TRACE: TRAJECTORY-ADAPTIVE ERROR MITIGATION FOR VARIATIONAL QUANTUM EIGENSOLVERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Variational quantum eigensolvers (VQEs) use repeatedly measured energy estimates as in-loop optimization signals, making the parameter trajectory itself vulnerable to hardware noise and finite-shot fluctuations. Existing mitigation methods are limited in this regime: physics-based schemes require costly per-query executions, while offline learning-based mitigators are usually frozen pointwise predictors and can fail under the trajectory-dependent distribution shift induced by VQE. We propose **TrACE**, a trajectory-aware pretrain-and-adapt framework for in-loop VQE error mitigation. For each target Hamiltonian, TrACE pretrains a reusable mitigator within a problem-scoped ansatz pool, learning a structured mitigation prior from circuit, parameter, and noisy-measurement features. During deployment, it serves as the optimizer-facing objective and absorbs trajectory context. It is intermittently adapted without ideal-energy labels under self-supervision, updating only the fusion and output modules through repeated noisy views, trajectory stability, and weak prior anchoring. We also introduce a trajectory-level evaluation protocol that measures the quality of the parameters and convergence of VQE, rather than rewarding overly optimistic pointwise energy estimates. Across TFIM, molecular-energy, and MaxCut benchmarks, including a 100-qubit TFIM instance, TrACE improves solution quality, convergence stability, and optimization speed over representative baselines under various noise settings.

1 INTRODUCTION

Quantum computing offers a promising computational paradigm for problems whose intrinsically quantum structure makes classical simulation difficult [50; 8]. In the Noisy Intermediate-Scale Quantum (NISQ) regime [54], the Variational Quantum Eigensolver (VQE) has become a representative framework for ground-state energy estimation, with applications in quantum chemistry [16; 23], condensed matter physics [20], materials science [47; 24], and drug discovery [34; 15; 9; 67]. Given a target Hamiltonian, VQE repeatedly executes a parameterized quantum circuit, estimates the Hamiltonian expectation value, and passes the resulting energy objective to a classical optimizer. Its practical effectiveness therefore depends not only on ansatz expressivity, but also on the reliability of the optimization signal supplied at each iteration. On noisy hardware, gate errors, decoherence, readout errors, and finite-shot fluctuations perturb the expectation values used to construct this signal, thereby distorting the optimization landscape, altering ansatz performance, biasing the estimated optimum, inducing spurious local minima, and ultimately degrading the quality of the optimized variational parameters [28; 27; 41; 66].

Since large-scale fault-tolerant quantum computation is not yet available [54; 7], quantum error mitigation (QEM) provides a practical route to improving expectation-value estimates from noisy quantum-circuit executions [54; 7; 12]. However, many widely used QEM methods follow a static, pointwise paradigm: they mitigate each circuit expectation value independently at a given parameter value. Such paradigm is insufficient for in-loop VQE mitigation, where the mitigated energy is not merely an output estimate, but the objective that drives the next optimizer query and shapes the sequence of visited parameters. This mismatch appears in different forms across existing QEM families. Physics-motivated methods such as zero-noise extrapolation (ZNE) [62; 33; 29] and probabilistic error cancellation (PEC) [62; 26; 64] can be effective, but deploying them at every VQE step is costly: ZNE requires repeated noise-scaled executions, while PEC incurs quasi-probability

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

sampling overhead that can grow rapidly with circuit size, depth, and noise strength. Machine-learning-based QEM methods [59; 44; 45; 3; 4] reduce deployment-time mitigation cost by training models offline and reusing them for target-circuit evaluations. Yet these models are typically frozen at deployment, making them vulnerable to local miscalibration under the optimizer-induced distribution shift: during VQE, queries are sequentially concentrated on trajectory-dependent regions of the parameter space rather than sampled independently from the offline training distribution. Recent VQE-specific Gaussian process regression (GPR) methods [36] introduce trajectory-aware regression for VQE, but their kernel design and posterior state are often tied to a particular ansatz or run, limiting reuse across ansatz pools and introducing per-run update and storage costs. These limitations reveal a missing capability: an in-loop VQE mitigator that reuses a learned problem-scoped prior across related ansatz instances while cheaply adapting to the trajectory encountered during deployment.

To address this gap, we propose **TrACE** (**T**rajectory-**A**daptive **C**ontextual **E**rror **M**itigation), a pretrain-and-adapt framework that supplies a mitigated objective throughout VQE optimization. Unlike physics-motivated QEM, TrACE amortizes mitigation through offline learning and avoids noise scaling or quasi-probability sampling at every optimization step. Unlike frozen learning-based QEM, TrACE combines VQE-native ansatz structure, parameter, measurement, and trajectory information with lightweight self-supervised recalibration during deployment. Unlike VQE-specific GPR methods that are often ansatz- or run-specific, TrACE learns a reusable problem-scoped prior across an ansatz pool and adapts it online without ideal-energy labels. In summary, our contributions are fourfold.

1) We formulate in-loop VQE mitigation as a problem-scoped amortized learning problem. Unlike pointwise QEM, which treats each circuit evaluation as an isolated expectation-value denoising task, our formulation optimizes for the quality of the VQE objective signal itself, so that mitigation is judged by whether it guides the optimizer toward parameters with low ideal energy.

2) We design a unified pretrained mitigation backbone for a problem-scoped ansatz pool. The backbone fuses circuit topology, block-aligned variational parameters, noisy Pauli-level Hamiltonian measurements, and expectation estimates to learn a reusable structural and noise-aware prior across related VQE runs, rather than training a separate mitigator for each target ansatz.

3) We develop a lightweight self-supervised adaptation mechanism for online trajectory recalibration. Starting from the pretrained mitigator, TrACE incorporates trajectory context during deployment and updates only the fusion module and output head during the VQE run, recalibrating the in-loop objective without ideal-energy labels or a separately trained mitigator for each optimization run.

4) We introduce a trajectory-level benchmarking protocol for in-loop VQE mitigation. Since the goal is to discover parameters with low ideal energy, pointwise expectation-value error alone is insufficient for evaluating an in-loop mitigator. Our protocol evaluates the quality of the visited parameters, best-so-far performance, convergence speed, and trajectory stability. Across TFIM, molecular-energy, and MaxCut benchmarks, including cross-ansatz transfer and larger instances up to 100 qubits, TrACE consistently improves in-loop VQE optimization over representative baselines.

2 Preliminaries

2.1 Parameterized Quantum Circuits and Observables

Denote a parameterized quantum circuit (PQC) by $U_s(\boldsymbol{\theta})$, where s indexes a fixed ansatz instance and $\boldsymbol{\theta} \in \mathbb{R}^{d_s}$ its variational parameters [5; 18]. The ansatz instance s specifies the circuit architecture, including the gate sequence, qubit layout, entangling pattern, and positions of trainable gates; only $\boldsymbol{\theta}$ changes during optimization [49; 37]. Applied to the reference input state, the circuit prepares the variational state

$$|\psi_s(\boldsymbol{\theta})\rangle = U_s(\boldsymbol{\theta})|0\rangle^{\otimes n}. \tag{1}$$

For a Hermitian observable O, the corresponding ideal expectation value is

$$y_s^{\mathrm{ideal}}(\boldsymbol{\theta}) = \langle 0|U_s^\dagger(\boldsymbol{\theta}) O U_s(\boldsymbol{\theta})|0\rangle. \tag{2}$$

This observable-expectation formulation is the standard objective-estimation primitive in variational quantum algorithms such as VQE [53; 49]. On real quantum backends, gate noise, measurement noise, and finite-shot sampling produce a noisy estimate $y_s^{\mathrm{noisy}}(\boldsymbol{\theta})$ rather than direct access to $y_s^{\mathrm{ideal}}(\boldsymbol{\theta})$ [54; 38; 65]. Thus, the observable value available to a PQC-based algorithm depends jointly on the ansatz structure s, the variational parameters $\boldsymbol{\theta}$, the measured observable O, and the backend noise condition.

2.2 Variational Quantum Eigensolver

The variational quantum eigensolver (VQE) is a hybrid quantum–classical algorithm for approximating the ground-state energy of a problem Hamiltonian H [18; 63]. Let E^{ref} denote the reference ground-state energy. For a fixed ansatz instance s, VQE optimizes the parameters of $U_s(\boldsymbol{\theta})$ by minimizing the ideal variational energy

$$E^{\mathrm{ideal}}(\boldsymbol{\theta}; s) = \langle 0|U_s^{\dagger}(\boldsymbol{\theta}) H U_s(\boldsymbol{\theta})|0\rangle. \tag{3}$$

The ansatz-optimal parameter is

$$\boldsymbol{\theta}_s^{\star} \in \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^{d_s}} E^{\mathrm{ideal}}(\boldsymbol{\theta}; s). \tag{4}$$

By the variational principle, $E^{\mathrm{ref}} \le E^{\mathrm{ideal}}(\boldsymbol{\theta}_s^{\star}; s) \le E^{\mathrm{ideal}}(\boldsymbol{\theta}; s)$, for any $\boldsymbol{\theta}$, as $E^{\mathrm{ideal}}(\boldsymbol{\theta}_s^{\star}; s)$ is also defined by the quality of the ansatz. During one VQE run, H and s are fixed, while the classical optimizer queries a trajectory $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$. At each step t, the energy constructed by the circuit is defined as $E_t^{ideal} := E^{\mathrm{ideal}}(\boldsymbol{\theta}_t; s)$. We evaluate the ideal quality of each visited parameter by its reference gap $\delta_t^{\mathrm{ideal}} = E^{\mathrm{ideal}}(\boldsymbol{\theta}_t; s) - E^{\mathrm{ref}}$. This gap shows the quality the parameter selected by the optimizer. On quantum hardware, errors such as gate noise, measurement noise, and finite-shot sampling prevent direct access to $E^{\mathrm{ideal}}(\boldsymbol{\theta}_t; s)$. The optimizer instead receives a noisy estimate E_t^{noisy}, whose distortion can guide the trajectory toward parameters with low reported energy but poor ideal quality. Therefore, in-loop VQE mitigation should provide an objective signal that helps the optimizer find parameters with smaller ideal gaps, rather than merely producing a lower reported energy.

2.3 Learning-based Quantum Error Mitigation

Learning-based quantum error mitigation (QEM) learns a parametric mitigation map from noisy circuit-execution information to a mitigated observable value. Given an offline supervised dataset $\mathcal{D} = \{(\mathbf{x}_k, y_k^{\mathrm{ideal}})\}_{k=1}^{N}$, where $\mathbf{x}_k$ contains the available circuit, measurement, and hardware-related information, including the noisy observable y_k^{noisy}, the ideal value y_k^{ideal} serves as the supervision label. The supervised training objective is

$$\mathcal{L}_{\mathrm{sup}}(\phi) = \frac{1}{N}\sum_{k=1}^{N}\left(h_{\phi}(\mathbf{x}_k) - y_k^{\mathrm{ideal}}\right)^2, \qquad \phi^{\star} \in \arg\min_{\phi} \mathcal{L}_{\mathrm{sup}}(\phi). \tag{5}$$

This formulation covers a broad class of neural and graph-based QEM methods that learn mitigation rules from paired noisy and ideal circuit data [6; 44; 3; 4].

Standard learning-based QEM is usually evaluated as an offline pointwise mitigation problem: each circuit-observable pair is mitigated independently, and the objective is to reduce the discrepancy between y_k^{miti} and the supervised label y_k^{ideal}. In VQE, however, the mitigated value is used inside a closed optimization loop. At step t, the mitigator provides the scalar objective observed by the classical optimizer, and this value influences the next queried parameter $\boldsymbol{\theta}_{t+1}$. Therefore, mitigation errors may alter not only a single energy estimate, but also the subsequent optimization trajectory. Moreover, during an actual noisy VQE run, the ideal energy label E_t^{ideal} at the queried parameter $\boldsymbol{\theta}_t$ is not available for supervised retraining. Hence, the standard offline supervised QEM paradigm is insufficient for in-loop VQE optimization, motivating a mitigation scheme that is trajectory-aware and can adapt during deployment without ideal-energy labels.

3 Method

3.1 Problem Setup: In-Loop VQE Error Mitigation

Following the ML-based QEM formulation in [3], we introduce a problem-scoped energy mitigator $h_{\phi_t} : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is a structured input space and ϕ_t denotes the parameters used at iteration t. For the structured input constructed from the current circuit execution $\mathbf{x}_t \in \mathcal{X}$, the mitigated objective passed to the classical optimizer is $h_{\phi_t}(\mathbf{x}_t)$, rather than E_t^{noisy}. Conceptually, if ideal energy labels along a VQE trajectory were available, as in standard supervised ML-QEM, the desired

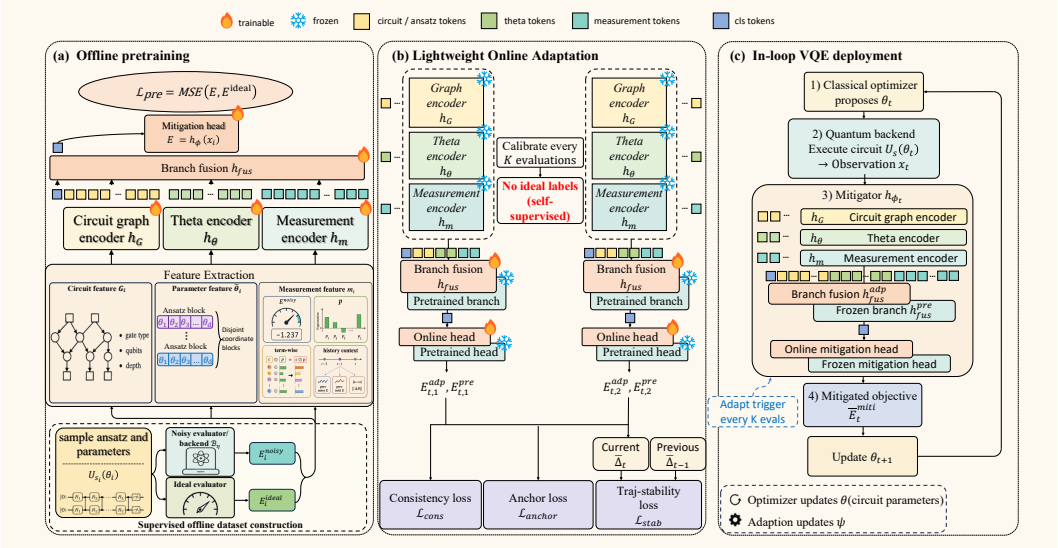


Figure 1: Overview of TrACE for trajectory-adaptive in-loop VQE mitigation. (a) **Offline pretraining:** for each target problem, TrACE builds a supervised noisy–ideal dataset across an ansatz pool and learns a structured mitigation prior from circuit topology, block-aligned parameters, and noisy measurement features. (b) **Lightweight online adaptation:** during deployment, the pretrained branch encoders remain frozen while only the fusion and output modules are adapted every K evaluations under self-supervision without ideal-energy labels. (c) **In-loop VQE deployment:** the adapted mitigator is inserted directly into the VQE loop and supplies the mitigated objective used by the classical optimizer to update $\boldsymbol{\theta}_{t+1}$, so mitigation acts during the VQE process rather than as post-hoc energy mitigation.

