

Distribution-adaptive dynamic shot optimization for variational quantum algorithms

Youngmin Kim ¹, Enhyeok Jang ¹, Hyungseok Kim ¹, Seungwoo Choi ¹, Changheon Lee ¹, Donghwi Kim ²,
Woomin Kyoung ², Kyujin Shin ^{2,*}, and Won Woo Ro ^{1,†}

¹*School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Republic of Korea*

²*Materials Research & Engineering Center, R&D Division, Hyundai Motor Company, Uiwang 16082, Republic of Korea*



(Received 29 December 2024; accepted 25 September 2025; published 5 December 2025)

Variational quantum algorithms (VQAs) have attracted remarkable interest over the past few years because of their potential computational advantages on near-term quantum devices. They leverage a hybrid approach that integrates classical and quantum computing resources to solve high-dimensional problems that are challenging for classical approaches alone. In the training process of variational circuits, constructing an accurate probability distribution for each epoch is not always necessary, creating opportunities to reduce computational costs through shot reduction. However, existing shot allocation methods that capitalize on this potential often lack adaptive feedback or are tied to specific classical optimizers, which limits their applicability to common VQAs and broader optimization techniques. Our observations indicate that the information entropy of a quantum circuit's output distribution exhibits an approximately exponential relationship with the number of shots needed to achieve a target Hellinger distance. In this work, we propose a distribution-adaptive dynamic shot (DDS) framework that efficiently adjusts the number of shots per iteration in VQAs using the entropy distribution from the prior training epoch. Our results demonstrate that the DDS framework sustains inference accuracy while achieving an $\sim 50\%$ reduction in average shot count compared to fixed-shot training, and $\sim 60\%$ higher accuracy than recently proposed tiered shot allocation methods. Furthermore, in noisy simulations that reflect the error rates of actual IBM quantum systems, DDS achieves an $\sim 30\%$ reduction in the total number of shots compared to the fixed-shot method with minimal degradation in accuracy and offers $\sim 70\%$ higher computational accuracy than tiered shot allocation methods.

DOI: [10.1103/vwhk-22b8](https://doi.org/10.1103/vwhk-22b8)

I. INTRODUCTION

Quantum computing represents a computational paradigm, offering the potential for enhanced computational capabilities beyond the reach of classical computers [1–6]. Variational quantum algorithms (VQAs) [7–11] are designed to leverage these advantages on noisy intermediate-scale quantum (NISQ) [12] devices available in the current era. Unlike classical counterparts, quantum computation requires multiple shots to approximate the expected probability distribution of a quantum state, as results cannot be fully observed in a single measurement. While increasing the number of shots can improve the statistical accuracy of results, it also prolongs execution time. Therefore, efficient optimization strategies for shot allocation are crucial to minimize overall computational overhead while maintaining training quality, particularly in iterative VQA training [13], where circuit evaluations are performed at each training epoch.

Previous research has examined various approaches to reducing the number of shots required to implement quantum circuits without compromising performance. One approach involves improving the optimizer used in the VQA process [14–16] by focusing on shot allocation based on specific Hamiltonian terms [17–19] and rearranging these terms to optimize shot distribution [20–22]. Additionally, variance-minimization-based shot assignment methods [18,22,24–28] have been explored to reduce measurement variance. However, these methods often introduce computational overhead, as they require a detailed analysis of Hamiltonian terms or the quantum circuit itself for predictive adjustments. For instance, SGLBO (stochastic gradient line Bayesian optimization) [23] integrates stochastic gradient descent with Bayesian optimization and incorporates an adaptive shot strategy based on gradient norm estimation. While effective in reducing shot costs and improving noise robustness, this method relies on accurate gradient estimation, which can introduce additional circuit executions and computational complexity. There is potential to efficiently allocate shots in order to minimize total execution time without incurring additional computational costs [28–30].

Recently, the tiered shot allocation method [31] has been proposed to enable dynamic shot allocation throughout the optimization process. This method deterministically reduces the overall number of measurement shots by statically defining a reduced shot count. However, reducing the number of shots without considering the specific VQA

*Contact author: shinkj@hyundai.com

†Contact author: wro@yonsei.ac.kr

circuit or variations in the quantum state during training does not guarantee the quality of training parameters, which limits its applicability. To address this limitation, a feedback mechanism is essential to dynamically adjust the shot count for each iteration, ensuring efficient resource utilization while maintaining the consistency and reliability of the VQA.

We have observed a correlation between the entropy of the probability distribution and the number of measurement shots required to guarantee a specific Hellinger distance. This finding suggests that the entropy of the probability distribution can serve as a practical feedback metric for adaptively determining the optimal number of measurement shots. During VQA iterations, probability distributions evolve, ranging from evenly distributed states with high entropy to those concentrated around a few high-probability states with low entropy. Our observations reveal a clear correlation between the characteristics of these distributions and the number of shots required to obtain reliable results. Specifically, evenly distributed probability distributions require a larger number of shots for accurate measurements, whereas distributions concentrated on a small subset of quantum states can be captured with fewer shots.

In this paper, we present the distribution-adaptive dynamic shot (DDS) approach, designed to optimize shot allocation and reduce execution time on quantum computers. Our method dynamically determines the optimal number of shots based on the specific quantum circuit and its probability distribution, leading to reduced shot usage with minimal impact on the overall programmability of the VQA process. By adjusting the shot count based on the entropy of the probability distribution, DDS effectively decreases the overall number of measurement shots without compromising fidelity. In each iteration, DDS calculates the entropy of the probability distribution and uses this information to determine the number of shots for the subsequent iteration.

Our approach dynamically adjusts the shot count for the quantum circuit of VQAs by exploiting the entropy of probability distribution obtained from the previous iteration. Unlike existing tiered shot allocation methods that rely on fixed heuristics, the DDS scheme leverages the gradual evolution of the probability distribution throughout the VQA process. As VQA gradually refines its parameters over multiple iterations, DDS maintains simulation accuracy while enhancing implementation efficiency. By dynamically adjusting the shot count for each iteration based on the estimated probability distribution, the DDS approach reduces the overall number of shots required to optimize the variational parameters. This reduction in shot count per iteration can lead to a substantial decrease in total execution time, further enhancing the overall efficiency of VQA implementations on quantum hardware.

Figure 1 illustrates the process of determining the number of shots associated with each iteration throughout the VQA process. We observed that as the VQA training progresses, the information entropy of the probability distribution generally decreases. During iterations with low-entropy distributions, trainability can be maintained by measuring with fewer shots while preserving accuracy. To evaluate the precision of VQA execution under adaptive shot allocation, we used the approximation ratio gap (ARG) [32]. The ARG quantifies the

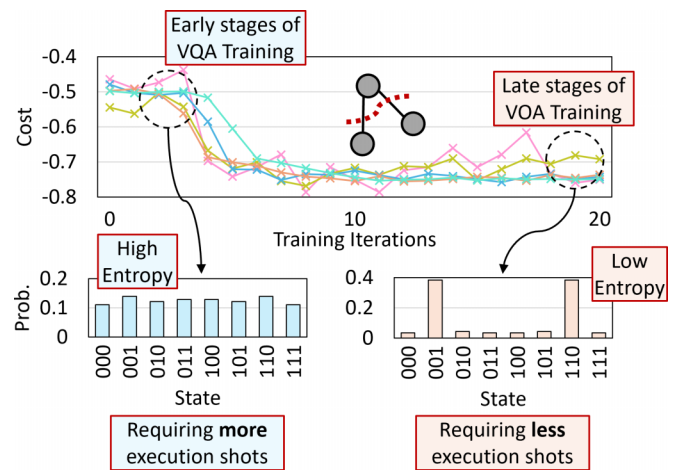


FIG. 1. The number of shots required to accurately estimate the probability distribution in a VQA depends on the information entropy of the quantum states at each iteration. In the early stages, the quantum state exhibits high entropy due to a relatively uniform probability distribution over computational basis states. As the VQA converges and the target state becomes dominant, the entropy decreases. By allocating a larger number of shots during high-entropy iterations and reducing the number of shots when the entropy is low, overall shot usage can be optimized efficiently.

difference between the final cost and the ideal cost, with a lower ARG indicating higher accuracy due to closer alignment with the ideal.

The DDS approach achieves a significant reduction in the total number of shots, with an average decrease of 50.69% compared to the baseline, even as the ARG shows a slight increase of 6.03% on average. Although the total number of iterations remains similar, DDS notably reduces the number of shots required per iteration, contributing to an overall decrease in computational resources. When compared to the existing tiered shot allocation method, DDS requires a higher number of shots; however, it provides a substantial improvement in accuracy, achieving a 63.34% increase in performance. Under noisy conditions, DDS achieves an average 31.15% reduction in the total number of shots while maintaining a slight increase in ARG compared to the baseline. Additionally, DDS provides a 68.91% improvement in ARG compared to the tiered-shot approach, despite requiring more shots. These results highlight the effectiveness of the DDS approach in balancing shot allocation and measurement precision during VQA training.

II. ENTROPY IN QUANTUM CIRCUIT IMPLEMENTATIONS

In this section, we present our observations regarding the probability distributions and the corresponding number of shots required for quantum measurements in VQAs. Specifically, we analyze the number of shots needed to accurately capture the probability distribution of the final quantum state and the implications for the overall VQA simulation process. Additionally, we discuss the relationship between the entropy of the probability distribution and the required number of shots, as well as how this relationship affects the efficiency of VQA implementations.

A. Information entropy of probability distribution

Information entropy, also referred to as *Shannon entropy* [33], quantifies the uncertainty associated with a random variable. Information about events that are less likely to occur is considered more informative than information about events that happen frequently. The greater the uncertainty, the more information can be obtained, and thus entropy is defined as the mathematical expected value of the information across all possible events. Let X be a random variable, which is a function of the probabilities in the sample space. For probability distributions $p_1, \dots, p_n$, the information entropy can be expressed as

$$H(X) \equiv H(p_1, \dots, p_n) \equiv - \sum_x p_x \log_2 p_x. \quad (1)$$

The entropy of a probability distribution is high for a uniform distribution and low for distributions that are unevenly distributed. In the context of a quantum circuit, the entropy of the Walsh-Hadamard transform applied to the initial $|0\rangle$ state is greater than the entropy of the GHZ (Greenberger-Horne-Zeilinger) state [34], which exhibits a more concentrated distribution with less uncertainty. Appendix C presents the entropy of the random number generation circuit, which is uniformly distributed and exhibits the maximum entropy among circuits with the same number of qubits. For simplicity, we will refer to information entropy simply as entropy throughout this paper.

B. Required shots for achieving accurate probability distributions

Probability distributions with more information have higher entropy; in other words, the higher the uncertainty, the higher the entropy. A uniform distribution is more informative than an unevenly distributed probability distribution that has a small number of solution states. Not only the evenness of the probability distribution but also the scalability of the sample space influences the entropy of a distribution. The sample space increases exponentially as the qubit count increases, requiring more shots to maintain accuracy. Figure 2 shows the Hellinger distance [35] of the distribution generated by a random number generation quantum circuit [36].

To assess the accuracy of the probability distribution of the final quantum state based on the number of measurement shots, we employ the Hellinger distance [35]. The Hellinger distance is a metric used to quantify the similarity between two probability distributions. A lower Hellinger distance indicates a closer match between the compared distributions, reflecting greater alignment in probability values and higher accuracy. For two probability distributions $P = (p_1, \dots, p_n)$ and $Q = (q_1, \dots, q_n)$, the Hellinger distance between P and Q is defined as

$$H_d(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_i (\sqrt{p_i} - \sqrt{q_i})^2}. \quad (2)$$

For a fixed number of qubits, an increase in the number of measurement shots leads to a reduction in the Hellinger distance. Moreover, as the qubit count increases, the number of shots required to achieve a low Hellinger distance also

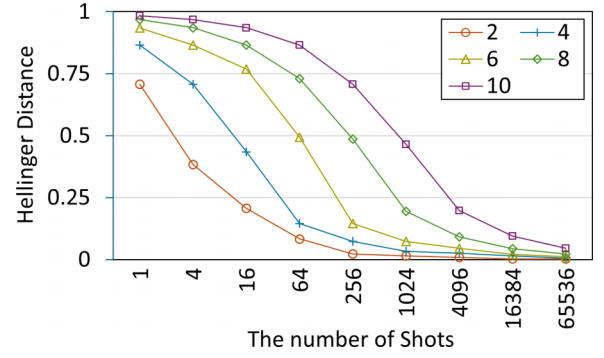


FIG. 2. The Hellinger distance between the ideal and measured probability distributions for random number generation circuits [36] with varying qubit counts, as a function of the number of shots. For each qubit count, circuits were executed with the number of shots increasing exponentially in powers of 4. The ideal distribution is defined as a uniform distribution over all possible quantum states, while the measured distribution is constructed from sampled quantum states. As the number of shots increases, the Hellinger distance decreases, indicating that the measured distribution converges toward the ideal uniform distribution.

increases. In circuits with higher qubit counts, a sufficiently large number of measurement shots is essential for accurately determining the quantum state. Additionally, we observe that quantum states with more informative probability distributions generally necessitate a greater number of measurement shots.

C. Optimizing shot requirements under a Hellinger distance constraint

The outcome we want from quantum circuit execution is the probability distribution of basis states at the final state. The distribution is composed of multiple collections of basis states, which are determined for each measurement. Increasing the shot count can achieve high accuracy for any distribution, but it would take a long time to run multiple measurements. A uniform distribution represents the state with the highest entropy and the largest number of correct answers, requiring a relatively high number of measurements for accurate representation. Conversely, when the number of correct answers is smaller, the distribution can be guaranteed with a comparable level of accuracy using fewer measurements.

To maintain the Hellinger distance while reducing the number of shots, it is essential to determine the minimum number of shots required for each probability distribution. Figure 3 illustrates the relationship between entropy and the number of shots needed to approximate the ideal probability distribution that the quantum state is theoretically expected to output. The x axis represents the entropy of the output distribution, while the y axis shows the number of shots required to achieve a Hellinger distance of 0.05. The results reveal an exponential relationship between the entropy of the target distribution and the number of shots necessary to achieve a specified level of accuracy. Therefore, by referencing the entropy of a distribution in VQA iterations, we can estimate the required shot count to ensure accurate measurement.

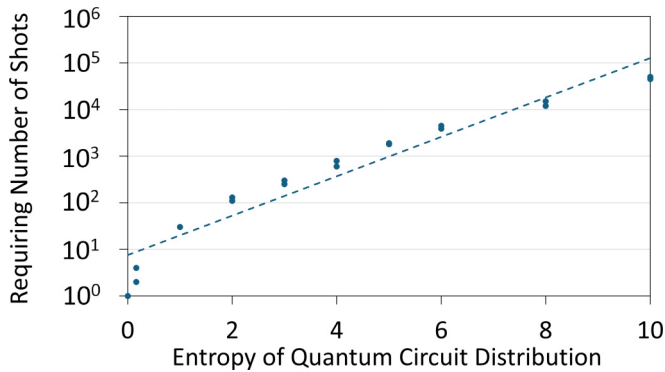


FIG. 3. Required number of shots to achieve a Hellinger distance below 0.05 between the measured and ideal probability distributions, evaluated for various quantum circuits with varying output distribution entropies. Each circuit is positioned according to the entropy of its probability distribution, and the corresponding shot count necessary to satisfy the accuracy threshold is shown on a logarithmic scale. As the entropy increases, the required number of shots grows, illustrating an exponential relationship between entropy and sampling cost.

III. DISTRIBUTION-ADAPTIVE DYNAMIC SHOT

In this section, we introduce an optimization method based on the distribution of quantum states that dynamically adjusts the shot count per iteration in VQAs to reduce the total number of measurements. We will discuss previous observations in the context of VQAs and show how these insights are leveraged in the DDS technique. Subsequently, we will provide a detailed explanation of the DDS approach.

A. Evolution of probability distributions in VQA

In VQA, the quantum circuit execution and classical optimization of circuit parameters are performed iteratively. For each iteration, measurements are taken based on the specified number of shots, with the resulting probability distribution derived from the measurement outcomes. Each measurement corresponds to a computational basis state, and the proportion of results that align with a specific basis relative to the total number of measurements defines the probability of that basis. As the quantum circuit parameters are updated, the probability distribution dynamically evolves with each iteration. Therefore, the probability distribution changes throughout the VQA process, with these variations being influenced by the algorithms or ansatz employed. A more detailed overview of VQA can be found in Appendix B.

For most VQA training or other quantum algorithms, the initial circuit preparation is typically performed by applying a Hadamard gate to the $|0\rangle$ state multiple times. At the beginning of the VQA training iterations shown in Fig. 1, the probability distributions are likely to be evenly distributed. For the quantum approximate optimization algorithm (QAOA) max-cut problem, the final state contains information about the edges for the max cut, and the number of correct states varies depending on the QAOA graph. In cases where edges are densely connected, the probability distribution will be evenly distributed, whereas in cases where edges are concentrated on one vertex or in a graph with low density, the

probability distribution will be concentrated on a few correct states. At the beginning of the training iterations, a large number of shots will be required, while in later training iterations, when the distribution becomes unevenly distributed, only a small number of shots will be needed.

