quantum algorithms on NISQ machines.

Besides classification tasks, it is crucial to benchmark QAS and its variants towards other learning problems in quantum chemistry and quantum many-body physics. In these two areas, the employed ansatz is generally Hamiltonian dependent [48,49,74]. As a result, the way of constructing the ansatz pool should be carefully conceived. In addition, another important research direction is understanding the capabilities of QAS for large-scale problems. How to find the near-optimal ansatz among the exponential candidates is a challenging issue.

We note that although QAS can reconcile the imperfection of quantum systems, a central law to enhance the performance of variational quantum algorithms is promoting the quality of quantum processors. For this purpose, a promising area for future exploration is to delve into carrying out QAS and its variants on more types of quantum machines to accomplish more real-world generation tasks with potential advantages.

Author Contributions: Y.D. and H.Y. conceived the research. K.L. and Y.Q. designed and performed the experiment. Y.Q. and Y.D. performed numerical simulations. Y.Q., X.W., R.W., M.-J.H. and D.E.L. analyzed the results. All authors contributed to discussions of the results and the development of the manuscript. Y.Q., R.W., Y.D. and K.L. wrote the manuscript with input from all co-authors. Y.D., X.W., D.T. and R.W. supervised the whole project. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the NSFC of China (No. 12104056, No. 12104055, No. 12404558, Grant No. 12004042), the NSF of Beijing (Grant No. Z190012), and the Key-Area Research and Development Program of Guang Dong Province (Grant No. 2018B030326001).

Data Availability Statement: Data will be shared when there is a demand.

Acknowledgments: We appreciate the helpful discussion with Weiyang Liu and Guangming Xue.

Conflicts of Interest: Yang Qian, Xingyao Wu, Yuxuan Du and Dacheng Tao are employed by JD Explore Academy. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Experiment Setup

Appendix A.1. Device Parameters

The qubit parameters, length, and fidelity for the single- and two-qubit gates of our device are summarized in Table A1.

Appendix A.2. Electronics and Control Wiring

The device is installed in a cryogenic setup in a dilution refrigerator. The control and measurement electronics which are connected to the device are shown in Figure A1. The electronic control module is divided into six areas with different temperatures. The superconducting quantum device is installed at the base plate with a cryogenic environment of 10 mK. For each qubit, the frequency is tuned by a flux control line by changing the magnetic flux through the SQUID loop, and the flux is controlled by a constant current, which is generated by a voltage source and inductively coupled to the SQUID. Four attenuators are connected in the circuit in series to act as the thermal precipitator. The XY control for each qubit is achieved by up-converting the intermediate frequency signals with an analog IQ-mixer module. The drive pulse is provided by the multi-channel AWGs with a sample rate of 2 GSa/s.
Table A1. Device parameters. $\omega_i$ and $\alpha_i$ represent the qubit frequency and qubit anharmonicity, respectively. $T_1$ and $T_2^*$ are the longitudinal and transverse relaxation times, respectively. $\bar{F}_s$ and $T_s$ are the average fidelity and length of the single-qubit gates. $g_i$ is the coupling strength between nearby qubits, and $I_{zz}$ is the effective ZZ coupling strength. $\bar{F}_{cz}$ are the fidelity of the CZ gates calibrated by quantum process tomography, and $T_{cz}$ are the length of the CZ gates.

