How Much Fault Tolerance Does a QNN Need? Rethinking Precision Through Convergence Analysis

Anonymous authors
Paper under double-blind review

Abstract

The transition from NISQ to fault tolerance changes how a quantum neural network (QNN) is executed. Trainable rotations are no longer applied directly, but synthesized into sequences of logical gates at a chosen approximation error. Convergence behavior under such error is barely understood. As a result, the error QNN training can tolerate to converge, and the quantum resources that tolerance demands, are set by convention rather than by analysis. We derive a convergence-based theory of synthesis error that gives a principled way to set precision during fault-tolerant QNN training. From a biased-gradient convergence bound, we obtain the gradient distortion an optimizer can tolerate at a given convergence state. We translate this into closed-form tolerance laws for synthesis precision and magic-state fidelity. The laws cover both deterministic synthesis and mixed synthesis, which averages over compilations of the same rotation. We then prove a T-cost theorem for reaching a target stationarity under each strategy, with a matching lower bound for generic rotation angles. The two strategies affect the gradient differently. Deterministic synthesis introduces a first-order bias that persists during training, so the allowable error must shrink as convergence approaches. Mixed synthesis cancels this bias in expectation, leaving only a second-order residual: the error becomes gradient variance, and the same target is reached with fewer T-gates. Experiments with explicit Clifford+T synthesis and end-to-end training on three datasets confirm these predictions: mixed synthesis roughly halves the T cost per circuit relative to convergence-scheduled deterministic synthesis without loss of accuracy, and scheduling magic-state fidelity reduces distillation work by up to 28%. These results show that the fault-tolerant precision of QNN training can be derived from learning rather than fixed by convention.

1 Introduction

Quantum computing is moving from noisy intermediate-scale (NISQ) devices toward fault-tolerant (FT) architectures, and this changes what it costs to train a quantum neural network (QNN). On NISQ hardware, a parameterized rotation is treated as a native operation. On FT hardware, each rotation must be approximated by a sequence of logical Clifford+T gates, whose T count grows as $\Theta(\log(1/\epsilon))$ in the tolerance ϵ for generic angles (Ross & Selinger, 2016), and every T gate consumes a distilled magic state (Bravyi & Kitaev, 2005; Litinski, 2019). Parameter-shift training repeatedly pays this rotation-precision cost. Precision therefore becomes a central resource of FT QNN training. But how much of this precision does learning actually need? Typically, FT execution fixes a synthesis accuracy and pays its cost throughout. We ask the reverse: *given a desired convergence target, what is the least FT precision that training requires?* Classical low-precision training shows that this matters, since numerical errors can be tolerated when they enter optimization in a favorable form, as with stochastic rounding (Gupta et al., 2015). In FT QNNs, however, the error originates from fault-tolerant execution itself. We therefore begin by examining how this error enters the learning process, leading to our first research question (RQ). **RQ1**: How do synthesis and magic-state errors distort the QNN gradient?

Characterizing this error is only part of the problem. Even if we know how FT error enters learning, we still need to know how much of it training can tolerate at different stages of optimization. This

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.

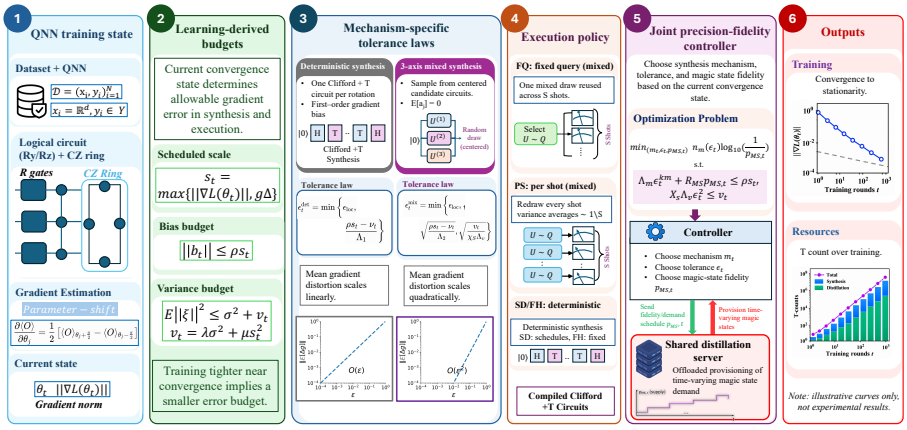


Figure 1: Overview of FT-QNN. (1) Training provides the current gradient norm. (2) This defines FT bias and variance budgets. (3) Deterministic and three-axis mixed synthesis map these budgets to mechanism-specific tolerances (Proposition 1). (4) Mixed synthesis uses fixed-query (FQ) or per-shot (PS) sampling. (5) The controller jointly selects synthesis, tolerance, and magic-state fidelity, while a shared distillation server provisions the requested states. (6) Outputs track training progress and T cost.

naturally leads to our second question. **RQ2:** How does the allowable FT error evolve as training converges? Once the allowable error is tied to convergence, the next question is how adapting FT precision to this changing requirement affects the cost of training. This gives our third question: **RQ3**: How much T cost does convergence-aware synthesis require during training?

A few studies have approached parts of this question from the FT execution side. Existing work usually takes FT precision as an input: synthesis methods optimize the cost of achieving a prescribed accuracy (Hastings, 2017; Campbell, 2017; Kliuchnikov et al., 2023; Ross & Selinger, 2016). Recent FT variational approaches adapt or relax that accuracy to reduce execution overhead (Sayginel et al., 2024; Kang et al., 2026). What remains unanswered is more fundamental: *determining what precision learning itself actually requires for convergence*. This connection remains unexplored in FT QNN training, where execution precision is typically chosen independently of the convergence state. Without that connection, it is unclear whether existing FT precision choices are necessary, overly conservative, or even aligned with the needs of learning. T cost, however, is only part of the FT resource problem. Every T gate must be supplied by a distilled magic state, so synthesis choices and magic-state fidelity are coupled. This leads to our fourth question: **RQ4**: Which synthesis mechanism and magic-state fidelity should the learner request? This naturally implies a time-varying magic-state demand over training. We therefore consider a shared distillation server, to which training offloads these requests. This gives our final question: **RQ5**: How should the distillation server provision this learning-dependent demand?

To answer these questions, we develop **FT-QNN**, a novel convergence-aware framework that links QNN learning directly to fault-tolerant execution. FT-QNN rethinks precision as a learning-dependent resource and traces how convergence requirements propagate from gradient error to synthesis cost, magic-state fidelity, and distillation demand. This goes beyond scheduling precision: FT execution can be matched to what learning actually requires. In doing so, FT-QNN exposes avoidable overhead in logical T execution and magic-state provisioning, reducing a central barrier toward practical FT-QNNs. Our chained contributions are as follows:

1. From convergence to required precision: We characterize how much FT precision QNN training needs, deriving bounds on the admissible synthesis & magic-state error from the optimization state, and prove a convergence bound under these budgets for SGD, momentum, & Adam scaling.

2. From required fault tolerance to T-cost reduction: We show that synthesis-error structure changes the precision needed for convergence. Three-axis mixed synthesis asymptotically halves the leading precision-dependent T cost and cuts cumulative T consumption by about half relative to convergence-scheduled deterministic synthesis.

3. From T demand to magic-state provisioning. We extend the convergence budget to magic-state fidelity and offload the resulting time-varying demand to a shared multi-fidelity distillation server, reducing distillation work while preserving the learning target.

4. We validate the predicted error scaling and convergence-aware precision laws with extensive explicit Clifford+T simulation, directly linking the theoretical budgets to observed training and resource behavior.

2 Related Work

A. Gate synthesis and FT variational algorithms: Arbitrary rotations are approximated in Clifford+T with a T count that grows logarithmically in the target accuracy (Ross & Selinger, 2016). Mixed synthesis combines several approximations of the same rotation to suppress coherent error quadratically (Hastings, 2017; Campbell, 2017), while optimized mixtures can further reduce synthesis cost (Kliuchnikov et al., 2023). More recent work also studies shaping the residual error channel produced by mixed synthesis (Yoshioka et al., 2025). Randomization has further been shown to reduce the logarithmic precision cost of quantum signal processing (Martyn & Rall, 2025). These works, in general, show how to realize a prescribed accuracy more efficiently, but do not determine what accuracy learning actually requires. Most QNN studies operate at the algorithmic level, treating parameterized rotations as directly available and modeling implementation errors as perturbations of the circuit or rotation parameters (Wang et al., 2022; Rabinovich et al., 2024). Moving toward FT execution, Sayginel et al. (2024) compile variational rotations into Clifford+T and adapt synthesis accuracy during optimization, while Kang et al. (2026) reduce QML overhead by relaxing fault tolerance for selected parameterized rotations. These works show that FT execution can be adapted or relaxed, but the required precision is not derived from a convergence condition.

B. Noisy optimization and low-precision training: Nonconvex SGD admits convergence guarantees under bounded gradient variance and controlled gradient bias (Demidovich et al., 2023; Hu et al., 2021), with related results for adaptive optimization (Défossez et al., 2020). Classical low-precision training exploits similar structure through stochastic rounding and unbiased quantization, preventing numerical error from accumulating systematically (Gupta et al., 2015; Alistarh et al., 2017). These results explain how classical learning can tolerate error, and motivate our approach.

C. Magic-state resources: Each logical T gate consumes a distilled magic state (Bravyi & Kitaev, 2005), and their production is a major component of FT resource estimates (Fowler et al., 2012; O'Gorman & Campbell, 2017; Beverland et al., 2022). Existing resource analyses therefore study synthesis and distillation under prescribed circuit-level accuracy requirements. They do not consider a fidelity demand derived from the evolving convergence requirements of learning.

Positioning of FT-QNN: Prior work in **A-C** studies the individual pieces of FT execution, optimization under error, and resource provisioning, but treats the required execution accuracy as given or externally chosen. FT-QNN instead derives this requirement from the learning process and carries it through synthesis cost, magic-state fidelity, and distillation demand. This creates a direct learning-to-resource connection that is absent from the individual lines of work above. Appendix G provides a broader comparison with related synthesis, FT-QML, optimization, and magic-state provisioning work.

3 FT-QNN: Method and Theory

3.1 Method

At each optimizer update, FT-QNN converts the current training state into an FT error budget, and the budget into a synthesis mechanism, tolerance, and magic-state fidelity. Rotations are compiled into Clifford+T circuits, and the requested magic states are supplied by a shared multi-fidelity distillation server (Figure 1).

Fault-tolerant QNN training: We consider a circuit $U(\theta)$ on n logical qubits with $\theta \in \mathbb{R}^P$, where each layer applies rotations $R_Q(\varphi) = e^{-i\varphi Q/2}$, $Q \in \{Y, Z\}$, followed by a CZ ring. For an encoded input $|\psi(x)\rangle$, the prediction is $f_\theta(x) = \langle\psi(x)|U(\theta)^\dagger O U(\theta)|\psi(x)\rangle$, and training minimizes the mean-squared loss $L(\theta)$. Parameter-shift differentiation on a batch of B samples needs $B(1+2P)$ circuit queries, each with S shots (Figure. 1, step 1). On FT hardware, each rotation is replaced by a Clifford+T circuit $\widetilde{U}$ with $\min_\alpha \|\widetilde{U} - e^{i\alpha} R_Q(\varphi)\|_2 \le \epsilon$, whose T count grows as $\Theta(\log(1/\epsilon))$ for generic angles, and every T gate consumes a distilled magic state. *The tolerance ϵ is therefore not*

only a compilation choice: it sets the FT resources training consumes. We measure this through the pre-shot T workload $W_t = \sum_j \mathrm{T}(\widetilde{U}_j)$ of update t, over all encoding, forward, and parameter-shift rotations, the executed workload $M_t = SW_t$, and the cumulative consumption $C_T(T) = \sum_{t<T} M_t$ (step 6).

Three-axis mixed synthesis: Synthesis error is not an angle error alone. We write the phase-aligned error of rotation j as $E_j = \widetilde{U}_j U_j^\dagger = \exp[-i(a_j \cdot \sigma)/2]$ with $a_j \in \mathbb{R}^3$ and $\sigma = (X, Y, Z)$. One component of a_j is an over- or under-rotation about the intended axis; the other two are transverse rotations, which angle-only error models, common for NISQ QNNs, can neither capture nor cancel. Deterministic synthesis uses one Clifford+T approximation per angle and tolerance. Our mixed construction instead samples from candidate circuits whose error generators satisfy $\mathbb{E}[a_j] = 0$ in all three directions (step 3). Candidates are drawn either once per query and reused across its S shots (*fixed-query*, FQ) or anew for every shot (*per-shot*, PS), which averages synthesis randomness across measurements (step 4). Base and shifted queries draw independently in both modes.

Convergence-aware precision and fidelity scheduling: Let $s_t = \max\{\|\nabla L(\theta_t)\|, g_\Delta\}$ be the convergence level, where the floor g_Δ keeps precision bounded near the target stationarity Δ. FT-QNN allows a gradient bias of ρs_t and a synthesis variance of v_t (step 2). The two mechanisms spend this budget differently: deterministic tolerance must shrink linearly with ρs_t, while mixed tolerance shrinks only as its square root until the variance limit binds (Proposition 1, step 3); the factor χ_S, equal to 1 for FQ and $1/S$ for PS, sets that limit. Magic-state error draws on the same bias budget. For a requested infidelity $p_{\mathrm{MS},t}$ with response bound R_{MS}, the controller requires $\Lambda_m \epsilon_t^{k_m} + R_{\mathrm{MS}} p_{\mathrm{MS},t} \le \rho s_t$, with $k_m = 1$ for deterministic and $k_m = 2$ for mixed synthesis, plus the variance limit for mixed actions. Among feasible $(m_t, \epsilon_t, p_{\mathrm{MS},t})$, it minimizes the proxy $n_m(\epsilon_t) \log_{10}(1/p_{\mathrm{MS},t})$, where n_m is the expected T count per rotation; the proxy ranks actions and is not a factory cost (step 5). In our experiments, $s_t = \|\nabla L(\theta_t)\|$ comes from an oracle gradient monitor, with $g_\Delta = 0$ and the tolerance grid bounded below by 10^{-6}, and the variance limit is enforced as an Adam-specific rule: a mixed tolerance is admitted only if it perturbs the next update no more than a reference parameter-shift estimator does (Appendix C).

Distillation-server provisioning: The schedule produces a time-varying demand for magic states at several fidelities. Instead of serving every T gate from one high-fidelity factory, FT-QNN sends each update's requests to a shared multi-fidelity server, which selects the smallest feasible distillation configuration for each fidelity (step 5). We replay the demand through a conditional distillation model that maps each fidelity to a code distance and normalized factory volume, and measure total factory work, the sustained capacity needed to avoid stalls, and the peak capacity (step 6). These are normalized quantities, not physical-qubit counts or runtime.

4 Theoretical Analysis

We show how three-axis synthesis error changes the gradient, turn this into tolerances and T cost, and then examine how optimizer noise and path length shape the advantage of mixed synthesis.

A. Gradient response to three-axis synthesis error *(answers RQ1; supports RQ2)*
Consider an update with N synthesized rotation instances, and let $a = (a_1, \ldots, a_N) \in \mathbb{R}^{3N}$ collect their error generators (Section 3). For fixed errors, let $\bar{G}(\theta, a)$ be the gradient estimator averaged over shot and mini-batch randomness, so $\bar{G}(\theta, 0) = \nabla L(\theta)$, and let $R_j = D_{a_j} \bar{G}(\theta, 0) \in \mathbb{R}^{P \times 3}$ be the first-order response of rotation j.

***Assumption* 1** (Local response)**.** *$\bar{G}(\theta, a)$ is three times continuously differentiable in a near $a = 0$. Each error satisfies $\|a_j\| \le c_a \epsilon$ for $\epsilon \le \epsilon_{\mathrm{loc}}$, and mixed draws are independent across all rotation instances, including those of base and shifted queries, and of shot and mini-batch randomness.*

***Lemma* 1** (Deterministic first-order response)**.** *For deterministic synthesis, the gradient perturbation is linear in the synthesis error: $\bar{G}(\theta, a) - \nabla L(\theta) = \sum_{j=1}^N R_j a_j + O(\epsilon^2)$. Hence there exists $\Lambda_1 > 0$, absorbing the remainder, such that $\|b_{\mathrm{det}}^{\mathrm{syn}}\| \le \Lambda_1 \epsilon$ for $\epsilon \le \epsilon_{\mathrm{loc}}$ (Proof: Appendix B.2).*

Since a_j has all three Pauli components, this term survives unless the error falls in a locally insensitive direction. Mixed synthesis centers the whole generator.

***Lemma* 2** (Three-axis centering)**.** *If $\mathbb{E}[a_j] = 0$ in all three directions, the first-order term leaves the mean gradient and enters its variance. With $C_j = \mathbb{E}[a_j a_j^\top]$ and $\Sigma_{\mathrm{FT}} = \mathrm{Cov}_a \bar{G}(\theta, a)$,*

$b^{\mathrm{syn}}_{\mathrm{mix}} = \frac{1}{2}\sum_j D^2_{jj}\bar{G}(\theta,0)[C_j] + O(\epsilon^3)$, *where* $D^2_{jj}\bar{G}[C_j] = \sum_{k,l}(C_j)_{kl}\partial_{a_{j,k}}\partial_{a_{j,l}}\bar{G}$, *and* $\Sigma_{\mathrm{FT}} = \sum_j R_j C_j R_j^\top + O(\epsilon^3)$. *Hence* $\|b^{\mathrm{syn}}_{\mathrm{mix}}\| \le \Lambda_2\epsilon^2$ *and* $\operatorname{tr}\Sigma_{\mathrm{FT}} \le \Lambda_v\epsilon^2$ *for* $\epsilon \le \epsilon_{\mathrm{loc}}$ *(Proof: Appendix B.3).*

Fixed-query synthesis keeps this covariance across the S shots, while per-shot redraws divide it by S, so $\sigma^2_{\mathrm{FT,mix}} \le \chi_S\Lambda_v\epsilon^2$ with $\chi_S \in \{1, 1/S\}$ (Appendix B.1). Centering only the rotation-axis component leaves the transverse terms of $\sum_j R_j\mathbb{E}[a_j]$, so the bias stays first order. Without independence, the joint covariance $\mathbb{E}[aa^\top]$ replaces the block sum, but the bias stays $O(\epsilon^2)$. The lemma concerns the per-draw estimator, not the gradient of the MSE of the channel-averaged prediction (Appendix B.1).

B. Convergence-compatible precision ***(answers RQ2; supports RQ4)***

Write $\hat{g}_t = \nabla L(\theta_t) + b_t + \xi_t$ with $b_t = \mathbb{E}[\hat{g}_t \mid \mathcal{F}_t] - \nabla L(\theta_t)$ and $\mathbb{E}[\xi_t \mid \mathcal{F}_t] = 0$. An oracle monitor gives $s_t = \max\{\|\nabla L(\theta_t)\|, g_\Delta\}$ with $g_\Delta = \phi\Delta$; noisy monitors are outside the present guarantee. Following Ajalloeian & Stich (2020), we require $\|b_t\| \le \rho s_t$ and allow a synthesis-variance budget $v_t = \lambda\sigma^2 + \mu s_t^2$, where σ^2 bounds intrinsic shot and mini-batch variance uniformly over $\|a_j\| \le c_a\epsilon_{\mathrm{loc}}$. Other FT errors share the bias budget through $\nu_t = R_{\mathrm{MS}}p_{\mathrm{MS},t} + R_{\mathrm{dep}}p_{\mathrm{dep}} < \rho s_t$, with trajectory-level response bounds $R_{\mathrm{MS}}, R_{\mathrm{dep}}$.

***Proposition* 1** (Mechanism-specific tolerance laws)**.** *Sufficient tolerances are* $\epsilon_t^{\mathrm{det}} = \min\{\epsilon_{\mathrm{loc}}, (\rho s_t - \nu_t)/\Lambda_1\}$ *and*

$$\epsilon_t^{\mathrm{mix}} = \min\Big\{ \underbrace{\epsilon_{\mathrm{loc}}}_{\textit{local validity}}, \underbrace{\sqrt{(\rho s_t - \nu_t)/\Lambda_2}}_{\textit{bias cap}}, \underbrace{\sqrt{v_t/(\chi_S\Lambda_v)}}_{\textit{variance cap}} \Big\} \tag{1}$$

(Proof: Appendix B.4).

When v_t is dominated by $\lambda\sigma^2$, let $\epsilon_v = \min\{\epsilon_{\mathrm{loc}}, \sqrt{\lambda\sigma^2/(\chi_S\Lambda_v)}\}$. Mixed synthesis is variance limited above the crossover $s_c = (\Lambda_2\epsilon_v^2 + \nu_t)/\rho$ and bias limited below it. Since $s_t \ge \phi\Delta$, a fixed residual bias needs $\bar{\nu} = \sup_t \nu_t < \rho\phi\Delta$, so tighter targets need smaller magic-state or residual logical errors. Under the cost model of Section C, the preferred mechanism switches at most once, from deterministic to mixed, and the bias split with magic-state error follows from equal marginal costs (Appendix B.10).

C. Cumulative T cost ***(answers RQ3)***

For generic rotations, synthesis cost is logarithmic in precision (Ross & Selinger, 2016); we assume $n_m(\epsilon) \le \kappa_m\log_{10}(1/\epsilon) + \beta_m$ for $m \in \{\mathrm{det}, \mathrm{mix}\}$. With N_t synthesized rotations per update before shot repetition, $\mathbb{E}[C_m(T)] \le S\sum_{t=0}^{T-1} N_t[\kappa_m\log_{10}(1/\epsilon_t^m) + \beta_m]$, so cost depends on both the schedule and the number of updates. The lower bound needs generic angles and a nondegenerate response.

