IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 December 2025, accepted 18 December 2025, date of publication 24 December 2025,
date of current version 31 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3647901

SURVEY

Strategies for Noise-Resilient Quantum Approximate Optimization Algorithms: A Review and Classification of Error Mitigation

KIMCHHOR CHIV[1], LEANGHOK HOUR[1], SANGHYEON LEE[1], TARA KIT[1], TAE-KYUNG KIM[2], AND YOUNGSUN HAN[1], (Member, IEEE)

[1]Department of AI Convergence, Pukyong National University, Nam-gu, Busan 48513, South Korea
[2]Department of Management Information Systems, Chungbuk National University, Seowon-gu, Cheongju-si, Chungcheongbuk-do 28644, South Korea

Corresponding author: Youngsun Han (youngsun@pknu.ac.kr)

This work was supported in part by the Institute for Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korea Government [Ministry of Science and ICT (MSIT)] under Grant 2020-0-00014; in part by the National Research Foundation of Korea (NRF) through the Quantum Computing Based on Quantum Advantage Challenge Research funded by Korean Government (MSIT) under Grant RS-2023-00257994; and in part by NRF through the Creation of the Quantum Information Science Research and Development Ecosystem funded by Korean Government (MSIT) under Grant RS-2023-00256050.

ABSTRACT The Quantum Approximate Optimization Algorithm (QAOA) is a leading candidate for solving combinatorial optimization problems on near-term quantum hardware. However, its practical deployment remains severely limited by hardware-induced noise and decoherence. In the absence of full fault tolerance, Quantum Error Mitigation (QEM) techniques offer a viable approach to enhance QAOA performance on noisy intermediate-scale quantum (NISQ) devices. In this work, we present a unified classification of QEM strategies specifically tailored for QAOA, systematically categorized into preprocessing, during execution, and post-processing phases. We further classify QAOA deployment settings—such as hardware-constrained devices, stochastic noise-dominated environments, and dense graphs problem—and recommend appropriate QEM techniques for each. A comparative analysis highlights the trade-offs among mitigation accuracy, resource overhead, and robustness to different noise types. By synthesizing recent advances and identifying open challenges, our paper provides both a comprehensive survey and a practical framework for designing robust, noise-resilient QAOA pipelines on near-term quantum platforms.

INDEX TERMS Combinatorial optimization, noisy intermediate-scale quantum (NISQ), noise resilience, quantum approximate optimization algorithm (QAOA), quantum computing, quantum error mitigation (QEM), variational quantum algorithms (VQAs).

I. INTRODUCTION

The advent of Variational Quantum Algorithms (VQAs) marks a promising frontier in quantum computing, particularly for the current generation of Noisy Intermediate-Scale Quantum (NISQ) devices [1]. VQAs operate in a hybrid quantum–classical fashion, where a parameterized quantum circuit is executed on quantum hardware, and a classical optimizer updates the parameters to minimize a problem-specific cost function [1], [2], [3]. This approach is well-suited for NISQ systems due to their shallower circuit depth and tolerance to limited qubit counts [4].

Among the most prominent VQAs is the Quantum Approximate Optimization Algorithm (QAOA), introduced by Farhi et al. [5]. QAOA is designed to solve combinatorial optimization problems [6], [7], such as MaxCut [8] and

The associate editor coordinating the review of this manuscript and approving it for publication was Siddhartha Bhattacharyya.

other Quadratic Unconstrained Binary Optimization (QUBO) instances [9], by alternating between two types of Hamiltonians: a cost Hamiltonian that encodes the problem and a mixer Hamiltonian that promotes exploration of the solution space [10]. The depth of the circuit, defined by the number of alternating layers p, controls the expressiveness of the ansatz. In theory, higher p leads to better approximation quality [11]; however, this also increases the vulnerability to hardware noise [12].

Despite QAOA's promise, its effectiveness on real devices is significantly hindered by decoherence, readout errors, crosstalk, and other noise sources that plague current quantum processors [13]. These noise processes distort the energy landscape, impair parameter optimization, and degrade the final solution quality. Increasing circuit depth further exacerbates these issues, creating a trade-off between algorithmic power and hardware robustness.

To combat this, QEM has emerged as a practical solution [14]. Unlike Quantum Error Correction (QEC), which requires significant overhead in the form of ancillary qubits and fault-tolerant operations, QEM techniques are lightweight and hardware-compatible. They are especially suited for NISQ-era applications, offering accuracy improvements without requiring full-blown error correction [15].

A variety of QEM techniques have been proposed—ranging from ZNE [16] and symmetry verification [17] to readout error mitigation [18], [19], circuit compilation [20] and more [14], [21]. However, these techniques are often studied in isolation and lack a cohesive framework that integrates them into QAOA workflows across different scenarios and noise models.

This review addresses this gap by presenting a unified classification of error mitigation techniques specifically designed for QAOA based on the recent published paper between 2020 to 2025. We aim to provide a structured understanding of how QEM methods can be categorized, deployed, and optimized based on their placement in the QAOA pipeline (preprocessing, execution, or post-processing). Furthermore, we evaluate their trade-offs in terms of accuracy gains, noise resilience, and resource efficiency.

Our paper makes the following key contributions:

- Classification of QEM Techniques for QAOA: We present a comprehensive, stage-wise classification of QEM techniques specific to the QAOA, organized across preprocessing, circuit execution, and post-processing stages.
- Prescriptive QEM Pipelines for QAOA: We recommend tailored, multi-stage QEM pipelines aligned with three distinct QAOA deployment archetypes.
- In-depth Comparative Analysis of Trade-offs: We provide a structured comparison of QEM techniques, analyzing the critical trade-offs between mitigation accuracy, resource overhead, and robustness to different noise models. This analysis serves as a practical decision-making guide for practitioners deploying QAOA on real hardware.
- Synthesis of Literature with Practical Implementation Insights: This contribution goes beyond a simple, direct comparison of individual methods. We bring together findings from a wide range of recent studies to identify major patterns, extract practical advice for implementation, and create a clear story about the challenges and solutions for noise-resilient QAOA. This overview provides the big-picture understanding needed to design effective, layered strategies for error mitigation.
- Discussion of Future Research Directions: We outline key limitations in current QEM approaches and identify promising future directions, including adaptive mitigation strategies, compiler-level integration, and ML-driven noise-aware pipelines.

This paper is organized as follows. Section II provides an essential background on the QAOA and noise-resilient quantum optimization. In Section III, we propose a classification of error mitigation strategies for QAOA, dividing them into preprocessing, circuit execution, and post-processing techniques, while also discussing their benefits and implementation contexts. Following this, Section IV classifies different QAOA settings—such as hardware-constrained, dense-graph, and general NISQ use cases—and maps them to suitable QEM strategies. Section V then offers a comparative analysis of these QEM methods in terms of their accuracy gains, resource efficiency, and scalability. Subsequently, Section VI outlines open challenges and future research directions, emphasizing the need for adaptive, compiler-integrated, and machine learning-assisted QEM solutions. Finally, Section VII concludes the paper with a summary of key insights and presents a vision for noise-resilient QAOA as a critical enabler of near-term quantum optimization.

II. BACKGROUND

This section provides the foundational knowledge required to understand the challenges of implementing QAOA on near-term quantum hardware. We first review the structure and components of the QAOA algorithm and then discuss the broader context of noise-resilient quantum optimization techniques.

A. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM (QAOA)

The QAOA is a hybrid quantum-classical algorithm designed for solving combinatorial optimization problems. This section details the operational principles of QAOA, from its Hamiltonian formulation to its variational structure and sensitivity to hardware noise.

1) OVERVIEW OF QAOA

The Quantum Approximate Optimization Algorithm (QAOA), introduced by Farhi et al. [5], is a prominent variational quantum algorithm designed to solve combinatorial optimization problems. QAOA operates in the *quantum-classical hybrid*

paradigm, a process clearly illustrated in Figure 1. In this loop, a quantum computer prepares a parameterized quantum state, and a classical optimizer updates the parameters to minimize a given cost function. The algorithm is particularly effective for problems that can be expressed in *Quadratic Unconstrained Binary Optimization* (QUBO) form. Given a cost function $C(x) : \{0, 1\}^n \to \mathbb{R}$, the goal is to find a binary string x^* that minimizes or maximizes $C(x)$. As shown in the "Problem Input" section of Figure 1, the QAOA process begins by constructing a quantum circuit from the problem's Hamiltonian. This involves two key components

- **Cost Hamiltonian** $\hat{H}_C$: Encodes the classical objective function. For instance, in the MaxCut problem, it takes the form

$$\hat{H}_C = \frac{1}{2} \sum_{(i,j)\in E} w_{ij}(I - Z_i Z_j),$$

  where $Z_i Z_j$ is the Pauli-Z operator acting on qubit i and j, respectively. w_{ij} denotes the edge weight between vertices i and j.
- **Mixer Hamiltonian** $\hat{H}_M$: Drives transitions between computational basis states, typically defined as

$$\hat{H}_M = \sum_{i=1}^{n} X_i,$$

  where X_i is the Pauli-X operator on qubit i.

The QAOA ansatz, depicted in the central part of Figure 1, applies the unitaries derived from these Hamiltonians in alternating layers:

$$|\psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \left(\prod_{k=1}^{p} e^{-i\beta_k \hat{H}_M} e^{-i\gamma_k \hat{H}_C} \right) |\psi_0\rangle,$$

where $|\psi_0\rangle = |+\rangle^{\otimes n}$ is the equal superposition over all basis states, and $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ are variational parameters to be optimized.

Following the flowchart, after the circuit is executed, a measurement is performed to estimate the expectation value of the cost Hamiltonian:

$$F_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) | \hat{H}_C | \psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle.$$

A classical optimizer is used to iteratively update the parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ to maximize (or minimize) F_p, depending on problem formulation. This hybrid loop continues until a target is reached, and a final solution is output. The quality of the QAOA solution is typically measured via the *approximation ratio* [22]:

$$\alpha = \frac{F_p(\boldsymbol{\gamma}, \boldsymbol{\beta})}{C_{\max}}.$$

QAOA is flexible and extensible, and several variants have been proposed to improve performance:

- **Warm-start QAOA** leverages classical approximations as initializations [23].
- **Multi-angle QAOA** assigns different parameters to each gate instance [24].

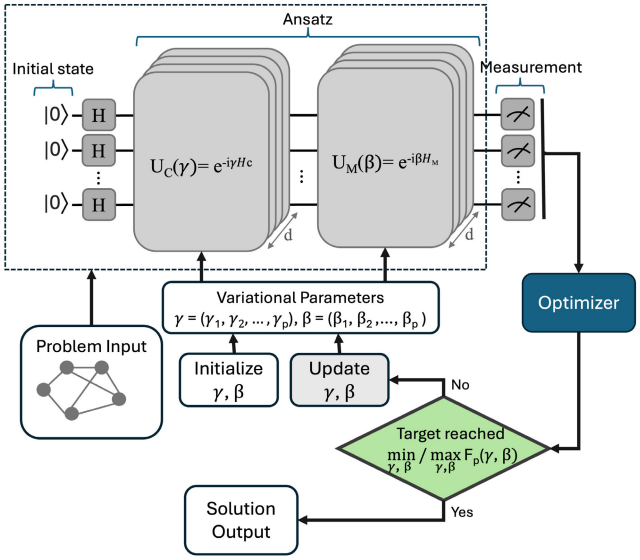

FIGURE 1. Quantum approximate optimization algorithm process flow.The figure illustrates the hybrid quantum-classical loop where a quantum computer executes the ansatz and a classical optimizer updates the variational parameters to minimize the cost function.

- **Adaptive QAOA (ADAPT-QAOA)** iteratively builds the circuit based on gradient feedback [25].
- **Fermionic QAOA and Constraint-preserving Mixers** target domain-specific constraints and quantum simulation [26].
- **Multiscale QAOA (MQAOA)** improves performance by combining shallow QAOA runs with a real-space renormalization group method in a feedback loop. [27].
- **Adaptive-Bias QAOA (ab-QAOA)** modifies the standard mixer Hamiltonian by incorporating iterative, local longitudinal fields (biases). This approach has been shown to yield notable reductions in the circuit depth p required to achieve high accuracy, effectively mitigating noise by shortening the total execution time [28], [29].

