

Error Attribution as a Resource-Allocation Principle for Quantum Error Correction: Sensitivity-Guided Noise Reduction and Its Analytic Limits

Error Attribution as a Resource-Allocation Principle for Quantum Error Correction: Sensitivity-Guided Noise Reduction and Its Analytic Limits

Abstract

Logical error rate is the standard benchmark for quantum error correction (QEC), but it is an aggregate quantity: it says nothing about which circuit components actually drive logical failure. Recent work introduced an error attribution scheme that computes per-component sensitivities $\partial P_L / \partial p_i$ of the logical error rate jointly from the same Monte Carlo samples used to estimate P_L , and showed that halving noise on the 5%–7% most sensitive components reduces logical error rates by roughly 15%–25%, about twice the improvement from randomly chosen components [1],[3]. This paper develops the quantitative structure underlying that result. We formalize the sensitivity map, derive the first-order intervention gain for targeted versus random noise-reduction budgets, and show analytically that the targeted-to-random advantage is $\rho = r / (fr + 1 - f)$, where f is the hotspot fraction and r the sensitivity contrast; for $f = 0.06$ and $r = 2.136$ this ratio is exactly 2, matching the reported factor. We further show that the reported 15%–25% reductions exceed the first-order prediction of about 6% by an amplification factor of approximately 3.15, which we attribute to nonlinear, multiplicative error propagation absent from independent-error models. We derive Monte Carlo sample-complexity requirements for the sensitivity estimator, situate attribution among channel-optimized, reinforcement-learning, and robustness-optimized QEC strategies, and identify failure modes: coherent noise, estimator variance at low P_L , sensitivity redistribution, and non-stationary drift.

1. Introduction

Quantum error correction protects quantum information by encoding logical qubits redundantly and decoding measured syndromes, and its performance is almost universally reported as a single number: the logical error rate P_L , the probability that a logical operator flips during a memory experiment or computation cycle. This aggregate benchmark is necessary but insufficient for engineering. Two circuit locations may carry identical physical error probabilities while contributing wildly different amounts to logical failure — a data qubit idling across many syndrome-extraction rounds is not equivalent to an ancilla used once — yet a standard P_L estimate is blind to this asymmetry.

The practical consequence is a resource-allocation problem. Experimental groups face finite budgets of calibration effort, gate-tuning time, and materials engineering. If logical failure is concentrated in a small set of "hotspot" components, then interventions targeted by sensitivity should outperform uniform or random improvement by a large factor. Recent work [1],[3] made this precise by introducing an error attribution estimator that computes all component sensitivities jointly from the Monte Carlo samples already being collected for P_L estimation, at essentially no additional simulation cost. In circuit simulations of rotated surface code memories and high-rate lifted-product codes, halving the noise strength on about 5%–7% of physical components selected by sensitivity reduced logical error rates by approximately 15%–25%, roughly twice the mean reduction from the same intervention applied to an equal number of randomly selected components.

This paper treats error attribution not merely as a simulation technique but as a resource-allocation principle, and asks three questions. First, what analytic structure governs the targeted-versus-random advantage, and under what sensitivity distributions does the reported factor of two arise? Second, why do the reported 15%–25% reductions exceed what a first-order, independent-error model predicts, and what does that gap tell us about error propagation in real decoding circuits? Third, what sample complexity does the sensitivity estimator require, and where can it fail? We answer these with explicit derivations on simplified but faithful models, and we position attribution within the broader landscape of noise-aware QEC optimization [2],[4],[5],[8].

Our contributions are: (i) a closed-form expression for the targeted-to-random gain ratio under a first-order intervention model, with full arithmetic; (ii) an amplification-factor analysis reconciling first-order predictions with reported nonlinear gains; (iii) a sample-complexity derivation for the joint sensitivity estimator; and (iv) a critical assessment of failure modes, including coherent noise [6], code-family dependence [7],[11], and the tension between optimized QEC and the robustness required at scale [5].

