

GSC-QEMit: A Telemetry-Driven Hierarchical Forecast-and-Bandit Framework for Adaptive Quantum Error Mitigation

Steven Szachara*, Sheeraja Rajakrishnan*, Dylan Jay Van Allen[†], Jason Pollack[†], Travis Desell*, Daniel Krutz*

*Department of Software Engineering, Rochester Institute of Technology, Rochester, USA

Emails: ss9270@rit.edu, sr8685@rit.edu, tjdvse@rit.edu, dxkvse@rit.edu

[†]Institute for Quantum & Information Sciences, Syracuse University, Syracuse, USA

Emails: djvanall@syr.edu, japollac@syr.edu

Abstract—Quantum error mitigation (QEM) is essential for extracting reliable results from near-term quantum devices, yet practical deployments must balance mitigation strength against runtime overhead under time-varying noise. We introduce *GSC-QEMit*, a telemetry-driven, context-forecast-bandit framework for *adaptive* mitigation that switches between lightweight suppression and heavier intervention as drift evolves. GSC-QEMit composes three coupled modules: (G) a Growing Hierarchical Self-Organizing Map (GHSOM) that clusters streaming telemetry into operating contexts; (S) an uncertainty-aware subsampled Gaussian-process forecaster that predicts short-horizon fidelity degradation; and (C) a cost-aware contextual multi-armed bandit (CMAB) that selects mitigation actions via Thompson sampling with explicit intervention cost. We evaluate GSC-QEMit on benchmark circuit families (GHZ, Quantum Fourier Transform, and Grover search) under nonstationary noise regimes simulated in Qiskit Aer, using an instrumented testbed where action labels correspond to graded mitigation intensity. Across Clifford, non-Clifford, and structured workloads, GSC-QEMit improves average logical fidelity by +9.0% relative to unmitigated execution while reducing unnecessary heavy interventions by reserving them for inferred noise spikes. The resulting policies exhibit a favorable fidelity–cost trade-off and transfer across the evaluated workloads without circuit-specific tuning.

I. INTRODUCTION

In the noisy intermediate-scale quantum (NISQ) regime, a central practical challenge is extracting reliable results from devices whose error processes drift over time and vary with operating context. Quantum error mitigation (QEM) addresses this challenge by reducing noise-induced bias at the *ensemble* level – through repeated executions, classical post-processing, and lightweight execution-time adjustments, without requiring full fault-tolerant overhead [1]–[4].

Most prior QEM work improves a specific mitigation primitive or its parameterization, such as zero-noise extrapolation, probabilistic error cancellation, and readout mitigation [4]–[6], or develops learned surrogate models that emulate a single mitigation workflow [7], [8]. Recent platform and learning-based advances have further improved the cost–accuracy trade-off of mitigation and, in some cases, enabled adaptive noise estimation under drift [9], [10]; however, these approaches generally operate within a fixed mitigation strategy and do not address the systems problem we study here: runtime

orchestration over a library of intervention options under nonstationary noise.

ML/AI efforts have accelerated progress in quantum reliability, including learned policies for feedback and control, models that map noisy signatures to decision rules, and hybrid methods supporting calibration, compilation, and execution-time tuning [11]–[17]. Yet many approaches are trained and deployed as effectively *static* controllers: optimized for a fixed device configuration, circuit family, or assumed noise process. In practice, static mitigation pays a constant computational “tax” even when noise is benign, while under-reacting when noise spikes, a mismatch that is especially costly when the appropriate response ranges from lightweight suppression (*e.g.*, dynamical decoupling, Pauli twirling) to heavier intervention (*e.g.*, code-based decoding) [16], [18], [19].

This motivates an adaptive, cost-aware view of quantum reliability: rather than committing to a single “best” mitigation method, a runtime system should *monitor telemetry* and *choose* when to intervene and how aggressively, explicitly trading fidelity improvement against intervention cost under uncertainty. We therefore formulate QEM orchestration as a closed-loop decision problem driven by streaming telemetry, where each cycle yields an updated state summary and an immediate post-action outcome. This setting naturally admits *contextual bandit* decision-making: actions are discrete, rewards are observed on a short horizon, and fast online adaptation is required without the instability and data demands of long-horizon reinforcement learning.

GSC-QEMit is differentiated by operating as a backend-agnostic policy layer above concrete mitigation primitives. Rather than learning a single controller tied to a specific code, circuit family, or mitigation method, it combines hierarchical telemetry context discovery, uncertainty-aware short-horizon forecasting, and cost-aware contextual bandit selection to decide *when* to intervene and *how aggressively* to intervene as operating conditions evolve.

To summarize, this work makes the following contributions:

- **Telemetry-driven orchestration layer:** We introduce GSC-QEMit, a modular runtime framework that orchestrates mitigation intensity from streaming telemetry via

hierarchical context discovery, uncertainty-aware forecasting, and a cost-aware bandit policy.

- **Context-forecast factorization:** We formalize a separation between (i) context discovery via Spark-GHSOM over streaming telemetry and (ii) short-horizon degradation forecasting with an SVGP that provides calibrated predictive uncertainty. This factorization enables nonstationary, context-dependent decision-making without end-to-end retraining.
- **Cost-aware contextual bandit control:** We cast mitigation selection as contextual decision-making and implement Thompson sampling with per-action Bayesian linear models (TS-CL), bootstrapped with expert demonstrations to encode fidelity-cost preferences.
- **Cross-benchmark evaluation under drift:** We validate the framework on GHZ, QFT, and Grover workloads under drifting noise in an instrumented simulator, demonstrating consistent fidelity gains and selective use of heavier interventions compared to static baselines.

