A subexponential algorithm for the erasure logical error rate of sparse stabilizer codes

Oliver D'Souza*

12 November 2026

Abstract

How well a quantum error-correcting code protects against the loss of qubits is measured by the logical error rate $I_H(p)$ of the optimal erasure decoder, as a function of the loss probability p. For a stabilizer code on n qubits this is an average over all 2^n erased sets; it is exponentially small at the loss rates of interest, where simulation sees no failures, and we know of no algorithm faster than enumeration. We show that the whole curve can be computed to relative accuracy with a SAT solver, using polynomially many satisfiability questions about formulas whose size is *linear* in the size of the code. Precisely, let H be a sparse check matrix and $F_H = 1 - I_H$ the optimal entanglement fidelity. For any bounded-density 3-SAT algorithm with time $\mathrm{poly}\cdot T(v)$ and polynomial space, a randomized algorithm computes, in time $\mathrm{poly}(\mathrm{input}, \varepsilon^{-1}, \log \delta^{-1})\, T(O(n))$ and polynomial space, a monotone polynomial that approximates *both* I_H and F_H to relative error ε, simultaneously at every $p \in [0,1]$, with probability $1-\delta$. Two features separate this from approximate counting with an oracle. The logical error rate is itself a projected count, through a second certificate complementary to the one for the fidelity, which gives relative accuracy in the tail that matters; and every query has $O(n)$ variables, without which a subexponential SAT algorithm would yield nothing. With the machine-checked subexponential 3-SAT algorithm of OpenAI the running time is $\mathrm{poly}\cdot 2^{O(n(\log\log n)^2/\sqrt{\log n})}$ with polynomial space, unconditionally, and the reduction, the engine and their composition are kernel-checked in one Lean 4 development with standard axioms only. Conversely, any output meeting the guarantee determines the code distance, and so does a single value of I_H at a small explicit p known up to an exponential factor; the task is therefore NP-hard already for sparse codes. The speed-up is asymptotic; for codes of practical size we combine the same certificate with sampling and compute logical error rates down to 10^{-28} for surface, toric and bivariate bicycle codes of up to 128 qubits, with the distance, the minimum-size failure sets and the leading coefficient certified by proof-checked SAT refutations. The reduction extends to correlated erasures generated by linear-size circuits from latent bits in a fixed number of probability classes. The Lean development, the code and the data are available at https://github.com/Oliverds321/stabilizer-erasure-sat-lean.

1 Introduction

A stabilizer code that is used on hardware whose dominant fault is a detectable loss, such as neutral-atom, trapped-ion and dual-rail superconducting qubits [77, 85, 102, 121], is characterized by its performance under *flagged erasure*: each qubit is lost independently with probability p, and the decoder is told which qubits were lost. For one fixed erased set E the question is linear algebra over $\mathbb{F}_2$: the optimal recovery succeeds with probability 2^{-d_E}, where d_E counts the logical degrees

*Copyright © 2026 Oliver D'Souza. This paper is licensed under Creative Commons Attribution 4.0 International (CC BY 4.0).

of freedom supported on E (Lemma 7; [39, 61]). Averaging over all 2^n erased sets is a different task. The result,

$$F_H(p) = \mathbb{E}_E[2^{-d_E}] = \sum_{j=0}^{n} f_j \binom{n}{j} p^j (1-p)^{n-j}, \qquad I_H(p) = 1 - F_H(p),$$

is the optimal entanglement fidelity of the code at erasure rate p, and I_H is the logical error rate of the optimal erasure decoder. The curve is what one reports when comparing codes, and its small-p tail, where $I_H(p)$ is exponentially small for good codes, is exactly what is hardest to estimate: direct sampling of erasure patterns needs $\Omega(1/I_H(p))$ samples to see a single failure (Section 6.4). This paper is about evaluating a code, not about decoding it: nothing here makes error correction itself faster.

The question. Given a *sparse* presentation of the code, a check matrix H with $2n$ columns and $O(n)$ nonzero entries, how fast can the whole curve be computed to relative accuracy ε in both tails? The trivial algorithm enumerates all 2^n sets and computes a rank for each, in $2^n \operatorname{poly}(n)$ time and polynomial space. We know of no faster algorithm for arbitrary sparse codes in the literature, and of no stated hardness result for the curve.

Main result. We show that the curve is no harder than bounded-density 3-SAT, up to a polynomial factor and a *linear* change of instance size. Write $T(v)$ for the running time of a deterministic polynomial-space algorithm for 3-CNF satisfiability on v variables with a fixed clause density. Theorem 1 gives a randomized algorithm that outputs rational Bernstein coefficients $1 = \widehat{f}_0 \geq \cdots \geq \widehat{f}_n = 4^{-k}$ such that, with probability $1-\delta$, simultaneously for every $p \in [0,1]$,

$$|\widehat{F}(p) - F_H(p)| \leq \varepsilon F_H(p) \quad \text{and} \quad |\widehat{I}(p) - I_H(p)| \leq \varepsilon I_H(p),$$

in time $\operatorname{poly}(L, \varepsilon^{-1}, \log \delta^{-1})\, T(C_c n)$ and polynomial space, where L is the input length and C_c depends only on the sparsity constant. The linear blow-up is essential: an $O(n \log n)$ reduction would turn a $2^{O(n/\sqrt{\log n})}$ SAT algorithm into a bound *worse* than 2^n. Figure 1 shows the structure of the reduction. The gain over enumeration is asymptotic: with the constants of the engines available today, enumeration is faster for every code that will be built (Remark 6), and Section 7 gives a practical method that uses the same certificate with an ordinary SAT solver.

The Lean-verified theorem of OpenAI [95] supplies a polynomial-space engine with running time $2^{O(v(\log\log v)^2/\sqrt{\log v})}$ on v variables. Corollary 3 then gives an unconditional algorithm with the same rate in n and polynomial space; the reduction, the engine and their composition are machine-checked (Section 9), so the corollary rests on no unrefereed proof. Remarks 4 and 5 give the rate $2^{O(n/\sqrt{\log n})}$ under two further instantiations.

Why both tails, and why a SAT oracle. Relative approximation of F_H alone is easy near $p = 0$, where $F_H \approx 1$; relative approximation of I_H alone is easy near $p = 1$. The content of the guarantee is that one coefficient vector controls both, including a tail that may be $e^{-\Theta(n)}$. In the classical theory of network reliability the two tails are genuinely different problems, with separate approximation schemes [62, 80] and in one case a hardness separation [48, 49]. Here a single counting event gives both, because the per-pattern fidelity 2^{-d_E} and infidelity $1 - 2^{-d_E}$ are the fractions of *complementary* projected sets of the same size $|N|$ (Proposition 8).

The oracle is needed. Proposition 12 shows that *any* output meeting the two-tail guarantee with $\varepsilon < 1$ determines the minimum distance of the code: the first nonzero infidelity coefficient

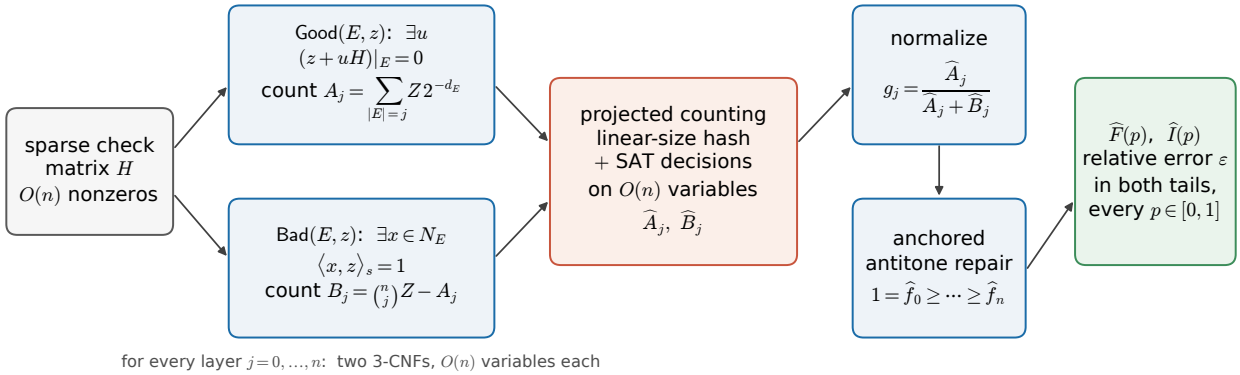


Figure 1: The reduction. For each erasure weight j, two existential certificates over the same pairs (E, z) have complementary projected counts A_j and B_j. Each count is estimated by hashing with a linear-size circuit and SAT decisions on $O(n)$ variables. Normalizing the two estimates and repairing the shape gives one coefficient vector that controls the fidelity and the infidelity curve at once.

sits at index d. Computing the distance of a stabilizer code is NP-hard [79, 119], and appending qubits that each carry a single-qubit check makes any presentation sparse without changing the curve (Corollary 14). So for every sparsity constant $c > 1$ no polynomial-time algorithm meets the guarantee unless NP $\subseteq$ BPP. The hardness is already present at a single point: one value $I_H(p)$ at an explicitly given small p, known only up to an exponentially large factor, determines the distance (Proposition 15). What remains open is a *fine-grained* lower bound, since the padding is not size-preserving, and hardness for presentations of bounded row and column weight (Section 11).

Why approximate rather than count. The external engine counts models exactly. We nonetheless reduce to *decision* and approximate, because the certificates are not parsimonious: the number of witnesses of a positive instance depends on E (Proposition 8), so witness totals are not the counts that define the curve. Projected counting onto the bits (E, z) removes the multiplicity, and hashing-based projected counting needs only a decision oracle and keeps every query linear in n (Appendix B).

Contributions.

1. *Complementary projected certificates with linear incidence* (Section 4). Two existential predicates on (E, z), z in the normalizer N, whose counts are $|N|2^{-d_E}$ and $|N|(1 - 2^{-d_E})$, each a projection of a 3-CNF with $O(n + \mathrm{nnz}(H))$ variables and clauses, where $\mathrm{nnz}(H)$ is the number of nonzero entries of H. This is a projected use of cleaning duality [15, 76]; it needs no logical-operator basis and no rank computation inside the circuit.

2. *A linear-size transfer with no loss in the shape step* (Sections 5 and 8, Appendices A and B). An all-length adaptation of the linear-size pairwise-independent hash of Ishai, Kushilevitz, Ostrovsky and Sahai [71] keeps every SAT query at $O(n)$ variables of bounded clause density, and an anchored antitone repair on a finite partial order (Lemma 11) restores monotone shape and exact endpoints while preserving the two-tail relative interval exactly. The same lemma handles the iid case, the uncorrectable-erasure probability, monotone latent circuits and the multiclass tensor case.

3. *An end-to-end machine-checked complexity bound* (Section 9). The reduction is formalized as a concrete multitape machine whose run, output, time and space are proved in Lean 4; the external verified SAT engine is linked through proved encoding and resource bridges in a single 6,157-module graph, so that the final statements have no SAT, hashing or solver hypothesis. To our knowledge this is the first machine-checked time and space bound for an algorithm that calls a machine-checked subexponential SAT engine.

We also give the hardness side (distance determination from the curve and from a single value, and NP-hardness for sparse presentations through a direct reduction from SAT to the distance of a CSS code with one logical qubit, whose correctness is machine-checked), an exact separation showing that ordinary weight enumerators do not determine the curve, a sampling-model lower bound, extensions to correlated erasures generated by linear-size circuits from latent bits, including a fixed number of latent probability classes with a multivariate monotone output (Theorem 20), and computed examples on codes of up to 128 qubits (Section 7).

Scope. The theorems are asymptotic: they establish the fine-grained transfer from satisfiability to the erasure curve, and Section 7 supplies the estimator for real codes, a hybrid whose error bars are statistical. The results concern flagged erasure of stabilizer codes with a maximally mixed logical input. The constants, the exact extent of the formalization and the open problems are collected in Sections 9 and 11.

2 Model and objective

Fix a constant $c > 0$. The input is a binary check matrix H with $2n$ columns and at most cn nonzero entries. Its row space S is isotropic for the standard symplectic form on $W = \mathbb{F}_2^{2n}$. Delete zero rows, but retain arbitrary remaining redundancies. Write

$$s = \dim S, \qquad k = n - s, \qquad N = S^{\perp}, \qquad Z = |N| = 2^{2n-s}.$$

A consistent choice of stabilizer signs defines an $[[n, k]]$ code; the quantities studied here do not depend on that choice. For each qubit, independently, erasure occurs with the same probability p. The recovery receives the erased set as a classical flag.

Let $D = 2^k$, let Φ_D be the normalized maximally entangled state of a logical system and a reference, and let $\mathcal{N}_{H,p}$ be the channel that encodes the logical system, erases each qubit independently with probability p by replacing it with a fixed state and recording the erased set in a classical flag register. Define

$$F_H(p) = \max_{\mathcal{R}} \langle \Phi_D | (\mathrm{id} \otimes \mathcal{R} \circ \mathcal{N}_{H,p})(|\Phi_D\rangle\langle\Phi_D|) | \Phi_D \rangle, \qquad I_H(p) = 1 - F_H(p). \tag{1}$$

The maximum is over all CPTP recoveries to the logical system that may read the flag. Thus F_H is the entanglement fidelity for the maximally mixed logical input; throughout, "entanglement fidelity" means this squared overlap. It is not worst-case fidelity, and it is not the probability that the erased region is perfectly correctable.

What the curve measures. An erased region is perfectly correctable exactly when $d_E = 0$ (Section 4). The probability $U_H(p) = \mathbb{P}(d_E > 0)$ of an uncorrectable pattern therefore differs from $I_H(p)$, although

$$I_H(p) \le U_H(p) \le 2 I_H(p).$$

Indeed $1 - 2^{-d_E}$ is zero when $d_E = 0$ and belongs to $[1/2, 1]$ otherwise; the comparison is noted in [4], where U_H is the decoding-error probability P_{de}. This constant-factor relation is not an ε-relative approximation of U_H. For an optimal syndrome/Pauli recovery, 2^{-d_E} is the probability of selecting the correct logical class in the induced Pauli mixture, so $I_H(p)$ is the logical error rate of the maximum-likelihood erasure decoder with uniform tie-breaking. That event supplies no herald of successful recovery. The Haar-average pure-state fidelity is instead $(DF_e + 1)/(D + 1)$, where F_e is the entanglement fidelity of (1) and $D = 2^k$. For one erased repetition-code qubit these three quantities are, respectively, correctability zero, entanglement fidelity $1/2$, and average fidelity $2/3$. Corollary 17 approximates U_H directly.

3 Main results

Theorem 1 (Quantitative transfer)**.** *Let $T\colon \mathbb{N} \to \mathbb{N}$ be nondecreasing, and suppose that for some fixed clause density $\Delta > 0$ there is a deterministic algorithm that decides satisfiability of 3-CNFs with v variables and at most Δv clauses in time $\mathrm{poly}(\ell)\, T(v)$ and space $\mathrm{poly}(\ell + v)$, where ℓ is the length of the binary encoding of the formula. Then there is a randomized algorithm that, given H as in Section 2 and rational $0 < \varepsilon, \delta < 1$, returns rational coefficients*

$$1 = \widehat{f}_0 \geq \widehat{f}_1 \geq \cdots \geq \widehat{f}_n = 4^{-k}$$

such that, writing

$$\widehat{F}(p) = \sum_{j=0}^{n} \widehat{f}_j \binom{n}{j} p^j (1-p)^{n-j}, \qquad \widehat{I}(p) = \sum_{j=0}^{n} (1 - \widehat{f}_j) \binom{n}{j} p^j (1-p)^{n-j},$$

with probability at least $1 - \delta$, simultaneously for every real $p \in [0, 1]$,

$$|\widehat{F}(p) - F_H(p)| \leq \varepsilon F_H(p), \qquad |\widehat{I}(p) - I_H(p)| \leq \varepsilon I_H(p). \tag{2}$$

Every run produces curves in $[0, 1]$ with exact endpoints, $\widehat{F} + \widehat{I} = 1$, nonincreasing $\widehat{F}$, and nondecreasing $\widehat{I}$. Zero infidelity coefficients are identified deterministically and retained exactly.

Let L be n plus the total bit length of the encodings of $H, n, \varepsilon, \delta$. The algorithm runs in time

$$\mathrm{poly}(L, \varepsilon^{-1}, \log \delta^{-1})\, T(C_c n)$$

for a constant C_c depending only on c, and in space polynomial in L. The case $n = 0$ is returned directly.

Proposition 2 (Resource accounting)**.** *In Theorem 1, for $n \geq 1$, the algorithm makes $O(n^3 \varepsilon^{-2} \log((n+1)/\delta))$ SAT queries, each a 3-CNF with $O_c(n)$ variables and $O_c(n)$ clauses, and each output coefficient has $O(n + \log \varepsilon^{-1})$ bits. The clause density of the queries is bounded by a constant depending only on c, independent of ε, δ; a bound for any specified positive density suffices after padding queries with unused variables at a constant-factor cost. The time bound of Theorem 1 uses only the time premise on the SAT algorithm; the space bound uses the space premise.*

The input-length convention includes reading long rational precision parameters, even when their numerical values remain bounded; without it the polynomial prefactor would omit the cost of reading ε, δ.

Instantiations. The following external results supply T. We separate what is machine-checked from what is claimed.

Corollary 3 (Unconditional instantiation)**.** *There is a randomized algorithm with the guarantees of Theorem 1 running in time*

$$\mathrm{poly}(L, \varepsilon^{-1}, \log \delta^{-1})\, 2^{O_c(n(\log\log n)^2/\sqrt{\log n})} \tag{3}$$

and polynomial space.

Proof. The Lean-verified theorem `SqrtLogSpace.sat_polynomial_space` of [95] (commit `4eaee875`; see Section 9) gives, for every $k \geq 3$, a fixed machine that decides satisfiability of width-k CNFs on v variables with formula bit length ℓ in time at most $C(\ell+1)^d 2^{A\rho(v)}$, where $\rho(v) = v(\ln\ln(v+16))^2/\sqrt{\ln(v+16)}$, with every execution prefix visiting at most $S(\ell+1)^e$ cells. Put $T(v) = \max_{s\leq v}\lceil 2^{A\rho(s)}\rceil + 1$, which is nondecreasing. Apply Theorem 1. □

The corollary is unconditional in the sense that every hypothesis is a kernel-checked theorem; Section 9 records that the composition of the reduction with this engine has itself been checked, including the translation between the engine's input convention and ours.

Remark 4 (Sharper conditional rate)*.* The manuscript [95] also claims, in an appendix that is not part of its Lean development, that bounded-density k-SAT is decidable in polynomial space and time $2^{O(n/\sqrt{\log n})}$. If that claimed algorithm is correct, then Theorem 1 gives time

$$\mathrm{poly}(L, \varepsilon^{-1}, \log \delta^{-1})\, 2^{O_c(n/\sqrt{\log n})} \tag{4}$$

with polynomial space. We do not rely on this rate anywhere else.

Remark 5 (Time-only rate from a second verified engine)*.* The Lean-verified main theorem of [31] decides k-CNFs with m clauses on n variables, and counts their models, in time $2^{C\min\{km/\sqrt{\log m},\, k^2 n \log\log n/\sqrt{\log n}\}}$ on Mathlib's multi-stack machine model, with no space bound. Our queries, after deletion of unused variables and duplicate clauses, have $m = O(v)$, so the time bound of Theorem 1 becomes $\mathrm{poly}\cdot 2^{O_c(n/\sqrt{\log n})}$ with no space bound. This is a paper-level observation: that machine model has not been connected formally to ours.