B. Mechanics of distribution-adaptive shot allocation

In VQA iterations, DDS evaluates the probability distributions and dynamically adjusts the number of measurement shots in each iteration to effectively minimize the total shot count. The probability distribution at each iteration is incrementally modified by the classical optimizer, reflecting changes in the quantum circuit parameters. Since consecutive iterations typically exhibit similar distributions, DDS leverages the entropy of the probability distribution from the previous iteration to determine the required shot count for the current iteration, ensuring accurate and reliable operation of the VQA process.

For each iteration, measurements are performed based on the specified shot count, and the resulting probability distribution is derived from the measurement outcomes. These outcomes, corresponding to measurement bases (commonly computational bases), are stored as counts. The probability distribution is then constructed from these counts, enabling the entropy H to be calculated in the estimator. This entropy serves as a key metric for effectively estimating the shot count required in subsequent iterations. For each measurement basis A_j , the entropy of the probability distribution is defined as

$$H = - \sum_j P(A_j) \log_2 P(A_j). \quad (3)$$

The entropy H is stored and utilized in the subsequent iteration to estimate the shot count. After the estimation process for the following iteration is completed, the calculated entropy is updated accordingly. This entropy calculation is repeated for each iteration until the parameters converge, as determined by the classical optimizer.

Determining the shot count for initializing the estimator is a crucial component of the DDS method. If no specific shot count is provided, the estimator defaults to 1024 shots. DDS dynamically adjusts the number of shots for each iteration using the entropy value stored from the previous iteration. Based on the observation that the required number of shots is exponentially proportional to the entropy, as discussed in Sec. II, the shot count S is calculated from the entropy H as follows:

$$S = k \times 2^H, \quad (4)$$

where k is a constant. This exponential relationship is further supported by theoretical analysis provided in Appendix D. Specifically, a bound is derived on the number of samples necessary to achieve a given Hellinger distance between the true and empirical distributions. The result shows that the required number of shots grows proportionally to the exponential of the Shannon entropy.

The value of k depends on the type of algorithm or ansatz used. The number of qubits also influences the selection of k : Large circuits typically require more shots, as noted in Sec. II. For circuits that are highly complex, where a significant

ALGORITHM 1. VQA with distribution-adaptive dynamic shot allocation.

Input: ansatz, initial_parameter
Output: optimal_parameter

1. parameter $\leftarrow$ initial_parameter
2. prev_entropy $\leftarrow$ 10 // initialize entropy
3. converged $\leftarrow$ False
4. **while** not converged **do**
5. entropy $\leftarrow$ prev_entropy
6. shots $\leftarrow$ constant $\times 2^{\text{entropy}}$
7. result $\leftarrow$ Estimator(ansatz, parameter, shots)
8. prev_entropy $\leftarrow$ ComputeEntropy(result)
9. cost $\leftarrow$ expectation_value(result)
10. parameter, converged $\leftarrow$ cobyala(ansatz, result)

number of shots is needed to achieve a sufficiently low Hellinger distance, k should be set to a larger value. Conversely, for circuits that require fewer shots, setting k to a smaller value is more appropriate to minimize the overall shot count.

Algorithm 1 outlines the process of dynamically adjusting the number of shots during the VQA process. At each iteration, the entropy of the probability distribution obtained from the measurement results in the previous iteration is calculated. This entropy value is then used to determine the number of shots required for the current iteration. By dynamically adapting the shot count based on the entropy, the algorithm effectively reduces the number of shots per iteration while maintaining the accuracy of the resulting probability distribution.

IV. EVALUATION

A. Evaluation methodology

In this section, we compare the proposed DDS method with the standard fixed-shot approach. Furthermore, we evaluate its performance relative to existing shot reduction strategies: the linear function-based and step function-based measurement shot allocation methods.

1. Benchmarks and backends

To evaluate the performance of the DDS approach, quantum circuits for both the QAOA and the variational quantum eigensolver (VQE) were constructed. For QAOA, we selected target graphs commonly used in network science, specifically power-law (PL), Barabási-Albert (BA), Watts-Strogatz (WS), and Sherrington-Kirkpatrick (SK) models. Each graph model exhibits a unique connection density between nodes. For all target graphs, the QAOA circuit was configured with 20 layers, and models with varying qubit counts were tested.

For the VQE experiments, we simulated the ground-state energies of H_2 , LiH, and BeH_2 in the STO-3G basis set. The VQE circuits were configured to meet the qubit requirements for each molecule. We used the unitary coupled cluster with single and double excitations (UCCSD) ansatz for VQE simulations, which requires 4, 12, and 14 qubits for H_2 , LiH, and BeH_2 , respectively. This approach allows us to map spin orbitals to individual qubits, and we use all orbitals without

freezing the core orbitals. The VQE calculations using the UCCSD ansatz were performed with bond lengths of 0.73 Å for H_2 , 1.57 Å for LiH, and 3.00 Å for BeH_2 .

All VQA benchmark circuits were simulated on the qiskit-aer state vector simulator [37], where dynamic shot allocation methods, including DDS, implemented with few modifications. For experiments conducted under noisy conditions, the qiskit-aer qasm simulator was employed. This noise simulation environments reflect the 1Q and 2Q gate error rates of ibm_marrakesh with Heron r2 and ibm_yonsei with Eagle r3, calibrated on December 13, 2024. In both QAOA and VQE experiments, the classical optimizer COBYLA (constrained optimization by linear approximation) was applied, given its widespread application in variational quantum algorithms.

2. Scenarios

The fixed-shot method uses 1024 shots as its configuration, aligning with the default setting in many existing quantum computing systems. The linear shot function [31], starting from 1000 shots (close to the default of 1024), gradually reduces the shot count over training iterations. This function is defined as $S_i = \max(20, S_{\text{begin}} - l \cdot i)$, where S_i represents the shot count at the i th iteration and l is the slope parameter that controls the rate of reduction. In this study, we set $S_{\text{begin}} = 1000$ and $l = 10$. To prevent impractically low shot counts, we enforce a minimum threshold of 20 shots. Similarly, the step function [31] reduces the shot count in discrete increments and is defined as $S_i = \max(20, S_{\text{begin}} - 10 \cdot l \cdot \lfloor \frac{i}{10} \rfloor)$, where $\lfloor \cdot \rfloor$ denotes the floor function. To ensure consistency with the linear shot function, we use $S_{\text{begin}} = 1000$ and set the slope parameter to $l = 10$. These values for S_{begin} and the slope parameter were applied uniformly across all VQA applications during the evaluation.

To investigate the effect of DDS across different graph complexities, we fixed the parameter k for each qubit count. In our QAOA experiments, k was adjusted based on the number of qubits, which corresponds to the number of nodes in the graph, since larger circuits with more qubits require more shots to maintain accuracy. As shown in Appendix C, the entropy H of an n -qubit random number generation circuit is equal to n . To ensure that the total number of shots, given by $k \times 2^H$, increases by a factor of two as the qubit count increases from 4 to 8 to 12, we set $k = 64, 8$, and 2, respectively. For the VQE scenarios, the value of k was fixed to 8 for all experiments.

3. Metrics

To measure the fidelity of VQA circuits, ARG [32] was used, defined as follows:

$$\text{ARG} = 100 \times \left| \frac{E_{\text{ideal}} - E_{\text{real}}}{E_{\text{ideal}}} \right|. \quad (5)$$

E_{ideal} is the ideal expectation value of the quantum state and E_{real} is the calculated expectation value. When the fidelity is high, the measured expectation value is closer to the ideal value, resulting in a smaller ARG value, which indicates a more accurate approximation. The smaller the ARG value, the better the performance.

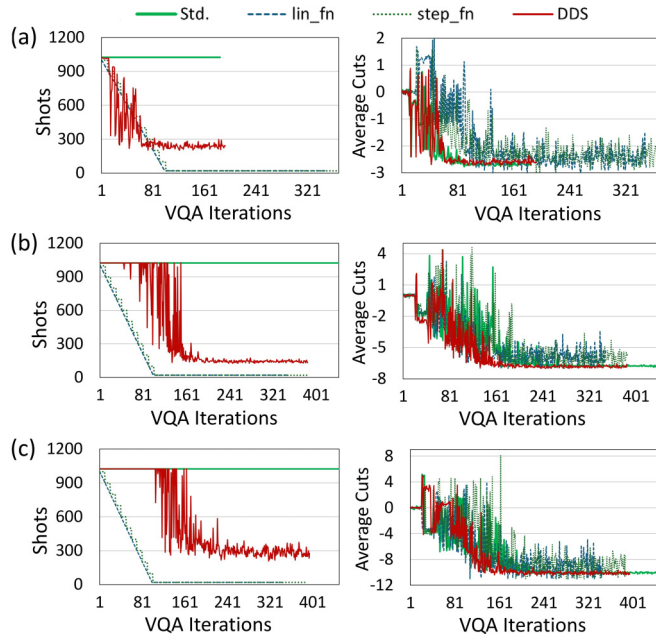


FIG. 4. The number of shots and the average cuts per iteration during the PL model QAOA training for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, linear function, step function, and DDS shot allocation methods. In each iteration, the probability distribution is constructed from quantum states sampled according to the assigned number of shots, and its expectation value corresponds to the average cuts.

S_{tot} is the total number of measurements in the VQA process, and S_{avg} is the average number of measurements taken per iteration. For the total number of iterations I_{tot} , the total number of shots is calculated as follows:

$$S_{\text{tot}} = S_{\text{avg}} \times I_{\text{tot}}. \quad (6)$$

B. Results and analysis

1. Evaluation on scalable VQAs

Dynamic shot allocation was applied to both QAOA and VQE models for several molecules. We tested four QAOA models and three VQE models, running each with the standard fixed-shot method of 1024 shots, as well as with two existing shot reduction methods: the linear function and step function-based shot allocation approaches [31]. The standard method, a conventional approach, is labeled as “Std.” in the graphs and tables. As described earlier, the linear and step functions begin with 1000 shots and gradually decrease the shot count over training. These methods are indicated as “lin_fn” and “step_fn” in the tables and graphs, respectively. For DDS, the implementation is the same as the approach outlined in the previous section, and an upper bound of 1024 shots was applied to allow for direct comparison with the standard method throughout the experiments.

Figure 4 illustrates the number of shots and the average cuts (i.e., the expectation value) per iteration for QAOA PL models with 4, 8, and 12 qubits, using four shot allocation methods: standard, linear_fn, step_fn, and DDS. Both linear_fn and step_fn follow a consistent downward trend until they reach the minimum shot limit of 20, maintaining that

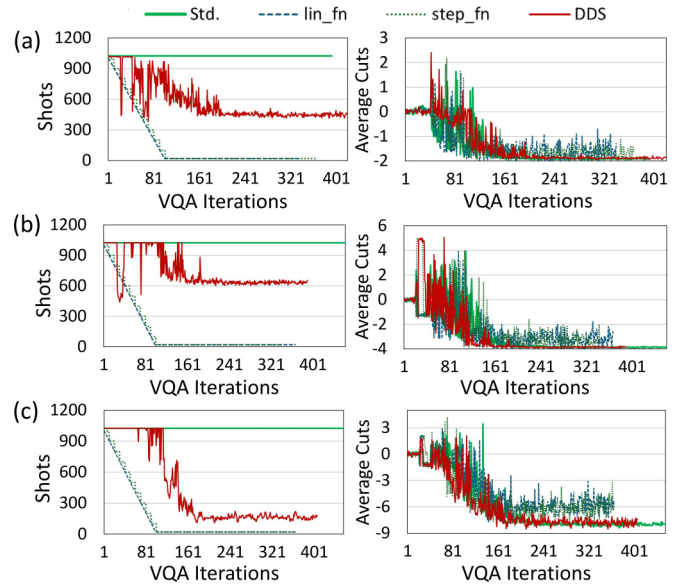


FIG. 5. The number of shots and the average cuts per iteration during the BA model QAOA training for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, linear function, step function, and DDS shot allocation methods.

number until the end. In contrast, DDS dynamically allocates shots based on the probability distribution of the results from the previous iteration. As a result, DDS initially requires a high number of shots, which gradually decreases over the course of iterations.

As the QAOA process advances, the quantum state approaches the solution, becoming more concentrated on certain states. Consequently, entropy decreases, leading to a reduced shot requirement in later iterations. When examining the cost (expectation value) across iterations, DDS performs comparably to the standard method. However, for linear_fn and step_fn, the cost does not converge consistently, displaying oscillations. While the total shot count may be lower for linear_fn and step_fn, the reduced number of shots hinders their ability to reliably converge to a low expectation value.

Figures 5, 6, and 7 display the shot count and the average cuts per iteration for the QAOA BA, WS, and SK models, respectively, under each shot allocation method. Similar to the PL model in Fig. 4, DDS begins with a high shot count and decreases it in later iterations. In terms of cost, DDS remains close to the performance of the standard method, whereas linear_fn and step_fn show fluctuations and fail to converge effectively to low expectation values.

For the SK model, due to its higher graph complexity, the required shot count per iteration is greater than for other models, and the cost also exhibits more significant fluctuations. In linear_fn and step_fn, cost fluctuations are more pronounced initially, but for models with lower graph complexity, DDS demonstrates less variability in cost. Given that the primary objective in QAOA is to achieve low expectation values, linear_fn and step_fn may face difficulties in achieving stable and consistent performance. In contrast, DDS effectively produces results comparable to the standard method in a stable manner.

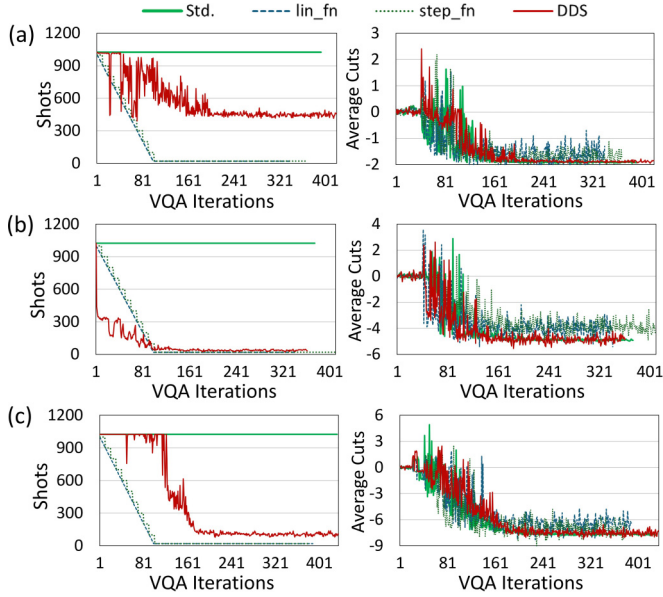


FIG. 6. The number of shots and the average cuts per iteration during the WS model QAOA training for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, linear function, step function, and DDS shot allocation methods.

Table II presents the average shot count (S_{avg}) for each QAOA model based on qubit count, the total number of iterations (*Iterations*) over which QAOA is simulated, and the *Average Cuts*, which represents the result of an additional execution following the last iteration. Due to variations in the shot allocation method, the number of shots used in the final iteration may differ. To ensure a fair comparison, this final execution was conducted with a fixed-shot count of 1024 for all methods.

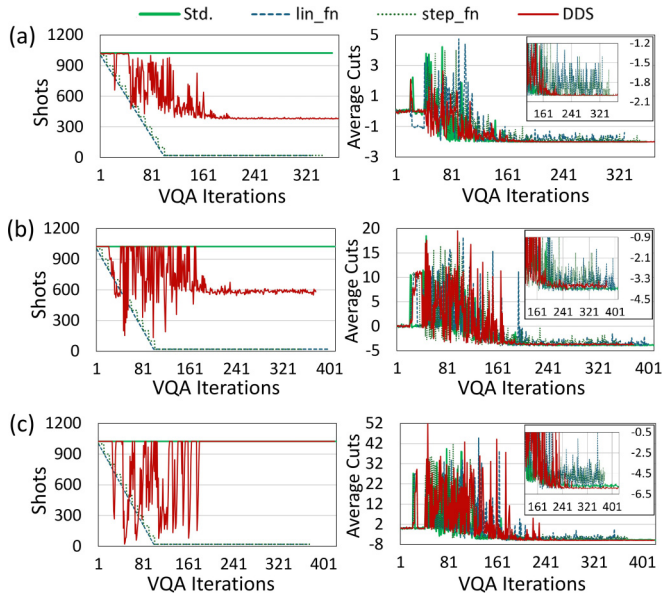


FIG. 7. The number of shots and the average cuts per iteration during the SK model QAOA training for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, linear function, step function, and DDS shot allocation methods.