The GSC-QEMit code repo and data is available upon request.

II. RELATED WORK

This paper sits at the intersection of (i) quantum error mitigation (QEM) for near-term devices, (ii) learning-based quantum control and decoding, and (iii) scalable telemetry modeling for nonstationary systems. Our core distinction is architectural: rather than proposing a new mitigation primitive or a single end-to-end controller, we study *mitigation orchestration*—how to select *when* to intervene and *which* intervention intensity to apply from streaming telemetry under explicit cost constraints.

A. QEM Primitives and Overhead Trade-offs

Canonical QEM protocols reduce noise-induced bias through repeated circuit executions and classical post-processing, including zero-noise extrapolation (ZNE) and probabilistic error cancellation (PEC) [4], [6]. These methods highlight a fundamental bias-variance-cost trade-off: bias reduction typically requires increased sampling overhead and/or stronger assumptions about the noise channel, which can become prohibitive as fault rates grow or as workloads scale [4]. This motivates approaches that adapt mitigation strength to runtime conditions, particularly under drift where the “right” level of mitigation may change over time [20].

B. Compiler-Integrated Detection and Intermediate Mitigation

Between purely post-processing-based QEM and full fault tolerance, quantum error detection (QED) provides a pragmatic intermediate strategy: errors are detected and runs are postselected rather than corrected, often reducing overhead relative to full correction while improving reliability. Systems work has emphasized that deploying QED at scale requires careful compilation and hardware mapping, including ancilla management and integration of detection structures into programs. QuantEM, for example, automates insertion

of QED subcircuits and supports a library of detection codes as a scalable intermediate mitigation approach [21]. Whereas such efforts focus on *compile-time* integration of detection structures, our focus is *runtime adaptation*: given a library of intervention options, decide online which to enable based on streaming telemetry and forecasted degradation.

C. Learning-Based Feedback Control Under Drift

A key driver of adaptive mitigation is nonstationarity: device performance can drift due to calibration changes, environmental variation, or slow parameter drift [22]. Prior work connects error signals (e.g., detection events or syndrome statistics) to feedback and control policies, including learned strategies that stabilize logical behavior in the presence of injected or observed drift [11], [23], [24]. Reinforcement learning and related adaptive controllers further demonstrate that streaming measurements can support closed-loop control on simulated or hardware-proxied systems [11], [12]. GSC-QEMit aligns with this telemetry-informed view, but operates at a higher policy layer: rather than directly optimizing pulse-level parameters, we select among discrete mitigation/correction *intensities* using learned contexts and calibrated forecasts under explicit cost constraints.

D. Neural Decoders and RL for QEC

Substantial literature treats decoding and correction as learning problems [12], [13], [15], [25]. Supervised neural decoders map syndrome patterns to corrections with low-latency inference suitable for real-time pipelines [13], [14], while RL-based agents can learn feedback strategies from interaction with simulated devices and streaming syndrome signals [11], [12]. These approaches often yield high performance within a fixed code/noise setting, but they typically produce a *single* policy tied to a particular code family and training distribution. In contrast, our emphasis is on *orchestration*: switching between intervention options (from lighter suppression to heavier intervention) as a function of context, predicted degradation, and action cost.

E. Hierarchical Context Discovery and Scalable Telemetry Modeling

GSC-QEMit draws from unsupervised structure discovery for high-volume telemetry. Growing Hierarchical Self-Organizing Maps (GHSOM) learn coarse-to-fine partitions via data-driven growth criteria, producing interpretable hierarchical regimes rather than a single flat clustering [26]. Spark-GHSOM extends this paradigm to distributed computation, making it a natural fit for streaming telemetry pipelines [26], [27]. While hierarchical SOM-style models have been applied broadly to regime discovery, they have rarely been integrated with online, uncertainty-aware decision layers for quantum reliability. Our work couples Spark-GHSOM contexts to (i) a Sparse Variational Gaussian Process (SVGP) forecaster and (ii) a contextual bandit policy, enabling interpretable state descriptors, calibrated near-horizon predictions, and cost-aware adaptation under drift.

F. Summary

Across these threads, prior work has produced strong mitigation primitives (ZNE, PEC, QED), learned decoders, and feedback strategies for nonstationary noise [6], [28]. GSC-QEMit targets a complementary systems problem: given streaming telemetry, learn hierarchical operating contexts, forecast near-horizon degradation with uncertainty, and select among intervention options using a cost-aware contextual bandit.

III. METHODOLOGY

We formulate GSC-QEMit as an adaptive quantum error mitigation problem: given a stream of cycle-level telemetry, the controller must decide when to intervene and how aggressively to intervene under uncertainty and explicit intervention cost.

TABLE I
EMULATOR TELEMETRY FIELDS AND PROVENANCE (PER CYCLE).