2. Background and Related Work

The immediate foundation is the error attribution scheme of [1],[3], which defines a per-component sensitivity $s_i = \partial P_L / \partial p_i$ and estimates all s_i simultaneously from standard Monte Carlo samples via a score-function decomposition of the sampled error configurations. The key efficiency claim is that no additional simulation passes are needed: the same samples that estimate P_L also estimate the full sensitivity map. The reported headline result — 15%–25% logical error reduction from halving noise on 5%–7% of components, roughly twice the random-selection baseline — is the empirical anchor for the analytic models developed in Sections 3 and 4 of this paper.

Noise-aware optimization has a longer history. Channel-optimized QEC [2] developed a theory of correction procedures tailored to a given noise channel, including robustness against uncertainty in that channel, and demonstrated numerically that optimized procedures always exceed the fidelity of noise-agnostic standard correction. Error attribution is complementary: rather than redesigning the code for a known channel, it identifies which parts of a fixed architecture deserve intervention first. Robustness-optimized QEC [5] makes the complementary point that in small-scale devices, where dominant error sources are understood, exploiting that knowledge yields more efficient correction than source-agnostic design; sensitivity maps are precisely the quantitative instrument for such exploitation at component granularity.

Reinforcement-learning approaches attack the same optimization problem with search rather than sensitivity. An RL framework for optimizing and fault-tolerantly adapting QEC codes when the error channel is unknown treats decoder and encoding choices as actions [4], and deep reinforcement learning has been extended to discover autonomous QEC encodings for bosonic systems under Knill–Laflamme constraints [9]. Both are powerful but sample-hungry black-box optimizers; attribution provides a gradient-like signal that could seed or constrain such searches, reducing the search dimensionality from the full circuit space to the sensitivity-ranked subspace.

Gauge-optimization work [8] identifies gauge freedoms in QEC codes and uses optimal-control algorithms to find decompositions into elementary operations easiest to realize on a given physical system. This is structurally analogous to attribution: both ask "given this hardware, which degrees of freedom of the code matter most?", though [8] optimizes implementation decomposition while attribution optimizes noise budget allocation.

Erasure-qubit protocols [7] analyze how converting unknown errors into flagged erasures improves surface code performance, at the cost of erasure-check operations that add noise and runtime. The trade-off analysis is structurally similar to attribution economics — one pays overhead at specific circuit locations to buy disproportionate logical gains — and attribution offers a principled way to decide whether those added operations pay for themselves, by comparing the sensitivity mass they remove against the sensitivity mass they introduce.

On the modeling side, coherent-versus-stochastic comparisons [6] of $d = 3$ Steane and surface code circuits found markedly different failure distributions, with no simple mapping between the models. This is a direct caution for attribution: sensitivity maps derived under a stochastic Pauli noise model may misrank components when the dominant noise is coherent, since coherent errors can interfere and cancel in ways stochastic errors cannot. Continuous-time QEC [10] treats noise and correction as continuous processes via weak measurement and feedback; in that setting the discrete notion of a "circuit component" must be replaced by a time- or channel-resolved attribution, an open generalization. Entanglement-assisted codes [11] expand the code-design space with pre-shared entanglement; attribution applies unchanged, but the component set grows to include the ebit-consuming gates.

Finally, survey treatments of QEC's implications for computing [12] situate logical error rate benchmarking within broader fault-tolerance economics, and the QEC–Darwinism tradeoff result [13] — that quantum error correction and Quantum Darwinism cannot coexist above a critical logical fidelity $F_L > 0.874$ — suggests that pushing P_L down via targeted intervention operates near fundamental information-theoretic boundaries, not merely engineering ones. The ultrametric-geometry proposal for passive error resilience [14] raises the possibility that some error suppression could be structural rather than calibration-driven; attribution would then quantify the residual, actively correctable error budget.

