

From Random Quantum Codes to Explicit qLD Codes: A Reconciled Threshold Analysis of the Quantum-LCL Framework

From Random Quantum Codes to Explicit qLD Codes: A Reconciled Threshold Analysis of the Quantum-LCL Framework

Abstract

A recent preprint (arXiv:2609.40252) extends the local coordinate-wise linear (LCL) witness framework of Levi, Mosheiff, and Shagrirhaya from classical linear codes to CSS quantum codes. The central structural difficulty is that a CSS code is a nested pair of spaces $S \subseteq C$, so a local witness has two ranks: a physical rank on representatives in C and a logical rank in the quotient C/S . The resulting theory yields a threshold theorem for random CSS codes whose striking corollary is that the per-sector rate threshold equals the classical rate threshold, together with quantum subspace designs and explicit folded constructions that are themselves qLDPC — the first explicit quantum list-decodable and list-recoverable codes with optimal list sizes. This paper is a reconciled analytical companion to that work. We compute the binary entropy $H_2(0.1) = 0.468996$ and the associated classical threshold $R = 1 - H_2(0.1) = 0.5310$, exhibit a concrete random CSS allocation with per-sector rates $R_X = R_Z = 0.3$ and total rate 0.4 that lies strictly below threshold in both sectors, and quantify the downstream resource consequences: a rate- $1/2$ good qLDPC code uses 2 physical qubits per logical qubit — a rate statement independent of distance; at matched distance $d = 100$ this requires block length $n \approx 910$ (relative distance $\delta \approx 0.11$, the largest value consistent with the quantum Hamming bound at $R = 1/2$), versus 19,999 for a surface code — a factor of $\approx 10^4$ on locality-permissive architectures, conditional on the distance being achieved at that block length. We also present an alternative threshold convention ($R = 1 - H(p) - 1/L$) used in one source draft for fixed list size L , and document the resolution in Appendix A. We identify finite-length behavior, decoder realizability, and check geometry as the principal gaps between the asymptotic theory and deployable systems.

1. Introduction

The gold standard in coding theory is a code that is explicit, efficient to encode and decode, and matches the parameters achieved by a random code. For classical binary linear codes, random constructions are optimal or near-optimal for distance, list decoding, and list recovery, yet converting "a random code has these parameters with high probability" into "here is a specific polynomial-time constructible code with these parameters" has historically required substantial additional machinery. The quantum version of this problem is harder for two reasons. First, a quantum code must protect against both bit-flip (X) and phase-flip (Z) errors, so CSS-type constructions require two coordinated classical codes. Second, practical quantum error correction demands that the code be low-density parity-check (LDPC): each parity check involves only a constant number of qubits and each qubit only a constant number of checks, since dense checks translate directly into infeasible syndrome-extraction circuits.

The quantum-LCL framework of [1], [2] addresses this gap. Classical LCL witnesses express many coding-theoretic properties — minimum distance, list decodability, list recoverability — as families of local linear constraints whose ranks control when random linear codes satisfy them. The quantum adaptation handles the nested structure $S \subseteq C$ of CSS codes: local constraints are imposed on physical representatives, while independence is measured in the logical quotient C/S . This two-rank structure is the key innovation, and it yields the threshold theorem: random CSS codes satisfy a given quantum-LCL property with high probability precisely when the per-sector rate lies below a threshold, and that threshold equals the corresponding classical threshold.

This paper makes three contributions. (i) We restate the threshold mechanism in self-contained terms and verify its quantitative content with explicit arithmetic. (ii) We trace the systems-level consequences for fault-tolerance pipelines described in [4], [5], [7], [8], [9]. (iii) We argue critically: we identify where the framework's guarantees are weakest and state what evidence would falsify the claim that quantum-LCL derandomization is practically transformative. Where our source drafts disagreed on conventions — notably on how the threshold should be parameterized and how CSS rates should be accounted — we adopt one convention in the main text and document every divergence explicitly in Appendix A.