| Parameter | Q1  | Q2  | Q3  | Q4  | Q5  | Q6  | Q7  | Q8  |
|-----------|-----|-----|-----|-----|-----|-----|-----|-----|
| $\omega_i/2\pi$ (GHz) | 5.202 | 4.573 | 5.146 | 4.527 | 5.099 | 4.47 | 5.118 | 4.543 |
| $\alpha_i/2\pi$ (GHz) | $-0.240$ | $-0.240$ | $-0.239$ | $-0.240$ | $-0.239$ | $-0.239$ | $-0.239$ | $-0.242$ |
| $T_1$ (µs) | 10.3 | 13.6 | 14.7 | 16.7 | 14.9 | 13.2 | 12.7 | 8.0 |
| $T_2^*$ (µs) | 11.7 | 2.0 | 15.2 | 1.9 | 15.9 | 1.8 | 12.6 | 1.6 |
| $\bar{F}_s$ | 99.67% | 98.96% | 99.76% | 99.55% | 99.05% | 99.69% | 98.51% | 99.09% |
| $T_s$ (ns) | 37 | 37 | 37 | 35 | 35 | 37 | 37 | 35 |
| $g_i/2\pi$ (MHz) | 19.7 | 19.6 | 19.1 | 19.1 | 19.2 | 19.3 | 19.6 | 0.675 | 0.8 | 0.7 | 0.78 | 0.626 | 0.655 | 0.819 |
| $I_{zz}/2\pi$ (MHz) | 97.23% | 96.11% | 97.48% | 93.54% | 93.72% | 94.95% | 95.54% | 18.5 | 21.5 | 23 | 19.5 | 23.5 | 21.5 | 18.5 |

Figure A1. Schematic diagram for the electronics and wiring setup for the superconducting quantum system.

The qubit readout is performed by a readout control system with a sampling rate of 1 GSa/s. The readout pulse is up-converted to the frequency band of the readout cavity with the analog IQ-mixer and transmitted through the readout line with attenuators and low-pass filters to the chip. At the output side, the constant and alternating currents are combined by a bias-tee, amplified by the parametric amplifier, and connected to the readout line through a circulator at 10 mK, as well as a high-electron-mobility transistor (HEMT) at 4 K and two more amplifiers at room temperature. The amplified signals finally are digitized by an analog-to-digital converter.

Appendix A.3. Noise Setup

Due to the ac Stark effect, photon number fluctuations from the readout cavity can cause qubit dephasing [75]. We implement a pure dephasing noisy channel in our device.
To every qubit, the noise photons are generated by a coherent source with a Lorentzian-shaped spectrum, which is centered at the frequency of $\omega_c$. $\omega_c$ is the center frequency of the readout cavity, which is over-coupled to the feedline at the input and output port, and capacitively coupled to the Xmon qubit. The Hamiltonian system including the readout cavity and the qubit can be written as

$$H/\hbar = \omega_c a^\dagger a + \frac{\omega_q}{2}\sigma_z + g_r (a^\dagger \sigma_- + a \sigma_+)$$  \hspace{1cm} (A1)

where $\sigma_\pm = \sigma_x \pm i\sigma_y$, and $\sigma_j (j = x, y, z)$ is the Pauli operator for the X-mon qubit. $a^\dagger$ ($a$) is the cavity photon creation (annihilation) operator. $\omega_q$ is the frequency between the ground and the first excited states of the qubit, and $g_r$ is the coupling strength between the qubit and the readout cavity.

By continuously sending the coherent photons to drive the readout cavity to maintain a coherent state, a noisy environment can be engineered. The noise channel can be described as the depolarization in the x-y plane of the Bloch sphere. The noise intensity can be tuned by changing the average number of coherent photons on the readout cavity’s steady state. The average number of photons is represented by the amplitude of the curve shown in the AWGs. Under different noise settings, the values of $T^*_2$ are shown in Table A2.

| Parameter | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 |
|-----------|----|----|----|----|----|----|----|----|
| $T^*_2 (p = 0)$ (µs) | 11.7 | 2.0 | 15.2 | 1.9 | 15.9 | 1.8 | 12.6 | 1.6 |
| $T^*_2 (p = 0.005)$ (µs) | 10.6 | 1.9 | 15.3 | 1.8 | 15.0 | 1.6 | 10.7 | 1.5 |
| $T^*_2 (p = 0.01)$ (µs) | 7.8 | 1.6 | 12.5 | 1.7 | 14.7 | 1.6 | 13.0 | 1.6 |
| $T^*_2 (p = 0.015)$ (µs) | 0.2 | 1.2 | 12.7 | 1.6 | 14.4 | 1.6 | 12.1 | 1.5 |