2) QAOA PERFORMANCE AND NOISE SENSITIVITY

In the ideal (noiseless) setting, QAOA's performance improves with increasing circuit depth p. As $p \to \infty$, QAOA theoretically approaches the performance of adiabatic quantum computation [5]. However, practical implementation on NISQ hardware introduces several challenges.

a: IMPACT OF CIRCUIT DEPTH

Deeper circuits suffer from greater decoherence, gate infidelity, and readout noise. As each QAOA layer increases the number of quantum gates, especially entangling gates, the cumulative noise can significantly reduce fidelity. While larger p may improve expressivity, empirical results indicate diminishing returns due to hardware noise [30].

b: NOISE MODELS AND HARDWARE LIMITATIONS

NISQ devices experience various types of noise that impact the reliability of quantum computations [4]. Depolarizing and

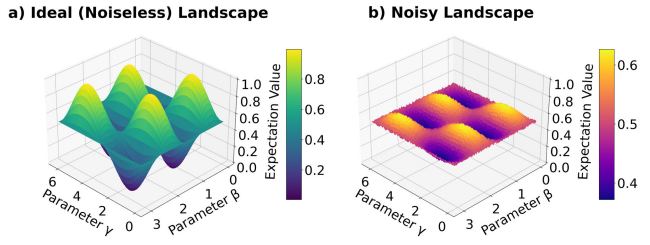

FIGURE 2. Noise-induced corruption of the QAOA optimization landscape for a 2-Qubit system. The figure contrasts the ideal, noiseless landscape (a), which has distinct peaks and valleys, and the noisy landscape (b), where noise flattens the features and obscures the path to the optimal solution.

dephasing noise affect both gate operations and idle qubits by introducing random errors and phase shifts, respectively [31]. Crosstalk leads to unintended interactions between neighboring qubits, often causing correlated errors during multi-qubit operations [31]. Additionally, readout errors introduce bit-flip inaccuracies during measurement, further degrading the fidelity of computational outcomes [31]. These noise effects corrupt the cost expectation values, which the classical optimizer uses for parameter updates. As a result, optimization may stagnate or converge to suboptimal solutions.

c: EFFECT ON OPTIMIZATION LANDSCAPE

The optimization process is most critically undermined by the severe distortion of the cost function landscape due to accumulated noise. A direct visual comparison is provided in Figure 2, which contrasts the ideal computational space with one compromised by real-world hardware effects. Figure 2(a) presents the pristine, theoretical landscape for a 2-qubit MaxCut problem, where prominent features and steep gradients create a clear path to the optimal solution. The corrosive effect of noise is starkly evident in Figure 2(b), where the landscape is visibly flattened and its features are suppressed. This not only blurs the distinction between superior and inferior solutions but also gives rise to the *barren plateaus* phenomenon, where gradients disappear across large areas of the parameter space [32]. Consequently, the optimizer is left to navigate an uninformative, featureless terrain, leading to stagnation or convergence to a poor-quality solution.

Classical Optimization Strategies: The choice of classical optimizer plays a pivotal role in navigating this noise-corrupted landscape. Gradient-free methods (e.g., COBYLA [33], Nelder and Mead [34]) are generally preferable for shallow circuits ($p = 1$) or environments dominated by stochastic noise, as they tolerate measurement fluctuations without the costly overhead of parameter-shift gradient calculations [35]. However, they scale poorly with dimensionality. Conversely, gradient-based methods become essential for deeper circuits ($p \geq 2$) where the parameter space is large, though they must contend with the phenomenon of barren plateaus [32]. While susceptible to these plateaus, stochastic gradient techniques like SPSA are particularly effective in NISQ settings because they approximate the gradient using a constant number of measurements regardless of dimension [13], [36]. Recent innovations in this domain include Quantum Natural Gradient (QNG) approaches that leverage the Hilbert space geometry to accelerate convergence [37], and sophisticated initialization strategies (e.g., warm-starting [23]) designed to bypass barren plateaus entirely.

d: EMPIRICAL EVIDENCE

Several studies have benchmarked QAOA on hardware and simulators, showing that even moderate noise levels can negate quantum advantage in realistic settings [22], [30]. For example, numerical simulations of QAOA on MaxCut instances under realistic noise indicate rapid degradation of the approximation ratio beyond $p = 1$ or $p = 2$ without error mitigation [38].

To address these limitations, recent research has focused on error mitigation techniques, which aim to reduce the impact of noise without requiring full QEC. These include strategies such as zero-noise extrapolation (ZNE) [15], symmetry verification [39], and hardware-aware compilation [40], all of which are further discussed in Section III.

B. NOISE-RESILIENT QUANTUM OPTIMIZATION TECHNIQUES

The practical implementation of quantum algorithms on NISQ devices is fundamentally constrained by hardware-induced noise. As such, enhancing the noise resilience of quantum optimization algorithms—particularly QAOA—is essential to extract meaningful performance on current and near-term quantum systems.

A variety of strategies have been developed to enable *noise-resilient quantum optimization*, broadly falling into three categories: (i) algorithm-level robustness, (ii) quantum error mitigation (QEM), and (iii) hardware-aware circuit optimization. Each of these approaches is designed to improve the fidelity of outcomes without relying on the full overhead of fault-tolerant QEC, which remains out of reach for most current hardware platforms.

1) ALGORITHM-LEVEL ROBUSTNESS

Some approaches aim to improve intrinsic robustness to noise by modifying the structure of the algorithm. For instance, variants of QAOA such as warm-start QAOA [23], adaptive ansätze [25], and low-depth QAOA (LD-QAOA) [22] attempt to reduce circuit depth while maintaining performance. These techniques mitigate the need for long coherent evolutions by leveraging classical priors or optimizing ansatz design.

In particular, LD-QAOA modifies parameter schedules to reduce entangling gate count, thereby suppressing error accumulation while preserving approximation ratios on specific problem classes. Similarly, constraint-preserving

mixers [41] eliminate invalid state transitions and reduce the state space, leading to less noisy expectation evaluations.

2) QUANTUM ERROR MITIGATION (QEM)

QEM refers to a set of techniques that reduce the impact of noise *without requiring ancilla qubits or fault-tolerant encoding* [15], [21]. QEM is particularly relevant to variational quantum algorithms such as QAOA and VQE, where the objective function is estimated via repeated measurements of noisy quantum states [13], [42].

Several notable QEM techniques include:

- **Zero-Noise Extrapolation (ZNE):** Involves executing circuits at scaled noise levels and extrapolating to zero noise using Richardson or polynomial fitting [16], [43], [44].
- **Readout Error Mitigation:** Uses calibration matrices to correct classical bitstring measurement biases [18], [19], [45].
- **Symmetry Verification:** Projects the final state onto the subspace satisfying known symmetries, removing errors that violate conservation laws [17], [39].
- **Probabilistic Error Cancellation (PEC):** Uses knowledge of noise channels to statistically invert their effect, at the cost of increased sampling overhead [15], [46], [47].

These techniques have been integrated into software frameworks such as Mitiq [48] to support automated and backend-aware QEM application.

3) HARDWARE-AWARE CIRCUIT OPTIMIZATIONS

Finally, certain mitigation techniques involve tailoring quantum circuits to the characteristics of the underlying hardware. These include:

- **Gate compilation and layout-aware transpilation:** Reducing SWAP gates and exploiting connectivity maps to minimize error-prone operations [20], [49].
- **Crosstalk-aware scheduling:** Avoiding simultaneous operations on physically adjacent qubits to suppress correlated noise [40].
- **Qubit remapping:** Assigning logical qubits to physical qubits with lower error rates, based on calibration data [50].

These approaches, while not strictly algorithmic or variational, are indispensable for deploying QAOA effectively on real devices.

4) SYNERGISTIC APPROACHES

Importantly, many of these techniques are *complementary* rather than mutually exclusive. For example, applying symmetry verification during circuit execution and ZNE during post-processing can yield compounded benefits. When integrated coherently, such techniques enable variational algorithms like QAOA to achieve higher fidelity with limited circuit depth and measurement shots.

In summary, noise-resilient quantum optimization hinges on the interplay between algorithmic innovations, error mitigation, and hardware-aware execution strategies. The classification of QEM techniques introduced in Section III will further categorize these strategies in a structured manner for QAOA-specific deployments.

III. CLASSIFICATION OF QUANTUM ERROR MITIGATION TECHNIQUES FOR QAOA

Given the inherent susceptibility of QAOA to noise and decoherence, an effective strategy involves implementing QEM techniques throughout the algorithmic pipeline. We present a structured classification of these techniques, categorized into three principal stages: preprocessing, during execution, and post-processing. It is important to recognize that several key techniques are not monolithic but form a complete workflow across multiple stages. For instance, the full ZNE pipeline involves running noise-scaled circuits during execution and then fitting the results in post-processing. These conceptual links are visually highlighted by the dashed frames in Figure 3, which provides a comprehensive overview of the QEM landscape. Table 1 provides a consolidated overview of this framework, outlining the specific techniques, their key benefits, and foundational papers for each stage. This section then systematically examines these approaches, highlighting their roles, benefits, and practical considerations within QAOA workflows.

A. PREPROCESSING TECHNIQUES

Preprocessing techniques are implemented prior to executing the quantum circuits. Their primary goal is to enhance noise resilience by optimizing circuit architecture, gate sequences, and qubit assignments. This approach, visually summarized in Figure 3(a), minimizes error accumulation from the outset and directly influences subsequent QAOA performance.

1) ANSATZ DESIGN OPTIMIZATION

Ansatz design optimization offers a proactive, preprocessing strategy for enhancing noise resilience by fundamentally re-engineering the QAOA circuit's structure [51], [52]. These methods modify the circuit based on the problem's topology or its underlying mathematical properties to reduce depth, gate count, or sensitivity to specific types of errors [51], [52].

a: SPANNING TREE HEURISTICS FOR DEPTH REDUCTION

A significant challenge in ansatz design is balancing the reduction of CNOT gates with the resulting increase in circuit depth [51]. Majumdar et al. [51] address this by proposing a greedy heuristic that constructs a low-depth spanning tree of the problem graph. Previous methods could reduce CNOT count by $n-1$ but often created deep circuits by following a Depth First Search (DFS) tree, making them prone to decoherence [51]. The key innovation is a cost function, shown in Equation (1), that guides the tree's construction to favor branching at lower levels, creating a trade-off between the tree's height and its branching factor to minimize

Algorithm 1 Cost Function Based Rooted Spanning Tree Generation [51]

Require: A Graph $G = (V, E)$, maximum branching factor B.
Ensure: A Rooted Spanning Tree T of the Graph G.

$T \leftarrow \{\}$
$u_{\text{bf}} \leftarrow 0$ for all vertex u.
$r \leftarrow$ randomly selected start vertex.
Visited $\leftarrow \{r\}$
$r_{\text{bf}} \leftarrow r_{\text{bf}} + 1$.
edges_to_add $\leftarrow$ neigh(r).
while |Visited| $< n$ **do**
  $e \leftarrow$ edges_to_add[0].
  $c \leftarrow 0$
  for all edge $= (u, v) \in$ edges_to_add **do**
    cost $\leftarrow (n - l_u) \cdot (B - u_{\text{bf}})$.
    if cost $> c$ **then**
      $c \leftarrow$ cost.
      $e \leftarrow$ edge.
  $T \leftarrow T \cup \{e\}$.
  Visited $\leftarrow$ Visited $\cup \{y\}$, where $e = (x, y)$.
  $x_{\text{bf}} \leftarrow x_{\text{bf}} + 1$.
  Remove all edges of the form $(*, y)$ from edges_to_add.
  for all edge $= (p, q) \in$ neigh(y) **do**
    if $q \notin$ Visited **then**
      edges_to_add $\leftarrow$ edges_to_add $\cup$ {edge}.

"delayed starts" in the circuit schedule [51].

$$C_v = (n - l_v) \cdot (B - v_{\text{bf}}) \tag{1}$$

Here, n is the number of vertices, l_v is the level of vertex v, v_{bf} is its current branching factor, and B is the maximum allowed branching factor. Algorithm 1 by Majumdar et al. [51] details the procedure for generating the rooted spanning tree using this cost function. By optimizing the tree structure, this method retains the $n - 1$ CNOT gate reduction while significantly reducing the circuit depth, leading to a more than 10-fold increase in success probability in noisy simulations.

b: PARITY ARCHITECTURE RE-ENCODING

A distinct paradigm demonstrated by Weidinger et al. [52] is to re-encode the entire optimization problem using the parity architecture (also known as the Lechner-Hauke-Zoller or LHZ scheme). This technique transforms a problem with all-to-all connectivity among N logical qubits into a larger, planar layout of $K = N(N - 1)/2$ physical qubits with only local interactions. This fundamental alteration of the ansatz obviates the need for complex SWAP networks that add noise and gate overhead [52]. The central contribution is an error mitigation strategy that modifies the QAOA objective function itself. Instead of minimizing the energy of the physical qubits, the algorithm minimizes the average energy of *decoded logical states* [52]. The cost function is computed by taking the trace of the problem Hamiltonian H_p with the decoded density matrix, averaged over a set of M spanning trees T, as shown in the equation below [52]:

$$\mathcal{C}^{(T)}(\beta, \gamma, \Omega) = \frac{1}{M} \sum_{m=1}^{M} \text{Tr}(H_P D_{t_m}[\tilde{\rho}(\beta, \gamma, \Omega)]) \tag{2}$$

This approach exploits the redundancy of the parity encoding; errors that violate the encoding's constraints are detected and corrected during the decoding step, making the optimization process itself robust to noise [52]. This method achieves a higher success probability than standard QAOA in noisy simulations, even when requiring more CNOT gates, because the ansatz is inherently more resilient [52].

c: ADAPTIVE-BIAS ANSATZ CONSTRUCTION

Within the scope of ansatz design, modifying the Hamiltonian structure itself serves as a potent error mitigation strategy. The Adaptive-Bias QAOA (ab-QAOA) [28], [29] modifies the standard transverse mixer $H_M = \sum X_j$ by incorporating iterative, local longitudinal fields (biases) to steer the evolution more effectively. Mathematically, the mixer Hamiltonian at layer l is transformed to:

$$H_M^{(l)} = \sum_{j=1}^{n} \left(X_j + h_{j,l} Z_j\right) \tag{3}$$

where $h_{j,l}$ represent adaptive bias parameters that are updated based on the gradients or measurement outcomes of previous steps [28], [29]. This modification breaks the symmetry constraints that often trap standard QAOA in local minima and significantly reduces the required circuit depth p to achieve a target approximation ratio. In our classification, this fits as a preprocessing technique because it fundamentally alters the ansatz definition to be inherently more robust to decoherence—by minimizing the gate count and total execution time required to reach solution convergence.