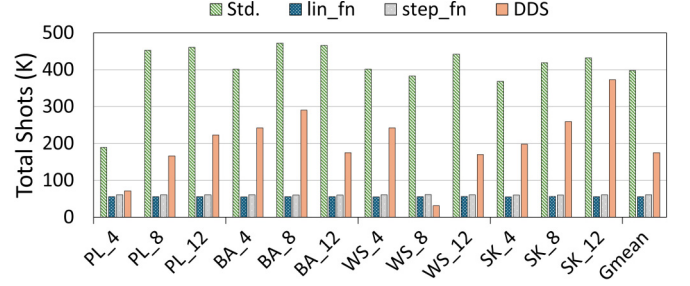


FIG. 8. Total shot counts (in thousands) required for the training of various QAOA models, including PL, BA, WS, and SK, across different qubit counts and model configurations. Each bar represents a shot allocation method—standard, linear function, step function, or DDS—and the total shots are computed as the sum of the number of shots used in all iterations during training. The final group, labeled Gmean, indicates the geometric mean of total shot counts across all qubit sizes and QAOA models.

For QAOA, the total number of iterations shows no substantial difference across methods. However, the average shot count per iteration is noticeably lower for DDS compared to the standard method. Figure 8 illustrates the total number of shots for each shot allocation method, indicating that DDS, on average, requires 50.69% fewer shots than the standard method. Meanwhile, the linear_fn and step_fn methods achieve greater reductions, with the total shot count decreasing by 85.49% and 84.24%, respectively, compared to the standard method.

To assess the fidelity of QAOA, the ARG metric was examined. As shown in Fig. 9, the ARG values for linear_fn and step_fn are substantially higher than the standard, with average increases of 241.18% and 230.73%, respectively. In contrast, the ARG for DDS remains close to the standard, showing only a 6.03% average difference. Thus, DDS achieves a substantial reduction in shot count with minimal fidelity loss. Although the linear_fn and step_fn methods reduce shot counts considerably, they incur significant fidelity losses, which may limit their suitability for accurate QAOA

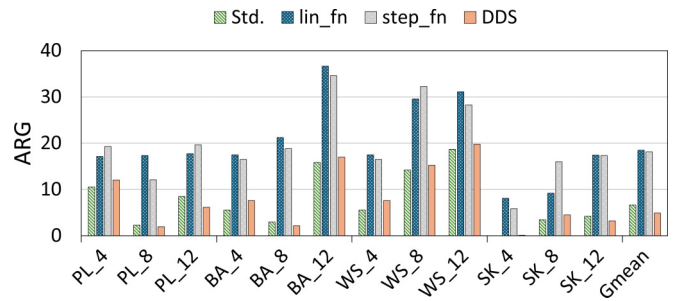


FIG. 9. The ARG values for the training of various QAOA models, including PL, BA, WS, and SK, across different qubit counts and model configurations. Each bar represents a shot allocation method—standard, linear function, step function, or DDS—and the ARG is computed by comparing the expectation value of the final probability distribution obtained through training with that of the ideal distribution. The final group, labeled Gmean, indicates the geometric mean of ARG values across all qubit sizes and QAOA models.

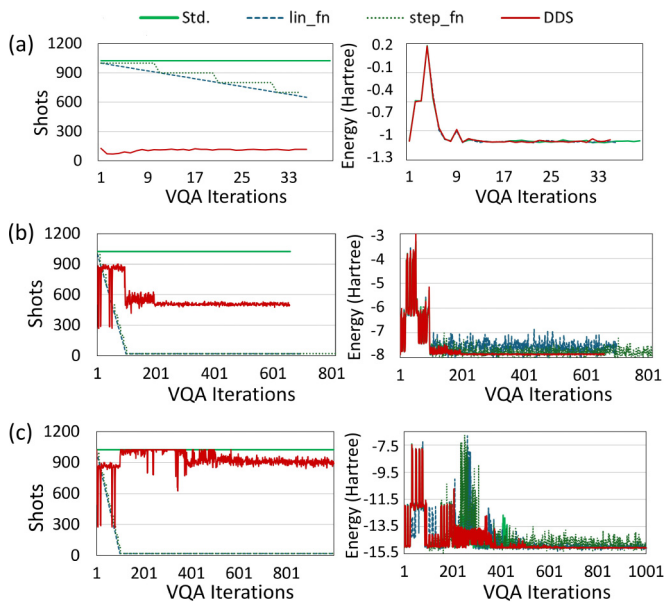


FIG. 10. The number of shots and the energy of each iteration for the VQE training process of (a) H₂, (b) LiH, and (c) BeH₂, using standard, linear function, step function, and DDS shot allocation methods. In each iteration, the probability distribution is constructed from quantum states sampled according to the assigned number of shots, and its expectation value corresponds to the energy.

simulation. Compared to these tiered shot allocation methods, DDS demonstrates a 63.34% improvement in accuracy.

Figure 10 illustrates the shot count and the energy per iteration for VQE simulations on H₂, LiH, and BeH₂ molecules, using the standard, linear_fn, step_fn, and DDS shot allocation methods. In these simulations, the expectation value computed through the VQE circuit represents the electronic ground-state energy, with the total ground-state energy obtained by adding the nuclear repulsion energy. For all three molecular structures, linear_fn and step_fn exhibit a steady initial reduction in shot count. However, as molecular complexity and qubit requirements increase, more iterations are needed for convergence. Consequently, when the shot count reaches the minimum too early, the cost fails to converge and instead fluctuates. In the case of H₂, the entropy of the distribution is low, allowing DDS to require fewer shots while maintaining a cost almost identical to the standard. For both LiH and BeH₂, the DDS method maintains a cost trajectory similar to the standard with minimal oscillations, demonstrating stable convergence.

Table III summarizes the average shot count per iteration, the total number of iterations required for the VQE process, and the final total ground-state energy after convergence. To ensure consistent comparison, the energy was calculated after one additional execution with 1024 shots across all shot allocation methods, same as the QAOA setup. For the VQE, the total number of iterations shows minimal difference across methods. However, the average number of shots per iteration varies significantly. As the complexity of the molecular ansatz increases, so does the number of required iterations. Consequently, while the average shot count per iteration for linear_fn and step_fn decreases over time, the average shot

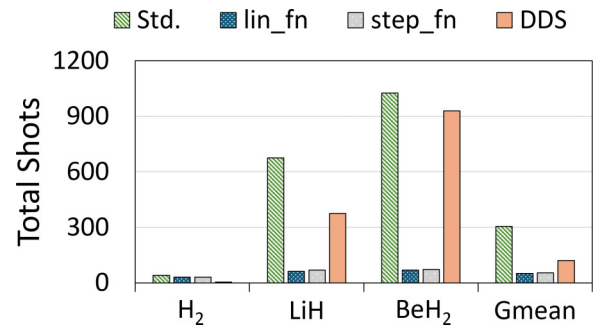


FIG. 11. Total shot counts (in thousands) required for the training of various molecules in VQE simulations, including H₂, LiH, and BeH₂. Each bar represents a shot allocation method—standard, linear function, step function, or DDS—and the total shots are computed as the sum of the number of shots used in all iterations during training. The final group, labeled Gmean, indicates the geometric mean of total shot counts across all molecules.

count for DDS increases as the entropy of the probability distribution grows, reflecting the increased uncertainty in the quantum state throughout the optimization process.

Figure 11 compares the total number of shots across shot allocation methods for the VQE process. As molecular complexity increases, reflected in the size and number of atoms in the molecular ansatz, linear_fn and step_fn exhibit relatively small changes in total shot count compared to other methods, with average reductions of 69.81% and 68.64%, respectively, compared to the standard. For LiH and BeH₂, shot reductions approach nearly 90%. DDS achieves an average reduction of 47.09%, with H₂ and LiH showing reductions of nearly 65%, though it still requires more shots than linear_fn and step_fn due to its dynamic convergence behavior.

In terms of ARG, as illustrated in Fig. 12, linear_fn and step_fn show increases of 172.65% and 66.79% compared to the standard, respectively. In contrast, DDS achieves a 24.39% reduction in ARG. For VQE, while linear_fn and step_fn significantly reduce shot count, they experience a notable increase in ARG, indicating reduced fidelity. DDS, however, not only decreases the number of shots but also main-

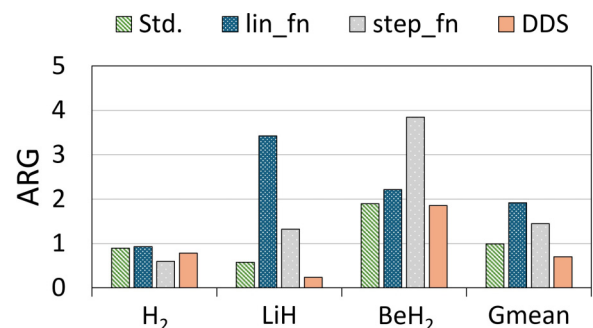


FIG. 12. The ARG values for the training of various molecules in VQE simulations, including H₂, LiH, and BeH₂. Each bar represents a shot allocation method—standard, linear function, step function, or DDS—and the ARG is computed by comparing the expectation value of the final probability distribution obtained through training with that of the ideal distribution. The final group, labeled Gmean, indicates the geometric mean of ARG values across all molecules.

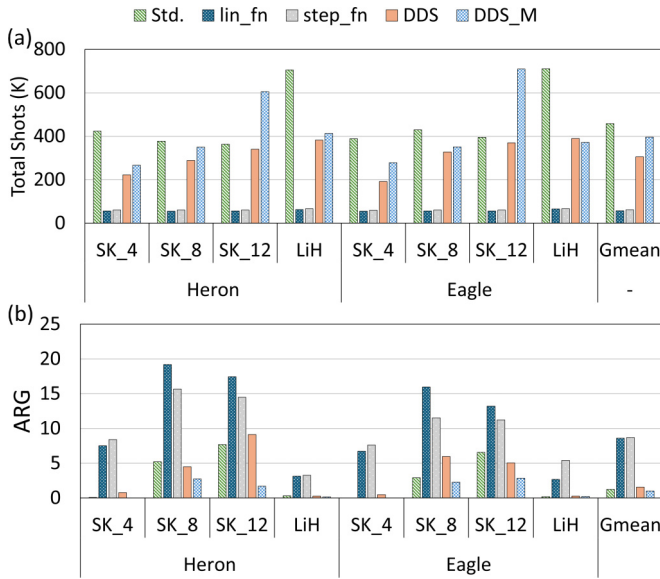


FIG. 13. (a) Total shot counts (in thousands) and (b) ARG values for the training of the SK QAOA model with 4, 8, and 12 qubits, and the LiH VQE under noisy conditions. Each bar represents a shot allocation method—standard, linear function, step function, or DDS. The final group, labeled Gmean, indicates the geometric mean of total shot counts and ARG values across all problem instances. The qiskit-aer qasm simulator was used, and the native gate set along with error information from the *ibm_marrakesh* and *ibm_yonsei* processors—based on the Heron and Eagle architectures—were employed.

tains ARG levels comparable to the standard, demonstrating its effectiveness in preserving fidelity while reducing shot count. Compared to the tiered shot allocation methods, DDS demonstrates a 37.78% improvement in accuracy. The results indicate that *linear_fn* and *step_fn* reduce shot counts but at the cost of significantly reduced fidelity. Thus, DDS is a more reliable approach for achieving precise quantum solutions.

2. Evaluation under noise condition

On actual quantum devices, experiments are typically conducted with a default configuration of 1024 shots [37], similar to quantum simulators. However, due to the presence of noise in real quantum hardware, the number of shots is sometimes increased to ensure more reliable results [39]. To simulate experiments under noisy conditions, the qiskit-aer qasm simulator was utilized, incorporating the native gate set and error information from the *ibm_marrakesh* and *ibm_yonsei* processors, which are based on the Heron and Eagle architectures. The noise model in the qasm simulator was configured to introduce depolarizing errors, accounting for both one-qubit and two-qubit gate errors.

The standard fixed-shot method and tiered shot allocation methods were used as before. Previously, the maximum shot count for DDS was set to 1024 for comparison with other shot allocation methods. However, since a larger number of shots may be required to ensure reasonable accuracy in noisy environments, DDS_M, with a maximum shot count of 100 000, was also included for comparison.

Figure 13 compares (1) the total shot counts across different shot allocation methods and (2) the ARG performance

under noisy conditions. The *linear_fn* and *step_fn* methods achieve significant reductions in total shot count, with average decreases of 87.13% and 86.17%, respectively, compared to the standard method. However, these methods exhibit substantial increases in ARG, by 599.19% and 608.02%, respectively. While *linear_fn* and *step_fn* effectively reduce shot counts even in noisy environments, they suffer from a more pronounced accuracy drop.

DDS achieves an average reduction in total shot count of 31.14%, while increasing ARG by 26.92% compared to the standard method. Although DDS requires more shots than *linear_fn* and *step_fn*, it still reduces the shot count compared to the standard approach, with ARG values 71.83% and 65.98% lower than those of *linear_fn* and *step_fn*, respectively. While DDS_M uses more shots than DDS, it demonstrates better accuracy. Specifically, DDS_M uses 4.30% fewer shots than the standard method, while reducing ARG by 18.05%. This indicates that DDS_M ensures higher accuracy without increasing the overall shot count, even when more shots are required during specific iterations of the training process. This further highlights the effectiveness of determining shot counts based on the entropy of the quantum state's probability distribution. Under noisy conditions, details regarding the number of iterations, cost, and shot usage during the training process for QAOA simulations with the SK model and VQE simulations for LiH can be found in Appendix F.

3. Evaluation with gradient-descent method

In previous sections, we demonstrated that DDS with COBYLA, a gradient-free optimization method, effectively reduces the number of shots while maintaining accuracy. However, gradient-based methods such as L-BFGS-B (Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Bound constraints) [38] depend on accurate gradient estimates and are consequently more vulnerable to shot-induced noise. In this section, we extend our investigation by performing gradient-based optimization experiments using the L-BFGS-B optimizer under both the standard and DDS shot allocation strategies.

Additionally, we include the weCANS [14] algorithm, a state-of-the-art adaptive shot allocation scheme for comparison. This algorithm is combined with the Adam optimizer, which adaptively adjusts the learning rate based on the exponential moving averages of the first- and second-order moments of the gradients. In the weCANS framework, the number of shots is determined by theoretical upper bounds derived from gradient variance and Lipschitz constants.

All gradients were estimated using the parameter-shift rule, which requires multiple circuit executions per parameter at each optimization step. To ensure consistency, a fixed number of shots was assigned to all circuit executions within a given iteration across all methods. Considering the increased sensitivity of gradient-based methods to shot noise, the standard approach was evaluated using 10 000 shots per circuit execution—an increase from the 1024 shots used in gradient-free scenarios. For DDS, the hyperparameter k was scaled up by a factor of ten, being set to $k = 640, 80, \text{ and } 20$ for 4, 8, and 12 qubits, respectively. A

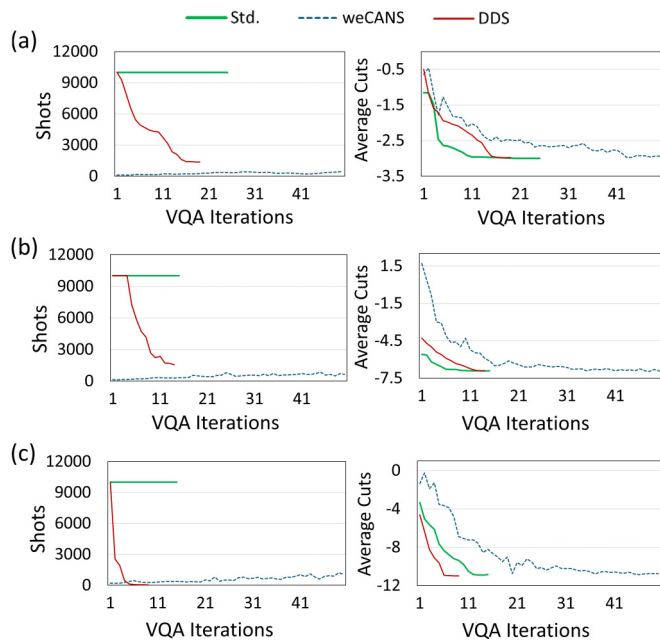


FIG. 14. The number of shots and the average cuts per iteration during the PL model QAOA training for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, DDS, and weCANS shot allocation methods. All methods are based on gradient-based optimization. The probability distribution is constructed from the sampled quantum states in each iteration, and its expectation value corresponds to the average cuts.

maximum shot cap of 10 000 shots per execution was also imposed.

Figure 14 presents the shot usage and average cut value per iteration for the PL QAOA model with 4, 8, and 12 qubits. In the DDS method, the optimization was initiated with the maximum number of shots, which gradually decreased as the training progressed. In contrast, weCANS started with a low shot count that increased over time. These trends reflect the differing philosophies of the two approaches: DDS emphasizes early-stage accuracy followed by shot efficiency, using entropy-based metrics to detect distributional stability, whereas weCANS begins conservatively and increases allocation based on theoretical uncertainty bounds. In all configurations, both the standard and DDS methods reached convergence in fewer iterations than weCANS, while all three methods exhibited consistent improvements in the average cut value throughout the optimization.

Figures 21, 22, and 23 extend the analysis to the BA, WS, and SK models. The DDS method consistently reduced shot consumption while maintaining comparable convergence behavior. In more complex instances—such as the 12-qubit SK model—high-entropy levels led DDS to maintain the maximum shot count throughout training, demonstrating its adaptive response to problem complexity. Nonetheless, convergence performance and final accuracy remained comparable to those of the standard method. In contrast, weCANS exhibited slower convergence across all models, likely due to its conservative initial shot allocation strategy.