***Assumption* 2** (Generic-angle lower bound)**.** *A set* $\mathcal{G}$*, containing a fraction* $q_{\mathrm{gen}} > 0$ *of synthesized rotations, consists of generic angles whose approximation at operator distance* δ *needs at least* $\kappa_-\log_{10}(1/\delta) - c_0$ *T gates. For these rotations,* $\|b^{\mathrm{syn}}_{\mathrm{det}}\| \ge \gamma\max_{j\in\mathcal{G}}\|a_j\|$ *for some* $\gamma > 0$*, so errors neither fall into insensitive directions nor cancel across gates.*

***Theorem* 1** (T-cost scaling at the convergence floor)**.** *Suppose Assumption 2 holds,* $\bar{\nu} \le \frac{1}{2}\rho\phi\Delta$*, and* ϵ_{loc} *is not binding. At* $s_t = \phi\Delta$*, any deterministic schedule meeting the bias budget needs at least* $\kappa_-\log_{10}(1/\Delta) - O(1)$ *T gates per generic rotation, and Proposition 1 attains* $\kappa_{\mathrm{det}}\log_{10}(1/\Delta) + O(1)$*. In the bias-limited regime, the mixed schedule costs between* $\frac{1}{2}\kappa_-\log_{10}(1/\Delta) - O(1)$ *and* $\frac{1}{2}\kappa_{\mathrm{mix}}\log_{10}(1/\Delta) + O(1)$*, since each candidate synthesizes a generic angle to distance at most* $\epsilon_t^{\mathrm{mix}} = \Theta(\sqrt{\Delta})$ *(Proof: Appendix B.5).*

***Corollary* 1** (Precision-cost regimes)**.** *Let* $r = \mathbb{E}[C_{\mathrm{det}}]/\mathbb{E}[C_{\mathrm{mix}}]$ *and* $h = T_{\mathrm{det}}/T_{\mathrm{mix}}$*, so* r/h *separates synthesis cost from path length. Without intrinsic noise and with variance shrinking as* $O(s_t^2)$*, both mechanisms have the same leading logarithmic scaling. With a noise floor and* $s_t > s_c$*, mixed synthesis is variance limited while deterministic cost grows with* $\log(1/\Delta)$*. Below the crossover,* $\epsilon_{\mathrm{det}} = O(\Delta)$ *and* $\epsilon_{\mathrm{mix}} = O(\sqrt{\Delta})$*, so the per-rotation cost ratio approaches "two" when* $\kappa_- = \kappa_{\mathrm{det}} = \kappa_{\mathrm{mix}}$ *and* β_m *is negligible (Proof: Appendix B.6).*

The factor of two is asymptotic: a finite-run saving near one half does not by itself place a run in the bias-limited regime, since the ratio also rises toward its crossover value in the variance-limited regime.

D. Stationarity and path length *(completes RQ3)*
The results above concern the gradient estimator and do not depend on the optimizer; only the convergence guarantee does. Assume L is ℓ-smooth and lower bounded, with $D = L(\theta_0) - \inf_\theta L(\theta) < \infty$, and take $\theta_{t+1} = \theta_t - \eta\hat{g}_t$. For a target Δ, let $\eta_\Delta = \min\{1/\ell,\ \rho^2/(2\ell\mu),\ (1-\rho^2)\Delta^2/[16\ell(1+\lambda)\sigma^2]\}$ and $T(\Delta) = \lceil 8D/[(1-\rho^2)\eta_\Delta\Delta^2]\rceil$, omitting terms with zero denominators.

***Theorem* 2** (Convergence under FT error budgets)**.** *Let $\rho \le 1/\sqrt{2}$, and suppose the bias and variance budgets hold at every update. For any $\eta \le \min\{1/\ell,\ \rho^2/(2\ell\mu)\}$ and horizon T, SGD satisfies*

$$\frac{1}{T}\sum_{t<T}\mathbb{E}\|\nabla L(\theta_t)\|^2 \le \underbrace{\frac{4D}{(1-\rho^2)\eta T}}_{\text{initial gap}} + \underbrace{\frac{3\rho^2 g_\Delta^2}{1-\rho^2}}_{\text{bias floor}} + \underbrace{\frac{2\ell\eta(1+\lambda)\sigma^2}{1-\rho^2}}_{\text{shot, batch, and synthesis variance}}. \tag{2}$$

With $g_\Delta = \phi\Delta$, $\phi = \sqrt{(1-\rho^2)/8}/\rho$, $\eta = \eta_\Delta$, and $T = T(\Delta)$, the terms are at most $\Delta^2/2$, $3\Delta^2/8$, and $\Delta^2/8$, so θ_τ with τ uniform on $\{0,\dots,T-1\}$ satisfies $\mathbb{E}\|\nabla L(\theta_\tau)\|^2 \le \Delta^2$. Deterministic synthesis is $\lambda = \mu = 0$, and for $\sigma > 0$ with separately tuned step sizes, $T_{\mathrm{det}}/T_{\mathrm{mix}} \to 1/(1+\lambda)$. With momentum, $m_t = \beta m_{t-1} + (1-\beta)\hat{g}_t$ and $\theta_{t+1} = \theta_t - \eta m_t$, all of this holds with ℓ replaced by $\ell_\beta = \ell(1+\beta)/(1-\beta)$. For Adam, let $P_t = \mathrm{diag}\big(1/(\sqrt{v_{t-1}}+\varepsilon)\big)$ have entries in $[p_{\min}, p_{\max}]$ from update t_0 onward (with D measured from θ_{t_0}), with $\kappa = p_{\max}/p_{\min}$ and $c_\kappa = 1-\kappa\rho^2 \ge 1/2$, and let gradients be bounded by G in norm and G_∞ per coordinate. With $\ell_A = 2\ell(1+\kappa\beta/(1-\beta))$, $C_A = 4\kappa p_{\max}\beta G^2 G_\infty/[(1-\beta)c_\kappa]$, and $\eta \le \min\{1/(\ell_A p_{\max}),\ \rho^2/(2\ell_A p_{\max}\mu)\}$,

$$\frac{1}{T}\sum_{t<T}\mathbb{E}\|\nabla L(\theta_t)\|^2 \le \frac{4D}{c_\kappa p_{\min}\eta T} + \frac{3\kappa\rho^2 g_\Delta^2}{c_\kappa} + \frac{2\ell_A\eta\,\kappa p_{\max}(1+\lambda)\sigma^2}{c_\kappa} + C_A\sqrt{1-\beta_2} + O(\eta), \tag{3}$$

where the last two terms come from the drift of P_t and vanish as $\beta_2 \to 1$ and $\eta \to 0$ (Appendices B.7 and B.8).

The descent step follows Ajalloeian & Stich (2020). The FT budgets leave a bias floor set by $\phi\Delta$, and synthesis variance limits the step size through v_t, lengthening the horizon. Momentum and scaling change only constants; Adam's changing scaling adds a floor that vanishes as $\beta_2 \to 1$. The Adam constants are worst-case: the extension shows that the guarantee's structure does not depend on the optimizer, while RQ1 to RQ3 verify the orders and tolerance laws on the Adam runs directly. With separately tuned step sizes, the certified cumulative-cost ratio tends to $2/(1+\lambda)$, so a lower T cost per update gives a certified cumulative saving only for $\lambda < 1$ (Appendix B.9).

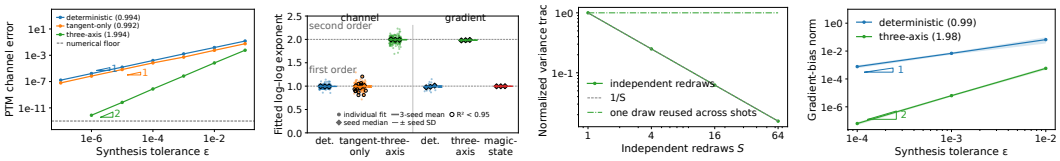
5 Experiments

5.1 Setup

Data and models: Since our focus is FT synthesis, convergence, and resource cost rather than dataset-specific accuracy, the datasets serve as representative learning workloads. Digits (0 vs. 1) Alpaydin & Kaynak (1998) is the primary task, while UNSW-NB15 Moustafa & Slay (2015) and Waveform-5000 Breiman & Stone (1984) test transfer beyond image data, each with 128/64/128 splits. The main circuit uses 6 logical qubits, 9 layers, $P = 108$, and a global parity observable; size studies span 4–8 qubits and 4–9 layers. (More details in Appendix C).

Training: We use Adam ($\beta = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, learning rate initially 0.1, reduced to 0.03, 0.01, 0.003, and 0.001 at updates 100, 250, 500, and 800) with parameter-shift gradients, so each selected example costs $1 + 2P$ circuits. Circuits are estimated with 500 or 1,000 shots, with batches of 32, 64, or 200. Arms are compared at a matched update (800 for digits, 500 otherwise), averaged over three seeds.

Execution policies and reference configurations: To our knowledge, no prior work studies fault-tolerant QNN training through the same fundamental questions considered here, so no direct baseline exists. The closest studies examine fixed or adaptive Clifford+T compilation for variational circuits (Beverland et al., 2022; Sayginel et al., 2024). We therefore use fixed-high deterministic synthesis **(FH)** as the fixed-accuracy reference and scheduled deterministic synthesis **(SD)** as the closest adaptive reference, with SD using the convergence-compatible tolerance of Proposition 1. FT-QNN evaluates three-axis mixed synthesis, extending mixed-unitary synthesis (Hastings, 2017;



(a) Channel error vs. ϵ (b) Fitted error order (c) σ^2_{FT} vs. S (d) Gradient bias vs. ϵ

Figure 2: Synthesis error and gradient effects (RQ1), mean over three seeds with $\pm$ SD. On log–log axes, slopes 1 and 2 indicate $O(\epsilon)$ and $O(\epsilon^2)$ scaling; ◿ marks fitted slopes. (a) Channel error over 64 angles. (b) Fitted exponents. (c) Gradient variance for independent vs. reused draws. (d) Gradient bias over 6 checkpoints.

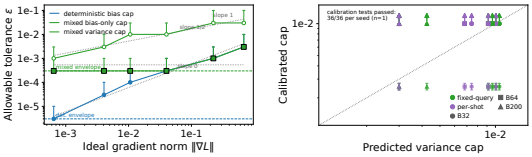
Campbell, 2017), through fixed-query **(FQ)** and per-shot **(PS)** policies, followed by joint fidelity scheduling (RQ4) and distillation provisioning (RQ5). Ideal training serves only as a quality reference. Appendix C.1 details how each configuration is chosen and run.

Environment: Experiments used a custom state-vector simulator on an NVIDIA RTX 4090 GPU, with Python 3.9.20, PyTorch 2.4.1, and CUDA 12.1. States used complex128 precision, while parameters and gradients used float64. Rotations were synthesized with `pygridsynth` 1.1.0.

5.2 Results on RQs

RQ1: How do synthesis and magic-state errors distort the gradient?

A key finding is that fault-tolerant execution error does not affect the optimizer uniformly: its gradient impact depends on how each rotation is synthesized. We compare three-axis mixed synthesis with deterministic Ross–Selinger synthesis (Ross & Selinger, 2016) and tangent-only mixing, which corrects only the intended rotation angle as in angle-perturbation noise models (Wang et al., 2022; Rabinovich et al., 2024). Figure 2a shows that deterministic error scales as $O(\epsilon)$, whereas three-axis mixed synthesis achieves $O(\epsilon^2)$ suppression across all three error axes, extending mixed-unitary results (Hastings, 2017; Campbell, 2017; Kliuchnikov et al., 2023). Tangent-only mixing remains first order because about 59% of the synthesis error lies on the two transverse axes. Figures 2b and 2d show the same behavior in digits training: the gradient-bias exponent is 0.9907 ± 0.0189 for deterministic and 1.9832 ± 0.0055 for mixed synthesis, matching Lemmas 1 and 2. The trade-off is gradient variance, which matches $\sum_j R_j C_j R_j^\top$ within 0.1% and averages as $1/S$ under per-shot resampling (Figure 2c). Magic-state infidelity adds a separate first-order bias (Appendix D.1). Thus, for equal gradient bias, mixed synthesis can use much looser tolerances and fewer T gates, provided the optimizer can absorb the added noise. RQ2 studies this trade-off.



(a) Allowable ϵ vs. $\|\nabla L\|$ (b) Predicted vs. calibrated

Figure 3: Allowable tolerance (RQ2). (a) Caps at six Adam checkpoints; dotted lines show the slopes of Proposition 1. (b) Frozen predictions vs. calibration; arrows mark censored brackets.

Answer to RQ1: *three-axis mixing makes synthesis bias second order in ϵ, turning the first-order error into averageable noise.*

RQ2: How does the allowable FT error evolve during training? Proposition 1 predicts that the deterministic tolerance must shrink in proportion to the gradient norm, while the mixed tolerance shrinks only as its square root until the variance cap binds. Deterministic synthesis is the baseline. Figure 3a shows this split over training. As $\|\nabla L\|$ falls from 0.68 to 6.5×10^{-4}, the deterministic cap tightens from 3×10^{-3} to 3×10^{-6}. The mixed cap stops at 3×10^{-4} from update 50 onward. It is set by variance: the bias-only mixed cap would allow 10^{-3}, which is $333\times$ looser than deterministic. Over the whole run, mixed synthesis can therefore use a $100\times$ looser tolerance than deterministic synthesis. Figure 3a uses a low-noise 4,096-shot reference. With the actual training noise, the variance cap rises, and the theory predicts when bias

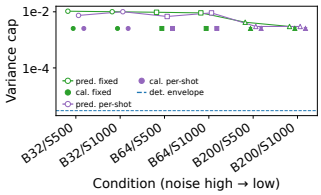


Figure 4: Predicted and calibrated variance caps, ordered by noise floor (RQ2).

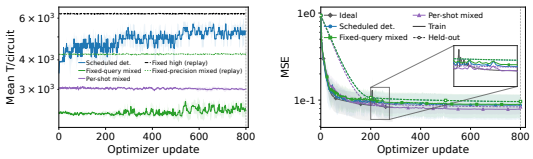
Table 1: Effect of the noise floor on the main circuit (digits, 6q/9L), at matched update 800. Entries are mean ± SD over three seeds. FH: fixed high; SD: scheduled deterministic; FQ: fixed-query mixed; PS: per-shot mixed. †Cost replay along the scheduled-deterministic trajectory.

	T per circuit ($\times 10^3$)				FQ saving (%)		Held-out MSE ($\times 10^{-2}$)			Acc.
Batch / shots	FH†	SD	FQ	PS	vs. FH	vs. SD	SD	FQ	PS	(%)
B32 / S500	6.26 ± 0.02	5.29 ± 0.33	2.44 ± 0.19	3.01 ± 0.03	61.1	54.0	9.59 ± 1.85	9.64 ± 1.91	8.36 ± 0.84	99.6 ± 0.4
B32 / S1000	6.27 ± 0.01	5.10 ± 0.01	2.29 ± 0.01	3.04 ± 0.01	63.4	55.1	8.07 ± 10.20	8.05 ± 10.08	7.87 ± 9.29	99.7 ± 0.3
B64 / S500	6.26 ± 0.01	5.64 ± 0.01	2.47 ± 0.01	2.96 ± 0.01	60.6	56.3	8.41 ± 10.40	8.18 ± 10.43	8.17 ± 10.06	99.9 ± 0.1
B64 / S1000	6.23 ± 0.01	5.66 ± 0.01	2.48 ± 0.01	3.03 ± 0.01	60.2	56.3	8.51 ± 11.05	9.16 ± 12.38	9.92 ± 11.74	99.4 ± 0.6
B200 / S500	6.28 ± 0.02	5.65 ± 0.01	2.90 ± 0.01	3.05 ± 0.01	53.9	48.8	8.33 ± 10.85	8.65 ± 11.58	7.89 ± 9.99	99.7 ± 0.3
B200 / S1000	6.27 ± 0.02	6.27 ± 0.02	3.05 ± 0.01	3.03 ± 0.01	51.3	51.3	7.92 ± 9.63	7.91 ± 9.79	7.77 ± 9.63	99.6 ± 0.4

takes over. Before any training run, we froze the variance caps and crossovers for six batch and shot settings. The predicted crossovers lie between $\|\nabla L\| = 0.0032$ and 0.035, so every setting should be bias-limited by update 368, where $\|\nabla L\| = 6.5 \times 10^{-4}$. Calibration with each setting's own noise confirms this. At update 368, the bias test fails above 0.0027 in every setting while both variance tests still pass, so all settings share one cap despite a $77\times$ range in intrinsic noise (Figure 4). This is the bias-limited regime of Corollary 1. Earlier in training, the calibrated caps exceeded 10^{-2} and are censored, so the predicted rise of the variance cap with noise could not be resolved. All 36 predeclared tests passed without retuning (Figure 3b). Note that the variance rule for Adam is an empirical next-update test, not part of Theorem 2.

Answer to RQ2:*deterministic tolerance tightens with the gradient, while mixed tolerance stays $100\times$ looser and becomes bias-limited near convergence, as predicted.*

RQ3: How much T cost does convergence-scheduled mixed synthesis save during training?



(a) T per circuit vs. update (b) MSE vs. update

Figure 5: Training cost and quality on the main evaluation circuit, mean ± SD over three seeds. (a) Fixed-high and fixed-precision mixed are cost replays. (b) Solid: training MSE; dashed: held-out MSE.

We compare two baselines: fixed high precision (FH), which compiles every rotation at the tightest tolerance used in training ($\epsilon = 3 \times 10^{-6}$), and scheduled deterministic synthesis (SD). Figure 5a shows the main-circuit cost. FH remains near 6.3k T per circuit, while SD rises from about 3.5k to 5.3k as its tolerance tightens with the gradient. Fixed-query mixed synthesis (FQ) stays near 2.4k. At update 800, FQ saves 61.1% versus FH and 54.0% versus SD. Scheduling alone saves 15.5%, while mixing alone saves 32.5%, showing that the two effects compound. The FQ–SD gap is close to the factor of two in Theorem 1, although this finite-run result is not itself the asymptotic theorem. Figure 5b shows comparable quality: held-out MSE is 0.0959 for SD and 0.0964 for FQ, while per-shot mixing (PS) reaches 0.0836 at 3.0k T per circuit.

Table 2: Effect of circuit size scaling. Bold marks the dimension held fixed. Digits, B32 / S500, update 800; mean ± SD over 3 seeds. Saving: FQ relative to SD.

	T per circuit ($\times 10^3$)		Saving	Held-out MSE ($\times 10^{-2}$)		Acc.
Model	SD	FQ	(%)	SD	FQ	(%)
4q / **9L**	3.39 ± 0.01	1.59 ± 0.01	53.1	8.91 ± 13.01	9.15 ± 13.19	99.9 ± 0.1
6q / **9L**⋆	5.29 ± 0.33	2.44 ± 0.19	54.0	9.59 ± 1.85	9.64 ± 1.91	99.6 ± 0.4
8q / **9L**	6.82 ± 0.01	3.35 ± 0.01	51.0	8.45 ± 10.84	10.05 ± 13.27	99.8 ± 0.2
6q / 4L	2.41 ± 0.02	1.01 ± 0.01	58.2	11.34 ± 13.89	10.58 ± 13.55	99.9 ± 0.1
6q / 6L	3.09 ± 0.01	1.61 ± 0.01	48.0	9.60 ± 11.46	8.72 ± 11.35	99.6 ± 0.4
6q / 9L⋆	5.29 ± 0.33	2.44 ± 0.19	54.0	9.59 ± 1.85	9.64 ± 1.91	99.6 ± 0.4

Table 1 extends the comparison across six batch/shot settings. FQ saves 51–63% versus FH and 49–56% versus SD, with held-out accuracy at least 99.4% throughout. The saving does not increase monotonically with the noise floor, consistent with RQ2: near convergence, the mixed tolerance becomes bias-limited. At B200/S1000, SD reaches the tightest tolerance by update 800 and matches FH cost. Table 2 varies model size from 4–8 qubits and 4–9 layers, where FQ saves 48–58% versus SD. Quality is similar except for 8q/9L, where FQ has higher held-out MSE (0.1005 vs. 0.0845).

Table 3 shows that the effect is not specific to digits. On UNSW-NB15 and Waveform with the 6q/4L model, FQ saves 62–66% versus FH and 50–58% versus SD, while held-out accuracy remains within one SD of the deterministic arm. These are per-circuit costs; one B32 round of the main circuit executes $32 \times 217 = 6{,}944$ circuits, so per-round workload scales accordingly. Total cost to a gradient target also depends on the number of updates; Appendix D.3 reports target passages and cumulative T consumption.

Answer to RQ3: *scheduled mixed synthesis cuts the T cost per circuit by about 60% against fixed high precision and by about half against scheduled deterministic synthesis, without loss of accuracy.*

Table 3: Cross-dataset validation of T cost and held-out quality (6q/4L, B32/S500; update 800 for digits, 500 otherwise). Mean ± SD over three seeds. †Cost replay along SD.

		Digits	UNSW-NB15	Waveform
T/circuit ($\times 10^3$)	FH†	2.98 ± 0.03	2.97 ± 0.02	2.97 ± 0.02
	SD	2.41 ± 0.02	2.30 ± 0.19	2.42 ± 0.02
	FQ	1.01 ± 0.01	1.14 ± 0.03	1.09 ± 0.04
MSE ($\times 10^{-2}$)	SD	11.34 ± 13.89	48.18 ± 3.82	35.17 ± 1.83
	FQ	10.58 ± 13.55	48.82 ± 4.78	34.60 ± 2.08
Acc. (%)	SD	100.0 ± 0.0	90.9 ± 2.7	94.5 ± 0.8
	FQ	100.0 ± 0.0	91.9 ± 2.3	94.0 ± 0.5
FQ saving (%)	vs. FH	66.2	61.6	63.1
	vs. SD	58.2	50.5	54.7

RQ4 and RQ5: Which mechanism and fidelity should the learner request, and how should the server provision them? From update 800 of the main run, we train 300 additional updates under two policies, applying the exact injection channel at every T gate. The baseline uses scheduled mixed synthesis with fixed high-fidelity magic states at $p_{\mathrm{MS}} = 3 \times 10^{-8}$, while joint selection chooses the mechanism, tolerance, and fidelity within the shared bias budget. It retains mixed synthesis at $\epsilon = 0.003$ throughout, with fidelity tightening as the gradient decreases (Figure 6). This reduces fidelity-weighted demand by $13.24 \pm 0.63\%$, while changing T count by only $0.24 \pm 0.35\%$ and held-out MSE by $0.03 \pm 0.03\%$. The absence of a mechanism switch is consistent with Proposition 2, since this continuation begins late in training.

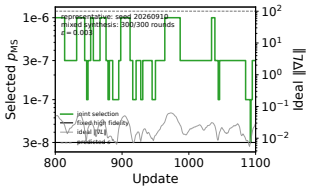


Figure 6: Magic-state fidelity chosen per update by joint selection (main circuit, B32/S500), with $\|\nabla L\|$ in gray.