3. Methods

3.1 Setup and notation

Consider a QEC memory circuit with n_c noise locations ("components"), where component $i \in \{1, \dots, n_c\}$ suffers an error with probability p_i per relevant time step. The logical error rate is

$$P_L(\mathbf{p}) = \Pr(\text{decoder output differs from the encoded logical state}),$$

a function of the noise vector $\mathbf{p} = (p_1, \dots, p_{n_c})$. The first-order sensitivity of component i is

$$s_i \equiv \left. \frac{\partial P_L}{\partial p_i} \right|_{\mathbf{p}}.$$

The first-order expansion $P_L \approx \sum_i s_i p_i$ is valid when $\sum_i p_i \ll 1$ and logical failure is dominated by single-component fault paths; higher-order terms scale as $O((\sum_i p_i)^2)$.

3.2 Component census for the rotated surface code

For a rotated surface code of distance $d = 3$ (9 data qubits, 8 ancilla qubits) run for $R = 4$ syndrome-extraction rounds, a standard phenomenological circuit model counts per round: 9 data-qubit idle locations, $4 \times 8 = 32$ two-qubit gate error locations (each ancilla touches 4 data qubits), 8 ancilla preparation locations, and 8 measurement locations, giving 57 locations per round and

$$n_c = 4 \times 57 = 228.$$

The reported hotspot fraction of 5%–7% [1],[3] therefore corresponds to $0.05 \times 228 \approx 11$ to $0.07 \times 228 \approx 16$ components.

3.3 Intervention model

An intervention halves the noise strength on a selected set S of m components: $p_i \mapsto p_i/2$ for $i \in S$. To first order,

$$\Delta P_L(S) = P_L(\text{after}) - P_L(\text{before}) \approx -\frac{1}{2} \sum_{i \in S} p_i s_i.$$

The fractional reduction is

$$G(S) = \frac{-\Delta P_L}{P_L} = \frac{1}{2} \frac{\sum_{i \in S} p_i s_i}{P_L}.$$

We analyze the canonical case of uniform baseline noise $p_i = p$ and a two-class sensitivity distribution: a fraction f of components ("hotspots") have sensitivity $s_{\text{hot}} = r s$, and the remaining $1 - f$ have $s_{\text{cold}} = s$. Then

$$P_L = n_c p s (1 + f(r - 1)).$$

3.4 Targeted versus random selection

With m components selected, targeted selection picks hotspots:

$$G_{\text{tgt}} = \frac{m r p s}{2 n_c p s (1 + f(r - 1))} = \frac{m r}{2 n_c (1 + f(r - 1))}.$$

Random selection picks hotspots in expectation $m f$ and cold components $m(1 - f)$:

$$G_{\text{rnd}} = \frac{m(f r + (1 - f))}{2 n_c (1 + f(r - 1))}.$$

The advantage ratio is therefore

$$\rho \equiv \frac{G_{\text{tgt}}}{G_{\text{rnd}}} = \frac{r}{f r + 1 - f}.$$

Equivalently, in the attribution-share language where $f_i = s_i p_i / \sum_j s_j p_j$ is the fraction of the logical error budget attributable to component i , the ratio ρ is the ratio of selected to population-mean sensitivity; the entire game is the sensitivity skew, independent of p , m , and n_c .

3.5 Joint sensitivity estimator and its variance

Following [1],[3], the estimator reuses Monte Carlo samples drawn for P_L . With N samples, logical failures flagged by indicator $Y_j \in \{0, 1\}$, and per-sample component-occupancy statistics, the plug-in estimators are $\hat{P}_L = \frac{1}{N} \sum_{j=1}^N Y_j$ and $\hat{s}_i$ constructed from the joint counts of component- i errors and logical failure. The variance of $\hat{P}_L$ is

$$\text{Var}(\hat{P}_L) = \frac{P_L(1 - P_L)}{N}.$$

3.6 Numerical inputs and their sources

All numerical inputs used in Section 4 are: (a) hotspot fraction $f \in [0.05, 0.07]$ and targeted-to-random ratio $\rho \approx 2$, from the reported simulations [1],[3]; (b) reported logical error reductions of 15%–25% [1],[3]; (c) component census $n_c = 228$ for $d = 3$, $R = 4$, derived above from standard rotated-surface-code structure; (d) a representative physical error rate $p = 10^{-3}$ per component per round, a standard phenomenological operating point; (e) Monte Carlo sample count $N = 10^6$, a typical simulation budget; and (f) a representative logical error rate $P_L = 10^{-3}$ at that operating point, consistent with below-threshold surface-code behavior. Items (c)–(f) are modeling assumptions of this paper, not measurements, and are labeled as such wherever used.