Appendix A.4. Readout Correction

The experimentally measured results of the final state for the eight qubits were corrected with a calibration matrix, which can be obtained in an experimental calibration process. The reconstruction process for readout results is based on Bayes’ rule. The colored schematic diagram for the calibration matrix is shown in Figure 2d. Assume that $p_{ij}$ stands for the probability of obtaining a measured population $|i\rangle$ when preparing a basis state $|j\rangle$. The calibration matrix is

$$F = \begin{pmatrix} p_{11} & p_{12} & \ldots & p_{12^n} \\ p_{21} & p_{22} & \ldots & p_{22^n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{2^{2^n}1} & p_{2^{2^n}2} & \ldots & p_{2^{2^n}2^n} \end{pmatrix}. \hspace{1cm} (A2)$$

If we prepare a state $|\psi\rangle$ on $n$ qubits, and the probability distribution of the prepared state in $2^n$ basis is $P = [p_1, p_2, \ldots, p_{2^n}]^T$, then we will obtain a measured state probability distribution as $\tilde{P}$ in the experiment. The relationship between the two probability distributions is

$$\tilde{P} = FP. \hspace{1cm} (A3)$$

Solving for $P$, we have

$$P = F^{-1}\tilde{P}. \hspace{1cm} (A4)$$

Appendix B. Implementation of Quantum Classifiers

In this section, we implement HAA, HEA, and QAS for the classification of the Iris dataset on the 8-qubit superconducting quantum processors with controllable dephasing noise. A detailed description of the dataset and hyper-parameters configuration is given below.
Appendix B.1. Dataset

The classification data employed in this paper are from the Iris dataset [76], which contains 150 instances characterized by 4 attributes and 3 categories. Each dimension of the feature vector is normalized to range \([0, 1]\). During training, the whole dataset is split into three parts, including the training set (60 samples), validation set (45 samples), and test set (45 samples). The sample distribution is visualized by selecting the first two dimensions of the feature vector. As shown in Figure A2, samples of class 1 and 2 cannot be distinguished by a linear classifier. It means that nonlinearity should be introduced into the quantum classifier to achieve higher classification accuracy.

![Figure A2](image_url)

Figure A2. The visualization of the Iris dataset based on its first two dimensions.

Appendix B.2. Objective Function and Accuracy Measure

Objective function. We adopt the mean square error (MSE) as the objective function for all quantum classifiers, i.e.,

\[
L(D, \theta) = \frac{1}{2n} \sum_{i=1}^{n} (\langle O \rangle_i - y_i)^2, \quad (A5)
\]

where \(\langle O \rangle_i = \langle 0 | U_E(x_i) \dagger U(\theta) \dagger O U(\theta) U_E(x_i) | 0 \rangle\), \(O\) refers to the observable, \(U_E(x_i)\) denotes the unitary operator that embeds classical feature vector \(x_i\) into the quantum circuit, and \(U(\theta)\) is the variational quantum circuit with the trainable parameters \(\theta\).

Definition of train, valid, and test accuracy. Given an example \(x_i\), the quantum classifier predicts its label as

\[
\tilde{y}_i = \begin{cases} 
0, & \langle O \rangle_i \leq \frac{1}{6} \\
1, & \frac{1}{6} < \langle O \rangle_i \leq \frac{1}{2} \\
2, & \frac{1}{2} < \langle O \rangle_i \leq \frac{5}{6}
\end{cases}, \quad (A6)
\]