Remark 6 (Comparison with the trivial algorithm)*.* Exhaustive enumeration takes $2^n\,\mathrm{poly}(n)$ time and polynomial space. The bound (3) improves on it only once $C_c A\,(\log\log n)^2/\sqrt{\log n} < 1$, that is, for $n > 2^{\Theta((C_c A)^2)}$ up to log-log factors. With the constants of the formalized construction (Section 11) that crossover is astronomically large. The contribution is the fine-grained transfer; Section 7 gives an estimator for real codes.

4 Sparse positive representations of both tails

For $E \subseteq [n]$, let $V_E \subset W$ be the subspace of Pauli vectors supported on E. Put

$$N_E = N \cap V_E, \qquad S_E = S \cap V_E, \qquad d_E = \dim N_E - \dim S_E.$$

The image of N_E in N/S has dimension d_E, so $0 \leq d_E \leq 2k$.

Lemma 7 (Known fixed-erasure optimum)**.** *For fixed E, the optimum in (1) is $F_{\mathrm{opt}}(E) = 2^{-d_E}$. Consequently $F_H(p) = \mathbb{E}_E[2^{-d_E}]$.*

Proof. Erasure is recovery-equivalent to replacing the erased subsystem by the maximally mixed state: one simulation appends that fixed state, and the other discards it. The replacement is the uniform mixture of the $4^{|E|}$ Pauli errors in V_E. The syndrome map on V_E has kernel N_E. Each reachable syndrome therefore has 2^{d_E} equally likely logical classes, since two errors in its fibre have the same logical action precisely when their difference is in S_E.

For completeness, optimization over all CPTP maps follows from Pauli orthogonality, not a restriction on the decoder. Write a general encoded Pauli channel with Kraus operators $\sqrt{p_{q,\ell}}V_qL_\ell$, where q is a syndrome, V_q maps the logical space isometrically into its syndrome sector, and the D^2 logical Paulis L_ℓ form an orthogonal unitary basis. For arbitrary recovery Kraus operators R_a, put $M_{a,q} = R_aV_q$. Trace preservation and Parseval give

$$\sum_a M_{a,q}^\dagger M_{a,q} = I_D, \qquad \sum_\ell |\operatorname{tr}(ML_\ell)|^2 = D\operatorname{tr}(M^\dagger M).$$

Thus

$$F_e = D^{-2}\sum_{a,q,\ell} p_{q,\ell}|\operatorname{tr}(M_{a,q}L_\ell)|^2 \le \sum_q \max_\ell p_{q,\ell}.$$

Syndrome measurement followed by a maximizing logical correction attains this bound, as in [51, Appendix A]. For erasure the conditional maximum is 2^{-d_E}. Orthogonal classical erasure flags permit independent optimal recoveries for every E, proving the expectation formula. □

The inverse-rank expression appears explicitly in [61, Eqs. (17)–(18)] and, in equivalent form, in the erasure-channel analysis of [39], in [4, Section III] as the reciprocal of the number of logical classes that have a representative supported on the erased set, and in the degenerate-decoding analysis of [87]. It gives fidelity one for empty erasure, 4^{-k} for full erasure, and one for every erasure when $k = 0$. The exponent d_E can be odd. For example, one erased qubit of the three-qubit repetition code has $d_E = 1$.

Proposition 8 (Complementary projected certificates)**.** *Let m be the number of retained rows of H. For $z \in N$, define*

$$\begin{aligned}\mathsf{Good}(E,z) &\iff \exists u \in \mathbb{F}_2^m :\ (z+uH)|_E = 0,\\ \mathsf{Bad}(E,z) &\iff \exists x \in N_E :\ \langle x,z\rangle_s = 1.\end{aligned}$$

These predicates are complementary, and

$$\#\{z \in N : \mathsf{Good}(E,z)\} = Z2^{-d_E}, \qquad \#\{z \in N : \mathsf{Bad}(E,z)\} = Z(1-2^{-d_E}). \tag{5}$$

A positive Good instance has $2^{m-\operatorname{rank} H_E}$ coefficient witnesses, where H_E retains both columns of each qubit in E. A positive Bad instance has $|N_E|/2$ witnesses.

Proof. Nondegenerate finite-dimensional orthogonality gives

$$(N \cap V_E)^\perp = N^\perp + V_E^\perp = S + V_{\bar{E}}.$$

Thus z annihilates N_E exactly when it differs from a vector $uH \in S$ by a vector supported outside E, precisely Good. Over $\mathbb{F}_2$, failure to annihilate means pairing to one with some $x \in N_E$, precisely Bad. This logical equivalence holds even for $z \in W$; the count in (5) uses $z \in N$.

Consider $\phi_E : N \to N_E^*$, with $\phi_E(z)(x) = \langle x, z\rangle_s$. The transpose map $N_E \to N^*$ has kernel $N_E \cap N^\perp = N_E \cap S = S_E$, using $S \subseteq N$. The two maps have equal rank, so $\operatorname{rank}\phi_E = d_E$ and

$$|\ker \phi_E| = 2^{\dim N - d_E} = Z2^{-d_E}.$$

Good is this kernel and Bad its complement in N. Finally $u \mapsto (uH)|_E$ has rank $\operatorname{rank} H_E$, giving the first fibre size. For a positive Bad instance, $x \mapsto \langle x, z\rangle_s$ is a nonzero linear functional on N_E, giving the second. □

This is a projected use of cleaning/annihilator duality [15, 76]. Its counting domain and projection are essential. For a Bell stabilizer state under full erasure, Good has probability one over $N = S$, but probability $1/4$ over the ambient W. Nor can one simply count witnesses: for a Bell pair on qubits $1, 2$ and two free qubits, the weight-two erasures $\{1, 3\}$ and $\{3, 4\}$ give, respectively, two and eight witnesses per positive Bad pair. This E-dependent multiplicity is why Section 5 counts projected assignments with a decision oracle rather than witnesses with a counting oracle.

Lemma 9 (Linear incidence)**.** *For each j, the predicates $(|E| = j, z \in N, \mathsf{Good}(E, z))$ and $(|E| = j, z \in N, \mathsf{Bad}(E, z))$ are projections onto the $3n$ bits (E, z) of 3-CNFs with $O(n + w)$ variables and clauses, where w is the number of nonzero entries of H. They are constructible in polynomial time.*

Proof. Write $H = (H_X \mid H_Z)$. Membership in N is the row-parity system $H_X z_Z + H_Z z_X = 0$, computed by $O(w)$ XOR gates. For Good, columnwise XOR chains compute uH with $O(w)$ gates, and $2n$ implications require equality with z on erased qubits. There are $m \le w$ coefficient-witness bits. For Bad, repeat the row checks for x, impose $2n$ support implications, and compute the symplectic pairing using $2n$ AND gates and an XOR chain.

A carry-save population counter imposes $|E| = j$ with $O(n)$ gates. Each three-to-two compressor reduces the total live-bit count by one; after at most n compressions, a ripple addition and comparison use $O(\log n)$ gates. Full-equivalence Tseitin clauses for binary gates have width at most three and constant size. Existentially quantify the witnesses and all gate auxiliaries, leaving only E, z projected. Both gate and wire counts are linear. □

Let A_j, B_j be the two projected counts in Lemma 9. Proposition 8 implies

$$A_j + B_j = \binom{n}{j} Z, \qquad f_j = \frac{A_j}{\binom{n}{j} Z}, \tag{6}$$

$$F_H(p) = Z^{-1} \sum_{j=0}^{n} A_j p^j (1-p)^{n-j}, \qquad I_H(p) = Z^{-1} \sum_{j=0}^{n} B_j p^j (1-p)^{n-j}. \tag{7}$$

There is no extra binomial factor in (7): the counts already include every set of size j. Every $A_j > 0$, since $z = 0$ is Good for every E.

5 From projected counts to both reliability curves

The following counting tool supplies the quantitative budget. Its all-length hash construction and proof are given in the appendices; the estimator is an adaptation of the hashing method of Stockmeyer [113] and Valiant and Vazirani [118] in the form used by modern approximate model counters [27, 58].

Proposition 10 (Projected counting with linear query size)**.** *For a 3-CNF with v variables and $O(v)$ clauses, and a specified set of $q \le v$ projection bits, its number of distinct projected models has a relative-η estimate with confidence $1-\zeta$ using $O((q+1)^2\eta^{-2}\log(1/\zeta))$ SAT calls for $0 < \zeta \le 1/2$. Each query has $O(v)$ variables and clauses. Zero counts are detected deterministically, and positive counts always receive positive estimates. The wrapper uses polynomial space.*

The repair lemma. Four places in this paper turn coefficientwise relative estimates into a monotone output without losing accuracy: the iid case, the uncorrectable-erasure probability, monotone latent circuits, and the multiclass tensor case. One lemma covers all of them. For $0 \le \varepsilon \le 1$, write the *two-tail interval* of $t \in [0,1]$ as $[L_\varepsilon(t), U_\varepsilon(t)]$, where

$$L_\varepsilon(t) = \max\{(1-\varepsilon)t, (1+\varepsilon)t - \varepsilon\}, \qquad U_\varepsilon(t) = \min\{(1+\varepsilon)t, (1-\varepsilon)t + \varepsilon\}. \tag{8}$$

A number g lies in this interval exactly when $|g-t| \le \varepsilon t$ and $|g-t| \le \varepsilon(1-t)$. Both endpoints are nondecreasing in t and enclose t.

Lemma 11 (Anchored antitone repair)**.** *Let $(J, \le)$ be a finite partial order, let $f\colon J \to [0,1]$ be antitone, and let $\mathrm{lo}, \mathrm{hi}\colon [0,1] \to \mathbb{R}$ be nondecreasing with $\mathrm{lo}(t) \le t \le \mathrm{hi}(t)$. Let $A \subseteq J$ be a set of anchors with known values $v_a = f_a$ for $a \in A$, and let $h\colon J \to \mathbb{R}$ be arbitrary. Replace h_a by v_a for $a \in A$, clip all other values to $[0,1]$, and define*

$$\ell_j = \min_{i \le j} h_i, \qquad u_j = \max\big(\{0\} \cup \{v_a : a \in A,\ j \le a\}\big), \qquad \widetilde{f}_j = \max\{u_j, \ell_j\}.$$

Then, for every input h, $\widetilde{f}$ is antitone with values in $[0,1]$ and $\widetilde{f}_a = v_a$ for every anchor. If moreover $h_j \in [\mathrm{lo}(f_j), \mathrm{hi}(f_j)]$ for every j, then $\widetilde{f}_j \in [\mathrm{lo}(f_j), \mathrm{hi}(f_j)]$ for every j.

Proof. If $j \le j'$ then $\{i : i \le j\} \subseteq \{i : i \le j'\}$, so $\ell_{j'} \le \ell_j$; and $\{a : j' \le a\} \subseteq \{a : j \le a\}$, so $u_{j'} \le u_j$. A maximum of two antitone functions is antitone. Values lie in $[0,1]$ because the clipped h and the anchors do. At an anchor a, every $v_{a'}$ with $a \le a'$ satisfies $v_{a'} = f_{a'} \le f_a = v_a$, and v_a itself is among them, so $u_a = v_a$; and $\ell_a \le h_a = v_a$. Hence $\widetilde{f}_a = v_a$.

For the interval claim, note first that clipping preserves membership: if $h_i > 1$ then $\mathrm{hi}(f_i) \ge h_i > 1 \ge f_i \ge \mathrm{lo}(f_i)$, and if $h_i < 0$ then $\mathrm{lo}(f_i) \le h_i < 0 \le f_i \le \mathrm{hi}(f_i)$; anchors satisfy $v_a = f_a$ trivially. For the lower bound, every $i \le j$ has $f_i \ge f_j$, so $h_i \ge \mathrm{lo}(f_i) \ge \mathrm{lo}(f_j)$, whence $\ell_j \ge \mathrm{lo}(f_j)$ and $\widetilde{f}_j \ge \mathrm{lo}(f_j)$. For the upper bound, $\ell_j \le h_j \le \mathrm{hi}(f_j)$; every anchor $a \ge j$ has $v_a = f_a \le f_j \le \mathrm{hi}(f_j)$; and $0 \le f_j \le \mathrm{hi}(f_j)$. Hence $u_j \le \mathrm{hi}(f_j)$ and $\widetilde{f}_j \le \mathrm{hi}(f_j)$. □

The lemma makes no use of how h was produced; the shape and anchor properties hold on every run, and accuracy is preserved on the event that the input was accurate. With $(J, \le) = (\{0, \dots, n\}, \le)$, anchors $\{0, n\}$ and the endpoints (8), the rule reads $\widetilde{f}_j = \max\{f_n, \min_{i \le j} h_i\}$, where h_0 and h_n have first been replaced by f_0 and f_n. The replacement is part of the rule: without it the running minimum at $j = n$ could exceed f_n. A nondecreasing target is handled by applying the lemma to $1 - f$. Monotone rearrangement of confidence bands is a classical device in shape-constrained estimation [30, 101]; the point here is that the *multiplicative* two-tail interval is preserved with no additional failure allowance.

Proof of Theorem 1. Apply Proposition 10 to all $2(n+1)$ counts with

$$\eta = \varepsilon/3, \qquad \zeta = \delta/[2(n+1)].$$

With probability at least $1-\delta$, every estimate is relatively accurate. On this single event, set

$$g_j = \frac{\widehat{A}_j}{\widehat{A}_j + \widehat{B}_j}.$$

Both fidelity and infidelity must be controlled after this normalization. For $f = A/(A+B) \in (0,1)$, write $\alpha = \widehat{A}/A,\ \beta = \widehat{B}/B$. Then

$$g - f = \frac{f(1-f)(\alpha-\beta)}{f\alpha + (1-f)\beta}. \tag{9}$$

Since $\alpha, \beta \in [1-\eta, 1+\eta]$, both relative errors are at most

$$\frac{2\eta}{1-\eta} = \frac{2\varepsilon}{3-\varepsilon} \le \varepsilon,$$

that is, $g_j \in [L_\varepsilon(f_j), U_\varepsilon(f_j)]$. Exact zeros handle $f = 0, 1$. The denominator is positive on every run because every Good count receives a positive estimate. The tolerance slack is real: choosing $\eta = \varepsilon/2$ would not work uniformly. With $f = 1/16, \varepsilon = 1/2, \alpha = 5/4, \beta = 3/4$, normalization gives $g = 1/10$, a relative fidelity error of $3/5$.

Next, the image of N_E in N/S is nested under inclusion of E. Hence d_E increases and 2^{-d_E} decreases under inclusion. Coupling uniform sets of consecutive sizes by initial segments of a uniform permutation shows

$$1 = f_0 \ge f_1 \ge \cdots \ge f_n = b, \qquad b = 4^{-k}.$$

Apply Lemma 11 on $\{0, \ldots, n\}$ with anchors $\{0, n\}$, values 1 and b, input $h = g$, and the endpoints (8). Writing $\bar{g}$ for g with the two anchor entries replaced, the rule is

$$\widehat{f}_j = \max\{b, \min_{i \le j} \bar{g}_i\}, \qquad \bar{g}_0 = 1, \quad \bar{g}_n = b, \quad \bar{g}_i = g_i \ \ (0 < i < n). \tag{10}$$

In particular $\widehat{f}_0 = 1$ and $\widehat{f}_n = b$; no clipping is needed because every g_i is a ratio in $[0,1]$. By the lemma, $\widehat{f}$ is nonincreasing with exact endpoints on every run, and on the accuracy event both relative bounds hold for every $\widehat{f}_j$.

Multiply the layerwise inequalities by the nonnegative Bernstein weights and sum to obtain (2) for every real p on that same event. No discretization or union bound over p is needed. The weights sum to one, giving complementarity and range. Monotonicity of the curve follows from

$$\widehat{F}'(p) = n \sum_{j=0}^{n-1} (\widehat{f}_{j+1} - \widehat{f}_j) \binom{n-1}{j} p^j (1-p)^{n-1-j} \le 0.$$

Zero infidelity layers are an initial segment. Their exact SAT zero tests give $g_j = 1$, which the repair retains. If $k > 0$, positive B_j gives $g_j < 1$ and $b < 1$, so the repair invents no extra zero layer. These zero layers are exactly $j < d$, with d the minimum nontrivial logical weight: a weight-d logical support can be extended to an erased set of any size $j \ge d$. For $k = 0$, the output is identically $(1, 0)$.

The resource claims are Proposition 2, proved next. □

Proof of Proposition 2. Here $q = 3n$. The query count follows by multiplying the bound in Proposition 10 by $2(n+1)$; query sizes follow from Lemma 9 together with the $O(q)$ hash variables and

clauses of Lemma 21. Accuracy affects the number of calls, not the query size. The queries have $O_c(n)$ clauses on $O_c(n)$ variables, so their density is bounded by a constant depending on c alone; padding with unused variables lowers the density to any prescribed positive value at constant-factor cost. Sequential processing of the queries gives polynomial space for the wrapper; the total space is the wrapper's plus the SAT algorithm's, which is where the space premise enters, and nowhere else. The estimator's integer outputs have $O(n + \log \varepsilon^{-1})$ bits, and normalization, exact endpoints and running minima preserve this bit bound. Evaluation at rational p has polynomial additional bit cost in its encoding length. The time bound is the number of queries times $\mathrm{poly}(\ell)T(v)$ with $v = O_c(n)$ and $\ell = \mathrm{poly}(L)$, plus polynomial work, and uses only the time premise and the monotonicity of T. □

6 Hardness and comparisons

6.1 Any accurate output determines the distance

Let $k > 0$ and let d be the minimum weight of a nontrivial logical operator, that is, the least $|E|$ with $d_E > 0$.

Proposition 12 (Distance determination)**.** *Let $0 \le \varepsilon < 1$ and let $\widehat{I}(p) = \sum_{j=0}^{n} c_j \binom{n}{j} p^j (1-p)^{n-j}$ with all $c_j \ge 0$ satisfy $|\widehat{I}(p) - I_H(p)| \le \varepsilon I_H(p)$ for every $p \in [0,1]$. Then $c_j = 0$ for $j < d$ and $c_d > 0$. In particular $d = \min\{j : c_j > 0\}$, and the order of vanishing of $\widehat{I}$ at $p = 0$ equals d.*

Proof. By (7), $I_H(p) = Z^{-1} \sum_{j \ge d} B_j p^j (1-p)^{n-j}$ with $B_d > 0$ and $B_j = 0$ for $j < d$, so $I_H(p) = (B_d/Z) p^d (1 + O(p))$ as $p \to 0^+$. The hypothesis gives $(1-\varepsilon) I_H \le \widehat{I} \le (1+\varepsilon) I_H$, hence $\widehat{I}(p)/p^d$ stays between $(1-\varepsilon)B_d/Z > 0$ and $(1+\varepsilon)B_d/Z < \infty$ for small $p > 0$. If some $c_j > 0$ with $j < d$, then $\widehat{I}(p)/p^d \ge c_j \binom{n}{j} (1-p)^{n-j} p^{j-d} \to \infty$, a contradiction. If $c_j = 0$ for all $j \le d$, then $\widehat{I}(p)/p^d \to 0$, a contradiction. Hence $c_j = 0$ for $j < d$ and $c_d > 0$. □

The proof uses the hypothesis only for small $p > 0$. Nonnegativity of the c_j is a convenience: for coefficients of arbitrary sign, the order of vanishing of $\widehat{I}$ at $p = 0$ is the index of its first nonzero Bernstein coefficient, and the two-sided bound forces that order to be d, so the first nonzero coefficient again sits at index d.