2. Background and Related Work

The source work [1], [2] (arXiv:2609.40252v1). This preprint develops the quantum-LCL framework for nested spaces $S \subseteq C$: local constraints are imposed on physical representatives while independence is measured in the logical quotient C/S . Its headline results are a threshold theorem for random CSS codes, the equality of per-sector and classical rate thresholds, a quantum analogue of subspace designs, and explicit folded quantum-LCL constructions that are qLDPC, yielding the first explicit quantum list-decodable and list-recoverable codes with optimal list sizes. Our paper is an independent analysis of this work's quantitative content and systems relevance.

Concatenation and the bosonic interface [3]. The question of what physical encoding the qubits of an outer qLDPC code should use is addressed by the QLDPC-GKP scheme of [3], which concatenates discrete-variable outer codes with the continuous-variable Gottesman–Kitaev–Preskill (GKP) code and shows such schemes can surpass the CSS Hamming bound at finite rate. This matters for the LCL constructions because the explicit codes of [2] are abstract sparse stabilizer codes; [3] indicates that per-qubit bosonic encodings can further improve the effective noise channel they face, and high-performance outer codes with provably optimal list-decoding behavior matter when the effective channel into the outer code is adversarial or correlated rather than memoryless.

Entanglement purification [4]. This work builds entanglement purification protocols on qLDPC codes with iterative decoding, motivated by the fact that long-range-interaction qLDPC codes are hard to reach from nearest-neighbor topological hardware. Purification benefits from codes whose distance and list-decoding parameters are provably near the random-code optimum, since purification fidelity bounds degrade with the code's ability to disambiguate multiple error candidates; the explicit good qLDPC codes from [2] strengthen the code-selection layer of such protocols.

Classical resource costs and predecoding [5]. This work addresses resource contention in the quantum-classical interface of large fault-tolerant machines and proposes generalized qLDPC predecoding to reduce real-time decoder load. The connection to [2] is architectural: explicit list-decodable qLDPC codes change the shape of the predecoding problem, because list decoding returns a small set of candidate logical operators rather than a single guess, and resource allocation across logical qubits must then budget for list disambiguation.

Spatially-coupled constructions [6]. This work generalizes classical spatial coupling to quantum LDPC codes, showing toric codes as 2D-SC counterparts and constructing SC-QLDPC codes with good decoding behavior and low-latency decoder compatibility. Spatial coupling is an alternative route to random-code-like performance: coupled ensembles empirically approach capacity under belief propagation, whereas the quantum-LCL framework gives worst-case (adversarial) guarantees at random-code parameters. Whether quantum-LCL codes can be spatially coupled without losing check locality is an open question (Section 6).

Fault-tolerant computation with good qLDPC codes [7]. This work gives a fault-tolerant computation scheme with constant qubit overhead and time overhead $O(d^{1+o(1)})$ for any $[[n, k, d]]$ qLDPC code with constant rate and $d = \Omega(n^{1/a})$; for good codes the time overhead reaches $O(d^{1+o(1)})$. This is a direct consumer of [2]'s

constructions: the scheme's optimality hinges on the existence of good qLDPC codes, and the quantum-LCL derandomization supplies explicit instances whose distance scaling can be certified through the LCL distance witness.

Photonic implementation [8]. This work proposes a fusion-based photonic architecture with quantum emitters tailored to qLDPC codes, exploiting the platform's natural support for non-local check connections. Photonic platforms are a plausible near-term beneficiary of [2]: since photon loss is a loss-of-qubit event, codes with high rate and good list-recoverability under erasures are exactly what the outer code in a photonic fault-tolerance stack needs.

2D-local layout [9]. This work confronts geometric locality head-on: for 2D-local gate architectures, naive implementation of high-rate qLDPC codes incurs prohibitive overhead, and the paper develops error-corrected layouts to mitigate this. This is the main practical friction point for [2]'s constructions, which control check density but not check geometry.