2) GATE REORDERING AND COMPILATION

Effective gate reordering and compilation are pivotal preprocessing steps explicitly designed to minimize depth and noise susceptibility through intelligent circuit transformations and execution scheduling [53], [54]. Given the significant error contributions from two-qubit operations, reordering gates to exploit parallelism and commutativity is particularly beneficial for QAOA [53], [54].

a: EQUIVALENT CIRCUIT AVERAGING FOR COHERENT ERROR MITIGATION

A key challenge in compilation is that systematic coherent errors can interfere constructively, leading to worse performance than predicted by randomized benchmarking [53]. To address this, Hashim et al. [53] introduced Equivalent Circuit Averaging (ECA), a technique that randomizes the circuit implementation to suppress such errors. The core of the method is to generate a set of logically equivalent but

physically distinct circuits by sampling from the various degrees of freedom available during compilation—such as different decompositions for the ZZ-SWAP gate. By executing each of these unique circuit variants for a fraction of the total shots and averaging the results, the systematic errors are effectively "washed out," leading to an incoherent, stochastic error channel that is easier to predict and manage [53]. This randomization is performed intelligently to preserve gate-level optimizations as much as possible, achieving an average error reduction of approximately 60% in total variation distance for $p = 1$ QAOA on a four-qubit superconducting processor [53].

This approach is particularly advantageous in regimes dominated by coherent noise sources, where fixed systematic errors would otherwise accumulate constructively. Functionally, ECA serves as a specific instance of noise tailoring: by averaging over logically equivalent implementations, it converts coherent systematic errors into incoherent stochastic noise [53]. However, this fidelity gain comes with an overhead in terms of compilation complexity (generating N unique circuit variants) and potentially the total number of shots required to maintain statistical precision given the reduction in success probability. Recent work by Marshman et al. [55] has formalized this behavior within the broader framework of unitary averaging. Mathematically, if one averages over N noisy implementations $\hat{U}_j = \hat{U}_T + \hat{E}_j$ (where $\hat{U}_T$ is the target unitary and $\hat{E}_j$ is the error), the effective transformation $\hat{\mathcal{U}}$ becomes:

$$\hat{\mathcal{U}}(N) = \frac{1}{N}\sum_{j=1}^{N}\hat{U}_j = \hat{U}_T + \frac{\hat{\epsilon}(N)}{N} \tag{4}$$

where $\hat{\epsilon}(N)$ represents the stochastic error contribution. This relationship explicitly demonstrates that while the target unitary is preserved, the error amplitude scales as $1/N$ [55]. Marshman et al. further note that while this averaging suppresses errors, it inherently requires a trade-off, potentially resulting in a finite decrease in the probability of success or necessitating increased resource redundancy to maintain fault tolerance [55].

b: VARIATION-AWARE REORDERING FOR CIRCUIT SUCCESS PROBABILITY

The commutativity of CPHASE gates in the QAOA cost Hamiltonian provides a powerful degree of freedom for compilation [54]. Alam et al. [54] leverage this property with a hardware-aware, multi-stage compilation policy designed to maximize the Circuit Success Probability (CSP) on devices with known gate-fidelity variations. Their procedure involves two main stages:

- **Instruction Parallelization**: First, the commutative CPHASE gates are grouped into the minimum possible number of parallel layers. This is framed as a bin-packing problem and solved using a first-fit decreasing heuristic, which sorts the logical qubits by their degree (number of CPHASE operations) and schedules the gates accordingly [54].
- **Variation-aware Iterative Mapping (VIM)**: After parallelization, the VIM procedure iteratively reorders these layers to find the sequence that yields the highest CSP. Starting with an initial layer order, the algorithm explores all possible two-layer swaps, compiles the circuit for each new order using a standard backend compiler, and calculates the resulting CSP based on hardware calibration data. The order that provides the best improvement becomes the new "root" for the next iteration. This process repeats until no further improvement in CSP is found [54].

This two-stage process, combined with an intelligent initial qubit placement (VQP), was shown to improve the circuit success probability by an average of 8.408X compared to the baseline compiler for a set of QAOA-MaxCut problems [54]. It should be noted that such substantial gains were specifically reported for circuit success probability on the 15-qubit ibmq_16_melbourne superconducting processor, using QAOA-MaxCut instances on 15-node erdos-renyi random graphs where gate error rates exhibited significant spatial variability [54].

However, the effectiveness of these compilation strategies strongly depends on hardware-specific features such as native gate sets, coherence times, and connectivity, thus requiring adaptive compilation strategies that dynamically consider real-time calibration data and hardware constraints [53], [54]. A critical limitation remains: optimal compilation typically demands significant classical pre-processing computational resources, potentially undermining scalability for larger-scale problem instances [54].

Algorithm 2 Factoring Out Semi-Symmetries [56]

Require: QUBO matrix Q, numAncillas, penalty z.

$n_{\text{new}} \leftarrow n$
cL $\leftarrow$ getConflictList(Q, n_{new})
while len(cL) $>$ 0 **do**
  syms, (i, j) $\leftarrow$ getMostSymQubits(Q, n_{new}, cL)
  if len(syms) $<$ 3**or**$n_{\text{new}} == n +$ numAncillas **then**
    break
  $n_{\text{new}} \leftarrow n_{\text{new}} + 1$
  $Q \leftarrow$ enhance(Q, n_{new}, (i, j), syms)
  cL $\leftarrow$ getConflictList(Q, n_{new})
return Q

3) SYMMETRY EXPLOITATION

Symmetry exploitation has emerged as a compelling preprocessing strategy, leveraging inherent problem symmetries to achieve profound circuit simplifications and noise reductions before execution [56]. The central premise is that by identifying and factoring out these symmetries, the complexity of the quantum implementation can be drastically reduced [56].

A prime example of this is the constructive approach introduced by Nüßlein et al. [56], which simplifies the QUBO matrix itself by identifying and factoring out a novel, weaker form of symmetry called ''semi-symmetries'' [56]. Two conflicting qubits (i, j) are defined as semi-symmetric if they share identical non-zero couplings to a shared subset U of at least three other qubits [56]. This relationship is formally defined as:

$$\exists U \subseteq \{1..n\} \setminus \{i, j\} \wedge |U| \geq 3 : \forall k \in U : Q_{i,k} = Q_{j,k} \neq 0 \tag{5}$$

The method, detailed in Algorithm 2, iteratively finds the qubit pair with the largest semi-symmetry and modifies the QUBO matrix by introducing an ancilla qubit [56]. This new ancilla effectively absorbs the shared couplings, which are then removed from the original qubits [56]. The 'enhance' function in the algorithm details how the QUBO matrix is rewritten to enforce the constraint that the ancilla is active if either of the original qubits is active [56]. This preprocessing technique directly reduces the number of non-zero couplings in the QUBO matrix, which in turn decreases the CNOT gate count and depth of the QAOA circuit [56]. Experiments demonstrated reductions in circuit couplings by up to 49% and circuit depth by up to 41% across a range of standard optimization problems [56].

While this symmetry-based preprocessing method has proven impactful, it requires a sophisticated classical algorithm for symmetry detection and introduces additional quantum resources in the form of ancilla qubits, thus posing resource overhead considerations in hardware-constrained contexts [56].

4) QUBIT MAPPING AND ROUTING OPTIMIZATION

Qubit mapping and routing optimization techniques are essential for addressing the limited qubit connectivity on current NISQ hardware, which often necessitates excessive SWAP gate insertions—greatly increasing circuit depth and cumulative noise [54], [57], [58]. These constraints severely impact the success probability of quantum algorithms such as QAOA [57], [58].

Recent advancements have focused on finding optimal initial mappings of logical qubits to physical qubits to minimize this overhead. A highly scalable approach, presented by Matsuo et al. [57], formulates the initial mapping problem as a Boolean Satisfiability (SAT) problem [57]. This technique determines whether a program graph $P = (V, E)$ can be embedded into a hardware connectivity graph C_l, which is the effective graph after applying l layers of a predetermined swap strategy [57]. The problem is translated into a SAT instance using boolean literals $x_{i,j}$, where $x_{i,j}$ is true if program qubit i is mapped to physical qubit j. The mapping is constrained by three conditions [57]:

1) Each program qubit must be assigned to exactly one physical qubit. This is enforced by clauses ensuring each program qubit is assigned to at least one physical qubit,

$$\bigvee_{j=0}^{|V_l|-1} x_{i,j} = 1, \tag{6}$$

and to at most one physical qubit,

$$\bigwedge_{j>k} (\neg x_{i,j} \vee \neg x_{i,k}) = 1. \tag{7}$$

2) Each physical qubit can be occupied by at most one program qubit.

$$\bigwedge_{i>j} (\neg x_{i,k} \vee \neg x_{j,k}) = 1. \tag{8}$$

3) Adjacent program qubits must be mapped to adjacent physical qubits in the effective graph C_l. For every edge $(i, j) \in E$, this is expressed as:

$$\neg x_{i,k} \vee \Big(\bigvee_{(k,k') \in E_l} x_{j,k'} \Big). \tag{9}$$

Algorithm 3 Binary Search for Optimal Initial Mapping [57]

Require: Program graph $P = (V, E)$, swap strategy $\mathcal{S}$.
Ensure: Initial layout that minimizes the number of swap layers.

```
l_L ← 0
l_R ← |V| − 2
while l_L < l_R do
    l ← ⌊(l_L + l_R)/2⌋
    Create connectivity graph C_l from S
    Create SAT instance SAT_embed(P, C_l)
    if SAT_embed(P, C_l) is satisfiable then
        l_R ← l
    else
        l_L ← l + 1
```

By leveraging powerful classical SAT solvers, this formulation can efficiently determine if a valid mapping exists for a given number of swap layers, l [57]. To find the *minimum* required l, Matsuo et al. employ a binary search algorithm, shown in Algorithm 3, which solves $\mathcal{O}(\log n)$ SAT instances to find an optimal initial mapping. This technique achieved a remarkable 65% reduction in CNOT gate count for 500-node random regular graphs [57]. Note that such high reduction factors are specific to sparse graph topologies (e.g., 3-regular graphs) where the initial mapping significantly influences the swap overhead, and gains may diminish for denser or fully-connected problem instances [57]. The practical utility of this method was later demonstrated by Sack and Egger [58], who applied it to successfully execute large-scale QAOA on nonplanar graphs with up to 40 qubits, a task that would have been infeasible otherwise.

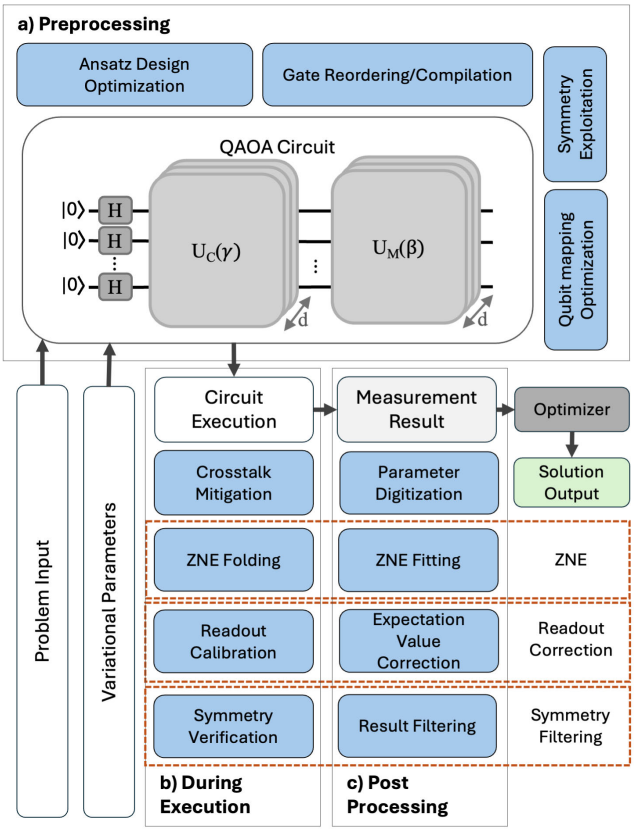

FIGURE 3. Overview of QEM techniques for the QAOA. The diagram classifies QEM strategies into three distinct stages: a) Preprocessing techniques, b) During execution techniques, and c) Post-processing techniques. The dashed frames highlight how specific techniques are conceptually linked and applied across multiple stages, illustrating the complete workflow for methods such as ZNE, Readout correction, and Symmetry-based filtering.

Other approaches, such as those introduced by Alam et al. [54], utilize variation-aware compilation methodologies that embed hardware-aware logic directly into the QAOA transpilation flow to improve circuit reliability under noisy conditions. Despite the success of these various methods, practical deployment of routing strategies remains challenged by the substantial classical overhead required for the optimization and the need for frequent recalibration due to hardware drift.

B. DURING EXECUTION TECHNIQUES

Circuit execution techniques, situated in the workflow as shown in Figure 3(b), aim to directly address and mitigate errors that arise during the physical execution of QAOA circuits on NISQ hardware. Unlike preprocessing strategies, these methods actively intervene or calibrate operations executed during runtime to maintain computational fidelity. They are critical in ensuring that quantum states prepared and manipulated in hardware closely match the intended theoretical quantum states.

This section critically examines four primary circuit execution-based QEM techniques: Zero-Noise Extrapolation (ZNE), Readout Error Mitigation, Crosstalk Mitigation, and Symmetry Verification.