Answer to RQ4: *Late in training, mixed synthesis remains preferable, while the requested magic-state fidelity can relax with the gradient.*

We then replay this demand through a conditional 15-to-1 distillation model. Each requested p_{MS} is mapped to the smallest surface-code distance that meets it for $p_{\mathrm{phys}} = 10^{-3}$ and exposure $V = 100$, giving distance 15 for the two loosest tiers and 17 for the two tightest. Compared with provisioning every state at the highest fidelity, scheduled provisioning reduces total factory work by $25.88 \pm 2.35\%$ and the sustained capacity required to avoid stalls by $25.34 \pm 2.11\%$, while peak capacity changes little ($0.45 \pm 0.58\%$). Across all recorded RQ3 traces (Table 4), the same rule saves 11–28% in work and sustained capacity, with peak savings below 2%. These are normalized factory metrics under the stated model, not physical-qubit counts or runtime; Appendix D.4 gives details and sensitivity analysis.

Answer to RQ5: *provisioning by the fidelity schedule cuts total factory work and sustained capacity by 11–28% on every trace, but not the peak.*

Table 4: Provisioning savings over fixed-high (primary scenario). Mean ± SD over 3 seeds; * joint selection.

	Saving (%)		
Trace	Work	Sust. cap.	Peak
Main circuit, 800–1,100*	25.88 ± 2.35	25.34 ± 2.11	0.45 ± 0.58
Noise floor (6 settings)	11.44–24.29	11.44–24.27	0.00–0.00
Model size (4 models)	12.32–28.20	12.11–28.17	0.00–0.10
UNSW-NB15	23.55 ± 3.87	24.14 ± 4.47	1.90 ± 2.83
Waveform	14.45 ± 7.88	15.77 ± 7.32	2.09 ± 1.83

6 Limitations and Conclusion

Limitations: Our experiments simulate logical circuits of up to 8 qubits, and the tolerance laws depend on constants that must be estimated at larger scales. The schedule uses an oracle gradient monitor, and the Adam case of the convergence theorem holds only with worst-case constants. Fixed-high and multi-trace provisioning results are replays of recorded training, and distillation costs come from a conditional model rather than calibrated hardware.

Conclusion: We asked how much fault tolerance a QNN needs to learn, and showed that the answer is set by the convergence state and by how synthesis error enters the gradient. Deterministic synthesis biases the gradient at first order, so its tolerance must shrink with the gradient; three-axis mixed synthesis reduces the bias to second order, which asymptotically halves the leading T cost at the convergence floor. In training, this lowers the T cost per circuit by about 60% against fixed high precision and by about half against scheduled deterministic synthesis without loss of accuracy, while scheduling magic-state fidelity lowers distillation work by 11–28%. These results show that fault-tolerant precision in QNN training should be derived from learning, not fixed by convention. As fault-tolerant hardware matures, treating precision as a learning-dependent resource can make QNN training substantially cheaper than fixed-accuracy execution suggests.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, organization, and grammatical suggestions during manuscript drafting. The authors independently developed and verified the research questions, methodology, theoretical results and proofs, experimental design, results, and conclusions, and reviewed all AI-assisted content for technical accuracy and consistency. The authors take full responsibility for the scientific content of the manuscript.

ETHICS STATEMENT

This work studies how fault-tolerant execution precision affects the training of quantum neural networks. All experiments use classical simulation of logical quantum circuits on established public datasets: a handwritten-digits benchmark, the UNSW-NB15 network-traffic dataset, and the Waveform dataset. The work does not involve human participants, collect personal or private data, or introduce a new dataset. The methods concern the resource cost of training on future fault-tolerant hardware and are not intended for deployment in high-stakes decision-making. We report the main limitations in Section 6: the circuit sizes studied, the use of an oracle gradient monitor, worst-case constants for Adam, cost replays for some baselines, and a conditional rather than hardware-calibrated distillation model. The authors are not aware of additional ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

We provide the details needed to reproduce our results in the main paper and the appendices. Appendix A lists all notation and abbreviations, and Appendix E explains the QNN model, parameter-shift gradients, and fault-tolerant execution. Appendix F describes rotation synthesis, the three-axis mixture construction, T-count accounting, the controller with its pseudocode, the simulated error channels, and the distillation-server model. Appendix C reports the datasets, preprocessing, encoding, architectures, compared methods, training hyperparameters, stopping rules, controller and server settings, seeds, simulators, software versions, and compute. Appendix B contains complete proofs of all theoretical results, and the remaining appendices report the full numerical results behind each research question. The implementation and scripts to reproduce all experiments are included in the supplementary material.

REFERENCES

Ahmad Ajalloeian and Sebastian U Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

E. Alpaydin and C. Kaynak. Optical Recognition of Handwritten Digits. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50P49.

Michael E Beverland, Prakash Murali, Matthias Troyer, Krysta M Svore, Torsten Hoefler, Vadym Kliuchnikov, Guang Hao Low, Mathias Soeken, Aarthi Sundaram, and Alexander Vaschillo. Assessing requirements to scale to practical quantum advantage. *arXiv preprint arXiv:2211.07629*, 2022.

Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Physical Review A—Atomic, Molecular, and Optical Physics*, 71(2):022316, 2005.

L. Breiman and C.J. Stone. Waveform Database Generator (Version 1). UCI Machine Learning Repository, 1984. DOI: https://doi.org/10.24432/C5CS3C.

Earl Campbell. Shorter gate sequences for quantum computing by mixing unitaries. *Physical Review A*, 95(4):042306, 2017.

Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of adam and adagrad. *arXiv preprint arXiv:2003.02395*, 2020.

Yury Demidovich, Grigory Malinovsky, Igor Sokolov, and Peter Richtárik. A guide through the zoo of biased sgd. *Advances in Neural Information Processing Systems*, 36:23158–23171, 2023.

Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A—Atomic, Molecular, and Optical Physics*, 86(3):032324, 2012.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International conference on machine learning*, pp. 1737–1746. PMLR, 2015.

Matthew B. Hastings. Turning gate synthesis errors into incoherent errors. *Quantum Info. Comput.*, 17(5–6):488–494, March 2017. ISSN 1533-7146.

Yifan Hu, Xin Chen, and Niao He. On the bias-variance-cost tradeoff of stochastic optimization. *Advances in Neural Information Processing Systems*, 34:22119–22131, 2021.

Haiyue Kang, Younghun Kim, Eromanga Adermann, Martin Sevior, and Muhammad Usman. Almost fault-tolerant quantum machine learning with drastic overhead reduction. *Quantum Science and Technology*, 11(1):015021, 2026.

Vadym Kliuchnikov, Kristin Lauter, Romy Minko, Adam Paetznick, and Christophe Petit. Shorter quantum circuits via single-qubit gate approximation. *Quantum*, 7:1208, 2023.

Daniel Litinski. Magic state distillation: Not as costly as you think. *Quantum*, 3:205, 2019.

John M Martyn and Patrick Rall. Halving the cost of quantum algorithms with randomization. *npj Quantum Information*, 11(1):47, 2025.

Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pp. 1–6. Ieee, 2015.

Joe O'Gorman and Earl T Campbell. Quantum computation with realistic magic-state factories. *Physical Review A*, 95(3):032338, 2017.

Daniil Rabinovich, Ernesto Campos, Soumik Adhikary, Ekaterina Pankovets, Dmitry Vinichenko, and Jacob Biamonte. Robustness of variational quantum algorithms against stochastic parameter perturbation. *Physical Review A*, 109(4):042426, 2024.

Neil J Ross and Peter Selinger. Optimal ancilla-free clifford+ t approximation of z-rotations. *Quantum Inf. Comput.*, 16(11&12):901–953, 2016.

Hasan Sayginel, Francois Jamet, Abhishek Agarwal, Dan E Browne, and Ivan Rungger. A fault-tolerant variational quantum algorithm with limited t-depth. *Quantum Science and Technology*, 9 (1):015015, 2024.

Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

Hanrui Wang, Jiaqi Gu, Yongshan Ding, Zirui Li, Frederic T Chong, David Z Pan, and Song Han. Quantumnat: quantum noise-aware training with noise injection, quantization and normalization. In *Proceedings of the 59th ACM/IEEE design automation conference*, pp. 1–6, 2022.

Nobuyuki Yoshioka, Seiseki Akibue, Hayata Morisaki, Kento Tsubouchi, and Yasunari Suzuki. Error crafting in mixed quantum gate synthesis. *npj Quantum Information*, 11(1):95, 2025.

APPENDIX

A NOTATION AND ABBREVIATIONS

This section collects the abbreviations (Table 5) and symbols (Table 6) used throughout the paper, grouped by the part of the method in which they first appear. Abbreviations of method names follow Table 7: FH and SD denote the deterministic reference configurations, and FQ and PS denote the two FT-QNN mixed-synthesis policies. Experimental settings are written compactly: for example, B32/S500 denotes a batch of 32 examples with 500 shots per circuit query, and 6q/9L denotes a circuit with 6 logical qubits and 9 layers.

Table 5: Abbreviations.

Abbreviation	Meaning
QNN	Quantum neural network
NISQ	Noisy intermediate-scale quantum
FT	Fault-tolerant
FT-QNN	Our convergence-aware framework
T gate	Non-Clifford gate $\mathrm{diag}(1, e^{i\pi/4})$
FH	Fixed-high deterministic synthesis (reference)
SD	Scheduled deterministic synthesis (reference)
FQ	Fixed-query three-axis mixed synthesis (ours)
PS	Per-shot three-axis mixed synthesis (ours)
PTM	Pauli transfer matrix
LP	Linear program
MSE	Mean-squared error
Bb/Ss	Batch size b, s shots per circuit query
nq/LL	Circuit with n logical qubits and L layers

We use the following conventions. Unless stated otherwise, $\|\cdot\|$ is the Euclidean norm for vectors and the spectral norm for operators and matrices, and $\|\cdot\|_2$ for unitaries denotes the operator norm used to define synthesis tolerance. Expectations $\mathbb{E}$ are over all randomness in an update, including mini-batch selection, measurement shots, and mixed-synthesis draws, and $\mathbb{E}[\cdot \mid \mathcal{F}_t]$ conditions on the history up to update t. The notation $O(\cdot)$ hides constants that do not depend on the synthesis tolerance ϵ or the target Δ, and $\Theta(\cdot)$ denotes matching upper and lower bounds. Logarithms of tolerances are base 10, so T-count slopes are per decade of tolerance. A T count always refers to logical T gates, each of which consumes one magic state.

B SUPPORTING THEORY AND PROOFS FOR SECTION 4

Question (RQ1). How do fault-tolerant synthesis errors distort the gradient that the optimizer receives?

Answer. Deterministic synthesis shifts the mean gradient at first order in the tolerance. Three-axis mixed synthesis removes this first-order shift: the mean is perturbed only at second order, and the first-order error reappears as variance. Drawing a new candidate for every shot divides this variance by the shot count and leaves the mean unchanged. Section B.1 fixes the estimator, and Sections B.2 and B.3 prove Lemmas 1 and 2.

B.1 THE GRADIENT ESTIMATOR

Setup. The training set is $\{(x_i, y_i)\}_{i=1}^n$ with labels ± 1. Each trainable parameter enters one Pauli rotation, so the parameter-shift rule is exact for ideal circuits:

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} \big(f_\theta(x_i) - y_i\big)^2, \tag{4}$$

$$\partial_k f_\theta(x) = \tfrac{1}{2}\big[f_{\theta+\frac{\pi}{2}e_k}(x) - f_{\theta-\frac{\pi}{2}e_k}(x)\big]. \tag{5}$$

Table 6: Notation.

Symbol	Meaning
Model and training	
$\theta \in \mathbb{R}^P$	Trainable parameters, P of them
$U(\theta)$, O	Parameterized circuit and measured observable
$f_\theta(x)$, $L(\theta)$	Prediction and mean-squared loss
B, S	Batch size and shots per circuit query
$\hat{g}_t$, b_t, ξ_t	Gradient estimate, its bias, its zero-mean noise
$\mathcal{F}_t$	History up to update t
ℓ, D	Smoothness constant and initial loss gap
η, T	Step size and number of updates
Δ	Target stationarity scale
Synthesis error	
ϵ	Synthesis tolerance (operator distance)
ϵ_{loc}	Largest tolerance for which the local expansion holds
E_j, $a_j \in \mathbb{R}^3$	Error unitary of rotation j and its generator
$R_j \in \mathbb{R}^{P\times 3}$	First-order gradient response of rotation j
$\bar{G}(\theta, a)$	Gradient estimator averaged over shots and batches, errors fixed
C_j	Second moment of the mixed error at rotation j
Σ_{FT}	Synthesis covariance of the gradient
χ_S	Resampling factor: 1 (FQ) or $1/S$ (PS)
$\Lambda_1, \Lambda_2, \Lambda_v$	Deterministic bias, mixed bias, and variance constants
Budgets and schedule	
s_t	Convergence level, $\max\{\|\nabla L(\theta_t)\|, g_\Delta\}$
$g_\Delta = \phi\Delta$	Schedule floor
ρ	Bias budget fraction
σ^2	Intrinsic shot and mini-batch variance bound
$v_t = \lambda\sigma^2 + \mu s_t^2$	Synthesis variance budget
ν_t	Bias from non-synthesis FT error
s_c, ϵ_v	Variance–bias crossover level and variance-limited tolerance
Magic states and cost	
p_{MS}	Logical infidelity of an injected magic state
R_{MS}	Magic-state gradient-response bound
p_{dep}, R_{dep}	Residual logical depolarizing rate and its response bound
$n_m(\epsilon)$	Expected T count per rotation for mechanism m
κ_m, β_m, κ_-	Cost slope, offset, and lower-bound slope
W_t, M_t, $C_T(T)$	Pre-shot, executed, and cumulative T workload
p_{phys}, p_L, d	Physical error rate, logical error per cycle, code distance
p_{in}, V	Raw magic-state error and exposure in the distillation model
Optimizers	
β, β_2, ε	Adam momentum, second-moment decay, and stabilizer
P_t, κ	Adam scaling and its conditioning
ℓ_β, ℓ_A, C_A	Effective smoothness (momentum, Adam) and Adam drift constant

For each example, the update evaluates the base query $q = 0$ and the shifted queries $q = (k, \pm)$. Each query is compiled independently. Let $F_{i,q}(a)$ be the expectation of the parity observable O in the compiled circuit, where a collects the error generators of all rotation instances. We index a over the queries of every example; queries of unselected examples are never executed. Each shot returns ± 1. Under fixed-query synthesis, a query is estimated from S shots with one frozen draw, and the gradient estimate is

$$\hat{g}_k = \frac{2}{B}\sum_{i\in I} X_i Y_{ik}, \qquad X_i = \hat{F}_{i,0} - y_i, \qquad Y_{ik} = \tfrac{1}{2}\big(\hat{F}_{i,(k,+)} - \hat{F}_{i,(k,-)}\big). \tag{6}$$

The mini-batch I is sampled uniformly and independently of the draws and shots. Each $F_{i,q}$ is an entire function of a, so the map $\bar{G}$ below is real-analytic in (θ, a) and 2π-periodic in θ. The smoothness in Assumption 1 therefore holds, and every derivative bound below is uniform in θ.

We use the following quantities, with $\mathrm{Var}(v) = \mathbb{E}\|v - \mathbb{E}v\|^2$:

$$G(\theta, a, I) = \mathbb{E}[\hat{g} \mid a, I], \qquad \bar{G}(\theta, a) = \mathbb{E}_I\, G(\theta, a, I), \tag{7}$$

$$v(a) = \mathbb{E}\big[\|\hat{g} - \bar{G}(\theta, a)\|^2 \,\big|\, a\big], \qquad \sigma^2 = \sup_{a \in \mathcal{B}} v(a), \tag{8}$$

$$\Sigma_{\mathrm{FT}} = \mathrm{Cov}_a\big(\bar{G}(\theta, a)\big), \tag{9}$$

where $\mathcal{B}$ is the set of errors with every block bounded by $c_a \epsilon_{\mathrm{loc}}$.

***Lemma* 3** (Estimator properties)**.** *Under Assumption 1:*

- *(a) A rotation synthesized to tolerance ϵ has $2\delta_j \le \|a_j\| \le c_a \epsilon$ with $c_a = \pi/\sqrt{2}$, where δ_j is its realized operator distance.*
- *(b) $\bar{G}(\theta, 0) = \nabla L(\theta)$, and fixed-query and per-shot synthesis give the same mean $\mathbb{E}\hat{g}$.*
- *(c) $\mathrm{Var}(\hat{g}) \le \sigma^2 + \chi_S \,\mathrm{tr}\,\Sigma_{\mathrm{FT}}$, with $\chi_S = 1$ for fixed-query and $\chi_S = 1/S$ for per-shot synthesis.*

Proof. (a) Let $E = \widetilde{U} U^\dagger = \exp[-i(a \cdot \sigma)/2]$, with the global phase chosen so that $\varphi = \|a\| \le \pi$. The eigenvalues of E are $e^{\pm i\varphi/2}$, which are mirror images across the real axis. The larger distance from $e^{i\alpha}$ to them is smallest when $e^{i\alpha} = \pm 1$, and $e^{i\alpha} = 1$ is the better choice since $\varphi \le \pi$. Using unitary invariance of the spectral norm,

$$\delta = \min_\alpha \|\widetilde{U} - e^{i\alpha} U\|_2 = \min_\alpha \|E - e^{i\alpha} I\|_2 = |e^{i\varphi/2} - 1| = 2\sin(\varphi/4). \tag{10}$$

Since $\sin x \le x$, we get $\delta \le \varphi/2$. Since $\sin$ is concave on $[0, \pi/4]$, it lies above its chord, $\sin x \ge (2\sqrt{2}/\pi)x$, so $\delta \ge (\sqrt{2}/\pi)\varphi$. Rearranging with $\delta \le \epsilon$ gives both bounds.

(b) Given a and I, the shots of different queries are independent, and each query estimate is unbiased for $F_{i,q}(a)$. The conditional mean of each product in equation 6 is therefore the product of the conditional means. Averaging over I then gives

$$\bar{G}_k(\theta, a) = \frac{2}{n} \sum_{i=1}^{n} \big(F_{i,0}(a) - y_i\big)\, \tfrac{1}{2}\big(F_{i,(k,+)}(a) - F_{i,(k,-)}(a)\big). \tag{11}$$

At $a = 0$, every compiled circuit is ideal, and the parameter-shift rule gives

$$\bar{G}_k(\theta, 0) = \frac{2}{n} \sum_{i=1}^{n} \big(f_\theta(x_i) - y_i\big)\, \partial_k f_\theta(x_i) = \partial_k L(\theta). \tag{12}$$

Now let $\bar{F}_{i,q} = \mathbb{E}_a F_{i,q}(a)$. The two factors in equation 11 depend on disjoint, independent blocks of a. Under fixed-query synthesis, the expectation therefore factorizes:

$$\mathbb{E}\hat{g}_k = \mathbb{E}_a \bar{G}_k(\theta, a) = \frac{2}{n} \sum_{i=1}^{n} \big(\bar{F}_{i,0} - y_i\big)\, \tfrac{1}{2}\big(\bar{F}_{i,(k,+)} - \bar{F}_{i,(k,-)}\big). \tag{13}$$

Under per-shot synthesis, every shot draws fresh errors, so each query estimate has mean $\bar{F}_{i,q}$. Queries remain independent, so the mean of $\hat{g}$ is again equation 13.

(c), fixed-query. Conditioning on a and applying the law of total variance,

$$\mathrm{Var}(\hat{g}) = \mathbb{E}_a\, \mathbb{E}\big[\|\hat{g} - \bar{G}(\theta, a)\|^2 \,\big|\, a\big] + \mathbb{E}_a \big\|\bar{G}(\theta, a) - \mathbb{E}_a \bar{G}\big\|^2 \tag{14}$$

$$= \mathbb{E}_a\, v(a) + \mathrm{tr}\,\Sigma_{\mathrm{FT}} \;\le\; \sigma^2 + \mathrm{tr}\,\Sigma_{\mathrm{FT}}. \tag{15}$$

(c), per-shot. We compare the two granularities query by query and then pass the comparison through the estimator. Suppress the example index and define the averaged shot variance and the synthesis variance of a query,

$$s_q = \frac{\mathbb{E}_a\big[1 - F_q(a)^2\big]}{S}, \qquad V_q = \mathrm{Var}_a\, F_q(a). \tag{16}$$

Under fixed-query synthesis, the law of total variance gives

$$\operatorname{Var} \hat{F}_q = \mathbb{E}_a \frac{1 - F_q(a)^2}{S} + \operatorname{Var}_a F_q(a) = s_q + V_q. \tag{17}$$

Under per-shot synthesis, the S outcomes are independent with mean $\bar{F}_q$. Using $1 - \bar{F}_q^2 = \mathbb{E}_a[1 - F_q^2] + V_q$,

$$\operatorname{Var} \hat{F}_q = \frac{1 - \bar{F}_q^2}{S} = \frac{\mathbb{E}_a[1 - F_q^2] + V_q}{S} = s_q + \frac{V_q}{S}. \tag{18}$$

The shot part is the same in both modes, and only the synthesis part is divided by S.

