



Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/239873/>

Version: Published Version

Article:

Kuo, K.-Y. and Ouyang, Y. (2026) Degenerate quantum erasure decoding. npj Quantum Information, 12. 75. ISSN: 2056-6387

<https://doi.org/10.1038/s41534-026-01212-3>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



Degenerate quantum erasure decoding

Kao-Yueh Kuo^{1,2}✉ & Yingkai Ouyang¹✉

Erasures are the primary type of errors in physical systems dominated by leakage errors. While quantum error correction (QEC) using stabilizer codes can combat erasure errors, it remains unknown which constructions achieve capacity performance. If such codes exist, decoders with linear runtime in the code length are also desired. In this paper, we present erasure capacity-achieving quantum codes under maximum-likelihood decoding (MLD), though MLD requires cubic runtime in the code length. For QEC, using an accurate decoder with the shortest possible runtime will minimize the degradation of quantum information while awaiting the decoder's decision. To address this, we propose belief propagation (BP) decoders that run in linear time and exploit error degeneracy in stabilizer codes, achieving capacity or near-capacity performance for a broad class of codes, including bicycle codes, product codes, and topological codes. We furthermore explore the potential of our BP decoders to handle mixed erasure and depolarizing errors, and also local deletion errors via concatenation with permutation invariant codes.

Erasure conversion^{1–10} is a recent technique that converts leakage errors into erasures^{11–13}. In systems dominated by leakage errors, such as ultracold Rydberg atoms^{1–5}, superconducting circuits^{6–8}, trapped ions⁹, or photonic systems¹⁰, erasure conversion ensures that the majority of errors are erasures, by paying necessary physical resource or time for the conversion. In the context of quantum error correction (QEC), the promised structure of errors affords potentially better error thresholds and decoding efficiency compared to unstructured errors. We aim to explore optimal QEC performance in erasure-dominated systems.

The rate of a QEC code is the ratio of the number of encoded qubits k to the number of physical qubits n . The quantum erasure channel capacity, $C(p) = 1 - 2p$ ^{12,13}, represents the maximum rate (k/n) for which a decoder can almost surely correct typical erasures, where p is the independent erasure probability per qubit. This capacity quantifies the highest rate of reliable quantum information transmission over a noisy erasure channel. A central question in quantum information theory is: how closely can practical QEC codes approach $C(p)$ with low-complexity decoders?

This question is particularly pertinent for quantum stabilizer codes with code parameters $[[n, k]]$ and low-weight commutative Pauli operators, known as quantum low-density parity-check (QLDPC) codes. Two outstanding open problems in this area are: (1) identifying stabilizer code constructions that achieve capacity $r = 1 - 2p$ for arbitrary rates $r = k/n$, and (2) developing decoders that enable these constructions to achieve near-capacity performance with linear runtime complexity. This paper solves both of these problems.

Previously, only two-dimensional (2D) topological codes, such as surface or toric codes, were known to achieve erasure capacity^{14,15} with linear decoding time¹⁶, but their code rate vanishes ($r \rightarrow 0$) as n increases.

A recent breakthrough by IBM significantly reduce qubit overhead ($1 - \frac{k}{n}$) compared to surface codes, using non-zero rate QLDPC codes—specifically bivariate bicycle codes—highlighting their non-vanishing rate and potential for nearly 2D layouts¹⁷. Bicycle-type codes with hundreds of qubits achieve comparable performance to surface codes with thousands, protecting a similar number of information qubits, offering a 10 times reduction in the number of physical qubits.

A further significant reduction in qubit overhead can be achieved using higher-rate codes, with long code lengths ensuring the correction of a large fraction of errors, which we include in this work. First, we show that the bicycle construction¹⁸ achieves capacity for moderate-to-high rate codes on the quantum erasure channel. Second, for low-rate (but non-zero rate) cases, product constructions with flexible rate, such as lifted-product (LP) codes¹⁹, also achieve erasure capacity. Third, to facilitate practical applications, we provide linear-time decoders universal for these codes, also supporting vanishing-rate 2D topological codes.

For erasures, the task of decoding (inferring the error) reduces to solving a system of linear equations (cf. (3)–(7)). The optimal decoder is the maximum-likelihood decoder (MLD), which is often implemented using a Gaussian decoder. This approach employs Gaussian elimination with a complexity of $O(n^3)$, or $O(n^{2.373})$ ^{20,21}.

In classical erasure error-correction, a binary bit-flipping belief propagation (Flip-BP₂) decoder is preferred over MLD for classical LDPC codes due to its linear runtime $O(n)$ and near-capacity performance^{22–28}. The success of BP is evident in 5G networks^{29–31}. However, naively applying Flip-BP₂ to QLDPC codes is problematic due to inevitable small stopping sets in the quantum case (see Supplementary Note 3 for more details). Small

¹School of Mathematical and Physical Sciences, University of Sheffield, Sheffield, UK. ²Department of Electrical Engineering, Yuan Ze University, Taoyuan City, Taiwan. ✉e-mail: kywukuo@saturn.yzu.edu.tw; y.ouyang@sheffield.ac.uk

stopping sets obstruct BP convergence^{24–28,32,33} and are avoided in the classical code design^{25–28}.

In the quantum case, various techniques have been proposed for decoding QLDPC codes under erasures. First, the peeling decoder was proposed to decode 2D topological codes, including peeling on toric or surface codes (in linear time)¹⁶ or extended peeling on color codes (with additional complexity)^{34–36}. (The peeling decoder includes a spanning-tree step³⁷, followed by a peeling step resembling a Flip-BP₂ in²³, as pointed in ref. 38). Second, for product codes, the pruned peeling (improved over peeling decoder) and a Vertical-Horizontal (VH) post-processing were proposed³⁸. The “pruned peeling + VH” has complexity $O(n^2)$, or $O(n^{1.5})$ for a probabilistic version. Third, modifications of minimum-weight perfect matching (MWPM) were showed good erasure decoding accuracy for 2D codes^{9,39}, though without linear time⁴⁰. Recently, BP was improved with guided decimation⁴¹, and cluster decomposition (extending the VH technique) was proposed in ref. 42 to balance accuracy and complexity between MLD and VH. Both works^{41,42} focused on product codes.

In this paper, we study a broad class of QLDPC codes, including 2D topological codes^{43–46} and their variants such as XZZX codes^{47–49}, as well as bicycle codes¹⁸, hypergraph-product (HP) codes⁵⁰, generalized HP (GHP) codes⁵¹, and LP codes¹⁹. These are CSS-type codes^{52,53}, except for the XZZX codes. General QLDPC codes admit low-complexity quantum circuits and offer the benefits of (1) enabling error-corrected quantum computation with constant encoding rates^{54,55} and (2) enhancing quantum network communication rates⁵⁶.

In our approach, we define *erasure bounded-distance decoding* (eBDD) (Def. 4) as a metric for evaluating decoding accuracy. We estimate the error threshold p_{th} of a code family by checking whether a constant correction radius t/n exists as n increases, where t is the expected number of corrected erasures. If so, we set $p_{th} = t/n$. For clarity, we may denote $p_{th}^{(BP)}$ for BP and $p_{th}^{(MLD)}$ for MLD if they differ. We emphasize that the threshold values in this paper apply to independent erasures; other error types, such as depolarizing errors, yield different thresholds.