4. Analysis

4.1 The targeted-to-random advantage ratio

From Section 3.4,

$$\rho = \frac{r}{fr + 1 - f}.$$

Setting $\rho = 2$ (the reported factor [1],[3]) and solving for the sensitivity contrast r as a function of hotspot fraction f :

$$2 = \frac{r}{fr + 1 - f} \Rightarrow 2fr + 2(1 - f) = r \Rightarrow r(1 - 2f) = 2(1 - f) \Rightarrow r = \frac{2(1 - f)}{1 - 2f}.$$

At the lower end of the reported hotspot range, $f = 0.05$:

$$r = \frac{2 \times 0.95}{1 - 0.10} = \frac{1.90}{0.90} \approx 2.111.$$

At the upper end, $f = 0.07$:

$$r = \frac{2 \times 0.93}{1 - 0.14} = \frac{1.86}{0.86} \approx 2.163.$$

So the reported factor-of-two advantage is explained by a remarkably modest sensitivity contrast: hotspots need only be about 2.1 to 2.2 times more influential than typical components, provided they occupy 5%–7% of the circuit. Conversely, solving $2(fr + 1 - f) = r$ for f gives $f = \frac{r-2}{2r-1}$; for $r = 3$: $f = \frac{1}{5} = 0.20$. A sharper contrast requires a smaller hotspot fraction to keep $\rho = 2$; both regimes are consistent with the reported 5%–7% band given moderate contrast.

4.2 First-order absolute gains

Take the midpoint parameters: $n_c = 228$, $f = 0.06$, $r = \frac{2(1-0.06)}{1-0.12} = \frac{1.88}{0.88} \approx 2.136$ (chosen so that $\rho = 2$ exactly at $f = 0.06$; check: $fr + 1 - f = 0.06 \times 2.136 + 0.94 = 0.1282 + 0.94 = 1.0682$, and $\rho = 2.136/1.0682 = 2.000$). With $p = 10^{-3}$ and unit cold sensitivity $s = 1$ in arbitrary units:

$$P_L = n_c p s (1 + f(r - 1)) = 228 \times 10^{-3} \times 1.0682 \approx 0.2435 s.$$

(We treat P_L here in the linearized model's units; the absolute normalization is absorbed into s , and only fractional quantities are physical.)

Targeted intervention on $m = 14$ components (6.1% of 228, within the reported 5%–7% band):

$$G_{\text{tgt}} = \frac{14 \times 2.136}{2 \times 228 \times 1.0682} = \frac{29.904}{487.10} \approx 0.0614.$$

Random selection of the same $m = 14$:

$$G_{\text{rnd}} = \frac{14 \times 1.0682}{2 \times 228 \times 1.0682} = \frac{14}{456} \approx 0.0307.$$

Check of the ratio: $\rho = 0.0614/0.0307 = 2.00$, as constructed.

4.3 The amplification factor: first order versus reported gains

The first-order model predicts a targeted reduction of $G_{\text{tgt}} \approx 6.1\%$, but the reported simulations [1],[3] show 15%–25%. Taking the geometric midpoint of the reported range, $\sqrt{0.15 \times 0.25} = \sqrt{0.0375} \approx 0.1936$, i.e., about 19.4%, the amplification factor is

$$A = \frac{0.1936}{0.0614} \approx 3.15.$$

At the range endpoints: $A_{\min} = 0.15/0.0614 \approx 2.44$ and $A_{\max} = 0.25/0.0614 \approx 4.07$. We interpret A as the signature of nonlinear error propagation: in a real decoding circuit, an error on a sensitive component is not merely directly logical; it also corrupts syndromes that mislead the decoder about other errors, so removing it eliminates both its direct contribution and an induced contribution on downstream components. A simple multiplicative model captures this: if each hotspot's effective first-order contribution is amplified by the average number of syndrome rounds its corruption influences, $A \approx \bar{c}$ with $\bar{c}$ the mean propagation fan-out. Our derived range $A \in [2.4, 4.1]$ is consistent with a fan-out of 2–4 syndrome rounds, plausible for $R = 4$ -round memories where a single early data-qubit error can be misidentified across multiple rounds.