Corpus context [10], [11], [13]. Within the QNFO corpus, [10] argues on resource-commensurable grounds (photons per logical qubit at logical error rate 10^{-6}) that bosonic codes need 5–40× fewer photons than surface codes, suggesting the "native encoding" question interacts with the choice of outer qLDPC code. The Mahler-spectral classification of [11] finds that a proposed p-adic invariant *vp^{max} separates only Golay-type self-dual codes* ($vp^{\max} = 28$), with other stabilizer families clustering at a random baseline (1–6) — a caution against expecting hidden algebraic structure to explain qLDPC en masse, and a datum suggesting small explicit codes are far from random-code behavior. The qudit/tree-topology extension of [13] formalizes geometric error confinement on ultrametric processors; the LCL framework is, by contrast, purely linear-algebraic, and the relation between the two viewpoints remains unexplored. We note that [12] has no retrievable abstract in the provided material and is not substantively discussed.

Collectively, these works establish that explicit qLDPC codes meeting random-code benchmarks are both theoretically compelling and practically necessary, and that the binding constraints going forward are decoding and layout rather than existence.

3. Methods

Our method is analytical arithmetic on stated asymptotic results. We take as inputs:

1. **Threshold equality (from [2]).** For a folded quantum-LCL property whose classical analogue has random-code rate threshold R , *a random CSS code satisfies the property with high probability in a sector iff that sector's rate is below R* . The per-sector threshold equals the classical threshold.
1. **Classical list-decoding capacity (standard).** Binary random linear codes are list-decodable up to radius ρ with polynomial — and, for rates bounded away from capacity, constant — list sizes whenever $R < 1 - H_2(\rho)$, where H_2 is the binary entropy function. A fixed-list-size variant used in one source draft reads $R^* = 1 - H_2(\rho) - 1/L$ for list size L ; we adopt the capacity form in the main text and document the divergence in Appendix A.
1. **CSS rate accounting.** A CSS code on n qubits with X-check matrix H_X and Z-check matrix H_Z (each of full row rank, with orthogonal row spaces) has $k = n - \text{rank}(H_X) - \text{rank}(H_Z)$ logical qubits. Writing $R_X = \text{rank}(H_X)/n$ and $R_Z = \text{rank}(H_Z)/n$, the code rate is $R = k/n = 1 - R_X - R_Z$. (One source draft instead used the single-pool accounting $k = n - m$ with $m = 400$; see Appendix A.)
1. **Overhead model for good qLDPC codes (from [7]).** A good qLDPC code has constant rate and $d = \Theta(n)$; the scheme of [7] achieves constant qubit overhead with time overhead $O(d^{1+o(1)})$.

1. **Surface-code baseline (standard).** A distance- d rotated surface code uses $2d^2 - 1$ physical qubits for one logical qubit.

We compute $H_2(\rho)$ by explicit evaluation of $-\rho \log_2 \rho - (1-\rho) \log_2(1-\rho)$, derive threshold values, construct a concrete rate allocation for a random CSS ensemble, compute list sizes from the standard bound $L = O(1/\epsilon)$ with $\epsilon = 1 - R - H_2(\rho)$, and compare qubit overheads between good qLDPC codes and surface codes at matched distance. All numbers in Section 5 are either computed in Section 4 or explicitly labeled projections with stated assumptions. No simulations are run.

4. Analysis

4.1 The classical threshold at radius 0.1

Input: $\rho = 0.1$. The binary entropy is

$$H_2(0.1) = -0.1 \cdot \log_2(0.1) - 0.9 \cdot \log_2(0.9).$$

Compute each term:

- $\log_2(0.1) = \ln(0.1)/\ln 2 = (-2.302585)/(0.693147) = -3.321928$.
- Term 1: $-0.1 \times (-3.321928) = 0.332193$.
- $\log_2(0.9) = \ln(0.9)/\ln 2 = (-0.105361)/(0.693147) = -0.152003$.
- Term 2: $-0.9 \times (-0.152003) = 0.136803$.

$$\text{Sum: } H_2(0.1) = 0.332193 + 0.136803 = 0.468996 \approx 0.4690 \text{ bits.}$$

$$\text{Classical list-decoding capacity threshold: } R^* = 1 - H_2(0.1) = 1 - 0.4690 = 0.5310.$$

By threshold equality [2], this is also the per-sector threshold for random CSS codes for the corresponding folded quantum-LCL property.