Figure 15(a) compares the total number of shots consumed across all experiments. DDS achieved an average reduction of 36.51% in shot count compared to the standard method,

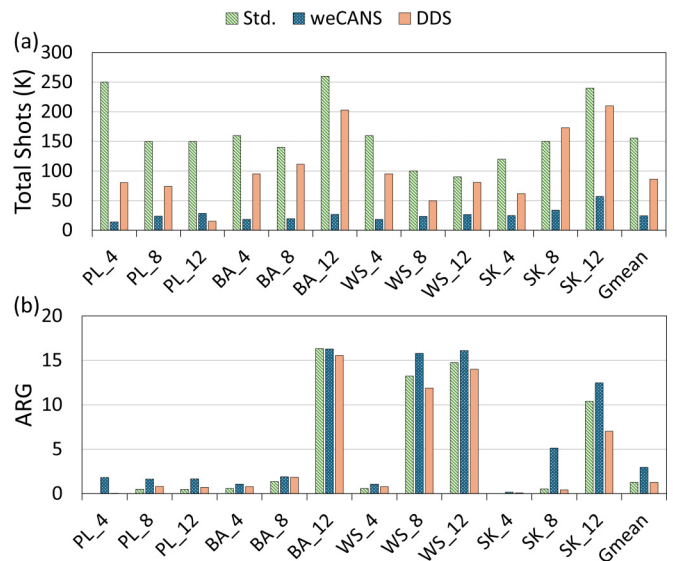


FIG. 15. (a) Total shot counts (in thousands) and (b) ARG values for the training of various QAOA models, including PL, BA, WS, and SK, across different qubit counts and model configurations. Each bar represents a shot allocation method—standard, DDS, or weCANS—all of which are based on gradient-based optimization. The final group, labeled Gmean, indicates the geometric mean of total shot counts and ARG values across all qubit sizes and QAOA models.

while weCANS reduced shot usage by 82.69%, albeit with differing trade-offs in convergence speed and solution quality. Figure 15(b) represents the ARG, a metric quantifying the deviation from the ideal solution. DDS exhibited a negligible 0.17% increase in ARG, suggesting minimal loss in solution fidelity. In contrast, weCANS incurred a substantial 131% increase, indicating that aggressive shot reduction may compromise optimization accuracy. These results underscore the capacity of DDS to maintain solution fidelity while significantly reducing resource usage.

4. Evaluation under noise condition with gradient-descent method

To assess robustness under realistic conditions, experiments were performed using the qiskit-aer qasm simulator with noisy quantum circuits, applying the same noise model settings as described previously. Noise models were constructed using calibration data from IBM's *ibm_marrakesh* and *ibm_yonsei* devices (Heron and Eagle architectures, respectively), with depolarizing noise applied to both single- and two-qubit gates. In this setting, an additional variant termed DDS_M, with an increased maximum shot count of 100 000, was introduced.

Figures 16 and 17 illustrate the number of shots and cost per iteration for QAOA simulations using the WS model, where standard, DDS, DDS_M, and weCANS shot allocation methods were employed with gradient-based optimizers. weCANS requires fewer shots overall compared to DDS, and although the number of shots gradually increases over iterations, the total shot count remains relatively low. However, the expectation value under weCANS converges more slowly and exhibits larger oscillations than DDS. In contrast, DDS

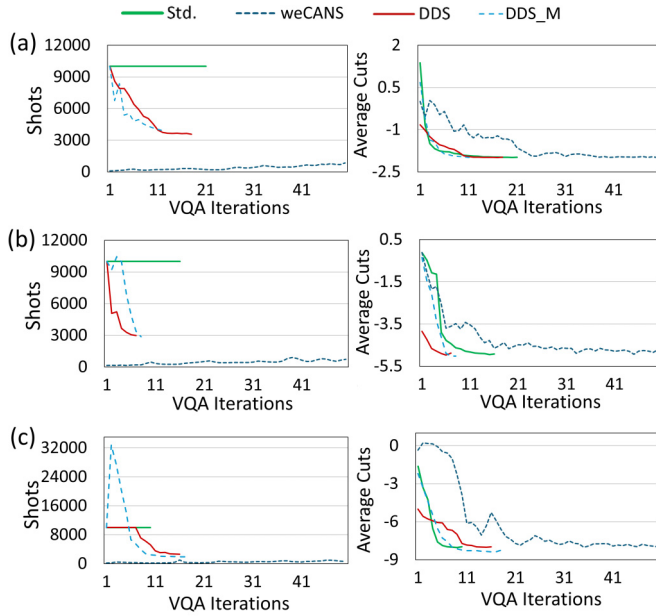


FIG. 16. Shot count and average cuts per iteration for QAOA on the WS model with (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, weCANS, DDS, and DDS_M shot allocation methods under noisy conditions. All methods employed gradient-based optimization. Simulations used the qiskit-aer qasm backend with the native gate set and error data from the *ibm_marrakesh* processor (Heron architecture).

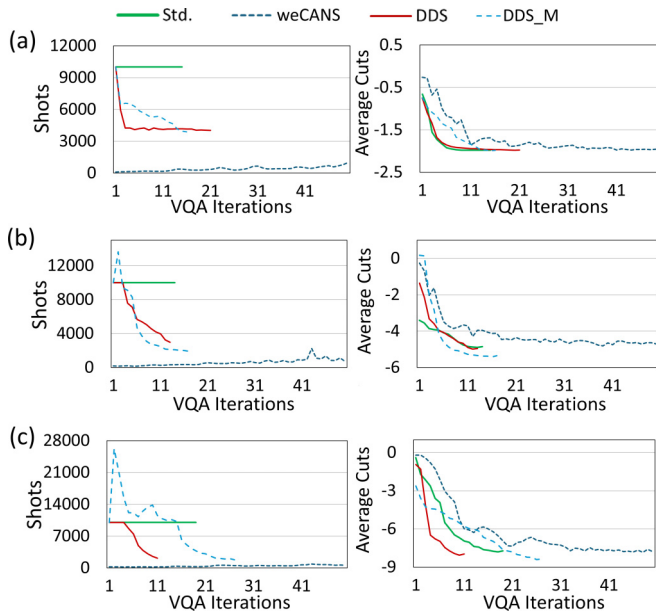


FIG. 17. Shot count and average cuts per iteration for QAOA on the WS model with (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, weCANS, DDS, and DDS_M shot allocation methods under noisy conditions. All methods employed gradient-based optimization. Simulations used the qiskit-aer qasm backend with the native gate set and error data from the *ibm_yonsei* processor (Eagle architecture).

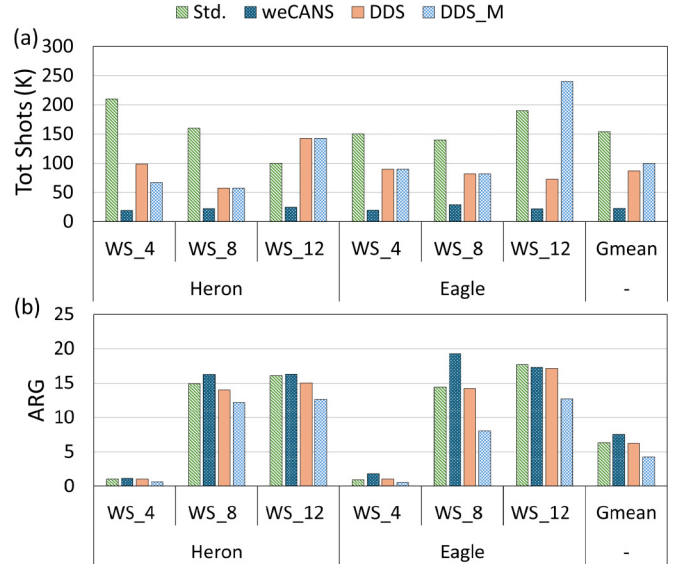


FIG. 18. (a) Total shot counts (in thousands) and (b) ARG values for the training of the WS QAOA model with 4, 8, and 12 qubits under noisy conditions. Each bar represents a shot allocation method—standard, DDS, or weCANS—all of which are based on gradient-based optimization. The final group, labeled Gmean, indicates the geometric mean of total shot counts and ARG values across all problem instances. The simulations were performed using the qiskit-aer qasm simulator, employing the native gate sets and error information from the *ibm_marrakesh* and *ibm_yonsei* processors—based on the Heron and Eagle architectures, respectively.

requires more shots than weCANS but still fewer than the standard method, while achieving faster and more stable convergence of the cost, closely following the decreasing trend of the standard approach.

Figure 18(a) summarizes total shot consumption under noisy conditions. DDS and DDS_M achieved reductions in shot usage of 44.61% and 24.08%, respectively, compared to the standard approach. weCANS demonstrated the highest reduction, at 84.40%. Figure 18(b) presents ARG values under noisy conditions. DDS slightly outperformed the standard method, achieving a 1.49% decrease in ARG, while DDS_M exhibited a more pronounced 31.98% reduction, suggesting that higher shot ceilings may help mitigate noise effects. Conversely, weCANS showed a 19.59% increase in ARG, indicating reduced robustness in the presence of hardware-level imperfections. These results, shown in Table I, confirm that DDS is compatible with gradient-based optimizers, enabling significant shot savings while preserving—and in some cases improving—optimization quality, even under noise.

5. Dynamic shots on entropy

As discussed in Sec. II C, determining the number of shots based on entropy requires a constant parameter. As the number of qubits increases, entropy also increases, often requiring a smaller constant value to maintain the desired shot count. In the QAOA experiments presented above, we set this constant to 64 for 4-qubit circuits, 8 for 8-qubit circuits, and 2 for 12-qubit circuits. The constant value can be adjusted depending on both the number of qubits and the complexity of the

TABLE I. Performance comparison of various shot allocation methods—Std., weCANS, DDS, and DDS_M—evaluated on PL, BA, WS, and SK QAOA models with 4, 8, and 12 qubits, and under noisy conditions on the WS model with the same qubit sizes. All methods employed gradient-based optimization. The table reports the geometric mean of total shot counts (in thousands) and ARG values across all considered qubit sizes and models.

Shot alloc.	Std.	weCANS	DDS	DDS_M
$S_{\text{tot}}(K)$	155.75	24.62	86.13	—
ARG	1.28	2.98	1.29	—
Noise condition (<i>ibm_marrakesh</i> and <i>ibm_yonsei</i>)				
$S_{\text{tot}}(K)$	154.13	22.68	87.01	99.55
ARG	6.32	7.56	6.23	4.30

quantum circuit. For each specific QAOA model and qubit count, we configured the constant k to identify an optimal value.

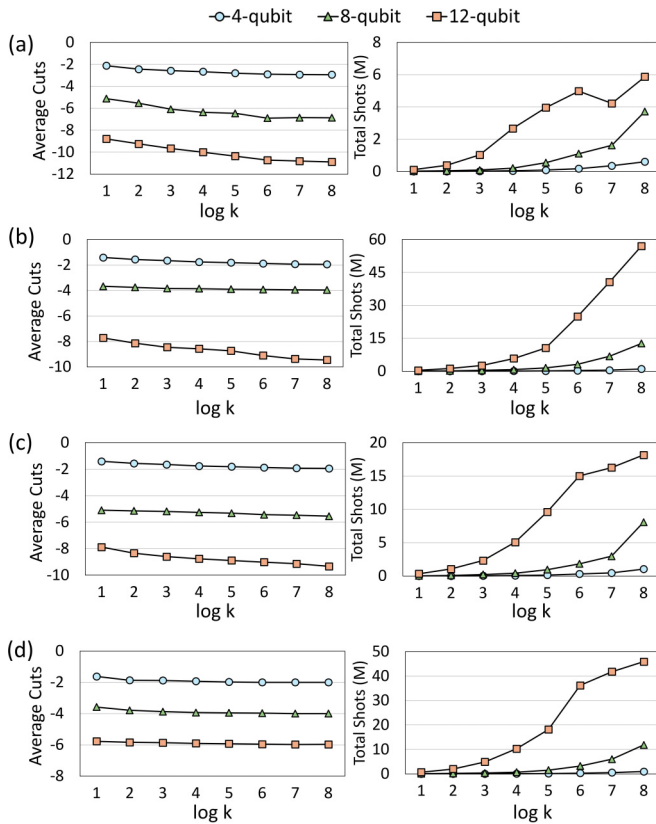


FIG. 19. Final average cuts (expectation values) and total shot counts for QAOA training of (a) PL, (b) BA, (c) WS, and (d) SK models. Each point corresponds to results obtained from 4-, 8-, and 12-qubit QAOA models under different hyperparameter k values, represented on a logarithmic scale (base 2). The final average cuts indicate the expectation values derived from the probability distributions at the end of training, while total shots, shown in millions (M), reflect the cumulative number of circuit executions consumed throughout all training iterations.

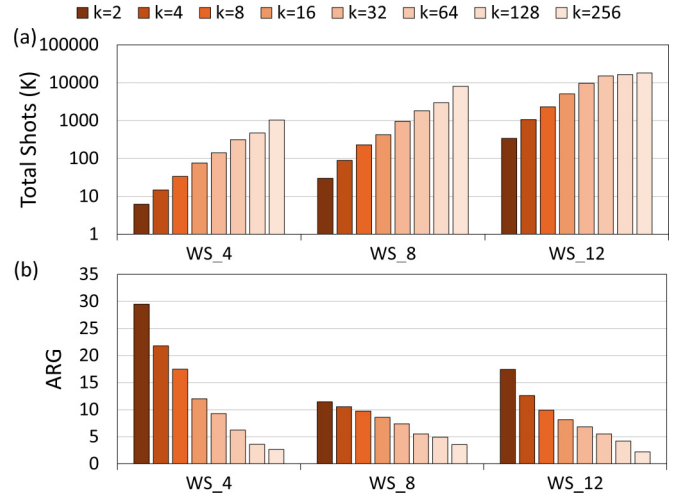


FIG. 20. (a) Total shot counts (in thousands, shown on a logarithmic scale) and (b) ARG values for the training of the WS QAOA model with 4, 8, and 12 qubits under varying hyperparameter k settings. Each bar corresponds to a DDS method with a different k value.

Figure 19 shows the total number of shots and the average cuts for various QAOA models and qubit counts, with k varied as powers of 2. In these experiments, the maximum shot count was increased to 100 000, compared to previous setups. For the PL model, entropy is relatively low because the quantum state in the final iteration has a limited number of high-probability states. Therefore, the final cost is more sensitive to changes in k compared to other models. In contrast, the SK model, which has the highest complexity, exhibits higher entropy. This results in a larger number of shots, as the higher entropy leads to a greater shot count, even for small values of k , ensuring sufficient sampling. The results show that the final cost for this model remains relatively stable across different values of k .

Figure 20 shows the total shot count as a function of k , revealing that as the number of qubits increases, the required shot count grows significantly. For circuits with a small qubit count, increasing k to enhance accuracy has minimal impact on the total shot count. In contrast, for circuits with a large number of qubits, a high k value can significantly increase the shot requirement. In such cases, adopting a smaller k may provide a more efficient strategy, facilitating the effective management of shot resources while incurring only a slight trade-off in accuracy. This approach is particularly relevant for complex circuits, where the final cost remains largely unaffected by k , thereby reducing the risk of using a smaller constant.

To further refine the selection of the constant k , both the qubit count and the complexity of the target graph can be jointly considered. In the presented experiments, k was configured based solely on the number of qubits. However, incorporating structural features of the graph enables more effective application of DDS.

The initial scaling parameter $\alpha(n)$ is defined for each qubit count n , assuming the maximum entropy case corresponding to an n -qubit random number generation circuit. The desired

total shot count without DDS is modeled as $\alpha(n) \times 2^n$, and $\alpha(n)$ is selected to match this target for each n . Once the problem instance is specified, the edge density $D(G)$ of the graph is computed as

$$D(G) = \frac{2|E|}{|V|(|V| - 1)}, \quad (7)$$

where $|V|$ is the number of nodes (qubits) and $|E|$ is the number of edges in the graph. The final value of k is then determined by normalizing $\alpha(n)$ with respect to the edge density:

$$k(n, G) = \frac{\alpha(n)}{D(G)}. \quad (8)$$

This formulation accounts for the empirical observation that sparse graphs tend to produce lower entropy in their final probability distributions, and therefore require larger k values to ensure sampling accuracy. Conversely, for denser graphs with inherently higher entropy, smaller values of k are sufficient. This method allows DDS to adaptively balance measurement cost and fidelity based on both the circuit size and problem structure.

V. CONCLUSION

Measurement-based quantum computers require repeated circuit executions to obtain the output distribution of the final quantum state. Dynamically adjusting the number of measurement shots per iteration can help reduce the overall execution time of VQA. Our observations show that, as the VQA training progresses, the information entropy of the probability distribution generally decreases. When entropy is low, fewer measurements are often sufficient to compute the cost function of VQA with adequate accuracy. To leverage this, we propose a DDS method that adjusts the shot count dynamically based on the information entropy observed in the previous VQA iteration.