Next, condition on I. The terms of equation 6 for different examples are independent, and X_i is independent of Y_{ik}. For independent X and Y,

$$\operatorname{Var}(XY) = \operatorname{Var} X \operatorname{Var} Y + \operatorname{Var} X \, (\mathbb{E}Y)^2 + \operatorname{Var} Y \, (\mathbb{E}X)^2, \tag{19}$$

$$\operatorname{Var} Y_{ik} = \tfrac{1}{4}\big(\operatorname{Var} \hat{F}_{i,(k,+)} + \operatorname{Var} \hat{F}_{i,(k,-)} \big). \tag{20}$$

By (b), the means are the same in both modes. Hence $\operatorname{Var}(\hat{g} \mid I) = \Phi_I(w)$, where w is the vector of query variances and Φ_I is a polynomial with nonnegative coefficients whose monomials are linear in w or products of two entries of w. For every such monomial and nonnegative s, V,

$$\Big(s_x + \frac{V_x}{S}\Big)\Big(s_y + \frac{V_y}{S}\Big) - s_x s_y = \frac{s_x V_y + s_y V_x}{S} + \frac{V_x V_y}{S^2} \tag{21}$$

$$\leq \frac{1}{S}\Big[(s_x + V_x)(s_y + V_y) - s_x s_y\Big], \tag{22}$$

and linear monomials satisfy the same inequality with equality. Summing over monomials,

$$\Phi_I\Big(s + \frac{V}{S}\Big) - \Phi_I(s) \leq \frac{1}{S}\Big[\Phi_I(s + V) - \Phi_I(s)\Big]. \tag{23}$$

Define the reference variance, in which synthesis fluctuations are removed but the means are kept:

$$\mathrm{Ref} = \mathbb{E}_I \, \Phi_I(s) + \operatorname{Var}_I \, \mathbb{E}[\hat{g} \mid I]. \tag{24}$$

We show that $\mathrm{Ref} \leq \sigma^2$. Given a, a product term has shot variance $s_X(a)s_Y(a) + s_X(a)m_Y(a)^2 + s_Y(a)m_X(a)^2$, where s and m are the shot variances and means at a. The blocks of a entering X and Y are independent, and $\mathbb{E}_a[m^2] \geq (\mathbb{E}_a m)^2$. Therefore

$$\begin{aligned} \mathbb{E}_a\big[s_X s_Y + s_X m_Y^2 + s_Y m_X^2\big] &= \bar{s}_X \bar{s}_Y + \bar{s}_X \, \mathbb{E}_a[m_Y^2] + \bar{s}_Y \, \mathbb{E}_a[m_X^2] \\ &\geq \bar{s}_X \bar{s}_Y + \bar{s}_X \, \bar{m}_Y^2 + \bar{s}_Y \, \bar{m}_X^2, \end{aligned} \tag{25}$$

and summing over terms gives $\Phi_I(s) \leq \mathbb{E}_a \operatorname{Var}_\omega(\hat{g} \mid a, I)$. For the batch term, convexity of the variance gives

$$\operatorname{Var}_I \, \mathbb{E}[\hat{g} \mid I] = \operatorname{Var}_I \, \mathbb{E}_a G(\theta, a, I) \leq \mathbb{E}_a \operatorname{Var}_I G(\theta, a, I). \tag{26}$$

Adding the two bounds and using the law of total variance for $v(a)$,

$$\mathrm{Ref} \leq \mathbb{E}_a\Big[\mathbb{E}_I \operatorname{Var}_\omega(\hat{g} \mid a, I) + \operatorname{Var}_I G(\theta, a, I)\Big] = \mathbb{E}_a \, v(a) \leq \sigma^2. \tag{27}$$

Finally, the law of total variance over I gives both variances in terms of Ref. Using equation 23, then $\mathrm{Ref} \leq \sigma^2$ and equation 15,

$$\begin{aligned} \operatorname{Var}_{\mathrm{ps}}(\hat{g}) &= \mathrm{Ref} + \mathbb{E}_I\Big[\Phi_I\Big(s + \frac{V}{S}\Big) - \Phi_I(s)\Big] \\ &\leq \mathrm{Ref} + \frac{1}{S}\,\mathbb{E}_I\Big[\Phi_I(s + V) - \Phi_I(s)\Big] \\ &= \Big(1 - \frac{1}{S}\Big)\mathrm{Ref} + \frac{1}{S}\operatorname{Var}_{\mathrm{fq}}(\hat{g}) \\ &\leq \Big(1 - \frac{1}{S}\Big)\sigma^2 + \frac{1}{S}\Big(\sigma^2 + \operatorname{tr}\Sigma_{\mathrm{FT}}\Big) = \sigma^2 + \frac{1}{S}\operatorname{tr}\Sigma_{\mathrm{FT}}. \end{aligned} \tag{28}$$

□

Scope of the convention. The term $v(a)$ contains the mini-batch variance at frozen errors, so σ^2 exceeds its ideal-circuit value by at most $O(\epsilon_{\mathrm{loc}})$. In exchange, Σ_{FT} is exactly the covariance expanded in Lemma 2.

Which objective is analyzed. Lemmas 1 and 2 bound the distance from the mean equation 13 to ∇L. This mean is generally not the gradient of the MSE of the channel-averaged prediction, because the circuits at $\theta_k \pm \pi/2$ are separate compilations and the shift rule does not transfer to them. If the base and shifted queries shared one frozen draw, the mean would acquire a covariance term:

$$\mathbb{E}_a\Big[(F_{i,0} - y_i)\,\tfrac{1}{2}\big(F_{i,(k,+)} - F_{i,(k,-)}\big)\Big] = (\bar{F}_{i,0} - y_i)\,\tfrac{1}{2}\big(\bar{F}_{i,(k,+)} - \bar{F}_{i,(k,-)}\big) + \tfrac{1}{2}\,\mathrm{Cov}_a\left(F_{i,0},\, F_{i,(k,+)} - F_{i,(k,-)}\right). \tag{29}$$

Each factor fluctuates at $O(\epsilon)$, so this term is $O(\epsilon^2)$, the same order as the mixed bias. This accounts for the difference between the frozen-error and mean-channel evaluations in our response study. Training compiles queries independently and therefore targets equation 13.

B.2 Proof of Lemma 1

Errors are controlled gate by gate, so we measure a by its largest block and multilinear maps by the matching operator norm:

$$|||a||| = \max_j \|a_j\|, \tag{30}$$

$$\|A\|_\star = \sup\big\{\|A[u_1,\dots,u_k]\| :\ |||u_1|||,\dots,|||u_k||| \le 1\big\}. \tag{31}$$

We use the following constants, all finite and uniform in θ:

$$M_1^0 = \|D_a\bar{G}(\theta,0)\|_\star \le \sum_j \|R_j\|_2,\ M_2^0 = \|D_a^2\bar{G}(\theta,0)\|_\star,\ M_k = \sup_{a\in\mathcal{B}} \|D_a^k\bar{G}(\theta,a)\|_\star \quad (k=2,3). \tag{32}$$

Proof of Lemma 1. For deterministic synthesis, a is fixed. By Lemma 3(b), the bias is $\bar{G}(\theta,a) - \bar{G}(\theta,0)$. The set $\mathcal{B}$ is convex, so Taylor's theorem with integral remainder along the segment from 0 to a gives

$$b_{\mathrm{det}}^{\mathrm{syn}} = \bar{G}(\theta,a) - \bar{G}(\theta,0) = \sum_{j=1}^{N} R_j a_j + r_1(a), \qquad r_1(a) = \int_0^1 (1-\tau)\, D_a^2\bar{G}(\theta,\tau a)[a,a]\, d\tau. \tag{33}$$

This is the stated expansion. By Lemma 3(a), $|||a||| \le c_a\epsilon$. For $\epsilon \le \epsilon_{\mathrm{loc}}$,

$$\begin{aligned}\|b_{\mathrm{det}}^{\mathrm{syn}}\| &\le \Big\|\sum_j R_j a_j\Big\| + \|r_1(a)\| \\ &\le M_1^0\,|||a||| + \tfrac{1}{2}M_2\,|||a|||^2 \\ &\le c_a M_1^0\,\epsilon + \tfrac{1}{2}c_a^2 M_2\,\epsilon_{\mathrm{loc}}\,\epsilon = \Lambda_1\epsilon,\end{aligned} \tag{34}$$

where $\Lambda_1 = c_a M_1^0 + \frac{1}{2}c_a^2 M_2\epsilon_{\mathrm{loc}}$. No independence is used, so the result also covers repeated instances of the same angle, which receive the same deterministic error. □

B.3 Proof of Lemma 2

We expand one order further than in Lemma 1 and take expectations. Centering removes the linear term from the mean, and independence removes the cross-gate second-order terms. The linear term then reappears in the covariance.

Proof of Lemma 2. Bias. Taylor's theorem to second order, with a third-order remainder, gives

$$\bar{G}(\theta,a) = \bar{G}(\theta,0) + \sum_j R_j a_j + \tfrac{1}{2}\sum_{j,l} D_{jl}^2\bar{G}(\theta,0)[a_j,a_l] + r_2(a), \tag{35}$$

$$\|r_2(a)\| \le \tfrac{1}{6}M_3\,|||a|||^3 \le \tfrac{1}{6}c_a^3 M_3\,\epsilon^3. \tag{36}$$

We take expectations term by term. The linear term vanishes because the mean of a_j is zero in all three components. For $j \neq l$, independence and zero means give

$$\mathbb{E}\, D^2_{jl}\bar{G}[a_j, a_l] = \sum_{r,s} \partial_{a_{j,r}}\partial_{a_{l,s}}\bar{G}\, \mathbb{E}[a_{j,r}]\, \mathbb{E}[a_{l,s}] = 0. \tag{37}$$

For $j = l$, the entries of C_j are the second moments $\mathbb{E}[a_{j,r}a_{j,s}]$, so

$$\mathbb{E}\, D^2_{jj}\bar{G}[a_j, a_j] = \sum_{r,s} (C_j)_{rs}\, \partial_{a_{j,r}}\partial_{a_{j,s}}\bar{G} = D^2_{jj}\bar{G}[C_j]. \tag{38}$$

Combining these with Lemma 3(b),

$$b^{\mathrm{syn}}_{\mathrm{mix}} = \mathbb{E}_a\bar{G}(\theta, a) - \bar{G}(\theta, 0) = \tfrac{1}{2}\sum_j D^2_{jj}\bar{G}(\theta, 0)[C_j] + \mathbb{E}\, r_2(a). \tag{39}$$

The diagonal sum equals the expectation of the full quadratic form, which gives the norm bound

$$\begin{aligned} \|b^{\mathrm{syn}}_{\mathrm{mix}}\| &\leq \tfrac{1}{2}\big\|\mathbb{E}\, D^2_a\bar{G}(\theta, 0)[a, a]\big\| + \tfrac{1}{6}c_a^3 M_3\, \epsilon^3 \\ &\leq \tfrac{1}{2}M_2^0 c_a^2\, \epsilon^2 + \tfrac{1}{6}c_a^3 M_3\, \epsilon_{\mathrm{loc}}\, \epsilon^2 = \Lambda_2\epsilon^2. \end{aligned} \tag{40}$$

Covariance. Since the linear term has mean zero, the centered gradient splits into a linear part and a remainder, with pointwise bounds:

$$\bar{G}(\theta, a) - \mathbb{E}_a\bar{G} = \mathcal{L}(a) + \mathcal{W}(a), \tag{41}$$

$$\mathcal{L}(a) = \sum_j R_j a_j, \qquad \|\mathcal{L}(a)\| \leq c_a M_1^0\, \epsilon, \tag{42}$$

$$\mathcal{W}(a) = r_1(a) - \mathbb{E}\, r_1(a), \qquad \|\mathcal{W}(a)\| \leq c_a^2 M_2\, \epsilon^2. \tag{43}$$

Independence and zero means make the linear part contribute only diagonal blocks:

$$\mathbb{E}[\mathcal{L}\mathcal{L}^\top] = \sum_{j,l} R_j\, \mathbb{E}[a_j a_l^\top]\, R_l^\top = \sum_j R_j C_j R_j^\top. \tag{44}$$

The remaining terms are bounded pointwise, so

$$\begin{aligned} \Big\|\Sigma_{\mathrm{FT}} - \sum_j R_j C_j R_j^\top\Big\|_2 &= \big\|\mathbb{E}[\mathcal{L}\mathcal{W}^\top + \mathcal{W}\mathcal{L}^\top + \mathcal{W}\mathcal{W}^\top]\big\|_2 \\ &\leq 2c_a^3 M_1^0 M_2\, \epsilon^3 + c_a^4 M_2^2\, \epsilon^4 = O(\epsilon^3). \end{aligned} \tag{45}$$

For the trace, the same decomposition and $\mathbb{E}\|R_j a_j\|^2 \leq \|R_j\|_2^2 c_a^2\epsilon^2$ give, for $\epsilon \leq \epsilon_{\mathrm{loc}}$,

$$\begin{aligned} \operatorname{tr}\Sigma_{\mathrm{FT}} &= \mathbb{E}\|\mathcal{L}\|^2 + 2\,\mathbb{E}\langle\mathcal{L}, \mathcal{W}\rangle + \mathbb{E}\|\mathcal{W}\|^2 \\ &\leq c_a^2\epsilon^2 \sum_j \|R_j\|_2^2 + 2c_a^3 M_1^0 M_2\, \epsilon^3 + c_a^4 M_2^2\, \epsilon^4 \\ &\leq \Lambda_v\epsilon^2, \end{aligned} \tag{46}$$

where $\Lambda_v = c_a^2 \sum_j \|R_j\|_2^2 + 2c_a^3 M_1^0 M_2\epsilon_{\mathrm{loc}} + c_a^4 M_2^2\epsilon_{\mathrm{loc}}^2$. □

Remarks. The leading variance constant is a sum of squares because independent errors add in quadrature. If draws were correlated but still centered, the bias bound would be unchanged, since it used only the norm of the full quadratic form. The covariance, however, would become $D_a\bar{G}\, C\, D_a\bar{G}^\top$ with the joint covariance C, and could grow toward the coherent value $(c_a M_1^0\epsilon)^2$. If only the tangent component is centered, the mean of a_j keeps its transverse part, and the linear term $\sum_j R_j\mathbb{E}[a_j]$ remains first order. This matches the unit slope measured for the tangent-only construction.

B.4 Proof of Proposition 1

Question (RQ2). How much synthesis error can an update tolerate, and how does this depend on the mechanism?

Answer. Deterministic tolerance must shrink linearly with the available bias budget. Three-axis mixed tolerance shrinks only as its square root, until the variance budget becomes the tighter constraint.

Throughout, ν_t bounds the part of the bias that does not come from synthesis, including cross terms between synthesis and other error sources:

$$\|b_t - b_t^{\mathrm{syn}}\| \le \nu_t. \tag{47}$$

Proof of Proposition 1. Both tolerances are capped at ϵ_{loc}, so Lemmas 1 and 2 apply.

Deterministic. Take $\epsilon_t^{\mathrm{det}} \le (\rho s_t - \nu_t)/\Lambda_1$. By Lemma 1,

$$\|b_t\| \le \|b_t^{\mathrm{syn}}\| + \nu_t \le \Lambda_1 \epsilon_t^{\mathrm{det}} + \nu_t \le (\rho s_t - \nu_t) + \nu_t = \rho s_t. \tag{48}$$

Deterministic synthesis adds no synthesis variance, so the variance budget holds trivially.

Mixed, bias. Take $\epsilon_t^{\mathrm{mix}} \le \sqrt{(\rho s_t - \nu_t)/\Lambda_2}$. By Lemma 2,

$$\|b_t\| \le \Lambda_2 \big(\epsilon_t^{\mathrm{mix}}\big)^2 + \nu_t \le (\rho s_t - \nu_t) + \nu_t = \rho s_t. \tag{49}$$

Mixed, variance. Take $\epsilon_t^{\mathrm{mix}} \le \sqrt{v_t/(\chi_S \Lambda_v)}$. By Lemma 3(c) and Lemma 2, the synthesis contribution to the variance satisfies

$$\chi_S \operatorname{tr} \Sigma_{\mathrm{FT}} \le \chi_S \Lambda_v \big(\epsilon_t^{\mathrm{mix}}\big)^2 \le v_t, \tag{50}$$

and hence the total variance satisfies

$$\mathbb{E}\big[\|\xi_t\|^2 \mid \mathcal{F}_t\big] \le \sigma^2 + v_t. \tag{51}$$

The minimum of the three mixed caps satisfies both budgets at once. □

Crossover and floor. Since $v_t \ge \lambda \sigma^2$, the variance cap is at least ϵ_v. At the crossover, the bias branch equals ϵ_v:

$$\sqrt{\frac{\rho s_c - \nu_t}{\Lambda_2}} = \epsilon_v \quad \Longleftrightarrow \quad s_c = \frac{\Lambda_2 \epsilon_v^2 + \nu_t}{\rho}. \tag{52}$$

The bias branch increases with s_t. It therefore exceeds ϵ_v for $s_t > s_c$, where the variance cap binds, and falls below it for $s_t < s_c$, where the bias cap binds. Both bias branches require $\rho s_t - \nu_t > 0$. At the floor $s_t = \phi\Delta$, this gives the condition $\bar{\nu} < \rho\phi\Delta$.

B.5 Proof of Theorem 1

Question (RQ3). What does each mechanism cost per rotation once training reaches the target precision?

Answer. At the convergence floor, deterministic synthesis needs about $\kappa \log_{10}(1/\Delta)$ T gates per generic rotation. Three-axis mixed synthesis in the bias-limited regime needs about half of that. Both statements carry matching upper and lower bounds.

Write $\mathsf{L} = \log_{10}(1/\Delta)$. At the floor, $s_t = \phi\Delta$, and the hypothesis $\bar{\nu} \le \frac{1}{2}\rho\phi\Delta$ gives

$$\tfrac{1}{2}\rho\phi\Delta \;\le\; \rho\phi\Delta - \nu_t \;\le\; \rho\phi\Delta. \tag{53}$$

Proof of Theorem 1. Deterministic lower bound. Consider any deterministic schedule that meets the bias budget at the floor. Its synthesis bias satisfies

$$\|b_{\mathrm{det}}^{\mathrm{syn}}\| \le \|b_t\| + \nu_t \le \rho\phi\Delta + \tfrac{1}{2}\rho\phi\Delta = \tfrac{3}{2}\rho\phi\Delta. \tag{54}$$

By Assumption 2, this bias controls every generic error: $\gamma \max_{j \in \mathcal{G}} \|a_j\| \le \frac{3}{2}\rho\phi\Delta$. By Lemma 3(a), $\|a_j\| \ge 2\delta_j$. Hence, for every generic rotation,

$$\delta_j \le \frac{\|a_j\|}{2} \le \frac{3\rho\phi}{4\gamma}\,\Delta. \tag{55}$$

The generic-angle bound then gives

$$\mathrm{T}_j \geq \kappa_- \log_{10} \frac{1}{\delta_j} - c_0 \tag{56}$$

$$\geq \kappa_- \mathsf{L} - \kappa_- \log_{10} \frac{3\rho\phi}{4\gamma} - c_0 \tag{57}$$

$$= \kappa_- \mathsf{L} - O(1). \tag{58}$$

Deterministic upper bound. By equation 53, the prescribed tolerance satisfies $\epsilon^{\mathrm{det}} \geq \rho\phi\Delta/(2\Lambda_1)$. The cost model gives

$$n_{\mathrm{det}}\big(\epsilon^{\mathrm{det}}\big) \leq \kappa_{\mathrm{det}} \log_{10} \frac{1}{\epsilon^{\mathrm{det}}} + \beta_{\mathrm{det}} \tag{59}$$

$$\leq \kappa_{\mathrm{det}} \mathsf{L} + \kappa_{\mathrm{det}} \log_{10} \frac{2\Lambda_1}{\rho\phi} + \beta_{\mathrm{det}} \tag{60}$$

$$= \kappa_{\mathrm{det}} \mathsf{L} + O(1). \tag{61}$$

Mixed upper bound. In the bias-limited regime, the bias cap binds. By equation 53, $\epsilon^{\mathrm{mix}} \geq \sqrt{\rho\phi\Delta/(2\Lambda_2)}$, so

$$n_{\mathrm{mix}}\big(\epsilon^{\mathrm{mix}}\big) \leq \kappa_{\mathrm{mix}} \log_{10} \frac{1}{\epsilon^{\mathrm{mix}}} + \beta_{\mathrm{mix}} \tag{62}$$

$$\leq \tfrac{1}{2}\kappa_{\mathrm{mix}} \mathsf{L} + \tfrac{1}{2}\kappa_{\mathrm{mix}} \log_{10} \frac{2\Lambda_2}{\rho\phi} + \beta_{\mathrm{mix}} \tag{63}$$

$$= \tfrac{1}{2}\kappa_{\mathrm{mix}} \mathsf{L} + O(1). \tag{64}$$

Mixed lower bound. Every candidate in the mixture synthesizes the same generic target angle. Its operator distance δ is therefore at most the tolerance, and Clifford conjugation changes neither the distance nor the T count. The tolerance is at most the bias cap, so by equation 53,

$$\delta \leq \epsilon^{\mathrm{mix}} \leq \sqrt{\frac{\rho\phi\Delta}{\Lambda_2}}. \tag{65}$$

Each candidate therefore satisfies

$$\mathrm{T} \geq \kappa_- \log_{10} \frac{1}{\delta} - c_0 \tag{66}$$

$$\geq \tfrac{1}{2}\kappa_- \mathsf{L} - \tfrac{1}{2}\kappa_- \log_{10} \frac{\rho\phi}{\Lambda_2} - c_0 \tag{67}$$

$$= \tfrac{1}{2}\kappa_- \mathsf{L} - O(1). \tag{68}$$

The expected T count is a weighted average of candidate counts, so it satisfies the same bound. □

The mixed lower bound concerns the prescribed schedule and uses no non-alignment condition. It does not claim that every mixed construction must cost this much.

B.6 Proof of Corollary 1

Proof of Corollary 1. No intrinsic noise. With $\sigma = 0$, the variance budget is $v_t = \mu s_t^2$, and the variance cap at the floor is linear in Δ:

$$\sqrt{\frac{v_t}{\chi_S \Lambda_v}} = \sqrt{\frac{\mu}{\chi_S \Lambda_v}}\, \phi\Delta. \tag{69}$$

For small Δ, this is below the bias cap, which scales as $\sqrt{\Delta}$. Hence $\epsilon^{\mathrm{mix}} = \Theta(\Delta)$. The argument of Theorem 1 then gives both mechanisms the same leading coefficient in L.

Variance-limited regime. For $\sigma > 0$ and $s_t > s_c$, the mixed tolerance satisfies $\epsilon^{\mathrm{mix}} \geq \min\{\epsilon_{\mathrm{loc}}, \epsilon_v\}$, which does not depend on Δ. Its cost is therefore bounded by a constant, while the deterministic cost is at least $\kappa_- \log_{10}(1/s_t) - O(1)$.

Bias-limited regime. The prescribed deterministic tolerance satisfies $\epsilon^{\mathrm{det}} \le \rho\phi\Delta/\Lambda_1$. Applying the candidate argument of Theorem 1 gives a lower bound of $\kappa_-\mathsf{L} - O(1)$ without any non-alignment condition. Combining this with the four bounds of the theorem, the per-rotation cost ratio of the two schedules satisfies

$$\frac{\kappa_-\mathsf{L} - O(1)}{\frac{1}{2}\kappa_{\mathrm{mix}}\mathsf{L} + O(1)} \;\le\; \frac{n_{\mathrm{det}}}{n_{\mathrm{mix}}} \;\le\; \frac{\kappa_{\mathrm{det}}\mathsf{L} + O(1)}{\frac{1}{2}\kappa_-\mathsf{L} - O(1)}. \tag{70}$$

Letting $\mathsf{L} \to \infty$,

$$\frac{2\kappa_-}{\kappa_{\mathrm{mix}}} \;\le\; \liminf \frac{n_{\mathrm{det}}}{n_{\mathrm{mix}}} \;\le\; \limsup \frac{n_{\mathrm{det}}}{n_{\mathrm{mix}}} \;\le\; \frac{2\kappa_{\mathrm{det}}}{\kappa_-}. \tag{71}$$

Both limits equal 2 when $\kappa_- = \kappa_{\mathrm{det}} = \kappa_{\mathrm{mix}}$. The offsets β_m enter only the $O(1)$ terms. Non-generic rotations cost $O(1)$ in both schedules, so the ratio of the per-update totals has the same limit whenever $q_{\mathrm{gen}} > 0$. □

B.7 Proof of Theorem 2

Question (RQ3). Does training under the FT error budgets reach the stationarity target, and after how many updates?