Consequently, any algorithm meeting the guarantee of Theorem 1 with some fixed $\varepsilon < 1$ and $\delta < 1/2$ computes the minimum distance with probability greater than $1/2$, by reading off the first positive infidelity coefficient. The minimum distance of a stabilizer code is NP-hard to compute [79], as it is for classical linear codes [10, 119], and hard to approximate [60, 78]. Those proofs reduce from the classical minimum-distance problem. For the purposes of this paper a shorter route is available, which we give in full because it is self-contained and machine-checked: a direct reduction from CNF satisfiability.

A direct reduction from SAT. Let φ be a CNF formula with variables $x_1, \dots, x_V$ and nonempty clauses (a formula with an empty clause is replaced by $x_1 \wedge \neg x_1$), and let K be its number of literal occurrences. For an occurrence o write $v(o)$ for its variable and $\nu_o = 1$ if it is negated, $\nu_o = 0$ otherwise.

Step A: a linear system. Take $2V + 3K$ binary coordinates p_i, n_i $(i \le V)$ and σ_o, w_o, c_o (one

triple for each occurrence o), and the equations over $\mathbb{F}_2$

$$p_i + n_i = 1, \qquad \sum_{o \in C} \sigma_o = 1 \ \text{ (each clause } C), \qquad w_o + \sigma_o + p_{v(o)} = \nu_o, \qquad c_o + p_{v(o)} = 1 + \nu_o,$$

written $Ax = s$. Put $b_o = p_{v(o)} + \nu_o$, the value of the literal. Since $n_i = 1 + p_i$, $w_o = \sigma_o + b_o$ and $c_o = 1 + b_o$, every solution has weight

$$\mathrm{wt}(x) = V + K + 2\,\big|\{o : \sigma_o = 1,\ b_o = 0\}\big| :$$

each pair (p_i, n_i) contributes 1, and each triple $(\sigma_o, \sigma_o + b_o, 1 + b_o)$ contributes 1, or 3 when $\sigma_o = 1$ and $b_o = 0$. If a satisfies φ, set $p_i = a_i$ and $\sigma_o = 1$ on one true literal of each clause; the weight is $V + K$. Conversely, read an assignment off the p_i: a clause that it falsifies has $b_o = 0$ on all its occurrences while $\sum_{o \in C} \sigma_o = 1$, so the weight is at least $V + K + 2$. Hence the minimum weight of a solution is $V + K$ if φ is satisfiable and at least $V + K + 2$ otherwise. The kernel of A is spanned by the explicit vectors $P_i = p_i + n_i + \sum_{v(o)=i}(w_o + c_o)$ and $D_o = \sigma_{o_0} + \sigma_o + w_{o_0} + w_o$, where o_0 is the first occurrence in the clause of o.

Step B: the code. Let $t = V + K$. Add one coordinate $*$ to the $2V + 3K$ coordinates, for the column s of $[A \mid s]$, and repeat every coordinate g on $t + 2$ qubits $(g, 0), \ldots, (g, t + 1)$; there are $n = (2V + 3K + 1)(V + K + 2)$ qubits. The stabilizer is generated by (i) X on copy 0 of the supports of the P_i and D_o; (ii) $X_{(g,0)}X_{(g,l)}$ for all g and l; (iii) for each equation r of $Ax = s$, Z on all copies of its support and, if $s_r = 1$, on all copies of $*$. The generators commute: those of (i) are orthogonal to the rows of A, and those of (ii) meet each Z-generator in zero or two qubits. The check matrix has $O((V + K)^3)$ nonzero entries.

Proposition 13 (Distance from satisfiability)**.** *The code of φ has exactly one logical qubit. If φ is satisfiable its distance is $t + 1$; otherwise it is at least $t + 2$. Equivalently, an uncorrectable erased set of size at most $t + 1$ exists if and only if φ is satisfiable.*

Proof. If $Ax = s$, the operator X on $\{(g, 0) : x_g = 1\} \cup \{(*, 0)\}$ commutes with (iii), and it is not in S: the X-parity on the copies of $*$ vanishes on every generator and equals one on it. Its weight is $\mathrm{wt}(x) + 1$, which is $t + 1$ for the solution of a satisfying assignment. A solution exists because no clause is empty, so $k \geq 1$. Conversely, let $X^a Z^b \in N \setminus S$. The code is CSS, so X^a and Z^b are in N and one of them is not in S. If $Z^b \notin S$, then $b \neq 0$, and commutation with (ii) makes b constant on the copies of each coordinate, so the weight is at least $t + 2$. If $X^a \notin S$, let $y_g = \sum_l a_{(g,l)}$. Commutation with (iii) gives $Ay = y_* s$ on the original coordinates. If $y_* = 0$, then $y \in \ker A$, and X^a is the product of the lift of y to copy 0, a product of generators (i), with the generators (ii) at the positions where a is nonzero; so $X^a \in S$, a contradiction. Hence $y_* = 1$ and y solves $Ax = s$. Every coordinate with $y_g = 1$, and $*$, carries a nonzero copy, so the weight is at least $\mathrm{wt}(y) + 1 \geq t + 1$, and a weight of at most $t + 1$ forces $\mathrm{wt}(y) \leq t$, which by Step A makes φ satisfiable. Finally $k = 1$: the argument above shows that an X-type element of N with $y_* = 0$ is in S, so the X-type elements of N form two classes modulo S; and a Z-type element of N, after multiplication by Z on all copies of $*$ if it anticommutes with the X-logical operator, commutes with all of N and therefore lies in $S = N^{\perp}$. Hence N is generated by S and two operators, and $\dim N - \dim S = 2$. □

The code and t are computed from φ in polynomial time, so by the Cook–Levin theorem [35] it is NP-hard to decide whether the distance of a CSS code with one logical qubit is at most a given number, and equally whether an uncorrectable erased set of a given size exists. The reduction combines two known devices. Placing the kernel of A in the stabilizer, so that the lightest X-logical

operator is a minimum-weight coset element [10], is the embedding by which Chen and Freedman show that a shortest nonbounding $\mathbb{F}_2$-cycle is NP-hard to find even when the homology has rank one [29], and it appears in the hardness of degenerate decoding [72]; protecting the other logical type by repetition is standard. To our knowledge this is the first proof that the minimum distance of a stabilizer code is NP-hard that does not pass through the hardness of the classical minimum-distance problem [119], on which the proofs of [60, 79] rest, and the first that gives hardness for codes with a single logical qubit. The construction, the proposition and its composition with Proposition 12 and Proposition 15 below are formalized (Section 9); the polynomial running time of the reduction and the Cook–Levin theorem are not. The codes produced are not sparse, but sparsity is restored by padding.

Corollary 14 (Hardness for sparse presentations)**.** *Fix $c > 1$, $\varepsilon < 1$ and $\delta < 1/2$. If a randomized polynomial-time algorithm meets the guarantee (2) of Theorem 1 on all check matrices with at most cn nonzero entries, then* NP $\subseteq$ BPP*. The same holds for the relative approximation of I_H alone, with the output in Bernstein form as in Proposition 12; Corollary 16 removes the assumption on the form of the output.*

Proof. Let H be an arbitrary check matrix on n qubits with w nonzero entries. Append n' new qubits and, for each of them, one new row whose only nonzero entry is a Z on that qubit. The new matrix H' has $n + n'$ qubits and $w + n'$ nonzero entries, so it satisfies the sparsity bound as soon as $n' \geq (w - cn)/(c - 1)$, which is polynomial in the size of H. Its row space is $S' = S \oplus \langle Z_i \rangle$, with normalizer $N' = N \oplus \langle Z_i \rangle$, the sums ranging over the new qubits i. For $E \subseteq [n + n']$ the two groups N'_E and S'_E each gain the same factor, generated by the Z_i with $i \in E$ new, so $d'_E = d_{E \cap [n]}$. Hence k and the minimum distance are unchanged and $F_{H'}(p) = \mathbb{E}_E\, 2^{-d_{E\cap[n]}} = F_H(p)$ identically in p. An algorithm as in the statement, run on H', returns with probability greater than $1/2$ an output from which Proposition 12 reads off the distance of the code of H. Applied to the codes of Proposition 13, this decides satisfiability in randomized polynomial time with two-sided bounded error. □

The padding multiplies the number of qubits by a factor of order w/n, which is not bounded, so Corollary 14 gives no fine-grained lower bound: it does not show that the curve of an n-qubit sparse code needs time $T(\Omega(n))$. It also says nothing about presentations in which every row and every column has bounded weight, where padding is not available. Both questions are open (Section 11). For $c \leq 1$ either some qubit is touched by no check, or every qubit is touched by exactly one nonzero entry and the code is a product of single-check blocks. In both cases $d \leq 1$ whenever $k > 0$, so the distance route gives nothing; we do not treat that regime. Conversely, sparsity enters Theorem 1 only through the query size: by Lemma 9 the same algorithm applies to an arbitrary check matrix with $T(C_c n)$ replaced by $T(C(n + w))$ for an absolute constant C.

Hardness at a single point. Proposition 12 uses the whole curve and its Bernstein form. Neither is needed: one number, at one explicitly given point, already determines the distance, even when it is known only up to an exponentially large factor.

Proposition 15 (A point estimate determines the distance)**.** *Let $k > 0$ and let d be the minimum distance. For every $p \in [0, 1]$,*

$$\tfrac{1}{2}\, p^d \leq I_H(p) \leq \binom{n}{d} p^d, \qquad p^d \leq U_H(p) \leq \binom{n}{d} p^d. \tag{11}$$

Consequently, let $R \geq 1$ and $0 < p \leq 1$ satisfy $2^{n+1}R^2 p < 1$, and let $\widehat{\imath}$ be any real number with $I_H(p)/R \leq \widehat{\imath} \leq R\, I_H(p)$. Then d is the unique natural number d' such that

$$\frac{p^{d'}}{2R} \leq \widehat{\imath} \leq 2^n R\, p^{d'}.$$

The same holds for U_H in place of I_H, with the test $p^{d'}/R \leq \widehat{u} \leq 2^n R\, p^{d'}$.

Proof. Let E_0 be an uncorrectable set of size d. Every $E \supseteq E_0$ is uncorrectable, so it contributes one to U_H and at least $1/2$ to I_H, and $\mathbb{P}(E \supseteq E_0) = p^d$. Conversely an uncorrectable set has at least d elements and therefore contains some d-subset, so its indicator is at most the number of d-subsets it contains; taking expectations gives $U_H(p) \leq \binom{n}{d} p^d$, and $I_H \leq U_H$. This proves (11). With $\binom{n}{d} \leq 2^n$, the true d passes the test. If some $d' < d$ also passed, then $p^{d'}/(2R) \leq \widehat{\imath} \leq R\, 2^n p^d \leq R\, 2^n p^{d'+1}$, so $1 \leq 2^{n+1}R^2 p$; if some $d' > d$ passed, then $p^d/(2R) \leq \widehat{\imath} \leq 2^n R\, p^{d'} \leq 2^n R\, p^{d+1}$, with the same contradiction. □

For example, at $p = 2^{-(n+2r+2)}$, a point with $O(n+r)$ bits, any estimate of $I_H(p)$ within a multiplicative factor 2^r determines d; a relative error $\varepsilon < 1$ corresponds to the constant $R = 1/(1-\varepsilon)$. Combined with the padding of Corollary 14, which changes neither I_H nor d, this gives hardness with no assumption on the form of the output:

Corollary 16 (Pointwise hardness)**.** *Fix $c > 1$ and a polynomial $h\colon \mathbb{N} \to \mathbb{N}$, and put $p_n = 2^{-(n+2h(n)+2)}$. Suppose a randomized polynomial-time algorithm, given a check matrix H on n qubits with at most cn nonzero entries, outputs with probability at least $2/3$ a number $\widehat{\imath}$ with $2^{-h(n)} I_H(p_n) \leq \widehat{\imath} \leq 2^{h(n)} I_H(p_n)$. Then* NP $\subseteq$ BPP*. The same holds for U_H.*

Proof. Given an arbitrary check matrix with $k > 0$, pad it as in Corollary 14 to a matrix H' on n' qubits that meets the sparsity bound; n' is polynomial, and H' has the same I_H, U_H and d. Run the algorithm on H' and apply Proposition 15 to the code of H' with $R = 2^{h(n')}$ and $p = p_{n'}$, for which $2^{n'+1}R^2 p = 1/2$. The point $p_{n'}$ has $O(n' + h(n'))$ bits, and the test of the proposition, tried for $d' = 1, \ldots, n'$, is polynomial-time arithmetic on the output. With probability at least $2/3$ it returns the distance. Applied to the codes of Proposition 13, which have $k = 1$, and repeated with a majority vote, this decides satisfiability with two-sided bounded error. For a general input the case $k = 0$ is recognized by Gaussian elimination. □

Thus the small-p tail is hard to approximate even within an exponential factor, for sparse inputs, at a single point. This is the regime in which Theorem 1 gives relative error ε.

What a SAT engine gives exactly. The same certificate shows what does *not* need approximation. For a threshold t, the statement $d \leq t$ is existential,

$$d \leq t \iff \exists E,\ |E| \leq t,\ \exists x \in N_E,\ \exists z \in N :\ \langle x, z \rangle_s = 1,$$

and by Lemma 9 it is the satisfiability of one 3-CNF with $O(n + \mathrm{nnz}(H))$ variables, with no projection and no hashing. A decision algorithm as in Theorem 1 therefore computes the distance exactly and deterministically, by binary search over t, in time $\mathrm{poly}(L)\, T(O(n + \mathrm{nnz}(H)))$; reductions of the distance to SAT are standard, and one has been formally verified [46]. If the algorithm also counts models, as the verified engine of [95] does, the ordinary weight enumerators follow exactly as well. The number of $x \in N$ of weight t is the model count of a 3-CNF on the $2n$ bits of x in which every

other variable is a function of x: the parity checks for $x \in N$, one OR gate per qubit for its support, and a counter for the weight; no set E appears. The elements of S are counted in the same way through $x = uH$, with the constant multiplicity 2^{m-s} for the free choice of u. These are exactly the quantities that Section 6.3 shows do not determine the curve. The curve weights each erased set by 2^{-d_E}, and it is this weighting that forces the projected, approximate route.

Note that the fidelity tail carries no distance-type obstruction near $p = 0$: $\widehat{F} = 1$ is a valid relative approximation of F_H there. The obstructions above live in the infidelity tail, which is also the quantity of practical interest.

Exact evaluation. Corollary 16 with $h \equiv 0$ shows that exact evaluation of I_H at the input-dependent point $p_n = 2^{-(n+2)}$ is NP-hard for sparse presentations. We prove no hardness for exact evaluation at a fixed constant p, and no #P-hardness. For a classical binary linear code the corresponding quantity is the reliability polynomial of a binary matroid, whose exact evaluation is #P-hard [19, 32, 73, 99]; the symplectic-matroid viewpoint is in [103]. The quantum curve is not such a polynomial in disguise: the correctable sets of the four-qubit CSS code with checks $ZZIZ$, $XIXX$, $XXXI$ violate the independent-set exchange axiom, and the weight-two fidelity polynomial of the three-qubit repetition code with five appended stabilized qubits is not log-concave, so the matroid-reliability approximation scheme of [3] does not apply directly.

6.2 Uncorrectable-erasure probability

Changing the projection gives a direct approximation of conventional uncorrectable-erasure probability, beyond the factor-two comparison of Section 2.

Corollary 17 (Uncorrectable-erasure probability)**.** *Under the hypotheses of Theorem 1, the same runtime and space bounds suffice to return a rational polynomial $\widehat{U}(p)$ such that, with probability at least $1 - \delta$,*

$$|\widehat{U}(p) - U_H(p)| \le \varepsilon U_H(p) \quad \textit{simultaneously for every } p \in [0,1].$$

The output can always be chosen nondecreasing, valued in $[0,1]$, with endpoints $\widehat{U}(0) = 0$ and $\widehat{U}(1) = \mathbf{1}_{k>0}$, and exact zero Bernstein coefficients. No relative guarantee for the probability $1 - U_H(p)$ of perfect correctability is asserted.

Proof. Since $N^{\perp} = S$,

$$d_E > 0 \quad \Longleftrightarrow \quad \exists x \in N \cap V_E,\ z \in N :\ \langle x, z\rangle_s = 1.$$

Indeed, a vector $x \in N_E \setminus S$ fails to annihilate N, and a pairing-one witness cannot lie in S. Use the Bad circuit of Lemma 9, but project only onto the n bits of E, existentially quantifying z as well as x and the auxiliaries. Its query size is unchanged. The projected count is now

$$C_j = \#\{E : |E| = j,\ d_E > 0\}, \qquad U_H(p) = \sum_{j=0}^{n} C_j p^j (1-p)^{n-j}.$$

Every failed set is counted once, regardless of d_E. Apply Proposition 10 with relative tolerance ε and failure probability $\delta/(n+1)$ to each coefficient; $n = 0$ is immediate. A union bound and positivity give the simultaneous relative guarantee, including exact zeros.

For the shape statement, write $u_j = C_j/\binom{n}{j}$. Inclusion monotonicity of $d_E > 0$ and the uniform-permutation coupling give $u_0 \le \cdots \le u_n$, with $u_0 = 0$ and $u_n = \mathbf{1}_{k>0}$. Apply Lemma 11 to

the antitone target $1-u$ with anchors $\{0,n\}$ and the one-sided endpoints $\mathrm{lo}(t)=(1+\varepsilon)t-\varepsilon$, $\mathrm{hi}(t)=(1-\varepsilon)t+\varepsilon$, which encode $(1-\varepsilon)u\le\widehat{u}\le(1+\varepsilon)u$. Translating back, the rule is

$$\widehat{u}_j=\min\{u_n,\max_{i\le j}\bar{c}_i\},\qquad \bar{c}_0=0,\quad \bar{c}_n=u_n,\quad \bar{c}_i=\min\Big\{1,\widehat{C}_i\Big/\binom{n}{i}\Big\}\ \ (0<i<n):$$

clip, set the two endpoints exactly, and then take running maxima. The lemma gives the claimed shape on every run and preserves relative accuracy on the accuracy event. The zero coefficients are exact on every run: the zero tests are deterministic, a positive count receives a positive estimate, and by inclusion monotonicity the layers with $u_j=0$ form a prefix, on which the running maximum is zero. Set $\widehat{U}(p)=\sum_j\widehat{u}_j\binom{n}{j}p^j(1-p)^{n-j}$. The oracle, bit and space budgets are those of Proposition 2. □

Remark 18 (A machine-checked route through repeated codes). Corollary 17 also follows from the latent-circuit results of Section 8, and in that form it is machine-checked (Section 9). Take $m\ge1$ copies of the code (in this remark m is the number of copies, not the number of rows of H), on nm qubits, and erase the same set E in every copy. This is a latent model with n latent bits and a duplication circuit, which is monotone and has no gates. The exponent of the repeated set is $m\,d_E$, so the infidelity of the repeated code is $I^{(m)}(p)=\sum_E p^{|E|}(1-p)^{n-|E|}(1-2^{-md_E})$ and

$$(1-2^{-m})\,U_H(p)\le I^{(m)}(p)\le U_H(p)\qquad(0\le p\le1).$$

With $2^{-m}\le\varepsilon/2$, an $(\varepsilon/2)$-relative approximation of $I^{(m)}$ is therefore an ε-relative approximation of U_H. The output of Theorem 20 with one class is nondecreasing, and its j-th coefficient vanishes exactly when no erased set of size j is uncorrectable. The price is that the SAT queries have $O(m(n+\mathrm{nnz}(H)))$ variables with $m=\lceil\log_2(2/\varepsilon)\rceil$, and that the value at $p=1$ is $1-4^{-mk}$ in place of $\mathbf{1}_{k>0}$.