predictor would be trained to align the mitigated objective with the latent ideal energy. For a trajectory $\tau = \{(\boldsymbol{\theta}_t, \mathbf{x}_t)\}_{t=1}^{T}$, this oracle trajectory-level objective can be written as

$$\mathcal{L}_{\text{sup}}(\phi; \tau) = \left(h_\phi(\mathbf{x}_t) - E_t^{\text{ideal}}\right)^2, \phi^* = \arg\min_\phi \mathcal{L}_{\text{sup}}(\phi) \tag{6}$$

for each step t, where E_t^{ideal} is defined in Sec. 2.2. Equivalently, the objective is to make the supplied mitigated objective $h_{\phi_t}(\mathbf{x}_t)$ approximate the ideal energy E_t^{ideal} at each visited parameter point. The resulting in-loop mitigation pipeline can be summarized as

$$\boldsymbol{\theta}_t \xrightarrow{\mathcal{B}_{\eta_t}} \left(E_t^{\text{noisy}}, \mathbf{x}_t\right) \xrightarrow{h_{\phi_t}} h_{\phi_t}(\mathbf{x}_t) \xrightarrow{\mathcal{O}} \boldsymbol{\theta}_{t+1},$$

where $\mathcal{B}_{\eta_t}$ denotes noisy backend execution; $\mathcal{O}$ denotes the classical VQE optimization rule.

However, during the VQE optimization loop, ideal-energy labels are unavailable. We therefore design a model which first pretrains the predictor offline with supervised ideal-energy labels to learn a structural prior, and adapt it online using label-free signals extracted from noisy measurements along the optimizer-induced trajectory. The goal is to provide an in-loop objective aligned with the ideal VQE landscape to obtain $\boldsymbol{\theta}^*$, rather than a pointwise fit to isolated noisy observations.

3.2 Problem-Scoped Cross-Ansatz Pretraining

Pretraining Dataset Construction. For the main benchmarks, we fix the target Hamiltonian H and construct an offline dataset from its ansatz pool $\mathcal{S}_H = \{s_H^{(1)}, \ldots, s_H^{(K)}\}$, where each $s_H^{(k)}$ specifies a parameterized ansatz associated with H. For instance, for a fixed TFIM Hamiltonian, $\mathcal{S}_H$ may include both HEA and HVA ansatz instances. We sample $s_i \sim \text{Unif}(\mathcal{S}_H)$, $\boldsymbol{\theta}_i \sim \text{Unif}\left([-\pi, \pi]^{d_{s_i}}\right)$, and use the ideal energy under H as the supervised label, i.e., $E_i^{\text{ideal}} = E^{\text{ideal}}(s_i, \boldsymbol{\theta}_i) = \langle 0|U_{s_i}^\dagger(\boldsymbol{\theta}_i) H U_{s_i}(\boldsymbol{\theta}_i)|0\rangle$. Appendix E.6 additionally evaluates a shared model across Hamiltonian coefficient values with fixed Pauli support, ansatz architecture, and parameter indexing. Prior ML-QEM datasets typically provide limited trajectory-level context [3] and do not explicitly encode

variational-parameter information [44; 45]. We therefore represent each offline circuit i using a hybrid feature tuple

$$\mathbf{x}_i = \left(\mathcal{G}_{s_i}, \widetilde{\boldsymbol{\theta}}_i, \mathbf{m}_i\right), \tag{7}$$

which jointly captures the ansatz structure $\mathcal{G}_{s_i}$, the variational state within a shared cross-ansatz coordinate system $\widetilde{\boldsymbol{\theta}}_i$, and noisy Hamiltonian measurement statistics $\mathbf{m}_i$. Accordingly, the pretraining dataset for a given H is denoted as $\mathcal{D}_{\text{pre}}^{H} = \left\{\left(\mathbf{x}_i, E_i^{\text{ideal}}\right)\right\}_{i=1}^{N}$.

For the *circuit branch*, we construct $\mathcal{G}_{s_i}$ from the circuit DAG of ansatz instance s_i. The graph contains input, output, and gate-operation nodes, with directed edges following the circuit dependency structure. Gate nodes are annotated with operation type, acted qubits, parameterization status, and normalized positional features such as operation and layer indices. We also include lightweight circuit-level statistics, including depth, size, qubit count, number of trainable parameters, and one- and two-qubit gate counts. This representation provides an ansatz-structure prior, enabling the model to exploit circuit topology beyond isolated noisy energy observations.

For the *parameter branch*, ansatz-dependent parameter vectors are embedded into a shared coordinate space defined by the ansatz pool. We assign each ansatz instance $s \in \mathcal{S}_H$ a disjoint coordinate block $\mathcal{I}_s \subseteq \{1, \ldots, d_{\text{blk}}(H)\}$, where $d_{\text{blk}}(H) = \sum_{s \in \mathcal{S}_H} d_s$, $|\mathcal{I}_s| = d_s$. We then define an injection $\Pi_s : \mathbb{R}^{d_s} \to \mathbb{R}^{d_{\text{blk}}(H)}$ that places the local parameter vector into the block $\mathcal{I}_s$ and sets all remaining coordinates to zero. For sample i, the resulting block-aligned parameter representation is $\widetilde{\boldsymbol{\theta}}_i = \Pi_{s_i}(\boldsymbol{\theta}_i) \in \mathbb{R}^{d_{\text{blk}}(H)}$. This shared representation allows a single parameter encoder to process different ansatz instances while preserving their ansatz-specific coordinate structure.

For the *measurement branch*, $\mathbf{m}_i$ includes finite-shot noisy Hamiltonian observations and available history context. Executing $U_{s_i}(\boldsymbol{\theta}_i)$ under backend condition η_t yields $E_t^{\text{noisy}} = \mathbf{c}^\top \widehat{\mathbf{p}}_t^{\text{noisy}}$, where $\mathbf{c} \in \mathbb{R}^L$ is the Hamiltonian coefficient vector and $\widehat{\mathbf{p}}_t^{\text{noisy}} \in \mathbb{R}^L$ is the noisy term-wise Pauli expectation vector. The associated term-wise contribution vector is $\widehat{\mathbf{r}}_t^{\text{noisy}} = \mathbf{c} \odot \widehat{\mathbf{p}}_t^{\text{noisy}}$, where $\odot$ denotes element-wise multiplication. Additionally, we define trajectory context as $\boldsymbol{\kappa}_t = \left[E_{t-1}^{\text{noisy}}; E_{t-1}^{\text{miti}}; \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}\|_2\right]$, when $t > 1$ for adaptation phase regularization (detailed in 3.3). It is set to $\mathbf{0}$ during offline pretraining. Overall, the measurement feature is $\mathbf{m}_i = \left[E_i^{\text{noisy}}; \widehat{\mathbf{p}}_i^{\text{noisy}}; \widehat{\mathbf{r}}_i^{\text{noisy}}; \boldsymbol{\kappa}_i\right]$. The ordered Pauli strings fix the meaning of each term-wise measurement position. Coefficient changes preserve this correspondence; a different operator support requires a new input alignment.

Multi-branch feature encoder. Given the hybrid input $\mathbf{x}_i = \left(\mathcal{G}_{s_i}, \widetilde{\boldsymbol{\theta}}_i, \mathbf{m}_i\right)$, we encode the circuit, parameter, and measurement branches into a latent space:

$$\mathbf{z}_i^c = h_{\phi_c}(\mathcal{G}_{s_i}), \qquad \mathbf{z}_i^p = h_{\phi_p}(\widetilde{\boldsymbol{\theta}}_i), \qquad \mathbf{z}_i^m = h_{\phi_m}(\mathbf{m}_i). \tag{8}$$

Here h_{ϕ_c}, h_{ϕ_p}, and h_{ϕ_m} denote the circuit, parameter, and measurement encoders, respectively.

The circuit encoder $h_{\phi_c}^c$ is implemented as a graph Transformer over the circuit DAG and outputs a circuit-level latent embedding $\mathbf{z}_i^c$. To make self-attention aware of the circuit topology and quantum system structure, we adopt a shortest-path structural bias to each attention logit:

$$\alpha_{uv}^{(\rho)} = \frac{(\mathbf{q}_u^{(\rho)})^\top \mathbf{k}_v^{(\rho)}}{\sqrt{d_\rho}} + \beta_\rho(\min(D_{uv}, D_{\max})), \tag{9}$$

where u and v denote circuit-graph nodes, ρ denotes the attention head, $\mathbf{q}_u^{(\rho)}, \mathbf{k}_v^{(\rho)} \in \mathbb{R}^{d_\rho}$ are the query and key vectors, and D_{uv} is the shortest-path distance between u and v on the circuit DAG. The learnable embedding $\beta_\rho(\cdot)$ maps the clipped distance bucket to an attention bias.

The parameter encoder h_{ϕ_p} and measurement encoder h_{ϕ_m} are multi-layer perceptrons with nonlinear activation, normalization, and a final projection to the same latent dimension as the circuit branch. After the three branch embeddings are obtained, we form a short modality-token sequence $\mathbf{T}_i = [\mathbf{z}_{\text{cls}}; \mathbf{z}_i^c + \mathbf{e}^c; \mathbf{z}_i^p + \mathbf{e}^p; \mathbf{z}_i^m + \mathbf{e}^m]$, where $\mathbf{e}^c$, $\mathbf{e}^p$, and $\mathbf{e}^m$ are learnable modality-type embeddings, following the common practice of adding segment embeddings to distinguish token sources in Transformer inputs [25; 19]. The fusion module h_{ϕ_f} aggregates the branch-token sequence $\mathbf{T}_i$ into a fused latent representation, from which the output head predicts the mitigated objective:

$$\mathbf{Z}_i = h_{\phi_f}(\mathbf{T}_i), \qquad \mathbf{z}_i = [\mathbf{Z}_i]_{\text{cls}}, \qquad E_i^{\text{miti}} = h_{\phi_{\text{out}}}(\mathbf{z}_i). \tag{10}$$

Offline ideal-energy learning. Given the offline dataset $\mathcal{D}_{\mathrm{pre}}^{H}$, we train the mitigator by supervised ideal-energy regression:

$$\phi^{\mathrm{pre}} \in \arg\min_{\phi} \mathcal{L}_{\mathrm{pre}}(\phi; H), \qquad \mathcal{L}_{\mathrm{pre}}(\phi; H) = \frac{1}{N} \sum_{i=1}^{N} \left(E_i^{\mathrm{miti}} - E_i^{\mathrm{ideal}}\right)^2. \tag{11}$$

Trajectory-induced distribution shift. Although the predictor is pretrained on offline samples from the problem-scoped ansatz pool, in-loop deployment exposes it to the non-uniform input sequence generated by the VQE optimizer. We denote this mismatch as

$$\mathcal{P}_H^{\mathrm{pre}}(\mathbf{x}) \neq \mathcal{P}_H^{\mathrm{traj}}(\mathbf{x}), \tag{12}$$

where $\mathcal{P}_H^{\mathrm{pre}}$ is the offline pretraining distribution for Hamiltonian H, and $\mathcal{P}_H^{\mathrm{traj}}$ is the trajectory-induced distribution encountered during in-loop optimization. This shift is important as VQE does not explore the parameter space uniformly; instead, it visits a local, history-dependent region shaped by previous noisy or mitigated objective values. Consequently, a frozen pretrained predictor may retain useful ansatz-level priors while remaining locally misaligned with the deployment trajectory. We therefore use the pretrained predictor $h_{\phi^{\mathrm{pre}}}$ as the initialization for trajectory-aware online adaptation.

3.3 Self-Supervised Trajectory-Aware Adaptation

Constrained adaptation around the pretrained objective prior. During online deployment, H and $s \in \mathcal{S}_H$ are fixed, while the VQE optimizer generates a trajectory $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$. Since the ideal energies E_t^{ideal} are unavailable, online adaptation must rely on label-free trajectory signals while preserving the representations learned during offline pretraining. We denote the pretrained parameter state by ϕ^{pre}. During online adaptation, the circuit, parameter, and measurement encoders are frozen at their pretrained values $\phi_c^{\mathrm{pre}}, \phi_p^{\mathrm{pre}}, \phi_m^{\mathrm{pre}}$, while only the fusion module and output head are updated. We collect these trainable parameters as

$$\psi_t = \left(\phi_c^{\mathrm{pre}}, \phi_p^{\mathrm{pre}}, \phi_m^{\mathrm{pre}}, \phi_{f,t}, \phi_{\mathrm{out},t}\right), \qquad \psi_{\mathrm{pre}} = \left(\phi_c^{\mathrm{pre}}, \phi_p^{\mathrm{pre}}, \phi_m^{\mathrm{pre}}, \phi_f^{\mathrm{pre}}, \phi_{\mathrm{out}}^{\mathrm{pre}}\right). \tag{13}$$

Omitting the fixed encoder parameters from the notation, we write the adapted predictor at step t as $h_{\psi_t}^{\mathrm{adp}}$. This constrained adaptation recalibrates the final objective mapping along the VQE trajectory without overwriting the pretrained branch representations.