Emulator telemetry (per cycle)			
Field	Source	Derived from	Description
cycle	Internal	Config	Control-cycle index $t \in \{0, \dots, T_{\text{run}} - 1\}$.
code_dist	Internal	Config	Repetition distance $d \in \{3, 5\}$.
p_phys	Simulator	Noise Model	Physical noise parameter (drift knob); used to form p_{eff} in Table II.
eps_logical	Decoder	Counts	Logical error ϵ_L (TVD to ideal).
fidelity	Derived	$1 - \epsilon_L$	Logical fidelity F_L .
features	Derived	Current + history	Feature vector $\mathbf{x}_t \in \mathbb{R}^{13}$ (Table II).

Rather than training a single monolithic decoder or end-to-end deep RL agent, we factor the system into three lightweight components that are trained independently and composed into a closed-loop runtime policy: (i) **Context Discovery**: Unsupervised clustering of 13-dimensional telemetry features using Spark-GHSOM to partition execution into hierarchical operating regimes; (ii) **Uncertainty Forecasting**: Short-horizon prediction of logical fidelity using a Sparse Variational Gaussian Process (SVGP), which provides both a predictive mean and calibrated uncertainty to guide risk-sensitive decisions; [29], [30] and (iii) **Cost-Aware Decision Making**: An online linear contextual bandit policy that selects among intervention labels using Thompson Sampling with per-action Bayesian linear reward models (TS-CL). Crucially, we bootstrap the bandit via imitation learning to embed expert heuristics (e.g., “avoid heavy correction when noise is low”) before online deployment [31], [32].

a) *Mitigation-controller terminology*.: We define three controller behaviors used in our evaluation: (1) Unmitigated (Identity): Applies no error mitigation, incurring zero cost but suffering full noise impact. (2) Fixed/Static Mitigation: Applies a constant strategy (e.g., Surface Code) every cycle, maximizing fidelity but maximizing cost. (3) Adaptive Mitigation (GSC-QEMit): Dynamically selects from an intervention library $\mathcal{A} = \{\text{Identity, Pauli Suppression, Surface Code}\}$ based on the forecasted regime. In our emulator, these actions

TABLE II
TELEMETRY FEATURE VECTOR $\mathbf{x}_t \in \mathbb{R}^{13}$ USED FOR CONTEXT DISCOVERY AND POLICY CONDITIONING. HERE $t \in \{0, \dots, T_{\text{run}} - 1\}$ INDEXES THE CONTROL CYCLE WITHIN A RUN OF LENGTH T_{run} . ALL FEATURES ARE NORMALIZED TO COMPARABLE SCALES WHERE INDICATED.

Idx	Definition
1	$t/(T_{\text{run}} - 1)$
2	$n/10$
3	$\text{depth}/100$
4	p_{eff}
5	ϵ_L (TVD to ideal)
6	$F_L = 1 - \epsilon_L$
7	$H(\hat{p})/10$
8	$N_T/50$ (T-gate count)
9	$N_{2q}/100$ (two-qubit gate count)
10	$\log_{10}(\max(\epsilon_L, 10^{-9}))$
11	$\log_{10}(\max(p_{\text{eff}}, 10^{-9}))$
12	recent-window mean of ϵ_L
13	recent-window variance of ϵ_L

are instantiated as logical-level interventions that modify the effective noise experienced by the circuit.

A. Telemetry in Our Emulator Experiments

Our experiments use an instrumented logical-memory emulator that emits cycle-level telemetry. Each cycle executes a benchmark circuit (e.g., GHZ, QFT) for N_{shots} shots. From these counts, we compute logical-memory diagnostics (detection rate, logical error rate) and assemble a fixed 13-dimensional feature vector used by all components.

a) *Telemetry feature vector (13-D)*.: At each control cycle $t \in \{0, \dots, T_{\text{run}} - 1\}$, the backend emits a 13-dimensional telemetry vector $\mathbf{x}_t$ (Table II) used for both hierarchical context discovery and policy conditioning. Here T_{run} denotes the total number of control cycles in a run (i.e., the experiment horizon), and time is normalized as $t/(T_{\text{run}} - 1)$. The feature set combines execution progress, circuit size and depth, gate-count statistics, and logical-performance indicators, with all quantities scaled to comparable ranges to stabilize unsupervised clustering and linear policy learning.

Importantly, in our simulator-based benchmark we expose p_{eff} as a controlled drift knob to enable paired counterfactual evaluation. On physical hardware, simulator-specific quantities such as p_{eff} can be replaced by *observable proxies* (e.g., calibration-reported error rates, detection or syndrome event rates, readout calibration drift, or short-window statistics of recent logical errors), while preserving the same telemetry interface to the context and bandit layers.

B. Applicability to Real Hardware

On real hardware, simulator-internal fields such as p_{phys} are not directly observable. In our emulator, p_{phys} is a controlled drift parameter used to generate noise, and we define p_{eff} as the scalar *noise/health proxy* consumed by the learning stack (Table II). In a physical deployment, p_{eff} is obtained from *observable proxies* computed from periodic calibration (e.g., gate error rates, readout error, and coherence metrics T_1/T_2) and/or online indicators (e.g., syndrome/detection-event rates, leakage rates, or short-window statistics of recent logical errors). Note that T_1 and T_2 here denote standard coherence

noise amplification for ZNE, detection/postselection, or code-based decoding), while the context–forecast–policy stack remains unchanged.

a) *Benchmark instantiation.*: In the simulator-backed benchmark used here, we instantiate these abstract levels in two complementary ways. First, **MODERATE** is realized as a structural redundancy-and-decode intervention by replicating circuit sampling and applying majority-vote decoding of measured bits to form a decoded logical distribution. Second, **SEVERE** is realized as a stronger intervention intensity via action-conditioned scaling of the effective noise parameters (and readout-flip reduction) in the Aer noise model. These choices provide a controlled test of adaptive switching under drift while preserving a general, backend-agnostic action interface.