An alternative reading, which we record in Appendix A, is that the reported gains reflect a stronger sensitivity skew than our two-class model assumes: in a single-hotspot parameterization, reproducing a 20% reduction requires the top components to carry a large share of the error budget (e.g., a $\sim 10\times$ skew on $\sim 7\%$ of components in a 100-component model). Both readings agree that the first-order two-class model with moderate contrast under-predicts the absolute gains; they differ on whether the missing factor is propagation nonlinearity or skew strength.

4.4 Sample complexity of the joint estimator

With $N = 10^6$ samples and $P_L = 10^{-3}$:

$$\text{Var}(\hat{P}_L) = \frac{10^{-3} \times (1 - 10^{-3})}{10^6} = \frac{9.99 \times 10^{-4}}{10^6} = 9.99 \times 10^{-10},$$

so the standard error is $\sigma(\hat{P}_L) = \sqrt{9.99 \times 10^{-10}} \approx 3.16 \times 10^{-5}$, a relative precision of $3.16 \times 10^{-5}/10^{-3} \approx 3.2\%$.

For the sensitivity estimator, the score-function construction [1],[3] effectively correlates the failure indicator with component occupancy; its variance scales as $\text{Var}(\hat{s}_i) \sim P_L/N$ up to a condition factor set by how often component i fires. For a component firing with per-sample probability q_i , the effective sample count for s_i is $N_i = Nq_i$, giving

$$\sigma(\hat{s}_i) \approx \sqrt{\frac{P_L}{Nq_i}} = \sqrt{\frac{10^{-3}}{10^6 q_i}} = \sqrt{\frac{10^{-9}}{q_i}}.$$

For a gate-error location with $q_i \approx 4 \times 10^{-3}$: $\sigma(\hat{s}_i) \approx \sqrt{10^{-9}/(4 \times 10^{-3})} = \sqrt{2.5 \times 10^{-7}} \approx 5.0 \times 10^{-4}$. Relative to a hotspot sensitivity of order $s_{\text{hot}} \approx 2.1$ (in units where the mean is 1), the relative error is $\approx 5.0 \times 10^{-4}/2.1 \approx 2.4 \times 10^{-4}$ — negligible. For a rare component with $q_i = 10^{-5}$: $\sigma(\hat{s}_i) \approx \sqrt{10^{-9}/10^{-5}} = \sqrt{10^{-4}} = 10^{-2}$, a relative error of $\approx 0.5\%$ — still acceptable. The estimator is therefore statistically cheap at these operating points; the cost grows only when P_L falls below $\sim 10^{-5}$, where $\sigma(\hat{P}_L)/P_L = \sqrt{(1 - P_L)/(NP_L)} \approx 1/\sqrt{NP_L}$ exceeds 10% unless $N \gtrsim 10^4/P_L = 10^9$ samples at $P_L = 10^{-5}$.

A conservative bound reinforces this picture: with attribution weights bounded by $w_i \leq 1$ and component error probability p_i , $\text{Var}(\hat{s}_i) \leq \frac{P_L}{Np_i^2}$; at $P_L = 1.10 \times 10^{-2}$, $N = 10^5$, $p_i = 10^{-4}$ this gives $\text{Var}(\hat{s}_i) \leq 11.0$, i.e., $\sigma(\hat{s}_i) \leq 3.3$ — far too noisy to rank individual components at $N = 10^5$, adequate only at $N = 10^7$ ($\sigma(\hat{s}_i) \leq 0.33$). The discrepancy between this bound and the optimistic estimate above is governed by the true attribution-weight distribution; we flag it as a modeling uncertainty in Appendix A.