Latent-energy consistency. At VQE step t, two independent finite-shot evaluations of the same circuit may yield different noisy measurement records, but they share the same latent ideal energy. We therefore require the adapted predictor to produce consistent mitigated objectives across such views, which provides a label-free calibration signal and discourages fitting view-specific shot fluctuations. For each view $a \in \{1, 2\}$, let $\mathbf{m}_{t,a}$ denote the corresponding measurement feature, and $\mathbf{x}_{t,a} = \left(\mathcal{G}_s, \widetilde{\boldsymbol{\theta}}_t, \mathbf{m}_{t,a}\right)$. We impose the consistency loss

$$\mathcal{L}_{\mathrm{cons}}(\psi; t) = \left(h_{\psi_t}(\mathbf{x}_{t,1}) - h_{\psi_t}(\mathbf{x}_{t,2})\right)^2. \tag{14}$$

Trajectory stability. To further stabilize online adaptation, we regularize the mitigation offset along the optimizer-induced trajectory, discouraging abrupt changes between consecutive VQE evaluations. Specifically, for each noisy view $a \in \{1, 2\}$, define the signed mitigation offset $\delta_{t,a} = h_{\psi_t}(\mathbf{x}_{t,a}) - E_{t,a}^{\mathrm{noisy}}$. The trajectory-stability loss is then defined as

$$\mathcal{L}_{\mathrm{stab}}(\psi; t) = \left(\bar{\delta}_t - \bar{\delta}_{t-1}\right)^2, \qquad \bar{\delta}_t = \frac{1}{2} \sum_{a=1}^{2} \delta_{t,a}. \tag{15}$$

where $\bar{\delta}_{t-1}$ is the recorded mean offset from the previous VQE evaluation and is treated as fixed during the current update. This regularizer does not assume that the ideal energy changes smoothly; it only constrains the local variation of the mitigation offset supplied to the optimizer.

Weak prior-anchor regularization. The preceding consistency and smoothness losses constrain relative behavior across noisy views and neighboring trajectory steps, but without a feasible anchor, the online predictor may converge to a view-consistent and trajectory-smooth surrogate that is

nevertheless misaligned with the latent ideal objective. We therefore use the frozen pretrained predictor as a weak online anchor by defining

$$\mathcal{L}_{\text{anchor}}(\psi; t) = \frac{1}{2} \sum_{a=1}^{2} \left(h_{\psi_t}(\mathbf{x}_{t,a}) - h_{\psi_{pre}}(x_{t,a}) \right)^2 . \tag{16}$$

This term prevents the self-supervised update from drifting to an arbitrary view-consistent surrogate.

The full online adaptation objective at step t is

$$\mathcal{L}_{\text{adapt}}(\psi; t) = \lambda_{\text{cons}} \mathcal{L}_{\text{cons}}(\psi; t) + \lambda_{\text{stab}} \mathcal{L}_{\text{stab}}(\psi; t) + \lambda_{\text{anchor}} \mathcal{L}_{\text{anchor}}(\psi; t), \tag{17}$$

where $\lambda_{\text{cons}}, \lambda_{\text{stab}}, \lambda_{\text{anchor}} \geq 0$ control the relative strength of the three regularizers. We set λ_{anchor} small so that the frozen pretrained predictor acts as a weak trust-region prior, rather than forcing the adapted predictor to follow the pretrained landscape exactly. After adaptation at step t, the view-averaged mitigated objective $\bar{E}_t^{\text{miti}}$ is passed to the classical optimizer. For efficiency, we trigger the online adaptation every K VQE steps and reuse the adapted predictor between two calibration events. Overall, the method is trajectory-adaptive: it incorporates history-aware context features from previous VQE evaluations and updates the objective predictor by self-supervised signals extracted solely from noisy measurements along the optimizer-induced trajectory.

4 Experiments

4.1 Setups

Benchmarks. We evaluate our method on five VQE settings spanning spin systems, molecular energy estimation, combinatorial optimization, and large-scale many-body optimization. For spin systems, we use the transverse-field Ising model (TFIM). The 4-qubit TFIM setting is evaluated with both a hardware-efficient ansatz (TFIM4-HEA) and a Hamiltonian variational ansatz (TFIM4-HVA). The large-scale setting uses a 50-qubit TFIM Hamiltonian with a depth-one HVA (TFIM50-HVA), serving as the main scalability benchmark. To test generalization beyond spin Hamiltonians, we also include an 8-qubit 3-regular MaxCut instance with a multi-angle QAOA ansatz (MaxCut-MA-QAOA) and an H_4 molecular Hamiltonian at bond factor 1.3 with a hardware-efficient no-Hartree–Fock ansatz (H_4-HEA). Details are given in Appendix C.1.

Noise Setting. Following the configurations set in [44; 3; 4], each benchmark is evaluated under three noise regimes: an incoherent noise setting including the combination of Pauli, depolarizing and readout error (`Incoh.`), an IBM fake-provider noise setting (`Prov.`), and a mixed coherent-plus-provider setting (`Mixed`) that adds systematic two-qubit over-rotations to provider-style noise. Detailed noise parameters and backend choices are provided in Appendix C.2.

Baselines. We compare with representative baselines that can be deployed inside the VQE loop. Each baseline must return an optimizer-facing energy estimate at every queried parameter, rather than only post-processing the final measurement trace. The comparison includes raw noisy VQE, Zero-Noise Extrapolation (ZNE) [62; 43; 38], Clifford Data Regression (CDR) [21; 48], active-learning Gaussian-process regression (GPR) [36], and flat-feature supervised ML-QEM regressors including OLS, RF, and MLP [39; 44]. For fairness, the supervised regressors are trained on the same offline data as our method, but use flat classical features and remain frozen during VQE. CDR is implemented as a periodically refreshed local calibration baseline, while GPR uses a trajectory-collected posterior surrogate whose mean is passed to the optimizer. Details are given in Appendix C.3.

4.2 Evaluation Metrics

Conventional QEM is usually evaluated pointwise, by how accurately a mitigator recovers the ideal expectation value of a fixed circuit. This criterion is insufficient for in-loop VQE mitigation: a mitigator may look favorable by reporting overly optimistic energies at queried points, while still failing to guide the optimizer toward better parameters. As a minor evaluation contribution, we therefore introduce a trajectory-level protocol for VQE mitigation. For each visited parameter $\boldsymbol{\theta}_t$, all metrics are computed from the ideal gap δ_t defined in Sec. 2.2, rather than from noisy or mitigated energies. This avoids rewarding a model that merely reports overly optimistic energies and instead measures whether the optimizer actually visits better points on the ideal landscape. **We introduce**

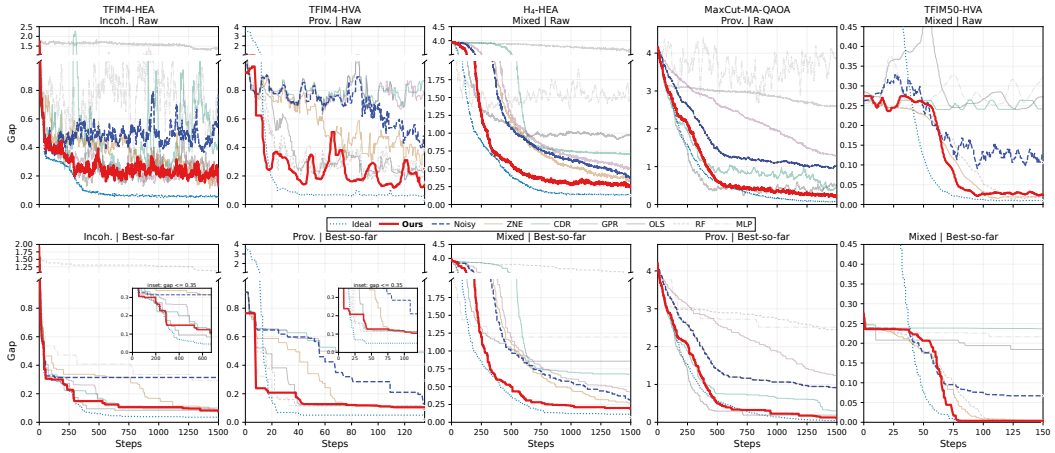


Figure 2: Historical individual-run VQE trajectories across benchmarks (not multi-seed means). The top row shows the raw training ideal gap, and the bottom row shows the cumulative best-so-far ideal gap. **Lower gap with faster and stable convergence is better. Additional results are in Fig. 6.**

Method	TFIM4-HEA Prov.				TFIM4-HVA Mixed				MaxCut-MA-QAOA Incoh.				H_4-HEA Prov.				TFIM50-HVA Incoh.			
	Min	Final	nAUBG	TTC	Min	Final	nAUBG	TTC	Min	Final	nAUBG	TTC	Min	Final	nAUBG	TTC	Min	Final	nAUBG	TTC
Ideal	0.0161	0.0161	0.087	561	0.0182	0.0182	0.009	35	0.0130	0.0130	0.170	783	0.0691	0.0691	0.093	542	0.0030	0.0030	1.063	127
Noisy	0.2827	0.3511	0.294	DNF	0.2762	2.1104	0.110	584	0.2336	0.2484	0.338	1596	0.2492	0.2492	0.211	926	0.0031	0.0044	0.683	536
ZNE	0.0977	0.3476	0.191	1498	0.7648	1.8424	0.273	DNF	0.3262	0.3334	0.368	1939	0.2140	0.2140	0.162	1017	0.0030	0.0030	0.584	461
CDR	0.1197	0.4746	0.166	638	0.1841	4.7215	0.099	133	0.1369	0.1369	0.195	952	0.6504	0.6504	0.334	2267	0.2394	0.2456	0.905	DNF
GPR	0.0673	0.1924	0.188	1505	0.7648	2.3322	0.273	DNF	2.5682	2.5910	0.744	DNF	0.4296	0.4296	0.258	1254	–	–	–	–
OLS	0.1417	0.1890	0.307	2969	0.2447	0.4986	0.209	503	0.1353	0.1800	0.245	1238	0.1418	0.1418	0.133	613	0.0031	0.0031	0.474	379
RF	1.0204	1.6560	1.103	DNF	0.7648	3.9426	0.273	DNF	2.5820	2.6742	0.660	DNF	3.7389	3.9846	0.966	DNF	0.2327	0.2448	0.904	DNF
MLP	0.4659	0.9280	0.464	DNF	0.5050	0.9903	0.202	214†	2.3321	3.0424	0.649	DNF	0.9448	1.3578	0.283	763†	0.0053	0.3756	0.407	DNF
TrACE	0.0608	0.0930	0.0893	407	0.1160	0.3418	0.0504	80	0.0932	0.1106	0.1428	701	0.0975	0.0975	0.0601	250	0.0031	0.0032	0.2467	210
± SD	±0.0046	±0.0076	±0.0011	±29	±0.0090	±0.0204	±0.0041	±9	±0.0054	±0.0090	±0.0123	±45	±0.0017	±0.0018	±0.0055	±28	±0.0002	±0.0002	±0.0089	±12

Table 1: Representative in-loop VQE mitigation results. TrACE shows means over three random seeds, with SD underneath; baselines are individual runs. Gap and nAUBG are lower-is-better. Stable-TTC ranks primary 20% convergence ahead of 30% fallback (†), then DNF; within each category, lower is better. Best and second-best non-Ideal values at source precision are coral red and navy blue. Full results and reporting protocol: Tab. 4 and Appendix C.4.

four lower-is-better metrics. *Min Gap* is $\min_t \delta_t$, measuring the best parameter discovered along the trajectory. *Final Gap* is δ_T, measuring the quality of the returned final iterate. *nAUBG* is the normalized area under the best-so-far gap curve, so lower values indicate that good parameters are found earlier within the optimization budget. *Stable-TTC* is the first objective-evaluation count at which the trajectory satisfies a sustained low-error criterion, capturing robust usable convergence rather than a transient good visit. Runs that never satisfy this criterion are marked as DNF. Details are given in Appendix C.5.

4.3 Experimental Results Analysis

We analyze whether mitigation improves the VQE optimization trajectory rather than only the reported energy estimate. Tab. 1 presents the five original representative settings; Appendix Tab. 4 gives all 15 benchmark–noise settings. At source precision and with the stated TTC category ordering, TrACE ranks first in 53 of 60 metric columns and within the top two in 56 of 60. All four metrics rank first in 9 of 15 settings and within the top two in 12 of 15. All 15 TrACE settings use three-run means with standard deviations; baseline entries retain individual-run results. These descriptive rankings are not paired significance tests. Reporting details are in Appendix C.4. **Performance across problem classes.** Tab. 4 shows improvements over noisy VQE across spin, molecular, and combinatorial benchmarks. On TFIM4-HEA, TrACE achieves the best Final Gap under all three noise regimes: 0.0669, 0.2740, and 0.0930 under Incoh., Mixed, and Prov., compared with Noisy values 0.6349, 0.6905, and 0.3511. On MaxCut-MA-QAOA, it retains the best Min Gap and Final Gap in all three regimes; the Final Gaps are 0.1106 ± 0.0090, 0.0723 ± 0.0071, and 0.0937 ± 0.0013. On

H_4-HEA/Mixed, the Final Gap is 0.2224 ± 0.0056, compared with the Noisy reference 0.3373. These results support the pretrain-plus-adapt recipe across distinct problem classes, beyond pointwise energy calibration. The individual-run trajectories in Fig. 2 and 6 illustrate earlier progress and sustained low-gap behavior in representative runs. **Performance across ansatz structures.** TFIM4-HEA and TFIM4-HVA use different ansatz families for the same Hamiltonian. This setting tests whether the shared pretraining prior supports ansatz-level transfer with separate online adaptation for each ansatz. While TrACE has the best Final Gap on TFIM4-HEA in all three regimes, on TFIM4-HVA it achieves the best Min Gap and nAUBG across the three regimes. All four metrics rank first only under Mixed noise. Incoh. also has the best Final Gap, but its fallback Stable-TTC ranks fourth; Prov. has the best Stable-TTC, but its Final Gap ranks seventh. This supports cross-ansatz reuse. These comparisons do not isolate architecture from other setting-specific configurations. Fig. 2 and 6 retain the original individual-run trajectories to illustrate optimization dynamics, rather than plotting multi-run means. **Performance on large-scale TFIM50.** On the 50-qubit TFIM-HVA benchmark, TrACE achieves the best Stable-TTC and best or second-best nAUBG in all three regimes. The three-run Final Gaps are 0.0032 ± 0.0002 under Incoh. and 0.003134 ± 0.000090 under Prov., with Stable-TTC 210 ± 12 and 190 ± 21. The Mixed-noise Final Gap is 0.0181 ± 0.0036, compared with 0.1756 for Noisy. The Prov. mean has the smallest endpoint; ZNE and OLS have smaller Incoh. endpoints, and RF has a smaller Mixed endpoint. Thus endpoint gains remain setting-dependent. These convergence gains support the pretrain-plus-adapt recipe at larger scale through earlier, sustained access to low-gap parameters.