2) *Instrumented benchmark backend with drifting noise.*: The backend executes a suite of measured benchmark circuits under a time-varying noise process simulated via Qiskit Aer. At each control cycle $t \in \{0, \dots, T_{\text{run}} - 1\}$, we generate a baseline physical error rate via a bounded sinusoidal drift around a nominal error rate p_{phys0} (the baseline physical error rate at the start of a run):

$$p_{\text{base}}(t) = \text{clip}\left(p_{\text{phys0}}[1 + \alpha \sin(2\pi t/T_{\text{run}}) + \xi_t], 10^{-4}, 0.05\right), \quad (1)$$

where α controls the drift amplitude, $\xi_t \sim \mathcal{N}(0, \sigma^2)$ models stochastic fluctuations, and $\text{clip}(\cdot)$ enforces physically reasonable bounds.

An intervention-dependent scale factor $s(a)$ is then applied to obtain the effective physical error probability:

$$p_{\text{eff}}(t, a) = \text{clip}(p_{\text{base}}(t) \cdot s(a), 10^{-5}, 0.5), \quad (2)$$

where $a \in \mathcal{A} = \{\text{NONE}, \text{MODERATE}, \text{SEVERE}\}$ denotes the selected intervention level. Stronger intervention levels correspond to smaller $s(a)$ (greater error suppression) but incur higher control cost $C(a)$ in the decision layer.

Importantly, in this simulator-based benchmark we expose p_{eff} as a controlled drift knob for repeatable evaluation. On physical hardware, simulator-specific quantities such as p_{eff} can be replaced by *observable proxies* (e.g., calibration-reported error rates, detection/syndrome event rates, readout drift indicators, or short-window statistics of recent errors) without changing the controller interface.

3) *Telemetry logging schema.*: Following the QDataSet design philosophy [33], we store each control cycle as a JSON object containing inputs, metadata, and derived diagnostics. Key fields include:

- "cycle": logical round index t .
- "p_eff": effective physical error proxy $p_{\text{eff}}(t, a)$ (simulator-controlled in this benchmark).
- "eps_logical": logical error ϵ_L computed as the TVD between the observed and ideal output distributions.
- "fidelity_logical": logical fidelity $F_L = 1 - \epsilon_L$.
- "features": a 13-D feature vector $\mathbf{f}_t$ used for context mapping (Table II).

- "action_level": selected intervention level in $\{\text{NONE}, \text{MODERATE}, \text{SEVERE}\}$.

C. Spark–GHSOM Context Mapping

We train a Growing Hierarchical Self-Organizing Map (GHSOM) on the 13-D features using Apache Spark. The objective is to discretize streaming telemetry into hierarchical operating regimes (contexts) that remain informative under drift. Training yields a context index $\mathcal{C} : \mathbb{R}^{13} \rightarrow \mathbb{Z}$ that maps each normalized feature vector $\tilde{\mathbf{f}}_t$ to a unique discrete context identifier c_t used at runtime.

D. Sparse Variational GP (SVGP) Forecaster

To predict near-horizon degradation with calibrated uncertainty, we employ a Sparse Variational Gaussian Process (SVGP). Unlike standard GPs, which scale cubically in the number of training points ($O(N^3)$), SVGP uses a set of inducing points to form a variational approximation to the posterior, enabling efficient training and fast runtime querying.

At each cycle t , the forecaster consumes a compact, interpretable state vector

$$\mathbf{x}_t = [c_t, p_{\text{eff},t}, d, t/T_{\text{run}}, F_{L,t}],$$

where c_t is the discrete context label output by the GHSOM, $p_{\text{eff},t}$ is the current scalar noise/health proxy, d is the code distance of the executed circuit, t/T_{run} is normalized execution time, and $F_{L,t}$ is the observed logical fidelity. Conditioned on $\mathbf{x}_t$, the SVGP returns a Gaussian predictive distribution over the near-horizon fidelity:

$$F_{L,t+\Delta} \sim \mathcal{N}(\hat{\mu}_{t+\Delta}, \hat{\sigma}_{t+\Delta}^2), \quad (3)$$

where $(\hat{\mu}_{t+\Delta}, \hat{\sigma}_{t+\Delta})$ are the posterior mean and standard deviation.

Rather than sampling from this distribution, GSC-QEMit treats these moments as summary statistics of predicted system health and appends them deterministically to the bandit context vector,

$$\mathbf{z}_t = [\mathbf{x}_t, \hat{\mu}_{t+\Delta}, \hat{\sigma}_{t+\Delta}].$$

The contextual bandit then applies Thompson sampling over its own reward model using $\mathbf{z}_t$, so that action selection depends on both expected future fidelity and forecast uncertainty, enabling a performance–risk trade-off when choosing mitigation intensity.