1) ZERO-NOISE EXTRAPOLATION (ZNE) FOLDING

Zero-Noise Extrapolation is a prominent QEM technique explicitly designed to reduce the systematic errors of quantum circuits by artificially amplifying hardware noise to multiple levels and subsequently extrapolating to estimate the zero-noise limit [43]. The "folding" stage of ZNE occurs during circuit execution and does not require additional qubit overhead, making it particularly attractive for resource-limited NISQ devices. The core method involves evaluating multiple, logically equivalent copies of the quantum circuit, each with an increased depth to amplify the noise intensity; this noise scaling is often achieved through techniques such as gate folding [59]. After collecting the noisy expectation values at these various noise levels, a classical extrapolation function is applied to the data points to predict the ideal, zero-noise expectation value [59]. Venere et al. [59] rigorously analyzed the effects of different gate-folding strategies within a QAOA workflow, illustrating how this controlled noise insertion, when combined with extrapolation models like Richardson extrapolation, can yield robust, noise-resilient expectation values. However, their study also noted the practical challenges of ZNE, including the optimal selection of scaling factors and extrapolation models, as well as sensitivity to coherent noise sources, which can undermine its effectiveness for complex QAOA circuits [59].

A significant practical challenge identified in the literature remains the optimal selection of extrapolation models and scaling factors, as overly simplistic extrapolations might underestimate noise, while overly complex extrapolations risk introducing additional numerical instability into the mitigation procedure.

2) READOUT CORRECTION: CALIBRATION

Readout errors, which arise during quantum state measurement, substantially impact the accuracy of results from QAOA circuits by distorting the final probability distribution of outcomes [60].

Readout Correction techniques are classified as circuit execution methods because they require active intervention at runtime. Specifically, they involve executing dedicated calibration circuits on the quantum hardware to characterize the measurement noise in real-time, distinguishing them from purely classical post-processing methods. The relationship between the ideal (noiseless) probability distribution p^{ideal} and the experimentally measured noisy distribution p^{noisy} can be described by a linear transformation involving a stochastic noise matrix Λ [60]:

$$p^{\text{noisy}} = \Lambda p^{\text{ideal}} \tag{10}$$

The core mitigation strategy involves first characterizing the matrix Λ and then applying its inverse to the measured data to recover an estimate of the ideal distribution [60]. However,

since the direct application of Λ^{-1} may yield unphysical, negative probabilities, a common practice is to find the closest valid probability distribution p via an optimization [60]:

$$p = \arg\min_{q} ||q - \Lambda^{-1} p^{\text{noisy}}||_2^2 \tag{11}$$

where the minimization is over all valid probability distributions q. The main challenge is that characterizing a general Λ for N qubits requires a number of measurements that scales exponentially, $O(2^N)$ [60].

To address this scalability bottleneck, recent approaches utilize tensored calibration methods that assume noise locality [61]. By characterizing the readout error of individual qubits or small disjoint clusters (e.g., via the M-matrix approach), the full calibration matrix Λ can be approximated as a tensor product of smaller, tractable matrices. This reduces the calibration overhead from exponential to linear or polynomial in the number of qubits, making readout mitigation feasible for large-scale QAOA instances on current hardware [61].

3) CROSSTALK MITIGATION

Crosstalk Mitigation is classified as a circuit execution technique because it relies on active, real-time characterization of the hardware to build an accurate model of these correlated errors [60]. The process begins by explicitly modeling the correlated noise, going beyond the assumption of independent noise channels to capture local and multi-qubit correlations that arise from physical coupling [60]. A key method for this was proposed by Maciejewski et al. [60], who use a technique called Diagonal Detector Overlapping Tomography (DDOT) to efficiently characterize these correlations during runtime.

From the DDOT data, a correlation metric $c_{j\to i}$ is calculated to quantify how strongly the state of qubit j affects the measurement noise on qubit i [60]:

$$c_{j\to i} = \frac{1}{2} ||\Lambda_{Q_i}^{Y_j=\prime 0\prime} - \Lambda_{Q_i}^{Y_j=\prime 1\prime}||_{1\to 1} \tag{12}$$

where $\Lambda_{Q_i}^{Y_j=\prime s\prime}$ is the single-qubit noise matrix for qubit i conditioned on qubit j being in state s. Based on this metric, qubits are assigned to ''clusters'' (strongly correlated groups) and ''neighborhoods'' (qubits that weakly influence a cluster) [60]. By calibrating and correcting for these modeled correlations, the fidelity of measured expectation values is substantially improved, with experiments on IBM hardware showing an error reduction by a factor of over 22 [60]. It is important to note that this magnitude of improvement is setting-dependent, specifically observed on the 15-qubit IBM Melbourne superconducting processor where significant non-local readout crosstalk was present, and actual gains may vary depending on the device's connectivity and noise profile [60]. However, comprehensive crosstalk mitigation remains challenging, as capturing these errors requires executing large sets of calibration experiments whose complexity can scale unfavorably with system size [60].

4) SYMMETRY FILTERING: VERIFICATION

The Symmetry Filtering workflow begins with a verification step during circuit execution to detect quantum states that have been corrupted by noise [62], [63]. This verification stage leverages known problem-specific or algorithmic symmetries. If an error pushes the quantum state outside the valid symmetry subspace, a verification procedure can project it back, thereby enhancing the final state fidelity [62]. For a noisy state ρ and an ideal noiseless state $|\psi\rangle$ that lies within the symmetry subspace, the fidelity of the post-projection state ρ_s is guaranteed to be greater than or equal to the original fidelity, as shown below [62]:

$$\mathbf{Tr}[|\psi\rangle\langle\psi|\rho_s] \geq \mathbf{Tr}[|\psi\rangle\langle\psi|\rho]. \tag{13}$$

This projection can be implemented intermittently between QAOA layers using mid-circuit measurements [63]. Botelho et al. [63] propose two distinct strategies to realize this:

- **Ancilla-Based Filtering**: This approach uses an ancilla qubit to check if a symmetry is preserved [63]. A controlled operation, corresponding to the symmetry operator, is applied from the main register to an ancilla qubit. The ancilla is then measured mid-circuit. If the measurement outcome indicates that the symmetry has been violated, the entire computation is discarded, and the qubits are reset for the next run [63]. This method, also foundational to the post-processing technique in [62], becomes a runtime technique when applied between QAOA layers.
- **Ancilla-Free Compression**: As a more resource-efficient alternative, this method avoids the need for extra ancilla qubits [63]. It uses a unitary transformation, U_{compress}, which is designed to map the entire valid subspace S to a smaller, known subspace, while mapping invalid states elsewhere. For example, it can map the valid subspace to states where a subset of qubits is always in the $|0\rangle$ state, as shown in Equation 14 from the paper [63]:

$$U_{\text{compress}}|s\rangle = \begin{cases} |\phi_s\rangle, & s \in S \\ |\psi_s\rangle, & s \notin S \end{cases} \tag{14}$$

  A mid-circuit measurement is then performed on the qubits expected to be $|0\rangle$. If any of these qubits are measured as $|1\rangle$, the state is identified as invalid and the run is discarded. If the check passes, the compression is reversed with $U_{\text{compress}}^{\dagger}$ before the computation continues [63].

Despite their promise, these methods introduce additional complexity and overhead. They require sophisticated hardware capabilities, such as reliable mid-circuit measurements and qubit reset, and reduce the effective throughput of the quantum computer by discarding a subset of the executed runs [62], [63].

C. POST-PROCESSING TECHNIQUES

Post-processing techniques focus on improving the output of quantum computations after circuit execution has concluded. As illustrated in Figure 3(c), these methods operate entirely on collected measurement data or optimized parameters, aiming to correct or refine the final results without requiring additional quantum operations. In QAOA, where expectation values derived from measurements are central to optimization, post-processing techniques play a crucial role in reducing error accumulation and enhancing solution quality. This section reviews four major post-processing strategies: ZNE fitting, Result filtering via symmetry constraints, Expectation value correction, and Digitization of variational parameters.

1) ZERO-NOISE EXTRAPOLATION (ZNE) FITTING

ZNE fitting is the crucial post-processing step that follows the collection of noisy measurement data at different, artificially amplified noise levels [59]. After gathering circuit outputs at scaled noise settings (e.g., via gate folding), a classical extrapolation algorithm is applied to predict the zero-noise limit of the expectation values [59].

Venere et al. [59] systematically demonstrated ZNE fitting within a QAOA workflow, primarily using **Richardson extrapolation**. This method fits a polynomial model to the data points (c_i, E_i), where c_i is the noise scale factor and E_i is the measured expectation value at that noise level. The goal is to evaluate this model at the zero-noise point ($c = 0$) to estimate the ideal, error-free expectation value. For a set of noisy expectation values $\{E(c_1), E(c_2), \ldots, E(c_k)\}$, the extrapolated zero-noise value $E(0)$ is found by solving a system of linear equations derived from the polynomial ansatz, effectively finding the y-intercept of the fitted curve. By applying this technique, Venere et al. [59] successfully improved QAOA approximation ratios even on noisy hardware.

However, ZNE fitting is sensitive to the number of noise-scaled data points collected and to the choice of extrapolation model (e.g., Richardson vs. Linear) [59]. Poor fitting can introduce numerical instabilities, particularly when using high-degree polynomials or when extrapolating far from the available data points [59]. Thus, careful model selection is critical for the reliable application of ZNE fitting in practical QAOA deployments.

2) SYMMETRY FILTERING: RESULT FILTERING

This is the second stage of the Symmetry Filtering workflow, where the information from the verification step is used to classically filter the final data [62]. Result filtering leverages known symmetries to selectively discard measurement outcomes that have been corrupted by noise. By post-selecting results that preserve symmetries such as bit-flip invariance, overall measurement fidelity can be improved [62].

This method, implemented for QAOA by Shaydulin and Galda [62], involves applying symmetry checks post hoc to the collected measurement data. In practice, a verification circuit is appended to the main QAOA evolution, which uses an ancilla qubit to check if a symmetry is violated. After running the circuit for a total of N_{shots}, the mitigation is performed by classically filtering the results. The mitigated expectation value, $\langle C \rangle_{\text{mitigated}}$, is then calculated as the sample mean over only the valid outcomes where the symmetry check passed:

$$\langle C \rangle_{\text{mitigated}} = \frac{1}{N_{\text{valid}}} \sum_{x_i \in X_{\text{filtered}}} f(x_i) \tag{15}$$

Here, X_{filtered} is the subset of measurement outcomes $\{x_i\}$ for which the corresponding ancilla measurement indicated that the symmetry was preserved, and $N_{\text{valid}} = |X_{\text{filtered}}|$ is the total number of such valid shots. While this filtering effectively boosts fidelity, it introduces a trade-off between error robustness and measurement efficiency, as it reduces the number of usable samples and thus may require an increased number of total shots to achieve statistical precision [62].

3) EXPECTATION VALUE CORRECTION

Expectation value correction is a post-processing technique that directly adjusts the measured observables to account for known systematic errors, typically arising from gate inaccuracies or decoherence effects [58]. Unlike filtering methods, this approach enhances result fidelity without discarding any measurement samples [58].

A powerful implementation of this technique, demonstrated by Sack and Egger [58], uses a trained neural network to map noisy expectation values to their corrected counterparts. The method involves training a feed-forward neural network (FFNN) on a dataset of classically simulable circuits. The FFNN learns a function that takes a vector of noisy observables (e.g., all single- and two-qubit Pauli-Z expectation values, $\langle \sigma_i^z \rangle_{\mathcal{N}}$ and $\langle \sigma_i^z \sigma_j^z \rangle_{\mathcal{N}}$) as input and outputs a vector of error-mitigated correlators, $\langle \sigma_i^z \sigma_j^z \rangle_{\mathcal{M}}$, corresponding to the edges in the problem graph [58].

Once the FFNN is trained, it is used as a classical post-processing step during the QAOA optimization loop. After each circuit execution, the noisy observables are measured and fed into the trained FFNN. The corrected cost function, $E(\beta, \gamma)_{\mathcal{M}}$, is then constructed by summing the network's outputs [58]:

$$E(\beta, \gamma)_{\mathcal{M}} = \sum_{(i,j) \in E} \langle \sigma_i^z \sigma_j^z \rangle_{\mathcal{M}} \tag{16}$$

This mitigated energy value is then passed to the classical optimizer. This data-driven approach was shown to successfully correct noise distortions in real-time QAOA runs, enabling improved convergence and the execution of circuits on an unprecedented scale of up to 40 qubits [58]. However, the success of this method depends critically on the quality of the training data and the ability of the chosen circuits

to accurately represent the noise characteristics of the target QAOA circuits [58].

Complementing neural network approaches, regression-based strategies such as Clifford Data Regression (CDR) [64] and its variable-noise variant (vnCDR) [65] have also been developed. CDR operates by constructing a linear error model derived from a dataset of training circuits that are structurally similar to the target circuit but computationally tractable to simulate (typically Clifford-dominated circuits) [64]. By comparing the noisy quantum measurements of these training circuits (x^{train}) against their exact classical values (y^{train}), a regression function is learned to map noisy observables to their ideal counterparts via a linear ansatz:

$$f(x^{\text{noisy}}) = \alpha x^{\text{noisy}} + \beta \tag{17}$$

where α and β are fitted parameters minimizing the training loss. vnCDR extends this by incorporating training data collected at multiple variable noise levels (e.g., via pulse stretching or identity insertion) [65]. Unlike CDR's scalar correction, vnCDR learns a multi-dimensional predictor:

$$f(\mathbf{x}^{\text{vec}}) = \mathbf{w} \cdot \mathbf{x}^{\text{vec}} \tag{18}$$

where $\mathbf{x}^{\text{vec}}$ is a vector of noisy expectation values measured at different noise scales $(c_1, c_2, \dots)$ and $\mathbf{w}$ is the learned weight vector [65]. In the context of the QAOA pipeline, these methods plug in as a hybrid post-processing step. Since the QAOA ansatz consists of non-Clifford rotations parameterized by γ and β, training data is generated by temporarily replacing these continuous parameters with Clifford angles (e.g., multiples of $\pi/2$) [64]. The quantum hardware executes these ''near-Clifford'' QAOA instances to gather noisy data, which is then regressed against the classical ground truth to learn the error model parameters, which are finally applied to mitigate the actual QAOA run.