For a single unencoded qubit, $U_H(p)=p$ whereas $I_H(p)=3p/4$; when $k=0$, both curves vanish identically.

6.3 Ordinary weight enumerators do not determine the curve

The quantum weight enumerators of Shor and Laflamme [107] and Rains [100] determine many properties of a code, and tensor-network enumerators [21, 22] compute exact logical error rates for structured codes. The ordinary enumerators do not determine the erasure curve. Consider two CSS $[[3,1,1]]$ codes. Code R has checks ZZI, IZZ; code B has checks XXI, ZZI, a Bell pair tensored with a free logical qubit. Both have total binary incidence four. Their ordinary Pauli-weight enumerators agree:

$$W_S(t)=1+3t^2,\quad W_N(t)=1+3t+3t^2+9t^3,\quad W_{N\setminus S}(t)=3t+9t^3.$$

For R, the normalizer is the union of the eight Z/I strings and the eight X/Y strings. For B, it is the product of the four Bell stabilizers with the four Paulis on the free qubit. Yet their layer sums are different:

j	0	1	2	3
$\sum_{\lvert E\rvert=j}F_{\mathrm{opt},R}(E)$	1	3/2	3/2	1/4
$\sum_{\lvert E\rvert=j}F_{\mathrm{opt},B}(E)$	1	9/4	3/2	1/4

For R, any one- or two-qubit erasure leaves one ambiguous logical Pauli bit. For B, only erasure of the free qubit matters. It follows that

$$F_R(p)=(1-p)^3+\tfrac{3}{2}p(1-p)^2+\tfrac{3}{2}p^2(1-p)+\tfrac{1}{4}p^3,$$
$$F_B(p)=1-\tfrac{3}{4}p,\qquad F_B(p)-F_R(p)=\tfrac{3}{4}p(1-p)^2.$$

Thus ordinary stabilizer and normalizer enumerators, even together, do not determine reliability. Support-sensitive enumerators [21, 104] and the two-parameter subcode enumerators of [4] retain more information; no priority is claimed for this elementary example.

6.4 A restricted independent-sampling comparison

Let $k > 0$ and let d be the minimum distance. For $0 < p < 1$,

$$\tfrac{1}{2}p^d \le I_H(p) \le \mathbb{P}\{\mathrm{Binomial}(n,p) \ge d\}, \tag{12}$$

by Proposition 15 and because fewer than d erased qubits give zero infidelity. For a family with $d \ge \alpha n$ and fixed $0 < p < \alpha < 1$, the Chernoff bound gives $I_H(p) \le e^{-n\mathcal{D}(\alpha\|p)}$ with $\mathcal{D}(a\|p) = a\log\frac{a}{p} + (1-a)\log\frac{1-a}{1-p} > 0$, so $I_H(p) = e^{-\Theta(n)}$; positive-rate, linear-distance qLDPC families with linear incidence exist [96, Theorem 2]. An empirical mean of M independent failure indicators with mean $\theta = I_H(p)$ is zero with probability $(1-\theta)^M$, so any relative tolerance below one at confidence $1-\delta$ needs $M \ge \log(1/\delta)/(2\theta)$ samples when $\theta \le 1/2$; averaging the exact values $1 - 2^{-d_E}$ needs $M \ge \log(1/\delta)/(4\theta)$ once $U_H(p) \le 1/2$. For these families direct sampling therefore needs $e^{\Omega(n)}$ trials, whereas the runtime (3) is subexponential. The comparison is with direct sampling only; it does not constrain importance sampling and splitting methods [11, 16, 33] or other algorithms given the code description. The statements of this subsection are machine-checked (Table 3).

7 Computed examples

The theorems above are asymptotic. This section runs the same certificates with off-the-shelf tools on real codes. It has three purposes: to test the formulas and the algorithm end to end against exhaustive enumeration; to show what the two-tail curve looks like and how far the leading-order approximation $I_H(p) \approx c\,p^d$ can be trusted; and to record how far present tools reach. The codes are the five-qubit code, the Steane and Shor codes, a $[[15, 7, 3]]$ Hamming CSS code, rotated surface codes [13], toric codes [84] and bivariate bicycle (BB) codes [14]. The programs and all outputs are in the artifact (`experiments/erasure`).

7.1 The algorithm of Theorem 1 with an off-the-shelf counter

For each code we generate the two 3-CNFs of Lemma 9 explicitly: XOR chains and AND gates with full-equivalence Tseitin clauses, and a cardinality constraint for $|E| = j$; every clause has width at most three. As a first check we count the projected models of these formulas exactly, by blocking-clause enumeration, and compare with exhaustive enumeration over all erased sets, in every layer: the (E, z)-projection of both formulas against A_j and B_j of (6) on the five-qubit and Steane codes; the projection through a logical basis described below against $A_j/|S|$ and $B_j/|S|$ on the Shor code and the nine-qubit surface code (the runs reported below use the full projection for these two codes); and the E-projection of the `Bad` formula against the number of uncorrectable sets on the Hamming code. All counts agree. The exhaustive tables themselves do not rest on our numerical code alone: for the seven codes of at most 25 qubits they were recomputed by a Lean function that is proved equal to the definition of f_j (`ExactTable.fRat_spec`) and agree entry by entry as exact fractions; the 32-qubit table used below is computed by our Python code only.

We then run the algorithm in the proof of Theorem 1, with four substitutions. The projected counts are estimated by ApproxMC [27, 28, 109], a hashing-based approximate model counter built

on a SAT solver with XOR constraints, with the same (η, ζ) guarantee per count; it hashes with dense random XOR constraints, where Lemma 21 uses a linear-size circuit. The SAT solver is CryptoMiniSat [110], where Corollary 3 uses the subexponential engine. The cardinality constraint is a totalizer from PySAT [67], which is not of linear size, where Lemma 9 uses a carry-save counter. And for all but the four smallest codes z is projected through a basis $L_1, \ldots, L_{2k}$ of N modulo S, that is, onto (E, a) with $z = \sum_i a_i L_i$. This is legitimate because Good and Bad depend on z only through its class modulo S: both counts are divided by $|S|$ and the ratio g_j is unchanged. It shortens the projection from $3n$ to $n + 2k$ bits at the price of needing a logical basis, which the certificates of Proposition 8 avoid. We take $\varepsilon = 0.3$ and $\delta = 0.1$, hence tolerance $\eta = 0.1$ and failure probability $\delta/(2(n+1))$ per count, test every count for zero with one SAT call, normalize, and apply the repair (10).

Code	n	proj.	vars	clauses	$\log_2$ count	err. I	err. F	time
$[[5, 1, 3]]$	5	15	92	227	10	exact	exact	0 s
Steane $[[7, 1, 3]]$	7	21	138	349	13	exact	exact	17 s
Shor $[[9, 1, 3]]$	9	27	170	417	17	exact	exact	35 s
rotated surface $[[9, 1, 3]]$	9	27	170	417	17	exact	exact	44 s
Hamming CSS $[[15, 7, 3]]$	15	29	298	831	27	0.00165	0.00313	14 min
toric $[[18, 2, 3]]$	18	22	287	811	20	0.00279	0.00132	6 min
rotated surface $[[25, 1, 5]]$	25	27	388	1203	24	0.00558	7.8×10^{-4}	23 min
toric $[[32, 2, 4]]$	32	36	547	1903	33	0.00239	0.00376	5.9 h

Table 1: The algorithm of Theorem 1 with ApproxMC and CryptoMiniSat, $\varepsilon = 0.3$, $\delta = 0.1$. "proj." is the number of projection bits; "vars" and "clauses" are for the larger of the two 3-CNFs at the middle layer; "$\log_2$ count" is the bit length of the largest projected count. The errors are the largest relative errors of $\widehat{I}$ and $\widehat{F}$ over a logarithmic grid of 400 values of p from 10^{-14} to 0.999, against exhaustive enumeration of all 2^n erased sets. "Time" is the wall-clock time of the counts, each a single-threaded process, summed over all counts.

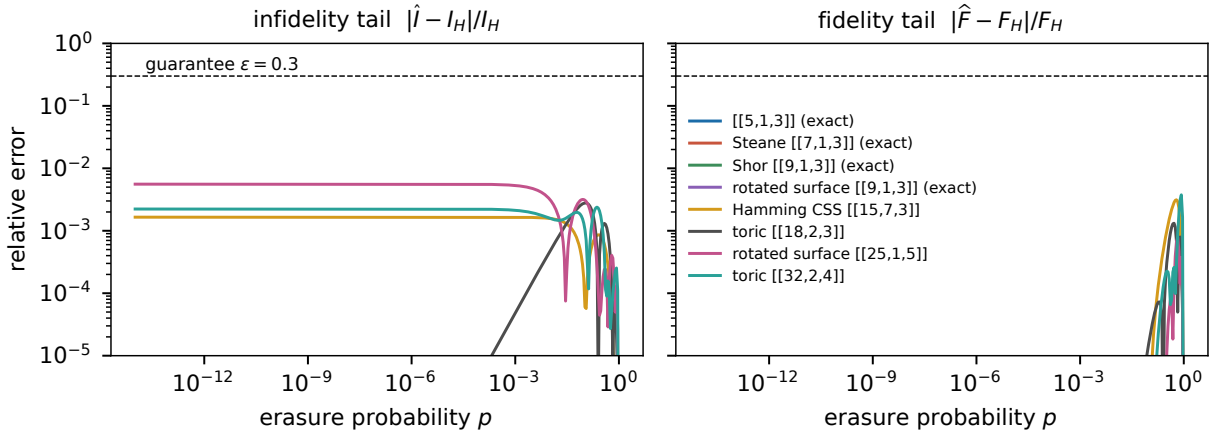


Figure 2: Relative error of the output of the algorithm of Theorem 1, run with off-the-shelf components, against the exact curves, in both tails. The dashed line is the guaranteed ε. Outputs that are exact do not appear on the logarithmic scale.

Table 1 and Figure 2 give the result. The guarantee holds in both tails down to $p = 10^{-14}$, with

observed errors at most 0.56%, a factor of more than fifty below ε; the zero infidelity coefficients are reproduced exactly, so the output has the right order of vanishing at $p = 0$; and on the four smallest codes the whole output is exact. The cost is another matter. The seven codes of at most 25 qubits took between less than a second and 23 minutes of single-thread time each. The 32-qubit toric code took 5.9 hours for its 66 counts, of which 2.7 went to three counts of the Bad formula, at $j = 9, 10, 11$. At a looser tolerance, single counts on the 49- and 50-qubit codes took up to 161 seconds, against 39 seconds at 32 qubits, and no count on the 72-qubit BB code completed within ten minutes; we did not run the full algorithm beyond 32 qubits. This is consistent with the purpose of Theorem 1, which is a statement about exponents and not a recipe.

7.2 A practical hybrid on the same certificate

For larger codes we use the SAT oracle only where sampling cannot work. Call an erased set *minimal uncorrectable* if $d_E > 0$ and every proper subset is correctable. By inclusion monotonicity, a set is uncorrectable exactly when it contains a minimal uncorrectable set.

Low layers. A SAT solver enumerates *all* minimal uncorrectable sets of size $1, 2, \dots, w$, level by level (in this section w denotes this size bound and not an incidence count). The formula is the E-projection of the Bad certificate in the form used for Corollary 17, except that the witness $z \in N$ is replaced by a fixed logical basis: there is $x \in N$ supported on E that anticommutes with some L_i. At level w' we ask for such a set of size at most w' that contains none of the sets found so far, and add each answer as a blocking clause, until the instance is unsatisfiable. Every answer E at level w' is minimal of size exactly w': it contains a minimal uncorrectable set M, and if $|M| < w'$ then M was found at an earlier, completed level and is blocked, so $|M| = w' \geq |E|$ and $E = M$. The final unsatisfiable answer says that the list is complete up to size w', and the first nonempty level is the distance; these answers are checked independently below. For the toric and BB codes we use their translation symmetry: every generating permutation is first checked to preserve the row space of H, so that the group they generate does, each minimal set is added together with its whole orbit, and the solver is asked only for sets through one representative qubit of each orbit of qubits. This loses nothing: a permutation that preserves S preserves d_E and maps minimal uncorrectable sets to minimal uncorrectable sets, and every nonempty set has a translate through a representative. For $j \leq w$ the layer sum $\sum_{|E|=j}(1 - 2^{-d_E})$ is supported on the union of the supersets of the listed sets. We estimate it by the union estimator of Karp, Luby and Madras [82]: draw a listed set M with probability proportional to its number of supersets of size j, draw a uniform superset $E \supseteq M$ of size j, compute d_E by Gaussian elimination, and weight $1 - 2^{-d_E}$ by the reciprocal of the number $\nu(E)$ of listed sets contained in E. The estimator is unbiased, and its relative variance is at most $\max_E \nu(E) \cdot g_{\max}/g_{\min}$, where g ranges over the weights of uncorrectable sets (here $g_{\max}/g_{\min} \leq 2$); it does not grow as uncorrectable sets become rare. Both facts are machine-checked (`UnionEstimator.mean_sample`, `UnionEstimator.second_moment_le`). We draw 4×10^5 union samples per layer (2×10^5 for the 128-qubit code). The layer $j = d$ is not sampled at all: its uncorrectable sets are exactly the listed sets of size d, so the leading coefficient $\binom{n}{d}(1 - f_d) = \sum_{|M|=d}(1 - 2^{-d_M})$ is a finite sum over the list.

High layers. The exponents of complementary sets satisfy $d_E + d_{E^c} = 2k$, so $f_j = 4^{-k}\,\mathbb{E}_{|E'|=n-j}[2^{d_{E'}}]$, and $4^k f_j - 1$ is again supported on uncorrectable sets, now of size $n - j$. For $n - j \leq w$ the same estimator applies with weight $2^{d_{E'}} - 1$. Here $g_{\max}/g_{\min}$ can be as large as $4^k - 1$, but the estimated quantity $x_j = 4^k f_j - 1$ enters $f_j = 4^{-k}(1 + x_j)$ only through $1 + x_j$, so a relative error in x_j reaches f_j multiplied by $x_j/(1 + x_j)$.

Bulk. For the remaining layers we sample 2×10^7 erased sets of size j uniformly and compute

d_E for each. This suffices when the relative standard errors of both f_j and $1 - f_j$ are at most 2%.

Layers just above w. Where it does not suffice, uncorrectable sets are too rare to sample and too many to list. We split the layer sum into the uncorrectable sets that contain a listed minimal set and the remainder, those whose minimal subsets all have size above w. The first part is estimated by the union estimator, whose identity holds for every j. The remainder is sampled uniformly, and it is a small part of the sum for j close to w, so that an absolute error which would be useless for the whole layer is enough for it. With M samples we take its standard error to be at least $3/M$, which is the 95% upper bound on the remainder when no remainder event is seen. The first sample is sized so that this floor meets the tolerance, and it is doubled until the combined relative standard error is at most 2%.

The final coefficients are obtained from these estimates by the repair (10), which by Lemma 11 preserves every two-tail relative interval that the estimates satisfy coefficientwise. Because both curves are positive combinations of the coefficients, the relative error of $\widehat{I}(p)$ and $\widehat{F}(p)$ at every p is at most the largest coefficientwise relative error. The result is a statistical statement, with standard errors from the central limit theorem for the unrepaired estimates, one coefficient at a time and with no claim of simultaneous coverage, and not the worst-case guarantee of Theorem 1. The duality, the union identity and its split form, the minimal-set characterization, the symmetry reduction and the passage from coefficients to curves are machine-checked (Table 3); the sampling errors are not.

Proof-checked completeness. The estimator needs three facts about a list: every listed set is uncorrectable, none is smaller than d, and every uncorrectable set of size at most w contains a listed set. The first two are checked by Gaussian elimination. For the third we rely neither on the solver that produced the list nor on a program that writes its input. The statement "there is an uncorrectable set of size at most w' that contains no listed set" is a CNF formula defined in Lean (`certFormula`), with a proof that its unsatisfiability gives the completeness of the list through w' (`certFormula_unsat`). A Lean program writes the formula, CaDiCaL [12] refutes it and emits a proof in the LRAT format [36], and the same program checks that proof with the LRAT checker of the Lean 4 core library, whose soundness is a theorem of that library (`LRAT.check_sound`). The symmetry reduction, and a case split of the instances whose proofs are too large to check in memory, are covered by the theorem. A second Lean program, whose checks are proved sound, verifies the side conditions, and one theorem (`ListCertificate.symmetric_certificate`) then gives the completeness of the list through w', the distance, the identification of the uncorrectable sets of size d with the listed ones, and the exact leading layer sum. For nine of the ten codes every level up to w is verified in this way. For the $[[128, 2, 8]]$ code the levels through size 12 of 15 are, which covers its distance, its sets of size d and its leading layer sum; sizes 13 to 15 were beyond our computing budget with this formula and are certified with an independently written encoding, whose refutations are proof-checked in the same way but whose meaning is tested against exhaustive enumeration on small codes and not proved. What remains trusted is the parser of the certificate file, the compiled code of the Lean functions and the matching, by SHA-256 hash, of the instances whose proofs were checked with those used in the final combination.

We checked the estimator against exhaustive enumeration on the $[[18, 2, 3]]$, $[[25, 1, 5]]$ and $[[32, 2, 4]]$ codes, the last over all 2^{32} erased sets. The repaired curves of the runs reported below differ from the exact ones by at most 1.4×10^{-4} in relative terms, in either tail, over the same grid of p as in Table 1. In separate validation runs with 4×10^6 uniform samples per bulk layer, no layer estimate of any kind deviates from the exact coefficient by more than 2.4 reported standard errors. These runs never need the split estimator. To exercise it we reran the $[[25, 1, 5]]$ and $[[32, 2, 4]]$ codes with $w = d$ and only 2×10^4 uniform samples per layer, which forces the split on three and five layers; all sixteen split estimates of $1 - f_j$ and of the uncorrectable fraction are within two standard

errors of the exact values.