E. Cost-Aware Contextual Bandit (TS-CL)

Adaptive mitigation is implemented as a Contextual Multi-Armed Bandit (CMAB) using Thompson Sampling with Linear Payoffs (TS-CL). At cycle t , the bandit context vector is formed by appending the SVGP forecast moments to the current compact state:

$$\mathbf{z}_t = [\mathbf{x}_t, \hat{\mu}_{t+\Delta}, \hat{\sigma}_{t+\Delta}]^\top \quad (4)$$

For each intervention level $a \in \mathcal{A}$, we maintain a Bayesian linear reward model characterized by a multivariate normal posterior over weight vectors $\boldsymbol{\theta}_a$:

$$\boldsymbol{\theta}_a \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_a, \boldsymbol{\Sigma}_a). \quad (5)$$

We sample $\tilde{\theta}_a$ from the posterior to obtain a stochastic value estimate for each action.

1) *Cost-Aware Reward Function*: Crucially, the objective is not just to maximize fidelity, but to maximize *value*: reliability improvement per unit cost. We define the realized reward as improvement in logical error minus a penalty for intervention overhead:

$$r_t(a) = (\epsilon_{L,t} - \epsilon_{L,t+1}) - \lambda \cdot C(a), \quad (6)$$

where $C(a) \in [0, 1]$ is the normalized computational/control cost of intervention level $a \in \{\text{NONE}, \text{MODERATE}, \text{SEVERE}\}$, and λ is a tunable Lagrange multiplier controlling the frugality of the agent.

2) *Imitation Learning (Behavioral Cloning)*: To overcome the cold-start problem where an uninformative prior may fail to discover effective switching behavior within a finite horizon, we bootstrap the bandit using offline imitation learning. We generate synthetic traces of an expert heuristic (e.g., selecting NONE under low inferred noise and SEVERE under high inferred noise) and perform Bayesian updates on the bandit’s priors $(\hat{\theta}_a, \Sigma_a)$ before online deployment. This seeds the agent with domain knowledge while allowing it to fine-tune its behavior online.

F. Runtime Orchestrator

The runtime orchestrator executes closed-loop control cycle:

- 1) **Observe**: receive telemetry $\mathbf{f}_t$ from the backend.
- 2) **Contextualize**: map features to GHSOM context c_t and query SVGP for forecast $(\hat{\mu}_{t+1}, \hat{\sigma}_{t+1})$.
- 3) **Decide**: build $\mathbf{z}_t$, sample action values via Thompson Sampling, subtract costs, and select the optimal intervention level a_t .
- 4) **Act**: execute a_t on the backend, observe the new state and error $\epsilon_{L,t+1}$, and compute the realized reward r_t .
- 5) **Update**: update the posterior distributions for the selected intervention level a_t .

This modular implementation supports direct comparisons among **unmitigated**, **static**, and **adaptive** strategies, isolating the specific value of dynamic switching under drift.

V. RESULTS

We evaluate GSC-QEMit on a suite of eight benchmark circuit workloads spanning three families: **Clifford** (e.g., GHZ state, Bell chain), **Non-Clifford** (e.g., CCX-heavy, T -gate sweep), and **Structured Algorithms** (e.g., QFT, Grover, Bernstein–Vazirani). All experiments use an instrumented Qiskit Aer backend with a controlled nonstationary drift process; the effective physical error rate varies sinusoidally over time (implemented through the simulator noise model) to emulate changing operating conditions.

We compare three strategies under identical drift realizations:

- 1) **Unmitigated (Baseline)**: No intervention is applied (equivalent to intervention level NONE at every cycle).
- 2) **Static Severe**: A fixed high-intensity intervention is applied at every cycle (equivalent to SEVERE at all times), incurring constant overhead.

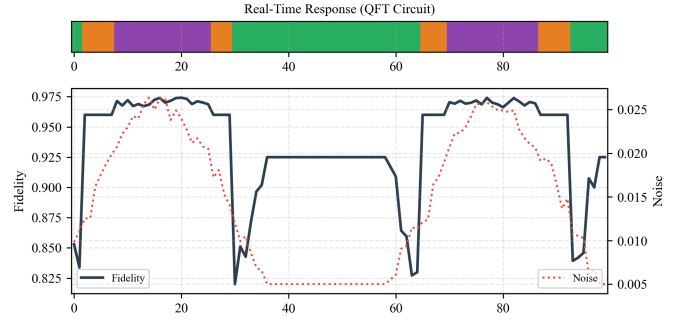


Fig. 2. **Real-time adaptive response (case study: QFT)**. A representative execution trace under drifting noise. **Top strip (barcode)**: the selected intervention level over time (NONE $\rightarrow$ MODERATE $\rightarrow$ SEVERE). **Barcode shading encodes action**: white/green = NONE, gray/orange = MODERATE, black/purple = SEVERE. **Bottom plot**: logical fidelity F_L (solid) alongside the drifting effective noise indicator (dotted). **Key insight**: during elevated-noise intervals, the policy increases intervention intensity (switching to SEVERE); during quiescent periods it returns to NONE/MODERATE, conserving cost while maintaining high fidelity.

- 3) **Adaptive (GSC-QEMit)**: The agent selects among NONE, MODERATE, and SEVERE each cycle based on telemetry, forecasted near-horizon degradation, and an explicit cost penalty.