Finally, a distinct approach to correcting expectation values involves targeting algorithmic errors rather than just hardware noise. Variance-Based Extrapolation [66] is a post-processing technique that exploits the physical relationship between the energy expectation value E and the energy variance $\Delta_H^2 = \langle H^2 \rangle - \langle H \rangle^2$. Since the energy of a variational state scales linearly with its variance near an eigenstate, one can measure E and Δ_H^2 at different parameter settings and linearly extrapolate to the zero-variance limit ($\Delta_H^2 \to 0$) to recover the corrected ground state energy. Crucially, this method has been shown to remain robust even in the presence of hardware noise [66], offering a complementary correction path that requires no additional quantum resources beyond the measurement of variance.

4) DIGITIZATION OF VARIATIONAL PARAMETERS

Digitization of variational parameters is a technique designed to mitigate the effects of analog control noise, such as the limited precision in setting gate rotation angles [67]. As analyzed by Quiroz et al. [67], this method directly addresses the finding that stochastic precision errors can cause an exponential degradation in QAOA performance.

The strategy involves a post-processing step where the continuous parameters, γ and β, found by the classical optimizer are quantized into a discrete binary representation before being implemented in the circuit. A unitary operator like $e^{-i\gamma H_C}$ is then compiled into a product of smaller, discrete gate operations corresponding to this binary expansion [67]. For an N_γ-bit representation of the angle γ, the unitary is decomposed as follows:

$$U_C(\gamma) = e^{-i\gamma H_C} \approx \prod_{j=1}^{N_\gamma} \left[\exp\left(-i \frac{2\pi}{2^{N_\gamma}} 2^{j-1} H_C \right) \right]^{A_j} \tag{19}$$

where $A_j \in \{0, 1\}$ are the bits of the binary representation of γ. This approach makes the algorithm robust against random, time-varying precision errors, which are otherwise difficult to correct [67].

While this technique improves the stability of the algorithm, it introduces a critical trade-off. The primary cost of digitization is an increase in circuit depth [67]. Quiroz et al. show that achieving a desired parameter precision for a p-layer QAOA requires a number of bits that grows logarithmically with p and the target accuracy, ϵ. Consequently, the circuit depth increases proportionally, balancing the benefit of noise resilience against the potential for increased decoherence in a deeper circuit [67].

D. IMPLICATIONS OF THE CLASSIFICATION

The classification of QEM techniques for QAOA circuits, as outlined in Sections III-A through III-C and summarized in Table 1, offers a structured framework for systematically analyzing and deploying mitigation strategies based on circuit characteristics, hardware constraints, and optimization goals. This classification has several important implications for both researchers developing new error mitigation methods and practitioners applying QAOA to real-world problems.

1) GUIDING TECHNIQUE SELECTION BASED ON CIRCUIT AND HARDWARE PROPERTIES

One major benefit of the classification is that it enables targeted selection of QEM techniques according to the specific features of a given QAOA instance. For example:

If circuit depth is the primary limitation due to decoherence, preprocessing techniques such as ansatz design optimization or gate reordering (Section III-A) should be prioritized.

If measurement fidelity is the bottleneck, circuit execution techniques like readout calibration or crosstalk mitigation (modeling)(Section III-B) become critical.

If classical post-processing resources are abundant and quantum resources are scarce, post-processing techniques such as ZNE fitting or expectation value correction (Section III-C) can offer significant improvements with minimal hardware demands.

Thus, the classification offers a context-aware decision map that allows practitioners to align mitigation choices

TABLE 1. Classification of error mitigation techniques for QAOA.

Stage	Technique	Description	Key Benefits	Papers	Year
Preprocessing Techniques	Ansatz Design Optimization	Use problem-aware, depth-reduced ansatz, or adaptive-bias ansatz	Reduces circuit depth and gate count.	[51], [52], [28], [29]	(2021), (2023), (2022), (2023)
	Gate Reordering / Compilation	Compile gates to match hardware topology and minimize two-qubit operations.	Minimizes SWAPs and coherence overhead.	([53], [54]	(2022), (2020)
	Symmetry Exploitation (Factoring/Reduction)	Reduce problem size or circuit complexity using known problem symmetries.	Less depth and fewer gates.	[56]	(2024)
	Qubit Mapping & Routing Optimization	Logical-to-physical qubit assignment that reduces crosstalk and routing.	Hardware-aware optimization.	[54], [57], [58]	(2020), (2023), (2024)
Execution Techniques	ZNE Folding	Run circuits at multiple noise levels (e.g., via gate folding), then extrapolate to zero-noise.	Noise-resilient expectation values.	[59]	(2024)
	Readout Calibration	Adjust measurement results based on hardware-calibrated noise models (including scalable tensored calibration).	Reduces measurement (readout) errors.	[60], [61]	(2021), (2021)
	Cross-Talk Mitigation	Explicit modeling of correlated errors between qubits during readout or execution.	Improves measurement fidelity.	[60]	(2021)
	Symmetry Verification	Reject or post-select runs that violate expected symmetry constraints.	Real-time result filtering.	[62], [63]	(2021), (2022)
Post-processing Techniques	ZNE Fitting	Interpolate/extrapolate results from folded circuit data to predict noise-free outcome.	Error suppression without modifying circuit.	[59]	(2024)
	Result Filtering	Post-select samples matching conserved quantities (e.g., parity, Hamming weight).	Filters out invalid outputs.	[62]	(2021)
	Expectation Value Correction	Adjust computed observables based on known bias/noise patterns (NN, CDR, vnCDR) or variance extrapolation.	Improves fidelity of variational optimization.	[58], [64]–[66]	(2024), (2021), (2021), (2022)
	Digitization of Parameters	Reduce precision of optimized parameters to mitigate analog control errors.	Increases robustness to control noise.	[67]	(2025)

with the noise profiles and resource limitations of specific quantum hardware platforms.

2) COMPOSABILITY AND SYNERGY OF TECHNIQUES

Another important implication is the composability of error mitigation techniques across different stages. Many QEM strategies are not mutually exclusive; they can be combined synergistically to yield greater improvements. For instance:

A hardware-optimized ansatz from preprocessing can be combined with mid-circuit symmetry verification and subsequently enhanced with ZNE fitting during post-processing.

In [58], large-scale QAOA runs combined mapping optimizations, machine-learning-based expectation correction, and calibration-based readout mitigation to successfully execute circuits beyond naïve noise thresholds.

Understanding where each technique fits in the classification promotes layered QEM strategies, enabling practitioners to incrementally add robustness without overburdening any single mitigation layer.

3) TRADE-OFF AWARENESS: ACCURACY VERSUS OVERHEAD

The classification also clarifies the trade-offs inherent to different classes of techniques:

Preprocessing techniques reduce error at the cost of increased classical compilation complexity or the need for problem-specific structural knowledge.

Circuit execution techniques (e.g., readout calibration, crosstalk modeling) often require substantial real-time calibration and may impose significant runtime overhead.

Post-processing techniques like result filtering or ZNE fitting improve fidelity but may discard samples (for result filtering) or introduce extrapolation errors (for ZNE) if improperly configured.

This trade-off awareness enables researchers and engineers to make informed decisions based on available computational

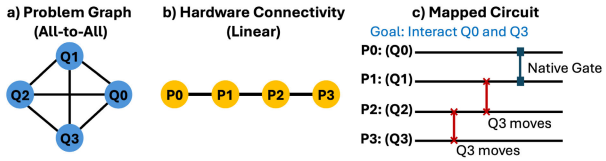

FIGURE 4. Illustration of SWAP gate overhead due to limited hardware connectivity. The diagram shows how a dense problem graph requiring an all-to-all interaction (a) must be compiled with additional SWAP gates (c) to execute on hardware with limited, linear connectivity (b).

resources, target error thresholds, and application-specific requirements. For example, in resource-constrained devices, lightweight post-processing may be preferable to heavy calibration-intensive methods.

4) ENABLING BENCHMARKING AND STANDARDIZATION

Finally, the classification provides a foundation for systematic benchmarking of QAOA error mitigation strategies. By categorizing techniques along the preprocessing–execution–post-processing pipeline, comparative studies can be structured around:

- Where the mitigation acts (circuit stage),
- How it affects resource usage (time, qubits, classical post-processing),
- What impact it has on solution quality (e.g., approximation ratio, optimization convergence rate).
- Such benchmarking would facilitate the standardization of QAOA error mitigation evaluation, a crucial step for advancing toward robust, reproducible, and scalable quantum optimization pipelines.

IV. CLASSIFICATION OF QAOA SETTINGS WITH ERROR MITIGATION STRATEGIES

The performance of the QAOA is highly sensitive to both hardware constraints and noise characteristics inherent to near-term quantum devices. Consequently, the effectiveness of QEM techniques is not uniform across all QAOA deployments but must instead be tailored to the specific setting in which the algorithm is applied.

In this section, we classify QAOA settings into three primary categories—*hardware-constrained QAOA*, *noisy NISQ QAOA*, and *QAOA for all-to-all or dense graphs*—and critically analyze the most appropriate QEM strategies for each category. This classification provides practical guidance for researchers and practitioners seeking to maximize QAOA performance under different noise models and device architectures.

A. HARDWARE-CONSTRAINED QAOA

Hardware-constrained QAOA arises when the quantum device exhibits restricted qubit connectivity, hardware-native gate set limitations, or short coherence times. This setting is typical for many current superconducting qubit architectures (e.g., heavy-hex or linear lattices) and certain trapped-ion layouts where non-local qubit interactions are not naturally supported. The primary challenge in this setting is the significant overhead introduced during the compilation, or transpilation, of the quantum circuit. When the required interactions of the problem graph do not match the physical qubit connectivity of the hardware, the compiler must insert additional SWAP operations to move logical qubit states around the processor. This process is visually demonstrated in Figure 4. The figure shows that to map a dense, all-to-all problem graph (a) onto a device with simple linear connectivity (b), multiple SWAP gates must be added to the circuit just to perform a single logical interaction between non-adjacent qubits (c). So there will be large overhead of SWAP gates required to execute interactions between non-adjacent qubits, which dramatically increases circuit depth and exposure to noise. Preprocessing techniques that optimize the circuit *before* execution are paramount.

Example 1 (Hardware-Constrained Setting): Consider a scenario where the task is to solve a MaxCut problem on a 16-qubit graph with a high average degree, using a quantum processor limited to a linear qubit connectivity chain. In this hardware-constrained setting, the primary challenge is the excessive circuit depth and error accumulation caused by the large number of SWAP gates needed to implement the cost Hamiltonian.

For this situation, a recommended QEM pipeline would be a two-stage process. First, during the Preprocessing stage, a sophisticated Qubit Mapping and Routing Optimization algorithm [54], [57] should be employed as the most critical step to minimize the SWAP overhead. Once the circuit structure has been optimized, the Execution stage should incorporate Readout Correction [60]. This is achieved by running calibration circuits to build a noise model that corrects for final-step measurement noise, serving as a crucial, low-overhead method to improve the final data quality.

Justification for Pipeline in Example 1: In the scenario described in Example 1, the recommended pipeline prioritizes tackling the most dominant source of error first. For a problem with a high-degree graph on hardware with limited connectivity, the primary challenge is structural, as the required SWAP operations significantly increase the overall circuit depth. Therefore, Qubit Mapping and Routing Optimization is selected as the initial step because it directly minimizes this SWAP overhead at the compilation stage. Research has demonstrated that hardware-aware compilation can achieve significant gate count reductions, with some methods reporting up to a 65% decrease for large graphs [57]. After the circuit depth has been optimized, Readout Correction is chosen as a complementary second step. This technique addresses a different, independent source of error—bit-flip inaccuracies during measurement—and does so with manageable overhead, making it an efficient way to enhance the fidelity of the already-optimized circuit.

B. NOISY NISQ QAOA

This setting describes deployments where stochastic (random) errors are the dominant performance limiters. The

primary challenge is that the quantum state is continuously corrupted by random noise, leading to a flattened and distorted energy landscape. A combination of execution-time and post-processing techniques is most effective.

Example 2 (Noisy NISQ Setting): Consider where a 50-qubit QUBO instance for portfolio optimization running on a general-purpose quantum computer. This device may have sufficient qubit connectivity but only moderate gate fidelities (e.g., 99.5%). The primary challenge here is not structural, but rather the continuous corruption of the quantum state by random, stochastic errors, which leads to a flattened and distorted energy landscape.

In this scenario, a powerful QEM pipeline would combine execution and post-processing techniques. The ideal first-line defense is ZNE [59], implemented by executing the circuit at scaled noise levels and extrapolating the results back to the zero-noise limit. As a second step, a Machine Learning-based Expectation Value Correction model [58] can be applied in post-processing. This advanced technique provides a final layer of data-driven refinement on the ZNE-mitigated results, correcting for complex noise correlations that simple extrapolation might miss.