Overall, TrACE helps the optimizer find better variational parameters with setting-dependent gains in convergence speed and stability, supporting its role as a practical in-loop mitigated objective rather than a post-hoc energy estimator.

4.4 Ablation Studies

Additional encoder, adaptation-interval, loss-weight, data-size, and label-fidelity studies are reported in Appendix E. Appendix E.6 demonstrates transfer to an unseen TFIM coefficient value with fixed Pauli support and aligned ansatz coordinates. Supplementary experiments on 100-qubit TFIM, depth-258 Fermi–Hubbard circuits, and wall-clock cost are reported in Appendices E.7–E.9. We ablate both the main components of TrACE and the design of the online adaptation mechanism. The component ablation compares raw noisy VQE, frozen pretraining, adaptation from scratch, and the full pretrain-plus-adapt recipe on representative settings including MaxCut-MA-QAOA/Incoh. and TFIM50-HVA/Prov. The trajectories in Fig. 5 show that frozen pretraining gives a fair optimizer-facing objective, while adaptation from scratch is less stable and typically slower. The full TrACE variant generally reaches useful low-gap regions earlier and improves the best-so-far trajectory, especially on MaxCut-MA-QAOA/Incoh. and TFIM50-HVA/Prov. These results indicate that pretraining provides a useful structural prior, while online adaptation improves deployment-time calibration and convergence dynamics. We further ablate the adaptation scope and loss terms. Fig. 5 shows that fuser-head-only adaptation is usually more stable than updating the full mitigator, particularly on the chemistry and MaxCut settings. Removing smoothness or consistency generally worsens convergence stability or best-so-far quality, supporting the final lightweight head-only adaptation with both regularizers.

5 Conclusion

We have introduced TrACE, a trajectory-adaptive learning framework for in-loop VQE error mitigation. By combining problem-scoped pretraining, structured hybrid circuit–measurement representations and lightweight label-free online adaptation, TrACE achieves best or second-best reported values in 14/15 settings for Min Gap, 13/15 for Final Gap, 15/15 for nAUBG, and 14/15 for Stable-TTC among physics-based, learning-based, and VQE-specific baselines. All 15 settings include three-run summaries; these descriptive comparisons do not establish population-wide robustness. Although there is additional cost from pretraining and online adaptation, this cost can be tuned through the ansatz-pool size, measurement budget, and adaptation interval, and amortized across related Hamiltonian–ansatz deployments, as discussed in Sec. B. In return, TrACE adapts the optimizer-facing objective to the trajectory encountered during noisy VQE, helping the classical optimizer discover lower-gap parameters and maintain more stable convergence.

AI USE STATEMENT

In this work, we used generative AI tools for drafting assistance and editing support during manuscript preparation, including assistance with LaTeX template conversion and result-table preparation. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops learning-based quantum error mitigation for variational quantum eigensolvers and evaluates it on simulated spin-system, molecular-energy, and combinatorial-optimization benchmarks under noise models, including models derived from IBM fake providers. It involves no human subjects or personal data. We do not foresee direct ethical concerns; potential societal impacts are indirect, through improvements in the reliability and efficiency of noisy quantum optimization. The scope and limitations of the method are discussed in Appendix B.

REPRODUCIBILITY STATEMENT

The benchmark Hamiltonians and ansatz families are described in Section 4.1 and Appendix C.1, and the noise configurations in Appendix C.2. The mitigation architecture and offline pretraining objective are specified in Section 3.2, and the self-supervised online adaptation procedure in Section 3.3. Optimizer settings and pretraining dataset sizes are listed in Tab. 2 and 3. Baseline implementations and trajectory-level metric definitions are described in Appendices C.3 and C.5, respectively. Ablation studies and additional results are provided in Section 4.4 and Appendices C.6 and E; the multi-seed aggregation protocol is given in Appendix C.4.

REFERENCES

[1] Panagiotis G. Anastasiou, Yanzhu Chen, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. TETRIS-ADAPT-VQE: An adaptive algorithm that yields shallower, denser circuit ansätze. *Physical Review Research*, 6(1):013254, 2024.

[2] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[3] Tianyi Bao, Xinyu Ye, Hang Ruan, Chang Liu, Wenjie Wu, and Junchi Yan. Beyond circuit connections: A non-message passing graph transformer approach for quantum error mitigation. In *International Conference on Learning Representations*, 2025.

[4] Tianyi Bao, Ruizhe Zhong, Xinyu Ye, Yehui Tang, and Junchi Yan. QEM-bench: Benchmarking learning-based quantum error mitigation and QEMFormer as a multi-ranged context learning baseline. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 2953–2967. PMLR, 2025.

[5] Marcello Benedetti, Erika Lloyd, Stefan H. Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.

[6] Elizabeth R. Bennewitz, Florian Hopfmueller, Bohdan Kulchytskyy, Juan Carrasquilla, and Pooya Ronagh. Neural error mitigation of near-term quantum simulations. *Nature Machine Intelligence*, 4:618–624, 2022.

[7] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S. Kottmann, Tim Menke, Wai-Keong Mok, Sukin Sim, Leong-Chuan Kwek, and Alán Aspuru-Guzik. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94:015004, 2022.

[8] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[9] Nick S. Blunt, Joan Camps, Ophelia Crawford, Róbert Izsák, Sebastian Leontica, Arjun Mirani, Alexandra E. Moylett, Sam A. Scivier, Christoph Sünderhauf, Patrick Schopf, Jacob M. Taylor, and Nicole Holzmann. A perspective on the current state-of-the-art of quantum computing for drug discovery applications, 2022.

[10] Xavi Bonet-Monroig, Ramiro Sagastizabal, M. Singh, and Thomas E. O'Brien. Low-cost error mitigation by symmetry verification. *Physical Review A*, 98:062339, 2018.

[11] Zhenyu Cai. Quantum error mitigation using symmetry expansion. *Quantum*, 5:548, 2021.

[12] Zhenyu Cai, Ryan Babbush, Simon C. Benjamin, Suguru Endo, William J. Huggins, Ying Li, Jarrod R. McClean, and Thomas E. O'Brien. Quantum error mitigation. *Reviews of Modern Physics*, 95:045005, 2023.

[13] Simone Cantori, Andrea Mari, David Vitali, and Sebastiano Pilati. Deep-learned error mitigation via partially knitted circuits for the variational quantum eigensolver. *arXiv preprint arXiv:2506.04146*, 2025.

[14] Chenfeng Cao, Hiroshi Yano, and Yuya O. Nakagawa. Accelerated variational quantum eigensolver with joint Bell measurement. *Physical Review Research*, 6(1):013205, 2024.

[15] Yudong Cao, Jonathan Romero, and Alán Aspuru-Guzik. Potential of quantum computing for drug discovery. *IBM Journal of Research and Development*, 62(6):6:1–6:20, 2018.

[16] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, et al. Quantum chemistry in the age of quantum computing. *Chemical Reviews*, 119 (19):10856–10915, 2019.

[17] Michele Cattelan, Sheir Yarkoni, and Wolfgang Lechner. Parallel circuit implementation of variational quantum algorithms. *npj Quantum Information*, 11(1):27, 2025.

[18] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3:625–644, 2021.

[19] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pp. 104–120, 2020.

[20] Mucio A. Continentino. *Key Methods and Concepts in Condensed Matter Physics*. IOP Publishing, 2021. ISBN 9780750333955.

[21] Piotr Czarnik, Andrew Arrasmith, Patrick J. Coles, and Lukasz Cincio. Error mitigation with clifford quantum-circuit data. *Quantum*, 5:592, 2021.

[22] Piotr Czarnik, Michael McKerns, Andrew T. Sornborger, and Lukasz Cincio. Improving the efficiency of learning-based error mitigation. *Quantum*, 9:1727, 2025.

[23] Peter Deglmann, Ansgar Schäfer, and Christian Lennartz. Application of quantum calculations in the chemical industry—an overview. *International Journal of Quantum Chemistry*, 115(3): 107–136, 2014.

[24] Anton Van der Ven, Zhi Deng, S. Banerjee, and Shyue Ping Ong. Rechargeable alkali-ion battery materials: Theory and computation. *Chemical Reviews*, 120(14):6977–7019, 2020.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[26] Suguru Endo, Simon C. Benjamin, and Ying Li. Practical quantum error mitigation for near-future applications. *Physical Review X*, 8:031027, 2018.

[27] Enrico Fontana, Nathan Fitzpatrick, David Muñoz Ramo, Ross Duncan, and Ivan Rungger. Evaluating the noise resilience of variational quantum algorithms. *Physical Review A*, 104: 022403, 2021.

[28] Daniel Stilck França and Raúl García-Patrón. Limitations of optimization algorithms on noisy quantum devices. *Nature Physics*, 17(11):1221–1227, 2021.

[29] Tudor Giurgica-Tiron, Yousef Hindy, Ryan LaRose, Andrea Mari, and William J. Zeng. Digital zero noise extrapolation for quantum error mitigation. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pp. 306–316. IEEE, 2020.

[30] Harper R. Grimsley, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, 10(1):3007, 2019.

[31] Daniel Gunlycke, C. Stephen Hellberg, and John P. T. Stenger. Cascaded variational quantum eigensolver algorithm. *Physical Review Research*, 6(1):013238, 2024.

[32] Erik J. Gustafson, Juha Tiihonen, Diana Chamaki, Farshud Sorourifar, J. Wayne Mullinax, Andy C. Y. Li, Filip B. Maciejewski, Nicolas P. D. Sawaya, Jaron T. Krogel, David E. Bernal Neira, and Norm M. Tubman. Surrogate optimization of variational quantum circuits. *Proceedings of the National Academy of Sciences*, 122(36):e2408530122, 2025.

[33] Andre He, Benjamin Nachman, Wibe A. de Jong, and Christian W. Bauer. Zero-noise extrapolation for quantum-gate error mitigation with identity insertions. *Physical Review A*, 102:012426, 2020.

[34] Alexander Heifetz (ed.). *Quantum Mechanics in Drug Discovery*. Springer US, 2020. ISBN 9781071602829.

[35] William J. Huggins, Sam McArdle, Thomas E. O'Brien, Joonho Lee, Nicholas C. Rubin, Sergio Boixo, K. Birgitta Whaley, Ryan Babbush, and Jarrod R. McClean. Virtual distillation for quantum error mitigation. *Physical Review X*, 11:041036, 2021.

[36] Tao Jiang, John Rogers, Marius S. Frank, Ove Christiansen, Yong-Xin Yao, and Nicola Lanatà. Error mitigation in variational quantum eigensolvers using tailored probabilistic machine learning. *Physical Review Research*, 6(3):033069, 2024.

[37] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

[38] Abhinav Kandala, Kristan Temme, Antonio D. Córcoles, Antonio Mezzacapo, Jerry M. Chow, and Jay M. Gambetta. Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, 567(7749):491–495, 2019.

[39] Changjun Kim, Kyungdeock Daniel Park, and June-Koo Rhee. Quantum error mitigation with artificial neural network. *IEEE Access*, 8:188853–188860, 2020.

[40] Bálint Koczor. Exponential error suppression for near-term quantum devices. *Physical Review X*, 11:031057, 2021.

[41] Tomochika Kurita, Hammam Qassim, Masatoshi Ishii, Hirotaka Oshima, Shintaro Sato, and Joseph Emerson. Synergetic quantum error mitigation by randomized compiling and zero-noise extrapolation for the variational quantum eigensolver. *Quantum*, 7:1184, 2023.

[42] Jeffrey Larson, Matt Menickelly, and Jiahao Shi. A novel noise-aware classical optimizer for variational quantum algorithms. *INFORMS Journal on Computing*, 37(1):63–85, 2025.

[43] Ying Li and Simon C. Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7:021050, 2017.

[44] Haoran Liao, Derek S. Wang, Iskandar Sitdikov, Ciro Salcedo, Alireza Seif, and Zlatko K. Minev. Machine learning for practical quantum error mitigation. *Nature Machine Intelligence*, 6:1478–1486, 2024.

[45] Manwen Liao, Yan Zhu, Giulio Chiribella, and Yuxiang Yang. Noise-agnostic quantum error mitigation with data augmented neural models. *npj Quantum Information*, 11:8, 2025.

[46] Wei-You Liao, Ge Yan, Yujin Song, Tian-Ci Tian, Wei-Ming Zhu, De-Tao Jiang, Yuxuan Du, and He-Liang Huang. Sample-efficient quantum error mitigation via classical learning surrogates. *arXiv preprint arXiv:2511.07092*, 2025.

[47] Vincenzo Lordi and John M. Nichol. Advances and opportunities in materials science for scalable quantum computing. *MRS Bulletin*, 46(7):589–595, 2021.

[48] Angus Lowe, Max Hunter Gordon, Piotr Czarnik, Andrew Arrasmith, Patrick J. Coles, and Lukasz Cincio. Unified approach to data-driven quantum error mitigation. *Physical Review Research*, 3:033098, 2021.

[49] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[50] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000. ISBN 9780521635035.

[51] Thomas E. O'Brien, Stefano Polla, Nicholas C. Rubin, William J. Huggins, Sam McArdle, Sergio Boixo, Jarrod R. McClean, and Ryan Babbush. Error mitigation via verified phase estimation. *PRX Quantum*, 2:020317, 2021.

[52] Srushti Patil, Dibyendu Mondal, and Rahul Maitra. Machine learning approach toward quantum error mitigation for accurate molecular energetics. *The Journal of Chemical Physics*, 163(2): 024129, 2025.

[53] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213, 2014.

[54] John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

[55] Yihui Quek, Daniel Stilck França, Sumeet Khatri, Johannes Jakob Meyer, and Jens Eisert. Exponentially tighter bounds on limitations of quantum error mitigation. *Nature Physics*, 20 (10):1648–1658, 2024.

