

Node Targeted Memory Strengths for RelayBP Decoding of Bivariate Bicycle Codes

Hongjun Kim
School of Electrical Engineering
Korea University
Seoul, Korea
hongjunkim98@korea.ac.kr

Jun Heo
School of Electrical Engineering
Korea University
Seoul, Korea
junheo@korea.ac.kr

Abstract—Bivariate bicycle codes suffer from low weight error floor caused by short cycles and degeneracy during the belief propagation decoding process. RelayBP mitigates this weakness by assigning evenly randomized memory strength to variable nodes. We show that failures that RelayBP still causes concentrate on a small subset of fault mechanisms. These nodes behave differently across the X and Z error sectors, and a structural analysis localizes the difference in prior log likelihood ratios, an asymmetry traced to the syndrome extraction circuit. Building on this, we propose localized memory broadening, which widens the memory strength sampling interval only for target nodes. At the detector level this reduces Z logical errors and X logical errors along with enhanced converged fraction.

Index Terms—qLDPC, BB code, RelayBP

I. INTRODUCTION

Quantum computers outperform classical computers by offering faster solutions in certain problems. Yet, the aspect of qubit fragility requires refinement for successful quantum operations, leading to the importance of fault-tolerant quantum computation (FTQC) [1]–[3]. Especially, quantum error correcting (QEC) codes enhance the robustness of information, preserving them against susceptible noise. Quantum low-density parity-check (qLDPC) codes, such as Bivariate Bicycle (BB) codes, arise as promising candidates due to their high encoding rate along with their implementable structure [4]. For these codes to be applicable in achieving FTQC, real-time decoding and low logical error rates remain necessary.

While qLDPC codes decode via BP, its limited performance due to short cycles and quantum degeneracy was enhanced through Ordered Statistics Decoding (OSD) post-process [5]. However, OSD requires $\mathcal{O}(n^3)$ complexity of operations, highlighting its incapability for real-time decoding. Recently, relay-ensembling with locally-averaged memory belief propagation (RelayBP) [6] emerged as a leading real-time decoder candidate, showing shortened operation time and enhanced performance compared to BP-OSD. RelayBP achieves low logical error rate through the mechanism of applying disordered memory weight γ during the message passing process.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00242396). This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00463, Development of a quantum repeater in optical fiber networks for quantum internet).

Meanwhile, RelayBP mechanism weighs parameter γ independently and randomly for each variable node (VN), leaving a question for refinement of memory strength γ choice. Given that BP failures concentrate among certain VN subsets, it is natural for weight γ to be set differently to those target VNs.

This paper identifies VNs responsible for these failures, traces their X/Z asymmetry to the prior log-likelihood ratios induced by the syndrome measurement circuit, and proposes localized memory broadening – a cost-free, per-node widening of the memory-strength interval that reduces hard-error logical failures by 74% (Z) and 46% (X) at the detector level.

II. BACKGROUND

A. Bivariate Bicycle (BB) Codes

BB codes are CSS-type stabilizer codes built from a pair of binary circulant matrices A and B generated by bivariate monomials[4], with check matrices

$$H_X = [A \mid B], \quad H_Z = [B^T \mid A^T], \quad (1)$$

which satisfy the CSS condition $H_X H_Z^T = 0$. We adopt the gross code $[[144, 12, 12]]$, for which $A = x^3 + y + y^2$ and $B = y^3 + x + x^2$, under circuit-level depolarizing noise model.

B. BP, DMem-BP and RelayBP

In QEC, maximum-likelihood (ML) decoding selects the most probable error sub-coset consistent with the syndrome s , but its cost grows exponentially with the number of logical qubits. Belief propagation (BP) replaces this with a tractable approximation that maximizes each fault's marginal $P(E_j \mid s)$ independently through iterative check-to-variable and variable-to-check message passing on the Tanner graph [7]. On BB codes, the dense short cycles and degeneracy of this graph limit the performance of BP.