Justification for Pipeline in Example 2: For the 50-qubit portfolio optimization problem in Example 2, the challenge is not excessive SWAPs but the aggregate impact of random noise from a high volume of gates. ZNE is the ideal first-line defense because it is specifically designed to mitigate errors that scale with gate count by extrapolating to the zero-noise limit. Venere et al. has shown ZNE can successfully recover a significant portion of the ideal performance in noisy QAOA workflows [59]. While powerful, ZNE's extrapolation may not capture all complex noise correlations. Therefore, a Machine Learning-based Expectation Value Correction model is added as a second step. This advanced technique provides a final layer of data-driven refinement, as demonstrated by Sack and Egger, who used a neural network to infer and correct noise distortions, leading to improved convergence in large-scale QAOA runs [58].

C. QAOA FOR ALL-TO-ALL OR DENSE GRAPHS

This setting involves problems requiring $\mathcal{O}(n^2)$ entangling gates. The primary challenge is the sheer number of gates, leading to extremely deep circuits. Techniques that exploit the problem's inherent structure are essential.

Example 3 (Dense Graph Setting): The third setting involves problems with dense or all-to-all connectivity, such as finding the MaxCut on a 20-vertex complete graph. If run on a device with full connectivity, like a trapped-ion quantum computer, the challenge is not qubit routing but the immense circuit depth from the sheer number of entangling gates required, as well as the significant degree of problem symmetry.

For this type of problem, the most effective QEM pipeline leverages the inherent structure of the graph. During Preprocessing, Symmetry Exploitation [56] should be used to simplify the problem by factoring out the permutation symmetries of the complete graph. This can fundamentally reduce the complexity and depth of the required circuit. As a complementary second step, Result Filtering [62] can be applied in Post-processing. This technique leverages the same symmetries to validate the final measurement outcomes, discarding any results that noise has pushed out of the valid solution subspace. It is worth noting that related symmetry verification methods [63] can also be implemented during the execution stage using mid-circuit measurements.

Justification for Pipeline in Example 3: The pipeline recommended for Example 3 is tailored for a dense, complete graph where the primary challenges are immense circuit depth and a high degree of symmetry. On a fully-connected device, Qubit Mapping is unnecessary. Symmetry Exploitation is the most potent strategy because it can achieve ''profound circuit simplifications'' by fundamentally reducing the problem's complexity. For example, factoring out ''semi-symmetries'' has been shown to reduce two-qubit couplings by up to 49% and circuit depth by up to 41% [56]. The choice of Result Filtering via Symmetry Constraints as a second step is a natural complement. This post-processing technique leverages the same symmetries used in preprocessing to validate the final measurement outcomes, discarding any results that noise has pushed out of the valid solution subspace [62].

This classification illustrates that no single error mitigation strategy universally optimizes QAOA performance. Instead, effective deployment requires careful alignment of QEM techniques with hardware capabilities, noise characteristics, and problem-specific demands. By tailoring mitigation strategies to the dominant challenges of each QAOA setting, researchers and practitioners can substantially enhance scalability, accuracy, and robustness of QAOA workflows on near-term quantum devices.

V. COMPARATIVE ANALYSIS AND TRADE-OFFS

The selection and deployment of an effective QEM strategy for the QAOA is not a monolithic decision but rather a nuanced exercise in balancing performance gains against resource overhead. As systematically detailed in Table 2, each QEM technique presents a unique profile of benefits and costs, which can be holistically analyzed across the preprocessing, circuit execution, and post-processing stages of a QAOA workflow. This section provides a comparative analysis of these methods, elucidating the critical trade-offs that practitioners must navigate when designing noise-resilient quantum optimization pipelines.

The primary trade-off observed in preprocessing techniques is the investment of substantial classical (and sometimes quantum) computational resources to generate quantum circuits that are inherently more robust and hardware-efficient. For instance, Qubit Mapping & Routing Optimization leverages powerful classical SAT solvers to achieve remarkable gate count reductions—such as a 65% decrease for 500-node graphs —at the cost of significant classical preprocessing [57]. Similarly, Symmetry Exploitation requires

$O(n^3)$ classical computation to identify semi-symmetries but also introduces quantum overhead in the form of additional ancilla qubits to deliver its impressive reductions in gate couplings (up to 49%) and depth (up to 41%) [56]. Ansatz Design Optimization pushes this trade-off further, with methods like the Parity Method demanding a quadratic increase in qubit overhead to enable a more noise-resilient encoding [52]. These strategies are therefore most suitable for scenarios where the primary limitation is the quantum circuit's depth and susceptibility to decoherence, justifying a heavy upfront investment to simplify the subsequent quantum execution. Similarly, Adaptive-Bias ansatz construction reduces the circuit depth required for convergence, effectively trading quantum gate depth for increased complexity in the classical optimization loop to update the bias parameters [28].

In contrast, circuit execution techniques shift the resource burden to the quantum runtime, requiring additional calibration circuits, ancillary logic, or repeated executions to actively manage errors as they occur. A clear illustration of this trade-off is seen when comparing Readout Calibration with advanced Cross-Talk Modeling. Standard Readout Calibration offers a moderate performance improvement but relies on an exponentially costly characterization requiring $O(2^N)$ circuits [60]. However, recent scalable approaches utilizing tensored calibration matrices can mitigate this overhead to linear scaling by assuming noise locality [61]. In contrast, explicitly modeling crosstalk via DDOT achieves superior error reduction—improving outcomes by a factor of over 22 on IBM hardware—with a much more scalable characterization overhead of only $O(k \cdot 2^k \log N)$ circuits [60]. It is important to note that this magnitude of improvement is setting-dependent, specifically observed on processors with significant readout crosstalk. This highlights a key trend: as QEM techniques become more sophisticated, they can offer greater performance with more efficient overhead. Other methods in this category, such as Symmetry Verification [63], introduce ancilla-based or compression circuits directly into the QAOA workflow, adding to the gate depth and overhead in exchange for the ability to filter corrupted states during the evolution.

Finally, post-processing techniques provide a flexible alternative by moving the mitigation workload entirely into the classical domain after measurement data has been collected. ZNE Fitting, for example, requires no modification to the quantum circuit itself but instead relies on classical modeling to extrapolate noisy data points to the zero-noise limit, successfully recovering a significant portion of the ideal performance (e.g., improving the approximation ratio from 0.58 to over 0.75) [59]. Result Filtering similarly operates on classical bitstrings, achieving a notable 23% average improvement in fidelity by discarding invalid outcomes, albeit at the cost of increased measurement shots to ensure statistical convergence [62]. However, this improvement was demonstrated on small-scale benchmarks (up to 5 qubits) on the 7-qubit ibmq_jakarta processor; for larger systems, the exponential reduction in valid sample yield may limit practical scalability. Similarly, Expectation Value Correction strategies present a trade-off between computational overhead and specific error targeting. Data-driven methods such as neural networks or CDR rely on generating classical training data, effectively shifting the cost to offline simulation [58], [64]. In contrast, Variance-Based Extrapolation does not require training data but instead incurs a real-time measurement overhead to estimate the energy variance Δ_{var}, offering the distinct advantage of mitigating algorithmic errors alongside hardware noise [66]. A unique case is the Digitization of Parameters, which mitigates analog control noise by trading circuit depth for robustness; this method requires a depth increase that scales as $\log_2\left(\frac{\sqrt{\epsilon}}{p}\right)$ to achieve the desired precision [67]. These post-processing strategies are particularly valuable when quantum resources are the primary bottleneck and classical computational power is readily available.

In summary, the comparative analysis reveals a clear spectrum of strategic choices. Preprocessing techniques are indispensable in hardware-constrained settings where minimizing circuit depth and SWAP overhead is critical. During execution stage methods offer the highest-fidelity corrections for specific, dominant noise sources like readout errors, provided the calibration overhead is manageable. Post-processing offers a versatile and low-quantum-overhead solution that is ideal for refining results classically. The most effective noise-resilient QAOA pipelines will ultimately employ a synergistic combination of these strategies, creating a multi-layered defense against noise that is tailored to the specific problem instance and hardware platform.

VI. OPEN CHALLENGES AND FUTURE DIRECTIONS

While significant progress has been made in developing and applying QEM techniques to QAOA, several open challenges remain. These challenges limit the scalability, generalizability, and long-term robustness of current mitigation approaches, especially as quantum devices evolve in size and architecture. Addressing these challenges is critical for transitioning QAOA from controlled experimental studies to real-world quantum applications.

A. LIMITATIONS OF CURRENT QEM METHODS

Many existing QEM techniques have been designed for small- to medium-scale systems and are not directly scalable to high-qubit-count architectures. For example, methods such as symmetry verification or readout calibration become increasingly expensive as the number of qubits grows, due to exponential increases in calibration experiments or the complexity of symmetry checks. Additionally, techniques like ZNE and circuit averaging rely on repeated circuit evaluations, which may become infeasible in low-latency or high-throughput settings.

Moreover, the assumptions underlying some mitigation techniques—such as static noise models or idealized symmetry constraints—may not hold on fluctuating or dynamically

TABLE 2. Comparative analysis of QEM techniques for QAOA.

Technique	QAOA Setting	Problem Benchmark	Performance	Resource Overhead	Noise Mitigated
Ansatz Design Optimization	Hardware-Constrained QAOA	Random QUBO instances (up to 7 logical qubits) [52], MaxCut with Adaptive Bias [28]	Achieves higher success probability than standard rerouting; Adaptive-bias reduces required depth p.	The Spanning Tree Method requires classical pre-processing with $O(\Delta \cdot n^2)$ complexity, and the Parity Method uses a quantum circuit with quadratic qubit overhead and requires classical post-processing for decoding; Adaptive-bias requires iterative parameter updates.	Gate errors, decoherence/relaxation and algorithmic errors
Gate Reordering & Compilation	Hardware-Constrained & Dense Graph	Max-Cut on Erdos-Renyi and regular graphs (up to 36 nodes) [54]	$\sim$53% depth and $\sim$23% gate-count reduction on average; and $\sim$25.8% improvement in Approximation Ratio Gap on hardware.	Requires advanced classical compilers/schedulers for optimization; some methods require multiple compilation steps or quantum executions	Gate errors and hardware variability (via variation-aware compilation)
Symmetry Exploitation	Hardware-Constrained QAOA	Max Clique, Hamilton Cycles, Graph Coloring, Vertex Cover, and Graph Isomorphism [56]	Reduces the number of two-qubit gates (couplings) by up to 49% and circuit depth by up to 41%	Requires $O(n^3)$ classical preprocessing to find semi-symmetries, while the quantum part requires additional ancilla qubits to factor out those symmetries.	Gate errors and decoherence
Qubit Mapping & Routing Optimization	Hardware-Constrained QAOA	Random three-regular (RR3) graphs and other sparse random graphs with up to 1000 nodes [57]	Achieved a 65% reduction in CNOT gate count for 500-node RR3 graphs. The heuristic method reduced SWAP layers by 25% for 1000-node graphs compared to trivial mappings.	Requires significant classical computation to solve a Boolean Satisfiability (SAT) problem within a binary search, which scales as $O(\log n)$ SAT instances, and for very large problems, a clustering heuristic is used to make the problem tractable.	Gate errors and decoherence
ZNE Folding	Noisy NISQ QAOA	Max-Cut on 10-node random graphs [59]	(Performance is realized after the fitting stage)	Requires executing multiple, deeper versions of the circuit to sample expectation values at different noise levels (e.g., via gate folding)	Cumulative effect of noise that scales with circuit depth (e.g., gate errors, decoherence)
Readout Calibration	Hardware-Constrained & Noisy NISQ	Energy estimation for random MAX-2-SAT [60]; Scalable Calibration [61]	Achieves a moderate reduction in energy estimation error by correcting for independent bit-flips; Scalable methods allow calibration on 50+ qubits.	Standard characterization is exponentially costly, requiring $O(2^N)$ circuits to build a full calibration matrix for N qubits; Tensored methods reduce this to linear/polynomial.	Uncorrelated and local measurement errors
Cross-Talk Mitigation	Hardware-Constrained & Noisy NISQ	Energy estimation for random MAX-2-SAT and fully-connected Hamiltonians (up to 23 qubits) [60]	Achieves a superior error reduction (by a factor of >22 on IBM hardware) by explicitly modeling and correcting for correlations	Uses efficient characterization via DDOT, requiring only $O(k \cdot 2^k \log N)$ circuits for k-local noise and avoiding exponential scaling.	Correlated readout noise (crosstalk)
Symmetry Verification	Hardware-Constrained & Noisy NISQ	Traveling Salesperson Problem (TSP) on 3 and 4 cities (QAOA+) [63]	Enables the filtering of states that have been corrupted by noise during the circuit's evolution	Requires additional verification circuits (e.g., ancilla-based filtering or ancilla-free compression), which adds gate and depth overhead	Bit-flip (X/Y) errors that violate problem symmetries, but not phase-flip (Z) errors that preserve the valid subspace
ZNE Fitting	Noisy NISQ QAOA	Max-Cut on 10-node random graphs [59]	Improved the average approximation ratio from 0.58 (noisy) to over 0.75, recovering most of the performance lost to noise compared to the ideal case (0.83)	Requires post-processing to fit a model (e.g., Richardson, Linear) to the noisy data points to extrapolate the zero-noise result	The cumulative effect of noise by calculating the zero-noise limit from the noise-scaled data.
Result Filtering	Hardware-Constrained & QAOA for All-to-All or Dense Graphs	MaxCut on small graphs (3-node path/complete, 4-node star/kite) on up to 5 qubits [62]	Achieved a 23% average improvement in quantum state fidelity by verifying bit-flip symmetry	Requires an ancilla qubit and an additional verification circuit (adds gate overhead) and needs post-selection, which increasing the total number of shots needed	Any errors (e.g., gate errors, decoherence) that push the quantum state out of the problem's known symmetry subspace
Expectation Value Correction	Hardware-Constrained & Dense Graphs	QAOA on Random Regular Graphs (40 nodes) [58]; Ising Model [64], [65]; Molecular Hamiltonians [66]	High accuracy via learned noise models (NN/CDR/vnCDR) and algorithmic error reduction (Variance Extrapolation)	Requires classical training data (NN/CDR) or measurement of energy variance Δ_{var} (Variance Extrapolation).	Gate/Readout errors, coherent bias, and algorithmic errors
Digitization of Parameters	Hardware-Constrained	Grover's Search via QAOA (up to 14 qubits), 1D Ising Model / GHZ State Preparation (up to 18 qubits) [67]	Allows mitigation of precision errors by trading circuit depth for robustness.	Required depth increase scales as $\log_2\left(\frac{\sqrt{\varepsilon}}{p}\right)$	Analog precision errors

calibrated hardware. These limitations highlight the need for more adaptive, efficient, and scalable QEM strategies that can generalize beyond small-scale benchmarks.