Quantum decoding relies on exploiting degeneracy, enabling a decoder to converge to any (degenerate) error equivalent to the occurred error up to a stabilizer, resulting in remarkable accuracy^{57–59}. Our constructed codes achieve capacity performance with $r = 1 - 2p_{th}^{(MLD)}$ using the Gaussian decoder (MLD), which always finds a solution likely to be a degenerate error, at the cost of $O(n^3)$ complexity. To achieve linear runtime $O(n)$, we first propose a gradient-descent Flip-BP₂ (GD Flip-BP₂) to address stopping sets via a simple GD scheme (which is supported by the symmetry of degenerate errors; see ref. 57 or Supplementary Note 4 for more discussions). Next, we introduce a soft binary BP with memory effects (MBP₂), its adaptive version (AMBP₂), and their quaternary counterparts (A)MBP₄. A soft BP decoder mimics a soft GD optimization, providing strong convergence to degenerate solutions. For our constructed codes, we achieve near-capacity performance with $r = 1 - 2.5p_{th}^{(BP)}$ using (A)MBP or GD Flip-BP₂ for non-zero rate codes, with AMBP providing very strong accuracy at low error rates. In addition, AMBP₄ achieves the optimal threshold of 0.5 for vanishing-rate 2D topological codes.

The success of (A)MBP stems from multiple factors. First, we apply message softization (15) to have non-extreme and non-zero BP message values, enabling updates within stopping sets. Second, we link BP to gradient descent to determine a proper update step-size and, when necessary, adjust the prior error distribution to ensure sufficient update gradients, further mitigating harmful stopping sets. Third, we extend binary decoders to quaternary ones, leveraging correlations between the X -portion and Z -portion of a QLDPC check matrix to prevent BP from being halted by stopping sets in one portion. Moreover, inspired by Hopfield nets^{60–62}, (A)MBP uses fixed inhibition (see Supplementary Note 5 (part b)), adding memory effects to BP to improve convergence. Additionally, we employ a group random schedule for message updates to improve convergence, similar to a serial schedule⁶³, while processing multiple messages in parallel (see Supplementary Note 8). Finally, we propose a strategy to make AMBP as efficient as MBP to facilitate practical applications.

We also explore the potential of our decoders to manage deletions of untracked particle losses (with the assistance of permutation-invariant (PI) codes^{64,65}) or mixed erasure and depolarizing errors.

Results

Figure 1 compares the decoding complexity of various decoders. BPs developed in this paper have accuracy comparable to the Gaussian decoder (MLD), but with linear complexity. The thresholds achieved by BP and MLD are analysed and compared with the channel capacity in Fig. 2. Physical fidelity decays exponentially over time (characterized by the decoherence time^{3,66}). We plot the erasure error rate p on a log scale to depict the exponential effort to maintain coherence. In Fig. 2, except for the 2D topological codes, each code family is represented by two data points, indicating the thresholds achieved by BP and MLD. There is a small gap between the accuracy of BP (linear-time decoder) and MLD (non-linear time decoder). The each code family has nearly fixed row and column weights to ensure linear BP complexity (see Supplementary Note 7 (part b)).

In Fig. 2, we also plot rate upper bounds estimated by [ref. 13, Theorem 3.8], which depend on stabilizer weight w . While not necessarily tight, these bounds align with our findings in that higher code rates require higher w .

To obtain a precise estimate of the rate currently achieved via BP at erasure probability p , we replot Fig. 2 on a linear scale in Supplementary Note 7 (part f). By interpolation, the rate achieved by BP is $r = 1 - 2.5p - 2p^2 + 6p^3$.

In Table 1, we compare our results with other linear-time or MLD decoders in the literature^{16,38,41,42}.

Quantum erasure channel

Erasures occur when certain qubits are lost, with a detector identifying their locations. A lost qubit is represented by an “erasure state” $|2\rangle$, orthogonal to the computational basis states $|0\rangle$ and $|1\rangle$. Thus, we consider^{12,13}:

Definition 1. A quantum erasure channel independently erases each qubit ρ_1 with probability $p \leq 0.5$, leaving it unchanged otherwise, resulting in the following error process:

$$\rho_1 \mapsto (1-p)\rho_1 + p|2\rangle\langle 2|, \quad (1)$$

with the location of the erased qubit known.

Assume that an $[[n, k]]$ stabilizer code is used. For decoding, we insert an ancilla qubit at each lost qubit’s location, creating a noisy codeword state of length n for stabilizer measurements. After measurements, each ancilla qubit collapses randomly, encountering Pauli I , X , Y , or Z with equal probability $1/4$. Thus, an erased qubit (with its location known) encounters the following error before decoding:

$$\rho_1 \mapsto (1-p)\rho_1 + p\frac{I}{2}, \quad (2)$$

where $\frac{I}{2} = \sum_{W \in \{I, X, Y, Z\}} \frac{1}{4} W \rho_1 W^\dagger$. The error process (2) is similar to the depolarizing channel, but with a key distinction: we know which qubits are completely depolarized.

Let $\mathbb{r} = \{1, 2, \dots, n\}$ be the set of *erased coordinates*. If $j \in \mathbb{r}$, then qubit j is erased. The complement, $\bar{\mathbb{r}} = \{1, 2, \dots, n\} \setminus \mathbb{r}$, contains the *non-erased coordinates*.

Let $E = E_1 \otimes \dots \otimes E_n$ be the n -fold Pauli error (up to global phase) after measurements, where $E_j = I, X, Y$, or Z with equal probability $\frac{1}{4}$ if $j \in \mathbb{r}$, and $E_j = I$ if $j \in \bar{\mathbb{r}}$.

A stabilizer decoder infers E from $\mathbb{r}$ and the *syndrome* of E obtained through measurements, as detailed below.

Erasure decoding problem in stabilizer codes

We refer to^{67,68} for background on stabilizer codes. For decoding, we represent the measured stabilizers $H^{(1)}, \dots, H^{(n-k)}$ as a (*quaternary*) Pauli

check matrix

$$H = \begin{bmatrix} H^{(1)} \\ \vdots \\ H^{(n-k)} \end{bmatrix} \in \{I, X, Y, Z\}^{(n-k) \times n}. \quad (3)$$

To have a linear decoding problem, we map I, X, Y, Z to $(0|0), (1|0), (1|1), (0|1)$, respectively. Conventionally, this defines a map $\varphi: \{I, X, Y, Z\}^n \rightarrow \{0, 1\}^{2n}$, and we have a $(n-k) \times 2n$ binary check matrix

$$H = \begin{bmatrix} \varphi(H^{(1)}) \\ \vdots \\ \varphi(H^{(n-k)}) \end{bmatrix} = \begin{bmatrix} H_1 \\ \vdots \\ H_{n-k} \end{bmatrix} = \begin{bmatrix} H_{1,1} & \cdots & H_{1,2n} \\ \vdots & \ddots & \vdots \\ H_{n-k,1} & \cdots & H_{n-k,2n} \end{bmatrix} \quad (4)$$