Disordered Memory BP (DMem-BP) mitigates this by replacing the fixed prior $\Lambda_{v_j}(0)$ with a per-iteration intrinsic value $\Lambda_{v_j}(t)$. After each iteration the marginal is updated as

$$M_{v_j}^{(t)} = \Lambda_{v_j}(t) + \sum_{c_{i'} \in M(v_j)} m_{c_{i'} \rightarrow v_j}^{(t)}, \quad (2)$$

and the intrinsic value $\Lambda_{v_j}(t)$ is updated as

$$\Lambda_{v_j}(t) = (1 - \gamma_j) \Lambda_{v_j}(0) + \gamma_j M_j(t - 1), \quad (3)$$

where the previous marginal $M_j(t-1)$ and the initial prior $\Lambda_{v_j}(0)$ are mixed by the memory strength γ_j [9].

RelayBP is formed by linking multiple DMem-BP instances in series. The first DMem-BP instance of the relay, initiates its marginal as $\Lambda_{v_j}(0) = \log \frac{1-p_{v_j}}{p_{v_j}}$.

The r^{th} instance's marginal of each VN is initialized as final marginal of $(r-1)^{th}$ instance, while memory strength is set uniformly random among the range $[\gamma_{low}, \gamma_{high}]$. While RelayBP operates R instances, of which each instance consists of max iteration T_r , the algorithm stops when R instances have been executed or S solutions have been discovered. The final solution of this algorithm returns to the lowest-weight solution found among all legs.

C. BP Failure Modes

In degenerate quantum codes, BP inevitably suffers from failures as final solution $\hat{E}$ that is based on qubit-wise marginal updates may not match into appropriate sub-coset $\hat{L}$. This mismatch manifests as three distinct failure modes [7-8].

1) *Oscillation (non-convergence)*: Message fails to settle to a fixed point due to short cycles in a Tanner graph – e.g., the 4-cycles induced by the bivariate structure of BB codes – whose associated trapping sets sustain self-reinforcing message oscillations. BP fails to produce a stable hard decision; its estimate continuously flips up to its iteration limit.

2) *Wrong syndrome convergence*: Message settles to a stable fixed point, but hard decision $\hat{E}$ violates the syndrome constraint, $H\hat{E} \neq s$. Bitwise marginal update process possibly leads to such a result, as VN's bitwise convergence may not result in the combination to satisfy the syndrome constraint.

3) *Wrong sub-coset convergence*: Messages settle to stable fixed point and yield hard decision that satisfies the syndrome condition $H\hat{E} = s$, but $\hat{E}$ lies in the wrong logical class.

III. TRAP CHARACTERIZATION IN RELAYBP

BP failure is unevenly distributed over VNs. Instead, they are concentrated among specific subsets. We identify the low-weight error configurations responsible for dominant failures, then extract VNs that compose them. These nodes form the target set a tailored memory strength is later applied.

We consider the gross code $[[144, 12, 12]]$ under the standard circuit-level depolarizing noise model, decoding the logical-idle syndrome cycle over four noisy syndrome cycles. In the resulting Detector Error Model (DEM), the decoding matrices H_X and H_Z are defined over fault mechanisms rather than data qubits. Each VN corresponds to a single circuit fault, and each Check Node (CN) corresponds to a detector formed from the parity change of a stabilizer between consecutive cycles. Faults that produce an identical syndrome are merged into a single VN. For our decoder, this yields the number of VNs (V_Z, V_X) as $V_Z = 2784$ and $V_X = 2747$.

Following [10], we target the weight-four error configurations that BP decodes slowly or incorrectly. For sets of fault VNs we denote by w the Hamming weight of their modulo-2 sum, n_u the number of distinct VNs in the union of their support, and $n_c = n_u - w$ the number of canceled checks.