Code	minimal uncorrectable sets, size: count	w	s.e.	$I_H(10^{-3})$	$I_H(10^{-2})$	$I_H(0.1)$	$p_{1/2}$
toric $[[18,2,3]]$	3: 12, 5: 72, 6: 78	6	0.02%	6.0×10^{-9}	6.0×10^{-6}	0.00626	0.443
surface $[[25,1,5]]$	5: 104, 7: 150	8	0.05%	5.2×10^{-14}	5.2×10^{-9}	4.7×10^{-4}	0.538
toric $[[32,2,4]]$	4: 16, 6: 192, 8: 1064, 10: 3136	10	0.04%	8.0×10^{-12}	8.0×10^{-8}	8.9×10^{-4}	0.456
surface $[[49,1,7]]$	7: 592, 9: 1954, 11: 4548, 13: 8076	13	0.2%	3.0×10^{-19}	2.9×10^{-12}	2.7×10^{-5}	0.530
toric $[[50,2,5]]$	5: 20, 7: 400, 9: 2200, 10: 2410, 11: 8000	11	0.2%	1.0×10^{-14}	1.0×10^{-9}	1.2×10^{-4}	0.463
BB $[[72,12,6]]$	6: 156, 8: 1980, 10: 26496	11	0.7%	8.1×10^{-17}	8.1×10^{-11}	9.2×10^{-5}	0.373
toric $[[72,2,6]]$	6: 24, 8: 720, 10: 5760, 12: 37572	13	0.9%	1.2×10^{-17}	1.2×10^{-11}	1.6×10^{-5}	0.467
surface $[[81,1,9]]$	9: 3112, 11: 16542, 13: 61350	14	2.0%	1.6×10^{-24}	1.5×10^{-15}	1.4×10^{-6}	0.525
BB $[[90,8,10]]$	10: 1116, 12: 20580	13	1.9%	5.6×10^{-28}	5.6×10^{-18}	6.7×10^{-8}	0.431
toric $[[128,2,8]]$	8: 32, 10: 1792, 12: 25984, 14: 195328	15	2.0%	1.6×10^{-23}	1.6×10^{-15}	2.6×10^{-7}	0.473

Table 2: The hybrid estimator. "Minimal uncorrectable sets" lists the number found at each size up to the level w through which the SAT enumeration is complete; sizes not listed have none, and the first size is the distance d. "s.e." is the largest relative standard error over all coefficients of either tail, which bounds the relative standard error of both curves at every p; in the split layers it includes the $3/M$ convention described in the text. $p_{1/2}$ is the erasure probability at which $I_H(p) = 1/2$.

Table 2 and Figure 3 give the results for ten codes of 18 to 128 qubits. The SAT enumeration took at most three minutes for each of the eight codes of at most 81 qubits, 43 minutes for the $[[90,8,10]]$ code and 13 minutes for the $[[128,2,8]]$ toric code, on up to twelve solver threads; the sampling took between 23 seconds and about 110 minutes per code on 8 to 24 threads. The split estimator was needed on one layer of the $[[81,1,9]]$ code, on eight layers ($j = 14,\dots,21$) of the $[[90,8,10]]$ code and on eleven ($j = 16,\dots,26$) of the $[[128,2,8]]$ code, with up to 1.6×10^9 samples in a layer; the sampled remainder was at most 3% of a layer sum. Every coefficient of every code has a relative standard error of at most 2% on both scales; in the fourteen of the twenty split layers where no remainder event was seen, the error assigned to the remainder is the 95% bound $3/M$. Three observations follow.

Small p. At $p = 10^{-3}$ the logical error rate ranges from 6×10^{-9} for the smallest code to 5.6×10^{-28} for the $[[90,8,10]]$ code, and at $p = 10^{-2}$ it is below 10^{-10} for six of the ten codes. No direct Monte Carlo simulation reaches these values; here they come with the same relative error as the rest of the curve, because the low layers are computed from the enumerated sets. The first nonempty level reproduces the distance of every code, and the leading coefficient $c = \binom{n}{d}(1 - f_d)$ is exact, with the completeness of the list of minimum-size sets proof-checked as described above: $c = 2L$ for the $L\times L$ toric code, whose $4L$ uncorrectable sets of size L all have $d_E = 1$; $c = 52$, 296 and 1556 for the surface codes of distance 5, 7 and 9; $c = 81$ for the $[[72,12,6]]$ code, which has 144 uncorrectable sets of size six with $d_E = 1$ and 12 with $d_E = 2$; and $c = 558$ for the $[[90,8,10]]$ code.

The leading term. As $p \to 0$ the curve approaches its leading term cp^d. Figure 3(b) shows, for the seven largest codes, how soon the two separate. At $p = 0.05$ the ratio $I_H(p)/(cp^d)$ is within 8% of one for nine codes and is 1.14 for the 128-qubit toric code. At $p = 0.1$ it is between 1.14 and 1.30 for the toric and BB codes of 50 to 90 qubits and 1.61 for the 128-qubit toric code, and it peaks between $p = 0.31$ and $p = 0.41$ at 2.6, 2.8, 4.8, 5.9 and 15.6 for the $[[72,12,6]]$, $[[50,2,5]]$, $[[72,2,6]]$,

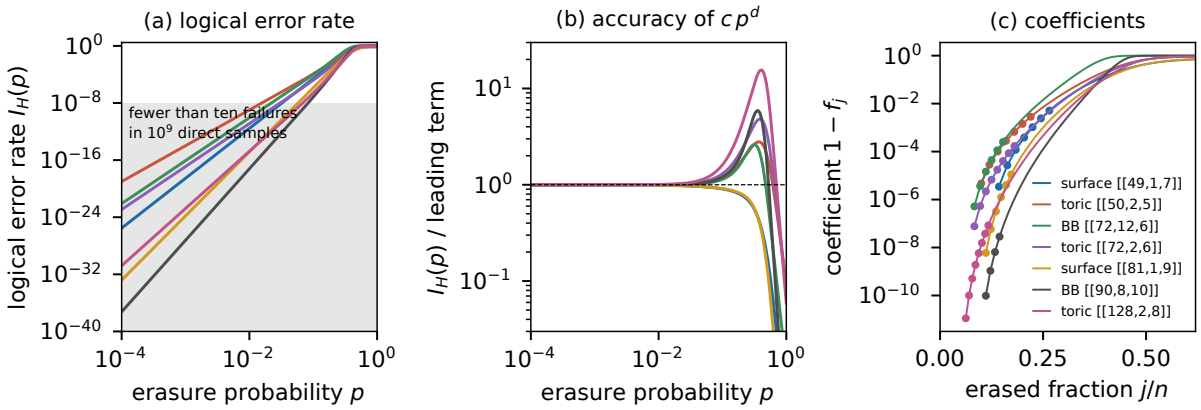


Figure 3: (a) The logical error rate $I_H(p)$ of the optimal erasure decoder for the seven largest codes of Table 2. In the shaded region a direct simulation with 10^9 samples expects fewer than ten failures. (b) The ratio of $I_H(p)$ to its leading term $\binom{n}{d}(1 - f_d)\,p^d$. (c) The infidelity coefficients $1 - f_j$; dots mark the layers $j \le w$, which are computed from the SAT-enumerated minimal sets alone.

$[[90, 8, 10]]$ and $[[128, 2, 8]]$ codes. For the surface codes the deviation has the other sign: the ratio is about 0.91 at $p = 0.1$ and about 0.6 at $p = 0.3$. The distance and the number of minimum-size uncorrectable sets therefore do not give the logical error rate at erasure probabilities of about ten percent and above, and the direction of the error depends on the code family; the graded coefficients of Figure 3(c) are what is needed.

The transition. The erasure probability $p_{1/2}$ at which $I_H = 1/2$ increases with the size of the toric code (0.443, 0.456, 0.463, 0.467, 0.473 for $L = 3, 4, 5, 6, 8$) and decreases with the size of the surface code (0.538, 0.530, 0.525), which is consistent with the erasure threshold $1/2$ of these families [112]. The two BB codes, which encode twelve and eight qubits, have $p_{1/2} = 0.373$ and 0.431; since I_H saturates at $1 - 4^{-k}$, these values should not be compared across different k.

7.3 Limits

The algorithm of Theorem 1, run with an off-the-shelf hashing counter, reached 32 qubits on a workstation. Its guarantee held with a wide margin wherever it could be checked, so the limit is cost; we made no attempt to tune the counter or the encoding. The hybrid estimator is limited by the unsatisfiable instance that closes a level, which becomes harder with the level and with the distance (we did not attempt the $[[144, 12, 12]]$ BB code), and by the size of the lists, 195,328 sets of size 14 for the 128-qubit code. In the layers between w and those where uncorrectable sets are common, the remainder is known only from uniform sampling. It cannot be missed silently, since it is sampled from the whole layer, but if w were further from the sampled layers it would dominate and the method would report a large error. The estimates are statistical; what is proof-checked is the completeness of the lists, with the exception stated above for the largest code. The layer structure used here is that of the iid channel, and we did not implement the multiclass or latent-circuit versions.

8 Correlated erasures generated by a latent circuit

The same reduction applies to a specified class of correlated erasures. Let $C : \{0,1\}^r \to \{0,1\}^n$ be a supplied deterministic circuit over a fixed bounded-arity Boolean basis, independent of the probability parameters. Count both gates and wires, including output wires, in s_C. Draw the latent bits independently with common probability p, set $E = C(b)$, and apply ordinary erasure of E to the encoded state. The draw is independent of that state. Recovery receives the exact set E, but not b. Write $F_{H,C}(p)$ for optimal entanglement fidelity of the maximally mixed logical input, and $I_{H,C}(p) = 1 - F_{H,C}(p)$.

Theorem 19 (Latent-circuit transfer)**.** *Assume the SAT algorithm of Theorem 1. Put*

$$V = n + r + w + s_C, \qquad q = r + 2n, \qquad L = n + r + \mathrm{len}(H, C, \varepsilon, \delta),$$

where $w = \mathrm{nnz}(H)$ and len *denotes the sum of encoding lengths. For $r \geq 1$, a randomized algorithm returns rational coefficients $\widehat{f}_0, \ldots, \widehat{f}_r$. Define*

$$\widehat{F}(p) = \sum_{j=0}^{r} \widehat{f}_j \binom{r}{j} p^j (1-p)^{r-j}, \qquad \widehat{I}(p) = 1 - \widehat{F}(p).$$

With probability at least $1 - \delta$, simultaneously for all $p \in [0,1]$,

$$|\widehat{F}(p) - F_{H,C}(p)| \leq \varepsilon F_{H,C}(p), \qquad |\widehat{I}(p) - I_{H,C}(p)| \leq \varepsilon I_{H,C}(p).$$

Every run has $0 \leq \widehat{f}_j \leq 1$, exact zero infidelity coefficients, and exact endpoints

$$\widehat{f}_0 = 2^{-d_{C(0^r)}}, \qquad \widehat{f}_r = 2^{-d_{C(1^r)}}.$$

If C is promised coordinatewise monotone, the coefficients can also be chosen nonincreasing, so $\widehat{F}$ is nonincreasing and $\widehat{I}$ is nondecreasing. No monotonicity is asserted for a general circuit.

The algorithm uses polynomial space and

$$O\left((r+1)(q+1)^2 \varepsilon^{-2} \log \frac{2(r+1)}{\delta}\right)$$

SAT calls, each with $O(V)$ variables and clauses of width at most three. Its time is

$$\mathrm{poly}(L, \varepsilon^{-1}, \log \delta^{-1}) T(KV)$$

for a constant K that depends only on the gate basis, and each coefficient has $O(n + r + \log \varepsilon^{-1})$ bits. For $r = 0$, one fixed erased set is evaluated exactly by binary linear algebra.

Proof. All latent causes with the same E induce the same erasure channel, so one optimal recovery for the revealed E serves every such cause, and Lemma 7 gives

$$F_{H,C}(p) = \sum_{b \in \{0,1\}^r} p^{|b|} (1-p)^{r-|b|} \, 2^{-d_{C(b)}}.$$

For each j, let A_j^C and B_j^C be the numbers of pairs (b, z) with $|b| = j$ and $z \in N$ that satisfy $\mathsf{Good}(C(b), z)$ and $\mathsf{Bad}(C(b), z)$, respectively. Append full-equivalence Tseitin clauses for C to the certificate circuits of Lemma 9, with a population counter for $|b| = j$, and project onto (b, z) only; this

uses $O(V)$ variables and clauses. The projection must be onto b and not onto E, because different b with the same E carry probabilities that must be added. Proposition 8 gives $A_j^C + B_j^C = \binom{r}{j} Z$ and $f_j^C = A_j^C / (\binom{r}{j} Z)$.

The rest follows the proof of Theorem 1, with $q = r + 2n$ projection bits and $2(r+1)$ counts: estimate, normalize, and replace the two endpoint coefficients by their exact values, computed by linear algebra. For an arbitrary circuit these coefficients are the output, without shape repair; positivity of the Bernstein weights gives both relative guarantees, and the SAT zero tests retain every zero infidelity coefficient, whether or not the zero layers form an initial segment. Under the monotonicity promise, $b \le b'$ implies $C(b) \subseteq C(b')$ and $d_{C(b)} \le d_{C(b')}$, so $f_0^C \ge \cdots \ge f_r^C$ by the same coupling, and Lemma 11 with anchors $\{0, r\}$ gives the shape. In that case the zero infidelity layers are exactly $j < d_C$, where $d_C = \min\{|b| : d_{C(b)} > 0\}$ (infinite if no latent assignment is uncorrectable) is a latent fault weight, not the code distance. □

If $r, w, s_C = O(n)$, the instantiation of Corollary 3 again gives the rate (3), and Remark 4 the rate (4) under its hypothesis.

Monotonicity genuinely needs an additional hypothesis. For one unencoded qubit and $C(b_1, b_2) = b_1$ XOR b_2, the exact fidelity coefficients are $(1, 1/4, 1)$, and

$$I_{H,C}(p) = \tfrac{3}{2} p(1-p).$$

The endpoint fidelities are both one, although the interior infidelity is positive.

8.1 A fixed number of latent probability classes

Partition the latent bits into a fixed number t of classes of sizes $r_1, \dots, r_t$, with independent bits of probability p_a in class a, and $\sum_a r_a = r$. Include the partition description in L. For a weight vector $\mathbf{j} = (j_1, \dots, j_t)$ with $0 \le j_a \le r_a$, define $A_{\mathbf{j}}^C$ and $B_{\mathbf{j}}^C$ by the same projected certificates, requiring weight j_a in each class. Then

$$A_{\mathbf{j}}^C + B_{\mathbf{j}}^C = Z \prod_{a=1}^{t} \binom{r_a}{j_a}, \qquad f_{\mathbf{j}}^C = \frac{A_{\mathbf{j}}^C}{Z \prod_a \binom{r_a}{j_a}},$$

and

$$F_{H,C}(\mathbf{p}) = \sum_{\mathbf{j}} f_{\mathbf{j}}^C \prod_{a=1}^{t} \binom{r_a}{j_a} p_a^{j_a} (1-p_a)^{r_a - j_a}.$$

There are $Q = \prod_a (r_a + 1) \le (r+1)^t$ coefficients. Order the index set $J = \prod_a \{0, \dots, r_a\}$ componentwise, and call $\mathbf{j}$ a *corner* if each $j_a \in \{0, r_a\}$; there are at most 2^t distinct corners, fewer when some $r_a = 0$.

Theorem 20 (Multiclass transfer with exact-corner tensor repair)**.** *Assume the SAT algorithm of Theorem 1, and fix t. There is a randomized algorithm that, given H, C, the class partition and rational $0 < \varepsilon, \delta < 1$, returns rational coefficients $(\widehat{f}_{\mathbf{j}})_{\mathbf{j} \in J}$ in $[0,1]$ such that, with $\widehat{F}(\mathbf{p}) = \sum_{\mathbf{j}} \widehat{f}_{\mathbf{j}} \prod_a \binom{r_a}{j_a} p_a^{j_a} (1-p_a)^{r_a - j_a}$ and $\widehat{I} = 1 - \widehat{F}$, with probability at least $1 - \delta$, simultaneously for every $\mathbf{p} \in [0,1]^t$,*

$$|\widehat{F}(\mathbf{p}) - F_{H,C}(\mathbf{p})| \le \varepsilon F_{H,C}(\mathbf{p}), \qquad |\widehat{I}(\mathbf{p}) - I_{H,C}(\mathbf{p})| \le \varepsilon I_{H,C}(\mathbf{p}).$$

On every run, $\widehat{F} + \widehat{I} = 1$, both curves take values in $[0,1]$, zero infidelity coefficients are retained exactly, and every corner coefficient equals the exact fidelity $2^{-d_{C(b_{\mathbf{j}})}}$ of the single latent assignment

$b_{\mathbf{j}}$ *in that layer. If* C *is coordinatewise monotone, the table* $\widehat{f}$ *is antitone in the componentwise order on every run, so* $\widehat{F}$ *is nonincreasing and* $\widehat{I}$ *nondecreasing in each* p_a*. The algorithm makes*

$$O\Big(Q(q+1)^2\varepsilon^{-2}\log\frac{2Q}{\delta}\Big)$$

SAT calls with $q = r + 2n$*, each with* $O(V)$ *variables and clauses of width at most three, and runs in time* $\mathrm{poly}(L, Q, \varepsilon^{-1}, \log\delta^{-1})\,T(KV)$ *and space* $\mathrm{poly}(L, Q)$*. The output has* Q *coefficients of* $O(n + r + \log\varepsilon^{-1})$ *bits each.*

The dependence on Q is stated explicitly: for fixed t it is polynomial in r, and no polynomial output bound is claimed for a separate probability for every bit.

Proof. The classwise population counters have total size $O(r)$, so the projected counting instances remain of size $O(V)$. Apply Proposition 10 to both counts in every layer with tolerance $\varepsilon/3$ and failure probability $\delta/(2Q)$, and normalize as in Theorem 19; on a single event of probability at least $1-\delta$ every $g_{\mathbf{j}}$ lies in its two-tail interval, and the tensor-Bernstein weights are nonnegative and sum to one. Each corner layer contains a single latent assignment, whose fidelity is computed exactly by linear algebra and substituted. For a general circuit this table is the output; whether the repair below is applied is part of the input, since monotonicity of C is not tested.

When C is coordinatewise monotone the true table is antitone: for $\mathbf{j} \le \mathbf{j}'$, extend within each class a uniform weight-j_a set to a uniform weight-j'_a superset; then $C(b) \subseteq C(b')$ and $f_{\mathbf{j}} \ge f_{\mathbf{j}'}$. Apply Lemma 11 on $(J, \le)$ with the corners as anchors and the endpoints (8). The output is antitone on every run, exact at the corners, and accurate on the accuracy event, and $\partial_{p_a}\widehat{F} \le 0$ by the one-variable Bernstein derivative. The repair makes $O(Q^2)$ comparisons of stored ratios, the oracle budget is $2Q$ applications of Proposition 10, and neither Q nor the accuracy parameters enter the argument of T. □

Theorem 20 with $t = 1$, $r = n$ and C the identity recovers the iid curve of Theorem 1. The machine-checked form of Theorem 1 is proved as exactly this specialization and then restated for the iid channel without tensor vocabulary (Section 9).

Examples. For independently triggered loss events with supports D_γ, $E_i = \bigvee_{\gamma:\, i\in D_\gamma} b_\gamma$ is monotone even when supports overlap, and has linear circuit size when $r = O(n)$ and $\sum_\gamma |D_\gamma| = O(n)$; this includes a constant-length burst triggered at each starting position. A second example is the deterministic-emission Gilbert–Elliott erasure model [2, Section VI] with a fixed initial state: $S_i = (\neg S_{i-1} \wedge A_i) \vee (S_{i-1} \wedge D_i)$ and $E_i = S_i$, with independent A_i, D_i of probabilities $\alpha, 1-\beta$, is an $O(n)$-size circuit with two probability classes. It is not monotone, so the accuracy guarantee applies without the shape. Correlated gate-associated atom loss [8, 97, 98] motivates such models; the extension covers loss patterns that are sampled independently of the encoded state and revealed exactly to the recovery, not state-dependent loss or noisy circuit dynamics.

9 Formal verification

The reduction of this paper has been formalized in Lean 4 as a concrete machine, and composed with a verified external SAT engine. This section states exactly what has been checked, in which toolchain, and what has not. All checked statements depend only on Lean's standard axioms `propext`, `Classical.choice` and `Quot.sound`, as reported by `#print axioms` for each named theorem; no `sorry`, added `axiom` or `native_decide` occurs.