The train (valid and test) accuracy aims to measure the differences between the predicted labels and true labels for examples in the training dataset \(D_T\) (valid dataset \(D_V\) and test dataset \(D_E\)), i.e.,

\[
\text{accuracy} = \frac{\sum_{(x_i, y_i) \in D} \mathbb{1}_{\tilde{y}_i = y_i}}{|D|}, \quad D = D_T \text{ or } D_V \text{ or } D_E, \quad (A7)
\]

where \(| \cdot |\) denotes the size of a set.
Appendix B.3. Training Hyper-Parameters

The trainable parameters for all ansätze are randomly initialized following the uniform distribution $U[-\pi, \pi]$. During training, the hyper-parameters are set as follows: the optimizer is stochastic gradient descent (SGD) [77], the batch size is 4, and the learning rate is fixed at 0.2. Specifically, the parameter shift rule [78] is applied to compute the gradient of the objective function with respect to a single parameter.

For QAS, we train 5 candidate supernets for 40 epochs to fit the training set. During the search phase, we randomly sample 100 ansatz and rank them according to their accuracy on the validation set. Finally, the ansatz with the highest accuracy is selected as the target ansatz. The ansätze pool is constructed as follows. For the single-qubit gate, the candidate set is $\{RY, RZ\}$. For the two-qubit gate, QAS automatically determines whether to apply $CZ$ gates to the qubit pair $(0, 1), (1, 2), (2, 3)$ or not, discarding all other combinations, such as $(0, 2)$ and $(0, 3)$. These nonadjacent qubits connections require more gates when running on the superconducting processor of 1-D chain topology, leading to bigger noise accumulation.

Appendix C. More Details of the Results

Appendix C.1. PCA Used in the Visualization of Loss Landscape

To visualize the loss landscape of HAA, HEA, and ansätze searched by QAS with respect to the parameter space, we apply principle component analysis (PCA) to the parameter trajectory collected in every optimization step and choose the first two components as the observation variable. To be concrete, given a sequence of trainable parameter vectors along the optimization trajectory $\{\theta(1), ..., \theta(t), ..., \theta(T)\}$, where $T$ is the number of total optimization steps and $\theta(t) \in \mathbb{R}^d$ denotes the parameter vector at the $t$-th step, we construct the matrix $\Theta = [\theta(1); \cdots; \theta(T)] \in \mathbb{R}^{T \times d}$. Once we apply PCA to $\Theta$ and obtain the first two principal components $E = [e_0, e_1] \in \mathbb{R}^{2 \times d}$, the loss landscape with respect to trainable parameters can be visualized by performing a 2D scan for $E\Theta^T$. Simultaneously, the projection vector $e_i$ of each component indicates the contribution of each parameter to this component, implying how many parameters determine the value of the objective function. Refer to [73] for details.

The optimization trajectory can provide certain information on the trainability and convergence of the employed ansatz in quantum classifiers. When the optimization path is exactly linear, it implies that the loss landscape is not intricate and the model can be easily optimized. On the contrary, the complicated nonlinear optimization curve indicates the difficulty of covering to the local minima.

Appendix C.2. Simulation Results

We conduct numerical experiments on classical computers to validate the effectiveness of QAS.

Dephasing noise. We simulate the dephasing noise channel as

$$\rho' = (1 - \tilde{\rho})\rho + \tilde{\rho}\sigma_z\rho\sigma_z,$$

where $\rho$ and $\rho'$ represent the ideal quantum state (density matrix) and noisy quantum state affected by the dephasing channel, $\sigma_z$ is the Pauli-Z operator, and $\rho = ap$ is the noise strength, representing the probability of applying a Pauli-Z operator to the quantum state. In the experiments, the noise strength $p$ is set as $\{0.05, 0.1, 0.15\}$, and the circuit layer $L$ is set as $\{2, 4, 6\}$. Each setting runs 10 times to suppress the effects of randomness.