The construction selects pairs of CNs (c_i, c_j) sharing the maximal number of VN faults $n_s(c_i, c_j) = 8$; there are 72 such pairs per sector in one syndrome cycle. A weight-four error is built from two shared VNs of a first pair $p_0 = (c_1, c_2)$ and two of a second pair $p_1 = (c_3, c_4)$, subject to

$$\begin{aligned} n_c(s_1, s_2) &= 2 & n_c(s_3, s_4) &= 2 \\ n_c(s_1, s_2, s_3, s_4) &= 8 \end{aligned}$$

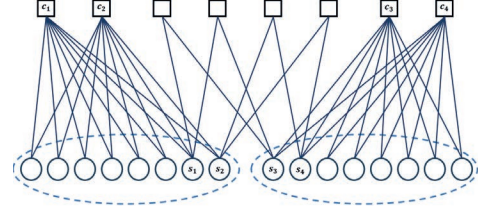


Fig. 1. A possible structure of part of the Tanner subgraph of weight-four errors that are slow to converge with BP iterations. Checks are drawn as squares and VNs as circles.

The first two conditions require that within each pair of VNs only its two shared CNs cancel, while the third requires that eight CNs cancel once the two pairs of VNs are combined, through cross-pair CN coincidences.

It is precisely this combination of intra-pair and cross-pair cancellation – dense short-cycle structure with extensive degeneracy – that drives the BP failure modes discussed in Section II. Fig. 1 schematically shows VNs of weight-four of which BP slowly converges.

Applying this construction to our decoder yields 2664 distinct weight-four errors in each of H_Z and H_X , in exact agreement with the count reported in [10]; this consistency validates both our DEM generation and our implementation of the selection criteria.

Taking the union of the support of all 2664 errors gives the set of VNs that participate in at least one such error. This set contains 360 nodes, about 13% of all fault mechanisms. We emphasize that the difficulty is a property of the four-node “combination” and its local tanner-graph neighborhood, not of any single node. These 360 nodes per sector, summarized in Table I, constitute the target set $\mathcal{H}_{X/Z}$ to which we apply a tailored memory strength in Section V.

The identical error and node count already suggests that the two sectors are symmetric at this level. As we show in Section IV, the relevant difference between them appears in the prior log-likelihood ratios $\Lambda_{v_j}(0)$ attached to the selected nodes.

TABLE I
HARD-ERROR STATISTICS FOR THE GROSS CODE

Quantity	Z	X
Fault columns (variable nodes)	2784	2747
Maximum-sharing pairs ($n_s = 8$)	72	72
Weight-four hard errors	2664	2664
Target nodes ($ \mathcal{H} $)	360	360

IV. STRUCTURAL ANALYSIS OF HARD-ERROR NODES

We now characterize these 360 nodes – asking how they differ from the rest of the code, and from each other across the two sectors. For each node we measure four quantities: its degree, its number of incident 4-cycles, its participation count among the 2664 hard errors, and its prior log-likelihood ratio $\Lambda_{v_j}(0)$.

TABLE II
MEAN OF THE STRUCTURAL QUANTITIES FOR TARGET NODES ($\mathcal{H}$) VERSUS THE FULL POPULATION, PER SECTOR.

Quantity	Z: $\mathcal{H}$	Z: all	X: $\mathcal{H}$	X: all
Degree	4.2	3.3	4.2	3.1
4-cycle count	19.0	11.7	19.8	9.9
Participation	29.6	3.8	29.6	3.9
$\Lambda_{v_j}(0)$	7.26	6.66	6.83	6.66

Each of the three topological quantities (degree, 4-cycle count, participation) of Table II reflects a distinct route to BP failure. The higher degree (4.2 versus 3.1-3.3) means target nodes couple to more detectors, spreading their influence across a larger neighborhood; while this carries more syndrome information in principle, for BP it co-occurs with denser short cycles, and the resulting message correlations violate BP's independence assumption. The higher count of 4-cycles (19 versus 11) pinpoints the concentrated trapping sets that self-reinforce oscillations of Section II. Finally, the participation count shows the sharpest gap (29.6 versus 3.8): target nodes are the shared hubs of the degenerate combinations, so a single node anchors many distinct hard errors – which is why intervening on this small set can affect a large share of the hard-error family. Together, these confirm that target nodes are not random but occupy the dense, degenerate regions of the Tanner graph.

Meanwhile, Fig. 2 shows the prior-LLR distributions of the X and Z target nodes. In both sectors the bulk of the nodes lie at high LLR, reflecting that the hard error nodes are predominantly low-probability fault mechanisms. The two sectors differ in their means: the Z target nodes sit at $\Lambda_{v_j}(0) = 7.26$, higher than the 6.83 of the X nodes, and the X distribution carries comparatively more weight at lower LLR, which lowers its mean.