[56] Stefan H. Sack and Daniel J. Egger. Large-scale quantum approximate optimization on nonplanar graphs with machine learning noise mitigation. *Physical Review Research*, 6:013223, 2024.

[57] Alireza Seif, Ze-Pei Cian, Sisi Zhou, Senrui Chen, and Liang Jiang. Shadow distillation: Quantum error mitigation with classical shadows for near-term quantum processors. *PRX Quantum*, 4:010303, 2023.

[58] Harjeet Singh, Luke W. Bertels, Daniel Claudino, Sophia E. Economou, Edwin Barnes, Nicholas P. Bauman, Karol Kowalski, and Nicholas J. Mayhall. Qubit-efficient quantum chemistry with the ADAPT variational quantum eigensolver and double unitary downfolding. *Journal of Chemical Theory and Computation*, 21(18):8799–8811, 2025.

[59] Armands Strikis, Dayue Qin, Yanzhu Chen, Simon C. Benjamin, and Ying Li. Learning-based quantum error mitigation. *PRX Quantum*, 2:040330, 2021.

[60] Ryuji Takagi, Suguru Endo, Shintaro Minagawa, and Mile Gu. Fundamental limits of quantum error mitigation. *npj Quantum Information*, 8(1):114, 2022.

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

[62] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119:180509, 2017.

[63] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. The variational quantum eigensolver: A review of methods and best practices. *Physics Reports*, 986:1–128, 2022.

[64] Ewout van den Berg, Zlatko K. Minev, Abhinav Kandala, and Kristan Temme. Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors. *Nature Physics*, 19(8):1116–1121, 2023.

[65] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F. Izmaylov. Measurement optimization in the variational quantum eigensolver using a minimum clique cover. *The Journal of Chemical Physics*, 152(12):124114, 2020.

[66] Samson Wang, Piotr Czarnik, Andrew Arrasmith, M. Cerezo, Lukasz Cincio, and Patrick J. Coles. Can error mitigation improve trainability of noisy variational quantum algorithms? *Quantum*, 8:1287, 2024.

[67] Billy J. Williams-Noonan, Elizabeth Yuriev, and David K. Chalmers. Free energy methods in drug design: Prospects of "alchemical perturbation" in medicinal chemistry. *Journal of Medicinal Chemistry*, 61(3):638–649, 2017.

[68] Andrey Zhukov and Walter V. Pogosov. Quantum error reduction with deep neural network applied at the post-processing stage. *Quantum Information Processing*, 21:93, 2022.

[69] Hang Zou, Martin Rahm, Anton Frisk Kockum, and Simon Olsson. Generative flow-based warm start of the variational quantum eigensolver. *npj Quantum Information*, 12(1):5, 2025.

A IMPACT

TrACE shows that learning-based mitigation can be used to improve the optimization performance of noisy VQE, rather than only to reduce pointwise energy-estimation error. By adapting to the optimizer-induced trajectory, TrACE provides an in-loop mitigated objective that leads to faster and more stable convergence and helps select variational parameters with better ideal quality. This direction is important for NISQ-era VQE, where hardware noise and finite-shot fluctuations can distort the optimization landscape and degrade the parameters found by the classical optimizer. Beyond the method itself, we propose a trajectory-level evaluation protocol for VQE mitigation, covering final quality, best-so-far quality, convergence speed, and stability. This protocol offers a more faithful assessment of whether mitigation improves the downstream variational optimization process, and can support more complete comparisons for future learning-based and hybrid mitigation methods.

B LIMITATIONS AND FUTURE WORK

TrACE has three main limitations. First, model reuse depends on aligned circuit and observable coordinates. The main benchmarks use target-specific models, while Appendix E.6 demonstrates shared pretraining and label-free adaptation within a TFIM family with fixed Pauli support and ordering, ansatz architecture, and parameter indexing. This provides evidence for transfer across coefficient changes within that family. Changing the Pauli operators can alter the physical meaning of term-wise measurement positions even when vector lengths match. Transfer that changes these operator or parameter semantics requires a new representation alignment and validation before reusing the model and is not evaluated here. Second, online adaptation introduces additional quantum and classical cost. In particular, the self-supervised consistency loss requires two noisy views of the same parameterized circuit when adaptation is performed. This increases the measurement budget relative to a single noisy VQE query. However, adaptation is not applied at every optimizer step in general; it can be performed every K steps or triggered only at selected trajectory points. Thus, unlike pointwise mitigation methods such as ZNE or PEC that attach extra executions to each circuit evaluation, the additional sampling cost of TrACE is intermittent and can be amortized over the optimization trajectory. Third, our experiments focus on shallow to moderate-depth ansatz circuits, including the large-qubit TFIM setting considered in this paper. The behavior of TrACE on substantially deeper circuits, where noise accumulation and trainability issues may be more severe, remains to be tested. This is an important direction for assessing the robustness of trajectory-aware learned mitigation beyond the present benchmark regime.

Future work will extend TrACE along three directions. First, we will extend the fixed-support TFIM transfer study to broader coefficient ranges and other Hamiltonian families. Changes in Pauli support or parameter semantics call for operator-aware representations and explicit alignment beyond the coefficient changes tested here. Second, the adaptation schedule and measurement allocation can be made adaptive, so that additional noisy views are requested only when the trajectory shows uncertainty, drift, or rapid local change. Third, we will evaluate and improve the method on deeper ansatz circuits, and study hybrid variants that combine trajectory-aware learned priors with selective physics-based mitigation when higher-fidelity calibration is needed.

C EXPERIMENT

C.1 EVALUATION BENCHMARK

We organize the experiments by problem class. Each experimental setting is specified by a Hamiltonian instance, an ansatz family, and an optimizer configuration.

TFIM. We first consider spin-system experiments based on the open-boundary transverse-field Ising model (TFIM), defined as

$$H_{\mathrm{TFIM}} = -J \sum_{i=1}^{n-1} Z_i Z_{i+1} - h \sum_{i=1}^{n} X_i. \tag{18}$$

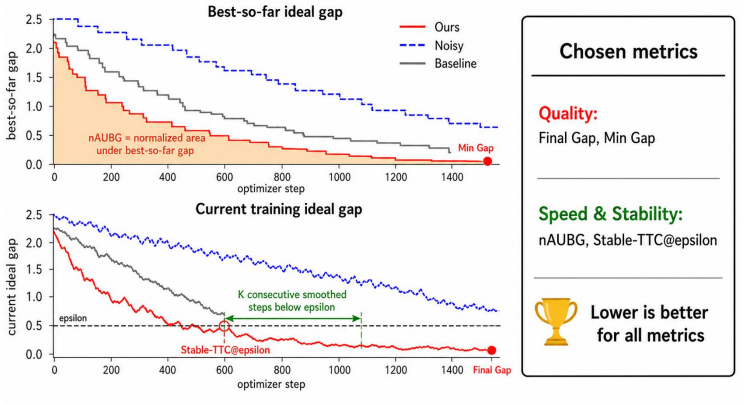


Figure 3: Schematic illustration of the selected VQE convergence metrics. nAUBG is computed on the best-so-far ideal-gap curve, while Stable-TTC is computed on the current training ideal-gap curve. Final Gap and Min Gap summarize end-of-run and best-reached solution quality.

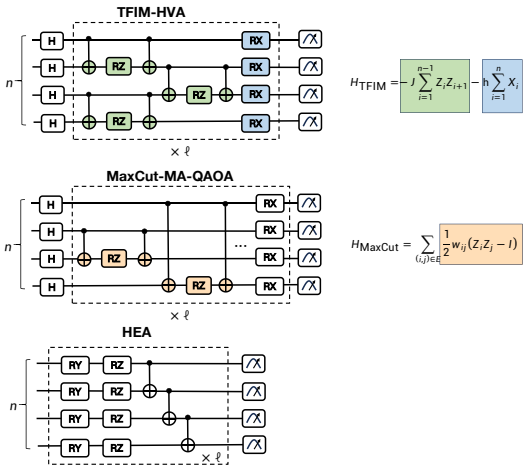


Figure 4: Illustration of Circuit Structure.

We consider two TFIM scales. The large-scale instance uses $n = 50$ qubits with $J = 0.1$ and $h = 0.5$. For this setting, we use a problem-inspired Hamiltonian variational ansatz (HVA),

$$U_{\mathrm{HVA}}(\theta) = \prod_{\ell=1}^{p} \exp\left(-i\beta_\ell \sum_{i=1}^{n} X_i\right) \exp\left(-i\gamma_\ell \sum_{i=1}^{n-1} Z_i Z_{i+1}\right), \tag{19}$$

with the packaged large-scale experiment using the depth-one instance `tfim_hva_d1`. The controlled small-scale TFIM instance uses $n = 4$ qubits with $J = 1.0$ and $h = 1.0$. On this instance, we evaluate hardware-efficient ansatzes (HEA), and use the corresponding HVA setting as the controlled cross-ansatz target. The HEA family is written abstractly as

$$U_{\mathrm{HEA}}(\theta) = \prod_{\ell=1}^{L} \left[U_{\mathrm{ent}}^{(\ell)} \prod_{q=1}^{n} U_{\mathrm{rot}}(\theta_{\ell,q}) \right], \tag{20}$$

where U_{rot} denotes single-qubit parameterized rotations and $U_{\mathrm{ent}}^{(\ell)}$ is a fixed entangling pattern. In the selected TFIM4 bundle, the packaged HEA instance is `hea_d2`; the matched HVA bundle is used to test whether the mitigation recipe transfers across ansatz families on the same Hamiltonian.

Combinatorial optimization. For combinatorial optimization, we study MaxCut on a weighted graph $G = (V, E, w)$. We use the cost Hamiltonian

$$H_{\text{MaxCut}} = \sum_{(i,j)\in E} \frac{1}{2} w_{ij}(Z_i Z_j - I), \tag{21}$$

so that minimizing $\langle H_{\text{MaxCut}} \rangle$ is equivalent to maximizing the cut value. The current selected instance is a 3-regular graph with $n = 8$ vertices, hence 8 qubits. We use a multi-angle QAOA ansatz with depth p,

$$U_{\text{MA-QAOA}}(\gamma, \beta) = \prod_{\ell=1}^{p} \left[\exp\left(-i \sum_{q\in V} \beta_{\ell,q} X_q \right) \exp\left(-i \sum_{(i,j)\in E} \gamma_{\ell,ij} \frac{1}{2} w_{ij}(Z_i Z_j - I) \right) \right], \tag{22}$$

initialized from $|+\rangle^{\otimes n}$. The selected MaxCut experiment uses `combopt_ma_qaoa_p4`, which gives separate angles for mixer terms and graph edges.

Molecular energy estimation. For molecular VQE, we consider electronic-structure Hamiltonians. In second quantization, the molecular Hamiltonian has the form

$$H_{\text{mol}} = \sum_{p,q} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s, \tag{23}$$

which is mapped to a qubit Hamiltonian

$$H_{\text{mol}}^{\text{qubit}} = \sum_{m=1}^{M} c_m P_m \tag{24}$$

by a fermion-to-qubit transformation. The selected chemistry setting uses an H_4 Hamiltonian at bond factor 1.3 and a hardware-efficient no-Hartree–Fock ansatz. Abstractly, the ansatz is

$$U_{\text{chem-HEA}}(\theta) = \prod_{\ell=1}^{L} \left[U_{\text{ent}}^{(\ell)} \prod_{q=1}^{n_{\text{chem}}} U_{\text{rot}}(\theta_{\ell,q}) \right], \tag{25}$$

applied to a computational-basis initial state rather than a Hartree–Fock initialized state. This setting tests whether the mitigation recipe extends beyond spin and graph-optimization Hamiltonians to molecular energy estimation.

The optimization protocol is summarized in Tab. 2. All runs use SPSA as the classical optimizer.

The pretraining dataset sizes for each setting is provided in Tab. 3. Experiments were conducted on a heterogeneous GPU cluster equipped with NVIDIA GeForce RTX 4090 and NVIDIA A100 GPUs. CPU hosts include Intel Xeon Gold 6530 and Intel Xeon CPU Max 9462 processors, with 512 GB–1 TB system memory.

C.2 Noise Configurations

Following [3; 44], we consider three primary noise configurations:

Fake Providers. To emulate the noise characteristics of real quantum devices, We utilize `FakeHanoiV2` and `FakeMelbourneV2`.

Incoherent Errors. Incoherent noise is simulated by applying Pauli-X and depolarizing errors to all gates, based on the Sycamore quantum device [2].

Coherent Errors. To further examine models' capabilities across different noise types, we introduce coherent noise through systematic over-rotation errors on two-qubit gates (CX, CY, CZ, and Swap). For each affected gate, the over-rotation angle is sampled uniformly from $[0.02\pi, 0.04\pi]$ and applied together with the noise model derived from the `FakeMelbourneV2` provider.

C.3 Baselines

We compare with four groups of in-loop VQE baselines. In all cases, the baseline must return an energy estimate at each optimizer query, so that the classical optimizer is driven by the mitigated objective rather than by a post-hoc correction of the final measurement trace.

Table 2: Nominal TrACE optimizer protocol for the selected benchmark settings. Historical baselines can use different method-specific evaluation horizons. All rows use SPSA. "# of Steps" denotes the number of SPSA iterations.

Benchmark	Noise setting	# of Steps	η_{SPSA}	c_{SPSA}	Last avg.
TFIM50-HVA	Incoh.	300	0.05	0.10	10
TFIM50-HVA	Mixed	150	0.05	0.10	10
TFIM50-HVA	Prov.	300	0.05	0.10	10
TFIM4-HEA	Incoh.	1500	0.05	0.08	5
TFIM4-HEA	Mixed	1500	0.05	0.08	5
TFIM4-HEA	Prov.	1500	0.05	0.08	5
TFIM4-HVA	Incoh.	500	0.010	0.05	1
TFIM4-HVA	Mixed	500	0.010	0.01	1
TFIM4-HVA	Prov.	500	0.005	0.01	1
H_4-HEA	Incoh.	1500	0.05	0.08	5
H_4-HEA	Mixed	1500	0.05	0.08	5
H_4-HEA	Prov.	1500	0.07	0.08	5
MaxCut-MA-QAOA	Incoh.	1500	0.02	0.06	5
MaxCut-MA-QAOA	Mixed	1500	0.01	0.04	5
MaxCut-MA-QAOA	Prov.	1500	0.01	0.04	5

Table 3: Offline pretraining dataset sizes for the selected benchmark settings.