4.5 Intervention budget arithmetic

At $n_c = 228$ and the reported 5%–7% band, the intervention touches $m \in [11.4, 16.0]$, i.e., 12 to 16 components. For the $d = 3$ rotated code this is a physically small set — comparable to the number of data qubits (9) plus a few ancillas — suggesting the hotspots concentrate on data-qubit idling locations and specific gate layers, consistent with the geometric intuition that central data qubits participate in more logical operators than peripheral ones.

4.6 Equivalent physical error budgets and distance scaling

For surface codes below threshold, $p_L \sim A_d (p/p_{\text{th}})^{(d+1)/2}$; for $d = 3$ the exponent is $(3 + 1)/2 = 2$. If attribution-guided intervention reduces p_L by a factor $(1 - r)$, the equivalent uniform physical error reduction factor is

$$\frac{p_{\text{eff}}}{p} = (1 - r)^{1/2}.$$

For $r = 0.20$: $(1 - 0.20)^{1/2} = \sqrt{0.8} = 0.894$, i.e., a 20% logical reduction is equivalent to a uniform $\approx 10.6\%$ physical error rate reduction — achieved by touching only 5%–7% of components. For our model's $r = 0.068$: $\sqrt{0.932} = 0.9654$, a 3.5% equivalent uniform reduction. At $d = 5$ the fractional gain is unchanged (it depends only on shares), but the absolute gain shrinks proportionally with p_L . Near threshold (e.g., $p/p_{\text{th}} = 0.5$, giving $p_L \approx 2.5 \times 10^{-2}$ for $d = 3$ with $A_d = 0.1$), higher-order terms $\frac{1}{2} \sum_{i,j} \frac{\partial^2 P_L}{\partial p_i \partial p_j} \Delta p_i \Delta p_j$ are no longer negligible and first-order predictions become unreliable.

5. Results

We report only quantities derived in Section 4, plus clearly labeled projections.

1. **Sensitivity contrast implied by the reported factor of two.** Given hotspot fraction $f \in [0.05, 0.07]$ and targeted-to-random ratio $\rho = 2$ [1],[3], the required sensitivity contrast is $r = 2(1 - f)/(1 - 2f)$, giving $r \approx 2.111$ at $f = 0.05$ and $r \approx 2.163$ at $f = 0.07$ (Section 4.1). Hotspots are only about 2.1–2.2× more influential than average components.
1. **First-order targeted gain.** With the derived model parameters ($n_c = 228$, $f = 0.06$, $r = 2.136$, $m = 14$), the first-order fractional reduction is $G_{\text{tgt}} = 29.904/487.10 \approx 6.14\%$, versus $G_{\text{rnd}} = 14/456 \approx 3.07\%$ for random selection; ratio $\rho = 2.00$ by construction (Section 4.2).
1. **Nonlinear amplification factor.** The reported 15%–25% reductions [1],[3] exceed the first-order prediction by $A \in [2.44, 4.07]$, midpoint $A \approx 3.15$ (Section 4.3), interpretable as syndrome-propagation fan-out of 2–4 rounds — or, under the alternative convention of Appendix A, as evidence of a stronger ($\sim 10\times$) sensitivity skew.
1. **Estimator precision (projection under stated assumptions).** Assuming $N = 10^6$ samples, $P_L = 10^{-3}$, and per-component firing probabilities $q_i \in [10^{-5}, 4 \times 10^{-3}]$, the sensitivity estimator's relative standard error ranges from $\approx 2.4 \times 10^{-4}$ (common gate locations) to $\approx 0.5\%$ (rare locations) (Section 4.4). This is a

projection from the variance model of Section 3.5, not a simulation measurement; the model omits covariance between component estimators, which would inflate these figures by an unknown factor of order the number of correlated components sharing a syndrome path. A conservative weight-bounded estimate (Section 4.4) instead indicates $N \sim 10^7$ is needed for reliable per-component ranking at low- P_L operating points.