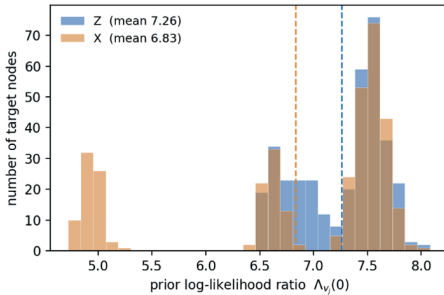


Fig. 2. Prior-LLR Distribution of X, Z Target Nodes. Mean value of the prior-LLR is expressed with dashed lines.

The separation of the two means is therefore not driven by a few outliers but reflects an overall shift between the sectors, substantiating the 7.26 versus 6.83 difference of Table II as robust property of the target sets. Equivalently, the Z target nodes correspond to rarer fault mechanisms – carrying a stronger prior belief that they are inactive – than their X counterparts.

This asymmetry is confined to the target sets. Over the full population the mean prior LLR is identical in the two sectors (6.66), as expected from a depolarizing noise model that treats X- and Z-type error symmetrically. The asymmetry therefore does not originate in the noise model: it is the selection of hard-error nodes – governed by the circuit-level structure of H_X and H_Z – that picks out lower-probability faults in Z.

The source of this selection asymmetry is the syndrome-measurement circuit. Although the depolarizing channel is X/Z-symmetric and the cross-code definition through $H_X = [A|B]$ and $H_Z = [B^T|A^T]$ is symmetric in form, the CNOT gate schedule of the extraction circuit is not: a CNOT gate propagates X and Z type errors in opposite directions, so a fixed gate ordering acts differently on the two error types. The resulting spacetime fault structures of H_X and H_Z differ, and with them the subset of fault mechanisms that satisfy the maximal-sharing and cancellation conditions of Section III. As a result, the two target sets are populated by different fault mechanisms, and – as measured in Table II – the Z target set happens to consist of rarer faults, giving it a higher mean prior LLR. We stress that this reflects which faults the circuit-induced structure selects, not any asymmetry in the noise model itself.

Crucially, the prior LLR is the quantity weighed against the propagated marginal by γ_j in the DMem-BP update. This identifies it as the natural variable for a per-node memory assignment, and shows that a sector-uniform γ_j ignores the asymmetry found here – motivating the scheme of Section V.

V. METHOD: LOCALIZED MEMORY BROADENING

Section IV identifies the prior LLR $\Lambda_{v_j}(0)$ as the variable that distinguishes the target nodes across sectors. The most direct way to exploit this is to make each target node's memory strength a function of its own prior. We therefore first consider a rule that assigns γ_j by linearly interpolating between two endpoints according to the node's normalized prior:

$$\gamma_j = \gamma_{low} + (\gamma_{high} - \gamma_{low})t_j, \quad \text{where} \quad t_j = \frac{\Lambda_{v_j}(0) - \Lambda_{min}}{\Lambda_{max} - \Lambda_{min}} \quad (4)$$

where Λ_{min} and Λ_{max} are the smallest and largest prior LLRs among the target nodes, so that t_j places each node on a $[0, 1]$ scale by its prior. A node with the lowest prior receives γ_{low} , one with the highest receives γ_{high} , and the rest interpolate. This shifts the center of each target node's memory strength according to how rare its fault is. We evaluated both assigning stronger memory to high-prior nodes and the reverse.

However, this would confine the target nodes to a narrow per-leg range, suppressing the stochastic exploration that RelayBP relies on – precisely at the most degenerate nodes. We

therefore expect such center-shifting to be counterproductive and instead propose broadening. Section VI confirms this.