Answer. Yes, for SGD. For any admissible step size, the average squared gradient norm is bounded by three terms: the initial gap, a bias floor, and the combined shot, batch, and synthesis variance. The step size and horizon make these terms at most $\Delta^2/2$, $3\Delta^2/8$, and $\Delta^2/8$. Mixed-synthesis variance enters only through the factor $1 + \lambda$, which lengthens the horizon by that factor.

Proof of Theorem 2. Write $g_t = \nabla L(\theta_t)$. The oracle makes s_t measurable with respect to $\mathcal{F}_t$, so the budgets can be enforced at every update. By Lemma 3(c) and Proposition 1, they give

$$\mathbb{E}[\hat{g}_t \mid \mathcal{F}_t] = g_t + b_t, \qquad \|b_t\| \le \rho s_t, \qquad \mathbb{E}\big[\|\xi_t\|^2 \mid \mathcal{F}_t\big] \le (1+\lambda)\sigma^2 + \mu s_t^2. \tag{72}$$

Step 1: one-step descent. By ℓ-smoothness,

$$L(\theta_{t+1}) \le L(\theta_t) - \eta\langle g_t, \hat{g}_t\rangle + \frac{\ell\eta^2}{2}\|\hat{g}_t\|^2. \tag{73}$$

Taking the conditional expectation and using $\mathbb{E}\|\hat{g}_t\|^2 = \|g_t + b_t\|^2 + \mathbb{E}\|\xi_t\|^2$,

$$\mathbb{E}[L(\theta_{t+1}) \mid \mathcal{F}_t] \le L(\theta_t) - \eta\langle g_t, g_t + b_t\rangle + \frac{\ell\eta^2}{2}\|g_t + b_t\|^2 + \frac{\ell\eta^2}{2}\big[(1+\lambda)\sigma^2 + \mu s_t^2\big]. \tag{74}$$

Step 2: removing the inner product. Since $\eta \le 1/\ell$, we have $\frac{\ell\eta^2}{2}\|g_t + b_t\|^2 \le \frac{\eta}{2}\|g_t + b_t\|^2$. Expanding the square,

$$-\langle g_t, g_t + b_t\rangle + \tfrac{1}{2}\|g_t + b_t\|^2 = -\|g_t\|^2 - \langle g_t, b_t\rangle + \tfrac{1}{2}\|g_t\|^2 + \langle g_t, b_t\rangle + \tfrac{1}{2}\|b_t\|^2 \tag{75}$$

$$= -\tfrac{1}{2}\|g_t\|^2 + \tfrac{1}{2}\|b_t\|^2. \tag{76}$$

Therefore

$$\mathbb{E}[L(\theta_{t+1}) \mid \mathcal{F}_t] \le L(\theta_t) - \frac{\eta}{2}\|g_t\|^2 + \frac{\eta}{2}\|b_t\|^2 + \frac{\ell\eta^2\mu}{2}s_t^2 + \frac{\ell\eta^2}{2}(1+\lambda)\sigma^2. \tag{77}$$

Step 3: bias and μ terms. We have $\|b_t\|^2 \le \rho^2 s_t^2$ and $s_t^2 = \max\{\|g_t\|^2, g_\Delta^2\} \le \|g_t\|^2 + g_\Delta^2$. Since $\eta \le \rho^2/(2\ell\mu)$, we have $\frac{\ell\eta^2\mu}{2} \le \frac{\eta\rho^2}{4}$. Hence

$$\frac{\eta}{2}\|b_t\|^2 + \frac{\ell\eta^2\mu}{2}s_t^2 \le \Big(\frac{\eta\rho^2}{2} + \frac{\eta\rho^2}{4}\Big)\big(\|g_t\|^2 + g_\Delta^2\big) = \frac{3\eta\rho^2}{4}\big(\|g_t\|^2 + g_\Delta^2\big). \tag{78}$$

Substituting,

$$\mathbb{E}[L(\theta_{t+1}) \mid \mathcal{F}_t] \le L(\theta_t) - \frac{\eta}{2}\Big(1 - \frac{3}{2}\rho^2\Big)\|g_t\|^2 + \frac{3\eta\rho^2}{4}g_\Delta^2 + \frac{\ell\eta^2}{2}(1+\lambda)\sigma^2. \tag{79}$$

Since $\rho^2 \le \frac{1}{2}$, we have $1 - \frac{3}{2}\rho^2 \ge \frac{1}{2}(1-\rho^2)$, because this rearranges to $2 - 3\rho^2 \ge 1 - \rho^2$. So

$$\mathbb{E}[L(\theta_{t+1}) \mid \mathcal{F}_t] \le L(\theta_t) - \frac{\eta(1-\rho^2)}{4}\|g_t\|^2 + \frac{3\eta\rho^2}{4}g_\Delta^2 + \frac{\ell\eta^2}{2}(1+\lambda)\sigma^2. \tag{80}$$

Step 4: telescoping (general bound). Take total expectations, sum over $t < T$, and use $L(\theta_0) - \mathbb{E}\, L(\theta_T) \le D$:

$$\frac{\eta(1-\rho^2)}{4}\sum_{t<T}\mathbb{E}\|g_t\|^2 \le D + T\Big[\frac{3\eta\rho^2}{4}g_\Delta^2 + \frac{\ell\eta^2}{2}(1+\lambda)\sigma^2\Big]. \tag{81}$$

Dividing by $T\eta(1-\rho^2)/4$ gives the general bound of the theorem:

$$\frac{1}{T}\sum_{t<T}\mathbb{E}\|g_t\|^2 \le \frac{4D}{(1-\rho^2)\eta T} + \frac{3\rho^2 g_\Delta^2}{1-\rho^2} + \frac{2\ell\eta(1+\lambda)\sigma^2}{1-\rho^2}. \tag{82}$$

Step 5: tuned constants. We bound the three terms of equation 82 in turn. With $g_\Delta = \phi\Delta$ and $\rho^2\phi^2 = (1-\rho^2)/8$,

$$\frac{3\rho^2 g_\Delta^2}{1-\rho^2} = \frac{3\rho^2\phi^2\Delta^2}{1-\rho^2} = \frac{3}{8}\Delta^2. \tag{83}$$

With $\eta = \eta_\Delta \le (1-\rho^2)\Delta^2/[16\ell(1+\lambda)\sigma^2]$,

$$\frac{2\ell\eta_\Delta(1+\lambda)\sigma^2}{1-\rho^2} \le \frac{2\ell(1+\lambda)\sigma^2}{1-\rho^2}\cdot\frac{(1-\rho^2)\Delta^2}{16\ell(1+\lambda)\sigma^2} = \frac{1}{8}\Delta^2. \tag{84}$$

With $T \ge 8D/[(1-\rho^2)\eta_\Delta\Delta^2]$,

$$\frac{4D}{(1-\rho^2)\eta_\Delta T} \le \frac{4D}{(1-\rho^2)\eta_\Delta}\cdot\frac{(1-\rho^2)\eta_\Delta\Delta^2}{8D} = \frac{1}{2}\Delta^2. \tag{85}$$

Adding the three bounds,

$$\frac{1}{T}\sum_{t<T}\mathbb{E}\|g_t\|^2 \le \Big(\frac{1}{2} + \frac{3}{8} + \frac{1}{8}\Big)\Delta^2 = \Delta^2. \tag{86}$$

For deterministic synthesis, $\lambda = \mu = 0$, and the step-size condition involving μ is absent.

Step 6: output iterate. Since τ is uniform on $\{0, \dots, T-1\}$ and independent of the run,

$$\mathbb{E}\|\nabla L(\theta_\tau)\|^2 = \frac{1}{T}\sum_{t<T}\mathbb{E}\|g_t\|^2 \le \Delta^2. \tag{87}$$

Step 7: horizon ratio. As $\Delta \to 0$, the third term in η_Δ tends to zero while the other two stay fixed, so it becomes the minimum for both mechanisms:

$$\eta_{\text{det}} = \frac{(1-\rho^2)\Delta^2}{16\ell\sigma^2}, \qquad \eta_{\text{mix}} = \frac{(1-\rho^2)\Delta^2}{16\ell(1+\lambda)\sigma^2}. \tag{88}$$

The horizon $T(\Delta)$ is inversely proportional to η_Δ up to the ceiling, which becomes negligible as $T \to \infty$. Hence

$$\frac{T_{\text{det}}}{T_{\text{mix}}} \to \frac{\eta_{\text{mix}}}{\eta_{\text{det}}} = \frac{1}{1+\lambda}. \tag{89}$$

□

B.8 Momentum and adaptive scaling

An evolved question: Does the convergence guarantee of Theorem 2 survive momentum and Adam's adaptive scaling?

Answer: Yes. Momentum recovers the same guarantee with an effective smoothness constant $\ell_\beta = \ell(1+\beta)/(1-\beta)$. Adam preserves the same structure up to the conditioning of its adaptive scaling and an additional drift term that vanishes as $\beta_2 \to 1$.

Shared setting. We retain the notation, smoothness assumption, and error budgets of Theorem 2 and Appendix B.7. Thus $\hat{g}_t = g_t + b_t + \xi_t$, where $g_t = \nabla L(\theta_t)$, $\|b_t\| \le \rho s_t$, and $\mathbb{E}[\|\xi_t\|^2 \mid \mathcal{F}_t] \le (1+\lambda)\sigma^2 + \mu s_t^2$. Only the optimizer dynamics change.

Momentum. Consider

$$m_t = \beta m_{t-1} + (1-\beta)\hat{g}_t, \qquad \theta_{t+1} = \theta_t - \eta m_t,$$

with $m_{-1} = 0$. Define

$$z_t = \theta_t + \frac{\beta}{1-\beta}(\theta_t - \theta_{t-1}), \qquad z_0 = \theta_0. \tag{90}$$

Unrolling the momentum recursion and applying convexity gives

$$\sum_{t=0}^{T-1} \|m_{t-1}\|^2 \le \sum_{t=0}^{T-1} \|\hat{g}_t\|^2. \tag{91}$$

Momentum extension. The auxiliary sequence removes the momentum recursion:

$$\begin{aligned} z_{t+1} - z_t &= \frac{\theta_{t+1} - \theta_t}{1-\beta} - \frac{\beta(\theta_t - \theta_{t-1})}{1-\beta} \\ &= \frac{-\eta m_t + \beta\eta m_{t-1}}{1-\beta} \\ &= -\eta \hat{g}_t, \end{aligned} \tag{92}$$

and

$$\|z_t - \theta_t\| = \frac{\beta\eta}{1-\beta}\|m_{t-1}\|. \tag{93}$$

By smoothness,

$$\begin{aligned} L(z_{t+1}) &\le L(z_t) - \eta\langle g_t, \hat{g}_t\rangle - \eta\langle \nabla L(z_t) - g_t, \hat{g}_t\rangle + \frac{\ell\eta^2}{2}\|\hat{g}_t\|^2 \\ &\le L(z_t) - \eta\langle g_t, \hat{g}_t\rangle + \frac{\ell\eta^2}{2}\|\hat{g}_t\|^2 + \frac{\ell\eta^2\beta}{2(1-\beta)}\left(\|m_{t-1}\|^2 + \|\hat{g}_t\|^2\right). \end{aligned} \tag{94}$$

Summing over t and applying equation 91 yields

$$L(z_T) - L(z_0) \le \sum_{t<T}\left[-\eta\langle g_t, \hat{g}_t\rangle + \frac{\ell_\beta\eta^2}{2}\|\hat{g}_t\|^2\right], \qquad \ell_\beta = \ell\frac{1+\beta}{1-\beta}. \tag{95}$$

Equation equation 95 has exactly the same form as the descent inequality used in the proof of Theorem 2, with ℓ replaced by ℓ_β. Therefore, Steps 2–7 of Appendix B.7 apply directly, giving

$$\frac{1}{T}\sum_{t<T}\mathbb{E}\|g_t\|^2 \le \frac{4D}{(1-\rho^2)\eta T} + \frac{3\rho^2 g_\Delta^2}{1-\rho^2} + \frac{2\ell_\beta\eta(1+\lambda)\sigma^2}{1-\rho^2}. \tag{96}$$

Hence the step size, stationarity guarantee, and horizon ratio of Theorem 2 remain valid after replacing ℓ by ℓ_β. □

Adam. Now let

$$\theta_{t+1} = \theta_t - \eta P_t m_t, \qquad P_t = \operatorname{diag}\left(\frac{1}{\sqrt{v_{t-1}} + \varepsilon}\right),$$

where $v_t = \beta_2 v_{t-1} + (1-\beta_2)\hat{g}_t^2$ coordinate-wise. Assume $P_{t,ii} \in [p_{\min}, p_{\max}]$, define $\kappa = p_{\max}/p_{\min}$ and $c_\kappa = 1 - \kappa\rho^2 \ge 1/2$, and assume $\|\hat{g}_t\| \le G$ and $|\hat{g}_{t,i}| \le G_\infty$. Then

$$\begin{aligned} p_{\min}\|u\|^2 \le \|u\|_{P_t}^2 \le p_{\max}\|u\|^2, \\ \|P_t u\|^2 \le p_{\max}\|u\|_{P_t}^2. \end{aligned} \tag{97}$$

Adam extension. Using the same auxiliary sequence,

$$\begin{aligned} z_{t+1} - z_t &= \frac{-\eta P_t m_t + \beta\eta P_{t-1} m_{t-1}}{1-\beta} \\ &= -\eta P_t \hat{g}_t - \frac{\eta\beta}{1-\beta}(P_t - P_{t-1}) m_{t-1} \\ &= -\eta P_t \hat{g}_t - e_t, \end{aligned} \tag{98}$$

where

$$e_t = \frac{\eta\beta}{1-\beta}(P_t - P_{t-1}) m_{t-1}. \tag{99}$$

The adaptive scaling changes by

$$\begin{aligned} &|v_{t-1,i} - v_{t-2,i}| \le (1-\beta_2) G_\infty^2, \\ &|P_{t,ii} - P_{t-1,ii}| \le p_{\max}^2 G_\infty \sqrt{1-\beta_2} =: \delta. \end{aligned} \tag{100}$$

Thus

$$\|e_t\| \le \frac{\eta\beta\delta G}{1-\beta}. \tag{101}$$

Applying smoothness to the auxiliary update, bounding the momentum term as in the momentum case, and using equation 97 gives

$$L(z_T) - L(z_0) \le \sum_{t<T} \left[-\eta \langle g_t, \hat{g}_t \rangle_{P_t} + \frac{\ell_A \eta^2 p_{\max}}{2} \|\hat{g}_t\|_{P_t}^2 + G\|e_t\| + \ell \|e_t\|^2 \right], \tag{102}$$

where

$$\ell_A = 2\ell \left(1 + \frac{\kappa\beta}{1-\beta} \right).$$

Apart from the two drift terms, Eq. equation 102 has the same structure as the descent inequality of Theorem 2, now in the P_t-weighted norm. Applying Steps 2–4 of Appendix B.7, together with equation 97, gives

$$\begin{aligned} \frac{1}{T} \sum_{t<T} \mathbb{E}\|g_t\|^2 \le{}& \frac{4D}{c_\kappa p_{\min} \eta T} + \frac{3\kappa\rho^2 g_\Delta^2}{c_\kappa} + \frac{2\ell_A \eta \kappa p_{\max}(1+\lambda)\sigma^2}{c_\kappa} \\ &+ \frac{4\beta\delta G^2}{(1-\beta) c_\kappa p_{\min}} + \frac{4\ell\eta\beta^2\delta^2 G^2}{(1-\beta)^2 c_\kappa p_{\min}}. \end{aligned} \tag{103}$$

Since $\delta = p_{\max}^2 G_\infty \sqrt{1-\beta_2}$, the fourth term is $O(\sqrt{1-\beta_2})$ and the fifth is $O(\eta(1-\beta_2))$. Thus Adam retains the same gap, bias-floor, and variance structure as Theorem 2, up to the conditioning factor κ, while the additional adaptive-scaling drift vanishes as $\beta_2 \to 1$. □

B.9 PATH-LENGTH CORRECTION

Corollary 2 (Path-length correction)**.** *In the bias-limited regime, if the synthesis and generic-angle constants match, the ratio of the certified cumulative-cost bounds approaches $2/(1+\lambda)$: a certified advantage for $\lambda < 1$, none at $\lambda = 1$, and a loss for $\lambda > 1$.*

Proof. Step 1: per-update cost at most the floor cost. Every update satisfies $s_t \ge \phi\Delta$. With ν_t replaced by $\bar{\nu}$, the tolerances of Proposition 1 are nondecreasing in s_t, and the cost bound $\kappa_m \log_{10}(1/\epsilon) + \beta_m$ is decreasing in ϵ. Every update therefore costs at most its floor value $\hat{n}_m(\Delta)$ per rotation, and

$$\mathbb{E}[C_m] \le S\, N\, T_m\, \hat{n}_m(\Delta). \tag{104}$$

Step 2: floor regime. As $\Delta \to 0$ with $\sigma > 0$, the mixed variance cap tends to the constant ϵ_v, while the bias cap tends to zero. The mixed schedule is therefore bias-limited at the floor. By Theorem 1 and Corollary 1 with matched constants,

$$\frac{\hat{n}_{\mathrm{det}}(\Delta)}{\hat{n}_{\mathrm{mix}}(\Delta)} \to 2. \tag{105}$$

Step 3: ratio of bounds. Combining Step 2 with the horizon ratio of Theorem 2,

$$\frac{S\,N\,T_{\mathrm{det}}\,\hat{n}_{\mathrm{det}}(\Delta)}{S\,N\,T_{\mathrm{mix}}\,\hat{n}_{\mathrm{mix}}(\Delta)} \to \frac{1}{1+\lambda} \cdot 2 = \frac{2}{1+\lambda}. \tag{106}$$

□

This is a ratio of certified upper bounds. The realized ratio also depends on how many updates each run spends above the floor, where both mechanisms are cheaper. For this reason, the experiments report realized path length and cumulative cost directly.

B.10 Mechanism and fidelity selection

Question (RQ4). When should the learner use deterministic or mixed synthesis, and how should the bias budget be shared with magic-state error?

Answer. Under a logarithmic cost model with a mixture overhead, mixed synthesis is preferred exactly below a single convergence level. The mechanism therefore switches at most once, from deterministic to mixed, as training approaches stationarity. At an interior optimum of the relaxed joint problem, the ratio of magic-state bias to synthesis bias equals the ratio of their marginal costs.

We use the cost model with equality and a common slope, and write the mixture overhead as $\Delta\beta$:

$$n_m(\epsilon) = \kappa \log_{10} \frac{1}{\epsilon} + \beta_m, \qquad \Delta\beta = \beta_{\mathrm{mix}} - \beta_{\mathrm{det}} \geq 0. \tag{107}$$

Fix the magic-state infidelity, so that ν_t is fixed and $u_t = \rho s_t - \nu_t > 0$. Define the level at which the deterministic tolerance reaches its cap, and the cost difference between the two schedules:

$$s_{\mathrm{loc}} = \frac{\Lambda_1 \epsilon_{\mathrm{loc}} + \nu_t}{\rho}, \qquad \Gamma(s_t) = n_{\mathrm{det}}\big(\epsilon_t^{\mathrm{det}}\big) - n_{\mathrm{mix}}\big(\epsilon_t^{\mathrm{mix}}\big) = \kappa \log_{10} \frac{\epsilon_t^{\mathrm{mix}}}{\epsilon_t^{\mathrm{det}}} - \Delta\beta. \tag{108}$$

Mixed synthesis is preferred when $\Gamma > 0$.

Proposition 2 (Switching and budget split)**.** *(i) For $s_t \geq s_{\mathrm{loc}}$, $\Gamma(s_t) \leq -\Delta\beta \leq 0$. For $s_t < s_{\mathrm{loc}}$, Γ is nonincreasing in s_t. Hence the set of levels at which mixed synthesis is preferred is an interval $[\phi\Delta, s^\star)$ with $s^\star \leq s_{\mathrm{loc}}$; it is empty if $\Gamma(\phi\Delta) \leq 0$. (ii) Consider minimizing $n_m(\epsilon)\log_{10}(1/p_{\mathrm{MS}})$ over continuous $\epsilon \in (0, \epsilon_{\mathrm{loc}})$ and $p_{\mathrm{MS}} \in (0,1)$, subject to $\Lambda_m \epsilon^{k_m} + R_{\mathrm{MS}} p_{\mathrm{MS}} \leq B_t$, where $B_t = \rho s_t - R_{\mathrm{dep}} p_{\mathrm{dep}}$ and the variance constraint is inactive. At any interior minimizer, the constraint is active and*

$$\frac{R_{\mathrm{MS}}\, p_{\mathrm{MS}}}{\Lambda_m \epsilon^{k_m}} = \frac{k_m\, n_m(\epsilon)}{\kappa \log_{10}(1/p_{\mathrm{MS}})}. \tag{109}$$

Proof. (i), above s_{loc}. For $s_t \geq s_{\mathrm{loc}}$, the deterministic tolerance equals its cap, and every mixed tolerance is at most that cap:

$$\epsilon_t^{\mathrm{det}} = \epsilon_{\mathrm{loc}} \geq \epsilon_t^{\mathrm{mix}} \quad \Longrightarrow \quad \Gamma(s_t) = \kappa \log_{10} \frac{\epsilon_t^{\mathrm{mix}}}{\epsilon_{\mathrm{loc}}} - \Delta\beta \leq -\Delta\beta. \tag{110}$$

(i), below s_{loc}. For $s_t < s_{\mathrm{loc}}$, the deterministic tolerance is $\epsilon_t^{\mathrm{det}} = u_t/\Lambda_1$. The tolerance ratio is then the minimum of three terms:

$$\frac{\epsilon_t^{\mathrm{mix}}}{\epsilon_t^{\mathrm{det}}} = \min\Big\{ \frac{\Lambda_1 \epsilon_{\mathrm{loc}}}{u_t},\ \frac{\Lambda_1}{\sqrt{\Lambda_2 u_t}},\ \frac{\Lambda_1}{u_t}\sqrt{\frac{\lambda\sigma^2 + \mu s_t^2}{\chi_S \Lambda_v}} \Big\}. \tag{111}$$

The first two terms decrease in s_t because u_t increases. For the third term, write its square up to a positive constant as

$$h(s) = \frac{\lambda\sigma^2 + \mu s^2}{(\rho s - \nu_t)^2}. \tag{112}$$

The sign of $h'(s)$ is the sign of

$$2\mu s(\rho s - \nu_t) - 2\rho(\lambda\sigma^2 + \mu s^2) = 2\mu\rho s^2 - 2\mu s\nu_t - 2\rho\lambda\sigma^2 - 2\rho\mu s^2 \tag{113}$$

$$= -2\big(\mu s\nu_t + \rho\lambda\sigma^2\big) \le 0. \tag{114}$$

A minimum of nonincreasing functions is nonincreasing, so the tolerance ratio, and hence Γ, is nonincreasing on $[\phi\Delta, s_{\mathrm{loc}})$.