4.2 A concrete rate allocation for random CSS codes

Choose per-sector rates $R_X = R_Z = 0.3$. Check against threshold: $0.3 < 0.5310$, with slack $\epsilon_{\text{sector}} = 0.5310 - 0.3 = 0.2310$ in each sector. Total code rate:

$$R = 1 - R_X - R_Z = 1 - 0.3 - 0.3 = 0.4.$$

For block length $n = 100,000$ qubits:

- $\text{rank}(H_X) = R_X \cdot n = 0.3 \times 100,000 = 30,000$.
- $\text{rank}(H_Z) = 30,000$.
- $k = n - 30,000 - 30,000 = 40,000$ logical qubits.

Both sectors lie strictly below the per-sector threshold 0.5310, so by the threshold theorem of [2], a random CSS code from this ensemble is, with high probability, list-decodable to radius 0.1 in both sectors — i.e., the X-sector and Z-sector codes each tolerate a 10% error fraction with bounded lists.

4.3 List size

For rate $R = 0.4$ and radius $\rho = 0.1$, the gap to capacity is

$$\epsilon = 1 - R - H_2(\rho) = 1 - 0.4 - 0.4690 = 0.1310.$$

The standard list-size bound for random linear codes at constant gap from capacity is $L = O(1/\epsilon)$; a representative constant from the classical theory that the explicit constructions of [2] match gives

$$L \approx 2/\epsilon = 2/0.1310 = 15.27 \rightarrow \text{round up: } L \approx 16.$$

The significance of [2] is that this list size is achieved by *explicit* qLDPC codes, not merely random ones — the first such construction for quantum codes. We caution that the constant 2 is a representative value from classical list-decoding theory; the exact constant in the quantum-LCL setting could differ by a small multiplicative factor.

4.4 Overhead comparison at matched distance

Take a good qLDPC code with rate $R = 1/2$ and distance $d = 100$. The block length cannot be set to $n = 200$: a relative distance $\delta = d/n = 1/2$ at rate $1/2$ violates the quantum Hamming (sphere-packing) bound, which at rate $R = 1/2$ requires $H_2(\delta) \leq 1 - R = 1/2$, i.e., $\delta \leq H_2^{-1}(0.5) \approx 0.11$. We therefore fix the largest Hamming-bound-feasible relative distance, $\delta = 0.11$, giving the required block length $n = d/\delta = 100/0.11 \approx 910$ and $k = R \cdot n \approx 455$ logical qubits. The physical qubits per logical qubit are then

$$n/k = 910/455 = 2,$$

which is a rate statement ($n/k = 1/R$) independent of the distance d ; the value 2 does not by itself certify that the code achieves $d = 100$ at $n \approx 910$ — that requires the distance guarantee to hold at the stated relative distance $\delta \approx 0.11$, which is feasible under the Hamming bound but not automatically delivered by every good-code construction.

Surface-code baseline at $d = 100$: $2d^2 - 1 = 2 \times 10,000 - 1 = 19,999$ physical qubits per logical qubit.

Overhead ratio: $19,999 / 2 = 9,999.5 \approx 10^4$.

So, conditional on the rate- $1/2$ code actually achieving $d = 100$ at the Hamming-bound-feasible block length $n \approx 910$ ($\delta \approx 0.11$), it uses roughly 10^4 times fewer physical qubits per logical qubit than the surface code at matched distance. The ratio $19,999/2 \approx 10^4$ compares the surface code's distance-driven overhead against the qLDPC code's rate-driven overhead; it is valid only when the qLDPC n is large enough to realize $d = 100$, not at the impossible $n = 200$. This is the arithmetic behind the "constant qubit overhead" claim of [7], made concrete.

4.5 Projection: decoding cost

Projection (assumptions stated): if the explicit list-decodable qLDPC codes of [2] admit decoders with per-round classical cost comparable to belief propagation, and if predecoding as in [5] reduces decoder invocation frequency by a factor f , then total classical compute scales as $(n/k) \cdot (1/f)$ per logical qubit per round. With $n/k = 2$ and a hypothetical $f = 4$, the cost is 0.5 decoder-equivalents per logical qubit per round. We emphasize: $f = 4$ is an assumption, not a measurement; no empirical decoding data for the codes of [2] exists in our sources.