The DDS method reduces the required total shot count by 50.69% compared to the conventional fixed-shot method [31] while maintaining similar accuracy and training epochs. Although our approach requires more shots than the tiered shot allocation method, it achieved a 63.34% improvement in accuracy, demonstrating its capability to reduce the total number of shots while maintaining the accuracy of VQA. Under noisy conditions, DDS achieves an average 31.15% reduction in the total shot count compared to the baseline while maintaining a slight increase in accuracy. Moreover, compared to the tiered-shot approach, DDS provides a 68.91% improvement in accuracy, even though it requires a higher shot count, demonstrating its robustness and effectiveness in noisy environments.

The DDS approach enables a significant reduction in the number of shots per iteration while maintaining the total number of VQA iterations. Moreover, DDS is applicable not only to current NISQ devices but also to future fault-tolerant quantum computers. As quantum hardware advances, achieving accurate probability distributions will entail substantial computational costs, particularly due to the large number of measurements required for tasks such as error correction and mitigation. DDS demonstrates its ability to enhance

performance even under such computationally demanding conditions. Additionally, the integration of DDS with current efficient VQA algorithms [40–46] can create a synergistic effect, further improving and accelerating the practical applicability of VQA algorithms.

For larger quantum circuits, high entropy can be observed even if the circuit is not highly complex. When elevated entropy persists throughout the iterations of the VQA, the potential for shot reduction becomes limited. This limitation stems from practical constraints observed on the IBM Quantum platform, where real QPU (quantum processing unit) executions were capped at 100 000 shots. As the performance and scalability of QPUs improve, enabling the execution of larger and more intricate circuits, the sample space will expand, leading to more complex probability distributions. Consequently, the maximum allowable shot count is also expected to increase. In such an environment, VQAs involving circuits with lower-entropy distributions are expected to benefit from reduced shot requirements relative to the enhanced maximum shot capacity. Additionally, even for VQAs exhibiting relatively high-entropy distributions during iterations, achieving a reasonable shot reduction remains feasible. These observations suggest that DDS can continue to be effective, even in scenarios that involve high-entropy circuits.

In cloud-based quantum computing systems, multiple tenants frequently submit jobs, resulting in high job volumes that lead to extended wait times for receiving results. For variational quantum algorithms, each iteration must be queued sequentially, as classically updated parameters are required for subsequent quantum circuits. Consequently, iterations are often delayed, as they cannot commence immediately after the previous one. Reducing the number of measurement shots can alleviate queuing delays for all tenants and decrease classical postprocessing overhead. Furthermore, the DDS approach presents a viable method for enhancing cost efficiency in cloud-based quantum computing [47]. By minimizing the total number of shots required, DDS significantly lowers cloud usage costs, particularly in scenarios where access to high-precision quantum devices is scarce.

ACKNOWLEDGMENTS

This research was supported by Creation of the quantum information science R&D ecosystem (based on human resources) through the National Research Foundation of Korea (NRF) funded by the Korean government (Ministry of Science and ICT (MSIT)) (RS-2023-00303229). The authors thank Dr. Seunghyo Noh at HMC for enlightening discussions.

DATA AVAILABILITY

The data supporting this study's findings are available within the article.

APPENDIX A: MEASUREMENT IN QUANTUM COMPUTING

In quantum computing, circuit execution concludes by measuring the final state of the qubits. A quantum state can be represented as a superposition of basis states, and each

measurement yields a single basis state. To construct the final probability distribution over these basis states, numerous measurements (referred to as “shots”) are required. The number of shots is determined by the programmer, and the maximum number of shots, along with default settings, is specified in the quantum system’s documentation. For each measurement, all qubits collapse to one of the basis states. After this collapse (typically to the computational basis), the circuit must be reconstructed to perform subsequent measurements.

The number of shots required can vary depending on the desired accuracy and the quantum system’s noise characteristics. Reducing the number of shots while maintaining accuracy is a key challenge, particularly in noisy intermediate-scale quantum devices. Techniques like dynamic shot allocation can optimize shot counts while preserving computational fidelity. By carefully balancing shot numbers with measurement strategies, the efficiency of quantum computations can be improved, maximizing the effective use of available quantum resources.

APPENDIX B: VARIATIONAL QUANTUM ALGORITHMS

VQA [7–11] is a quantum algorithm that minimizes the value of a given function, such as finding the ground-state energy of a molecule [48–51] or solving the maximum cut problem in QAOA [52,53]. The two main components of VQAs are the parameterized quantum part, which executes the quantum circuit that reflects the loss function, and the classical optimization part, which calculates the cost value from the results and optimizes the parameters of the quantum circuit. The quantum part includes preparing the quantum circuit and executing it to generate measurement outcomes. An ansatz is used to define the VQA circuit, representing the function to be solved.

An ansatz is a parameterized circuit composed of gates that can span the entire Bloch sphere for a single qubit or cover all norm vectors for multiple qubits. This ansatz defines the quantum circuit that represents the problem to be solved. In the circuit preparation process, the programmer initially sets the ansatz and its parameters, creating the quantum circuit. The circuit is then decomposed into native gates and executed on the quantum processor. After measuring the quantum state, the classical computer processes the measurement results to calculate the cost value and updates the quantum circuit’s parameters to minimize this cost. With the updated parameters, the ansatz is re-prepared, and the next iteration is executed. This process is repeated until the classical optimizer concludes that the cost value is minimized.

APPENDIX C: ENTROPY OF RANDOM NUMBER GENERATION CIRCUIT

In the context of quantum computing, the entropy of a system represents the uncertainty or randomness associated with its quantum state distribution of possible outcomes. The random number generation circuit is particularly relevant in this regard, as it produces quantum states with varying entropy levels based on the number of qubits and the nature of the transformations applied.

Entropy, specifically *Shannon entropy*, quantifies the uncertainty of a probability distribution $P = \{p_1, p_2, \dots, p_n\}$, where p_i represents the probability of observing the i th outcome among all possible outcomes. The formula for Shannon entropy is expressed as follows:

$$H(P) = - \sum_{i=1}^n p_i \log_2 p_i, \quad (C1)$$

where $H(P)$ is the entropy, n is the total number of possible outcomes (determined by the number of qubits), and p_i is the probability of the i th basis state or outcome.

For a random number generation circuit, the probability distribution P is derived from the circuit’s measurements. A uniform distribution of $p_i = \frac{1}{N}$ for all i results in the maximum entropy for n -qubit quantum circuits:

$$H_{\text{rand}} = - \sum_{i=1}^N \frac{1}{N} \log_2 \frac{1}{N} = \log_2 N = n. \quad (C2)$$

Conversely, nonuniform distributions result in lower entropy values, reflecting reduced randomness. In these cases, certain outcomes become more probable, which increases the system’s overall unpredictability and indicates a distinct bias in the distribution of quantum states.

APPENDIX D: EXPONENTIAL RELATIONSHIP BETWEEN ENTROPY AND REQUIRED SHOTS

Theorem D1 (Exact Hellinger-based sample-size bound).

Let $P = (p_1, \dots, p_k)$ be a probability distribution with $p_i > 0$. Let the accuracy be $\varepsilon > 0$ and the confidence level be $1 - \alpha$ ($0 < \alpha < 1$). If

$$N \geq \frac{\sum_{i=1}^k \frac{1}{p_i} \ln(2k/\alpha)}{4\varepsilon^2}, \quad (D1)$$

then the empirical distribution $\hat{P}_N$ satisfies

$$\Pr[H(\hat{P}_N, P) \leq \varepsilon] \geq 1 - \alpha, \quad (D2)$$

where $H(\cdot, \cdot)$ denotes the Hellinger distance.

Proof. Consider a Bernoulli random variable

$$X_{ij} = \begin{cases} 1 & \text{if the } j\text{th sample is } i \in \{1, \dots, k\} \\ 0 & \text{otherwise} \end{cases} \quad (D3)$$

By Hoeffding’s inequality,

$$\Pr[|\hat{p}_i - p_i| \geq \delta] \leq 2e^{-2N\delta^2}, \quad (D4)$$

where $\hat{p}_i = \frac{1}{N} \sum_j X_{ij}$. Applying Boole’s inequality,

$$\Pr[\max_i |\hat{p}_i - p_i| \geq \delta] \leq 2ke^{-2N\delta^2}, \quad (D5)$$

so requiring $2ke^{-2N\delta^2} \leq \alpha$ gives

$$N \geq \frac{\ln(2k/\alpha)}{2\delta^2}. \quad (D6)$$

The square of the Hellinger distance is given by

$$H^2(\hat{P}, P) = \frac{1}{2} \sum_i (\sqrt{\hat{p}_i} - \sqrt{p_i})^2. \quad (D7)$$

TABLE II. Performance comparison of various shot allocation methods, including PL, BA, WS, and SK, is presented across a range of qubit sizes to evaluate the effectiveness of these approaches under varying quantum circuit complexities. The table provides a comprehensive overview, detailing the average cuts after all iterations, the total number of iterations (epochs) performed, and the average shots per iteration (S_{avg}), which is calculated by dividing the total number of shots by the total number of iterations. The shot allocation methods considered include Std., a conventional fixed-shot approach utilizing 1024 shots per iteration; lin_fn, a tiered shot allocation method based on a linear function that adjusts shot counts progressively; step_fn, a tiered shot allocation method based on a step function that reduces shots in discrete intervals; and DDS, the proposed entropy-based dynamic shot allocation method, which adapts the number of shots dynamically based on the entropy of the quantum state throughout the optimization process.

Application	4-qubit PL QAOA				8-qubit PL QAOA				12-qubit PL QAOA			
	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS
S_{avg}	1024	159.05	164.08	370.3	1024	159.45	157.36	431.4	1024	160.26	157.36	565
Iterations	185	350	369	193	442	349	387	385	450	347	387	395
Average cuts	-2.68	-2.49	-2.42	-2.64	-6.77	-5.73	-6.10	-6.80	-9.98	-8.98	-8.77	-10.24
Application	4-qubit BA QAOA				8-qubit BA QAOA				12-qubit BA QAOA			
	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS
S_{avg}	1024	163.16	164.47	577.6	1024	152.60	174.58	745.2	1024	153.32	165.66	431.13
Iterations	392	340	368	420	461	367	344	390	455	365	365	406
Average cuts	-1.89	-1.65	-1.67	-1.85	-3.86	-3.13	-3.23	-3.89	-8.03	-6.04	-6.24	-7.92
Application	4-qubit WS QAOA				8-qubit WS QAOA				12-qubit WS QAOA			
	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS
S_{avg}	1024	163.16	164.47	577.6	1024	162.32	149.63	87.1	1024	145.08	158.80	391.3
Iterations	392	340	368	420	374	342	410	362	432	389	383	435
Average cuts	-1.89	-1.65	-1.67	-1.85	-4.94	-4.05	-3.90	-4.88	-7.76	-6.57	-6.84	-7.65
Application	4-qubit SK QAOA				8-qubit SK QAOA				12-qubit SK QAOA			
	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS
S_{avg}	1024	168.41	171.92	534.5	1024	141.33	169.77	688.4	1024	151.16	161.01	887.9
Iterations	360	328	350	371	409	401	355	377	422	371	377	420
Average cuts	-2.00	-1.84	-1.88	-2.00	-3.86	-3.63	-3.36	-3.82	-5.72	-4.94	-4.94	-5.79

Since

$$(\sqrt{\widehat{p}_i} - \sqrt{p_i})^2 = \frac{(\widehat{p}_i - p_i)^2}{(\sqrt{\widehat{p}_i} + \sqrt{p_i})^2} \leq \frac{(\widehat{p}_i - p_i)^2}{p_i}, \quad (\text{D8})$$

$$H^2(\widehat{P}, P) \leq \frac{1}{2} \sum_i \frac{\Delta_i^2}{p_i} \leq \frac{\delta^2}{2} \sum_i \frac{1}{p_i}, \quad (\text{D9})$$

where $\widehat{p}_i - p_i \equiv \Delta_i$ and $\max_i |\Delta_i| \leq \delta$. Choose

$$\delta = \varepsilon \sqrt{\frac{2}{\sum_i \frac{1}{p_i}}} \quad (\text{D10})$$

and substitute back. ■

Corollary D1 (Weaker bound via Rényi and Shannon entropy). The Rényi entropy is defined as

$$H_\alpha(P) = \frac{1}{1-\alpha} \log \sum_i p_i^\alpha. \quad (\text{D11})$$

Note that the Shannon entropy appears as α approaches 1. Since $H_{-1}(P) \geq H_1(P) = -\sum_i p_i \ln p_i$, any N satisfying Theorem D1 also obeys

$$N \geq \frac{\exp\{2H_1(P)\} \ln(2k/\alpha)}{4\varepsilon^2}, \quad (\text{D12})$$

which is a necessary (but not sufficient) Shannon-only condition.

TABLE III. Performance comparison of shot allocation methods applied to VQE simulations for H_2 , LiH , and BeH_2 molecules. In the VQE simulations, the energy calculated as the expectation value corresponds to the electronic ground-state energy. To obtain the total ground-state energy, the nuclear repulsion energy was added. All energy values are reported in Hartree units.

Application	H_2 VQE				LiH VQE				BeH_2 VQE			
	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS	Std.	lin_fn	step_fn	DDS
S_{avg}	1024	848.57	897.06	137.7	1024	89.63	85.30	570.3	1024	68.51	73.00	928.32
Iterations	40	36	35	37	659	698	813	658	1001	1001	1001	1001
Energy (E_h)	-1.12	-1.12	-1.13	-1.12	-7.83	-7.58	-7.76	-7.86	-15.02	-14.96	-14.69	-15.02

TABLE IV. Performance comparison of various shot allocation methods—Std., weCANS, and DDS—evaluated on PL, BA, WS, and SK models across increasing qubit sizes. Final average cuts after training convergence, total numbers of optimization iterations, and average shots per iteration (S_{avg})—calculated by dividing total shot counts by the number of iterations—are presented. Gradient-based optimization was employed using the L-BFGS-B optimizer for both the Std. and DDS methods, and the Adam optimizer for weCANS. The Std. method used 10 000 shots per circuit execution, while DDS dynamically adjusted the shot count based on quantum state entropy, with k set to 640, 80, and 20 for 4-, 8-, and 12-qubit models, respectively, and a maximum of 10 000 shots per execution. weCANS adopted a theoretically guided adaptive strategy starting from a minimum of 100 shots.

Application	4-qubit PL QAOA			8-qubit PL QAOA			12-qubit PL QAOA		
	Std.	weCANS	DDS	Std.	weCANS	DDS	Std.	weCANS	DDS
S_{avg}	10 000	278.56	4225.68	10 000	477.30	5300.29	10 000	570.48	1691.11
Iterations	25	50	19	15	50	14	15	50	9
Average cuts	−3.00	−2.94	−2.99	−6.89	−6.81	−6.87	−10.86	−10.73	−10.83
Application	4-qubit BA QAOA			8-qubit BA QAOA			12-qubit BA QAOA		
	Std.	weCANS	DDS	Std.	weCANS	DDS	Std.	weCANS	DDS
S_{avg}	10 000	367.32	5275.78	10 000	386.92	7966.93	10 000	538.06	8828.09
Iterations	16	50	18	14	50	14	26	50	23
Average cuts	−1.98	−1.97	−1.98	−3.92	−3.90	−3.90	−7.97	−7.98	−8.05
Application	4-qubit WS QAOA			8-qubit WS QAOA			12-qubit WS QAOA		
	Std.	weCANS	DDS	Std.	weCANS	DDS	Std.	weCANS	DDS
S_{avg}	10 000	367.32	5275.78	10 000	468.58	4972.50	10 000	532.14	5757.43
Iterations	16	50	18	10	50	10	9	50	14
Average cuts	−1.98	−1.97	−1.98	−4.99	−4.84	−5.0684	−8.13	−8.00	−8.20
Application	4-qubit SK QAOA			8-qubit SK QAOA			12-qubit SK QAOA		
	Std.	weCANS	DDS	Std.	weCANS	DDS	Std.	weCANS	DDS
S_{avg}	10 000	494.70	6175.80	10 000	677.88	7865.55	10 000	1142.86	10 000
Iterations	12	50	10	15	50	22	24	50	21
Average cuts	−2.00	−1.99	−1.99	−3.97	−3.79	−3.98	−5.35	−5.23	−5.55

APPENDIX E: IMPACT OF BARREN PLATEAUS ON DDS

An important consideration regarding the DDS framework is its behavior in VQAs affected by barren plateau (BP), where cost function landscapes become exponentially flat and gradients vanish [54,55]. Since DDS dynamically adjusts shot counts based on the entropy of the probability distributions, the presence of BPs—where gradients scale as $O(2^{-n})$ for n qubits—can result in probability distributions that evolve very slowly or remain nearly uniform throughout the iterations. In such cases, entropy may not decrease as expected during VQA training (Sec. II), potentially causing DDS to allocate a consistently high number of shots (e.g., $S = k \times 2^H$ with $H \approx n$ for uniform distributions) thereby limiting achievable shot reductions.