(i), conclusion. Combining the two cases, $\{\Gamma > 0\}$ is contained in $[\phi\Delta, s_{\mathrm{loc}})$. Since Γ is nonincreasing there, this set is an initial interval $[\phi\Delta, s^\star)$.

(ii) Write $\ell_p = \log_{10}(1/p_{\mathrm{MS}})$ and $f(\epsilon, p_{\mathrm{MS}}) = n_m(\epsilon)\,\ell_p$. Both partial derivatives of f are negative:

$$\frac{\partial f}{\partial \epsilon} = -\frac{\kappa\,\ell_p}{\epsilon \ln 10} < 0, \qquad \frac{\partial f}{\partial p_{\mathrm{MS}}} = -\frac{n_m(\epsilon)}{p_{\mathrm{MS}} \ln 10} < 0. \tag{115}$$

If the constraint were slack at a minimizer, increasing ϵ slightly would lower f. The constraint is therefore active. The Karush–Kuhn–Tucker conditions give a multiplier $\zeta > 0$ with

$$\frac{\kappa\,\ell_p}{\epsilon \ln 10} = \zeta\, k_m \Lambda_m \epsilon^{k_m - 1}, \qquad \frac{n_m(\epsilon)}{p_{\mathrm{MS}} \ln 10} = \zeta\, R_{\mathrm{MS}}. \tag{116}$$

Solving each equation for ζ and equating,

$$\frac{\kappa\,\ell_p}{k_m \Lambda_m \epsilon^{k_m}} = \frac{n_m(\epsilon)}{R_{\mathrm{MS}}\, p_{\mathrm{MS}}} \quad \Longleftrightarrow \quad \frac{R_{\mathrm{MS}}\, p_{\mathrm{MS}}}{\Lambda_m \epsilon^{k_m}} = \frac{k_m\, n_m(\epsilon)}{\kappa\,\ell_p}. \tag{117}$$

□

Part (i) states when switching occurs, not that it must be observed. A trajectory that starts below $s^\star$ selects mixed synthesis throughout, which is consistent with the late-training continuation in our experiments. Part (ii) is a necessary condition for the continuous relaxation. The controller used in the experiments searches a finite action set, so its choices need not satisfy this condition exactly.

C Setup Details

C.1 Compared methods

Table 7 summarizes the compared methods. Ideal training uses exact parameterized rotations and serves only as a quality reference. FH and SD use deterministic Ross–Selinger synthesis. FH fixes the synthesis tolerance at the tightest value required over training, while SD adapts the deterministic tolerance using Proposition 1. FT-QNN replaces deterministic synthesis with three-axis mixed synthesis under either fixed-query (FQ) or per-shot (PS) sampling. The full FT-QNN controller additionally selects magic-state fidelity jointly with synthesis precision and drives the provisioning experiment in RQ5.

Table 7: Summary of compared methods. FH and SD represent deterministic FT compilation strategies, while FQ and PS are FT-QNN mixed-synthesis variants.

Method	Rotation synthesis	ϵ schedule	Mixed	Redraw policy	Adaptive p_{MS}	Role
Ideal	Exact	–	No	–	–	Quality reference
FH	Deterministic Clifford+T	Fixed, tightest required	No	–	No	Fixed-accuracy reference (Beverland et al., 2022)
SD	Deterministic Clifford+T	Adaptive (Prop. 1)	No	–	No	Adaptive reference (cf. Sayginel et al., 2024)
FQ	Three-axis mixed	Prop. 1	Yes	Per query	No	FT-QNN variant
PS	Three-axis mixed	Prop. 1	Yes	Per shot	No	FT-QNN variant
FT-QNN	Three-axis mixed	Jointly scheduled	Yes	Per shot (RQ4)	Yes	Full method

FH follows the fixed-accuracy compilation setting of prior resource and variational FT studies (Sayginel et al., 2024; Beverland et al., 2022). SD uses the same deterministic synthesis but replaces a fixed accuracy with the convergence-aware tolerance of Proposition 1; it adopts the adaptive-accuracy idea of Sayginel et al. (2024) rather than their adaptation rule. FQ and PS use the same mixed-synthesis construction and differ only in when the candidate is redrawn: once per circuit query for FQ, or independently for every shot for PS. Joint magic-state fidelity selection is enabled only in the RQ4 experiment, which uses per-shot redraws in both arms, and RQ5 replays the resulting fidelity demand for server provisioning. FH, fixed-loose deterministic synthesis ($\epsilon = 3 \times 10^{-3}$), and fixed-precision mixed synthesis ($\epsilon = 3 \times 10^{-4}$) are evaluated as cost replays along the recorded scheduled-deterministic trajectory.

C.2 Data, encoding, and models

Table 8 lists the datasets. Every preprocessing step is fitted on the training subset only, and transformed values outside the fitted range are clipped. Each processed feature x_q is encoded once on qubit q as $R_Y(x_q)|0\rangle$, so a circuit on q qubits uses q features.

Table 8: Datasets and preprocessing. Angle scaling maps each feature to $[-\pi/2, \pi/2]$.

Dataset	Classes	Raw features	Preprocessing	Split (train/val/test)
Digits	0 vs. 1	64 pixels	Standardize, PCA, angle scaling	200 / – / 160
UNSW-NB15	Normal vs. Generic	39 numeric	Median imputation, standardize, PCA, angle scaling	128 / 64 / 128
Waveform	1 vs. 2	40	Standardize, PCA, angle scaling	128 / 64 / 128

Each circuit has q encoding rotations and $P = 2qL$ trainable rotations, one R_Y and one R_Z per qubit and layer, followed by a CZ ring. Parameters are initialized independently and uniformly in $[-\pi, \pi)$. Table 9 lists the architectures.

Table 9: Circuit architectures. One B32 round runs $32(1 + 2P)$ circuits before shot repetition.

Architecture	Qubits q	Layers L	Parameters P	Circuits per B32 round
4q/9L	4	9	72	4,640
6q/4L	6	4	48	3,104
6q/6L	6	6	72	4,640
6q/9L (main)	6	9	108	6,944
8q/9L	8	9	144	9,248

C.3 Training protocol

All runs use Adam with $\beta = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. The learning rate starts at 0.1 and drops to 0.03, 0.01, 0.003, and 0.001 at updates 100, 250, 500, and 800. Gradients use the parameter-shift rule on the mean-squared loss with a global parity observable.

Stopping and convergence. Runs in RQ3, the size study, and the equal-budget study train for at least 800 and at most 1,500 updates. After update 800, a run stops when it is operationally stable, or when it passes a rolling stationarity test together with a quality gate. A run is operationally stable if, over its last 50 updates, the relative loss range is at most 5×10^{-4}, the absolute fitted loss slope is at most 10^{-6} per update, and the training accuracy is constant. The rolling stationarity test uses a window of 10 updates: the median ideal gradient norm must be at most the target, and at least 8 of the 10 values at most 1.5 times the target, with targets 0.002 (finite shots) and 0.001 (exact). Gradient-target passages are reported separately for targets 0.02, 0.01, 0.005, 0.003, 0.002, 0.001, and 0.0008. The UNSW-NB15 and Waveform runs train for exactly 500 updates. The ideal runs used for the RQ1 and RQ2 checkpoints stop when the full-data gradient norm stays below 0.001 for 20 consecutive updates. RQ4 is a fixed continuation from update 800 to 1,100. Table 10 summarizes each experiment.

Seeds. Results are averaged over three independent seeds, drawn from $\{20260910, 20260922, 20260925, 20260928, 20260932\}$. Each seed sets the data split, the

Table 10: Training settings per experiment.

Experiment	Batch / shots	Max updates	Matched update	Arms
RQ1/RQ2 checkpoints	ideal Adam	1,500	–	Ideal
RQ3 noise floor	B32, B64, B200 / S500, S1000	1,500	800	SD, FQ, PS
Size study	B32/S500	1,500	800	SD, FQ
Equal budget	B32/S500	1,500	800	SD, FQ
UNSW-NB15, Waveform	B32/S500	500	500	Ideal, SD, FQ
RQ4	B32/S500	800 $\rightarrow$ 1,100	1,100	PS, fixed vs. joint fidelity

preprocessing fit, the parameter initialization, and all synthesis draws and shot samples, so the reported variation includes data variation.

C.4 Controller, synthesis, and server settings

Table 11: Controller, synthesis, and distillation-server settings.

Setting	Value
Controller	
Bias budget	$\Delta_{b,t} = \rho\|\nabla L(\theta_t)\|$, $\rho = 0.1$, oracle norm at every update ($g_\Delta = 0$)
Bias constants	$A = 75$ (deterministic, $A\epsilon$), $B_2 = 35$ (mixed, $B_2\epsilon^2$)
Tolerance grid	10^{-1} to 10^{-6}, 15 values (Appendix F.4)
Tolerance cap	3×10^{-3} in the main experiments
Variance rule	256 baseline noise repetitions, 16 synthesis draws per probe, reference B200/S1000
Fixed tolerances	FH 3×10^{-6}; fixed-loose 3×10^{-3}; fixed-precision mixed 3×10^{-4}
Equal budget	Tightest grid tolerance with expected pre-shot T per B32 round $\leq$ 17M, 19M, or 22M
Magic states (RQ4)	
Fidelity tiers	$p_{\mathrm{MS}} \in \{10^{-6}, 3 \times 10^{-7}, 10^{-7}, 3 \times 10^{-8}\}$; fixed arm 3×10^{-8}
Response bound	$R_{\mathrm{MS}} = 2571.3$
Channel	Exact feed-forward injection superoperator at every compiled T gate
Distillation server (RQ5)	
Protocol	One-level 15-to-1, $p_{\mathrm{in}} = 5 \times 10^{-4}$, success probability $1 - 15p_{\mathrm{in}}$
Output error	$35p_{\mathrm{in}}^3 + Vp_L$, $p_L = 0.1(100p_{\mathrm{phys}})^{(d+1)/2}$, odd $d \in \{3, \ldots, 49\}$
Scenarios	$p_{\mathrm{phys}} \in \{10^{-4}, 3 \times 10^{-4}, 10^{-3}\} \times V \in \{30, 100, 300\}$; primary $(10^{-3}, 100)$
Volume and buffer	$15d^3/(1 - 15p_{\mathrm{in}})$ per state; one-round predictive warm-up
Reconfiguration	Break-even delay $\tau^\star = (W_{\mathrm{fixed}} - W_{\mathrm{sched}})/(N_{\mathrm{switch}}\, C_{\mathrm{sched,peak}})$

Response tensors and calibration. The RQ1 and RQ2 analyses use frozen checkpoints of ideal Adam runs at updates 0, 10, 50, 100, 200, and the terminal update; RQ2 calibration uses updates 0, 200, and 368. Response tensors are central finite differences of full-data adjoint MSE gradients under individual gate-frame perturbations, with step 5×10^{-5} for first-order and 10^{-3} for diagonal second-order responses, each selected from a step sweep. Adjoint gradients agree with parameter shift to 1.12×10^{-15}.

C.5 Simulation and compute

All simulators are custom and written in PyTorch with complex128 states and float64 parameters and gradients. Training in RQ3, the size study, the dataset study, and the equal-budget study, as well as the RQ1 response derivatives and the RQ2 calibration, use a state-vector simulator. The RQ1 exact mixed-channel checks and RQ4 use a density-matrix simulator, in which mixed synthesis averages the candidate and conjugation channels before propagation and every compiled T gate is replaced by the injection superoperator. Rotations are synthesized with `pygridsynth` 1.1.0, and the mixture LP is solved with SciPy's HiGHS interface. Runs used one NVIDIA RTX 4090 (24 GB) with Python 3.9.20, PyTorch 2.4.1, CUDA 12.1, NumPy 1.26.4, SciPy 1.13.1, and scikit-learn 1.5.2. Recorded wall time, including simulation, compilation, and monitoring, totals about 8.3 h for RQ3,

2.3 h for the size study, 7.8 h for the equal-budget study, 0.8 h for the dataset study, 6.1 h for RQ4, and 0.5 h for the RQ2 calibration. GPU-active time and peak memory were not recorded separately.

D Additional Results

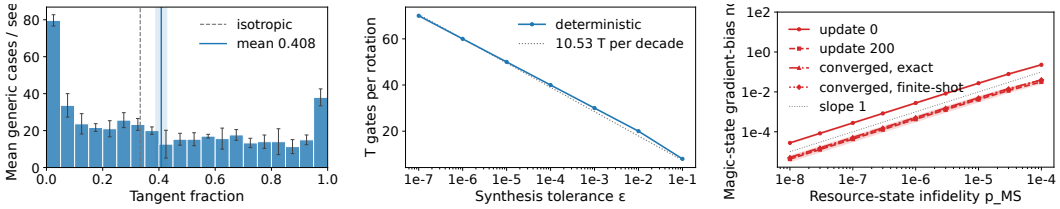
D.1 RQ1: Synthesis and gradient response

This section gives the details behind Figure 2. It covers the synthesis characterization, the magic-state response, and the numerical checks behind the error orders and the variance law. All values are mean $\pm$ sample SD over three seeds ($n = 3$).

Synthesis characterization: Each seed synthesized 64 generic angles at 7 tolerances, giving 448 cases, together with exactly synthesizable control angles. The controls were excluded from all fits. Figure 7a shows how much of each error generator lies along the rotation axis. The mean share is 0.408 ± 0.020, not the isotropic value $1/3$, so we do not assume isotropy anywhere. Figure 7b shows the deterministic T count. Over the sampled range, it grows by 10.53 ± 0.09 T gates per decade of tolerance. Mixed-synthesis points below the predefined 10^{-13} numerical floor were left out of the slope fits. Table 13 lists the channel exponents for each seed.

Magic-state response: Figure 7c shows the gradient bias under exact T-gate injection through the compiled QNN. The fitted exponent in p_{MS} is 0.9972 ± 0.0011, and the coefficient $\|b_{\mathrm{MS}}\|/p_{\mathrm{MS}}$ is 519.0 ± 96.4. The coefficient changes over training, which is why R_{MS} enters the theory as a trajectory-level bound.

Numerical checks: Table 12 collects the checks behind Lemmas 1 and 2 and Lemma 3(c). The covariance, $1/S$, and cross-gate checks support the independent-draw model at the tested sizes. They are not a proof that every training circuit has decorrelated errors. The last row compares the frozen-error diagnostic, in which base and shifted circuits share one draw, with the mean-channel evaluation. Their difference is the covariance term discussed in Appendix B.1. The tolerance budget used the larger of the two values.



(a) Tangent fraction (b) T count vs. ϵ (c) Magic-state bias vs. p_{MS}

Figure 7: Synthesis and magic-state characterization (RQ1), mean over three seeds with $\pm$ SD. (a) Share of the synthesis error along the rotation axis, over all generic cases. (b) Deterministic T count, median over angles, with the fitted cost per decade. (c) Gradient bias under exact injection, one line per checkpoint.

D.1.1 Cross-dataset check

The error orders in Lemmas 1 and 2 do not depend on the task, so one model is enough to test them. To check that nothing in the digits setup drives the result, we repeated the gradient-level measurements on the two external datasets used in RQ3: UNSW-NB15 (Normal vs. Generic) and Waveform (class 1 vs. 2). Both use the smaller 6-qubit, 4-layer model with 48 parameters. We took three checkpoints from each deterministic training run, at updates 0, 250, and 500, and applied the same synthesis, tolerance grid, and exact channels as in the digits study.

Table 14 shows the results. On UNSW-NB15 and Waveform, deterministic bias exponents range from 0.861 to 1.044, and mixed bias exponents from 1.951 to 1.999. The covariance trace ratio stays within 0.995 to 1.004, and redraws follow $1/S$ within 2.2%. The magic-state bias remains first order, with exponents from 0.997 to 0.999. The response coefficient $\|b_{\mathrm{MS}}\|/p_{\mathrm{MS}}$ differs between tasks, which is expected: it sets the constant, not the order.

Table 12: Numerical checks for RQ1. Mean ± sample SD over three seeds ($n = 3$).

Quantity	Value
Channel-error exponent, deterministic	0.9943 ± 0.0008
Channel-error exponent, tangent-only	0.9918 ± 0.0138
Channel-error exponent, three-axis	1.9940 ± 0.0028
Mean tangent fraction	0.408 ± 0.020
Deterministic T count per decade of ϵ	10.53 ± 0.09
Gradient-bias exponent, deterministic	0.9907 ± 0.0189
Gradient-bias exponent, three-axis mixed	1.9832 ± 0.0055
Covariance trace ratio, linear audit	0.9989 ± 0.0030
Covariance trace ratio, nonlinear at $\epsilon = 10^{-2}$	1.0029 ± 0.0084
Max deviation from $1/S$ ($S = 1, 4, 16, 64$)	$1.84 \pm 0.74\%$
Max normalized cross-gate covariance	0.00881 ± 0.00009
Magic-state bias exponent in p_{MS}	0.9972 ± 0.0011
$\|b_{\mathrm{MS}}\|/p_{\mathrm{MS}}$	519.0 ± 96.4
Frozen / mean-channel bias ratio	1.147 ± 0.041

Table 13: Channel-error exponents by seed.

Construction	Seed 1	Seed 2	Seed 3
Deterministic	0.9952	0.9941	0.9937
Tangent-only	0.9913	0.9783	1.0059
Three-axis	1.9908	1.9960	1.9953

D.2 RQ2: Tolerance budgets and regime predictions

Tolerance budgets over training. Table 15 lists the tolerance caps at six Adam checkpoints of the digits model under the bias budget $\rho = 0.1$. Each cap is the largest tested tolerance satisfying its constraint, and the bracket gives the adjacent infeasible grid point. A maximum is taken over finer tested tolerances to handle compiler nonmonotonicity. The variance cap uses an Adam-specific rule: one mixed draw per trainable gate per update, fixed across shots, must not perturb the next Adam update more than the 4,096-shot parameter-shift estimator does. The conservative envelopes are 3×10^{-6} for deterministic synthesis, 10^{-3} for the mixed bias-only cap, and 3×10^{-4} for the selected mixed cap, where variance binds.

Frozen predictions and calibration. Before any probe or training run, we froze the variance caps and crossovers for six batch and shot settings, using the empirical Adam constants $A = 75$, $B_2 = 35$, and $\rho = 0.1$ (Table 16). The prediction formula is normalized to the B200/S1000 setting, but each calibration uses the setting's own noise. For every setting, we generated 256 baseline Adam-update perturbations with that batch size and shot count, and used the same baseline for both mixed policies. Each of the 36 tests (3 checkpoints $\times$ 6 settings $\times$ 2 mixed policies) passed if the tolerance predicted for that checkpoint met two conditions: a bias test, $\|g_{\mathrm{mixed}} - g_{\mathrm{ideal}}\| \leq 0.1\|g_{\mathrm{ideal}}\|$, and a policy-specific variance test. For fixed-query synthesis, the Adam-update perturbation from synthesis must be at most that from the setting's own noise, estimated with 16 synthesis draws. For per-shot synthesis, the total perturbation must be at most twice the noise alone, estimated with 256 repetitions. All 36 tests passed, and the controllers were not retuned afterward.

Resolution of the calibrated caps. Thresholds were located by geometric bisection between 10^{-6} and 10^{-2} with six refinements. At updates 0 and 200, every setting and policy passed at 10^{-2}, so these caps are upper-censored. At update 368, all twelve setting and policy combinations fell in the same bracket, $[0.00237, 0.00274]$, a factor of 1.15 wide. At its lower end, both tests pass. At its upper end, the bias test fails in every setting, while both variance tests still pass. The shared cap therefore reflects the bias limit, not the noise floor. Finer differences between settings cannot be resolved within this bracket.

Table 14: RQ1 checks across datasets. Ranges are over checkpoints.

Dataset (model)	Det. bias exp.	Mixed bias exp.	Cov. ratio	Max $1/S$ dev.	MS exp.
Digits (6q/9L)	0.913–1.149	1.945–2.002	0.996–1.008	1.5%	0.994–0.998
UNSW-NB15 (6q/4L)	0.861–1.044	1.965–1.999	0.995–1.004	2.2%	0.997–0.999
Waveform (6q/4L)	0.877–0.957	1.951–1.982	0.999–1.001	1.3%	0.998–0.999

Table 15: Tolerance caps at Adam checkpoints of the digits model. Brackets give the largest feasible and smallest infeasible tested tolerance.