	TFIM4 HVA–HEA	Chemistry H_4-HEA	TFIM50-HVA	MaxCut-MA-QAOA
Train	1600	800	800	800
Val.	400	200	100	200

Zero-noise extrapolation. We first include Zero-Noise Extrapolation (ZNE), a standard non-learning QEM method that estimates the zero-noise limit by evaluating the circuit at multiple amplified noise levels and extrapolating back to the noiseless expectation value [62; 43; 38]. Whenever ZNE is practical for the considered setting, we apply it at each queried parameter point and pass the ZNE-mitigated energy to the classical optimizer as the in-loop objective.

Practical ML-QEM baselines. Following the implementation of the VQE setting in [44], we implement three classical supervised regressors: a multilayer perceptron, a random forest, and ordinary least squares regression. The MLP follows the two-layer architecture used in [39; 44], while the random forest uses an ensemble of decision trees trained with bootstrap sampling and feature subsampling. OLS serves as a linear calibration baseline. For a fair in-loop comparison, all three models are trained on the same random-θ pretraining data as our method, use the same flat classical features, and directly regress from noisy observations to ideal-energy targets. During VQE, the trained regressor replaces the raw noisy energy by its predicted mitigated energy at every optimizer step.

Clifford data regression. We compare with Clifford Data Regression (CDR) [21; 48], which learns a noisy-to-ideal calibration map from classically simulable near-Clifford circuits and applies it to the target circuit. To make CDR compatible with the VQE loop, we use a periodic local variant: every fixed number of optimizer iterations, we construct a near-Clifford training bank around the current variational circuit, fit a linear or ridge calibration map, and reuse this map for subsequent optimizer queries until the next refresh. This yields an in-loop CDR objective while keeping the additional online cost controlled.

Active-learning GPR. We implement an active-learning Gaussian-process baseline following the tailored probabilistic mitigation framework of [36]. The model maintains a database of noisy VQE evaluations collected along the optimization trajectory and uses the Gaussian-process posterior mean as the objective value supplied to the optimizer. When the posterior uncertainty at a queried parameter is high, the baseline performs an additional noisy measurement and updates the database before returning the in-loop estimate.

C.4 MULTI-SEED REPORTING PROTOCOL

For all 15 TrACE settings, Tab. 4 reports the mean and standard deviation across three random seeds, including the five settings in the main-text Tab. 1. Each SD appears beneath its corresponding mean. Baselines retain their original individual-run values.

Within each setting, all four metrics are computed from the same three runs, using a fixed offline dataset.

Baseline comparisons may differ in method-specific configurations and horizons, so rankings are descriptive rather than matched statistical tests. Rankings use source precision where available and reported precision otherwise; display rounding can hide differences. Historical figures and supplementary ablations retain their own reported protocols.

C.5 EVALUATION METRICS

The goal of in-loop VQE mitigation is to guide the classical optimizer toward a better variational parameter vector, not merely to report a lower energy value. This distinction is important because a learned mitigator may output an artificially low objective without improving the parameter selected by VQE. We therefore evaluate the quality of a VQE trajectory by mapping each visited parameter vector back to the ideal Hamiltonian and measuring its gap to the reference energy.

For a VQE trajectory $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$ under ansatz instance s, we compute the ideal energy of each visited parameter vector as

$$E_t^{\mathrm{ideal}} = E^{\mathrm{ideal}}(\boldsymbol{\theta}_t; s) = \langle 0|U_s^{\dagger}(\boldsymbol{\theta}_t) H U_s(\boldsymbol{\theta}_t)|0\rangle, \tag{26}$$

where E_t^{ideal} is obtained from a noiseless statevector simulator, an MPS evaluator, or the best available reference evaluator for the corresponding problem setting. Let E_{ref} denote the exact or selected reference energy of the problem instance. Following the notation in Sec. 3.1, the ideal gap at step t is defined as

$$\delta_t = E_t^{\mathrm{ideal}} - E_{\mathrm{ref}}. \tag{27}$$

All reported metrics are computed from the ideal-gap trajectory $\{\delta_t\}_{t=1}^{T}$, rather than from the noisy or mitigated energies observed during optimization. Smaller values indicate better performance.

Final Gap. The final ideal gap is the ideal-energy gap of the final parameter vector returned by VQE:

$$\text{Final Gap} = \delta_T = E_T^{\mathrm{ideal}} - E_{\mathrm{ref}}. \tag{28}$$

This metric measures the final optimization outcome, i.e., whether the in-loop objective used by a method ultimately guides the optimizer to a high-quality parameter vector.

Min Gap. The minimum ideal gap is the smallest ideal-energy gap achieved by any visited parameter vector:

$$\text{Min Gap} = \min_{1\le t\le T} \delta_t. \tag{29}$$

This metric measures the best solution discovered during the trajectory, regardless of whether the final iterate remains at that point.

Normalized AUBG (nAUBG). The normalized area under the best-so-far gap curve measures how quickly a method discovers high-quality parameter vectors. Let

$$\delta_t^{\mathrm{bsf}} = \min_{1\le s\le t} \delta_s \tag{30}$$

denote the best-so-far ideal gap found up to step t. For each problem setting, let $\delta^{\star}_{\mathrm{ideal}}$ be the ideal-floor gap given by the best value reached by the ideal trajectory, and let $\delta^{0}_{\mathrm{noisy}}$ denote the initial noisy-VQE gap, computed as the median ideal gap over the first few noisy-VQE evaluations. We define the setting-specific normalization scale

$$\Delta_{\mathrm{norm}} = \delta^{0}_{\mathrm{noisy}} - \delta^{\star}_{\mathrm{ideal}}. \tag{31}$$

Let e_t denote the objective-evaluation count associated with the t-th recorded trace point. We compute nAUBG as the evaluation-count-weighted average of the best-so-far ideal gap after shifting by the ideal floor and normalizing by the noisy-to-ideal initial gap range:

$$\mathrm{nAUBG} = \frac{1}{Z}\sum_{t=1}^{T} \omega_t \left[\frac{\delta_t^{\mathrm{bsf}} - \delta^{\star}_{\mathrm{ideal}}}{\Delta_{\mathrm{norm}} + \varepsilon}\right]_{+}, \tag{32}$$

Table 4: Unified in-loop VQE comparison across all settings. All TrACE entries are means over three random seeds, with SD underneath (Appendix C.4); baselines are individual runs. Gap and nAUBG are lower-is-better. Stable-TTC ranks primary 20% convergence ahead of 30% fallback (†), then DNF ($W = 10$, $K = 10$); within a category, lower is better. Best and second-best non-Ideal values at source precision are coral red and navy blue; display rounding can hide differences. TFIM50-HVA uses 150 optimizer steps for Mixed and 300 for Incoh./Prov.; GPR is omitted due to computational cost.

Setting	Method	Min Gap ↓			Final Gap ↓			nAUBG ↓			Stable-TTC ↓		
		Incoh.	Mixed	Prov.	Incoh.	Mixed	Prov.	Incoh.	Mixed	Prov.	Incoh.	Mixed	Prov.
TFIM4-HEA	Ideal	0.0161	0.0161	0.0161	0.0161	0.0161	0.0161	0.086	0.084	0.087	561	556	561
	Noisy	0.3143	0.3362	0.2827	0.6349	0.6905	0.3511	0.304	0.322	0.294	DNF	DNF	DNF
	ZNE	0.0783	0.2714	0.0977	0.1718	1.0963	0.3476	0.239	0.379	0.191	1805	DNF	1498
	CDR	0.0999	0.3065	0.1197	0.4300	1.5747	0.4746	0.154	0.292	0.166	1073	DNF	638
	GPR	0.0648	0.1362	0.0673	0.0935	0.4625	0.1924	0.159	0.225	0.188	1110	2377	1505
	OLS	0.0842	0.2462	0.1417	0.2030	0.3836	0.1890	0.125	0.302	0.307	660	2791†	2969
	RF	1.1167	1.5450	1.0204	1.3640	2.1756	1.6560	1.237	1.446	1.103	DNF	DNF	DNF
	MLP	0.2926	0.4021	0.4659	0.8453	0.9908	0.9280	0.369	0.420	0.464	DNF	DNF	DNF
	TrACE	0.0641	0.1264	0.0608	0.0669	0.2740	0.0930	0.0984	0.1680	0.0893	375	808†	407
	± SD	±0.0050	±0.0051	±0.0046	±0.0069	±0.0200	±0.0076	±0.0056	±0.0037	±0.0011	±4	±96	±29
TFIM4-HVA	Ideal	0.0182	0.0182	0.0182	0.0182	0.0182	0.0182	0.033	0.009	0.029	48	35	46
	Noisy	0.0989	0.2762	0.1089	0.2481	2.1104	0.3023	0.157	0.110	0.207	4039†	584	DNF
	ZNE	0.0855	0.7648	0.1112	0.1956	1.8424	0.3067	0.158	0.273	0.163	532†	DNF	DNF
	CDR	0.3752	0.1841	0.4069	0.6014	4.7215	0.7166	0.528	0.099	0.546	DNF	133	DNF
	GPR	0.1245	0.7648	0.2940	0.3573	2.3322	0.5099	0.287	0.273	0.496	DNF	DNF	DNF
	OLS	0.0912	0.2447	0.1037	0.1846	0.4986	0.1169	0.180	0.209	0.157	323†	503	83†
	RF	0.0875	0.7648	0.1098	0.1847	3.9426	0.1327	0.168	0.273	0.149	119†	DNF	81†
	MLP	0.1049	0.5050	0.0934	0.2114	0.9903	0.1152	0.168	0.202	0.128	DNF	214†	78†
	TrACE	0.0827	0.1160	0.0888	0.1766	0.3418	0.5230	0.1218	0.0504	0.1026	594†	80	56
	± SD	±0.0090	±0.0090	±0.0116	±0.0135	±0.0204	±0.0143	±0.0318	±0.0041	±0.0130	±85	±9	±3
MaxCut-MA-QAOA	Ideal	0.0130	0.0130	0.0130	0.0130	0.0130	0.0130	0.170	0.174	0.171	783	803	785
	Noisy	0.2336	0.0933	0.9176	0.2484	0.1319	0.9777	0.338	0.227	0.381	1596	1186	1226†
	ZNE	0.3262	0.0798	0.9176	0.3334	0.0931	0.9777	0.368	0.157	0.381	1939	761	1226†
	CDR	0.1369	0.0892	0.3019	0.1369	0.1497	0.3594	0.195	0.203	0.255	952	888	1111
	GPR	2.5682	0.7713	1.2098	2.5910	0.7838	1.2266	0.744	0.397	0.554	DNF	1631†	DNF
	OLS	0.1353	0.0823	0.1895	0.1800	0.0939	0.4611	0.245	0.228	0.160	1238	1163	634
	RF	2.5820	2.7686	2.4456	2.6742	2.8407	2.5753	0.660	0.708	0.683	DNF	DNF	DNF
	MLP	2.3321	1.4967	2.5129	3.0424	2.5286	3.8938	0.649	0.631	0.671	DNF	DNF	DNF
	TrACE	0.0932	0.0710	0.0821	0.1106	0.0723	0.0937	0.1428	0.1222	0.1273	701	557	626
	± SD	±0.0054	±0.0068	±0.0072	±0.0090	±0.0071	±0.0013	±0.0123	±0.0079	±0.0129	±45	±57	±49
H_4-HEA	Ideal	0.0691	0.0691	0.0691	0.0691	0.0691	0.0691	0.093	0.093	0.093	539	539	542
	Noisy	0.1422	0.3139	0.2492	0.1422	0.3373	0.2492	0.275	0.311	0.211	1042	1366	926
	ZNE	0.1157	0.2664	0.2140	0.1157	0.2955	0.2140	0.206	0.307	0.162	899	1318	1017
	CDR	0.4037	0.6638	0.6504	0.4086	0.6638	0.6504	0.452	0.459	0.334	2348	1396	2267
	GPR	0.2095	0.4246	0.4296	0.2175	0.4705	0.4296	0.336	0.355	0.258	1282	1517	1254
	OLS	0.4831	0.8569	0.1418	0.4881	0.9551	0.1418	0.207	0.284	0.133	1002	722†	613
	RF	3.5891	3.4625	3.7389	3.8617	3.7478	3.9846	0.948	0.940	0.966	DNF	DNF	DNF
	MLP	1.1367	1.1957	0.9448	1.5622	1.4640	1.3578	0.295	0.310	0.283	684†	DNF	763†
	TrACE	0.1067	0.1898	0.0975	0.1067	0.2224	0.0975	0.0573	0.1803	0.0601	196	731	250
	± SD	±0.0019	±0.0071	±0.0017	±0.0019	±0.0056	±0.0018	±0.0039	±0.0132	±0.0055	±19	±87	±28
TFIM50-HVA	Ideal	0.0030	0.0041	0.0030	0.0030	0.0095	0.0030	1.063	2.221	1.063	127	127	127
	Noisy	0.0031	0.0670	0.0867	0.0044	0.1756	0.1454	0.683	0.525	0.850	536	DNF	DNF
	ZNE	0.0030	0.0047	0.0031	0.0030	0.0235	0.0031	0.584	0.392	0.397	461	188	348
	CDR	0.2394	0.2364	0.2392	0.2456	0.2421	0.2456	0.905	0.945	0.905	DNF	DNF	DNF
	OLS	0.0031	0.1842	0.0030	0.0031	0.3689	0.0032	0.474	0.801	0.300	379	DNF	260
	RF	0.2327	0.0048	0.0036	0.2448	0.0076	0.0065	0.904	0.426	0.360	DNF	203	318
	MLP	0.0053	0.2165	0.0445	0.3756	0.2872	0.4386	0.407	0.891	0.175	DNF	DNF	DNF
	TrACE	0.0031	0.0046	0.003003	0.0032	0.0181	0.003134	0.2467	0.3180	0.2012	210	160	190
	± SD	±0.0002	±0.0006	±0.000030	±0.0002	±0.0036	±0.000090	±0.0089	±0.0305	±0.0248	±12	±18	±21

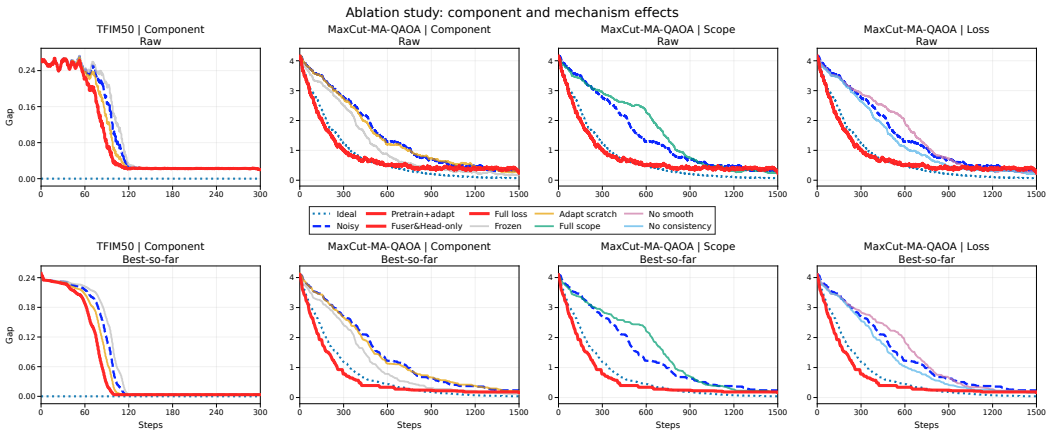


Figure 5: Model ablations of TrACE on TFIM50-provider and MaxCut-MA-QAOA-`Incoh.`, showing historical individual-run ideal-gap trajectories and best-so-far gaps for the full method, noisy control, frozen/adapt-scratch components, adaptation-scope variants, and loss-term removals.

where $[x]_+ = \max(x, 0)$, $\varepsilon > 0$ is a small numerical constant used only to avoid degeneracy when the noisy and ideal initial gaps coincide, ω_t is the objective-evaluation interval represented by the recorded point, and $Z = \sum_{t=1}^{T} \omega_t$.