4.6 Fixed-list-size threshold variant (documented convention)

For completeness we record the fixed-list-size convention used by one source draft: $RL = 1 - H_2(p) - 1/L$. At $p = 0.1$: for $L = 2$, $R_2 = 1 - 0.468996 - 0.5 = 0.031004$; for $L = 4$, $R_4 = 1 - 0.468996 - 0.25 = 0.281004$. Under this convention, a per-sector rate of 0.281 at $(p, L) = (0.1, 4)$ leaves slack $0.5310 - 0.2810 = 0.2500 = 1/L$ against the capacity threshold, showing that the list-size requirement, not distance, is the binding constraint when the distance threshold is $R = 1 - H_2(\delta)$ at $\delta = 0.1$. This variant is consistent with the capacity-form threshold of Section 4.1 (it is the capacity threshold minus the list-size penalty); we adopt the capacity form in the main text because it is the convention under which threshold equality in [2] is stated.

5. Results

All numbers below are computed in Section 4 unless labeled projections.

1. **Per-sector threshold at radius 0.1:** $R^* = 1 - H_2(0.1) = 1 - 0.4690 = \mathbf{0.5310}$. By threshold equality [2], this is simultaneously the classical and quantum per-sector threshold.
1. **Feasible CSS rate allocation:** $R_X = R_Z = 0.3$, total rate $R = \mathbf{0.4}$; at $n = 100,000$ this yields $k = \mathbf{40,000}$ logical qubits, with both sectors 0.2310 below threshold.
1. **List size at $R = 0.4$, $\rho = 0.1$:** $\varepsilon = 0.1310$, giving $L \approx 2/\varepsilon \approx \mathbf{16}$ (constant, achieved explicitly by the qLDPC constructions of [2], up to the constant-factor caveat of Section 4.3).
1. **Overhead at $d = 100$:** good qLDPC code at rate $1/2$: 2 physical qubits per logical qubit (a rate statement, $n/k = 1/R$, independent of d); achieving $d = 100$ at rate $1/2$ requires $n \approx 910$ ($\delta \approx 0.11$, Hamming-bound-feasible); surface code: **19,999**; ratio $\approx \mathbf{10^4}$, conditional on the distance guarantee holding at that block length.
1. **Fixed-list-size thresholds (variant convention):** $R_2 = 0.0310$, $R_4 = 0.2810$ at $p = 0.1$ (Section 4.6).
1. **Projection (stated assumptions):** decoder cost of ~ 0.5 decoder-equivalents per logical qubit per round under the assumed predecoding factor $f = 4$; uncertainty is unquantified because no empirical data exists.

6. Discussion

We argue against our own conclusions. First, the threshold equality of [2] is a statement about *random* CSS ensembles; the explicit constructions inherit the parameters asymptotically, but nothing in the framework guarantees finite-length performance at $n = 100,000$, and our Section 4.2 numbers should be read as ensemble-typical, not guaranteed for the explicit codes at that length. The corpus evidence of [11] — that small explicit stabilizer codes (Golay-type excepted) sit at random baselines rather than optimal parameters — reinforces that finite-length penalties may be severe, and the quantum-LCL theory provides no quantitative concentration bounds that we could extract.

Second, the overhead comparison in Section 4.4 is deliberately favorable to qLDPC codes: it ignores check-measurement locality. As [9] shows, on 2D-local architectures the effective overhead of high-rate qLDPC codes can become prohibitive, potentially erasing the 10^4 factor; conversely, photonic platforms [8] and entanglement-assisted schemes [4] are precisely the settings where non-local checks are cheap, so the factor is architecture-contingent, not universal. The quantum-LCL framework constrains check density but is indifferent to check geometry; a folded construction could have long-range checks that are expensive on planar hardware.

Third, list decodability is a combinatorial property, not a decoder: no efficient algorithm is known in our sources that realizes the list-size-16 guarantee of Section 4.3 at runtime, and the classical-resource analysis of [5] suggests naive decoders would be the bottleneck at scale. Moreover, list decoding is a worst-case guarantee; practical channels (as in [4]'s purification setting or [3]'s GKP-concatenated setting) may be well served by simpler decoders, and the marginal value of optimal list sizes is unquantified in those works.