Update	$\|\nabla L\|$	Det. cap	Mixed bias cap	Mixed var. cap	Selected mixed
0	0.681	3×10^{-3} $[3\times10^{-3}, 10^{-2}]$	3×10^{-2} $[3\times10^{-2}, 10^{-1}]$	3×10^{-3} $[3\times10^{-3}, 10^{-2}]$	3×10^{-3}
10	0.223	10^{-3} $[10^{-3}, 3\times10^{-3}]$	3×10^{-2} $[3\times10^{-2}, 10^{-1}]$	10^{-3} $[10^{-3}, 3\times10^{-3}]$	10^{-3}
50	0.0400	3×10^{-4} $[3\times10^{-4}, 10^{-3}]$	10^{-2} $[10^{-2}, 3\times10^{-2}]$	3×10^{-4} $[3\times10^{-4}, 10^{-3}]$	3×10^{-4}
100	0.00407	3×10^{-5} $[3\times10^{-5}, 10^{-4}]$	3×10^{-3} $[3\times10^{-3}, 10^{-2}]$	3×10^{-4} $[3\times10^{-4}, 10^{-3}]$	3×10^{-4}
200	0.0106	10^{-4} $[10^{-4}, 3\times10^{-4}]$	10^{-2} $[10^{-2}, 3\times10^{-2}]$	3×10^{-4} $[3\times10^{-4}, 10^{-3}]$	3×10^{-4}
368	0.000645	3×10^{-6} $[3\times10^{-6}, 10^{-5}]$	10^{-3} $[10^{-3}, 3\times10^{-3}]$	3×10^{-4} $[3\times10^{-4}, 10^{-3}]$	3×10^{-4}
Envelope	–	3×10^{-6}	10^{-3}	3×10^{-4}	3×10^{-4}

D.3 RQ3: Training cost details

This section gives the details behind RQ3 results: how costs are counted, how the savings split between scheduling and mixing, cumulative cost to gradient targets, the equal-budget comparison, and the dataset setup.

Cost accounting: We report the mean T count per circuit execution, averaged over all base and shifted queries of an update. One update with batch B runs $B(1+2P)$ circuits before shot repetition, which is $32\times217 = 6{,}944$ circuits for the main circuit at B32; the per-round workload is this number times the T count per circuit. Executed magic-state demand multiplies this again by the shot count. Every executed logical T gate consumes a new magic state, so cumulative counts measure execution work, not reusable hardware. Fixed high (FH) and fixed-precision mixed are cost replays: every query of the stored scheduled-deterministic trajectory is recompiled at $\epsilon = 3\times10^{-6}$ (deterministic) or $\epsilon = 3\times10^{-4}$ (mixed, mixture-weighted expected T), with the same angles, batches, and encoder inputs. They give cost only, not training outcomes.

Scheduling versus mixing: On the main circuit at B32/S500, the mean T count per circuit over all 800 updates is 6,263 for FH, 4,878 for scheduled deterministic, 4,227 for fixed-precision mixed, 3,027 for per-shot mixed, and 2,409 for fixed-query mixed. At update 800, scheduling alone saves 15.5% against FH, mixing at a fixed tolerance alone saves 32.5%, and the two together save 61.1%. Fixed-query mixing costs 19.2% less than per-shot mixing.

Cumulative cost and gradient targets: A cheaper update does not guarantee a cheaper run, because arms can take different numbers of updates to reach a gradient target. Figure 8 plots the ideal gradient norm against cumulative executed T, and Table 17 lists target passages. Stable training and passing a tight gradient target are separate outcomes: every run stabilized, while tight targets were passed only by some runs, and unpassed targets are reported as censored.

Equal budget: Table 18 gives both methods the same expected pre-shot T ceiling per round. Each method then uses the tightest tolerance that fits. Deterministic synthesis stabilized under every ceiling, so this comparison shows a budget–quality trade-off rather than a target that only mixed synthesis can reach.

Datasets: UNSW-NB15 uses the Normal and Generic attack classes, with 128 training, 64 validation, and 128 official-test examples, about 0.12% of the official rows. Waveform uses classes 1 and 2, with the same split sizes and six principal components. All preprocessing is fit on the training split only. The Waveform class pair was chosen after results on another pair, so its result is specific to that pair. As a further check, we evaluated the original Waveform checkpoints on the 2,988 re-

Table 16: Frozen predictions and calibration for six batch and shot settings. Intrinsic noise is relative to B200/S1000. Calibrated caps at updates 0 and 200 are upper-censored at 10^{-2}.

Setting	Noise ratio	Pred. fixed-query	Pred. per-shot	Pred. crossover $\|\nabla L\|$	Cal. (0, 200)	Cal. (368)
B32/S500	76.6	0.0105	0.00743	0.035	$\geq 10^{-2}$	[0.00237, 0.00274]
B32/S1000	70.3	0.0101	0.0101	0.035	$\geq 10^{-2}$	[0.00237, 0.00274]
B64/S500	32.2	0.00963	0.00681	0.0324	$\geq 10^{-2}$	[0.00237, 0.00274]
B64/S1000	29.1	0.00915	0.00915	0.0293	$\geq 10^{-2}$	[0.00237, 0.00274]
B200/S500	2.00	0.00424	0.00300	0.00631	$\geq 10^{-2}$	[0.00237, 0.00274]
B200/S1000	1.00	0.00300	0.00300	0.00315	$\geq 10^{-2}$	[0.00237, 0.00274]

maining untouched class-1/2 rows: accuracy is 93.94% for scheduled deterministic and 93.88% for fixed-query mixed synthesis, and MSE is 0.3626 and 0.3704.

Table 17: Gradient-target passages. Top: number of trajectories passing each target, out of 24. Bottom: main circuit at B32/S500, update of first passage and cumulative executed T. "–" denotes that no seed passed the target.

Target $\|\nabla L\|$	0.02	0.01	0.005	0.003	0.002	0.001	0.0008
Runs passing	18	18	9	3	1	0	0

	Target 0.01		Target 0.005	
Arm	Update	Cum. T ($\times 10^{12}$)	Update	Cum. T ($\times 10^{12}$)
SD	1016 ± 180	17.46 ± 3.35	–	–
FQ	1012 ± 107	8.46 ± 0.88	–	–
PS	828 ± 90	8.67 ± 0.89	1237	12.92

Table 18: Equal expected pre-shot T ceiling per round (main circuit, B32/S500, update 800).

	Tolerance ϵ		Train MSE ($\times 10^{-2}$)		Held-out MSE ($\times 10^{-2}$)	
Ceiling	SD	FQ	SD	FQ	SD	FQ
17M	0.006	0.008	8.37	7.65	9.76	8.70
19M	0.003	0.006	7.11	7.24	7.98	8.14
22M	0.003	0.003	7.69	7.16	8.38	8.62

D.3.1 Training trajectories on UNSW-NB15 and Waveform

Figure 9 shows how cost and training quality evolve on the two external datasets with the 6q/4L model at B32/S500. The T demand is the expected pre-shot count per complete parameter-shift round, which runs $32(1 + 2 \times 48) = 3{,}104$ circuits, before multiplying by the 500 shots. The MSE is the training loss evaluated with ideal circuit execution at every update.

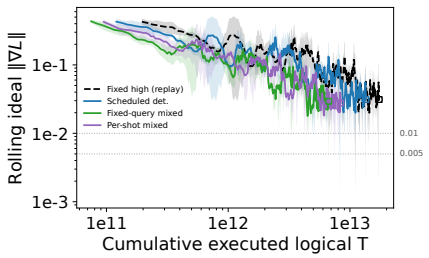
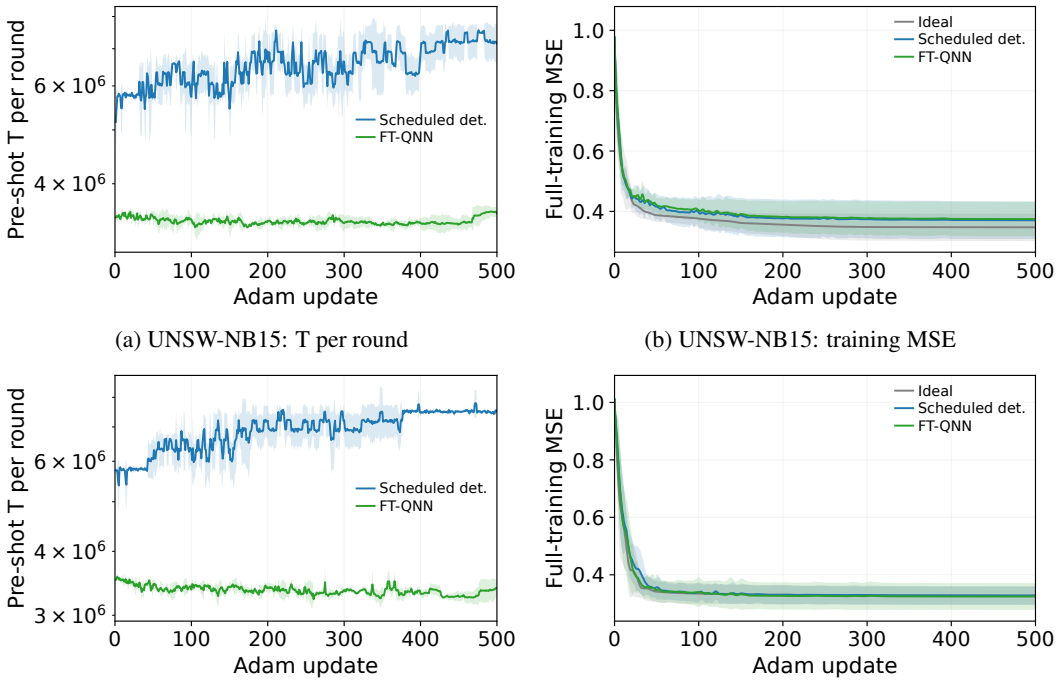


Figure 8: Ideal gradient norm against cumulative executed T on the main circuit (B32/S500), mean over three seeds with $\pm$ SD. Markers show target passages.

On both datasets, the cost of scheduled deterministic synthesis rises as its tolerance tightens with the gradient, from 5.17 ± 0.62M to 7.15 ± 0.58M T gates per round on UNSW-NB15 and from 5.76 ± 0.09M to 7.50 ± 0.05M on Waveform. FT-QNN stays nearly flat, between 3.3M and 3.5M. The saving therefore grows during training, from 32.6% to 50.5% on UNSW-NB15 and from 38.9% to 54.8% on Waveform, with means over all 500 updates of 47.3% and 51.0%. This is the behavior Proposition 1 predicts: early in training, both mechanisms operate near the tolerance cap, and the gap opens as

the deterministic tolerance must follow the shrinking gradient. Training quality is essentially unchanged. At update 500, the training MSE of FT-QNN is 1.06% above scheduled deterministic synthesis on UNSW-NB15 (0.3745 vs. 0.3706) and 1.07% below it on Waveform (0.3244 vs. 0.3279), with both within one seed SD. On UNSW-NB15, both FT arms end slightly above ideal training (0.3473), while on Waveform all three arms end within 1% of each other.



(a) UNSW-NB15: T per round

(b) UNSW-NB15: training MSE

(c) Waveform: T per round

(d) Waveform: training MSE

Figure 9: Training trajectories on UNSW-NB15 and Waveform (6q/4L, B32/S500), mean over three seeds with ± SD. (a, c) Expected pre-shot T gates per parameter-shift round (log scale), scheduled deterministic versus FT-QNN. (b, d) Training MSE with ideal circuit execution for ideal training, scheduled deterministic synthesis, and FT-QNN.

D.4 RQ4 AND RQ5: JOINT SELECTION AND PROVISIONING DETAILS

This section gives the details behind RQs 4 and 5 results: the joint selection rule, the full RQ4 comparison, the distillation model, the provisioning results, their sensitivity to hardware assumptions, and the multi-trace replay.

Joint selection: Both RQ4 policies continue the main run (B32/S500) from update 800 to 1,100 and apply the exact injection-twirl channel at every compiled T gate. The fixed policy uses scheduled mixed synthesis with $p_{\mathrm{MS}} = 3\times10^{-8}$. At each update, joint selection searches a finite set of actions: the mechanism (deterministic or mixed), the tolerance, and the fidelity tier $p_{\mathrm{MS}} \in \{10^{-6}, 3\times10^{-7}, 10^{-7}, 3\times10^{-8}\}$. It keeps only actions whose predicted bias satisfies $b^{\mathrm{syn}}(m,\epsilon)+R_{\mathrm{MS}}p_{\mathrm{MS}} \le 0.1\|\nabla L\|$, with the Adam constants $A = 75$ and $B_2 = 35$ and the conservative $R_{\mathrm{MS}} = 2571.3$. Among these, it minimizes the proxy $n_m(\epsilon)\log_{10}(1/p_{\mathrm{MS}})$, where $n_m(\epsilon)$ is the predicted T count per rotation. This proxy is used only to rank actions; it is not a factory cost. Table 19 gives the full comparison.

Distillation model: The server uses a conditional one-level 15-to-1 model with input error $p_{\mathrm{in}} = 5\times10^{-4}$, leading distillation error $35p_{\mathrm{in}}^3$, and success probability $1-15p_{\mathrm{in}}$. The logical error per cycle at code distance d is $p_L = 0.1(100p_{\mathrm{phys}})^{(d+1)/2}$. A state meets the requested fidelity if $35p_{\mathrm{in}}^3 + Vp_L \le p_{\mathrm{MS}}$, where V is the exposure. Each tier uses the smallest feasible odd distance, and the normalized output volume is $15d^3/(1-15p_{\mathrm{in}})$. Executed T demand is multiplied by the shot

count before factory accounting, and the server buffers one update ahead. These are model choices, not calibrated hardware measurements.

Provisioning on the joint-selection trace: Table 20 and Figure 10 compare fixed-high and scheduled provisioning in the primary scenario ($p_{\text{phys}} = 10^{-3}$, $V = 100$), where tiers 10^{-6} and 3×10^{-7} use distance 15 and tiers 10^{-7} and 3×10^{-8} use distance 17. The trace changes fidelity tier about 27 times but needs only about 15 physical configuration switches, since neighboring tiers often share a distance. At 80% of the fixed-high peak capacity, scheduled provisioning runs without stalls, while fixed-high provisioning adds $22.36 \pm 0.39\%$ to the round time.

Sensitivity: Table 21 repeats the replay for nine combinations of physical error rate and exposure. Work and sustained-capacity savings range from 8.1 to 52.8%. They are not monotone in V, because the savings depend on which tiers share a code distance. The peak falls only where the highest tier needs a larger distance than all other tiers. Separate pools for each distance need 1.690 ± 0.004 times the peak capacity of one flexible shared pool, and the shared pool is preferable as long as a configuration switch costs less than 5.63 ± 2.16 update intervals.

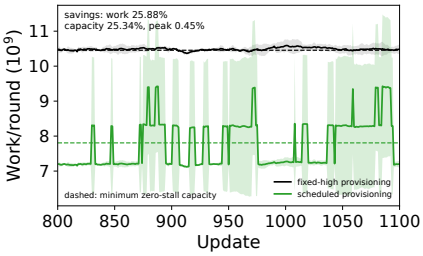


Figure 10: Normalized factory work per update on the joint-selection trace, fixed-high versus scheduled provisioning (primary scenario). Dashed lines mark each policy's zero-stall sustained capacity.

Multi-trace replay: For Table 4, we applied the same fidelity rule to the recorded trace of every RQ3 run: the ideal gradient norm, the synthesis mechanism and tolerance, and the per-round T demand, each unchanged. For each model, R_{MS} was measured by exact injection at three checkpoints and set to the maximum (Table 22). These runs were not retrained with injected magic-state noise, so the replay gives provisioning cost only. The quality evidence for joint selection is the trained continuation in Table 19.

Table 19: Joint selection versus fixed high fidelity, updates 800–1,100. Mean ± SD over three seeds. Accuracy changes are in percentage points.

	Fixed high fidelity	Joint selection	Change (%)
Train MSE ($\times 10^{-2}$)	7.801 ± 0.994	7.799 ± 0.993	-0.02 ± 0.01
Held-out MSE ($\times 10^{-2}$)	8.356 ± 0.937	8.358 ± 0.935	0.03 ± 0.03
Train accuracy (%)	99.67 ± 0.29	99.67 ± 0.29	0.00 ± 0.00
Held-out accuracy (%)	99.58 ± 0.36	99.58 ± 0.36	0.00 ± 0.00
Pre-shot T per round ($\times 10^6$)	20.92 ± 0.20	20.97 ± 0.14	0.24 ± 0.35
T per circuit	3013 ± 29	3020 ± 21	0.24 ± 0.35
Fidelity proxy per round ($\times 10^6$)	157.4 ± 1.5	136.6 ± 2.1	-13.24 ± 0.63
Joint-selection updates per tier (of 300)			
$p_{\text{MS}} = 10^{-6}$	–	54.7 ± 22.7	–
$p_{\text{MS}} = 3 \times 10^{-7}$	–	193.3 ± 16.3	–
$p_{\text{MS}} = 10^{-7}$	–	51.3 ± 23.1	–
$p_{\text{MS}} = 3 \times 10^{-8}$	–	0.7 ± 0.6	–

E Background: QNNs and fault-tolerant execution

This section introduces the quantum computing concepts the paper relies on, for readers from machine learning and from quantum computing alike.

Qubits, gates, and measurement. A qubit is the quantum analogue of a bit. Its state is a unit vector in a two-dimensional complex space, and a register of several qubits has a state in the tensor

Table 20: Provisioning on the joint-selection trace, primary scenario ($p_{\mathrm{phys}} = 10^{-3}$, $V = 100$). Mean $\pm$ SD over three seeds.

	Fixed high	Scheduled	Change (%)
Total normalized work ($\times 10^{12}$)	3.146 ± 0.022	2.332 ± 0.087	-25.88 ± 2.35
Zero-stall sustained capacity ($\times 10^{9}$)	10.455 ± 0.067	7.806 ± 0.267	-25.34 ± 2.11
Peak one-round capacity ($\times 10^{9}$)	10.683 ± 0.095	10.635 ± 0.051	-0.45 ± 0.58
Round-time overhead at 80% of peak (%)	22.36 ± 0.39	0.00 ± 0.00	–
Fidelity-request switches	0	26.7 ± 3.8	–
Configuration switches	0	14.7 ± 4.2	–

Table 21: Sensitivity of provisioning savings to physical error rate and exposure, joint-selection trace. Mean $\pm$ SD over three seeds. Distances are listed for tiers $10^{-6}/3 \times 10^{-7}/10^{-7}/3 \times 10^{-8}$ and are the same across seeds.

p_{phys}	V	Work (%)	Sust. cap. (%)	Peak (%)	Distances
10^{-4}	30	52.83 ± 0.10	52.82 ± 0.12	18.74 ± 29.63	7/7/7/9
10^{-4}	100	43.78 ± 3.98	42.75 ± 3.49	0.45 ± 0.58	7/7/9/9
10^{-4}	300	9.64 ± 4.01	9.52 ± 4.19	0.00 ± 0.00	7/9/9/9
3×10^{-4}	30	45.13 ± 0.09	45.12 ± 0.10	16.17 ± 25.17	9/9/9/11
3×10^{-4}	100	37.40 ± 3.40	36.54 ± 3.00	0.45 ± 0.58	9/9/11/11
3×10^{-4}	300	8.24 ± 3.43	8.14 ± 3.57	0.00 ± 0.00	9/11/11/11
10^{-3}	30	35.60 ± 1.77	35.56 ± 1.84	11.53 ± 17.13	13/15/15/17
10^{-3}	100	25.88 ± 2.35	25.34 ± 2.11	0.45 ± 0.58	15/15/17/17
10^{-3}	300	32.39 ± 1.66	32.36 ± 1.71	10.55 ± 15.44	15/17/17/19

product of their spaces, whose dimension doubles with every added qubit. Computation applies gates, which are unitary matrices acting on one or two qubits at a time. A measurement returns a random outcome whose probabilities are set by the state. Repeating the same circuit many times, each repetition being called a shot, estimates the expected value of the measured quantity. Any finite number of shots gives only an estimate, so every quantity read from a quantum circuit carries sampling noise.

Quantum neural networks. A quantum neural network (QNN) is a parameterized quantum circuit trained in the same way as a classical neural network (Mitarai et al., 2018; Schuld et al., 2019). The input is first loaded into the quantum state by rotations whose angles depend on the data. A trainable circuit then follows, built from layers of single-qubit rotations whose angles are the model parameters, interleaved with two-qubit gates that create correlations, or entanglement, between qubits. In our circuits, each layer applies rotations about the Y and Z axes to every qubit and then a ring of controlled-Z gates between neighboring qubits. The prediction is the expected value of a measured observable,

$$f_\theta(x) = \langle \psi(x) | \, U(\theta)^\dagger \, O \, U(\theta) \, | \psi(x) \rangle, \tag{118}$$

where the encoded state is written $|\psi(x)\rangle$, the trainable circuit $U(\theta)$, and the observable O. We measure the parity of all qubits, which takes the values $+1$ and -1, matching binary labels. Training minimizes the mean-squared error between predictions and labels,

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \big(f_\theta(x_i) - y_i \big)^2. \tag{119}$$

The rotation angles thus play the role of neural-network weights, and the entangling layers play the role of the nonlinear interactions between units.

Parameter-shift gradients. A QNN cannot be differentiated by backpropagation through the quantum state, because the state is not observable. Instead, the parameter-shift rule computes each partial derivative exactly from two extra runs of the circuit, with one angle shifted up and down by

Table 22: Magic-state response used in the multi-trace replay. Exponent and coefficient ranges are over three checkpoints; R_{MS} is the maximum coefficient, mean $\pm$ SD over three seeds.

Model	Exponent in p_{MS}	$\|b_{\mathrm{MS}}\|/p_{\mathrm{MS}}$	R_{MS}
Digits 4q/9L	0.9951–0.9986	418.7–3586.1	3586.1 ± 1823.0
Digits 6q/4L	0.9974–0.9989	262.4–649.0	649.0 ± 213.8
Digits 6q/6L	0.9968–0.9985	298.9–1478.0	1478.0 ± 662.8
Digits 6q/9L	0.9963–0.9977	427.3–2773.3	2571.3 ± 1002.6
Digits 8q/9L	0.9963–0.9969	451.9–1757.1	1757.1 ± 738.9
UNSW-NB15 6q/4L	0.9970–0.9987	525.1–746.0	746.0 ± 295.9
Waveform 6q/4L	0.9980–0.9988	145.1–427.1	427.1 ± 89.6

a quarter turn:

$$\frac{\partial f_\theta(x)}{\partial \theta_k} = \frac{1}{2}\Big[f_{\theta+\frac{\pi}{2}e_k}(x) - f_{\theta-\frac{\pi}{2}e_k}(x)\Big]. \tag{120}$$

The full gradient for one example therefore needs the original circuit plus two shifted circuits per parameter, and each of these circuits is itself estimated from many shots. This is the main difference from classical training. A single gradient step runs hundreds or thousands of circuits, each many times.