This metric evaluates convergence speed in a scale-normalized way. A lower nAUBG indicates that the optimizer discovers low-gap parameter vectors earlier in the optimization budget.

Stable-TTC. Stable time-to-convergence measures when the current VQE iterate first enters a reliably low-gap region. For the ideal-gap trajectory $\{\delta_t\}$, define

$$\bar{\delta}_t = \frac{1}{|\mathcal{W}_t|} \sum_{s \in \mathcal{W}_t} \delta_s, \qquad \mathcal{W}_t = \{\max(1, t - W + 1), \ldots, t\}. \tag{33}$$

We define a target threshold by interpolating between the ideal floor and the initial noisy-VQE gap:

$$\epsilon_\rho = \delta^\star_{\text{ideal}} + \rho \left(\delta^0_{\text{noisy}} - \delta^\star_{\text{ideal}}\right). \tag{34}$$

Stable-TTC is reported as the first objective-evaluation count at the beginning of a run of K consecutive recorded trace points satisfying the threshold condition:

$$\text{Stable-TTC} = e_{\tau^\star}, \quad \tau^\star = \min\left\{t : \bar{\delta}_{t+j} \le \epsilon_\rho \text{ for all } j = 0, \ldots, K-1\right\}. \tag{35}$$

We use $W = 10$ and $K = 10$, with a primary threshold $\rho = 0.20$ and, when needed, a fallback $\rho = 0.30$ marked with a dagger. Runs that satisfy neither criterion are DNF. Stable-TTC ranks finite primary-threshold convergence ahead of any finite fallback, and finite fallback ahead of DNF; counts are compared only within the same category. A three-run entry is marked as fallback if any constituent run uses it; all three runs in the TFIM4-HEA/Mixed and TFIM4-HVA/Incoherent entries use fallback. DNF is never substituted by a finite count when computing SD. See Appendix C.4 for the aggregation protocol. Unlike Min Gap, which can be improved by a single transient visit to a good point, Stable-TTC measures when the live VQE iterate becomes stably usable.

C.6 Additional Experiment Results

Values for the four metrics of all 15 settings are listed in Tab. 4, with three-run variability for TrACE and individual-run baseline references. Fig. 6 displays the training trajectories in addition to Fig. 2. Fig. 5 provides the ablation results.

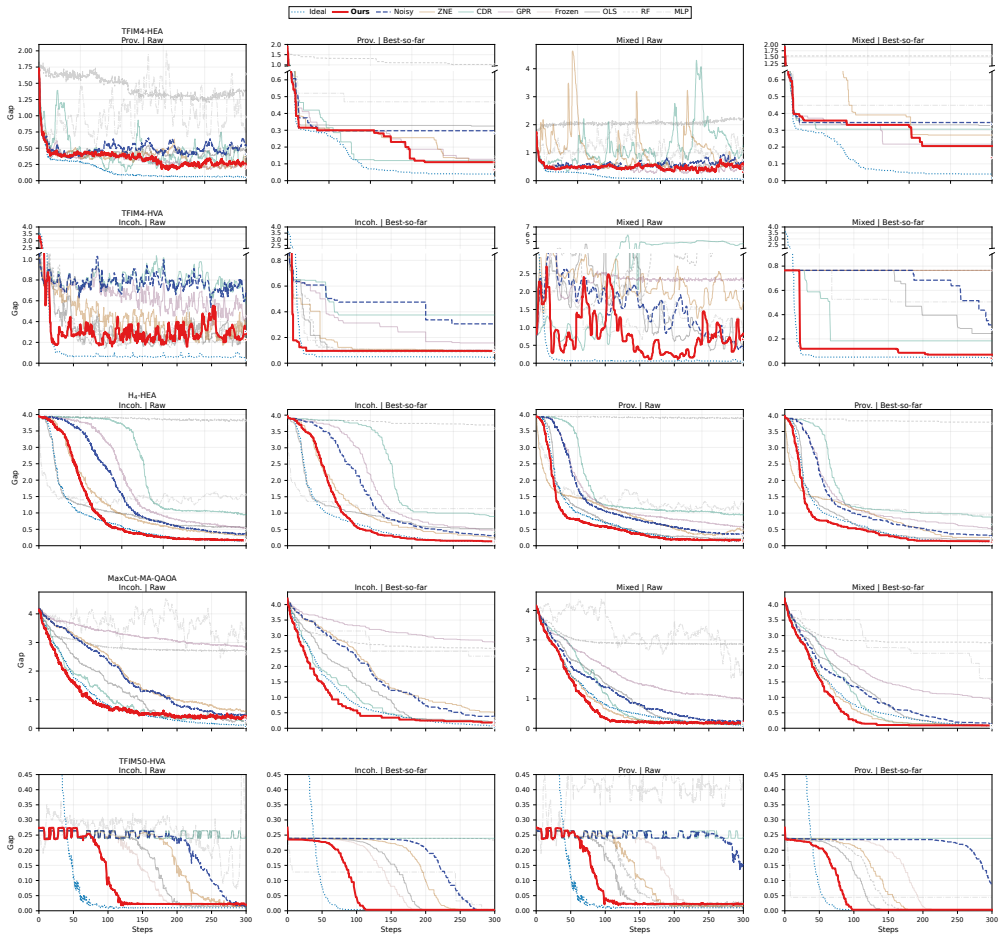


Figure 6: Additional small- and medium-scale VQE trajectories for the remaining packaged noise settings. Columns correspond to TFIM4-HEA, TFIM4-HVA, H_4-HEA, and MaxCut-MA-QAOA; each pair of rows shows the raw ideal gap and the cumulative best-so-far ideal gap for one non-main noise setting. The HVA provider panel is truncated to the first 150 optimizer steps for readability. Lower gap is better.

D Related Works

D.1 Variational Quantum Eigensolvers

The variational quantum eigensolver (VQE) is a canonical hybrid algorithm for near-term ground-state and Hamiltonian-optimization problems [53; 18; 63]. Given a parameterized circuit $U(\boldsymbol{\theta})$, VQE prepares $|\psi(\boldsymbol{\theta})\rangle$ and uses a classical optimizer to minimize the measured expectation $E(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$. This formulation has been widely used in molecular electronic-structure problems, spin and many-body Hamiltonians, and Hamiltonian encodings of combinatorial optimization, matching the chemistry, TFIM, and MaxCut settings studied in this work.

One line of VQE research improves the variational representation. Adaptive and problem-tailored constructions, including ADAPT-VQE and qubit-ADAPT-VQE, grow compact ansätze from operator pools rather than fixing the circuit a priori [30; 61]. Recent variants further target shallower circuits or reduced active-space requirements, for example through TETRIS-ADAPT-VQE and downfolded ADAPT-VQE [1; 58]. These methods improve the ansatz or Hamiltonian representation, whereas TrACE keeps the target Hamiltonian–ansatz setting fixed within each VQE run and improves the optimizer-facing objective used during noisy VQE.

Another line reduces the execution or measurement burden of VQE. Representative directions include joint measurement strategies, cascaded VQE formulations, and parallel circuit execution [14; 31; 17]. These methods aim to make energy evaluation more efficient. In contrast, TrACE addresses a complementary question: given noisy finite-shot measurements along the VQE loop, how can the objective values supplied to the classical optimizer be made more useful for selecting low-energy parameters?

A complementary body of work improves the optimizer side of VQE, including noise-aware derivative-free optimization, surrogate-assisted search, and learned warm starts or parameter transfer [42; 32; 69]. These methods change how variational parameters are initialized or proposed. Our focus is different: we keep the Hamiltonian, ansatz, and classical optimization protocol fixed within each mitigation comparison, and study how the energy values supplied to that optimizer can be mitigated in-loop under noisy finite-shot measurements. Thus, optimizer-side methods are compatible with our approach, but they address a different control axis from the mitigation baseline comparisons in this paper.

D.2 Quantum error mitigation

D.2.1 Non-machine-learning-based methods

Quantum error mitigation (QEM) aims to reduce noise-induced bias without full fault-tolerant encoding [12]. Physics-motivated approaches such as zero-noise extrapolation (ZNE) and probabilistic error cancellation (PEC) remain foundational baselines [62; 43; 26; 64]. However, their cost profile is poorly matched to in-loop VQE optimization: ZNE requires repeated noise-scaled executions of each queried circuit, while PEC relies on noise characterization and quasi-probability sampling whose overhead can grow rapidly with circuit size and noise strength [60; 55]. Since VQE repeatedly evaluates many neighboring parameterized circuits, applying these procedures at every optimizer step can make mitigation substantially more expensive than the underlying variational search.

Other non-ML techniques exploit problem or circuit structure, including symmetry verification and expansion [10; 11], verified phase estimation [51], and purification- or multi-copy-based methods such as virtual distillation and shadow distillation [40; 35; 57]. These methods can be effective when their structural assumptions are available, but they typically require additional circuit executions, copies, calibration, or problem-specific constraints. Clifford data regression (CDR), variable-noise CDR, and related low-capacity regression schemes provide a practical middle ground by fitting simple calibration maps from classically tractable circuits [21; 48; 22]. Yet they remain largely pointwise mitigation routines: their performance depends on the quality of the calibration circuit bank and they do not explicitly adapt to the trajectory-dependent distribution induced by VQE optimization.

D.2.2 Machine-learning-based methods

Machine-learning-based QEM amortizes part of the mitigation cost by training predictors from noisy and ideal data. Early neural and learning-based methods showed that classical models can mitigate

noisy output distributions, observables, and VQE energies without additional quantum executions at inference time [39; 59; 6; 68; 56]. Recent work further improves scalability and representation quality through practical ML-QEM, data-augmented mitigation, graph-based molecular mitigation, GTranQEM, QEM-Bench, and QEMFormer [44; 45; 52; 3; 4]. Despite these advances, most ML-QEM methods are trained offline and deployed as frozen pointwise predictors. In VQE, however, the optimizer induces a non-stationary and highly concentrated parameter distribution, so a frozen mitigator can become locally miscalibrated along the trajectory even if it performs well on the offline training distribution.

Several recent methods move closer to VQE-specific mitigation. Tailored probabilistic or GPR-style methods exploit ansatz-induced correlations across variational parameters, while in-loop deep mitigation and amortized ZNE variants seek to reduce the cost of repeated mitigation over parameterized circuit families [36; 13; 46]. These approaches highlight the importance of trajectory-aware mitigation, but they are often ansatz- or run-specific, rely on supervised targets generated near the current trajectory, or maintain posterior/calibration state that does not naturally transfer across circuit structures. In contrast, TRACE targets problem-scoped, cross-ansatz reusable mitigation: it learns a structured offline prior across an ansatz pool and performs lightweight self-supervised online adaptation inside the VQE loop, without ideal-energy labels during deployment. Appendix E.6 further evaluates shared pretraining across coefficient changes within a fixed Pauli representation.

E Additional Ablations and Sensitivity Analyses

The following experiments examine encoder contributions, adaptation frequency, loss weights, offline data requirements, transfer within a Hamiltonian family, larger and deeper circuits, and wall-clock cost. The sensitivity tables report individual experimental configurations rather than multi-seed uncertainty estimates; they should not be interpreted as significance tests. Multi-seed aggregates are reported separately in Tab. 4.

E.1 Encoder-branch ablation

We retrain the mitigator after removing each encoder route from the fused token sequence and rerun the in-loop protocol on H_4-HEA under mixed noise. Tab. 5 reports the resulting trajectory metrics. Removing any of the three routes increases the final gap and delays Stable-TTC in this comparison. The nAUBG effect is not uniform: removing the circuit route slightly reduces nAUBG despite worsening the endpoint. Thus these results indicate complementary endpoint contributions, rather than uniform dominance on every metric. Removing the measurement encoder refers to its learned token route; it does not establish a measurement-free method when measured quantities remain available to the output head.

Table 5: Encoder-route removal on H_4-HEA/Mixed. All metrics are lower-is-better.