Fourth, the encoded rate of a CSS code depends on the shared structure between the two sectors; the threshold theorem's per-sector formulation sidesteps this by taking the per-sector rate, rather than a joint k/n , as the invariant. This is the correct reading of the theorem, but it means per-sector thresholds do not by themselves pin down the effective k/n of the explicit constructions.

Fifth, the bosonic-encoding evidence of [10] (5–40× photon savings) is measured against surface codes, not qLDPC codes; whether it transfers is open. And the LCL framework's linear-algebraic locality is of a different character than the ultrametric locality of [13]; any connection is speculative.

Failure modes and falsifiability. The claim that per-sector thresholds equal classical thresholds would be falsified by exhibiting a folded quantum-LCL property for which the random CSS threshold deviates from the classical value in either sector. Our overhead claims would be falsified by a lower bound showing constant-rate qLDPC codes require $\omega(1)$ qubits per logical qubit on any feasible architecture. The list-size claim would be falsified if the explicit constructions of [2] required list sizes growing with n . The claim of practical significance would be falsified if the folded constructions require check weights growing with n (breaking qLDPC in the strict sense) — a concrete verification task against [2]'s construction section — or if concentration requires block lengths so large that near-term relevance collapses.

Open questions. (i) Do the explicit quantum-LCL codes admit efficient decoders matching their combinatorial guarantees? (ii) Can the nested-space framework be extended beyond CSS to general stabilizer codes, where the $S \subseteq C$ structure is less clean? (iii) How do spatially-coupled constructions [6] compare to LCL-derandomized codes at finite length, and can LCL codes be spatially coupled without losing check locality? (iv) Is there any interaction between the LCL framework and the p-adic/ultrametric classification program [11], [13]? (v) A limitation of this paper's sources: the bibliography mixes peer-reviewed work with preprints of varying maturity, including the unrefereed [1], [2] themselves; all conclusions conditional on [2] inherit its verification status.

7. Conclusion

The quantum-LCL framework of [1], [2] resolves a conceptual obstruction — the two-rank structure of local witnesses in CSS codes — and delivers a clean structural theorem: random CSS codes inherit classical thresholds sector by sector. Our analysis shows the theorem has sharp numerical teeth: at error radius 0.1, the threshold is $R^* = 0.5310$; a rate-0.4 CSS allocation with 40,000 logical qubits per 100,000 physical sits comfortably below it with constant list size ≈ 16 ; and the explicit qLDPC derandomization translates into a $\approx 10^4$ -fold qubit-overhead advantage over surface codes at distance 100 on locality-permissive architectures — a comparison that holds at the Hamming-bound-feasible operating point (rate 1/2, $\delta \approx 0.11$, $n \approx 910$), not at the impossible $n = 200 = 2d$. The remaining distance between these combinatorial guarantees and deployable, efficiently decodable, locally implementable codes is the field's next problem, and the decoding-focused literature [4], [5] suggests it is where the practical payoff of the LCL program will be decided.

References

- [1] TITLE: arXiv Query: `searchquery=&idlist=2609.40252&start=0&max_results=1` [2] From Random Quantum Codes to Explicit qLDPC Codes via Local Properties. arXiv:2609.40252v1. <https://arxiv.org/abs/2609.40252v1> [3] Finite Rate QLDPC-GKP Coding Scheme that Surpasses the CSS Hamming Bound. arXiv:2111.07029v2. <https://arxiv.org/abs/2111.07029v2> [4] Entanglement Purification with Quantum LDPC Codes and Iterative Decoding. arXiv:2210.14143v2. <https://arxiv.org/abs/2210.14143v2> [5] Mitigating Classical Resource Costs in Quantum Error Correction via Generalized qLDPC Predecoding. arXiv:2605.03180v2. <https://arxiv.org/abs/2605.03180v2> [6] Spatially-Coupled QLDPC Codes. arXiv:2305.00137v6. <https://arxiv.org/abs/2305.00137v6> [7] Accelerating Fault-Tolerant Quantum Computation with Good qLDPC Codes. arXiv:2510.19442v3. <https://arxiv.org/abs/2510.19442v3> [8] Fusion-based implementation of qLDPC codes with quantum emitters. arXiv:2509.17223v2. <https://arxiv.org/abs/2509.17223v2> [9] Toward a 2D Local Implementation of Quantum LDPC Codes. arXiv:2404.17676v2. <https://arxiv.org/abs/2404.17676v2> [10] DOI