Simulation results. As shown in Figure A3, QAS achieves the highest test accuracy over all noise and layer settings. When $L = 2$ and $p = 0.05$, the performance gap between HEA (95.7%) and QAS (97.1%) is relatively small. With both the depth and noise strength increasing, HEA witnesses a rapid accuracy drop (60% for $L = 6$ and $p = 0.15$). By contrast, the test accuracy for QAS with $L = 6$ and $p = 0.15$ is 95%, which slightly decreases 2%. This behavior accords with the results on the superconducting processor (the test accuracy of QAS running a superconducting device decreases from 97.8% to 95.6% when $p$ increases.
from 0 to 0.015; refer to Figure 3 for more details), further illustrating the advantage of QAS in error mitigation and model expressivity.

Figure A3. The test accuracy achieved by HEA and QAS under various numbers of layers and noise strength when simulating on classical devices.

Appendix C.3. QAS in Quantum Approximate Optimization Algorithm (QAOA)

The quantum approximate optimization algorithm [41] is a variational quantum algorithm used to solve combinatorial optimization problems. In our work, QAOA is employed to search the maximum cut of an undirected graph. To this end, the system Hamiltonian generated from a graph $G = (V, E)$ with vertices $V$ and edges $E$ is given as

$$ C = \sum_{j,k \in E} Z_j Z_k, $$

(A9)

where $Z_j$ is the Pauli-Z operator. Theoretically, the optimal partition $\psi$ is exactly deduced from the state $|\psi\rangle$, minimizing the expectation $\langle \psi | C | \psi \rangle$. In practice, the target state is approximated by following variation quantum circuit with depth $p$:

$$ |\gamma, \beta\rangle = \prod_{i=1}^{p} U_B(\beta_i) U_C(\gamma_i) |+\rangle, $$

(A10)

where the unitary gate $U_C(\gamma_i)$ is generated by the Hamiltonian $U_C(\gamma_i) = \exp(-i\gamma_i C)$, $U_B(\beta_i)$ is usually selected as $\exp(-i\beta_i X)$, $\gamma_i$ and $\beta_i$ are trainable parameters. Following the hybrid quantum–classical optimization scheme, we can asymptotically approach the solution by minimizing the expectation $\langle \gamma, \beta | C | \gamma, \beta \rangle$ with infinite $p$.

Considering the system noise and limited qubit connectivity of quantum hardware, a natural generalization of $U_B$ is a generic unitary operator with free rotation axis [79] or more trainable parameters [80–82]. Inspired by the success of QAS in classification, we also apply QAS to automatically search the optimal mixer Hamiltonian to match the qubit connectivity and mitigate error.

We study the 8-qubit system for the max-cut problem of graph with 8 vertices. Due to the chain connection of physical qubits, edges connecting vertices $V_i$ and $V_j$ with $|i-j| > 2$ are discarded as shown in Figure A4a. The graph node indexed from 1 to 8 corresponds to the qubits from Q3 to Q10. An optional graph partition achieving the maximum cut 7 is depicted in Figure A4b. Following the protocol of QAOA and the target graph, we employ the ansatz in Figure A4c to search the max-cut, where $U_C$ is generated by the system Hamiltonian $C$, and $U_B$ is generated by the mixer Hamiltonian. As drawn in Figure A4d, the heuristic layout of $U_B$ is described by $U_B(\beta_i) = R_X(\beta_i) \otimes 8$, while QAS introduces more candidate Pauli operators and entangling gates to enhance the flexibility of $U_B$. The performance of QAS for QAOA is demonstrated in Figure A4e,f. It is obvious that the mixer Hamiltonian searched by QAS achieves smaller loss $\langle \psi | C | \psi \rangle$ than the vanilla mixer Hamiltonian for various Trotter steps and noise scales, except for the extreme setting.
of $p = 2$ and 0.015 noise strength. We further measure the approximation ratio $r = \frac{A}{\Lambda_{\text{max}}}$, where $A$ is the graph cut solved by QAOA, and $\Lambda_{\text{max}}$ is the theoretical maximum cut. When corrupted by noise with increasing scales, the standard QAOA witnesses a gradual reduction in the average approximation ratio. Instead, the quantum ansatz generated by QAS can guarantee to find the optimal solution even when suffering from noise of intensity 0.015.