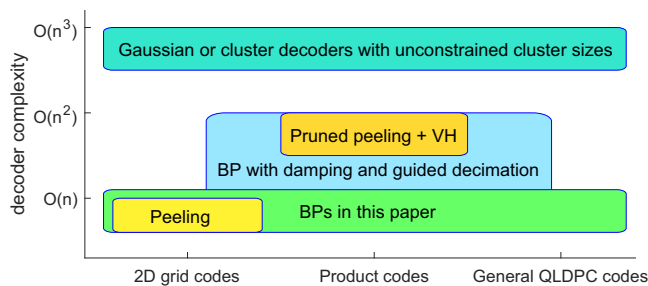
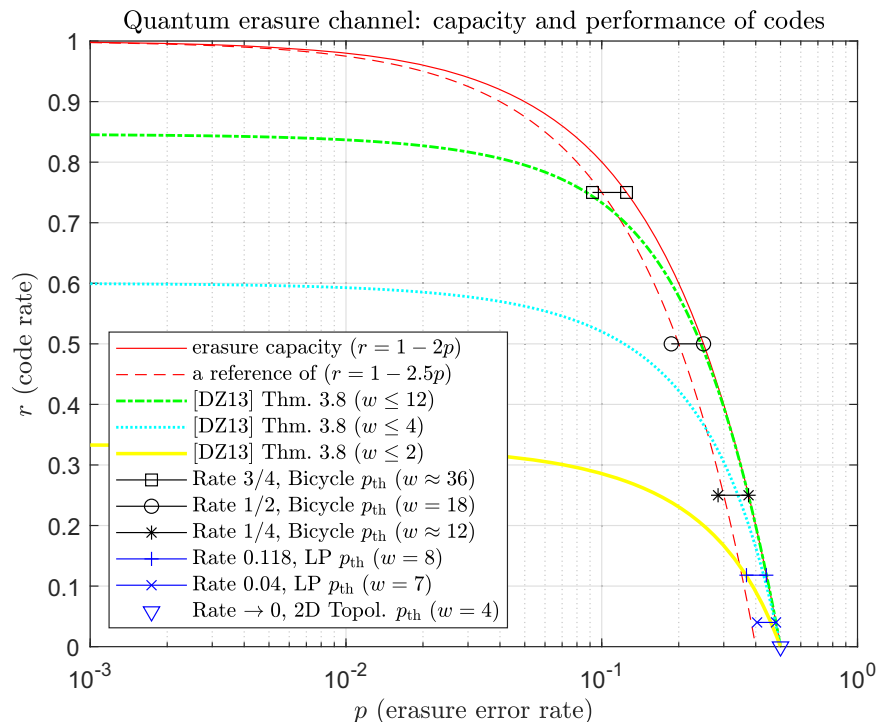


Fig. 1 | Comparison of decoders. The vertical axis indicates complexity, while the horizontal axis shows code types with increasing connectivity from left to right. BPs in this paper perform accurately across various code types, similar to the Gaussian decoder, with a slight threshold gap but with linear complexity. The achieved thresholds are shown in Fig. 2.

Fig. 2 | Quantum erasure capacity and thresholds of codes. For 2D topological codes, BP and MLD share a threshold of $p_{th} = 0.5$. For other codes, a small accuracy gap is indicated by two points: $p_{th}^{(BP)}$ (left) and $p_{th}^{(MLD)}$ (right). The constructed codes achieve capacity with MLD, $r = 1 - 2p_{th}^{(MLD)}$, and near-capacity with BP, $r = 1 - 2.5p_{th}^{(BP)}$. We summarize our decoders and tabulate their accuracy figures in Table 1. For bicycle codes, BP thresholds are achieved using GD Flip-BP₂ or (A)MBP_q. For LP codes, BP thresholds are achieved using AMBP_q and GD Flip-BP₂ shows similar accuracy for rate 0.118 LP codes. For 2D topological codes, a threshold of 0.5 is achieved using AMBP₄. Higher-rate codes require larger stabilizer weight w , consistent with the trend from¹³ (plotted as curves labeled [DZ13]).



$$= [H^X | H^Z] = \begin{bmatrix} H_{1,1}^X & \cdots & H_{1,n}^X & H_{1,1}^Z & \cdots & H_{1,n}^Z \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ H_{n-k,1}^X & \cdots & H_{n-k,n}^X & H_{n-k,1}^Z & \cdots & H_{n-k,n}^Z \end{bmatrix}. \quad (5)$$

Consider the operation under $\text{mod } 2$ for the binary field $\mathbb{F}_2 = \{0, 1\}$. For two binary matrices A and B of $2n$ columns, define a bilinear form (called *symplectic inner product*)

$$\langle A, B \rangle = AAB^T,$$

where $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $(\cdot)^T$ is the matrix transpose. Two operators $E, F \in \{I, X, Y, Z\}^n$ commute if the corresponding E, F satisfy $\langle E, F \rangle = 0$, and anticommute if $\langle E, F \rangle = 1$. We denote zero and identity matrices by $\mathbf{0}$ and $\mathbf{I}$, respectively, using the subscript n to specify dimension $n \times n$. Notice that $\langle H, H \rangle = \mathbf{0}$. The row-space of H is a linear code $C = \text{Row}(H) \subset \mathbb{F}_2^{2n}$, which is symplectic self-orthogonal. That is, $C \subseteq C^\perp$, where $C^\perp$ is the symplectic dual of C in $\mathbb{F}_2^{2n}$.

For any stabilizer codes, we have this **degeneracy**: for an error $E \in \{0, 1\}^{2n}$, its *logical coset* is $E + C$; any errors $\tilde{E} \in E + C$ are *degenerate errors* of E with an equivalent effect on the code space. Degenerate errors share the same syndrome $s = \langle E, H \rangle = \langle \tilde{E}, H \rangle$, but syndrome matched errors form a larger error set called *error coset* ($E + C^\perp$) of syndrome s .

Definition 2. (Quantum erasure decoding problem) Given a binary matrix $H \in \{0, 1\}^{m \times 2n}$, $m \geq n - k$, erasure information $\mathbb{I} \subseteq \{1, 2, \dots, n\}$, and a syndrome $s \in \{0, 1\}^m$, where $s = \langle E, H \rangle$ for some $E \in \{0, 1\}^{2n}$ drawn from the erasure channel, a decoder is asked to find an $\hat{E} \in \{0, 1\}^{2n}$ such that $\hat{E} \in E + C$ with probability as high as possible.

Given $\mathbb{I}$ and s , an erasure- and syndrome-matched solution satisfies a system of linear equations:

$$\begin{cases} (H \Lambda) E^T = [H^Z | H^X] (E^X | E^Z)^T = s^T, \\ (E_j^X | E_j^Z) = (0|0) \forall j \in \mathbb{I}. \end{cases} \quad (6)$$