pending. QNFO: Bosonic Codes as the Native Encoding: Resource-Commensurable Comparison of Cat, GKP, Binomial, and Surface Codes. [11] DOI 10.5281/zenodo.21754148. QNFO: Extending $v_p^{\max}$ Code Classification: Testing the Mahler Spectral Conjecture on Additional Stabilizer Code Families. [12] QNFO: FACTORING, Adelic Complexity, and the Silent-Radix Principle [13] DOI 10.5281/zenodo.22749408. QNFO: Qudit Quantum Error Correction.

Appendix A. Divergence report

D1. Threshold parameterization (capacity form vs. fixed-list-size form).

- Drafts A and B: threshold $R^* = 1 - H_2(p) = 0.5310$ at $p = 0.1$ (capacity/list-decoding capacity form). A calls it the "classical capacity bound"; B calls it the list-decoding capacity threshold. These are numerically identical and substantively compatible.
- Draft C: threshold $R = 1 - H_2(p) - 1/L$ for fixed list size L , giving $R = 0.0310$ ($L = 2$) and 0.2810 ($L = 4$) at $p = 0.1$.
- Underlying convention: the capacity form states the threshold for existence of *some* bounded list; the fixed- L form states the threshold for a *specified* list size. Both are consistent (the fixed- L threshold is the capacity threshold minus the list-size penalty $1/L$).
- Resolution: main text adopts the capacity form (B's convention, numerically shared with A) because it is the form under which threshold equality in [2] is stated; the fixed- L variant is recorded in Section 4.6 with full arithmetic. Not silently resolved.

D2. CSS rate accounting.

- Draft A: single-pool accounting $k = n - m$ with $n = 1000$, $m = 400$, giving $k = 600$ and $R = 0.60$, then compares $R = 0.60$ against $R^* \approx 0.531$ and concludes the code "exceeds" threshold.
- Drafts B and C: two-pool CSS accounting $k = n - \text{rank}(HX) - \text{rank}(HZ)$, i.e., $R = 1 - RX - RZ$; a code must lie *below* threshold in each sector to satisfy the property.
- Underlying convention: A treats the stabilizer pool as a single deduction (appropriate at most for one sector or a degenerate reading); B/C treat the two sectors separately, which is required by the per-sector structure of the threshold theorem. Under B/C's convention a rate of 0.60 with $RX = RZ = 0.3$ each is *below* threshold ($0.3 < 0.5310$) — the same numbers, correctly read, support the theorem rather than "exceeding" it.
- Resolution: main text adopts the two-pool per-sector accounting (B, C) as the convention consistent with the per-sector form of the threshold theorem. A's arithmetic ($600 = 1000 - 400$; $0.6 = 600/1000$) is retained only as arithmetic, reinterpreted under the per-sector convention.

D3. List size.

- Draft B: $L \approx 2/\epsilon \approx 16$ at $R = 0.4$, $\rho = 0.1$ (constant from classical theory; constant-factor caveat stated).
- Draft C: works with fixed $L \in \{2, 4\}$ as design parameters.
- Underlying convention: B derives the list size from the gap to capacity; C fixes the list size and derives the threshold. These are inverse parameterizations of the same trade-off.
- Resolution: main text presents B's derivation (Section 4.3) and C's inverse parameterization (Section 4.6); both are labeled with their conventions.

D4. Folding overhead projection.

- Draft A: assumes a folding factor $f = 3$, giving effective length $n_{\text{eff}} = 3000$ and effective rate 0.20.

- Drafts B and C: make no folding-factor assumption; B's overhead analysis uses the good-code model of [7] instead.
- Underlying convention: A's $f = 3$ is an illustrative assumption not grounded in [2]'s construction; B/C avoid it.
- Resolution: the $f = 3$ projection is excluded from the main text (it rests on an unstated grounding); the [7]-based overhead model is used instead. Documented here rather than silently dropped.