Figure A4. Experimental results for the QAOA. (a) An instance of a graph with 8 vertices. (b) An optional partition achieving the max cut. (c) The QAOA circuit. (d) QAS helps design $U_B$. (e) The loss comparison between standard QAOA (termed ‘baseline’) and QAS. (f) The approximation ratio achieved by standard QAOA and QAS.

References
1. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* 2019, 574, 505–510. [CrossRef] [PubMed]
2. Wu, Y.; Bao, W.S.; Cao, S.; Chen, F.; Chen, M.C.; Chen, X.; Chung, T.H.; Deng, H.; Du, Y.; Fan, D.; et al. Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.* 2021, 127, 180501. [CrossRef] [PubMed]
3. Zhong, H.S.; Wang, H.; Deng, Y.H.; Chen, M.C.; Peng, L.C.; Luo, Y.H.; Qin, J.; Wu, D.; Ding, X.; Hu, Y.; et al. Quantum computational advantage using photons. *Science* 2020, 370, 1460–1463. [CrossRef] [PubMed]
4. Zhu, Q.; Cao, S.; Chen, F.; Chen, M.C.; Chen, X.; Chung, T.H.; Deng, H.; Du, Y.; Fan, D.; Gong, M.; et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling. *Sci. Bull.* 2021, 67, 240–245. [CrossRef]
5. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J.S.; Menke, T.; et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* 2022, 94, 015004. [CrossRef]
6. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* 2021, 3, 625–644. [CrossRef]
7. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* 2018, 2, 79. [CrossRef]
8. Abbas, A.; Sutter, D.; Zoufal, C.; Lucchi, A.; Figalli, A.; Woerner, S. The power of quantum neural networks. *Nat. Comput. Sci.* 2021, 1, 403–409. [CrossRef]
9. Banchi, L.; Pereira, J.; Pirandola, S. Generalization in quantum machine learning: A quantum information standpoint. *PRX Quantum* 2021, 2, 040321. [CrossRef]
10. Bu, K.; Koh, D.E.; Li, L.; Luo, Q.; Zhang, Y. Statistical complexity of quantum circuits. *Phys. Rev. A* 2022, 105, 062431. [CrossRef]
11. Caro, M.C.; Datta, I. Pseudo-dimension of quantum circuits. *Quantum Mach. Intell.* 2020, 2, 14. [CrossRef]
12. Caro, M.C.; Huang, H.Y.; Cerezo, M.; Sharma, K.; Sornborger, A.; Cincio, L.; Coles, P.J. Generalization in quantum machine learning from few training data. *Nat. Commun.* 2022, 13, 4919. [CrossRef]
13. Du, Y.; Hsieh, M.H.; Liu, T.; You, S.; Tao, D. Learnability of Quantum Neural Networks. *PRX Quantum* 2021, 2, 040337. [CrossRef]
14. Du, Y.; Tu, Z.; Yuan, X.; Tao, D. Efficient measure for the expressivity of variational quantum algorithms. *Phys. Rev. Lett.* 2022, 128, 080506. [CrossRef]
15. Huang, H.Y.; Kueng, R.; Preskill, J. Information-theoretic bounds on quantum advantage in machine learning. *Phys. Rev. Lett.* 2021, 126, 190505. [CrossRef] [PubMed]
16. Huang, H.Y.; Broughton, M.; Mohseni, M.; Babbush, R.; Boixo, S.; Neven, H.; McClean, J.R. Power of data in quantum machine learning. *Nat. Commun.* 2021, 12, 1–9. [CrossRef]
17. Huang, H.Y.; Kueng, R.; Torlai, G.; Albert, V.V.; Preskill, J. Provably efficient machine learning for quantum many-body problems. *Science* 2022, 377, eabk3333. [CrossRef] [PubMed]
18. Endo, S.; Sun, J.; Li, Y.; Benjamin, S.C.; Yuan, X. Variational quantum simulation of general processes. *Phys. Rev. Lett.* 2020, 125, 010501. [CrossRef]

19. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.M.; Gambetta, J.M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* 2017, 549, 242–246. [CrossRef]

20. Pagano, G.; Bapat, A.; Becker, P.; Collins, K.S.; De, A.; Hess, P.W.; Kaplan, H.B.; Kyprianidis, A.; Tan, W.L.; Baldwin, C.; et al. Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proc. Natl. Acad. Sci. USA* 2020, 117, 25396–25401. [CrossRef]

21. Cerezo, M.; Poremba, A.; Cincio, L.; Coles, P.J. Variational quantum fidelity estimation. *Quantum* 2020, 4, 248. [CrossRef]

22. Du, Y.; Tao, D. On exploring practical potentials of quantum auto-encoder with advantages. *arXiv* 2021, arXiv:2106.15432.

23. Carolan, J.; Mohseni, M.; Olson, J.P.; Prabhu, M.; Chen, C.; Bunandar, D.; Niu, M.Y.; Harris, N.C.; Wong, F.N.; Hochberg, M.; et al. Variational quantum unsampling on a quantum photonic processor. *Nat. Phys.* 2020, 16, 322–327. [CrossRef]

24. Holmes, Z.; Sharma, K.; Cerezo, M.; Coles, P.J. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum* 2022, 3, 010313. [CrossRef]

25. McClean, J.R.; Boixo, S.; Smelyanskiy, V.N.; Babbush, R.; Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* 2018, 9, 4812. [CrossRef]

26. Cerezo, M.; Sone, A.; Volkoff, T.; Cincio, L.; Coles, P.J. Cost function dependent barren plateaus in shallow parameterized quantum circuits. *Nat. Commun.* 2021, 12, 1–12. [CrossRef]

27. Pesah, A.; Cerezo, M.; Wang, S.; Volkoff, T.; Sornborger, A.T.; Coles, P.J. Absence of Barren Plateaus in Quantum Convolutional Neural Networks. *Phys. Rev. X* 2021, 11, 041011. [CrossRef]

28. Grant, E.; Wossnig, L.; Ostaszewski, M.; Benedetti, M. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum* 2019, 3, 214. [CrossRef]

29. Bravyi, S.; Kliessch, A.; Koenig, R.; Tang, E. Obstacles to variational quantum optimization from symmetry protection. *Phys. Rev. Lett.* 2020, 125, 260505. [CrossRef]

30. Havlíček, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised learning with quantum-enhanced feature spaces. *Nature* 2019, 567, 209–212. [CrossRef]

31. Huang, H.L.; Du, Y.; Gong, M.; Zhao, Y.; Wu, Y.; Wang, C.; Li, S.; Liang, F.; Lin, J.; Xu, Y.; et al. Experimental quantum generative adversarial networks for image generation. *Phys. Rev. Appl.* 2021, 16, 024051. [CrossRef]

32. Peters, E.; Caldeira, J.; Ho, A.; Leichenauer, S.; Mohseni, M.; Neven, H.; Spentzouris, P.; Strain, D.; Perdue, G.N. Machine learning of high dimensional data on a noisy quantum processor. *npj Quantum Inf.* 2021, 7, 161. [CrossRef]

33. Rudolph, M.S.; Toussaint, N.B.; Katabarwa, A.; Johri, S.; Peropadre, B.; Perdomo-Ortiz, A. Generation of high-resolution handwritten digits with an ion-trap quantum computer. *Phys. Rev. X* 2022, 12, 031010. [CrossRef]

34. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Boixo, S.; Broughton, M.; Buckley, B.B.; Buell, D.A.; et al. Hartree-Fock on a superconducting qubit quantum computer. *Science* 2020, 369, 1084–1089.