A. Real-Time Stabilization Under Drift

Figure 3 illustrates representative telemetry traces for the full benchmark suite under a controlled, time-varying noise process. In all the experiments, circuits are executed repeatedly over a fixed horizon T_{run} while the effective physical error rate $p_{\text{eff}}(t)$ is deliberately modulated according to the drifting-noise model described in Section IV-B. No additional faults or adversarial perturbations are injected beyond this explicit drift process.

The drift schedule is designed to probe two complementary regimes. In the first half of each run, the noise level is rapidly increased to a peak, producing a pronounced high-noise interval near the center of the trace. This stress-test phase evaluates whether the controller can detect elevated risk and escalate mitigation quickly. In the second half of the run, the noise level varies more slowly and gradually relaxes, allowing us to observe whether the policy correspondingly de-escalates intervention when aggressive mitigation is no longer beneficial.

Across all six benchmarks, the unmitigated baseline (dashed curves) exhibits a clear fidelity collapse during the high-noise interval, producing the characteristic mid-run trough visible in each panel. In contrast, the adaptive controller (solid curves) responds to the rising predicted risk by escalating intervention intensity, substantially reducing the depth of these troughs. As the noise later decreases and stabilizes, the controller relaxes mitigation, causing the adaptive fidelity to converge toward the unmitigated trace. This convergence reflects intentional behavior: when noise is low and stable, the optimal policy favors minimal intervention to avoid unnecessary overhead.

Overall, the reduced sensitivity of the adaptive fidelity traces to the imposed noise peak demonstrates a genuine

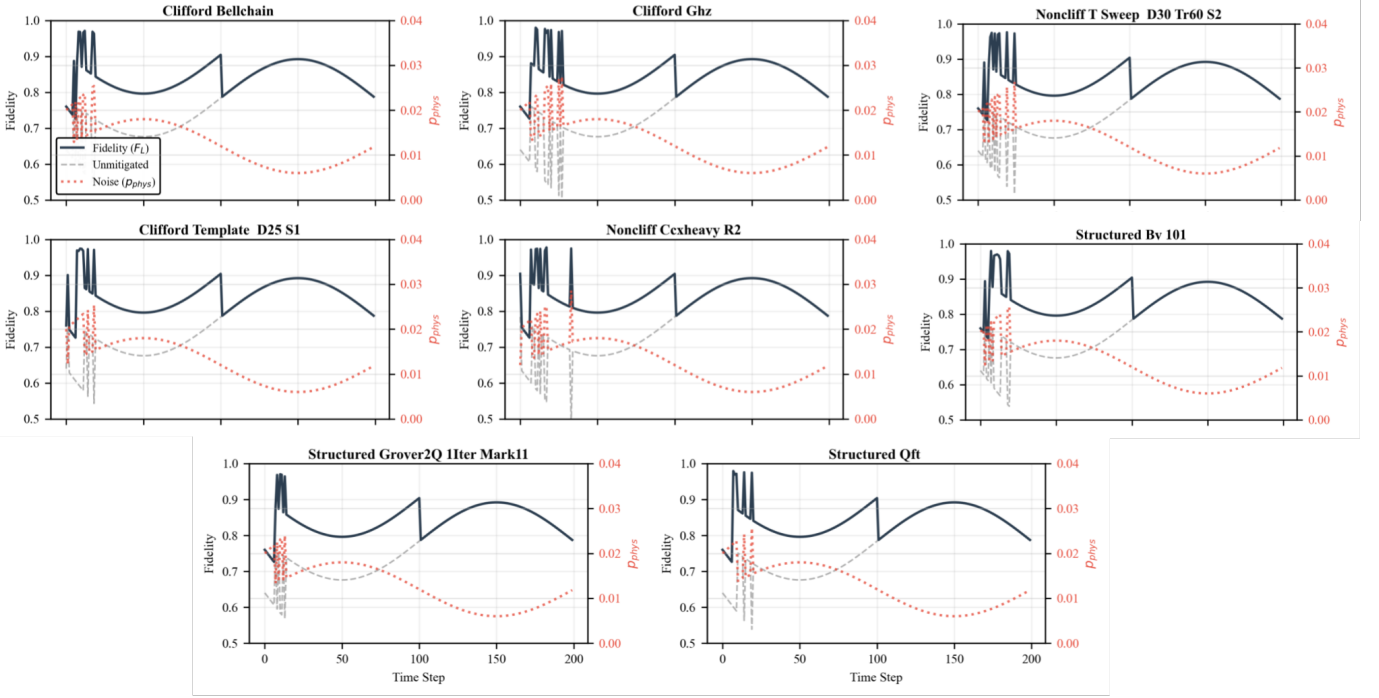


Fig. 3. **Cross-benchmark stabilization under drift.** Time-series traces for the evaluation suite under a controlled, time-varying noise schedule with a deliberate mid-run noise peak. **Dotted:** imposed effective noise indicator $p_{\text{eff}}(t)$. **Dashed:** unmitigated baseline logical fidelity. **Solid:** adaptive fidelity under GSC-QEMit. The reduced depth of the mid-run fidelity trough under the adaptive policy reflects active mitigation escalation during high-noise intervals, followed by de-escalation as noise relaxes.

stabilization effect. Rather than merely tracking drift, GSC-QEMit actively counteracts time-varying noise by modulating mitigation strength in response to forecasted system health.