1. **Intervention budget.** For the $d = 3$, $R = 4$ rotated surface code census ($n_c = 228$), the reported 5%–7% band corresponds to 12–16 components (Section 4.5).
1. **Equivalent physical budget (model).** A 20% logical reduction corresponds to a uniform physical error reduction factor of 0.894 ($\approx 10.6\%$); the model's 6.14% reduction corresponds to $\sqrt{0.9386} \approx 0.9688$ ($\approx 3.1\%$) (Section 4.6).
1. **Sample-complexity wall (projection).** Maintaining 10% relative precision on P_L requires $N \gtrsim 10^4/P_L$ samples; at $P_L = 10^{-5}$ this is $N \geq 10^9$, a roughly $10^3\times$ increase over the $P_L = 10^{-3}$ operating point (Section 4.4).

6. Discussion

What the analytic model explains and what it does not. The closed-form ratio $\rho = r/(fr + 1 - f)$ shows that the reported factor-of-two advantage does not require exotic sensitivity landscapes: a 6% hotspot fraction with a $2.14\times$ contrast suffices. This is good news for practitioners — moderate sensitivity heterogeneity, which is generic in real circuits, already yields a twofold resource-allocation advantage. Conversely, it is a caution for interpretation: a factor of two is the *generic* outcome of any moderately skewed sensitivity distribution, so empirical factors near two do not by themselves indicate that attribution has found uniquely important components. Only the absolute gain (15%–25% versus the first-order 6%) reveals the nonlinear structure, and our amplification factor $A \approx 3.15$ is an inference from a toy multiplicative model, not a derivation from the actual circuit dynamics.

Limitations and failure modes. First, the first-order framework assumes independent stochastic errors. Coherent-versus-stochastic comparisons [6] showed that coherent and stochastic error models yield markedly different failure distributions with no simple mapping between them; under coherent noise, sensitivities can be signed (interference can make some components *reduce* P_L when their error amplitude increases), and halving an error strength is not equivalent to halving a probability. An attribution scheme validated only on stochastic Pauli noise may misrank components catastrophically on hardware with dominant coherent errors. Second, our component census ($n_c = 228$) and all operating points ($p = 10^{-3}$, $P_L = 10^{-3}$, $N = 10^6$) are modeling assumptions; the lifted-product code results in [1],[3] involve different censuses and possibly different hotspot fractions, and our f -to- r inversion should be redone per code family. Third, the estimator variance model ignores covariances; correlated error paths (a single physical defect firing on many nominal components) can inflate $\text{Var}(\hat{s}_i)$ beyond our projection. Fourth, sensitivity maps are snapshots: under calibration drift, the map itself drifts, and a hotspot list computed once may go stale. Fifth, after intervening on a hotspot, the attribution map changes — *sensitivity redistribution* — and a greedy iterate-intervene-recompute loop may oscillate or saturate; the linear model predicts no saturation, so observed saturation would itself be informative. Sixth, at very low P_L (deep sub-threshold), the sample-complexity wall of Section 4.4 makes direct Monte Carlo attribution expensive; importance-sampling or zero-variance-transition extensions would be needed.

Arguing against ourselves. A skeptic could object that the targeted-versus-random comparison is not the right baseline: a fair comparison is against the best simple heuristic (e.g., prioritizing measurement locations, known error hotspots) without any attribution machinery. If the sensitivity skew correlates trivially with component type, attribution reduces to a lookup table. The counterargument is that attribution is data-driven and catches non-obvious hotspots (e.g., schedule-dependent idling locations), but this remains to be demonstrated against type-based

baselines. A second objection: if sensitivities must be estimated per device, the approach may not scale to large codes where P_L is tiny and N explodes as $1/P_L$; mitigations — importance sampling, conditioning estimators, and the joint-estimation trick of [1],[3] — partially address this but remain an open engineering question. A third objection: physical error probability and logical sensitivity are distinct, and only the product $s_i p_i$ ranks intervention value — which is precisely why attribution adds information over a "fix the noisiest component" heuristic.