B. ADAPTIVE AND DYNAMIC ERROR MITIGATION

A key direction for future research is the development of adaptive QEM techniques that can respond dynamically to real-time hardware noise conditions. This includes strategies that monitor device performance during execution and adjust parameters, calibration routines, or error models on-the-fly. A promising approach in this area is pulse-based adaptive mitigation, which uses a specific pulse-level inverse of the circuit to probe the noise strength and adjust mitigation coefficients accordingly, without requiring complex tomography or a machine-learning stage [68]. Such methods are designed to handle time-dependent noise and have been shown to be robust to calibration drifts over long experimental runs, allowing circuits to maintain fidelity despite hardware fluctuations [68]. Other promising examples include dynamic symmetry detection, real-time fidelity-aware qubit remapping, or online ZNE fitting using streaming measurements. These techniques would benefit from tighter integration between QEM modules and quantum runtime systems or cloud-based quantum platforms.

C. INTEGRATION WITH QUANTUM COMPILERS AND SCHEDULERS

Many QEM strategies—especially in the preprocessing category—can be enhanced through deeper integration with quantum compilers and hardware-aware schedulers. For example, compiler-level gate synthesis and routing can incorporate noise-aware metrics, such as error rates or crosstalk profiles, to automatically generate more robust QAOA circuits.

Currently, such integration remains limited or ad hoc. Future QAOA pipelines could benefit from standardized QEM-aware compilation frameworks that seamlessly combine qubit mapping, gate ordering, and symmetry exploitation with circuit-specific error profiling.

D. STANDARDIZATION AND BENCHMARKING OF QEM TECHNIQUES

A significant challenge facing the QEM community is the lack of standardized benchmarks for evaluating mitigation techniques across QAOA settings. Most studies test techniques on specific problems (e.g., Max-Cut, portfolio optimization) under unique hardware conditions, making direct comparison difficult.

To advance the field, there is a pressing need for:

- Standardized datasets and QAOA benchmark instances. State-of-the-art research demonstrating quantum utility has been required to develop its own rigorous, large-scale benchmarking protocols to validate its claims [69]. The absence of a community-wide standard necessitates such complex, resource-intensive efforts on a case-by-case basis, highlighting the need for more universal evaluation frameworks.
- Unified metrics for accuracy, noise resilience, and cost efficiency.
- Cross-platform experimental protocols for evaluating QEM effectiveness.

Such efforts would enable more reproducible research and guide the development of generalized QEM strategies that are effective across architectures and applications.

E. MACHINE LEARNING AND DATA-DRIVEN FOR QEM

Machine learning (ML) offers a promising pathway to transcend the limitations of static error models by learning noise patterns directly from data. While early ML techniques demonstrated success in mapping noisy measurement distributions to their ideal counterparts via Artificial Neural Networks (ANNs) [70], recent advancements have addressed the critical challenges of scalability.

1) SCALABILITY AND EFFICIENCY

A key limitation of traditional error mitigation is the exponential cost of characterizing noise on larger devices. Recent work by Tu et al. [71] and Cantori [72] has introduced scalable deep learning frameworks that exploit the synergy between noisy quantum hardware and classical high-performance computing. By training on efficiently simulable ''Clifford-heavy'' or smaller-scale circuits, these models can generalize to larger, relevant quantum instances without the overhead of full tomography, making them highly promising for near-term scaling.

2) OVERHEAD ANALYSIS

Regarding computational cost, a major advantage of these strategies is their impact on circuit repetitions. Unlike standard techniques like ZNE or PEC which incur a multiplicative overhead in measurement shots (often scaling exponentially with noise strength), ML surrogates shift the overhead from the quantum processor (QPU) to the classical training phase. As demonstrated by Liao et al. [73], classical learning surrogates allow for *sample-efficient* mitigation: once the model is trained, the inference cost is negligible. This makes them the most computationally efficient strategy for runtime-constrained QAOA workflows where reducing QPU latency is paramount, provided the one-time classical training cost can be amortized.

VII. CONCLUSION

As NISQ-era quantum hardware continues to evolve, the development of effective and efficient error-mitigation strategies is essential to realizing the practical potential of quantum optimization algorithms. This paper addressed that need by presenting a comprehensive classification of QEM techniques for QAOA, organized across the stages of algorithm execution. By classifying techniques into preprocessing, during execution, and post-processing categories,

we have created a structured framework for understanding the landscape of mitigation approaches.

Furthermore, we moved beyond a simple survey by establishing a prescriptive framework that maps tailored, multi-stage QEM pipelines to distinct QAOA deployment archetypes—hardware-constrained, noise-dominated, and dense-graph settings. Each recommendation is supported by an evidence-based rationale, providing practitioners with actionable guidance for their specific use cases. This framework is supported by an in-depth comparative analysis of the trade-offs among mitigation accuracy, resource efficiency, and noise resilience, synthesized from a holistic review of the state-of-the-art literature.

While significant progress has been made, our discussion of open challenges highlights the ongoing need for scalable, adaptive, and compiler-integrated QEM approaches. We envision future QAOA pipelines that leverage dynamic error tracking, machine learning-based mitigation, and standardized benchmarking to improve robustness and reproducibility.

In summary, this work not only organizes existing knowledge but also provides actionable insights to support the design of robust QAOA implementations on today's imperfect quantum devices. By making informed choices about when and how to mitigate errors, researchers and practitioners can move closer to achieving meaningful quantum advantage in real-world optimization tasks.

REFERENCES

[1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," *Nature Rev. Phys.*, vol. 3, no. 9, pp. 625–644, Aug. 2021, doi: 10.1038/s42254-021-00348-9.

[2] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New J. Phys.*, vol. 18, no. 2, Feb. 2016, Art. no. 023023, doi: 10.1088/1367-2630/18/2/023023.

[3] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, "Variational quantum linear solver," *Quantum*, vol. 7, p. 1188, Nov. 2023, doi: 10.22331/q-2023-11-22-1188.

[4] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018, doi: 10.22331/q-2018-08-06-79.

[5] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*.

[6] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, P. Balaprakash, S. Khairy, R. Shaydulin, and L. Cincio, "Learning to optimize variational quantum circuits to solve combinatorial problems," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 3, pp. 2367–2375, doi: 10.1609/aaai.v34i03.5616.

[7] D. F. Perez-Ramirez, "Variational quantum algorithms for combinatorial optimization," 2024, *arXiv:2407.06421*.

[8] Y.-C. Lin, C.-C. Wang, C.-H. Tu, and S.-H. Hung, "Towards optimizations of quantum circuit simulation for solving max-cut problems with QAOA," in *Proc. 39th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2024, pp. 1487–1494, doi: 10.1145/3605098.3635897.

[9] C. Moussa, H. Wang, T. Bäck, and V. Dunjko, "Unsupervised strategies for identifying optimal parameters in quantum approximate optimization algorithm," *EPJ Quantum Technol.*, vol. 9, no. 1, p. 11, Dec. 2022, doi: 10.1140/epjqt/s40507-022-00131-4.

[10] D. Amaro, C. Modica, M. Rosenkranz, M. Fiorentini, M. Benedetti, and M. Lubasch, "Filtering variational quantum algorithms for combinatorial optimization," *Quantum Sci. Technol.*, vol. 7, no. 1, Jan. 2022, Art. no. 015021, doi: 10.1088/2058-9565/ac3e54.

[11] J. Basso, E. Farhi, K. Marwaha, B. Villalonga, and L. Zhou, "The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the sherrington-kirkpatrick model," 2021, *arXiv:2110.14206*.

[12] M. P. Harrigan et al., "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor," *Nature Phys.*, vol. 17, no. 3, pp. 332–336, Mar. 2021, doi: 10.1038/s41567-020-01105-y.

[13] 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*, vol. 549, no. 7671, pp. 242–246, Sep. 2017, doi: 10.1038/nature23879.

[14] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *Rev. Modern Phys.*, vol. 95, no. 4, Dec. 2023, Art. no. 045005, doi: 10.1103/revmodphys.95.045005.

[15] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," *Phys. Rev. Lett.*, vol. 119, no. 18, Nov. 2017, Art. no. 180509, doi: 10.1103/physrevlett.119.180509.

[16] V. R. Pascuzzi, A. He, C. W. Bauer, W. A. de Jong, and B. Nachman, "Computationally efficient zero-noise extrapolation for quantum-gate-error mitigation," *Phys. Rev. A, Gen. Phys.*, vol. 105, no. 4, Apr. 2022, Art. no. 042406, doi: 10.1103/physreva.105.042406.

[17] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O'Brien, "Low-cost error mitigation by symmetry verification," *Phys. Rev. A, Gen. Phys.*, vol. 98, no. 6, Dec. 2018, Art. no. 062339, doi: 10.1103/physreva.98.062339.

[18] A. W. R. Smith, K. E. Khosla, C. N. Self, and M. S. Kim, "Qubit readout error mitigation with bit-flip averaging," *Sci. Adv.*, vol. 7, no. 47, p. 8009, Nov. 2021, doi: 10.1126/sciadv.abi8009.

[19] R. Hicks, B. Kobrin, C. W. Bauer, and B. Nachman, "Active readout-error mitigation," *Phys. Rev. A, Gen. Phys.*, vol. 105, no. 1, Jan. 2022, Art. no. 012419, doi: 10.1103/physreva.105.012419.

[20] Y. Guo and S. Yang, "Efficient quantum circuit compilation for near-term quantum advantage," *EPJ Quantum Technol.*, vol. 12, no. 1, p. 69, Dec. 2025, doi: 10.1140/epjqt/s40507-025-00368-9.

[21] S. Endo, S. C. Benjamin, and Y. Li, "Practical quantum error mitigation for near-future applications," *Phys. Rev. X*, vol. 8, no. 3, Jul. 2018, Art. no. 031027, doi: 10.1103/physrevx.8.031027.

[22] 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*, vol. 10, no. 2, Jun. 2020, Art. no. 021067, doi: 10.1103/physrevx.10.021067.

[23] D. J. Egger, J. Mareček, and S. Woerner, "Warm-starting quantum optimization," *Quantum*, vol. 5, p. 479, Jun. 2021, doi: 10.22331/q-2021-06-17-479.

[24] R. Herrman, P. C. Lotshaw, J. Ostrowski, T. S. Humble, and G. Siopsis, "Multi-angle quantum approximate optimization algorithm," *Sci. Rep.*, vol. 12, no. 1, p. 6781, Apr. 2022, doi: 10.1038/s41598-022-10555-8.

[25] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," *Nature Commun.*, vol. 10, no. 1, p. 3007, Jul. 2019, doi: 10.1038/s41467-019-10988-2.

[26] Z. Wang, S. Hadfield, J. Zhang, and E. Rieffel, "Quantum approximate optimization algorithm for MaxCut: A fermionic view," *Phys. Rev. A, Gen. Phys.*, vol. 97, no. 2, Feb. 2018, Art. no. 022304, doi: 10.1103/physreva.97.022304.

[27] P. Zou, "Multiscale quantum approximate optimization algorithm," *Phys. Rev. A, Gen. Phys.*, vol. 111, no. 1, Jan. 2025, Art. no. 012427, doi: 10.1103/physreva.111.012427.

[28] Y. Yu, C. Cao, C. Dewey, X.-B. Wang, N. Shannon, and R. Joynt, "Quantum approximate optimization algorithm with adaptive bias fields," *Phys. Rev. Res.*, vol. 4, no. 2, Jun. 2022, Art. no. 023249, doi: 10.1103/physrevresearch.4.023249.

[29] Y. Yu, C. Cao, X.-B. Wang, N. Shannon, and R. Joynt, "Solution of SAT problems with the adaptive-bias quantum approximate optimization algorithm," *Phys. Rev. Res.*, vol. 5, no. 2, Jun. 2023, Art. no. 023147, doi: 10.1103/physrevresearch.5.023147.

[30] V. Akshay, H. Philathong, M. E. S. Morales, and J. D. Biamonte, "Reachability deficits in quantum approximate optimization," *Phys. Rev. Lett.*, vol. 124, no. 9, Mar. 2020, Art. no. 090504, doi: 10.1103/physrevlett.124.090504.