B. Quantitative Performance Across Workloads

To summarize performance across workloads, Table III reports mean fidelity under unmitigated execution and under the adaptive policy, along with relative gain $\Delta(\%) = 100 \times \left(\frac{F_L^{\text{adaptive}}}{F_L^{\text{unmitigated}}} - 1 \right)$.

TABLE III
PERFORMANCE BY BENCHMARK. WE REPORT MEAN UNMITIGATED FIDELITY, MEAN ADAPTIVE FIDELITY UNDER GSC-QEMIT, AND RELATIVE GAIN $\Delta(\%)$.

Benchmark Performance Summary				
Class	Benchmark	Unmitigated	Adaptive	Gain (%)
Clifford	Bell Chain	0.774	0.843	+8.9%
Clifford	GHZ State	0.771	0.844	+9.4%
Clifford	Template	0.774	0.844	+9.1%
Non-Clifford	CCX-Heavy	0.773	0.844	+9.2%
Non-Clifford	T-Sweep	0.773	0.844	+9.1%
Structured	BV (Oracle)	0.773	0.845	+9.3%
Structured	Grover (Search)	0.775	0.841	+8.5%
Structured	QFT (Algo)	0.774	0.842	+8.7%

Across all benchmarks, GSC-QEMit yields consistent improvements in mean logical fidelity (**+8.5% to +9.4%**) relative to unmitigated execution. Gains are comparable across the three workload families, suggesting that the telemetry-driven

interface captures degradation patterns that are not tied to a single circuit type.

Finally, the adaptive policy achieves these fidelity gains while reducing unnecessary high-intensity intervention. In our runs, the controller selects NONE during low-risk intervals for approximately 40% in the reported setting, thereby reducing aggregate intervention cost by $\sim 35\%$ compared to high-intensity intervention at every cycle.

VI. CONCLUSION

We present *GSC-QEMit*, a telemetry-driven framework for adaptive quantum error mitigation that formulates mitigation selection as a cost-aware policy under nonstationary drift. The approach composes three independently trained, lightweight modules: a Spark-GHSOM context mapper, an SVGP forecaster that predicts near-horizon fidelity with calibrated uncertainty, and a TS-CL contextual bandit that selects among abstract intervention levels (NONE, MODERATE, SEVERE).

Evaluated on an instrumented Qiskit Aer benchmark suite spanning eight circuit workloads, GSC-QEMit stabilizes logical fidelity by escalating mitigation during high-risk intervals and relaxing it under benign conditions. Across workloads, it improves mean logical fidelity by **+9.0%**, reducing aggregate intervention cost by about 35% relative to a static severe strategy. Results show that combining context discovery with uncertainty-aware forecasting enables robust, budgeted mitigation that transfers across workloads without circuit-specific tuning.

ACKNOWLEDGEMENTS

This material is based upon work supported by the United States National Science Foundation (#2225354).