Falsifiability. The analytical claims here are conditional on the linear model and stated inputs; they are falsified if (a) circuit-level simulation with realistic noise shows that targeted intervention on sensitivity-ranked components yields no better than random improvement (contradicting both [1],[3] and our $\rho > 1$ analysis), (b) near-uniform sensitivity distributions are measured across circuit components, (c) sensitivity rankings drift rapidly under recalibration, or (d) coherent-error dominance invalidates stochastic sensitivities per [6].

Open questions. How do sensitivity profiles behave under code deformation and gauge choices [8]? Can attribution be performed online from syndrome statistics, connecting to continuous-time formulations [10]? Can sensitivity maps seed the reinforcement-learning agents of [4],[9] to accelerate code discovery? Do erasure-conversion overheads [7] and entanglement-assisted checks [11] interact multiplicatively with attribution-guided noise budgets? Speculatively, if the QEC–Quantum Darwinism no-go at $F_L > 0.874$ [13] is correct, attribution-guided gains push devices toward a fundamental tradeoff boundary rather than an unbounded improvement path; and passive ultrametric resilience schemes [14] would need their own attribution theory.

7. Conclusion

Error attribution converts the logical error rate from a single benchmark number into a spatial map of engineering leverage. We showed analytically that the advantage of sensitivity-guided over random noise reduction is $\rho = r/(fr + 1 - f)$, that the reported factor of two [1],[3] follows from a modest $2.1\text{--}2.2\times$ sensitivity contrast on 5%–7% of components, and that the reported 15%–25% absolute gains exceed the first-order prediction by an amplification factor $A \in [2.4, 4.1]$ consistent with syndrome-propagation fan-out. The framework's costs are a demanding (but standard) Monte Carlo budget — with a $10^4/P_L$ wall for aggregate precision — and an assumption of stochastic, stationary noise. As QEC moves from demonstrating thresholds to optimizing deployed devices, attribution provides the missing link between benchmarking and resource-constrained design.

References

-
- [1] TITLE: arXiv Query: `searchquery=&idlist=2610.03399&start=0&max_results=1` [2] Channel-Optimized Quantum Error Correction. arXiv:0810.2524v1. <https://arxiv.org/abs/0810.2524v1> [3] Optimizing quantum error correction through error attribution. arXiv:2610.03399v1. <https://arxiv.org/abs/2610.03399v1> [4] Optimizing Quantum Error Correction Codes with Reinforcement Learning. arXiv:1812.08451v5. <https://arxiv.org/abs/1812.08451v5> [5] Robustness-optimized quantum error correction. arXiv:1909.05156v2. <https://arxiv.org/abs/1909.05156v2> [6] Quantum error correction failure distributions: comparison of coherent and stochastic error models. arXiv:1704.03961v1. <https://arxiv.org/abs/1704.03961v1> [7] Optimizing quantum error correction protocols with erasure qubits. arXiv:2408.00829v3. <https://arxiv.org/abs/2408.00829v3> [8] Optimized Quantum Error Correction Codes for Experiments. arXiv:1411.1779v2. <https://arxiv.org/abs/1411.1779v2> [9] Discovering autonomous quantum error correction via deep reinforcement learning. arXiv:2511.12482v2. <https://arxiv.org/abs/2511.12482v2> [10] Continuous-time quantum error correction. arXiv:1311.2485v2. <https://arxiv.org/abs/1311.2485v2> [11] Entanglement-Assisted Quantum Error-Correcting Codes. arXiv:1610.04013v1. <https://arxiv.org/abs/1610.04013v1> [12] DOI 10.5281/zenodo.21979060. QNFO: Implications

for Computing and Quantum Error Correction. [13] DOI 10.5281/zenodo.21964674. QNFO: Archimedean Shadows: The QEC-Darwinism Tradeoff in Ultrametric Spaces. [14] DOI 10.5281/zenodo.21748299. QNFO: Passive Error Resilience Through Ultrametric Geometry: A Proposal for p-Adic Quantum Metrology.