Concretely, broadening leaves the center of the sampling distribution at the baseline but enlarges its support for the target nodes. At each leg, the memory strength of node v_j is

$$\gamma_j \sim \begin{cases} \mathcal{U}[\gamma_{low}^H, \gamma_{high}^H], & v_j \in \mathcal{H} \\ \mathcal{U}[\gamma_{low}, \gamma_{high}], & \text{otherwise,} \end{cases} \quad (5)$$

where $\mathcal{U}[a, b]$ is the uniform distribution on $[a, b]$ and the target interval strictly contains the baseline one, $[\gamma_{low}^H, \gamma_{high}^H] \supset [\gamma_{low}, \gamma_{high}]$. In contrast to the center-shifting rule, this assigns no fixed value to any node: each target node still draws an independent, uniformly random memory strength at every leg, preserving the leg-to-leg stochasticity of RelayBP. Only the support of that distribution changes – widened for the target nodes so that they additionally sample more negative memory strengths, which perturb and can break the self-reinforcing fixed points underlying the oscillation and wrong sub-coset failures of Section II. We use the baseline interval $[\gamma_{low}, \gamma_{high}] = [-0.24, 0.66]$ and the broadened target interval $[\gamma_{low}^H, \gamma_{high}^H] = [-0.5, 0.9]$.

The scheme alters neither the decoding matrices, the code, nor the RelayBP algorithm; only the per-node sampling support is changed, for the $|\mathcal{H}| = 360$ target nodes. The modification is therefore essentially cost-free at runtime – a constant per-node interval array – keeping it compatible with the real-time decoding setting that motivates RelayBP.

Substituting (3) into (2) and holding the check messages fixed over one iteration gives

$$M_{v_j}(t) = \gamma_j M_{v_j}(t-1) + (1 - \gamma_j) \Lambda_{v_j}(0) + S_j(t), \quad (6)$$

where $S_j(t) = \sum_{c_{i'} \in M(v_j)} m_{c_{i'} \rightarrow v_j}^{(t)}$. This is a first-order recursion with fixed point $M_{v_j}^* = \Lambda_{v_j}(0) + S_j/(1 - \gamma_j)$; writing $e(t) = M_{v_j}(t) - M_{v_j}^*$, the deviation obeys $e(t) = \gamma_j^t e(0)$. Two consequences follow. The recursion settles only for $|\gamma_j| < 1$, beyond which the prior no longer anchors the marginal; and for $\gamma_j < 0$ the deviation alternates in sign at every iteration, which is the mechanism by which negative memory strengths perturb the self-reinforcing fixed points of Section II-C.

The condition $|\gamma_j| < 1$ constrains the broadened interval. Writing the baseline interval by its center $c = 0.21$ and width $w = 0.90$, we specify the target interval as $[c + \delta - \rho w/2, c + \delta + \rho w/2]$ with dilation $\rho \geq 1$ and center offset δ , so that (5) is the center-preserving case $\delta = 0$. The upper endpoint is then binding, giving $\rho < \rho_{\max} = 2(1 - c)/w = 1.76$; the interval $[-0.5, 0.9]$ is $(\rho, \delta) = (1.56, -0.01)$, a center-preserving dilation at 89% of $\rho_{\max}$. Broadening therefore introduces a single degree of freedom whose admissible range is fixed analytically by (6), the baseline endpoints being taken unchanged from [6]. We do not claim $\rho = 1.56$ is optimal within $(1, \rho_{\max})$; predicting the optimum from the local Tanner-subgraph structure is left to Section VII.

VI. RESULTS

We evaluate localized memory broadening in three complementary settings. First, a detector-level evaluation injects

the 2664 hard errors of Section III directly into the decoder to measure its effect on the targeted failures. Second, a floor-contribution estimates these per-error outcomes by their occurrence probabilities to gauge the impact on the logical error floor. Third, a Monte Carlo evaluation samples random circuit-level errors to confirm that generic-error decoding is unaffected. All runs use the gross code over four noisy syndrome cycles and RelayBP with $R = 200$ legs, $T_r = 60$ iterations per leg, 80 pre-iterations, and $\gamma_0 = 0.125$.

A. Detector-Level Evaluation

Initially, we inject the 2664 hard errors of each sector directly into the decoder and compare baseline RelayBP with localized broadening over the target interval $[\gamma_{low}^H, \gamma_{high}^H] = [-0.5, 0.9]$. Results are averaged over three random seeds and summarized in Table III. Broadening reduces the logical error count substantially in both sectors – from 154.7 to 40.7 in Z (74% reduction) and from 536.0 to 288.7 in X (46% reduction) – while raising the converged fraction from 0.863 to 0.937.