9.1 The checked statements

The composition with a verified engine. Under Lean v4.34.1 and Mathlib commit `d13f23b7`, the external development of [95] (commit `4eaee875`, with proof-internal and import-only edits to 108 of its files that leave every exported statement unchanged and are listed in the artifact) and the reduction are checked in one 6,157-module graph. The engine is consumed through its theorem `SqrtLogSpace.sat_polynomial_space`, described in the proof of Corollary 3. The reduction is stated against a contract for a SAT machine, `ResourceEncodedSAT`, which charges the ambient variable count in unary plus the binary query length, and the bridge theorem `ConcreteSolver.exists_resource_machine` proves that one fixed engine satisfies it (for requests with at most as many clauses as variables; the reduction pads its queries to meet this), with polynomial envelopes and the monotone runtime envelope $T(v) = \max_{s \le v} \lceil 2^{A\rho(s)} \rceil + 1$. Its proof uses the verified encoding bridge, the satisfiability-preserving formula translation and a proved polynomial bound on the frontend's own time and space. The closed theorems below therefore have no SAT, hashing, certificate or physical-lift hypothesis. The same guarantee for an arbitrary engine that satisfies the contract, with T as a parameter and with the hash certificate and the physical lift as hypotheses, is `TensorMachine.Runtime.tensor_machine_guarantee`, stated for probability classes.

Theorem 1 and Corollary 3 as one declaration. The theorem

`QecReliability.IidErasure.iid_erasure_machine_guarantee`

states the iid case directly: the channel, the coefficients and the guarantee are expressed without latent circuits or tensor vocabulary. There exist an engine, a rate constant $A > 0$ and polynomial envelopes P_t, P_s such that, for every n, every list of check rows on n qubits with isotropic row space S, and all natural numbers $0 < a \le b$, $0 < c$, $0 < d$ (so $\varepsilon = a/b$, $\delta = c/d$):

(i) the entanglement fidelity $F(p)$ of one fixed flag decoder composed with the channel that erases each qubit independently with probability p equals $\sum_j f_j \binom{n}{j} p^j (1-p)^{n-j}$, where f_j is the average of 2^{-d_E} over $|E| = j$, and no recovery channel has larger fidelity, for every $p \in [0,1]$;

(ii) for *every* string of N coin bits appended to the input, the fixed machine M has a completed run that leaves a rational vector $(\widehat{f}_0, \dots, \widehat{f}_n)$ on its output tape, in time at most $P_t(L_*)(B+1)(T(\kappa'(L_{\mathrm{code}}+1))+1)$ and at most $P_s(L_*)$ visited cells, with $1 = \widehat{f}_0 \ge \widehat{f}_1 \ge \cdots \ge \widehat{f}_n = F(1) = 4^{-(n-\dim S)}$ and $\widehat{f}_j = 1$ if and only if $f_j = 1$;

(iii) with probability at least $1 - c/d$ over the uniform law on the N coins, the curve $\widehat{F}(p) = \sum_j \widehat{f}_j \binom{n}{j} p^j (1-p)^{n-j}$ satisfies $|\widehat{F}(p) - F(p)| \le \varepsilon F(p)$ and $|\widehat{F}(p) - F(p)| \le \varepsilon(1 - F(p))$ simultaneously for every $p \in [0,1]$.

Here N is a number of coin bits and not the normalizer, $T(v) = \max_{s \le v} \lceil 2^{A\rho(s)} \rceil + 1$, $B = \lceil 144 b^2/a^2 \rceil$, κ' is a fixed constant, $L_{\mathrm{code}} = 2n + \mathrm{nnz}(H) + 1$, and $L_* = L + t + Q$ in the notation of Theorems 19 and 20, with one class of size n ($t = 1$, $Q = n+1$) and the identity circuit. Item (ii) is the monotone shape, the exact endpoints and the deterministic identification of zero infidelity coefficients of Theorem 1, for the coefficients the machine writes. The statement differs from the prose of Theorem 1 in three ways. It makes no sparsity assumption, and the argument of T is linear in $n + \mathrm{nnz}(H)$, where $\mathrm{nnz}(H)$ stands for the total length of the coordinate lists that present the rows. The confidence parameter enters the time and space bounds only through L_*, which includes the

encodings of c and d. And the machine is the multiclass machine, run on one class and an encoding of the identity circuit; the theorem is proved by this specialization and a proved identity between the one-class tensor channel and the iid channel.

Resources and the other closed theorems. A second form of the theorem,

`IidErasure.iid_erasure_machine_guarantee_resources`,

adds the content of Proposition 2 with explicit constants, for every coin string: each output coefficient is a pair of natural numbers of at most $3n + \lceil \log_2(4B+1) \rceil + 1$ bits; the counting procedure makes at most $512(L_* + 1)^4(B + 1)$ SAT calls; and every query is a width-three request with at most $\kappa'(L_{\mathrm{code}} + 1)$ variables and as many clauses. The calls counted are those of the source-level counting procedure whose output the machine writes; they are tied to the machine run through the time bound, and not by an identification with tape-level engine invocations.

The other models have closed theorems of the same form (Table 3):

Declaration	Model
`isotropic_polynomial_endpoint_machine_guarantee`	latent circuit, arbitrary (Theorem 19)
`latent_monotone_machine_guarantee`	latent circuit, monotone
`isotropic_tensor_machine_guarantee_strong`	probability classes, monotone (Theorem 20)
`isotropic_tensor_machine_guarantee_general`	probability classes, arbitrary
`uncorrectable_machine_guarantee_sized`	U_H, through the repeated code of Remark 18

In every one of them the engine, the rate constant and the polynomial envelopes are chosen *before* the dimensions and inputs, and the machine depends only on the engine and a fixed program. The theorem for an arbitrary latent circuit has no monotone-shape conclusion, because its machine applies no repair; below it is called the latent theorem, and the theorem for monotone probability classes the strengthened multiclass theorem. All statements, and that of the external engine, are reproduced verbatim in the artifact's `docs/` folder, with the audit files in `tools/audit/`; their full text is longer than a page because every bound is written out.

Two audits of the compiled graph, one printing the axioms of the 147 declarations cited in this section and in Table 3 by name, the other collecting the axioms of all 2,386 declarations of the thirty-four final modules, report only the three standard axioms.

9.2 Correspondence with the paper

Kernel checking establishes the Lean statements, not their identity with the prose. Table 3 in Appendix C records the correspondence claim by claim. In summary: every theorem, proposition, corollary and lemma of Sections 3–8 has a kernel-checked counterpart, at machine level for the algorithmic statements, with four exceptions that are stated there: the rates of Remarks 4 and 5, the final step to NP-hardness, where the reduction from SAT is checked but the Cook–Levin theorem and the polynomial running time of the reduction are not, the explicit hash of Appendix A, for which the machines use a different certificate, and the direct algorithm in the proof of Corollary 17, which is checked at the level of its formula while the machine-level theorem runs a different algorithm. The exact-distance and weight-enumerator algorithms of Section 6, which are not numbered statements, are checked at the level of their formulas only. The numerical content of Section 7 is experimental, with two exceptions described there: the exhaustive tables of the seven codes of at most 25 qubits were recomputed by a verified Lean function, and the lists of minimal uncorrectable sets are certified

complete. These certificates are results of running compiled Lean programs whose checks are proved sound; they are not theorems reduced in the kernel.

9.3 Trusted base, constants, and artifact

The trusted base is the Lean 4 kernel, Mathlib, and the definitions on both sides of the bridge: the external development's CNF semantics, machine model and input encoding, and our definitions of the erasure channel, entanglement fidelity, the multitape machine, completed runs, visited cells and the output predicate. The machine model is a deterministic finite-state multitape machine; the external `PrimitiveMachine` scans one cell per step and is transported into it with a proved time and space overhead. Two kinds of non-transparent declaration occur and introduce no axiom: the external solver seals six finite instruction tables as `private opaque` constants, and our tactic support code contains three `partial def` helpers. An independent kernel replay with `leanchecker`, which re-checks every declaration of a module in the kernel from its compiled imports, was run on all 6,157 modules of the graph, Mathlib and the external solver included, and passed.

Every closed statement is existential in the engine, the rate constant and the polynomial envelopes, and the reliability program depends on a fixed hash certificate. The constants are doubly exponential: the certificate's expander walk has 2^{56} steps, so the random-bit length N and the SAT-size constant κ both exceed $2^{2^{58}}$ for every input, including $n = 0$. The theorems are therefore asymptotic statements, as Remark 6 says for the paper constants.

The Lean development, the audit files, and the code and data of the experiments are public at https://github.com/Oliverds321/stabilizer-erasure-sat-lean; this paper describes release `v1.0.0`.

10 Related work

Erasure in quantum codes. The quantum erasure channel and its capacity are classical topics [9, 59]. The per-pattern optimum 2^{-d_E}, the rank-of-H_E combinatorics and equal coset probabilities appear in Delfosse and Zémor's rate bounds for LDPC stabilizer codes [39], in the optimal recovery probability of Gullans et al. [61], and in degenerate erasure decoding [87]. Fast maximum-likelihood erasure decoders exist for surface and hypergraph-product codes [34, 40]. Ashikhmin [4] studies finite-length codes on the same channel. That paper states the recoverability criterion, the per-pattern fidelity and the comparison $I_H \le U_H \le 2I_H$, and then bounds the probability P_{de} of an unrecoverable pattern, our U_H, over all codes of given parameters, by MacWilliams-type identities, linear programming and random-coding arguments; it does not consider computing the curve of a given code. Our enumeration reproduces the seventeen invariants printed in its Example 1, and the verified table of `ExactTable` its numbers of unrecoverable sets (script `ashikhmin_example.py`). Mandal, Gökduman and Pfister [89] prove area theorems and an erasure-folding identity for a related but different curve, the entropy of the logical correction class given the syndrome, which for erasure is linear in d_E where ours is 2^{-d_E}; the expression of erasure failure as a sum over uncorrectable patterns is used in Freire et al. [54]. Rare-event estimation of logical error rates by splitting and subset sampling [11, 16, 65, 91], coherent-information thresholds by Monte Carlo over erasure configurations [33], statistical-mechanics estimates of logical error curves under Pauli noise [120], and shot-free leading-order predictors [106] are the practical alternatives; none gives a worst-case relative guarantee on the whole curve. The hybrid estimator of Section 7.2 belongs with these methods: it stratifies by erasure weight, as subset sampling does, and replaces sampling in the rare layers by a complete SAT enumeration of minimal uncorrectable sets.

Enumerators and hardness in quantum coding. Quantum weight enumerators [100, 107] and their tensor-network generalizations [21, 22] compute exact logical error rates when a low-complexity contraction exists; support-sensitive enumerators [104] contain erasure information. Decoding stabilizer codes is NP-hard [66] and optimally decoding them is #P-hard [72], with refinements in [50, 86]; the minimum distance problem is NP-hard [79] and hard to approximate [60, 78], extending the classical results [10, 119]. Proposition 12 connects our guarantee to the last of these.

Reliability polynomials. The Bernstein form $\sum_j f_j \binom{n}{j} p^j (1-p)^{n-j}$ is the N-form of a reliability polynomial [32, 93]. Exact network reliability is #P-hard [99, 117]; matroid reliability is a Tutte-polynomial specialization [19], #P-hard to evaluate for binary matroids outside special points [73] and hard to approximate in large regions of the plane [57]. Relative approximation of the two tails has separate histories: Karger's scheme for unreliability [80] and its successors [25, 26, 64, 81]; Guo and Jerrum's scheme for reliability [62]; and, for two-terminal reliability, a scheme for one tail while the other is #BIS-hard [48, 49]. Anari et al. approximate matroid reliability for all p coefficientwise via log-concavity [3]; Section 6 shows the quantum weights need not be log-concave. We are not aware of a prior notion of simultaneous relative approximation of both tails from one coefficient vector.

Approximate counting with a decision oracle. Approximate counting with pairwise-independent hashing and an NP oracle goes back to [24, 74, 108, 113, 118]. Practical SAT-based counters follow the same design [6, 27, 28, 58], with dense XOR constraints or with short or sparse ones that give up strong universality [1, 47, 92, 123]. We keep exact pairwise independence at linear size, following Ishai, Kushilevitz, Ostrovsky and Sahai [71], who use Spielman's linear-time codes [111]; see also [17, 18, 45, 63, 90]. Approximate model counting has been formally verified in Isabelle/HOL [114] as a per-run certificate; our formalization of the estimator is a resource-bounded machine proof.

Fine-grained complexity. The Exponential Time Hypothesis and sparsification are due to Impagliazzo, Paturi and Zane [20, 68, 70]. Fine-grained reductions from approximate counting to decision are in [42, 43, 116], and lower bounds for the Tutte polynomial, the classical cousin of our curve, under the counting version of ETH in [41]. The refutation of ETH by OpenAI [95] and the Lean-verified $2^{o(n)}$ 3-SAT algorithm of Anthropic [31] are recent and under public examination [44, 53, 69, 83, 88, 94]. Given the kernel-checked engine theorems, lower bounds conditional on ETH no longer constrain anything, and we use no such bound. The only quantum consequence we know of is an announced classical sampling result for linear-size quantum circuits [56]; as of 10 November 2026 we know of none that concerns codes or reliability.

Formalized computation and quantum codes. Mechanized complexity theory in proof assistants includes Turing machine formalizations [5, 23, 122], the Cook–Levin theorem [7, 55], cost-model reasonability [52] and descriptive complexity [105]. In Lean 4 [37] with Mathlib [115], stabilizer codes have been formalized end to end with a verified reduction of distance to SAT [46], and the quantum Singleton bound via erasure correctability [38]. Our development formalizes a randomized resource-bounded algorithm together with the external engine it calls; we know of no earlier machine-checked complexity bound that composes a formalized reduction with a formalized subexponential engine.

11 Limitations and open problems

The results give a complete chain from a SAT oracle to the erasure curve: a reduction with linear-size queries, its composition with a subexponential engine checked in one Lean development, NP-hardness for sparse presentations, and certified logical error rates for codes of up to 128 qubits. The points below delimit these results and record what remains open.

1. *Constants.* Lemma 21 has absolute but large constants, and the formalized certificate has constants exceeding $2^{2^{58}}$. The crossover against exhaustive enumeration (Remark 6) is far beyond any physical code size. Reducing the hash constants, or formalizing the explicit construction of Appendix A, is the first step toward any practical reading.

2. *Dependence on an unrefereed result.* The rate (4) depends on a claimed algorithm outside the Lean development of [95]; the rate (3) depends on a kernel-checked theorem whose informal interpretation as a Word-RAM bound is still under examination [69]. Our machine-level composition removes the interpretation step for Corollary 3 but not the question of whether the external definitions model what they claim.

3. *Matching lower bound.* Corollary 14 gives NP-hardness for sparse presentations, but through a padding that is not size-preserving, so it yields no fine-grained lower bound. A converse reduction from bounded-density SAT on v variables to the curve of a sparse code on $O(v)$ qubits would turn Theorem 1 into an equivalence. Hardness for presentations of bounded row and column weight, where padding is unavailable, is also open; for classical regular LDPC codes the minimum distance problem is NP-complete [75], and we do not know the stabilizer analogue.

4. *Exact evaluation.* We do not know whether exact evaluation of $F_H(p)$ is #P-hard for sparse stabilizer codes, or whether it is NP-hard at a fixed constant p; Corollary 16 covers only p exponentially small in n.

5. *Scope.* The results concern flagged erasure of stabilizer codes with a maximally mixed logical input. Unflagged noise, worst-case fidelity, nonstabilizer inputs, state-dependent loss and noisy circuit dynamics are outside scope. Correlated erasures are covered only through a supplied size-controlled circuit.

6. *Extent of the formalization.* Section 9 and Table 3 state claim by claim what is kernel-checked. The main statements that are not are the explicit hash of Appendix A, in whose place the machines use a fixed certificate, and the Cook–Levin theorem behind the NP-hardness statements.

Acknowledgement. The AI systems Claude (Anthropic) and Codex (OpenAI) assisted with development, auditing, formalization, the computational experiments and manuscript preparation; the author is responsible for the mathematical claims and submission.

A All-length linear-size pairwise-independent hashing

Theorem 1 requires *Boolean gate and wire* bounds, not merely an arithmetic-operation or RAM bound. Ishai, Kushilevitz, Ostrovsky and Sahai [71, Theorems 3.2–3.3] construct linear-size pairwise-independent hash circuits, but the stated length guarantee is for infinitely many input lengths.

We record the all-length alignment used here. The construction imports the explicit encoder and expander results below; it does not re-prove their number-theoretic ingredients. The formalized development uses a different certificate (Section 9).

Lemma 21. *There are absolute constants C, a and a deterministic polynomial-time procedure that, for every $q \geq 1$, constructs a fan-in-two, fan-out-two Boolean circuit*

$$h_U : \{0,1\}^q \longrightarrow \{0,1\}^q$$

using at most Cq gates, wires and seed bits. For uniform seed U and distinct x, y, the pair $(h_U(x), h_U(y))$ is uniform on $\{0,1\}^{2q}$. The description is constructible in $O(q^a)$ time. Every output prefix is likewise exactly pairwise independent.

Proof. Classical ingredients. Spielman's construction [111, Lemma 2 and Theorem 19] provides polynomial-uniform injective binary linear encoders

$$G_\ell : \mathbb{F}_2^{L_\ell/4} \longrightarrow \mathbb{F}_2^{L_\ell}, \qquad L_\ell = L_0\, 2^\ell,$$

with distance at least γL_ℓ, for an absolute $\gamma > 0$, and $O(L_\ell)$-size XOR circuits. Fixed local linear maps give Boolean XOR circuits, not just unit-cost arithmetic algorithms. The explicit expander constructions discussed in [111, Section 3, especially Proposition 6, Definition 9 and the following density paragraph] supply a dense polynomial-uniform family of regular constant-degree graphs with a fixed spectral gap. Uniformity includes selecting an eligible size: for all sufficiently large L, a graph with $L \leq m \leq 2L$ can be found and constructed in time polynomial in L. For example, fix the prime parameter 5 in that source's Theorem 5 and scan primes $\varrho \equiv 1 \pmod{20}$ with $m = \varrho(\varrho^2 - 1)/2$ in this interval. The displayed arithmetic conditions are then satisfied; the density paragraph's prime-gap input guarantees such a size for large L. Trial division, the finite scan and Proposition 6 give polynomial time. Thus we do not assume that a suitable size is supplied as advice. All statements here allow finitely many small exceptions, absorbed by fixed hardwired circuits.

Distance amplification. Choose the least $L = L_\ell$ with $L/4 \geq q$, pad the input by zeros and apply G_ℓ; thus $L = O(q)$. Choose an expander on m vertices with $L \leq m \leq 2L$ and pad the encoded word to m coordinates. Make the walk lazy by adding one holding choice for each outgoing edge. Its symmetric transition matrix P is positive semidefinite, its gap $g > 0$ is constant, and its degree D_0 is constant.