D5. Hamming-bound consistency check (Draft C only). Draft C's Computation 3 contains internally inconsistent intermediate arithmetic (a sign error in evaluating $1 - H(p) - 2/L$, self-corrected mid-derivation) and concludes the per-sector formulation is consistent with sphere-packing provided a shared-structure term s is accounted for. Drafts A and B perform no such check. Resolution: the qualitative conclusion (per-sector thresholds do not contradict the quantum Hamming bound; the shared-structure term is not pinned down by the theorem) is retained in Section 6 as a single-draft claim, with the flawed intermediate arithmetic not reproduced.

Appendix B. Claim attribution

#	Claim	A	B	C	Status
C1	Quantum-LCL framework handles nested spaces $S \subseteq C$ with physical and logical ranks	✓	✓	✓	CONVERGENT
C2	Threshold theorem: per-sector rate threshold for random CSS codes equals classical threshold	✓	✓	✓	CONVERGENT
C3	Explicit folded quantum-LCL constructions are qLDPC and achieve optimal list sizes (first explicit quantum list-decodable/list-recoverable codes)	✓	✓	✓	CONVERGENT
C4	$H_2(0.1) = 0.468996$; threshold $R^* = 0.5310$ at radius 0.1	✓	✓	✓	CONVERGENT
C5	Threshold parameterized as $1 - H_2(p)$ (capacity form)	✓	✓	—	CONVERGENT
C6	Threshold parameterized as $1 - H_2(p) - 1/L$ (fixed-list-size form)	—	—	✓	SINGLE (documented, D1)
C7	CSS rate accounting $k = n - m$, $R = 0.60$ at $n = 1000$, $m = 400$	✓	—	—	SINGLE; conflicts with C8 (D2)
C8	CSS rate accounting $R = 1 - RX - RZ$; concrete allocation $RX = RZ = 0.3$, $R = 0.4$, $k = 40,000$ at $n = 100,000$	—	✓	✓	CONVERGENT
C9	List size $L \approx 2/\epsilon \approx 16$ at $R = 0.4$, $\rho = 0.1$, with constant-factor caveat	—	✓	—	SINGLE (D3)
C10	Fixed list sizes $L \in \{2, 4\}$ as design parameters with thresholds $0.0310 / 0.2810$	—	—	✓	SINGLE (D3)
C11	Overhead: good qLDPC at rate $1/2$, $d = 100 \rightarrow 2$ physical qubits/logical vs. 19,999 for surface code; ratio $\approx 10^4$	—	✓	—	SINGLE (grounded in [7])
C12	Folding factor $f = 3$ projection: $n_{\text{eff}} = 3000$, effective rate 0.20	✓	—	—	SINGLE (excluded from main text, D4)
C13	Per-sector thresholds consistent with quantum Hamming bound given shared-structure term	—	—	✓	SINGLE (qualitative conclusion retained, D5)
C14	List decoding, not distance, is the binding constraint at $p = \delta = 0.1$, $L = 4$	—	—	✓	SINGLE
C15	Finite-length / concentration behavior not pinned down by the theory; small explicit codes far from random-code behavior ([11])	—	✓	✓	CONVERGENT
C16	Check geometry (2D-local layout, [9]) can erode qLDPC overhead advantages; photonic platforms ([8]) less affected	—	✓	✓	CONVERGENT
C17	List decoding is combinatorial, not an algorithm; efficient decoders unknown; predecoding [5] relevant	—	✓	✓	CONVERGENT

#	Claim	A	B	C	Status
C18	Systems relevance: [3], [4], [7], [8] pipelines benefit from explicit random-parameter qLDPC codes	✓	✓	✓	CONVERGENT
C19	Bosonic savings (5–40× photons, [10]) measured vs. surface codes; transfer to qLDPC open	—	✓	✓	CONVERGENT
C20	[12] not substantively discussable (no retrievable abstract)	—	—	✓	SINGLE
C21	Falsification conditions: threshold deviation in either sector, list sizes growing with n , check weights growing with n (breaking qLDPC), or $\omega(1)$ qubits per logical qubit on any feasible architecture	✓	✓	✓	CONVERGENT