TABLE III
DETECTOR-LEVEL PERFORMANCE ON THE 2664 HARD ERRORS PER SECTOR (MEAN $\pm$ STD, THREE SEEDS)

Metric	Baseline	Broadening
Logical fails, Z (/2664)	154.7 $\pm$ 3.3	40.7 $\pm$ 2.4
Logical fails, X (/2664)	536.0 $\pm$ 6.2	288.7 $\pm$ 7.9
Converged fraction	0.863 $\pm$ 0.001	0.937 $\pm$ 0.002

Improvement shown in both metrics is significant as it demonstrates that broadened nodes not only converge more often but also converge to the correct sub-coset. The larger reduction in Z is consistent with the prior asymmetry of Section IV – the Z target nodes carry stronger priors and are thus more susceptible to the prior-driven failures that broadening disrupts, leaving more room for improvement.

To confirm that this gain stems from broadening the sample support – rather than from any node-targeted change – we compare a range of alternative interventions on the same target set. Table IV reports the change in failures relative to baseline.

TABLE IV
ABLATION: CHANGE IN LOGICAL FAILS VS. BASELINE (Δ , NEGATIVE=IMPROVEMENT), MEAN OVER THREE SEEDS.

Intervention on $\mathcal{H}$	ΔZ	ΔX
Broaden $[-0.5, 0.9]$	-114.0	-247.3
Broaden $[-0.7, 0.9]$	-72.7	-106.0
Broaden $[-0.9, 0.9]$	+1171.3	+1686.3
Center shift, prior $\uparrow \rightarrow \gamma \uparrow$	+397.0	+114.0
Center shift, prior $\uparrow \rightarrow \gamma \downarrow$	+369.0	+81.0

In the (ρ, δ) coordinates of Section V, the three broadening rows are not a pure width sweep: holding $\gamma_{high}^H = 0.9$ while dilating beyond $\rho_{\max} = 1.76$ necessarily displaces the center downward, so $[-0.7, 0.9]$ and $[-0.9, 0.9]$ combine over-dilation with $\delta = -0.11$ and $\delta = -0.21$, respectively.

The diminished gain of $[-0.7, 0.9]$ and the sharp degradation of $[-0.9, 0.9]$ are consistent with the center-shift rows, in which the prior-based rule of Section V moves the center without widening and worsens decoding in both directions. The tested interval that is both center-preserving ($\delta \approx 0$) and admissible ($\rho < \rho_{max}$) is $[-0.5, 0.9]$, and it performs best in both sectors: the gain comes from dilating the sampling support about the baseline center, and displacing that center — whether by a deterministic prior-based assignment or as a by-product of over-dilation—diminishes the gain and, when large, reverses it outright. These isolate localized memory broadening as the operative mechanism.

B. Floor-Contribution Estimate

The detector-level results count each hard error equally, whereas its contribution to the logical error floor is weighted by its frequency. To capture this we estimate the weight four floor contribution by weighting each hard error e by its leading-order occurrence probability and summing over the errors the decoder fails:

$$P_{floor}^{(4)} = \sum_{e=1}^{2664} \left(\prod_{i \in e} p_{v_i} \right) \cdot P_{fail}(e), \quad (7)$$

where p_{v_i} is the prior of fault v_i and $P_{fail}(e)$ is the measured failure probability of error e measured at the detector level. The first factor is computed from the decoder’s fault priors; the second is taken from the runs of Section VI-A. Fig. 3 compares this estimate for baseline and broadening.

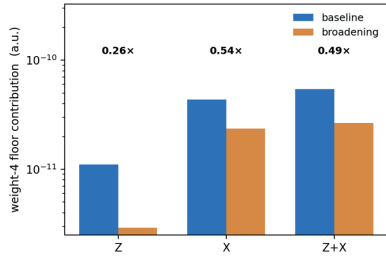


Fig. 3. Estimated weight-four floor contribution for baseline and broadening: $P_{floor}^{(4)} = \sum_e \left(\prod_{i \in e} p_{v_i} \right) \cdot P_{fail}(e)$.