For distinct inputs let $\mathcal{A}$ be the agreement set of their padded codewords. Then $|\mathcal{A}|/m \leq 1 - \gamma/2$. Writing $Q_\mathcal{A}$ for coordinate projection, every vector v supported on $\mathcal{A}$ satisfies

$$\langle v, Pv\rangle \leq (1-g)\|v\|^2 + \frac{g}{m}\langle v, \mathbf{1}\rangle^2 \leq (1 - g\gamma/2)\|v\|^2.$$

Put $\vartheta = 1 - g\gamma/2 < 1$. Positivity implies $\|Q_\mathcal{A} P Q_\mathcal{A}\| \leq \vartheta$. A stationary walk of τ visited vertices therefore lies entirely in $\mathcal{A}$ with probability at most

$$m^{-1}\mathbf{1}_\mathcal{A}^T (Q_\mathcal{A} P Q_\mathcal{A})^{\tau-1}\mathbf{1}_\mathcal{A} \leq \vartheta^{\tau-1}.$$

Fix τ with $\vartheta^{\tau-1} < \gamma/4$, and form a τ-bit symbol for each of the $M = mD_0^{\tau-1} = O(q)$ walks. Distinct inputs agree on fewer than $\gamma M/4$ symbols. These symbols require only $O(\tau M)$ wires.

Exact length alignment. Choose the least available dyadic block length $L' \geq M$. Then $L' \leq 2M$ and $L'/4 \geq q$, because $M \geq L \geq 4q$. Repeat the *whole* τ-bit symbols as evenly as possible to obtain exactly L' symbols. Each is used at most twice, so distinct inputs agree at fewer than

$$2(\gamma M/4) \leq \gamma L'/2 < \gamma L'$$

positions. No partial-width input symbol is introduced.

Local hashing and extraction. For each repeated position i, independently choose uniform $a_i \in \mathbb{F}_2^\tau, b_i \in \mathbb{F}_2$ and output $r_i(v) = a_i \cdot v + b_i$. Distinct symbol values give independent uniform output bits; equal values give equal bits. Seeds on repeated copies are *fresh and independent*; copying an already hashed bit would not justify this argument.

Let $G' : \mathbb{F}_2^{L'/4} \to \mathbb{F}_2^{L'}$ be the encoder at length L', and let $f = (G')^T$. Deleting fewer than $\gamma L'$ columns from f leaves full row rank: otherwise a nonzero row combination would be a nonzero codeword of weight below the distance of G'. Thus f maps independent uniform bits on the remaining coordinates to a uniform $L'/4$-bit string, whatever the fixed values on the deleted coordinates.

Condition on seeds at the equal-symbol positions of two inputs. At every other position the two output bits are independent uniform, independently across positions. The number of fixed positions is below the distance of G'. It follows that $f(r(x))$ and $f(r(y))$ are independent uniform strings under every such conditioning, hence unconditionally. Retaining their first q bits proves the desired pairwise independence.

Size and uniformity. The encoders have linear-size XOR circuits; walk symbols and local affine evaluations use $O(q)$ wires and gates, and $(\tau+1)L' = O(q)$ random bits. For explicit wire accounting, a circuit with S gates of fan-in at most two and r output sinks has at most $2S + r$ incoming edges in total. This includes every use of an input, seed bit or zero-padding source, and every output edge. Replace each fan-out by a binary copy tree; the number of new gates and edges is linear in this edge total. Apply this normalization globally, including sources and output sinks. All these counts are $O(q)$. Transposition preserves linear size: reverse the normalized linear encoder circuit, interchanging XOR and duplication and supplying zero sources for unused reverse inputs. At most one is needed per existing vertex, so these sources also number $O(q)$. The resulting hash circuit has fan-in and fan-out at most two and $O(q)$ edges; no dense parity rows are hidden in the bound. All graphs, walks, repetitions and circuits are generated in polynomial time. For the finitely many small q, fixed affine hash circuits are covered by enlarging C. □

The seed in Lemma 21 is hardwired before constructing each SAT query. It is never existentially quantified. The preprocessing exponent and the circuit-size constant are absolute and may be large. Their independence from the accuracy parameters is what the runtime transfer uses.

B Projected counting and oracle accounting

Let $\Phi(x, y)$ be a 3-CNF and $U = \{x : \exists y,\ \Phi(x, y)\} \subseteq \{0,1\}^q$. If $q = 0$, one SAT query determines whether the unique empty projection is present, so return that count exactly. For a fixed seed append a full-equivalence Tseitin encoding of an output-prefix condition $h_r(x) = 0$. Lemma 21 adds $O(q)$ variables, clauses and wires.

Capped enumeration. To enumerate at most K distinct projected assignments, traverse the binary prefix tree of x. Test SAT at each visited prefix using unit clauses; prune unsatisfiable nodes, count satisfiable leaves, and stop upon reaching K leaves. The number of queries is at most $1 + 2(q + 1)K$: successful prefixes lie on paths to encountered leaves, and unsuccessful sibling tests charge to those paths. If the root is unsatisfiable, one query suffices. The wrapper stores a depth-q stack, a counter and one formula. Existential witnesses are not enumerated, and no collection of long blocking clauses is accumulated.

One estimation trial. First test SAT, returning zero exactly if U is empty. For $0 < \eta < 1$, set

$$B_0 = \lceil 16/\eta^2 \rceil, \qquad \tau_0 = 4B_0.$$

Draw one full q-bit hash and consider its nested prefixes $r = 0, \ldots, q$. At each scale count $C_r = |U \cap \{h_r = 0\}|$, capped at $\tau_0 + 1$. At the first count at most τ_0, return $\max\{1, 2^r C_r\}$. If no scale stops, return one. These conventions make the estimator total and keep every positive instance's estimate positive, including on a failed trial. Only the initial SAT test certifies zero.

Lemma 22. *One trial is relatively accurate within η with probability at least $7/8$.*

Proof. Write $M = |U|$. For $M \le \tau_0$, scale zero is exact. Otherwise choose

$$r_* = \lfloor \log_2(M/B_0) \rfloor, \qquad B_0 \le \mu_* = M/2^{r_*} < 2B_0.$$

Since $M \le 2^q$, $r_* \le q$. Pairwise independence gives $\mathbb{E}C_r = \mu_r = M/2^r$ and $\mathrm{Var}(C_r) \le \mu_r$. Chebyshev's inequality and the geometric sum yield

$$\mathbb{P}\{\exists r \le r_* : |C_r - \mu_r| > \eta\mu_r\} \le \sum_{r=0}^{r_*} \frac{1}{\eta^2 \mu_r} < \frac{2}{\eta^2 \mu_*} \le \frac{1}{8}.$$

No independence between scales is required. On the complementary event, $C_{r_*} < (1+\eta)2B_0 < \tau_0$, so a stop occurs by r_*. Every earlier possible stop is also relatively accurate and has a nonzero count. Rescaling proves the claim; the positive fallback conventions do not affect this event. □

Repeat independently an odd number $s \ge (32/9)\log(1/\zeta)$ of times and take the median. Hoeffding's inequality bounds the probability that at least half the trials fail by

$$\exp\{-2s(7/8 - 1/2)^2\} = \exp(-9s/32) \le \zeta.$$

Each trial uses $O((q+1)^2(\tau_0+1))$ SAT queries. All prefix restrictions contain at most q unit clauses, so the query size remains linear in the original formula size. Sequential processing and storage of the polynomially many trial outputs give polynomial space. Each output is at most $\max\{1, 2^q\tau_0\}$, hence has $O(q + \log \eta^{-1})$ bits. This proves Proposition 10.

C Paper-to-Lean correspondence

Table 3: Paper-to-Lean correspondence. "Checked" means a kernel-checked declaration with standard axioms whose statement the authors judge to express the paper claim; "mathematical core" means that the combinatorial and analytic content is checked but no machine or resource statement is. The reader is referred to the artifact for the exact text.

Paper statement	Formal status
Lemma 7, optimality over all CPTP recoveries	Checked: conjunct (i) of the iid theorem, the corresponding conjunct of the strengthened multiclass theorem, and `PhysicalTargetOptimality` in the latent theorem.

Paper statement	Formal status
Proposition 8, Lemma 9	Checked inside the machine construction (CNF query transcribed and proved equal to the specification).
Lemma 11 (anchored antitone repair)	Checked in `PosetShapeRepair`: `repair_antitone`, `repair_range`, `repair_anchor`, and the interval claim for general lo, hi as `repair_interval`.
Theorem 1: one algorithm, two-tail accuracy, monotone coefficients, exact endpoints, deterministic identification of zero infidelity coefficients, time and space	Checked as one declaration, `IidErasure.iid_erasure_machine_guarantee`, with the differences from the prose listed in Section 9; also, for an arbitrary engine satisfying the contract, as `tensor_machine_guarantee` (the form for probability classes, with the shape stated for the curves).
Proposition 2: query count, query size, coefficient bit length	Checked as conjuncts of `iid_erasure_machine_guarantee_resources` and of its multiclass form, with explicit constants; Section 9 says what is counted.
Corollary 3	Checked: the same declaration, in which the engine is the concrete verified one and T is its rate envelope.
Remarks 4, 5, 6	Not formalized; paper-level.
Proposition 12 (distance determination)	Checked as a mathematical statement, for coefficients of arbitrary sign: `DistanceDetermination.zero_prefix_iff` (analytic form) and `Lift.distance_determination` (for the iid erasure channel: the zero prefix of any accurate infidelity table is exactly the set of sizes at which every erased set is correctable).
Corollary 14 (hardness for sparse presentations)	Padding step checked: `Padding.erasureScore_pad`, `Padding.isotropicLift_fidelity_pad` (the curve is unchanged), `Padding.rowSpace_padRows` and `Padding.rowIncidences_padRows` (exactly one new nonzero entry per new qubit). The reduction itself is a pair of Lean functions with a correctness theorem (`DistanceExtraction.cor_sparsehard_extraction`): pad to the target sparsity, run any algorithm meeting the guarantee, and extract the distance. The reduction from SAT is checked as well: Proposition 13 is `SatReduction.reduce_correct` with `reduce_isDistance_le_iff`, `reduce_isDistance_eq`, `reduce_logical_qubits` ($k = 1$) and the size bound `reduce_size`, and its composition with the extraction is `SatReduction.sat_iff_table`: on the padded code of a formula φ, any output meeting the guarantee has first nonzero index at most $t + 1$ exactly when φ is satisfiable. Not formalized: the Cook–Levin theorem and the polynomial running time of the reduction (its output size is bounded).
Proposition 15 (a point estimate determines the distance)	Checked (module `PointHardness`): the bounds (11) as `tail_sandwich`, and the uniqueness of the exponent as `infidelity_point_determines_distance`, its channel form, and `uncorrectableProb_point_determines_distance`.

Paper statement	Formal status
Corollary 16; exact distance and enumerators from a SAT engine	The pointwise reduction is checked as Lean functions with a correctness theorem (`DistanceExtraction.cor_pointhard_extraction`, and its form for U_H). The certificate equivalence for $d \le t$ is checked (`distance_le_iff_certificate`), and so is its CNF form: $d \le t$ exactly when the formula `certFormula` at level t is satisfiable, and d is the least such t (`distance_le_iff_sat`, `distance_isLeast`), which is the correctness of the binary search. The reduction from SAT to one point estimate is `SatReduction.sat_iff_point` (and `sat_iff_pointU` for U_H). The enumerator formulas are checked as well (module `EnumeratorFormula`): a 3-CNF on $(2t+5)n+\mathrm{nnz}(H)+2$ variables whose number of models is the number of elements of N of weight t (`normWeightFormula_count`), and one on $(2t+3)n+\mathrm{nnz}(H)+m+2$ variables whose number of models is 2^{m-s} times the number of elements of S of weight t (`stabWeightFormula_count`), with the clause width and the variable counts. These are statements about the formulas; a machine that calls a counting engine on them is not formalized, and neither are the Cook–Levin theorem and the polynomial running time of the reduction.
Section 6.3 (enumerators do not determine the curve) and the worked examples of the text	Checked (module `EnumeratorExample`): the per-pattern values, layer sums and curves of both three-qubit codes, their difference $\frac{3}{4}p(1-p)^2$ for the optimal physical fidelities, and the equality of the weight counts of both stabilizer groups and both normalizers; also the single unencoded qubit ($I = 3p/4$, $U = p$) and the list of correctable sets of the four-qubit code of Section 6. Module `PaperExamples` checks the others: the Bell-state counting domain (Good has probability $1/4$ over the ambient space and one over N); the witness multiplicities two and eight, for every positive Bad pair; correctability zero, entanglement fidelity $1/2$ and average fidelity $2/3$ for one erased repetition-code qubit (the Haar-average formula itself is quoted, not formalized); the failure of log-concavity for the repetition code with five appended stabilized qubits, with the positive definite two-dimensional restriction of the Hessian; and the latent XOR example, for the optimal physical fidelity.
Section 6.4 (sampling comparison)	Checked. Both bounds of (12): `PaperExamples.tail_bound`, with the binomial tail in closed form and $I_H \le U_H$ in between; the identification of d with the minimum weight of a nontrivial logical operator: `isDistance_iff_minWeight`. Module `SamplingBounds`: the Chernoff bound for the binomial tail, positivity of the divergence, the resulting $\frac{1}{2}p^n \le I_H(p) \le e^{-n\mathcal{D}(\alpha\|p)}$, the probability $(1-\theta)^M$ of the all-zero sample, and the sample-size inequality. Module `RankSampling`: for M independent erasure patterns, the probability $(1-U_H(p))^M$ that all are correctable and the bound $M \ge \log(1/\delta)/(4\,I_H(p))$ for the average of the rank values (`rank_sample_size`).

Paper statement	Formal status
Corollary 17 and the bounds $I_H \le U_H \le 2I_H$	Mathematical core checked (module `UncorrectableProbability`): the certificate identity, monotonicity under inclusion, the Bernstein form of U_H, the two inequalities against the channel infidelity, and the repair step (`repairedCurve_rel`, `repairedCurve_monotoneOn`, `repairedFraction_eq_zero_iff`). The algorithm is checked at machine level in the repeated-code form of Remark 18 (`UncorrectableMachine.uncorrectable_machine_guarantee`, and `uncorrectable_machine_guarantee_sized` with the explicit time bound; the exponent identity $d_{\widetilde{E}} = m\, d_E$ in module `ProductCode` is used): run, time and space on every coin string, a nondecreasing output with exact zero coefficients, and the two-sided bound $(1-\varepsilon')(1-2^{-m})U_H \le \widehat{U} \le (1+\varepsilon')U_H$ with probability at least $1-\delta$. The direct algorithm of the proof is checked at the level of its formula (module `UncorrectableFormula`): a 3-CNF on at most $29n + 4\,\mathrm{nnz}(H) + 12$ variables whose models, projected onto the n erasure bits, are exactly the uncorrectable sets of size j (`uncFormula_projected`, `uncFormula_count`), with $U_H(p) = \sum_j C_j p^j (1-p)^{n-j}$ (`uncorrectableProb_eq_counts`). A machine that runs it, with its exact right endpoint, is not formalized.
Theorem 19, arbitrary C	Checked (latent theorem): run, time, space, exact endpoints, range, two-tail accuracy, optimality. Its second form, `isotropic_latent_machine_guarantee_resources`, adds on every run the exact endpoint coefficients, the identification $\widehat{f}_j = 1 \Leftrightarrow f_j = 1$ of the zero infidelity coefficients, the coefficient bit lengths, the number of SAT calls and the size of every query (the last two for $r > 0$).
Theorem 19, monotone C	Checked as its own declaration, `LatentMonotone.latent_monotone_machine_guarantee`, the strengthened multiclass theorem with one class: antitone coefficients, exact endpoints, exact identification of zero infidelity coefficients, the fact that these form a prefix $j < d_C$, optimality and accuracy.
Theorem 20, monotone C	Checked (strengthened multiclass theorem): antitone table, exact corners, exact zero infidelity coefficients, accuracy, optimality. Space $P_s(L_*)$ charges Q as in the statement.
Theorem 20, general C (accuracy without repair)	Checked: `isotropic_tensor_machine_guarantee_general`, with no monotonicity hypothesis: run, time and space of the same form, values in $[0,1]$, exact corners, exact identification of the coefficients equal to one, optimality of the target and two-tail accuracy. Its machine substitutes the exact corner values and applies no repair; monotonicity is a promise on the input, as in the theorem.
Lemma 21 (explicit all-length hash)	Not formalized as stated. The machine uses a fixed pairwise-independent certificate of parameters 16^{384} and 2^{1528}, with expander walks of 2^{56} steps. It is not a fan-out-two circuit and is not polynomial-time constructible in the sense of Appendix A.
Proposition 10 and Appendix B (estimator)	Checked inside the machine construction.

Paper statement	Formal status
Section 7: identities behind the hybrid estimator	Checked (modules `HybridEstimator`, `HybridSymmetry`): the duality $d_E + d_{E^c} = 2k$ and its layer form, existence of a minimal uncorrectable subset, the union identity of Karp, Luby and Madras, its split form with a remainder, the passage from coefficientwise to curvewise relative error, the invariance of d_E and of minimal uncorrectable sets under a qubit permutation that preserves S, and the reduction of the search to orbit representatives. Module `CompletenessFormula` defines the CNF formula used to certify that a list of minimal uncorrectable sets is complete and proves what its unsatisfiability means (`certFormula_unsat`, `certFormula_checked`); the refutations are checked by the LRAT checker of Lean core. Module `ListCertificate` proves sound the remaining checks of a certificate (commuting check rows, uncorrectable listed sets, symmetry, closure, orbit representatives, case-split cover) and derives completeness, the distance and the exact leading layer sum (`symmetric_certificate`, `plain_certificate`). Module `UnionEstimator` gives the mean and the second-moment bound of the union estimator, and `ExactTable` a verified exhaustive enumeration, used for the reference tables of the seven codes of at most 25 qubits; the 32-qubit table is computed by our Python code only. The computations themselves and the sampling errors are numerical experiments and are not formalized.

References

[1] Dimitris Achlioptas and Panos Theodoropoulos. Model counting with error-correcting codes. *Constraints*, 24(2):162–182, 2019.

[2] Nitzan Adler and Yuval Cassuto. Burst-erasure correcting codes with optimal average delay. *IEEE Transactions on Information Theory*, 63(5):2848–2865, 2017.

[3] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, and Cynthia Vinzant. Log-concave polynomials II: High-dimensional walks and an FPRAS for counting bases of a matroid. *Annals of Mathematics*, 199(1):259–299, 2024. STOC 2019; arXiv:1811.01816.

[4] Alexei Ashikhmin. Fidelity of finite length quantum codes in qubit erasure channel. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 1880–1885. IEEE, 2020.

[5] Andrea Asperti and Wilmer Ricciotti. A formalization of multi-tape Turing machines. *Theoretical Computer Science*, 603:23–42, 2015.

[6] Rehan Abdul Aziz, Geoffrey Chu, Christian Muise, and Peter J. Stuckey. #∃SAT: Projected model counting. In *Theory and Applications of Satisfiability Testing (SAT 2015)*, LNCS, 2015. arXiv:1507.07648.

[7] Frank J. Balbach. The Cook–Levin theorem. Archive of Formal Proofs, https://isa-afp.org/entries/Cook_Levin.html, 2023.

[8] Gefen Baranes, Madelyn Cain, J. Pablo Bonilla Ataides, Dolev Bluvstein, Josiah Sinclair, Vladan Vuletić, Hengyun Zhou, and Mikhail D. Lukin. Leveraging qubit loss detection in fault-tolerant quantum algorithms. *Physical Review X*, 16:011002, 2026. arXiv:2502.20558.

[9] Charles H. Bennett, David P. DiVincenzo, and John A. Smolin. Capacities of quantum erasure channels. *Physical Review Letters*, 78:3217–3220, 1997. arXiv:quant-ph/9701015.