Variant	Min Gap	Final Gap	nAUBG	Stable-TTC
Full TrACE	0.1898	0.2224	0.1803	731
w/o measurement encoder	0.3313	0.4091	0.2557	847
w/o parameter encoder	0.2426	0.3720	0.2019	693
w/o circuit encoder	0.2655	0.4051	0.1802	705

E.2 Adaptation interval and measurement overhead

On MaxCut-MA-QAOA/Incoherent, we vary the adaptation interval K while retaining the single-seed comparison protocol. Tab. 6 uses the nominal extra-measurement accounting of one additional noisy view per adaptation event, giving approximately $1/K$ overhead relative to the base VQE queries. These percentages describe that accounting convention, rather than measured wall-clock speedups or the offline pretraining cost. Every tested adapted configuration improves all four metrics over the noisy control. The dependence on K is non-monotonic: $K = 5$ gives the lowest Min Gap, $K = 10$ the lowest Final Gap, and $K = 50$ the lowest nAUBG and earliest Stable-TTC. The results support a practical trade-off between adaptation frequency and measurement cost rather than a universally optimal interval.

Table 6: Adaptation-interval sensitivity on MaxCut-MA-QAOA/Incoherent.

Variant	Extra shots	Min Gap	Final Gap	nAUBG	Stable-TTC
$K = 5$	20%	0.0659	0.1433	0.1166	620
$K = 10$ (default)	10%	0.0932	0.1106	0.1428	701
$K = 25$	4%	0.0819	0.1755	0.1173	607
$K = 50$	2%	0.0869	0.1648	0.1113	540
$K = 100$	1%	0.1269	0.1714	0.1287	627
$K = 500$	0.2%	0.1000	0.1446	0.1312	623
Frozen pretrained	0%	0.1206	0.1298	0.2683	1215
Noisy	0%	0.2336	0.2484	0.3385	1596

E.3 Adaptation losses and weight sensitivity

We study leave-one-out removal and one-at-a-time weight sweeps on MaxCut-MA-QAOA/Incoherent, around $(\lambda_{\text{cons}}, \lambda_{\text{stab}}, \lambda_{\text{anchor}}) = (1, 0.1, 0.02)$. Removing consistency produces the largest deterioration in Final Gap, from 0.1106 to 0.2614. Removing stability or anchoring also worsens the endpoint,

although removing anchoring slightly improves Stable-TTC. The weight sweeps in Tab. 8 further show that endpoint quality can be more sensitive than the integrated or convergence-time metrics. The reported Min Gap and Final Gap use the trajectory-minimum and returned-final parameter conventions of the experiment, respectively; final averaging can make the returned-final gap smaller than the minimum among logged query points.

Table 7: Adaptation-loss removal on MaxCut-MA-QAOA/Incoherent.

Variant	Min Gap	Final Gap	nAUBG	Stable-TTC
Full	0.0932	0.1106	0.1428	701
$\lambda_{\text{cons}} = 0$	0.2513	0.2614	0.1661	723
$\lambda_{\text{stab}} = 0$	0.1325	0.1796	0.1605	704
$\lambda_{\text{anchor}} = 0$	0.1687	0.1375	0.1617	684

Table 8: Observed metric ranges in one-at-a-time loss-weight sweeps. Consistency and stability use $\{0.1, 0.3, 0.5, 0.8, 1, 2\}$; anchoring uses $\{0.01, 0.02, 0.05, 0.08, 0.2\}$. Other weights remain at their defaults. Ranges are descriptive extrema, not confidence intervals.

Swept weight	Min Gap	Final Gap	nAUBG	Stable-TTC
λ_{cons}	0.0981–0.2778	0.0981–0.3157	0.1520–0.1665	669–748
λ_{stab}	0.0981–0.2387	0.0981–0.3147	0.1520–0.1654	684–804
λ_{anchor}	0.0981–0.2408	0.0981–0.2367	0.1520–0.1704	621–754

E.4 Pretraining dataset size

On MaxCut-MA-QAOA/Incoherent, we vary the number of pretraining samples from 200 to 1200, retrain the mitigator, and rerun the in-loop optimization. Tab. 9 shows that several hundred labels already improve on the noisy control. Min Gap varies only between 0.094 and 0.101 for $N_{\text{train}} \geq 800$, while Final Gap is non-monotonic and is lowest at the default 800-sample configuration. Larger sets improve Stable-TTC in this comparison, without uniformly improving the endpoint. This is a sensitivity study for the tested setting rather than a universal sample-complexity claim.

Table 9: Pretraining data-size sensitivity on MaxCut-MA-QAOA/Incoherent.

N_{train}	Min Gap	Final Gap	nAUBG	Stable-TTC
Noisy	0.234	0.248	0.338	1596
200	0.115	0.225	0.273	739
400	0.133	0.211	0.152	715
800 (default)	0.0932	0.1106	0.1428	701
1000	0.094	0.159	0.130	634
1200	0.101	0.139	0.134	598

E.5 Pretraining label fidelity

On TFIM4-HEA/Provider, we replace the exact labels in the training and validation sets with bond-truncated MPS labels ($\chi = 2$), ZNE-mitigated labels, or Gaussian-corrupted labels $y' = y_{\text{exact}} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Tab. 10 shows that approximate-label variants retain improvements over the noisy control but reach Stable-TTC later than the exact-label configuration. For the Gaussian variants, Min Gap and Final Gap increase with σ over the tested range, whereas nAUBG is not monotonic. MPS labels achieve a slightly smaller Final Gap than exact labels in this comparison, so the results do not imply that exact labels dominate every endpoint metric.

Table 10: Label-fidelity sensitivity on TFIM4-HEA/Provider.

Pretraining labels	Min Gap	Final Gap	nAUBG	Stable-TTC
Noisy control	0.283	0.351	0.294	DNF
Exact	0.0608	0.0930	0.0893	407
MPS ($\chi = 2$)	0.101	0.128	0.183	1678
ZNE	0.083	0.135	0.175	1670
Gaussian, $\sigma = 0.03$	0.064	0.137	0.165	1266
Gaussian, $\sigma = 0.06$	0.075	0.146	0.186	1704
Gaussian, $\sigma = 0.125$	0.077	0.163	0.210	2011
Gaussian, $\sigma = 0.25$	0.112	0.183	0.199	2134

E.6 Transfer within a Hamiltonian family

We investigate transfer within $H(h) = \sum_{P \in \mathcal{S}} c_P(h) P$. Only the coefficients $c_P(h)$ vary: the Pauli-string set $\mathcal{S}$ and its ordering, ansatz architecture, and parameter indexing remain fixed. Each term-wise measurement position therefore refers to the same Pauli operator, and each parameter position retains its circuit meaning across family members. For TFIM4-HVA/Mixed, a shared mitigator is pretrained on $h \in \{0.5, 0.8, 1.2, 1.5\}$ using 350 training and 50 validation samples per value (1400/200 in total). No samples at the target $h = 1.0$ are included in either split. The model is deployed at $h = 1.0$ with label-free online adaptation. Tab. 11 shows a smaller Final Gap but later Stable-TTC than the target-specific model. This comparison supports coefficient transfer within the aligned TFIM family. Transfer to a different Pauli support, such as from TFIM to MaxCut, changes the meaning of the input positions and requires a new representation alignment and validation; it is not tested here.

Table 11: Transfer to unseen $h = 1.0$ on TFIM4-HVA/Mixed. These are individual configuration results; † denotes the historical 30% fallback threshold.

Method	Min Gap	Final Gap	nAUBG	Stable-TTC
Noisy	0.2762	2.1104	0.110	584
ZNE	0.7648	1.8424	0.273	DNF
CDR	0.1841	4.7215	0.099	133
GPR	0.7648	2.3322	0.273	DNF
OLS	0.2447	0.4986	0.209	503
RF	0.7648	3.9426	0.273	DNF
MLP	0.5050	0.9903	0.202	$214^{\dagger}$
Target-specific TrACE	0.1160	0.3418	0.0504	80
Cross-Hamiltonian TrACE	0.0522	0.3389	0.038	331

E.7 SCALING TO 100 QUBITS

We extend the evaluation to TFIM100-HVA under Mixed noise. TrACE and the frozen ML-QEM baselines (MLP, OLS, and RF) use 800 training and 100 validation samples, with ideal labels computed using MPS. Table 12 reports the four trajectory metrics; Ideal is a reference and is excluded from the mitigation-method ranking. Active GPR did not complete within the evaluation period and is omitted. These supplementary results are reported without multi-seed SDs. Among non-Ideal methods, TrACE has the best Final Gap, nAUBG, and Stable-TTC, and the second-best Min Gap behind OLS. Its Final Gap is approximately $8.9\times$ smaller than raw noisy VQE ($0.2427/0.0273$). Raw VQE, MLP, CDR, and ZNE are DNF under the reported convergence criterion. Fig. 7 illustrates the optimization dynamics over 100 optimizer steps in an additional individual-run comparison of Ideal, Noisy, and TrACE. These trajectories are separate from the runs summarized in Tab. 12.

Table 12: Supplementary TFIM100-HVA/Mixed comparison. Ideal is a reference; all four metrics are lower-is-better, with DNF denoting failure to reach stable convergence.

Method	Min Gap	Final Gap	nAUBG	Stable-TTC
Ideal	0.0062	0.0062	0.0176	36
Raw VQE	0.1542	0.2427	0.0916	DNF
MLP	0.0157	0.4121	0.0825	DNF
OLS	0.0072	0.1023	0.0716	110
RF	0.0100	0.0351	0.0889	282
CDR	0.0167	0.2885	0.0526	DNF
ZNE	0.0332	0.1295	0.1315	DNF
TrACE	0.0081	0.0273	0.0254	76

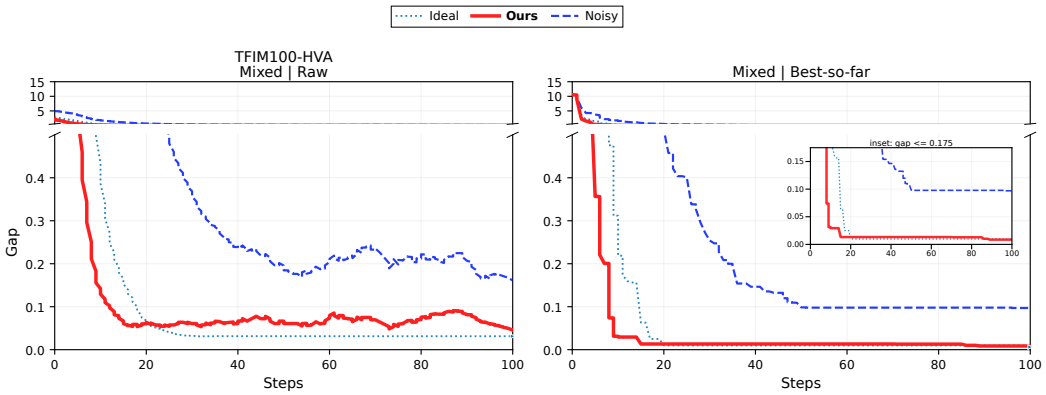


Figure 7: Additional individual-run TFIM100-HVA trajectories under Mixed noise. The left panel shows the current ideal gap (Raw), and the right panel shows the cumulative best-so-far ideal gap. All methods are shown over the same first 100 optimizer steps.

E.8 DEEP-CIRCUIT FERMI–HUBBARD BENCHMARK

To probe circuit depth, we evaluate a four-site Fermi–Hubbard system (eight qubits) using a parameter-tied Hamiltonian variational ansatz with $p = 8$, 1,302 native gates, and circuit depth 258 under Incoherent noise. Table 13 reports the supplementary comparison without multi-seed SDs. TrACE ranks first among non-Ideal methods on all four metrics and reduces Final Gap from 0.5976 to 0.1556, approximately $3.8\times$. Its Stable-TTC of 88 is earlier than the Ideal reference's 105 in this experiment; this does not imply a general advantage over noiseless optimization.

Table 13: Eight-qubit Fermi–Hubbard comparison at circuit depth 258 under Incoherent noise. Best gap in the source experiment is reported as Min Gap; † marks the fallback convergence threshold.

Method	Min Gap	Final Gap	nAUBG	Stable-TTC
Ideal	0.0391	0.0437	0.0857	105
Raw VQE	0.2155	0.5976	0.3372	DNF
MLP	0.1194	0.4311	0.2420	521
OLS	0.1014	0.3299	0.2439	608
RF	0.1043	0.5780	0.2694	501
CDR	0.1947	0.2714	0.3487	DNF
ZNE	0.1377	0.4927	0.3224	$849^{\dagger}$
GPR	0.0904	0.1684	0.1628	435
TrACE	0.0875	0.1556	0.1188	88

E.9 WALL-CLOCK COMPARISON

We measure TFIM4-HVA/Provider using 4,000 shots per qubit-wise commuting group and 500 SPSA iterations on NVIDIA RTX 5090 GPUs, with the same software environment and resource allocation across methods. Table 14 separates offline data generation and pretraining from online execution; online entries give the median and observed minimum–maximum range, not SDs. Data generation is a one-time shared cost for TrACE and the frozen ML-QEM baselines. The total offline cost of TrACE is $315.07 + 39.57 = 354.64$ seconds. Its median online time is 367.03 seconds, or $1.180\times$ raw VQE, and is 58.1%, 28.8%, and 47.5% lower than ZNE, CDR, and active GPR, respectively. The frozen ML-QEM medians slightly below raw VQE lie within the observed timing variability and do not establish speedups. These timings are hardware-dependent. With one additional noisy view every tenth query, the nominal online measurement multiplier is $1 + 1/10 = 1.1$, distinct from the measured wall-clock ratio.

Table 14: Wall-clock costs on TFIM4-HVA/Provider. Times are in seconds. The final column divides each median online time by the raw-VQE median.

Method	Data gen.	Pretrain	Online median [min–max]	Per query	× Raw
Raw VQE	0	0	311.13 [292.98–329.11]	0.311	1.000
ZNE	0	0	875.39 [857.09–941.95]	0.875	2.814
CDR	0	0	515.60 [495.81–576.47]	0.516	1.657
Active GPR	0	0	699.28 [497.45–1160.41]	0.699	2.248
ML-QEM (OLS)	315.07	0.83	302.56 [299.06–334.06]	0.303	0.972
ML-QEM (RF)	315.07	3.05	302.56 [298.48–303.75]	0.303	0.972
ML-QEM (MLP)	315.07	1.08	296.15 [295.16–315.54]	0.296	0.952
TrACE	315.07	39.57	367.03 [364.69–386.37]	0.367	1.180