Table 1 | Comparison of linear-time and MLD decoders for quantum erasure errors on data qubits

decoder	type	optimized complexity	worst case complexity	tested codes	comment
Peeling ¹⁶	spanning-tree (MLD for 2D codes)	$O(n)$ (toric/surface)	$O(nT_{\max})$	2D gird (toric/surface) ¹⁶ ; color codes need additional complexity ^{34–36}	spanning-tree step is linear-time ³⁷ ; peeling step is iterative, but guaranteed to finish within $T_{\max} = O(1)$ iterations for toric/surface codes.
Pruned peeling + VH ³⁸	spanning tree + local GE	$O(n^{1.5})$ (if use ^{14,39})	$O(n^2)$	HP	Peeling needs VH to decode HP codes but underperforms AMBP
BP with damping and guided decimation ⁴¹	soft BP + guided decimation	$O(n)$ (reduced accuracy)	$O(n^2 T_{\max})$	HP and GHP	outperform peeling+VH, although not as accurate as AMBP
Peeling + cluster decomposition ⁴²	spanning tree + tuneable GE	$O(n)$ (reduced accuracy)	$O(n^3)$ (MLD)	HP and GHP	flexible accuracy/complexity tradeoff; can beat BPs but need complexity $O(n^3)$
GD Flip-BP ₂ (this paper, Algorithms 1)	bit-flipping BP	$O(n \log \log n)$	$O(nT_{\max})$	LP (Fig. 8), bicycle (Fig. 6), HP and GHP (Fig. 11)	performs well on bicycle and rate 0.118 LP codes; extremely low complexity
(AMBP ₂ ($q = 2$ or 4) (this paper, Algorithms 2–4)	soft BP	$O(n \log \log n)$	$O(nT_{\max})$	LP (Fig. 7), toric and XZZX (Fig. 9), HP and GHP (Fig. 11), bicycle (Supplementary Note 7 (part c))	accurate and universal for QLDPC codes; also decode mixed errors with high accuracy (Fig. 5)
Gaussian elimination (GE) (see (6)–(7))	MLD	$O(n^{2.373})$ (see refs. 20,21)	$O(n^3)$	we provide the accuracy achieved by Gaussian decoder for all tested codes	MLD is difficult to simulate at low error rates, but eBDD provides good curve extrapolation

Upper portion: decoders in the literature. Lower portion: our decoders and GE. GHP codes are considered as special LP codes^{19,41,42}. We construct flexible-rate and flexible-length LP codes from random quasi-cyclic matrices and bicycle codes with random generator vectors. High-rate bicycle codes ($r = 3/4$) are particularly suitable to work with PI codes for local deletion errors (cf. Fig. 4). Bicycle codes are generalized for fault-tolerant error correction on IBM hardware implementations, emphasizing their non-zero code rate and potential for nearly 2D layouts¹⁷. Figures comparing our decoders with those in refs. 38,41,42 are provided in Supplementary Note 7 (part a).

Definition 3. A solution $\hat{E} \in \{0, 1\}^{2n}$ satisfying (6) is called a *feasible solution* for the erasure decoding.

Solving (6) can be simplified by solving

$$[H_{\mathbb{I}}^Z | H_{\mathbb{I}}^X](E_{\mathbb{I}}^X | E_{\mathbb{I}}^Z)^T = s^T, \quad (7)$$

where $A_{\mathbb{I}}$ denotes the submatrix of a matrix A restricted to the columns specified by $\mathbb{I}$. The cardinality $|\mathbb{I}| \approx np \propto n$. A solution to (6) or (7) can be obtained by Gaussian elimination with complexity $O(n^3)$. Since the syndrome s is produced from the physical error E , there always exists at least one feasible solution (e.g., E itself) to the linear equations in (6). However, the solution may not be unique, especially at high erasure error rates. The uniqueness of the solution depends on $|\mathbb{I}|$ and the code distance d (discussed in detail in Supplementary Note 2). An $[[n, k]]$ code with distance d is called an $[[n, k, d]]$ code.

Other error models, such as those with syndrome errors or circuit-level noise, can lead to non-unique solutions even at low physical error rates; these cases are beyond the scope of this paper. We assume noiseless syndrome extraction. Similar deductions (as those in Lemma 1, Theorem 2, and Supplementary Note 1) still apply under syndrome errors or circuit-level noise, given appropriate check matrices and logical generators. Decoding still proceeds based on the check matrix constructed as in refs. 69,70, but the problem becomes more challenging, as it may yield many degenerate solutions even at low physical error rates.

Paralleling [ref. 16, Lemma 1], we present the following lemma and theorem, proved by group theory and probability theory in Supplementary Note 1.

Lemma 1. If two distinct logical cosets both contain feasible solutions, then the number of feasible solutions in both cosets is equal, and they both are maximum likelihood logical cosets.

Theorem 2. For a stabilizer code with a check matrix $H \in \{0, 1\}^{m \times 2n}$, if erasures $\mathbb{I} \subseteq \{1, 2, \dots, n\}$ and syndrome $s \in \{0, 1\}^m$ are given for decoding, then any feasible solution to (6) is an optimal solution for the decoding problem in Def. 2. A decoder always outputting a feasible solution is an MLD.

Obtaining a feasible solution is the best a decoder can do for erasure decoding, although it does not always guarantee successful decoding, especially when $|\mathbb{I}|$ is large.

Example 1. Consider a $[[4, 1]]$ code with this check matrix

$$H = [H^X | H^Z] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

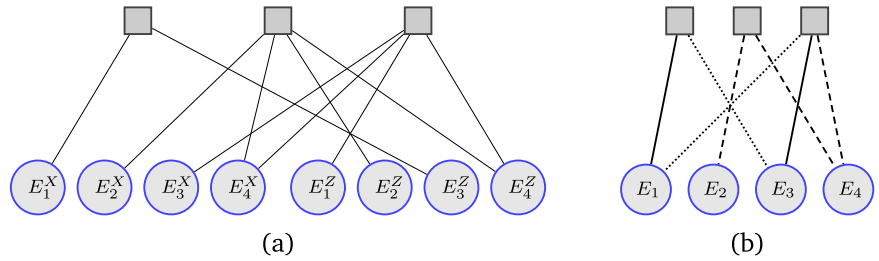
or equivalently, its Pauli check matrix

$$H = \begin{bmatrix} X & I & Z & I \\ I & Y & I & Y \\ Z & I & X & Y \end{bmatrix}. \quad (9)$$

There are $2^{2k} = 4$ logical cosets for a given syndrome s . When $s = (0 \ 1 \ 0)$, the corresponding 4 logical cosets, in Paulis, are:

$$\begin{bmatrix} I & Z & I & I & \leftarrow \\ X & Z & Z & I \\ I & X & I & Y & \leftarrow \\ X & X & Z & Y \\ Z & Z & X & Y \\ Y & Z & Y & Y \\ Z & X & X & I \\ Y & X & Y & I \end{bmatrix}, \begin{bmatrix} I & I & Z & X \\ X & I & I & X \\ I & Y & Z & Z \\ X & Y & I & Z \\ Z & I & Y & Z \\ Y & I & X & Z \\ Z & Y & Y & X \\ Y & Y & X & X \end{bmatrix}, \begin{bmatrix} I & I & Z & Z \\ X & I & I & Z \\ I & Y & Z & X \\ X & Y & I & X \\ Z & I & Y & X \\ Y & I & X & X \\ Z & Y & Y & Z \\ Y & Y & X & Z \end{bmatrix}, \begin{bmatrix} I & Z & I & Y & \leftarrow \\ X & Z & Z & Y \\ I & X & I & I & \leftarrow \\ X & X & Z & I \\ Z & Z & X & I \\ Y & Z & Y & I \\ Z & X & X & Y \\ Y & X & Y & Y \end{bmatrix},$$

Fig. 3 | Tanner graphs of Example 1. **a** Tanner graph induced by the binary matrix H in (8). **b** Tanner graph induced by the Pauli matrix H in (9), with edge types X (solid line), Y (dashed line), and Z (dotted line).



where left arrows indicate the errors to be discussed.

When only the second qubit is erased, we easily identify a unique solution $IZII$. MLD always succeeds in this error case.