REFERENCES

- [1] P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, “Error mitigation with Clifford quantum-circuit data,” *Quantum*, vol. 5, p. 592, Nov. 2021. [Online]. Available: <https://doi.org/10.22331/q-2021-11-26-592>
- [2] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O’Brien, “Low-cost error mitigation by symmetry verification,” *Phys. Rev. A*, vol. 98, p. 062339, Dec 2018. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.98.062339>
- [3] S. Bravyi, S. Sheldon, A. Kandala, D. C. McKay, and J. M. Gambetta, “Mitigating measurement errors in multiqubit experiments,” *Phys. Rev. A*, vol. 103, p. 042605, Apr 2021. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.103.042605>
- [4] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O’Brien, “Quantum error mitigation,” *Reviews of Modern Physics*, vol. 95, no. 4, p. 045005, 2023.
- [5] K. Temme, S. Bravyi, and J. M. Gambetta, “Error mitigation for short-depth quantum circuits,” *Physical review letters*, vol. 119, no. 18, p. 180509, 2017.
- [6] S. Endo, S. C. Benjamin, and Y. Li, “Practical quantum error mitigation for near-future applications,” *Physical Review X*, vol. 8, no. 3, p. 031027, 2018.
- [7] Z. Liao *et al.*, “Machine-learning-based quantum error mitigation,” *arXiv preprint arXiv:2309.17368*, 2023.
- [8] Z. Wang and H. Tang, “Artificial intelligence for quantum error correction: A comprehensive review,” *arXiv preprint arXiv:2412.20380*, 2024.
- [9] M. Daguerre and M. Sarovar, “Real-time adaptive estimation of noise channels for quantum error mitigation,” *Physical Review A*, vol. 111, p. 062609, 2025.
- [10] Y. Alexeev *et al.*, “Artificial intelligence for quantum computing,” *arXiv preprint arXiv:2411.09131*, 2024.
- [11] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, “Reinforcement learning with neural networks for quantum feedback,” *Phys. Rev. X*, vol. 8, p. 031084, Sep 2018. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevX.8.031084>
- [12] R. Sweke, M. S. Kesselring, E. P. L. van Nieuwenburg, and J. Eisert, “Reinforcement learning decoders for fault-tolerant quantum computation,” *Machine Learning: Science and Technology*, vol. 2, no. 2, p. 025005, dec 2020. [Online]. Available: <https://doi.org/10.1088/2632-2153/abc609>
- [13] S. Varsamopoulos, B. Criger, and K. Bertels, “Decoding small surface codes with feedforward neural networks,” *Quantum Science and Technology*, vol. 3, no. 1, p. 015004, nov 2017. [Online]. Available: <https://doi.org/10.1088/2058-9565/aa955a>
- [14] P. Baireuther, M. D. Caio, B. Criger, C. W. J. Beenakker, and T. E. O’Brien, “Neural network decoder for topological color codes with circuit level noise,” *New Journal of Physics*, vol. 21, no. 1, p. 013003, jan 2019. [Online]. Available: <https://doi.org/10.1088/1367-2630/aaf29e>
- [15] U. U. Shinde and R. Bandaru, “Quantum error-correction using humming sparrow optimization based self-adaptive deep cnn noise correction module,” *Scientific Reports*, vol. 14, no. 1, p. 14289, 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-65182-2>
- [16] Y. Alexeev, M. H. Farag, T. L. Patti, M. E. Wolf, N. Ares, A. Aspuru-Guzik, S. C. Benjamin, Z. Cai, Z. Chandani, F. Fedele, N. Harrigan, J.-S. Kim, E. Kyoseva, J. G. Lietz, T. Lubowe, A. McCaskey, R. G. Melko, K. Nakaji, A. Peruzzo, S. Stanwyck, N. M. Tubman, H. Wang, and T. Costa, “Artificial intelligence for quantum computing,” 2024. [Online]. Available: <https://arxiv.org/abs/2411.09131>
- [17] Z. Wang and H. Tang, “Artificial intelligence for quantum error correction: A comprehensive review,” 2024. [Online]. Available: <https://arxiv.org/abs/2412.20380>
- [18] L. Viola, E. Knill, and S. Lloyd, “Dynamical decoupling of open quantum systems,” *Physical Review Letters*, vol. 82, no. 12, p. 2417–2421, Mar. 1999. [Online]. Available: <http://dx.doi.org/10.1103/PhysRevLett.82.2417>
- [19] J. J. Wallman and J. Emerson, “Noise tailoring for scalable quantum computation via randomized compiling,” *Phys. Rev. A*, vol. 94, p. 052325, Nov 2016. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.94.052325>
- [20] J. a. Gama, I. Zliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Comput. Surv.*, vol. 46, no. 4, Mar. 2014. [Online]. Available: <https://doi.org/10.1145/2523813>
- [21] J. Liu, Q. Langfitt, M. J. Jeng, A. Gonzales, N. Agyeman-Bobie, K. J. S. Vijaymurugan, D. Dilley, Z. H. Saleem, N. Hardavellas, and K. N. Smith, “Quantem: The quantum error management compiler,” *arXiv preprint arXiv:2509.15505*, 2025.
- [22] A. Bifet and R. Gavaldà, “Learning from time-changing data with adaptive windowing,” in *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, 2007, pp. 443–448. [Online]. Available: <https://epubs.siam.org/doi/abs/10.1137/1.9781611972771.42>
- [23] D. Dong and I. R. Petersen, “Quantum control theory and applications: a survey,” *IET control theory & applications*, vol. 4, no. 12, pp. 2651–2671, 2010.
- [24] C. Ahn, A. C. Doherty, and A. J. Landahl, “Continuous quantum error correction via quantum feedback control,” *Physical Review A*, vol. 65, no. 4, p. 042301, 2002.
- [25] F. Battistel, C. Chamberland, K. Johar, R. W. Overwater, F. Sebastiano, L. Skoric, Y. Ueno, and M. Usman, “Real-time decoding for fault-tolerant quantum computing: Progress, challenges and outlook,” *Nano Futures*, vol. 7, no. 3, p. 032003, 2023.
- [26] A. Malondkar, R. Corizzo, I. Kiringa, M. Ceci, and N. Japkowicz, “Spark-ghsom: Growing hierarchical self-organizing map for large scale mixed attribute datasets,” *Information Sciences*, vol. 496, pp. 572–591, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0020025518309496>
- [27] The Apache Software Foundation, *SparkR: R Front End for 'Apache Spark'*, 2025, <https://www.apache.org> <https://spark.apache.org>.
- [28] K. Temme, S. Bravyi, and J. M. Gambetta, “Error mitigation for short-depth quantum circuits,” *Phys. Rev. Lett.*, vol. 119, p. 180509, Nov 2017. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.119.180509>
- [29] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. [Online]. Available: <https://doi.org/10.7551/mitpress/3206.001.0001>
- [30] J. Hensman, A. Matthews, and Z. Ghahramani, “Scalable variational gaussian process classification,” in *Artificial intelligence and statistics*. PMLR, 2015, pp. 351–360.
- [31] S. Agrawal and N. Goyal, “Thompson sampling for contextual bandits with linear payoffs,” in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 127–135. [Online]. Available: <https://proceedings.mlr.press/v28/agrawal13.html>
- [32] D. Russo and B. V. Roy, “An information-theoretic analysis of thompson sampling,” *Journal of Machine Learning Research*, vol. 17, no. 68, pp. 1–30, 2016. [Online]. Available: <http://jmlr.org/papers/v17/14-087.html>
- [33] E. Perrier, A. Yousry, and C. Ferrie, “Qdataset, quantum datasets for machine learning,” *Scientific data*, vol. 9, no. 1, p. 582, 2022.