35. Robert, A.; Barkoutsos, P.K.; Woerner, S.; Tavernelli, I. Resource-efficient quantum algorithm for protein folding. *npj Quantum Inf.* 2021, 7, 38. [CrossRef]

36. Kais, S. Introduction to quantum information and computation for chemistry. In *Quantum Information and Computation for Chemistry*; Wiley: Hoboken, NJ, USA, 2014; pp. 1–38.

37. Wecker, D.; Hastings, M.B.; Wiebe, N.; Clark, B.K.; Nayak, C.; Troyer, M. Solving strongly correlated electron models on a quantum computer. *Phys. Rev. A* 2015, 92, 062318. [CrossRef]

38. Cai, X.; Fang, W.H.; Fan, H.; Li, Z. Quantum computation of molecular response properties. *Phys. Rev. Res.* 2020, 2, 033324. [CrossRef]

39. Harrigan, M.P.; Sung, K.J.; Neeley, M.; Satzinger, K.J.; Arute, F.; Arya, K.; Atalaya, J.; Bardin, J.C.; Barends, R.; Boixo, S.; et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* 2021, 17, 332–336. [CrossRef]

40. Lacroix, N.; Hellings, C.; Andersen, C.K.; Di Paolo, A.; Remm, A.; Lazar, S.; Krinner, S.; Norris, G.J.; Gabureac, M.; Heinsoo, J.; et al. Improving the performance of deep quantum optimization algorithms with continuous gate sets. *PRX Quantum* 2020, 1, 110304. [CrossRef]

41. Zhou, L.; Wang, S.T.; Choi, S.; Pichler, H.; Lukin, M.D. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Phys. Rev. X* 2020, 10, 021067. [CrossRef]

42. Hadfield, S.; Wang, Z.; O’Gorman, B.; Rieffel, E.G.; Venturelli, D.; Biswas, R. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* 2019, 12, 34. [CrossRef]

43. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* 1997, 1, 67–82. [CrossRef]

44. Poland, K.; Beer, K.; Osborne, T.J. No free lunch for quantum machine learning. *arXiv* 2020, arXiv:2003.14103.

45. Gard, B.T.; Zhu, L.; Barron, G.S.; Mayhall, N.J.; Economou, S.E.; Barnes, E. Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *npj Quantum Inf.* 2020, 6, 10. [CrossRef]

46. Ganzhorn, M.; Egger, D.J.; Barkoutsos, P.; Ollitrault, P.; Salis, G.; Moll, N.; Roth, M.; Fuhrer, A.; Mueller, P.; Woerner, S.; et al. Gate-efficient simulation of molecular eigenstates on a quantum computer. *Phys. Rev. Appl.* 2019, 11, 044092. [CrossRef]

47. Choquette, A.; Di Paolo, A.; Barkoutsos, P.K.; Sénéchal, D.; Tavernelli, I.; Blais, A. Quantum-optimal-control-inspired ansatz for variational quantum algorithms. *Phys. Rev. Res.* 2021, 3, 020920. [CrossRef]
79. Govia, L.; Poole, C.; Saffman, M.; Krovi, H. Freedom of mixer rotation-axis improves performance in the quantum approximate optimization algorithm. *arXiv* 2021, arXiv:2107.13129. [CrossRef]

80. Farhi, E.; Goldstone, J.; Gutmann, S.; Neven, H. Quantum algorithms for fixed qubit architectures. *arXiv* 2017, arXiv:1703.06199.

81. Bapat, A.; Jordan, S. Bang-bang control as a design principle for classical and quantum optimization algorithms. *arXiv* 2018, arXiv:1812.02746. [CrossRef]

82. Yu, Y.; Cao, C.; Dewey, C.; Wang, X.B.; Shannon, N.; Joynt, R. Quantum Approximate Optimization Algorithm with Adaptive Bias Fields. *arXiv* 2021, arXiv:2105.11946. [CrossRef]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.