When the 2nd and 4th qubits are erased, i.e., $\mathbb{r} = \{2, 4\}$, we have $E_j = I$ for $j \in \bar{\mathbb{r}} = \{1, 3\}$. As indicated by the left arrows above, the occurred error is a Pauli operator in either

$$\begin{bmatrix} I & Z & I & I \\ I & X & I & Y \end{bmatrix} \text{ or } \begin{bmatrix} I & Z & I & Y \\ I & X & I & I \end{bmatrix}. \quad (10)$$

An MLD will output a solution in one of the cosets, but the actual error may lie in the other coset, leading to a logical error rate of 1/2 in this error case.

Using the above error cases, we illustrate the coset structure in Supplementary Note 1 (see S Fig. 1 therein).

A good code can resolve many erasures with $|\mathbb{r}| = np$ at large p , and we aim to resolve them in linear time.

Bit-Flipping BP algorithm

Given (6), BP can approximate the solution with high accuracy, by running a message passing on the Tanner graph induced by the code's check matrix. For classical BP, see refs. 71–75. For quantum decoding, we can perform binary decoding based on a binary check matrix $H \in \{0, 1\}^{m \times 2n}$, or a quaternary decoding based on a Pauli check matrix $H \in \{I, X, Y, Z\}^{m \times 2n}$. For example, matrices in (8) and (9) induce the two Tanner graphs shown in Fig. 3.

For a Tanner graph, let $\mathcal{N}(i)$ be the set of neighbouring variable node indices of check node i . Let $\mathcal{M}(j)$ be the set of neighbouring check node indices of variable node j . For example, for a Tanner graph induced by an $H \in \{0, 1\}^{m \times 2n}$,

$$\begin{aligned} \mathcal{N}(i) &= \{j \in \{1, 2, \dots, 2n\} : H_{ij} = 1\}, \\ \mathcal{M}(j) &= \{i \in \{1, 2, \dots, m\} : H_{ij} = 1\}. \end{aligned}$$

For BP erasure decoding, important code properties include *stopping sets* and *girth*. For details, see Supplementary Note 3. However, due to the degeneracy of stabilizer codes, the solutions are generally not unique. Specifically, the number of feasible solutions in a logical coset (Lemma 1) increases with the number of erasures (cf. Example 1). When decoding a code, we aim for the decoder to combat an erasure rate p as high as possible, which leads to many erasures and, consequently, many feasible solutions in the logical coset where the actual error is located. In particular, we observe strong symmetry of degenerate errors (see⁵⁷ or more details in Supplementary Note 4). This motivates us to use heuristics to converge to any feasible solution. Thus, we propose a bit-flipping algorithm with a gradient descent (GD) step, referred to as GD Flip-BP₂, as outlined in Algorithm 1. (This algorithm behaves more like a greedy approach to minimize a discrete function (11), but can be linked to GD optimization for a differentiable function (18)).

We explain Algorithm 1. The binary check matrix has $2n$ columns. Given $\mathbb{r} \subseteq \{1, 2, \dots, n\}$, we have

$$\mathbb{r}_2 = \mathbb{r} \cup (\mathbb{r} + n),$$

where $\mathbb{r} + n = \{j + n : j \in \mathbb{r}\}$. The GD step performs when no updates are performed during an iteration. This step tries to minimize an objective

function defined as the sum of the numbers of unresolved bits in each check, expressed as

$$\sum_{i=1}^m |\mathcal{N}(i) \cap \mathbb{r}_2|. \quad (11)$$

Algorithm 1. GD Flip-BP₂

Input:

A binary matrix $H \in \{0, 1\}^{m \times 2n}$, a syndrome $\mathbf{s} \in \{0, 1\}^m$, erased coordinates $\mathbb{r}_2 \subseteq \{1, 2, \dots, 2n\}$, and an integer $T_{\max} > 0$. ($H = [B^Z|B^X]$ from a binary check matrix $B = [B^X|B^Z]$).

Initialization.

- Let $\Gamma_j = 0$ if $j \in \mathbb{r}_2$, and $\Gamma_j = +1$ if $j \in \bar{\mathbb{r}}_2$.

Iterative Update.

- For $i \in \{1, 2, \dots, m\}$: Compute $\Delta_i = |\mathcal{N}(i) \cap \mathbb{r}_2|$; Then if $\Delta_i = 1$, let j be the single element in $\mathcal{N}(i) \cap \mathbb{r}_2$ and set

$$\Gamma_j = (-1)^{s_i} \prod_{j' \in \mathcal{N}(i) \setminus \{j\}} \Gamma_{j'}.$$

- (GD Step) If no Γ_j is updated in the previous sub-step: Find a $j \in \mathbb{r}_2$ such that the corresponding column in $H_{\mathbb{r}_2}$ has a maximum column weight. Set $\Gamma_j = -1$.
- If any Γ_j is updated during this iteration, remove j from $\mathbb{r}_2$.

Judgement.

- If $\mathbb{r}_2 = \emptyset$: For all $j = 1, \dots, 2n$, if $\Gamma_j = +1$, set $\hat{E}_j = 0$, and if $\Gamma_j = -1$, set $\hat{E}_j = 1$. Generate $\hat{\mathbf{s}} = \langle \hat{E}, \mathbf{B} \rangle = \hat{E}H^T$.
 - if $\hat{\mathbf{s}} = \mathbf{s}$: return “CONVERGE”;
 - else: return “GD_FAIL”.
- Else: if the maximum number of iterations $T_{\max}$ is reached, halt and return “FAIL”.
- Repeat from the step of Iterative Update.

Algorithm 2. MBP₂ (with an optional soft GD step)

Input:

A binary matrix $H \in \{0, 1\}^{m \times 2n}$, a syndrome $\mathbf{s} \in \{0, 1\}^m$, an integer $T_{\max} > 0$, a real $\alpha > 0$, and initial LLRs $\Lambda_j = \ln(p_j^{(0)}/p_j^{(1)})$ for $j = 1, 2, \dots, 2n$ (cf. (14)). ($H = [B^Z|B^X]$ from a binary check matrix $B = [B^X|B^Z]$). GD parameters: an integer $T_{\text{GD}} > 0$ and a real $|\Lambda_{\text{GD}}| > 0$.

Initialization.

- For $j \in \{1, 2, \dots, 2n\}$ and $i \in \mathcal{M}(j)$: Let $\Gamma_{j \rightarrow i} = \text{soft}(\Lambda_j)$.

Check-Node Computation

- For $i \in \{1, 2, \dots, m\}$: Let $\Delta_i = (-1)^{s_i} \boxplus_{j \in \mathcal{N}(i)} \Gamma_{j \rightarrow i}$.
- For $i \in \{1, 2, \dots, m\}$ and $j \in \mathcal{N}(i)$: Let $\Delta_{i \rightarrow j} = \Delta_i \boxminus \Gamma_{j \rightarrow i}$.

Variable-Node computation.

- For $j \in \{1, 2, \dots, 2n\}$: Let $\Gamma_j = \Lambda_j + (1/\alpha) \sum_{i \in \mathcal{M}(j)} \Delta_{i \rightarrow j}$.