However, it is important to note that BPs primarily hinder the efficiency of the VQA optimization rather than directly affecting the shot allocation process. This indicates that while DDS may still obtain marginal reductions in shots by responding to subtle shifts in the probability distribution, its overall effectiveness is likely diminished in BP-dominated landscapes. A more comprehensive analysis could involve simulating circuits affected by BP (e.g., deep random ansatzes [54]) and analyzing the stability of their entropy. Nonetheless, mitigating BPs through strategies such as using shallow circuits or localized ansatzes [56] would likely be necessary for DDS to retain meaningful efficiency. Despite its advantages,

the DDS encounters challenges in the presence of BPs, and this limitation will be addressed in future work.

APPENDIX F: PERFORMANCE COMPARISON OF SHOT APPROACHES

Tables II and III present the final cost, total iterations (i.e., epochs), and the average number of shots per iteration for different shot allocation methods used in the training of each VQA model. The total number of iterations (denoted as *Iterations*) excludes the final common 1024-shot evaluation described in Sec. IV B. Additionally, S_{avg} represents the total number of shots used throughout the training process, divided by the total number of iterations, providing an estimate of shot efficiency for each method. The shot allocation strategies vary in how they distribute shots across training iterations, affecting both training efficiency and accuracy.

For each shot allocation approach, there are the fixed-shot method, tiered shot allocation method from previous work, and DDS. The “Std.” method represents a standard shot approach, maintaining a constant 1024 shots per iteration without adjustments. The “lin_fn” and “step_fn” methods are tiered shot allocation strategies. The lin_fn starts with 1000 shots and reduces the number of shots linearly over iterations, while the step_fn decreases the shot count by 10 every 10 iterations. Lastly, the “DDS” method dynamically determines

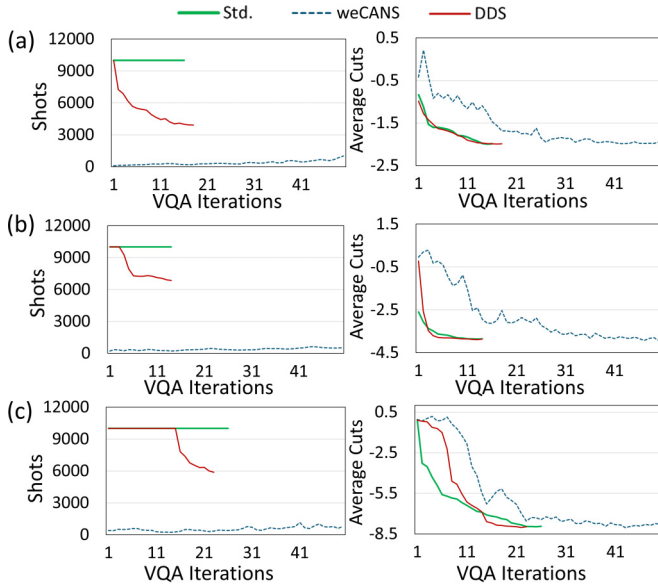


FIG. 21. The number of shots and the average cuts per iteration during the BA model QAOA training for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, DDS, and weCANS shot allocation methods based on gradient-based optimization. The expectation value corresponds to the average cuts.

the number of shots based on the entropy of the probability distribution from the previous iteration, as detailed in the paper.

For the QAOA models (PL, BA, WS, and SK), simulations were conducted with qubit counts of 4, 8, and 12. For the VQE models, molecular systems including H_2 , LiH , and BeH_2 were simulated. Across all VQA applications and varying qubit

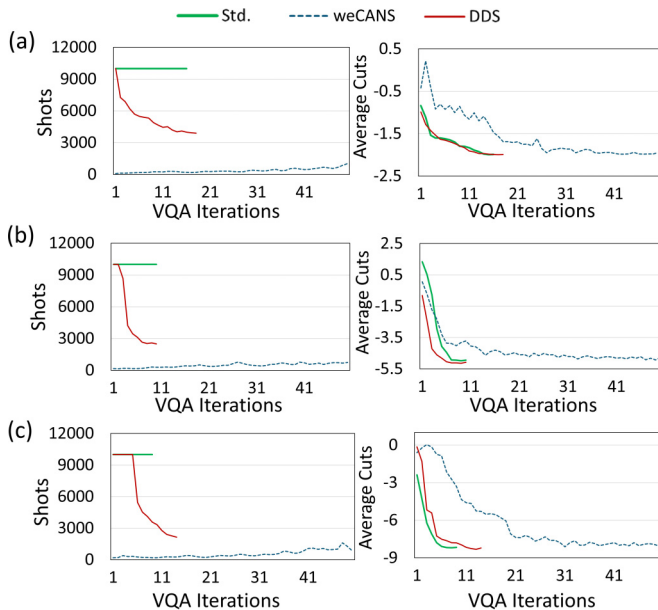


FIG. 22. The number of shots and the average cuts per iteration during the WS model QAOA training for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, DDS, and weCANS shot allocation methods based on gradient-based optimization.

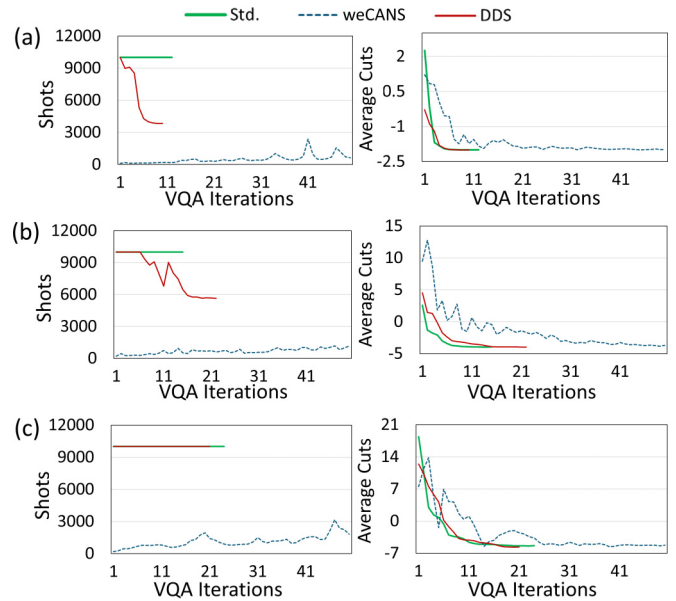


FIG. 23. The number of shots and the average cuts per iteration during the SK model QAOA training for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, using standard, DDS, and weCANS shot allocation methods based on gradient-based optimization.

numbers, DDS demonstrated a significant reduction in the required number of shots compared to the standard method, while maintaining a similar total number of iterations. Specifically, for QAOA, the average reduction in shot count was 50.69%, and for VQE, it was 47.09%. Notably, the final cost values showed negligible differences between DDS and the standard method, indicating that DDS achieves comparable accuracy with fewer shots. When comparing the final cost to the ideal value using the ARG metric, an average increase of only 6.03% was observed for QAOA and VQE. Interestingly, for VQE, the ARG value actually decreased by 24.39%, indicating that DDS may offer an efficient approach to cost estimation with fewer shots.

The `lin_fn` and `step_fn` methods reduce the shot count by 85.49% and 84.24%, respectively, compared to the standard method. However, these methods incur significant accuracy losses, as indicated by ARG increases of 241.18% and 230.73%, respectively. In contrast, DDS dynamically adjusts the number of shots, minimizing accuracy degradation while still requiring fewer shots than the conventional fixed-shot approach. Although DDS requires more shots than the tiered shot allocation methods, it results in an overall ARG reduction of 63.34%, demonstrating a substantial improvement in accuracy.

In addition to the gradient-free optimization experiments conducted using COBYLA, we evaluated the shot allocation methods under gradient-based optimization, shown in Table IV. The standard and DDS methods employed the L-BFGS-B optimizer, while Adam was used for weCANS. Given the higher sensitivity of gradient-based methods to shot noise, the standard method was executed with 10 000 shots per circuit, an increase from the 1024 shots used in the gradient-free setting. For DDS, the entropy-based shot allocation approach was retained, with the hyperparameter k scaled by a factor of

ten—set to 640, 80, and 20 for 4-, 8-, and 12-qubit simulations, respectively. A maximum of 10 000 shots per circuit execution was imposed. For weCANS, the minimum number of shots was set to 100.

As in the previous experiments, simulations were performed for QAOA models (PL, BA, WS, and SK) with 4, 8, and 12 qubits, as shown in Figs. 14, 21, 22, and 23. Across these models, DDS reduced the number of shots required during training by an average of 36.51% compared to the standard method, while maintaining a similar number of iterations. The final cost values showed little deviation from those obtained using the standard method. When evaluated using the ARG metric, the average increase was 0.17%, indicating that the accuracy was preserved.

In contrast, weCANS achieved a greater reduction in the number of shots by an average of 82.69%, but resulted in a larger accuracy loss, with an ARG increase of 131%. DDS used more shots than weCANS but achieved a lower ARG, with a 56.70% reduction relative to weCANS. These results suggest that DDS provides a more balanced trade-off between shot efficiency and optimization accuracy in gradient-based settings.

APPENDIX G: EVALUATION OF SHOT ALLOCATION METHODS AGAINST THE STANDARD

When employing different shot allocation methods in VQA simulations, the final cost typically converges to a specific value as the iterations progress. This value represents the average cuts in the case of QAOA and the energy in the case of VQE. It is considered more effective if the cost converges faster to a lower value and exhibits minimal fluctuations during the convergence process. Conversely, if the cost fails to converge accurately in later iterations and exhibits significant fluctuations, it may hinder the accurate measurement of the final value.

Figure 24 presents the energy variations at each iteration during VQE simulations of H_2 , LiH , and BeH_2 using different shot allocation methods compared to the standard fixed-shot allocation approach. In the case of H_2 , due to its small number of qubits, the total number of iterations is relatively low, and the differences between shot allocation methods are not particularly pronounced. However, for LiH and BeH_2 , it is evident that the `linear_fn` and `step_fn` methods exhibit slower convergence and display oscillatory behavior until the final iteration.

Figure 25 presents the average cuts calculated at each iteration for QAOA simulations applied to the PL, BA, WS, and SK models, using various shot allocation methods compared to the standard shot allocation approach. The results indicate that both `linear_fn` and `step_fn` exhibit slower convergence compared to DDS across all QAOA models. Furthermore, these methods do not converge to a steady value but instead display oscillatory behavior throughout the iterations, persisting until the final iteration.

Both QAOA and VQE utilize fewer shots compared to the standard fixed-shot method when applying the shot allocation approach. Notably, in the case of DDS, there are instances where the computed cost is lower than that of the standard

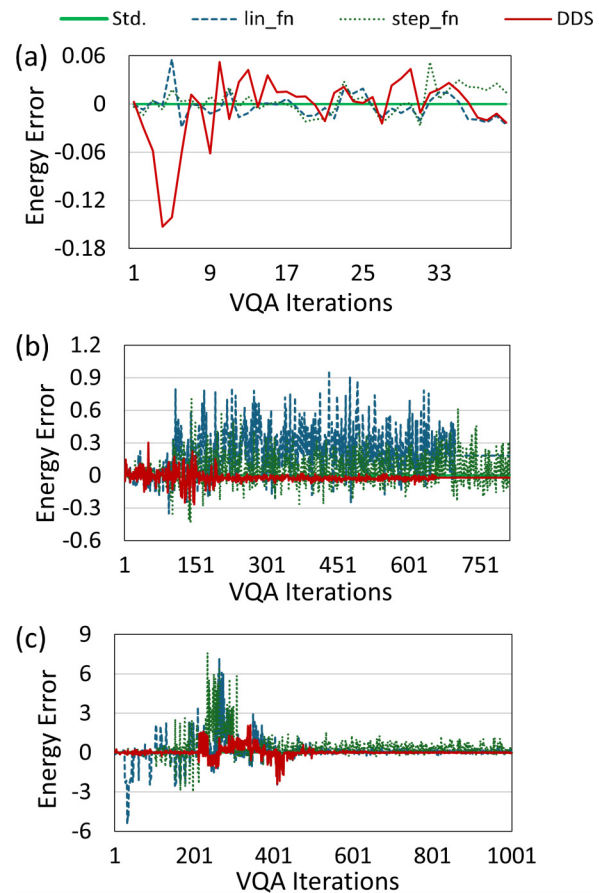


FIG. 24. Iteration-wise deviation in energy values from the standard shot allocation baseline during VQE simulations for (a) H_2 , (b) LiH , and (c) BeH_2 . The energy error was computed at each iteration as the deviation in energy values from the baseline established by the standard shot allocation method, which is fixed at zero. This metric quantifies the relative performance of alternative shot allocation strategies throughout the optimization process.

method. A lower calculated cost during iterations indicates the possibility for the optimization process to converge more quickly to the lowest cost. As observed in the figures, better training performance is indicated when the error approaches a negative value. DDS demonstrates the potential for improved convergence while requiring fewer shots than the fixed-shot allocation method.

When using gradient-based optimization methods—L-BFGS-B for the standard and DDS approaches, and Adam for weCANS—the evolution of the average cuts error provides insight into the stability and convergence characteristics of each shot allocation strategy. The average cuts error is defined as the deviation of each method’s average cut value from that of the standard shot allocation method. Since the standard method serves as the baseline, its corresponding error remains fixed at zero throughout the training process. For DDS and weCANS, however, this error varies with iteration, reflecting differences in the convergence trajectory due to adaptive shot allocation.

Figure 26 shows that in the early phase of training, both DDS and weCANS exhibit relatively large fluctuations in

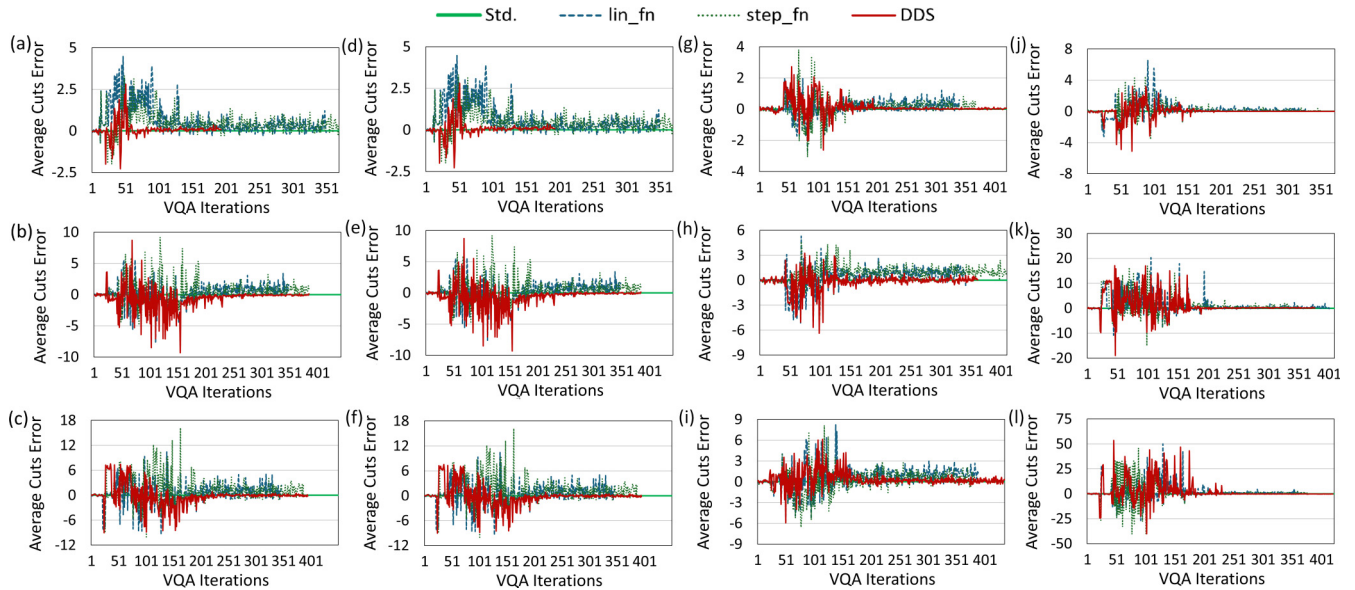


FIG. 25. Iteration-wise deviation in average cut values from the standard shot allocation baseline during QAOA training. Results are shown for the model for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits; BA model for (d) 4 qubits, (e) 8 qubits, and (f) 12 qubits; WS model for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits; SK model for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits. The average cuts error was computed at each iteration as the deviation in average cut values from the baseline established by the standard shot allocation method, which is fixed at zero. Each line corresponds to a different shot allocation strategy: linear function, step function, and DDS.

average cuts error when compared to the standard method. This is attributed to the reduced number of shots used per iteration, which amplifies the effect of statistical noise on cost estimation. These fluctuations are particularly pronounced for weCANS, which begins training with a minimal shot count of 100. Despite these initial variances, both methods show a trend toward convergence as training progresses. DDS, in particular, demonstrates faster stabilization of the average cuts error compared to weCANS, indicating that its entropy-

informed shot adjustment leads to more consistent training dynamics.