Broadening lowers the weight-four floor contribution by a factor of 0.26 in Z and 0.54 in X. These reductions closely track the detector-level reductions of Section VI-A, indicating that broadening improves not merely rate configurations but the errors that dominate the floor. This estimate is a leading order — it includes only the identified weight-four hard errors and assumes independent faults — so its absolute scale may not be meaningful. However, the reduction factor is robust: the occurrence weights enter both baseline and broadening identically, so the ratio is independent of the physical error rate, and it reproduces to within 2% across seeds.

C. Monte Carlo Evaluation

A full circuit-level Monte Carlo over four physical error rates (3×10^{-4} to 1.2×10^{-3} , 1.2×10^5 shots/point) confirms

that broadening leaves generic-error decoding unaffected, with per-cycle logical error rates matching baseline to within 0.2%. These curves do not reflect the detector-level and floor improvements, as expected: the accessible error rates lie above the floor regime, where generic errors dominate and the 2664 targeted configurations are a negligible fraction. The target errors dominate only deep in the floor regime at lower p , which is infeasible to sample directly on the gross code and is therefore assessed through the estimate of Section VI-B.

VII. DISCUSSION AND CONCLUSION

We identified the VNs responsible for low-weight BP failures in the gross code and showed that they occupy the densest, most degenerate regions of the Tanner graph. A structural analysis revealed that the X/Z asymmetry of these nodes lies not in their topology but in their prior log-likelihood ratios.

Building on this, we proposed localized memory broadening. The key insight is that these nodes need more exploration rather than a prescribed memory center — shifting the center degrades decoding, whereas localized broadening improves it. At the detector level, broadening substantially reduces logical errors and raises the converged fraction in both sectors, with a correspondingly large reduction of the estimated weight-four floor contribution. The scheme modifies only a per-node sampling interval, which is essentially cost-free at runtime and well suited to real-time decoding. The improvement is established on the specific family of weight-four hard errors that prior work identifies as a primary source of the BP floor; we do not independently establish their dominance. The gross-code floor is too low for Monte Carlo verification, so its reduction is assessed through a probability-weighted estimate whose absolute scale is model-dependent.

These limitations point to extensions: sector-specific tuning informed by prior asymmetry, a structure-driven rule predicting the optimal dilation ρ within $(1, \rho_{max})$ — removing the hyperparameter, and evaluation on codes or syndrome cycles.

REFERENCES

- [1] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Physical review A*, vol. 52, no. 4, pp. R2493, 1995.
- [2] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Physical Review A*, vol. 54, no. 2, pp. 1098, 1996.
- [3] A. G. Fowler *et al.*, “Surface codes: Towards practical large-scale quantum computation,” *Physical Review A*, vol. 86, no. 3, pp. 032324, 2012.
- [4] S. Bravyi *et al.*, “High-threshold and low-overhead fault-tolerant quantum memory,” *Nature*, vol. 627, no. 8005, pp. 778–782, 2024.
- [5] J. Roffe *et al.*, “Decoding across the quantum low-density parity-check code landscape,” *Physical Review Research*, vol. 2, no. 4, pp. 043423, 2020.
- [6] T. Müller *et al.*, “Improved belief propagation is sufficient for real-time decoding of quantum memory,” *arXiv preprint arXiv:2506.01779*, 2025.
- [7] D. Poulin and Y. Chung, “On the iterative decoding of sparse quantum codes,” *arXiv preprint arXiv:0801.1241*, 2008.
- [8] N. Raveendran and B. Vasić, “Trapping sets of quantum LDPC codes,” *Quantum*, vol. 5, pp. 562, 2021.
- [9] J. Chen *et al.*, “Improved belief propagation decoding algorithms for surface codes,” *IEEE Transactions on Quantum Engineering*, 2025.
- [10] H. Landa, “Low-weight quantum syndrome errors in belief propagation decoding,” *arXiv preprint arXiv:2603.19126*, 2026.