- For $j \in \{1, 2, \dots, 2n\}$ and $i \in \mathcal{M}(j)$: Compute

$$\begin{aligned}\Gamma_{j \rightarrow i} &= \Gamma_j - \Delta_{i \rightarrow j}, \\ \Gamma_{j \rightarrow i} &= \text{soft}(\Gamma_{j \rightarrow i}).\end{aligned}\quad (12)$$

- (Optional soft GD Step) When the number of iterations is a multiple of T_{GD} : For $j \in \{1, 2, \dots, 2n\}$,

$$\text{if } |\Gamma_j| < |\Lambda_{\text{GD}}|, \text{ then set } \Lambda_j = \text{sign}(\Gamma_j) |\Lambda_{\text{GD}}|. \quad (13)$$

Judgement.

- For all $j = 1, \dots, 2n$, if $\Gamma_j \geq 0$, set $\hat{E}_j = 0$, and if $\Gamma_j < 0$, set $\hat{E}_j = 1$. Generate $\hat{\mathbf{s}} = \langle \hat{\mathbf{E}}, \mathbf{B} \rangle = \hat{\mathbf{E}} \mathbf{H}^T$.
 - if $\hat{\mathbf{s}} = \mathbf{s}$: return “CONVERGE”;
 - else if the maximum number of iterations T_{max} is reached: halt and return “FAIL”.
 - else: repeat from the Check-Node Computation step.

MBP₂ and its adaptive version (AMBP₂)

A soft-valued decoder improves decoding accuracy. A bit could be 0, 1 or erased (denoted by ?). The values 0, ?, 1 are mapped as +1, 0, −1 in GD Flip-BP₂ for computation. Once a decision is made, ? becomes a bit value 0 or 1 and remains fixed, though it may be incorrect. Generally, we consider a soft-valued decoder, where the messages are represented as soft values, allowing them to adjust their signs and magnitudes during iterations, improving decision accuracy.

We describe this decoder generalization. We have the initial probabilities $p_j^{(0)} = \Pr(E_j = 0)$ and $p_j^{(1)} = \Pr(E_j = 1)$ for $j = 1, 2, \dots, 2n$, where

$$(p_j^{(0)}, p_j^{(1)}) = (p_{j+n}^{(0)}, p_{j+n}^{(1)}) = \begin{cases} (\frac{1}{2}, \frac{1}{2}) & \text{if } j \in \mathbb{I}, \\ (1, 0) & \text{if } j \in \bar{\mathbb{I}}. \end{cases} \quad (14)$$

The initial *log-likelihood ratios* (LLRs) are $\Lambda_j = \ln(p_j^{(0)}/p_j^{(1)})$ for $j = 1, 2, \dots, 2n$. The (hard) values 0, ?, 1 of E_j map to $+\infty, 0, -\infty$ as Λ_j . We will update these LLRs, and denote the updated LLRs as Γ_j , $j = 1, 2, \dots, 2n$, to estimate $\hat{\mathbf{E}}$.

We restrict variable-to-check messages from taking values in $-\infty, 0, \infty$, avoiding representing the three (hard) values 0, ?, 1. The function achieves our purpose:

$$\text{soft}(x) = \begin{cases} \text{sign}(x) \text{LLR_MAX}, & \text{if } |x| > \text{LLR_MAX}, \\ \text{sign}(x) \text{LLR_MIN}, & \text{if } |x| < \text{LLR_MIN}, \\ x, & \text{otherwise.} \end{cases} \quad (15)$$

This maintains LLR numerical stability and enables updates even within stopping sets. Suggested values for LLR_MAX and LLR_MIN are provided in Supplementary Note 9.

Remark 3. We do not apply the function $\text{soft}(\cdot)$ in (15) to initial LLRs Λ_j . This ensures $\Gamma_j = \Lambda_j = +\infty$ for $j \in \bar{\mathbb{I}}$, making our BP output always erasure-matched, though convergence (syndrome matching, $\hat{\mathbf{s}} = \mathbf{s}$) is not guaranteed.

We will update LLRs to meet the target syndrome. For two LLRs Γ_1 and Γ_2 , following Gallager⁷¹, we define

$$\Gamma_1 \boxplus \Gamma_2 := \text{sign}(\Gamma_1 \Gamma_2) f^{-1}(f(|\Gamma_1|) + f(|\Gamma_2|)), \quad (16)$$

where $|\cdot|$ is absolute value, $f(x) = \ln \frac{e^x + 1}{e^x - 1} = \ln(\coth(\frac{x}{2}))$, and notice that $f^{-1}(x) = f(x)$. An equivalent expression is $\Gamma_1 \boxplus \Gamma_2 = 2 \tanh^{-1}(\tanh(\frac{\Gamma_1}{2}) \tanh(\frac{\Gamma_2}{2}))$ ⁷⁵, and in general,

$$\boxplus_{j=1}^w \Gamma_j := 2 \tanh^{-1} \left(\prod_{j=1}^w \tanh \frac{\Gamma_j}{2} \right). \quad (17)$$

To resolve all erasures (minimizing (11)) while satisfying the target syndrome, we define a check-satisfaction reward function for maximization:

$$S(\Gamma_1, \Gamma_2, \dots, \Gamma_{2n}) := \sum_{i=1}^m (-1)^{s_i} \boxplus_{j \in \mathcal{N}(i)} \Gamma_j. \quad (18)$$

Algorithm 2 presents MBP₂, performing a soft message-passing to approximate a solution to (6) by updating LLRs. Specifically, it approximates Γ_j to $\ln \frac{\Pr(E_j=0 | \text{syndrome is } \mathbf{s})}{\Pr(E_j=1 | \text{syndrome is } \mathbf{s})}$ by maximizing (18) (see (22)–(27)).

The check-to-variable messages $\Delta_{i \rightarrow j}$ are normalized by some $\alpha > 0$, like classical normalized BP^{76,77}. Typically, $\alpha \geq 1$ is chosen to suppress over-estimated messages. Here, we also consider $\alpha < 1$ to enlarge the BP step-size to search more possible solutions. Larger step-size may cause divergence. Inspired by Hopfield nets^{60–62}, MBP uses fixed inhibition (12), adding memory effects to BP to improve convergence even with $\alpha < 1$ ⁵⁹ (see Supplementary Note 5 (part b)).

We can select a proper α value using GD optimization to maximize (18). In METHODS, we validate this by linking BP update rules with GD. Given the physical error rate p , we can determine α from a locally linear function of p using the continuity of the reward function (18) (see (22)–(27)). Furthermore, we introduce an optional step (13) to adjust the initial LLR Λ_j to have enough magnitude $|\Lambda_{\text{GD}}|$, ensuring enough update gradients in harmful stopping sets. This is performed whenever the updated LLR Γ_j for index j does not have enough magnitude (i.e., $|\Gamma_j| < |\Lambda_{\text{GD}}|$) after every T_{GD} iterations.

In BP, the message $\Delta_{i \rightarrow j}$ is typically computed by $\Delta_{i \rightarrow j} = (-1)^{s_i} \boxplus_{j' \in \mathcal{N}(i) \setminus \{j\}} \Gamma_{j' \rightarrow i}$. For efficiency, we compute it by

$$\Delta_{i \rightarrow j} = \Delta_i \Gamma_{j \rightarrow i} := 2 \tanh^{-1} \left(\tanh \frac{\Delta_i}{2} / \tanh \frac{\Gamma_{j \rightarrow i}}{2} \right), \quad (19)$$

where $\Delta_i = (-1)^{s_i} \boxplus_{j' \in \mathcal{N}(i)} \Gamma_{j' \rightarrow i}$ is computed first and every $\Gamma_{j \rightarrow i} \neq 0$. This ensures $0 < |\Delta_i| < |\Gamma_{j \rightarrow i}|$ for numerical stability, which is guaranteed by the function $\text{soft}(\cdot)$ in (15). Compared to the typical computation, this reduces the BP complexity from $O(2nj_2(j_2 - 1))$ to $O(2nj_2)$ per-iteration, where j_2 is the mean column-weight of $\mathbf{H}$.