An additional consideration arises due to the earlier convergence of the standard method. Because gradient-based optimization with a fixed, high number of shots (10 000 per execution) allows the standard method to reach convergence in fewer iterations, there are instances where weCANS continues training beyond the point at which the standard method has terminated. To enable a consistent comparison across the

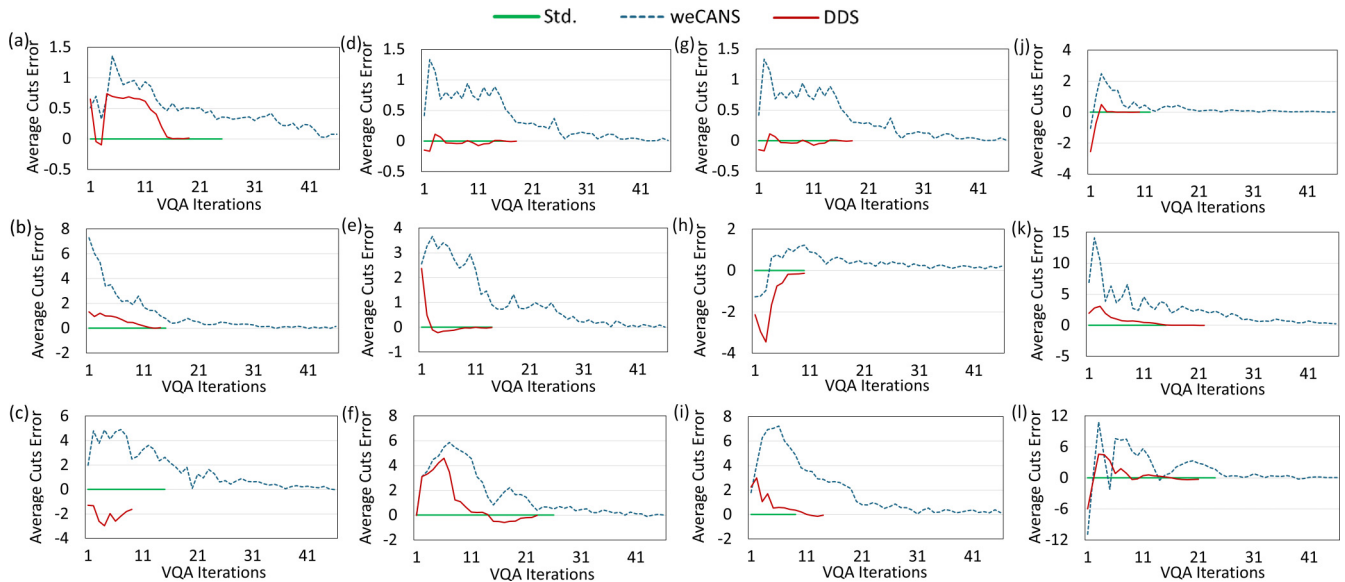


FIG. 26. Iteration-wise deviation in average cut values from the standard shot allocation baseline during QAOA training. Results are shown for the model for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits; BA model for (d) 4 qubits, (e) 8 qubits, and (f) 12 qubits; WS model for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits; SK model for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits. The average cuts error was computed at each iteration as the deviation in average cut values from the baseline established by the standard shot allocation method, which is fixed at zero. Gradient-based optimization was used, and each line represents a different shot allocation strategy: standard, weCANS, and DDS.

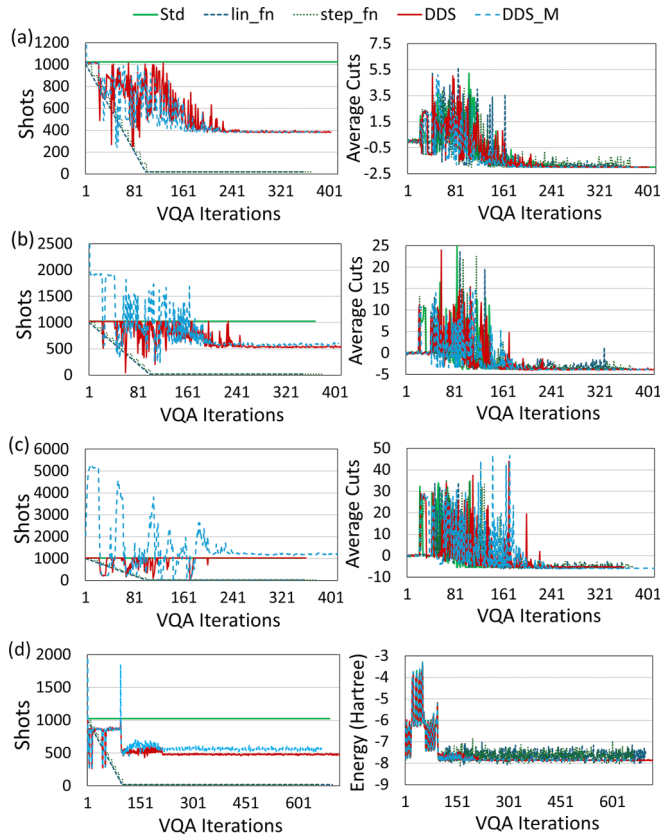


FIG. 27. Shot count and expectation values per iteration for QAOA on the SK model with (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, and for VQE on (d) LiH, using standard, linear function, step function, DDS, and DDS_M shot allocation methods under noisy conditions. Expectation values correspond to average cuts (QAOA) or electronic energy (VQE), computed from quantum states sampled with the assigned shot count. Simulations used the qiskit-aer qasm backend with the native gate set and error data from the *ibm_marrakesh* processor (Heron architecture).

full range of iterations, we define the error in these later stages as the deviation of each method's expectation value from the final expectation value obtained in the last iteration of the standard method. This approach ensures that the error metric continues to reflect relative performance throughout the entire training trajectory, even after the standard method has completed optimization.

Overall, these results suggest that while adaptive shot allocation methods such as DDS and weCANS introduce greater variance in the early stages of optimization, they are capable of achieving convergence with fewer measurement resources. In particular, DDS not only converges faster than weCANS but also maintains closer proximity to the baseline standard in the final stages of training, highlighting its robustness and efficiency in gradient-based quantum optimization settings.

APPENDIX H: DYNAMIC SHOT ALLOCATION IN NOISY SIMULATOR

We applied various shot allocation methods to VQA in a noisy environment and analyzed their impact on the total number of shots and accuracy. The qiskit-aer qasm simulator

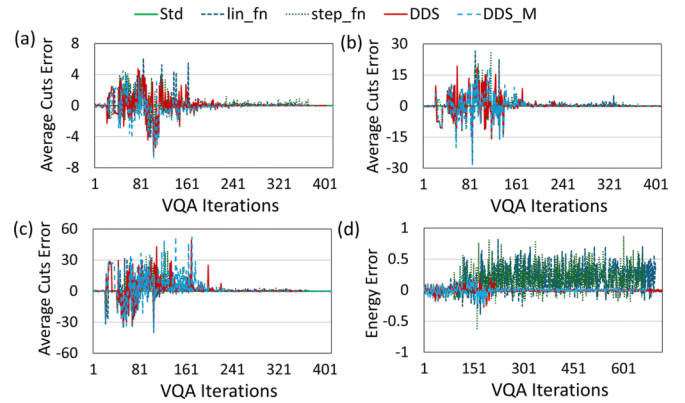


FIG. 28. Deviation in average cuts (QAOA) for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, and energy (VQE) for (d) LiH per iteration from the standard shot allocation baseline, fixed at zero. Each line represents a shot allocation strategy: linear function, step function, DDS, or DDS_M. Simulations were performed with the qiskit-aer qasm backend using the native gate set and error data from the *ibm_marrakesh* processor (Heron architecture).

was employed to simulate noise, using the native gate sets and qubit errors of the *ibm_marrakesh* and *ibm_yonsei* processors, which are based on the Heron and Eagle architectures. For the VQA models, we utilized QAOA with SK models across various qubit numbers and VQE for LiH.

Figures 27 and 28 illustrate the shot count and cost per iteration for QAOA simulations with the SK model and VQE simulations for the LiH molecule, using the standard, linear_fn, step_fn, DDS, and DDS_M shot allocation methods. In these simulations, the expectation value computed through the VQE circuit corresponds to the electronic ground-state energy, with the total ground-state energy obtained by adding the nuclear repulsion energy. Even in a noisy environment, DDS demonstrates good convergence, whereas linear_fn and step_fn fail to fully converge by the final iteration, displaying oscillatory behavior in their graphs. DDS_M, which allows iterations with shot counts exceeding 1024, shows improved convergence in the cost function compared to other methods.

Figures 29 and 30 illustrate the average cuts error calculated at each iteration for QAOA simulations applied to the SK model and VQE simulations for LiH, comparing various shot allocation methods to the standard shot allocation approach. The results demonstrate that both linear_fn and step_fn exhibit slower convergence compared to DDS across all VQA models. Moreover, these methods fail to converge to a steady value and instead show oscillatory behavior throughout the iterations, persisting until the final iteration. In the QAOA SK model training process, significant fluctuations in the average cuts error are observed during earlier iterations, which makes the fluctuations in the final iteration less apparent. However, during the LiH VQE training process, these differences are more pronounced and clearly visible.

Tables V and VI present the final cost, total iterations (i.e., epochs), and the average number of shots (S_{avg}) per iteration for different shot allocation methods used in the training of each VQA model under noisy conditions. The linear_fn and step_fn methods exhibit a much smaller average number of shots per iteration compared to other shot allocation methods.

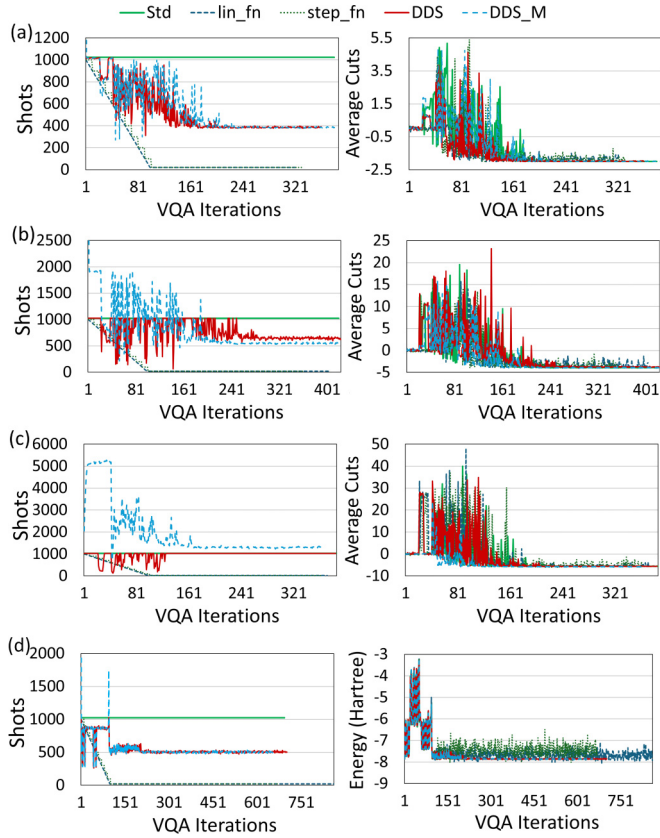


FIG. 29. Shot count and expectation values per iteration for QAOA on the SK model with (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, and for VQE on (d) LiH, using standard, linear function, step function, DDS, and DDS_M shot allocation methods under noisy conditions. Expectation values correspond to average cuts (QAOA) or electronic energy (VQE), computed from quantum states sampled with the assigned shot count. Simulations used the qiskit-aer qasm backend with the native gate set and error data from the *ibm_yonsei* processor (Eagle architecture).

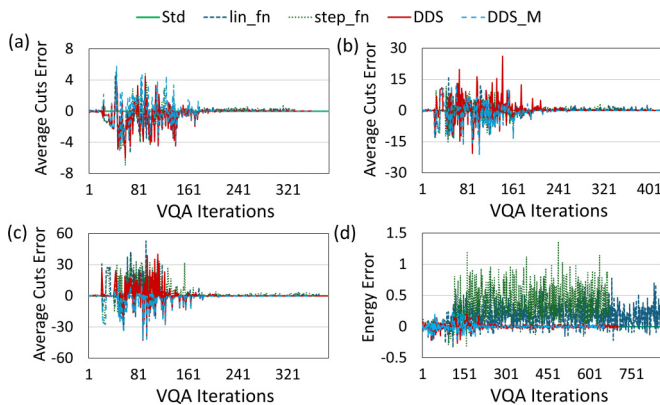


FIG. 30. Deviation in average cuts (QAOA) for (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits, and energy (VQE) for (d) LiH per iteration from the standard shot allocation baseline, fixed at zero. Each line represents a shot allocation strategy: linear function, step function, DDS, or DDS_M. Simulations were performed with the qiskit-aer qasm backend using the native gate set and error data from the *ibm_yonsei* processor (Eagle architecture).

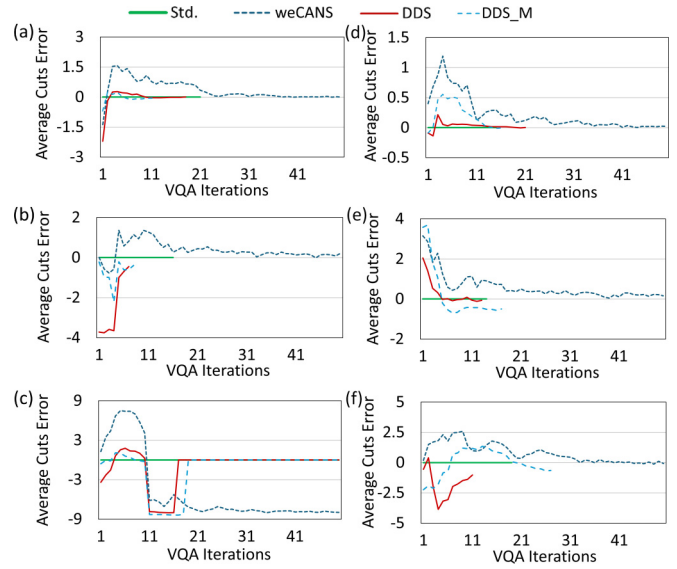


FIG. 31. Deviation in average cuts per iteration for QAOA on the WS model with (a) 4 qubits, (b) 8 qubits, and (c) 12 qubits using noise data from the *ibm_marrakesh* processor (Heron architecture), and with (d) 4 qubits, (e) 8 qubits, and (f) 12 qubits using noise data from the *ibm_yonsei* processor (Eagle architecture). Each line represents a shot allocation strategy: weCANS, DDS, or DDS_M. All methods employed gradient-based optimization. Simulations used the qiskit-aer qasm backend with the native gate set and error data from each processor.

However, when examining the final cost values, average cuts, and energy, it is evident that these values do not converge as well as those obtained using the standard method. In the case of DDS, the overall number of iterations is similar, but the average number of shots is reduced by half, with the final cost remaining almost the same. DDS_M requires fewer average shots than the standard for small models, but more for the large 12-qubit SK QAOA models. Nevertheless, the final cost, average cuts, and energy are lower than the standard, indicating better accuracy.

To further evaluate the effectiveness of various shot allocation methods under realistic noise conditions in conjunction with gradient-based optimization, we conducted additional simulations using the same noise models described previously. The qiskit-aer qasm simulator was used with noise profiles derived from IBM's *ibm_marrakesh* and *ibm_yonsei* devices, reflecting the Heron and Eagle architectures. Depolarizing noise was applied to both single- and two-qubit gates to model the effect of hardware-level imperfections.

The number of shots and cost per iteration for QAOA simulations using the WS model are evaluated, where standard, DDS, DDS_M, and weCANS shot allocation methods were employed with gradient-based optimizers. In this setting, DDS dynamically adjusted the number of shots per iteration within a cap of 10 000, while DDS_M extended the maximum to 100 000 shots. The weCANS method, incorporating a learning-based shot allocation mechanism, was also evaluated with a minimum shot threshold of 100.

Figure 31 presents the average cuts error at each iteration during QAOA training with the SK model. The results indicate

TABLE V. Performance comparison of various shot allocation methods under noisy condition. SK is presented across a range of qubit sizes to evaluate the effectiveness of these approaches under varying quantum circuit complexities. For each quantum processor architecture (Heron and Eagle), the `ibm_marrakesh` and `ibm_yonsei` processors are used to provide error information for the noise simulator. The table details the average cuts after all iterations, the total number of iterations (epochs) performed, and the average shots per iteration (S_{avg}). The shot allocation methods considered include Std., `lin_fn`, `step_fn`, DDS, and DDS_M. DDS_M is the DDS method that does not have the 1024-shot limit.