[31] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A quantum engineer's guide to superconducting qubits," *Appl. Phys. Rev.*, vol. 6, no. 2, Jun. 2019, Art. no. 021318, doi: 10.1063/1.5089550.

[32] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren Plateaus in quantum neural network training landscapes," *Nature Commun.*, vol. 9, no. 1, p. 4812, Nov. 2018, doi: 10.1038/s41467-018-07090-4.

[33] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*. Cham, Switzerland: Springer, 1994, pp. 51–67.

[34] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965, doi: 10.1093/comjnl/7.4.308.

[35] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. de Jong, "Classical optimizers for noisy intermediate-scale quantum devices," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Oct. 2020, pp. 267–277.

[36] J. C. Spall, "An overview of the simultaneous perturbation method for efficient optimization," *Johns Hopkins Apl. Tech. Dig.*, vol. 19, no. 4, pp. 482–492, 1998.

[37] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, "Quantum natural gradient," *Quantum*, vol. 4, p. 269, May 2020, doi: 10.22331/q-2020-05-25-269.

[38] C. Xue, Z.-Y. Chen, Y.-C. Wu, and G.-P. Guo, "Effects of quantum noise on quantum approximate optimization algorithm," *Chin. Phys. Lett.*, vol. 38, no. 3, Mar. 2021, Art. no. 030302.

[39] S. McArdle, X. Yuan, and S. Benjamin, "Error-mitigated digital quantum simulation," *Phys. Rev. Lett.*, vol. 122, no. 18, May 2019, Art. no. 180501, doi: 10.1103/physrevlett.122.180501.

[40] P. Murali, D. C. Mckay, M. Martonosi, and A. Javadi-Abhari, "Software mitigation of crosstalk on noisy intermediate-scale quantum computers," in *Proc. 25th Int. Conf. Architectural Support Program. Lang. Operating Syst.* New York, NY, USA: Association for Computing Machinery, Mar. 2020, pp. 1001–1016, doi: 10.1145/3373376.3378477.

[41] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, Feb. 2019, doi: 10.3390/a12020034.

[42] 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 quantum processor," 2013, *arXiv:1304.3061*.

[43] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, "Digital zero noise extrapolation for quantum error mitigation," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Oct. 2020, pp. 306–316, doi: 10.1109/QCE49297.2020.00045.

[44] A. He, B. Nachman, W. A. de Jong, and C. W. Bauer, "Zero-noise extrapolation for quantum-gate error mitigation with identity insertions," *Phys. Rev. A, Gen. Phys.*, vol. 102, no. 1, Jul. 2020, Art. no. 012426, doi: 10.1103/physreva.102.012426.

[45] S. Bravyi, S. Sheldon, A. Kandala, D. C. Mckay, and J. M. Gambetta, "Mitigating measurement errors in multiqubit experiments," *Phys. Rev. A, Gen. Phys.*, vol. 103, no. 4, Apr. 2021, Art. no. 042605, doi: 10.1103/physreva.103.042605.

[46] R. S. Gupta, E. van den Berg, M. Takita, D. Ristè, K. Temme, and A. Kandala, "Probabilistic error cancellation for dynamic quantum circuits," *Phys. Rev. A, Gen. Phys.*, vol. 109, no. 6, Jun. 2024, Art. no. 062617, doi: 10.1103/physreva.109.062617.

[47] B. McDonough, A. Mari, N. Shammah, N. T. Stemen, M. Wahl, W. J. Zeng, and P. P. Orth, "Automated quantum error mitigation based on probabilistic error reduction," in *Proc. IEEE/ACM 3rd Int. Workshop Quantum Comput. Softw. (QCS)*, Nov. 2022, pp. 83–93, doi: 10.1109/QCS56647.2022.00015.

[48] R. LaRose, A. Mari, S. Kaiser, P. J. Karalekas, A. A. Alves, P. Czarnik, M. El Mandouh, M. H. Gordon, Y. Hindy, A. Robertson, P. Thakre, M. Wahl, D. Samuel, R. Mistri, M. Tremblay, N. Gardner, N. T. Stemen, N. Shammah, and W. J. Zeng, "Mitiq: A software package for error mitigation on noisy quantum computers," *Quantum*, vol. 6, p. 774, Aug. 2022, doi: 10.22331/q-2022-08-11-774.

[49] Z. Du, S. Kan, S. Stein, Z. Liang, A. Li, and Y. Mao, "Hardware-aware compilation for chip-to-chip coupler-connected modular quantum systems," 2025, *arXiv:2505.09036*.

[50] A. Zulehner, A. Paler, and R. Wille, "An efficient methodology for mapping quantum circuits to the IBM QX architectures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 7, pp. 1226–1236, Jul. 2019.

[51] R. Majumdar, D. Bhoumik, D. Madan, D. Vinayagamurthy, S. Raghunathan, and S. Sur-Kolay, "Depth optimized ansatz circuit in QAOA for max-cut," 2021, *arXiv:2110.04637*.

[52] A. Weidinger, G. B. Mbeng, and W. Lechner, "Error mitigation for quantum approximate optimization," *Phys. Rev. A, Gen. Phys.*, vol. 108, no. 3, Sep. 2023, Art. no. 032408, doi: 10.1103/physreva.108.032408.

[53] A. Hashim, R. Rines, V. Omole, R. K. Naik, J. M. Kreikebaum, D. I. Santiago, F. T. Chong, I. Siddiqi, and P. Gokhale, "Optimized SWAP networks with equivalent circuit averaging for QAOA," *Phys. Rev. Res.*, vol. 4, no. 3, Jul. 2022, Art. no. 033028, doi: 10.1103/physrevresearch.4.033028.

[54] M. Alam, A. Ash-Saki, J. Li, A. Chattopadhyay, and S. Ghosh, "Noise resilient compilation policies for quantum approximate optimization algorithm," in *Proc. 39th Int. Conf. Computer-Aided Design*, Nov. 2020, pp. 1–7, doi: 10.1145/3400302.3415745.

[55] R. J. Marshman, D. Singh, T. C. Ralph, and A. P. Lund, "Unitary averaging with fault and loss tolerance," *Phys. Rev. A, Gen. Phys.*, vol. 109, no. 6, Jun. 2024, Art. no. 062436, doi: 10.1103/physreva.109.062436.

[56] J. Nüßlein, L. Sünkel, J. Stein, T. Rohe, D. Schuman, C. Linnhoff-Popien, and S. Feld, "Reducing QAOA circuit depth by factoring out semi-symmetries," 2024, *arXiv:2411.08824*.

[57] A. Matsuo, S. Yamashita, and D. J. Egger, "A SAT approach to the initial mapping problem in SWAP gate insertion for commuting gates," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E106.A, no. 11, pp. 1424–1431, Nov. 2023, doi: 10.1587/transfun.2022eap1159.

[58] S. H. Sack and D. J. Egger, "Large-scale quantum approximate optimization on nonplanar graphs with machine learning noise mitigation," *Phys. Rev. Res.*, vol. 6, no. 1, Mar. 2024, Art. no. 013223, doi: 10.1103/physrevresearch.6.013223.

[59] M. Venere, A. Lusso, V. Onofre, A. Maldonado-Romo, and M. D. Santambrogio, "Characterizing the effects of zero-noise extrapolation on a QAOA workflow," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Sep. 2024, pp. 426–427, doi: 10.1109/QCE60285.2024.10338.

[60] F. B. Maciejewski, F. Baccari, Z. Zimborás, and M. Oszmaniec, "Modeling and mitigation of cross-talk effects in readout noise with applications to the quantum approximate optimization algorithm," *Quantum*, vol. 5, p. 464, Jun. 2021, doi: 10.22331/q-2021-06-01-464.

[61] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, "Scalable mitigation of measurement errors on quantum computers," *PRX Quantum*, vol. 2, no. 4, Nov. 2021, Art. no. 040326, doi: 10.1103/prxquantum.2.040326.

[62] R. Shaydulin and A. Galda, "Error mitigation for deep quantum optimization circuits by leveraging problem symmetries," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Oct. 2021, pp. 291–300, doi: 10.1109/QCE52317.2021.00046.

[63] L. Botelho, A. Glos, A. Kundu, J. A. Miszczak, Ö. Salehi, and Z. Zimborás, "Error mitigation for variational quantum algorithms through mid-circuit measurements," *Phys. Rev. A, Gen. Phys.*, vol. 105, no. 2, Feb. 2022, Art. no. 022441, doi: 10.1103/physreva.105.022441.

[64] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, "Error mitigation with Clifford quantum-circuit data," *Quantum*, vol. 5, p. 592, Nov. 2021, doi: 10.22331/q-2021-11-26-592.

[65] A. Lowe, M. H. Gordon, P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, "Unified approach to data-driven quantum error mitigation," *Phys. Rev. Res.*, vol. 3, no. 3, Jul. 2021, Art. no. 033098, doi: 10.1103/physrevresearch.3.033098.

[66] C. Cao, Y. Yu, Z. Wu, N. Shannon, B. Zeng, and R. Joynt, "Mitigating algorithmic errors in quantum optimization through energy extrapolation," *Quantum Sci. Technol.*, vol. 8, no. 1, Jan. 2023, Art. no. 015004, doi: 10.1088/2058-9565/ac969c.

[67] G. Quiroz, P. Titum, P. Lotshaw, P. Lougovski, K. Schultz, E. Dumitrescu, and I. Hen, "Quantifying the impact of precision errors on quantum approximate optimization algorithms," *Phys. Rev. Res.*, vol. 7, no. 2, Jun. 2025, Art. no. 023240, doi: 10.1103/physrevresearch.7.023240.

[68] I. Henao, J. P. Santos, and R. Uzdin, "Adaptive quantum error mitigation using pulse-based inverse evolutions," *npj Quantum Inf.*, vol. 9, no. 1, p. 120, Nov. 2023, doi: 10.1038/s41534-023-00785-7.

[69] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, "Evidence for the utility of quantum computing before fault tolerance," *Nature*, vol. 618, no. 7965, pp. 500–505, Jun. 2023, doi: 10.1038/s41586-023-06096-3.

[70] C. Kim, K. D. Park, and J.-K. Rhee, "Quantum error mitigation with artificial neural network," *IEEE Access*, vol. 8, pp. 188853–188860, 2020.

[71] Z. Tu, J. Xu, X. Zhou, Y. Zhu, Y. Liu, Q. Du, H. Lian, and B. Zhou, "QEMOS: A scalable quantum error mitigation method to overcome qubit sensitivity," *Mach. Learn., Sci. Technol.*, vol. 6, no. 4, Dec. 2025, Art. no. 045033.
[72] S. Cantori, A. Mari, D. Vitali, and S. Pilati, "Synergy between noisy quantum computers and scalable classical deep learning for quantum error mitigation," *EPJ Quantum Technol.*, vol. 11, no. 1, pp. 1–13, Dec. 2024, doi: 10.1140/epjqt/s40507-024-00256-8.
[73] W.-Y. Liao, G. Yan, Y. Song, T.-C. Tian, W.-M. Zhu, D.-T. Jiang, Y. Du, and H.-L. Huang, "Sample-efficient quantum error mitigation via classical learning surrogates," 2025, *arXiv:2511.07092*.

KIMCHHOR CHIV received the bachelor's degree in computer science from the Royal University of Phnom Penh, Cambodia, in 2022. She is currently pursuing the M.S. degree with the Department of AI Convergence, Pukyong National University, Busan, South Korea. Her research interests include quantum machine learning and quantum error mitigation.

LEANGHOK HOUR received the bachelor's degree in management information systems from SETEC Institute, Phnom Penh, Cambodia, in 2019. He is currently pursuing the integrated M.S. and Ph.D. degree with the Department of AI Convergence, Pukyong National University, Busan, South Korea. His research interests include compilers, deep learning, and memory systems.

SANGHYEON LEE received the integrated B.S. and M.S. degree from the Department of AI Convergence, Pukyong National University, Busan, South Korea, in 2024, where he is currently pursuing the Ph.D. degree. His research interests include compiler and quantum computing.

TARA KIT received the bachelor's degree in computer science and engineering from the Royal University of Phnom Penh, Cambodia, in 2020. He is currently pursuing the M.S. degree with the Department of AI Convergence, Pukyong National University, Busan, South Korea. His current research interests include quantum machine learning and quantum blockchain.

TAE-KYUNG KIM received the M.E. and Ph.D. degrees in information industry engineering from Chungbuk National University, Republic of Korea, in 2005 and 2010, respectively. From 2013 to 2019, he directed Korea Software HRD Center, Phnom Penh, Cambodia, where he led large-scale talent-development programs. Previously, he was an AI Faculty with Incheon Jaeneung University. He is currently a Professor with the Department of Management Information Systems, Chungbuk National University. His research interests include artificial intelligence, big data analytics, and quantum machine learning, with an applied interest in software education.

YOUNGSUN HAN (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2003 and 2009, respectively. He was a Senior Engineer with the System LSI, Samsung Electronics, Suwon, South Korea, from 2009 to 2011. He was an Assistant/Associate Professor with the Department of Electronic Engineering, Kyungil University, Gyeongsan-si, South Korea, from 2011 to 2019. He is currently a Professor with the Department of Computer and AI Engineering, Pukyong National University, Busan, South Korea. His research interests include quantum computing, compiler construction, microarchitecture, and high-performance computing.