Additionally, we observe that altering the message-update schedule can significantly improve decoding convergence for highly degenerate codes (Supplementary Note 8).

Algorithm 3. AMBP₂

Input: The same as in Algorithm 2 but further with a sequence of real values $\alpha_1 > \alpha_2 > \dots > \alpha_\ell > 0$.

Initialization: Let $i = 1$.

MBP Step: Run MBP MBP₂($\mathbf{H}, \mathbf{s}, T_{\text{max}}, \alpha_i, \{\Lambda_j\}$), which returns an indicator “CONVERGE” or “FAIL” with an estimated $\hat{\mathbf{E}} \in \{0, 1\}^{2n}$.

Adaptive Check:

- If the return indicator is “CONVERGE”, return “CONVERGE” (with $\alpha^* = \alpha_i$ and the estimated $\hat{\mathbf{E}}$);
- Otherwise, let $i \leftarrow i + 1$; if $i > \ell$, return “FAIL”;
- Otherwise, repeat from the MBP Step.

For erasure decoding, all MBP instances with different α values can run in parallel, halting all instances as soon as one finds a feasible solution.

As discussed earlier, we can select the α value from GD, but it is based on the reward function (18) and not necessarily optimal. First, the function in (18) is not concave and has vanishing gradients when encountering stopping sets, so a gradient-based step-size search does not ensure convergence. Second, directly using a large step-size may lead to oscillations and prevent the algorithm from converging. To mitigate this, we can test a list of decoders with step-sizes ranging from small to large step-sizes ($1/\alpha$). Thus, we also propose an adaptive MBP₂ (AMBP₂), as in Algorithm 3, which tests a list of $\alpha_1 > \dots > \alpha_\ell$ and adaptively select a proper α^* .

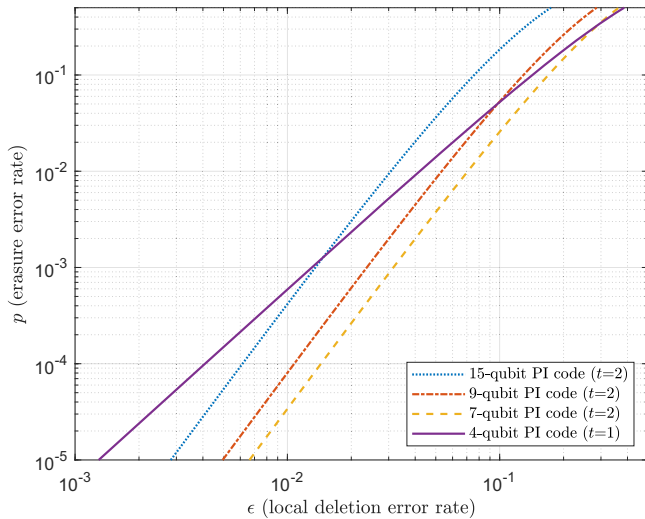


Fig. 4 | Local deletion to erasure rate conversion using $[[\ell, 1]]$ PI codes for $\ell = 4, 7, 9, 15$ (^{64,90,84,83,87}). The $[[4, 1]]$ PI code converts $\epsilon = 0.385$ to $p = 0.5$, which can then be handled by topological codes. At $\epsilon = 0.1$, an $[[\ell < 15, 1]]$ PI code achieves $p < 0.1$, allowing a QLDPC code rate above $3/4$ (Fig. 2). At $\epsilon = 0.01$, using a 7-qubit PI code achieves $p < 10^{-4}$, which allows QLDPC codes to attain a rate of essentially 1.

MBP₄, and unified (A)MBP_q notation, $q = 2$ or 4

We also consider quaternary BP (BP₄). This BP leverages the correlation between bits j and $j + n$ via the Pauli check matrix $H \in \{I, X, Y, Z\}^{m \times n}$ for error correction, enhancing accuracy in both classical⁷⁸ and quantum settings⁶³, when compared to binary BP. Indirectly, this can be interpreted as a turbo update between the two error components ($E^X|E^Z$)⁷⁹, reducing the likelihood of the decoder being halted by binary stopping sets in either the X or Z portion. Preliminary simulations show very good accuracy, so we do not design a GD step for quaternary BP.

For quaternary BP in the quantum setting, we express the variable-to-check ($j \rightarrow i$) message, following^{63,80}, as

$$\ln \frac{\Pr(\text{Pauli error } E_j \text{ and entry } H_{ij} \text{ commute})}{\Pr(\text{Pauli error } E_j \text{ and entry } H_{ij} \text{ anticommute})}, \quad (20)$$

$$\text{e.g., } = \ln \frac{p_j^I + p_j^X}{p_j^Y + p_j^Z} \text{ for initial LLR } (j \rightarrow i) \text{ if } H_{ij} = X. \quad (21)$$

Extending the derivations in (22)–(27), we obtain quaternary MBP (MBP₄) and its adaptive version (AMBP₄)⁵⁹.

Algorithm 4. Erasure decoding with (A)MBP_q, $q = 2$ or 4

Input:

A check matrix $H \in \{I, X, Y, Z\}^{m \times n}$, a syndrome $\mathbf{s} \in \{0, 1\}^m$, erased coordinates $\mathbb{r} \subseteq \{1, 2, \dots, n\}$, an integer $T_{\max} > 0$, and a set of real $\alpha_1 > \alpha_2 > \dots > \alpha_\ell > 0$.

Include T_{GD} and $|\Lambda_{\text{GD}}|$ if the GD step is applied to (A)MBP₂.

Initialization.

- Set $(p_j^I, p_j^X, p_j^Y, p_j^Z) = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ for $j \in \mathbb{r}$ and $(p_j^I, p_j^X, p_j^Y, p_j^Z) = (1, 0, 0, 0)$ for $j \in \bar{\mathbb{r}}$. (Convert to binary distributions (14) if $q = 2$.)

Decoding.

- Run (A)MBP_q($H, \mathbf{s}, T_{\max}, \{p_j^I, p_j^X, p_j^Y, p_j^Z\}_{j=1}^n, \{\alpha_i\}_{i=1}^\ell$), with optional GD step with input $(T_{\text{GD}}, |\Lambda_{\text{GD}}|)$ if applied, and obtain an estimate $\hat{E} \in \{I, X, Y, Z\}^n$.

Output.

- Return $\hat{E} \in \{0, 1\}^{2n}$ and $\hat{\mathbf{s}} = (\hat{E}, \mathbf{H})$, where $\hat{E}$ and $\mathbf{H}$ are the binary version of $\hat{E}$ and H , respectively.

Given a Pauli check matrix $H \in \{I, X, Y, Z\}^{m \times n}$, which uniquely determines a binary check matrix $\mathbf{H} \in \{0, 1\}^{m \times 2n}$, we consider a unified (A)MBP notation: let $\ell = 1$ for MBP and $\ell > 1$ for AMBP, and refer the decoder by

$$(A)MBP_q(H, \mathbf{s}, T_{\max}, \{p_j^I, p_j^X, p_j^Y, p_j^Z\}_{j=1}^n, \{\alpha_i\}_{i=1}^\ell),$$

with $q = 4$ for (A)MBP₄ and $q = 2$ for (A)MBP₂. We state a unified (A)MBP_q algorithm in Algorithm 4.