QPU arch.		Heron														
Application		4-qubit SK QAOA					8-qubit SK QAOA					12-qubit SK QAOA				
		Std.	<code>lin_fn</code>	<code>step_fn</code>	DDS	DDS_M	Std.	<code>lin_fn</code>	<code>step_fn</code>	DDS	DDS_M	Std.	<code>lin_fn</code>	<code>step_fn</code>	DDS	DDS_M
S_{avg}		1024	155.56	162.14	551.34	687.35	1024	159.05	159.16	703.74	861.09	1024	155.18	161.39	945.89	1471.97
Iterations		414	359	374	403	388	368	350	382	410	406	355	360	376	360	411
Average cuts		-1.998	-1.85	-1.83	-1.98	-2.00	-3.79	-3.23	-3.37	-3.82	-3.89	-5.52	-4.93	-5.11	-5.43	-5.88
QPU arch.		Eagle														
Application		4-qubit SK QAOA					8-qubit SK QAOA					12-qubit SK QAOA				
		Std.	<code>lin_fn</code>	<code>step_fn</code>	DDS	DDS_M	Std.	<code>lin_fn</code>	<code>step_fn</code>	DDS	DDS_M	Std.	<code>lin_fn</code>	<code>step_fn</code>	DDS	DDS_M
S_{avg}		1024	170.71	179.70	532.92	720.47	1024	140.42	168.10	775.18	832.68	1024	150.81	166.87	961.62	1959.25
Iterations		380	323	333	360	386	420	404	359	423	422	386	372	362	385	362
Average cuts		-2.000	-1.87	-1.85	-1.99	-2.00	-3.88	-3.36	-3.54	-3.76	-3.91	-5.58	-5.19	-5.31	-5.67	-5.81

that both DDS and DDS_M converge stably in noisy environments. DDS displays more consistent and faster convergence compared to other methods, while DDS_M, benefiting from a higher shot ceiling, further improves the quality of convergence. In contrast, weCANS shows greater error fluctuation throughout the training process, especially in later iterations.

Table VII shows the final cost, average shot count, and total iterations for each shot allocation method under noisy conditions with gradient-based optimization. DDS and DDS_M reduced the total shot count by 44.61% and 24.08%, respectively, relative to the standard method. Despite this reduction, DDS achieved a slight improvement in accuracy, as reflected in a 1.49% decrease in ARG. DDS_M demonstrated a more substantial accuracy gain, reducing ARG by 31.98%, indicating that increasing the shot ceiling can be an effective strategy for mitigating the effects of noise in gradient-based optimization. On the other hand, weCANS, while achieving the greatest shot savings by 84.40% reduction, incurred an ARG increase of 19.59%, suggesting reduced robustness under noisy conditions.

These results demonstrate that DDS-based methods are compatible with gradient-based optimizers and particularly effective in noisy quantum environments. By dynamically adjusting the number of shots according to the optimization landscape, DDS enables efficient use of quantum resources

while preserving convergence quality. Moreover, DDS_M, with its enhanced shot flexibility, shows improved robustness under higher noise conditions, further enhancing performance. Although the weCANS achieves the largest reduction in shots, it compromises the stability of convergence, whereas both DDS and DDS_M offer a more balanced trade-off between efficiency and accuracy. Overall, these findings highlight the practicality of DDS-based shot allocation for scalable and reliable optimization on NISQ devices.

APPENDIX I: SHOT REDUCTION WITH ADJUSTING OPTIMIZER

Previous efforts to reduce shot counts in VQAs have introduced methods such as iCANS [17], Rosalin [18], gCANS [19], weCANS [14], and SantaQlaus [15], which optimize the shot allocation process by modifying the optimizer within the VQA. Specifically, iCANS [17] determines the sample count per iteration to maximize expected gain in the stochastic gradient descent process. While iCANS allows for individual variation of shot counts across gradient components, gCANS [19] applies a global optimization strategy to maximize expected gain for the entire gradient vector, and weCANS [14] focuses on maximizing gain per unit of time. Although integrating DDS with these optimizer-based schemes presents

TABLE VI. Performance comparison of shot allocation methods applied to VQE simulations for LiH under noisy condition. In the VQE simulations, the energy calculated as the expectation value corresponds to the electronic ground-state energy. To obtain the total ground-state energy, the nuclear repulsion energy was added. All energy values are reported in Hartree units.

QPU arch.		Heron					Eagle				
Application		LiH VQE					LiH VQE				
		Std.	<code>lin_fn</code>	<code>step_fn</code>	DDS	DDS_M	Std.	<code>lin_fn</code>	<code>step_fn</code>	DDS	DDS_M
S_{avg}		1024	89.73	100.21	532.97	616.26	1024	76.43	99.25	554.69	569.45
Iterations		689	697	662	717	670	694	861	670	702	653
Energy (E_h)		-7.86	-7.64	-7.62	-7.86	-7.87	-7.87	-7.67	-7.46	-7.86	-7.86

TABLE VII. Performance comparison of various shot allocation methods—Std., weCANS, DDS, and DDS_M—under noisy conditions using the WS QAOA model across a range of qubit sizes (4, 8, and 12). Simulations were performed using noise models derived from calibration data of IBM’s *ibm_marrakesh* and *ibm_yonsei* devices, representing Heron and Eagle quantum processor architectures, respectively. For each setting, results were collected under gradient-based optimization, with the L-BFGS-B optimizer applied for Std. and DDS, and the Adam optimizer used for weCANS. Metrics reported include the final average cut value after training, the total number of optimization iterations, and the average number of shots per iteration (S_{avg}). DDS employed entropy-based adaptive shot allocation with a shot cap of 10 000 per execution, and DDS_M denotes a variant of DDS without this constraint. In weCANS, shot allocation was initiated from a minimum of 100 shots and scaled up based on gradient-based uncertainty estimates.

QPU arch.					Heron							
Application	4-qubit WS QAOA				8-qubit WS QAOA				12-qubit WS QAOA			
	Std.	weCANS	DDS	DDS_M	Std.	weCANS	DDS	DDS_M	Std.	weCANS	DDS	DDS_M
S_{avg}	10 000	386.22	5464.72	5562.92	10 000	450.84	4755.86	7199.63	10 000	496.96	6642.81	7929.50
Iterations	21	50	18	12	16	50	7	8	10	50	16	18
Average cuts	−1.97	−1.97	−1.97	−1.98	−4.89	−4.81	−4.94	−5.05	−8.00	−7.98	−8.10	−8.33
QPU arch.					Eagle							
Application	4-qubit WS QAOA				8-qubit WS QAOA				12-qubit WS QAOA			
	Std.	weCANS	DDS	DDS_M	Std.	weCANS	DDS	DDS_M	Std.	weCANS	DDS	DDS_M
S_{avg}	10 000	394.58	4512.33	5623.00	10 000	577.44	6135.15	4829.47	10 000	442.14	6604.82	8891.44
Iterations	15	50	21	16	14	50	13	17	19	50	11	27
Average cuts	−1.98	−1.96	−1.97	−1.98	−4.92	−4.64	−4.93	−5.28	−7.85	−7.89	−7.91	−8.33

challenges, modifying the optimizers within VQAs makes it difficult to apply them together with other shot optimization methods. Future work could explore combining these approaches to improve shot efficiency in VQA.

APPENDIX J: VARIANCE-MINIMIZATION-BASED SHOT ASSIGNMENT

Variance-minimization-based shot assignment strategies [18,22,24–28] aim to reduce total measurement variance by grouping commutable Hamiltonian terms into cliques, allowing simultaneous measurement of each clique. In this

approach, the number of measurements for each clique, N , is optimized to minimize variance across all measurements. The Variance-Preserving Shot Reduction (VPSR) [28] technique further refines this process by reducing the total number of shots while maintaining variance. VPSR initially measures k shots for each clique to estimate its standard deviation and then uses the aggregated standard deviation data to effectively allocate an optimized shot count per clique. Although VPSR effectively reduces total shot usage while preserving variance, the initial k shot per clique, followed by a standard deviation aggregation and shot allocation phase, introduces additional resource and computational overhead.

- [1] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.* **41**, 303 (1999).
- [2] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (ACM Press, New York, NY, USA, 1996), pp. 212–219.
- [3] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature (London)* **574**, 505 (2019).
- [4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2010).
- [5] E. Farhi and A. W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, [arXiv:1602.07674](https://arxiv.org/abs/1602.07674).
- [6] R. Jozsa, Quantum algorithms and the Fourier transform, *Proc. R. Soc. London Ser. A* **454**, 323 (1998).
- [7] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, Variational quantum algorithms, *Nat. Rev. Phys.* **3**, 625 (2021).
- [8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature (London)* **549**, 195 (2017).
- [9] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, *Phys. Rev. Lett.* **113**, 130503 (2014).
- [10] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, *New J. Phys.* **18**, 023023 (2016).
- [11] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning

- with quantum-enhanced feature spaces, *Nature (London)* **567**, 209 (2019).
- [12] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [13] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth *et al.*, The variational quantum eigensolver: A review of methods and best practices, *Phys. Rep.* **986**, 1 (2022).
- [14] K. Ito, Latency-aware adaptive shot allocation for run-time efficient variational quantum algorithms, [arXiv:2302.04422](https://arxiv.org/abs/2302.04422).
- [15] K. Ito and K. Fujii, SantaQlaus: A resource-efficient method to leverage quantum shot-noise for optimization of variational quantum algorithms, [arXiv:2312.15791](https://arxiv.org/abs/2312.15791).
- [16] D. Luo, J. Shen, R. Dangovski, and M. Soljacic, QuACK: Accelerating gradient-based quantum optimization with Koopman operator learning, in *Advances in Neural Information Processing Systems*, edited by A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Curran Associates, Inc., Red Hook, NY, 2023), Vol. 36, pp. 25662–25692.
- [17] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles, An adaptive optimizer for measurement-frugal variational algorithms, *Quantum* **4**, 263 (2020).
- [18] A. Arrasmith, L. Cincio, R. D. Somma, and P. J. Coles, Operator sampling for shot-frugal optimization in variational algorithms, [arXiv:2004.06252](https://arxiv.org/abs/2004.06252).
- [19] A. Gu, A. Lowe, P. A. Dub, P. J. Coles, and A. Arrasmith, Adaptive shot allocation for fast convergence in variational quantum algorithms, [arXiv:2108.10434](https://arxiv.org/abs/2108.10434).
- [20] S. Choi, I. Loaiza, and A. F. Izmaylov, Fluid fermionic fragments for optimizing quantum measurements of electronic Hamiltonians in the variational quantum eigen-solver, *Quantum* **7**, 889 (2023).
- [21] T.-C. Yen, A. Ganeshram, and A. F. Izmaylov, Deterministic improvements of quantum measurements with grouping of compatible operators, non-local transformations, and covariance estimates, *npj Quantum Inf.* **9**, 14 (2023).
- [22] S. Choi, T.-C. Yen, and A. F. Izmaylov, Improving quantum measurements by introducing “ghost” pauli products, *J. Chem. Theory Comput.* **18**, 7394 (2022).
- [23] S. Tamiya and H. Yamasaki, Stochastic gradient line Bayesian optimization for efficient noise-robust optimization of parameterized quantum circuits, *npj Quantum Inf.* **8**, 90 (2022).
- [24] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, *Phys. Rev. A* **92**, 042303 (2015).
- [25] O. Crawford, B. van Straaten, D. Wang, T. Parks, E. Campbell, and S. Brierley, Efficient quantum measurement of Pauli operators in the presence of finite sampling error, *Quantum* **5**, 385 (2021).
- [26] S. M. Mniszewski, P. A. Dub, S. Tretiak, P. M. Anisimov, Y. Zhang, and C. F. Negre, Reduction of the molecular Hamiltonian matrix using quantum community detection, *Sci. Rep.* **11**, 4099 (2021).
- [27] Z.-J. Zhang, K. Nakaji, M. Choi, and A. Aspuru-Guzik, A composite measurement scheme for efficient quantum observable estimation, [arXiv:2305.02439](https://arxiv.org/abs/2305.02439).
- [28] L. Zhu, S. Liang, C. Yang, and X. Li, Optimizing shot assignment in variational quantum eigensolver measurement, *J. Chem. Theory Comput.* **20**, 2390 (2024).
- [29] S. Liang, L. Zhu, X. Liu, C. Yang, and X. Li, Artificial-intelligence-driven shot reduction in quantum measurement, *Chemical Physics Reviews* **5**, 041401 (AIP Publishing, Melville, NY, 2024).
- [30] S. S. Kahani and A. Nobakhti, A novel framework for shot number minimization in quantum variational algorithms, [arXiv:2307.04035](https://arxiv.org/abs/2307.04035).
- [31] K. Phalak and S. Ghosh, Shot optimization in quantum machine learning architectures to accelerate training, *IEEE Access* **11**, 41514 (2023).
- [32] R. Ayanzadeh, N. Alavisamani, P. Das, and M. Qureshi, Frozen-Qubits: Boosting fidelity of QAOA by skipping hotspot nodes, in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM Press, New York, NY, USA, 2023), Vol. 2, pp. 311–324.
- [33] C. E. Shannon, A mathematical theory of communication, *Bell Sys. Tech. J.* **27**, 379 (1948).
- [34] D. M. Greenberger, M. A. Horne, and A. Zeilinger, Going beyond Bell’s theorem, in *Bell’s Theorem, Quantum Theory and Conceptions of the Universe*, edited by M. Kafatos (Springer, Dordrecht, 1989), pp. 69–72.
- [35] E. Hellinger, Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen, *J. Angew. Math.* **1909**, 210 (1909).
- [36] A. Li, S. Stein, S. Krishnamoorthy, and J. Ang, QASMBench: A low-level quantum benchmark suite for NISQ evaluation and simulation, *ACM Trans. Quantum Comput.* **4**, 1 (2023).
- [37] IBM, IBM quantum, 2024, <https://quantum-computing.ibm.com/>.
- [38] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization, *ACM Trans. Math. Software* **23**, 550 (1997).
- [39] T. Miki, D. Tsukayama, R. Okita, M. Shimada, and J. Shirakashi, Quantum-classical variational approaches with single-qubit operation on near-term quantum processors, in *2022 IEEE 22nd International Conference on Nanotechnology (NANO)* (IEEE, Piscataway, NJ, 2022), pp. 303–306.
- [40] L. Zhao, Q. Wang, J. J. Goings, K. Shin, W. Kyoung, S. Noh, Y. M. Rhee, and K. Kim, Enhancing the electron pair approximation with measurements on trapped-ion quantum computers, *npj Quantum Inf.* **10**, 76 (2024).
- [41] A. Eddins, M. Motta, T. P. Gujarati, S. Bravyi, A. Mezzacapo, C. Hadfield, and S. Sheldon, Doubling the size of quantum simulators by entanglement forging, *PRX Quantum* **3**, 010309 (2022).
- [42] W. J. Huggins, B. A. O’Gorman, N. C. Rubin, D. R. Reichman, R. Babbush, and J. Lee, Unbiasing fermionic quantum Monte Carlo with a quantum computer, *Nature (London)* **603**, 416 (2022).
- [43] T. P. Gujarati, M. Motta, T. N. Friedhoff, J. E. Rice, N. Nguyen, P. K. Barkoutsos, R. J. Thompson, T. Smith, M. Kagele, M. Brei *et al.*, Quantum computation of reactions on surfaces using local embedding, *npj Quantum Inf.* **9**, 88 (2023).

- [44] D. Wu, R. Rossi, F. Vicentini, N. Astrakhantsev, F. Becca, X. Cao, J. Carrasquilla, F. Ferrari, A. Georges, M. Hibat-Allah *et al.*, Variational benchmarks for quantum many-body problems, *Science* **386**, 296 (2024).
- [45] V. Cimini, M. Valeri, S. Piacentini, F. Ceccarelli, G. Corrielli, R. Osellame, N. Spagnolo, and F. Sciarrino, Variational quantum algorithm for experimental photonic multiparameter estimation, *npj Quantum Inf.* **10**, 26 (2024).
- [46] M. A. Rahman, M. S. Akter, E. Miller, B. Timofti, H. Shahriar, M. Masum, and F. Wu, Fine-tuned variational quantum classifiers for cyber attacks detection based on parameterized quantum circuits and optimizers, in *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)* (IEEE, Piscataway, NJ, 2024), pp. 1067–1072.
- [47] M. Wang, P. Das, and P. J. Nair, Qoncord: A multi-device job scheduling framework for variational quantum algorithms, in *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)* (IEEE, Piscataway, NJ, 2024), pp. 735–749.
- [48] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* **5**, 4213 (2014).
- [49] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, *Nature (London)* **549**, 242 (2017).
- [50] H. R. Grimsely, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, *Nat. Commun.* **10**, 3007 (2019).
- [51] X. Liu, A. Angone, R. Shaydulin, I. Safro, Y. Alexeev, and L. Cincio, Layer VQE: A variational approach for combinatorial optimization on noisy quantum computers, *IEEE Trans. Quantum Eng.* **3**, 1 (2022).
- [52] K. Blekos, D. Brand, A. Ceschini, C.-H. Chou, R.-H. Li, K. Pandya, and A. Summer, A review on quantum approximate optimization algorithm and its variants, *Phys. Rep.* **1068**, 1 (2024).
- [53] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, *Phys. Rev. X* **10**, 021067 (2020).
- [54] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [55] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, *Nat. Commun.* **12**, 1791 (2021).
- [56] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, *Quantum* **3**, 214 (2019).