[10] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[11] Michael E. Beverland, Malcolm Carroll, Andrew W. Cross, and Theodore J. Yoder. Fail fast: Techniques to probe rare events in quantum error correction, 2025. arXiv:2511.15177.

[12] Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. CaDiCaL 2.0. In *Computer Aided Verification – CAV 2024, Part I*, volume 14681 of *LNCS*, pages 133–152. Springer, 2024.

[13] H. Bombin and M. A. Martin-Delgado. Optimal resources for topological two-dimensional stabilizer codes: Comparative study. *Physical Review A*, 76:012305, 2007.

[14] Sergey Bravyi, Andrew W. Cross, Jay M. Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J. Yoder. High-threshold and low-overhead fault-tolerant quantum memory. *Nature*, 627:778–782, 2024.

[15] Sergey Bravyi and Barbara Terhal. A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes. *New Journal of Physics*, 11:043029, 2009. arXiv:0810.1983.

[16] Sergey Bravyi and Alexander Vargo. Simulation of rare events in quantum error correction. *Physical Review A*, 88:062308, 2013. arXiv:1308.6270.

[17] Martijn Brehm, Yuval Ishai, and Nicolas Resch. Fast bounded-independence functions and their duals. In Yevgeniy Dodis, editor, *7th Conference on Information-Theoretic Cryptography (ITC 2026)*, volume 385 of *LIPIcs*, pages 8:1–8:23, 2026. arXiv:2606.07009.

[18] Martijn Brehm and Nicolas Resch. Linear time encodable binary code achieving GV bound with linear time encodable dual achieving GV bound, 2025. arXiv:2509.07639.

[19] Thomas Britz. MacWilliams identities and matroid polynomials. *Electronic Journal of Combinatorics*, 9(1):R19, 2002.

[20] Chris Calabro, Russell Impagliazzo, Valentine Kabanets, and Ramamohan Paturi. The complexity of unique k-SAT: An isolation lemma for k-CNFs. In *Proceedings of the 18th IEEE Conference on Computational Complexity (CCC)*, 2003.

[21] ChunJun Cao, Michael J. Gullans, Brad Lackey, and Zitao Wang. Quantum Lego expansion pack: Enumerators from tensor networks. *PRX Quantum*, 5:030313, 2024. arXiv:2308.05152.

[22] ChunJun Cao and Brad Lackey. Quantum Lego: Building quantum error correction codes from tensor networks. *PRX Quantum*, 3:020332, 2022. arXiv:2109.08158.

[23] Mario Carneiro. Formalizing computability theory via partial recursive functions. In *10th International Conference on Interactive Theorem Proving (ITP 2019)*, volume 141 of *LIPIcs*, pages 12:1–12:17, 2019. arXiv:1810.08380.

[24] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[25] Ruoxu Cen, William He, Jason Li, and Debmalya Panigrahi. Beyond the quadratic time barrier for network unreliability. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1542–1567. SIAM, 2024. arXiv:2304.06552.

[26] Ruoxu Cen, Jason Li, and Debmalya Panigrahi. Network unreliability in almost-linear time. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC 2025)*, pages 120–131. ACM, 2025. arXiv:2503.23526.

[27] Supratik Chakraborty, Kuldeep S. Meel, and Moshe Y. Vardi. A scalable approximate model counter. In *Principles and Practice of Constraint Programming (CP 2013)*, volume 8124 of *LNCS*, pages 200–216, 2013.

[28] Supratik Chakraborty, Kuldeep S. Meel, and Moshe Y. Vardi. Algorithmic improvements in approximate counting for probabilistic inference: From linear to logarithmic SAT calls. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3569–3576, 2016.

[29] Chao Chen and Daniel Freedman. Hardness results for homology localization. *Discrete & Computational Geometry*, 45(3):425–448, 2011. Conference version: SODA 2010, pp. 1594–1604.

[30] Victor Chernozhukov, Iván Fernández-Val, and Alfred Galichon. Improving point and interval estimators of monotone functions by rearrangement. *Biometrika*, 96(3):559–575, 2009.

[31] Claude (Anthropic). A $2^{o(n)}$ time algorithm for 3-SAT. Hexagon preprint 2610.00577, https://hexagonmath.org/abs/2610.00577, October 2026. v1, 27 Oct 2026. Lean 4 formalization: https://github.com/anthropics/formal-math/tree/main/eth.

[32] Charles J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, 1987.

[33] Luis Colmenarez, Seyong Kim, and Markus Müller. Fundamental thresholds for computational and erasure errors via the coherent information. *PRX Quantum*, 6:040327, 2025. arXiv:2412.16727.

[34] Nicholas Connolly, Vivien Londe, Anthony Leverrier, and Nicolas Delfosse. Fast erasure decoder for hypergraph product codes. *Quantum*, 8:1450, 2024. arXiv:2208.01002.

[35] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 1971.

[36] Luís Cruz-Filipe, Marijn J. H. Heule, Warren A. Hunt, Jr., Matt Kaufmann, and Peter Schneider-Kamp. Efficient certified RAT verification. In *Automated Deduction – CADE 26*, volume 10395 of *LNCS*, pages 220–236. Springer, 2017.

[37] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, volume 12699 of *LNCS*, pages 625–635, 2021.

[38] Frederick Dehmel and Shilun Li. A symplectic proof of the quantum Singleton bound, 2026. arXiv:2602.20186.

[39] Nicolas Delfosse and Gilles Zémor. Upper bounds on the rate of low density stabilizer codes for the quantum erasure channel. *Quantum Information & Computation*, 13(9–10):793–826, 2013. arXiv:1205.7036.

[40] Nicolas Delfosse and Gilles Zémor. Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel. *Physical Review Research*, 2:033042, 2020. arXiv:1703.01517.

[41] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Transactions on Algorithms*, 10(4):21:1–21:32, 2014.

[42] Holger Dell and John Lapinskas. Fine-grained reductions from approximate counting to decision. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 281–288, 2018. Journal version: ACM Trans. Comput. Theory 13(2), article 8, 2021, doi:10.1145/3442352; arXiv:1707.04609.

[43] Holger Dell, John Lapinskas, and Kitty Meeks. Approximately counting and sampling small witnesses using a colorful decision oracle. *SIAM Journal on Computing*, 51(4):849–899, 2022.

[44] Fabio Dias and Yvens Lima. Limits and constants of OpenAI's claimed ETH refutation, 2026. arXiv:2611.00517.

[45] Erez Druk and Yuval Ishai. Linear-time encodable codes meeting the Gilbert–Varshamov bound and their cryptographic applications. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2014.

[46] Mattias Ehatamm, Yi Lee, Xiaodi Wu, and Runzhou Tao. End-to-end formalization of quantum error correction, 2026. arXiv:2605.16523.

[47] Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Low-density parity constraints for hashing-based discrete integration. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

[48] Weiming Feng, Yucheng Fu, and Heng Guo. Approximating two-terminal network reliability, 2026. arXiv:2608.02523.

[49] Weiming Feng and Heng Guo. An FPRAS for two terminal reliability in directed acyclic graphs. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *LIPIcs*, pages 62:1–62:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. arXiv:2310.00938.

[50] Alex Fischer and Akimasa Miyake. Hardness results for decoding the surface code with Pauli noise. *Quantum*, 8:1511, 2024.

[51] Andrew S. Fletcher, Peter W. Shor, and Moe Z. Win. Structured near-optimal channel-adapted quantum error correction. *Physical Review A*, 77(1):012320, 2008.

[52] Yannick Forster, Fabian Kunze, and Marc Roth. The weak call-by-value λ-calculus is reasonable for both time and space. *Proceedings of the ACM on Programming Languages*, 4(POPL):27:1–27:23, 2020.

[53] Lance Fortnow. The exponential time refutation. Computational Complexity blog, https://blog.computationalcomplexity.org/2026/10/the-exponential-time-refutation.html, October 2026. Posted 28 Oct 2026.

[54] Bruno Costa Alves Freire, François-Marie Le Régent, and Anthony Leverrier. Quantum Maxwell erasure decoder for qLDPC codes, 2026. arXiv:2601.10713.

[55] Lennard Gäher and Fabian Kunze. Mechanising complexity theory: The Cook–Levin theorem in Coq. In *12th International Conference on Interactive Theorem Proving (ITP 2021)*, volume 193 of *LIPIcs*, pages 20:1–20:18, 2021.

[56] Alexandru Gheorghiu and Aparna Gupte. Exact classical sampling of quantum circuits via subexponential-time #SAT. Result announced 3 Nov 2026 in the discussion thread "New Results Related to ETH", https://ethdiscussion.blog/2026/11/01/announce-your-results/, 2026. Descriptive title; no preprint located at the time of writing.

[57] Leslie Ann Goldberg and Mark Jerrum. Approximating the Tutte polynomial of a binary matroid and other related combinatorial polynomials. *Journal of Computer and System Sciences*, 79(1):68–78, 2013. arXiv:1006.5234.

[58] Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Model counting: A new strategy for obtaining good bounds. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, pages 54–61. AAAI Press, 2006.

[59] Markus Grassl, Thomas Beth, and Thomas Pellizzari. Codes for the quantum erasure channel. *Physical Review A*, 56(1):33–38, 1997. arXiv:quant-ph/9610042.

[60] Elena Grigorescu, Vatsal Jha, and Eric Samperton. On the hardness of approximating distances of quantum codes. In *45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2025)*, volume 360 of *LIPIcs*, pages 34:1–34:18, 2025. arXiv:2509.21469.

[61] Michael J. Gullans, Stefan Krastanov, David A. Huse, Liang Jiang, and Steven T. Flammia. Quantum coding with low-depth random circuits. *Physical Review X*, 11:031066, 2021. arXiv:2010.09775.

[62] Heng Guo and Mark Jerrum. A polynomial-time approximation algorithm for all-terminal network reliability. *SIAM Journal on Computing*, 48(3):964–978, 2019. ICALP 2018 best paper.

[63] Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005.

[64] David G. Harris and Aravind Srinivasan. Improved bounds and algorithms for graph cuts and network reliability. *Random Structures & Algorithms*, 52(1):74–135, 2018. Preliminary version in Proc. SODA 2014, pp. 259–278; arXiv:1602.08730.

[65] Sascha Heußen, Don Winter, Manuel Rispler, and Markus Müller. Dynamical subset sampling of quantum error-correcting protocols. *Physical Review Research*, 6:013177, 2024. arXiv:2309.12774.

[66] Min-Hsiu Hsieh and François Le Gall. NP-hardness of decoding quantum error-correction codes. *Physical Review A*, 83:052331, 2011. arXiv:1009.1319.

[67] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *Theory and Applications of Satisfiability Testing – SAT 2018*, volume 10929 of *LNCS*, pages 428–437. Springer, 2018.

[68] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[69] Russell Impagliazzo, Ramamohan Paturi, and Ryan Williams. ETH refutation: Examination and discussion. Blog, https://ethdiscussion.blog/2026/10/26/eth-refutation-examination-and-discussion/, October 2026. Launched 26 Oct 2026; states that the examiners' involvement is not a certification of the claim.

[70] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[71] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008)*, pages 433–442. ACM, 2008.

[72] Pavithran Iyer and David Poulin. Hardness of decoding quantum stabilizer codes. *IEEE Transactions on Information Theory*, 61(9):5209–5223, 2015. arXiv:1310.3235.

[73] F. Jaeger, D. L. Vertigan, and D. J. A. Welsh. On the computational complexity of the Jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, 108(1):35–53, 1990.

[74] Mark R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43(2–3):169–188, 1986.

[75] Chenyuan Jia, Qingqing Peng, Ke Liu, Guanghui Wang, and Guiying Yan. On the intractability of the minimum distance problem for regular LDPC codes, 2026. arXiv:2606.23161.

[76] Gleb Kalachev and Sergey Sadov. On the cleaning lemma of quantum coding theory. *Linear Algebra and its Applications*, 649:96–121, 2022. arXiv:2204.04699.

[77] Mingyu Kang, Wesley C. Campbell, and Kenneth R. Brown. Quantum error correction with metastable states of trapped ions using erasure conversion. *PRX Quantum*, 4:020358, 2023.

[78] Upendra Kapshikar. Hardness of approximating quantum code distance beyond $\sqrt{N}$, 2026. Submitted 27 Sep 2026; arXiv:2609.33956.

[79] Upendra Kapshikar and Srijita Kundu. On the hardness of the minimum distance problem of quantum codes. *IEEE Transactions on Information Theory*, 69(10), 2023. arXiv:2203.04262.

[80] David R. Karger. A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM Journal on Computing*, 29(2):492–514, 1999.

[81] David R. Karger. A phase transition and a quadratic time unbiased estimator for network reliability. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2020)*, pages 485–495. ACM, 2020.

[82] Richard M. Karp, Michael Luby, and Neal Madras. Monte-Carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 10(3):429–448, 1989.

[83] Shiva Kintali. A simplified proof of ETH is false. https://shivakintali.github.io/papers/ETH.pdf, October 2026. Dated 26 Oct 2026.

[84] A. Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.

[85] Aleksander Kubica, Arbel Haim, Yotam Vaknin, Harry Levine, Fernando Brandão, and Alex Retzker. Erasure qubits: Overcoming the T_1 limit in superconducting circuits. *Physical Review X*, 13:041022, 2023.

[86] Kao-Yueh Kuo and Chung-Chin Lu. On the hardnesses of several quantum decoding problems. *Quantum Information Processing*, 19:123, 2020. arXiv:1306.5173.

[87] Kao-Yueh Kuo and Yingkai Ouyang. Degenerate quantum erasure decoding. *npj Quantum Information*, 12:75, 2026. arXiv:2411.13509.

[88] Ruhollah Majdoddin. Lowered crossover points of Anthropic's subexponential 3-SAT algorithm, 2026. arXiv:2610.41622.

[89] Avijit Mandal, Mert Gökduman, and Henry D. Pfister. Area theorems for quantum error correction, 2026. Submitted 15 Oct 2026; arXiv:2610.22710.

[90] Yishay Mansour, Noam Nisan, and Prasoon Tiwari. The computational complexity of universal hashing. *Theoretical Computer Science*, 107(1):121–133, 1993.

[91] Carolyn Mayer, Anand Ganti, Uzoma Onunkwo, Tzvetan Metodi, Benjamin Anker, and Jacek Skryzalin. Rare event simulation of quantum error-correcting circuits. *Quantum*, 10:2237, 2026. arXiv:2509.13678.

[92] Kuldeep S. Meel and S. Akshay. Sparse hashing for scalable approximate model counting: Theory and practice. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 728–741, 2020.

[93] E. F. Moore and C. E. Shannon. Reliable circuits using less reliable relays. *Journal of the Franklin Institute*, 262:191–208 and 281–297, 1956. Parts I and II.

[94] Ryan O'Donnell. Some exposition of "The Exponential Time Hypothesis Is False". https://www.cs.cmu.edu/~odonnell/some_eth_false_exposition.pdf, October 2026. Dated 30 Oct 2026.

[95] OpenAI. The exponential time hypothesis is false. https://github.com/openai/math, October 2026. Preprint dated 17 Oct 2026, revised 25 Oct 2026. GitHub repository openai/math, preprints/The-Exponential-Time-Hypothesis-is-False-October-25-2026/paper.pdf; Lean scope lean/docs/378.md.

[96] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical LDPC codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2022)*, pages 375–388. ACM, 2022. arXiv:2111.03654.

[97] Hugo Perrin, Sven Jandura, and Guido Pupillo. Quantum error correction resilient against atom loss. *Quantum*, 9:1884, 2025. arXiv:2412.07841.

[98] Hugo Perrin, Gatien Roger, and Guido Pupillo. Correlated atom loss as a resource for quantum error correction, 2026. arXiv:2603.24237.

[99] J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.

[100] Eric M. Rains. Quantum weight enumerators. *IEEE Transactions on Information Theory*, 44(4):1388–1394, 1998. arXiv:quant-ph/9612015.

[101] Tim Robertson, F. T. Wright, and R. L. Dykstra. *Order Restricted Statistical Inference*. Wiley, Chichester, 1988.

[102] Kaavya Sahay, Junlan Jin, Jahan Claes, Jeff D. Thompson, and Shruti Puri. High-threshold codes for neutral-atom qubits with biased erasure errors. *Physical Review X*, 13:041013, 2023.

[103] Pradeep Sarvepalli. Quantum codes and symplectic matroids, 2011. arXiv:1104.1171.

[104] A. J. Scott. Probabilities of failure for quantum error correction. *Quantum Information Processing*, 4:399–431, 2005. arXiv:quant-ph/0406063.

[105] Pierre Senellart and Anton Gnatenko. Descriptive complexity in Lean: Completeness by first-order reductions, 2026. arXiv:2609.18261.

[106] Sayam Sethi, Aditi Awasthi, Maxwell Poster, Joshua Viszlai, and Jonathan Mark Baker. Staying sober: A shot-free predictor of logical error rates for concatenated quantum error correcting codes, 2026. arXiv:2610.06606.

[107] Peter W. Shor and Raymond Laflamme. Quantum analog of the MacWilliams identities for classical coding theory. *Physical Review Letters*, 78:1600–1602, 1997. arXiv:quant-ph/9610040.

[108] Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 330–335, 1983.

[109] Mate Soos and Kuldeep S. Meel. BIRD: Engineering an efficient CNF-XOR SAT solver and its applications to approximate model counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1592–1599, 2019.

[110] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In *Theory and Applications of Satisfiability Testing – SAT 2009*, volume 5584 of *LNCS*, pages 244–257. Springer, 2009.

[111] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996.

[112] Thomas M. Stace, Sean D. Barrett, and Andrew C. Doherty. Thresholds for topological codes in the presence of loss. *Physical Review Letters*, 102:200501, 2009. arXiv:0904.3556.

[113] Larry J. Stockmeyer. On approximation algorithms for #p. *SIAM Journal on Computing*, 14(4):849–861, 1985.

[114] Yong Kiam Tan, Jiong Yang, Mate Soos, Magnus O. Myreen, and Kuldeep S. Meel. Formally certified approximate model counting. In Arie Gurfinkel and Vijay Ganesh, editors, *Computer Aided Verification (CAV 2024)*, volume 14681 of *Lecture Notes in Computer Science*, pages 153–177. Springer, 2024. arXiv:2406.11414.

[115] The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020)*, pages 367–381, 2020.

[116] Marc Thurley. An approximation algorithm for #k-SAT. In *29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 14 of *LIPIcs*, pages 78–87, 2012.

[117] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[118] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[119] Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, 1997.

[120] Leon Wichette, Hans Hohenfeld, Elie Mounzer, and Linnea Grans-Samuelsson. A partition function framework for estimating logical error curves in stabilizer codes. *Quantum*, 10:2175, 2026. arXiv:2505.15758.

[121] Yue Wu, Shimon Kolkowitz, Shruti Puri, and Jeff D. Thompson. Erasure conversion for fault-tolerant quantum computing in alkaline earth Rydberg atom arrays. *Nature Communications*, 13:4657, 2022.

[122] Jian Xu, Xingyuan Zhang, and Christian Urban. Mechanising Turing machines and computability theory in Isabelle/HOL. In *Interactive Theorem Proving (ITP 2013)*, volume 7998 of *LNCS*, pages 147–162, 2013.

[123] Shengjia Zhao, Sorathan Chaturapruek, Ashish Sabharwal, and Stefano Ermon. Closing the gap between short and long XORs for model counting. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 3322–3329, 2016. arXiv:1512.08863.