Why fault tolerance is needed. Physical qubits are noisy: every gate and measurement fails with some small probability, and the errors accumulate over a circuit. Current devices, often called noisy intermediate-scale quantum (NISQ) devices, run circuits directly on physical qubits and tolerate this noise. Fault-tolerant (FT) quantum computing instead encodes each logical qubit in many physical qubits using a quantum error-correcting code, such as the surface code (Fowler et al., 2012). Errors are detected and corrected continuously, so the logical error rate can be made as small as needed by enlarging the code. The size of the code is measured by its distance, and larger distances suppress logical errors exponentially at the cost of more physical qubits and time.

Clifford+T gates and rotation synthesis. Error-correcting codes can only implement a discrete set of logical gates fault-tolerantly. The standard choice is the Clifford+T set. Clifford gates, such as the Hadamard, phase, and controlled-Z gates, are comparatively cheap in the surface code. The T gate is a fixed rotation by one eighth of a turn; it is the only non-Clifford gate in the set, and it is the expensive one. A QNN, however, needs rotations by arbitrary angles. Such a rotation cannot be applied exactly and must be approximated by a sequence of Clifford and T gates, a step called gate synthesis. The approximation error is measured by the operator distance to the target rotation, up to an irrelevant global phase:

$$\min_{\alpha} \left\| \widetilde{U} - e^{i\alpha} R_Q(\varphi) \right\|_2 \le \epsilon. \tag{121}$$

The number of T gates needed grows with the requested precision. For almost all angles, the best possible sequences need about three T gates per bit of precision (Ross & Selinger, 2016),

$$\text{T count} \;\approx\; 3 \log_2 \frac{1}{\epsilon}. \tag{122}$$

Each tenfold tightening of the tolerance therefore costs roughly ten more T gates per rotation. Rotations about the Y axis are obtained from Z rotations by conjugating with Clifford gates, at no extra T cost.

Magic states and distillation. A T gate is not applied directly. It is carried out by consuming a special resource state, called a magic state, through a short teleportation circuit. Every executed T gate uses up one magic state, which cannot be reused. Magic states produced directly on hardware are too noisy, so they are purified in dedicated regions of the machine, called factories, by distillation. The standard 15-to-1 protocol (Bravyi & Kitaev, 2005) consumes fifteen noisy states and outputs one whose error is roughly cubic in the input error. Higher output fidelity needs larger codes or more rounds of distillation, and therefore more space and time. In many FT resource estimates, magic-state factories dominate the total cost (O'Gorman & Campbell, 2017; Litinski, 2019; Beverland et al., 2022). A magic state delivered with a small residual error acts as a small error on the T gate that consumes it, which in turn perturbs the circuit output and the gradient.

Why training multiplies FT cost. These costs compound during training. Each rotation in a circuit is synthesized at the chosen tolerance, so every circuit contains many T gates. Each update runs the base and shifted circuits for every example in the batch, and repeats each circuit for many shots. For the main circuit in our experiments, one update with a batch of 32 examples executes

$$32 \times (1 + 2 \times 108) = 6{,}944 \tag{123}$$

circuits before shot repetition, and each of their T gates consumes a fresh magic state. The synthesis tolerance is therefore paid for on every rotation, circuit, shot, and update. This is why the precision schedule, and not only the circuit design, determines the fault-tolerant cost of training a QNN.

F METHOD DETAILS

This section describes each component of FT-QNN at the level needed to reproduce it: how rotations are synthesized and their errors extracted, how the mixture is built, how resources are counted, how the controller chooses precision and fidelity, which error channels are simulated, and how magic-state demand is offloaded to the distillation server.

F.1 ROTATION SYNTHESIS AND ERROR GENERATORS

Every rotation in the circuit, both encoding and trainable, is synthesized separately for each circuit query. For a Z rotation by angle φ at tolerance ϵ, we call the Ross–Selinger algorithm (Ross & Selinger, 2016), implemented in `pygridsynth` 1.1.0, which returns a Clifford+T sequence $\widetilde{U}$ satisfying

$$\min_{\alpha} \left\| \widetilde{U} - e^{i\alpha} R_Z(\varphi) \right\|_2 \leq \epsilon. \tag{124}$$

Only Z rotations are synthesized. A Y rotation is obtained by conjugating a Z rotation with the Clifford gate $F = SH$,

$$R_Y(\varphi) = F\, R_Z(\varphi)\, F^\dagger = SH\, R_Z(\varphi)\, HS^\dagger, \tag{125}$$

which adds no T gates. Synthesized sequences are cached by angle and tolerance.

For each synthesized rotation, we extract its error in the form the theory uses. We remove the global phase, compute the error unitary relative to the target,

$$E = \widetilde{U}\, R_Z(\varphi)^\dagger, \tag{126}$$

and write it as a rotation about some axis,

$$E = \exp\Big[-\tfrac{i}{2} \left(a_x X + a_y Y + a_z Z\right) \Big], \qquad a = (a_x, a_y, a_z) \in \mathbb{R}^3. \tag{127}$$

The component along the intended rotation axis, a_z, is the tangent error: an over- or under-rotation. The other two components are transverse errors, which tilt the rotation axis. The phase is chosen so that the error angle is at most π, which makes the generator unique, and its length is at most about twice the synthesis tolerance (Lemma 3).

F.2 THREE-AXIS MIXED SYNTHESIS

The mixed construction replaces one deterministic circuit per rotation by a probability distribution over candidate circuits, chosen so that the average error generator is zero in all three directions. It has two stages: a linear program cancels the tangent component, and Clifford conjugation cancels the transverse components.

Candidates. For a generic angle φ and tolerance ϵ, we build three candidates: $R_Z(\varphi)$ synthesized at tolerance ϵ, and the two neighboring rotations $R_Z(\varphi - \epsilon)$ and $R_Z(\varphi + \epsilon)$, each synthesized at tolerance $\epsilon/4$. All three are evaluated against the original target $R_Z(\varphi)$. The offset candidates deliberately under- and over-rotate, so their tangent errors lie on opposite sides of zero, which makes a zero-mean mixture possible. Each offset candidate differs from the target by about $\epsilon/2$ from the angle offset plus at most $\epsilon/4$ from synthesis, so every candidate stays within operator distance ϵ of the target. Angles that are exact multiples of $\pi/4$ are synthesized exactly with a single candidate.

Tangent component: a linear program. We choose mixture weights that make the average tangent error zero while using as few T gates as possible:

$$\min_{w \geq 0} \sum_k w_k \, \mathrm{T}_k \qquad \text{subject to} \qquad \sum_k w_k = 1, \qquad \sum_k w_k \, a_{z,k} = 0, \tag{128}$$

where candidate k has T count T_k and tangent error $a_{z,k}$. There are no other constraints. If the program is infeasible, the synthesis step stops with an error rather than substituting a deterministic circuit, so every mixed rotation in our runs is exactly tangent-centered.

Transverse components: Clifford conjugation. The phase gate S commutes with $R_Z(\varphi)$, so conjugating a candidate by S leaves the intended rotation unchanged. It rotates the transverse part of the error by a quarter turn about the Z axis:

$$S\,E\,S^\dagger = \exp\Big[-\tfrac{i}{2}\,(-a_y X + a_x Y + a_z Z)\Big]. \tag{129}$$

For every candidate, we apply the four conjugations $S^k U S^{-k}$, $k = 0, 1, 2, 3$, each with probability $1/4$, so candidate k receives final weight $w_k/4$. The four quarter-turns average the transverse components to zero, while the tangent component and the T count are unchanged, since S is a Clifford gate. The final distribution has at most $3 \times 4 = 12$ entries, or 4 for an exactly synthesizable angle. Over all synthesized cases, the largest remaining mean generator was below 3×10^{-16} times the tolerance.

Tangent-only ablation. The tangent-only baseline in RQ1 uses the LP weights without the conjugations. Its mean tangent error is zero, but its mean transverse error is not.

F.3 Resampling policies and T-count accounting

Under fixed-query (FQ) synthesis, one entry of the mixture is drawn for each rotation in each circuit query and reused for all shots of that query. Under per-shot (PS) synthesis, a new entry is drawn for each rotation in every shot. In both policies, the base circuit and every shifted circuit draw independently, so their errors are uncorrelated.

The T count of a circuit is the sum of the T counts of all its synthesized rotations, including encoding rotations. The pre-shot workload of an update sums over every circuit it executes,

$$W_t = \sum_{\text{circuits in update } t} \; \sum_{\text{rotations } j} \mathrm{T}(\widetilde{U}_j), \tag{130}$$

and the executed workload multiplies each circuit by the number of shots, $M_t = S\,W_t$. For deterministic and FQ synthesis this count is exact. For PS synthesis, each shot uses a different entry, so we report the expected T count under the mixture weights. The cumulative consumption is the sum of the executed workloads over updates. Every executed T gate consumes one magic state, so the cumulative count also measures magic-state consumption.

F.4 Convergence-aware controller

Convergence level. At every update, including update 0, the controller computes the exact loss and gradient norm on the full training set, which we call the oracle monitor. In our experiments, the bias budget is set directly from this norm,

$$\Delta_{b,t} = \rho \, \|\nabla L(\theta_t)\|, \qquad \rho = 0.1, \tag{131}$$

which corresponds to $g_\Delta = 0$ in the schedule of Section 4. The floor $g_\Delta > 0$ is needed only for the convergence proof. In practice, the tolerance grid supplies the floor: no tolerance below 10^{-6} is ever requested.

Tolerance grid. Tolerances are chosen from the fixed grid

$$\big\{10^{-1},\, 3{\times}10^{-2},\, 10^{-2},\, 8{\times}10^{-3},\, 6{\times}10^{-3},\, 5{\times}10^{-3},\, 4{\times}10^{-3},\, 3{\times}10^{-3},\\ 10^{-3},\, 3{\times}10^{-4},\, 10^{-4},\, 3{\times}10^{-5},\, 10^{-5},\, 3{\times}10^{-6},\, 10^{-6}\big\}, \tag{132}$$

with an upper cap, usually 3×10^{-3}, that bounds the tolerance even when the budget would allow more. This cap plays the role of ϵ_{loc}, keeping every tolerance in the regime where the local expansion is accurate. Synthesis error directions vary between tolerances, so the bias is not strictly monotone in ϵ. To stay conservative, a tolerance is accepted only if it and every finer grid value meet the budget.

Bias budget. The controller predicts the synthesis bias with two constants measured before training,

$$\|b_{\mathrm{det}}(\epsilon)\| \le A\,\epsilon, \qquad \|b_{\mathrm{mix}}(\epsilon)\| \le B_2\,\epsilon^2, \qquad A = 75, \quad B_2 = 35. \tag{133}$$

They are the measured counterparts of Λ_1 and Λ_2 in Lemmas 1 and 2, and include the response accumulated across all gates and queries of an update. A tolerance is admissible if the predicted synthesis bias plus the magic-state bias fits the budget:

$$A\,\epsilon + R_{\mathrm{MS}}\,p_{\mathrm{MS}} \le \Delta_{b,t} \quad \text{(deterministic)}, \qquad B_2\,\epsilon^2 + R_{\mathrm{MS}}\,p_{\mathrm{MS}} \le \Delta_{b,t} \quad \text{(mixed)}. \tag{134}$$

Variance rule for Adam. The variance budget of Section 4 is stated for SGD. For our Adam runs, we enforce an empirical rule on the next update instead. A mixed tolerance is admitted only if the perturbation it causes in the next Adam update is no larger than the perturbation caused by the setting's own shot and mini-batch noise. For per-shot synthesis, the total perturbation must be at most twice the noise alone. If no tolerance satisfies both the bias budget and the variance rule, the run stops and is reported as infeasible; this did not occur in the reported runs.

Joint selection. In the RQ4 runs, the controller also chooses the magic-state fidelity. It searches every combination of mechanism, grid tolerance, and fidelity tier $p_{\mathrm{MS}} \in \{10^{-6}, 3\times10^{-7}, 10^{-7}, 3\times10^{-8}\}$ that meets the bias budget, with the conservative $R_{\mathrm{MS}} = 2571.3$, and the variance rule. It picks the one that minimizes

$$n_m(\epsilon)\,\log_{10}\frac{1}{p_{\mathrm{MS}}}, \tag{135}$$

the expected T count per rotation weighted by the fidelity demand. This proxy only ranks actions; the physical cost of the chosen fidelity is evaluated separately by the server model. Algorithm 1 summarizes one update.

Algorithm 1 One FT-QNN update

1: Compute $\|\nabla L(\theta_t)\|$ with the oracle monitor and set $\Delta_{b,t} = \rho\|\nabla L(\theta_t)\|$
2: **for** each action $(m, \epsilon, p_{\mathrm{MS}})$ on the grid **do**
3: Check the bias budget with $A\epsilon$ (deterministic) or $B_2\epsilon^2$ (mixed), plus $R_{\mathrm{MS}}p_{\mathrm{MS}}$
4: If m is mixed, check the variance rule
5: **end for**
6: Choose the feasible action minimizing $n_m(\epsilon)\log_{10}(1/p_{\mathrm{MS}})$; stop if none is feasible
7: Synthesize every rotation of every base and shifted query at (m, ϵ)
8: Send the magic-state request, count and fidelity, to the distillation server
9: Execute all queries for S shots and form the parameter-shift gradient
10: Take one Adam step

F.5 Error channels in simulation

Synthesis error. Synthesis error is simulated exactly: every circuit uses the actual compiled Clifford+T sequence for each rotation, not a noise model.

Magic-state error. A magic state delivered with infidelity

$$p_{\mathrm{MS}} = 1 - \langle A|\rho_A|A\rangle \tag{136}$$

perturbs the T gate that consumes it. In the RQ4 training runs, every compiled T gate applies the exact averaged injection channel built from a Z-twirled noisy magic state, including the feed-forward correction. This channel is equivalent to a Pauli channel that applies the ideal T gate with probability $1 - p_{\mathrm{MS}}$ and ZT with probability p_{MS}. The response study (Appendix D.1) compared three models: this Pauli proxy, explicit injection with a Z-twirled resource state, and explicit injection with an isotropically depolarized resource state. The first two agreed end to end to within 1.7×10^{-12}.

Residual logical noise. Logical Clifford gates are treated as error-corrected, and no residual logical depolarizing channel is simulated, so $p_{\mathrm{dep}} = 0$ in all experiments. The term $R_{\mathrm{dep}} p_{\mathrm{dep}}$ in the theory shows how such noise would share the bias budget. The depolarized resource state in the response study is a model of magic-state preparation, not of gate noise. Physical noise enters only through the distillation model, via the physical error rate p_{phys} of the surface code.

F.6 Distillation-server offloading

For each update, the server receives the number of magic states requested at each fidelity, equal to the executed T gates of that update. It serves each request from a factory configured for that fidelity.

Factory model. Each factory runs one level of 15-to-1 distillation (Bravyi & Kitaev, 2005) with input error $p_{\mathrm{in}} = 5 \times 10^{-4}$, output error $35 p_{\mathrm{in}}^3$, and success probability $1 - 15 p_{\mathrm{in}}$. It runs on surface-code patches of distance d with logical error per cycle

$$p_L = 0.1\,(100\,p_{\mathrm{phys}})^{(d+1)/2}. \tag{137}$$

A requested fidelity is met if the output error, including logical errors during distillation, satisfies

$$35 p_{\mathrm{in}}^3 + V\,p_L \le p_{\mathrm{MS}}, \tag{138}$$

with exposure V. For each fidelity, the server uses the smallest odd distance that meets it. The normalized cost of one output state is

$$\frac{15\,d^3}{1 - 15 p_{\mathrm{in}}}, \tag{139}$$

and demand is measured in these normalized, high-fidelity-state-equivalent units per update interval.

Buffering and capacity. Because the fidelity schedule follows the gradient, requests can be predicted ahead of time. The server starts with a buffer of one update's production and may carry inventory forward between updates. For per-update demands $d_1, \ldots, d_N$, the smallest sustained capacity that serves every update without a stall is

$$C_{\min} = \max_{1 \le k \le N} \frac{\sum_{t=1}^{k} d_t}{k+1}, \tag{140}$$

where the $+1$ counts the initial one-update buffer. We also report the total normalized work, the peak capacity needed by the busiest single update, the stall overhead when capacity is fixed at 80% of the fixed-high peak, and the number of physical reconfigurations, since neighboring fidelity tiers often share a code distance.

Pooling. A flexible shared pool reassigns factories between code distances as requests change. Separate pools reserve factories for each distance. The flexible pool needs less peak capacity but pays a delay at each reconfiguration, and we report the delay below which it remains preferable.

G Extended related work

This section expands Section 2 and states, for each line of work, what it establishes and how FT-QNN differs. Table 23 summarizes the comparison.

G.1 Gate synthesis and mixed synthesis

Deterministic synthesis. Ross and Selinger give an optimal ancilla-free algorithm for approximating Z rotations in Clifford+T, with a T count of about $3 \log_2(1/\epsilon)$ for generic angles and a matching lower bound for almost all angles (Ross & Selinger, 2016). We use their algorithm for every deterministic circuit and every mixture candidate, and their lower bound underlies Assumption 2. Their work fixes the accuracy in advance. It does not ask what accuracy a downstream computation needs, nor how the error affects a learning algorithm.

Mixed synthesis. Hastings (2017) and Campbell (2017) show that choosing randomly among several approximations of the same unitary turns coherent error into incoherent error and suppresses the mean channel error quadratically. Kliuchnikov et al. (2023) optimize such mixtures to shorten circuits, and Yoshioka et al. (2025) study how to shape the residual error channel that mixing leaves behind. These works establish the mechanism we build on. They analyze the error of a gate or a circuit at a prescribed accuracy. We analyze how the error reaches a training gradient, show that in this setting all three components of the error generator must be centered, derive how the tolerance may loosen as a consequence, and study how often candidates should be redrawn within a training update.

Randomized compilation. Martyn & Rall (2025) show that randomization halves the logarithmic cost of approximating functions with quantum signal processing, by turning a first-order approximation error into a second-order one. Theorem 1 rests on the same first-to-second-order change. Their setting is a fixed algorithm with a fixed target accuracy. Ours is an optimization loop whose admissible error changes with every update, which is why our result is stated at the convergence floor and must also account for how mixing changes the training path.

G.2 Fault-tolerant variational algorithms and QML

Error models in QNN training. Most QNN studies treat parameterized rotations as native gates and model their imperfections as perturbations of the rotation angles or as generic device noise (Wang et al., 2022; Rabinovich et al., 2024). This is appropriate for NISQ hardware. On FT hardware, however, the dominant rotation error is the synthesis error, which is a rotation about all three axes rather than an angle error. Our RQ1 results show that correcting only the angle leaves the gradient bias first order.

Fault-tolerant variational algorithms. Sayginel et al. (2024) compile VQE rotations into Clifford+T with Ross–Selinger synthesis, study the resulting T count, and show that adapting the synthesis accuracy during optimization preserves convergence. This is the closest prior work to ours and motivates our scheduled deterministic reference. Their adaptation is not derived from a convergence condition, they use deterministic synthesis only, and they do not consider magic-state fidelity or provisioning. FT-QNN derives the tolerance from a gradient-bias budget, gives mechanism-specific tolerance laws with cost bounds and a convergence guarantee, and extends the same budget to magic-state fidelity.

Partial error correction. Kang et al. (2026) reduce the overhead of FT QML by error-correcting Clifford gates while leaving parameterized rotations without distillation, and show that models remain trainable with little loss of accuracy. Their approach removes the T cost of rotations but leaves a fixed error floor on them. FT-QNN keeps every rotation fault tolerant and instead lowers its cost by matching precision to the convergence state. The two approaches address different regimes: a fixed rotation error suits early training and coarse classification, while our budgets show that the admissible error must keep shrinking as training approaches stationarity.

G.3 Noisy optimization and low-precision training

Optimization with biased and noisy gradients. Nonconvex stochastic optimization converges to stationarity under bounded gradient variance and under gradient bias that is controlled relative to the gradient norm (Ajalloeian & Stich, 2020; Demidovich et al., 2023; Hu et al., 2021), and related guarantees exist for Adam (Défossez et al., 2020). We use this theory as a tool. Our contribution is to identify the bias and variance that FT execution produces, turn their budgets into synthesis tolerances and magic-state fidelities, and extend the guarantee to momentum and Adam-type scaling under these FT error sources.

Low-precision training. Stochastic rounding makes quantization error unbiased, so training still converges at low numerical precision (Gupta et al., 2015), and unbiased gradient quantization reduces communication with convergence guarantees (Alistarh et al., 2017). Three-axis mixed synthesis plays a similar role for FT circuits. It differs in three ways: its error is a rotation in three dimensions rather than a scalar rounding error, its cost grows logarithmically in the precision rather

than linearly in the number of bits, and its precision is scheduled by the convergence state rather than fixed.

G.4 Magic-state distillation and FT resource estimation

Distillation and factories. The 15-to-1 protocol of Bravyi & Kitaev (2005) is the basis of most magic-state factories. Surface-code resource analyses (Fowler et al., 2012; O'Gorman & Campbell, 2017; Litinski, 2019) and end-to-end estimates (Beverland et al., 2022) show that factories often dominate the cost of FT algorithms, and design factories for a fixed target output fidelity. We use a conditional one-level 15-to-1 model with the standard surface-code error heuristic. Our contribution is not a new factory design. It is the observation that learning produces a time-varying fidelity demand, and the evaluation of how a shared multi-fidelity server can exploit it.

G.5 Summary

Existing work studies the individual components of the problem: synthesis at a given accuracy, optimization under generic error, and resource provisioning for a given circuit. FT-QNN connects them by deriving the required accuracy from the learning process and carrying it through synthesis cost, magic-state fidelity, and distillation demand.

Table 23: Comparison with related work. Precision from learning: the synthesis accuracy is derived from the optimization state. Three-axis: error is modeled and cancelled in all three generator directions. Fidelity: magic-state fidelity is chosen by the method. Guarantee: a convergence or cost guarantee for training.

Work	Clifford+T	Mixed	Three-axis	Precision from learning	Fidelity	Guarantee
Ross & Selinger (2016)	✓					
Hastings (2017); Campbell (2017)	✓	✓				
Kliuchnikov et al. (2023); Yoshioka et al. (2025)	✓	✓				
Martyn & Rall (2025)		✓				
Wang et al. (2022); Rabinovich et al. (2024)						
Sayginel et al. (2024)	✓			Heuristic		
Kang et al. (2026)	Partial					
Bravyi & Kitaev (2005); Litinski (2019); Beverland et al. (2022)	✓				Fixed	
FT-QNN (ours)	✓	✓	✓	✓	Scheduled	✓