For (A)MBP₂, if the optional GD step is applied, we refer the algorithm as GD (A)MBP₂ in simulations.

Discussions

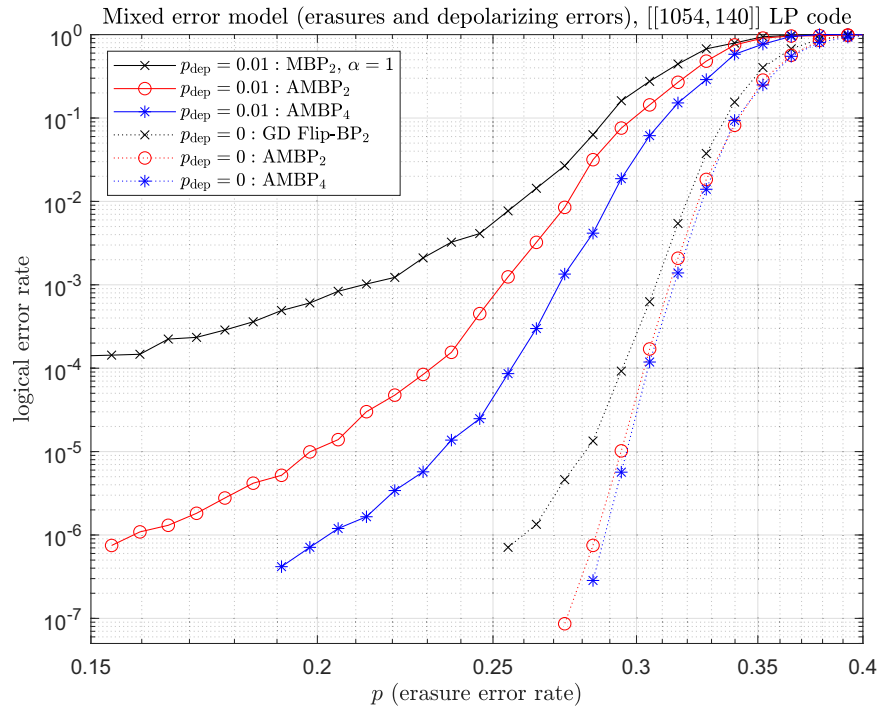
We analysed the general erasure decoding problem, and proposed efficient decoders suitable for this task. Topological codes were known to achieve capacity with vanishing code rate^{14–16}. We construct various rate bicycle and LP codes, and their MLD thresholds attain the erasure capacity (Fig. 2). Soft (A)MBP decoders behave like MLD at low logical error rate, achieving near-capacity performance with a small gap. The GD flipping decoder performs well on bicycle and LP codes, despite its extremely low complexity.

The Gaussian decoder resembles the core idea of ordered statistics decoding (OSD)⁸¹, solving linear equations restricted to unreliable error estimates. Combining BP with OSD improves BP accuracy for decoding depolarizing errors^{58,82}. For erasures, a similar BP + Gaussian approach can achieve MLD performance. In this setting, Gaussian elimination targets only erasures left by BP. This reduces complexity compared to using the Gaussian decoder alone, though the overall complexity remains $O(n^3)$; see Supplementary Note 10 for details.

Our results extend beyond erasure errors. First, we consider deletion errors, which follow the same model as erasure errors but without knowledge of the error locations. We apply our results on concatenated QLDPC codes with permutation-invariant (PI) codes^{64,65,83–87} as inner codes. Deletion errors can be catastrophic for stabilizer codes, but PI codes can correct them^{65,88}. Unlike erasures, a primary challenge of deletion errors is the lack of error location information, making decoding more difficult, as misidentifying even one location can cause failure. Accurately locating all deletions to obtain erasures aids decoding but is time-consuming, increasing the physical error rate. Thus, accepting imperfect erasure conversions, where some leakage errors remain as localized deletions, can be more effective. We assume a local deletion model where each block of ℓ physical qubits has an identifiable boundary, protected by an $[[\ell, 1]]$ PI code, converting deletions into erasures. Figure 4 shows the deletion-to-erasure conversion performance in a channel with deletion probability ϵ , using a PI code that corrects t deletions (see legend). The $[[4, 1]]$ PI code^{64,89,90} converts local deletions with probability $\epsilon = 0.385$ into erasure errors of rate $p < 0.5$, which topological codes can handle. At $\epsilon = 0.1$, an $[[\ell, 1]]$ code with $\ell < 15$ achieves a converted $p < 0.1$, allowing a QLDPC code rate above $3/4$ (Fig. 2). In particular, if the per-qubit deletion probability is 0.01, then using a 7-qubit PI code can bring the logical erasure rate to below 10^{-4} , which allows QLDPC codes to attain a rate of essentially 1, which is close to the maximum possible rate.

Second, in a large system, after converting dominant noise into erasures at known locations, we model additional errors that occur as qubit depolarizing errors. This mixed error model is discussed in⁹¹ using peeling on 2D codes. In contrast, (A)MBP can decode general QLDPC codes in this error model. We show this using a $[[1054, 140]]$ LP code (from¹⁹) with the simulation setup in METHODS. With (A)MBP₄, we adapt by adjusting the initial error distribution for the non-erased qubits to $(p^I, p^X, p^Y, p^Z) = (1 - p_0, \frac{p_0}{3}, \frac{p_0}{3}, \frac{p_0}{3})$, where p_0 can be set to either the depolarizing error rate p_{dep} or a suitable fixed value $< p_{\text{dep}}$ ^{59,92}. With (A)MBP₂, the initial error distribution is $(p^{(0)}, p^{(1)})$, where $p^{(1)} = p^X + p^Y = p^Z + p^Y$ and $p^{(0)} = 1 - p^{(1)}$. The adjustment enables soft BP decoders to simultaneously correct both erasure and depolarizing errors. Figure 5 shows our decoder accuracy, assuming $p_{\text{dep}} = 0.01$. GD Flip-BP₂ cannot handle depolarizing errors, so we use MBP₂ with $\alpha = 1$ as a soft extension. All decoders successfully handle mixed errors.

Fig. 5 | BP accuracy for mixed erasure and depolarizing errors using the $[[1054, 140]]$ LP code. The curves with only erasure errors are provided for reference ($p_{\text{dep}} = 0$). AMBP₄ maintains strong accuracy in the mixed error model ($p_{\text{dep}} = 0.01$).



Among them, AMBP₄ shows the best accuracy, as depolarizing errors exhibit strong correlations between X and Z errors.

Our findings highlight the proposed decoders' adaptability and accuracy in handling various error types, including erasures, deletions, and depolarizing errors. We expect the versatility and high performance of our proposed decoders to contribute to the development of quantum LDPC codes and fault-tolerant quantum computation.

For practical use, the tanh rule (17) can be approximated by the min-sum rule, which selects the minimum magnitude among the input LLRs and combines signs separately^{77,93,94}. It retains soft information and may outperform flip BP, motivating its study in quantum erasure correction.

Quantum channel limits have long been known^{68,95,12}. For the depolarizing channel, the hashing bound (also known as the quantum Hamming bound) sets the capacity for non-degenerate codes, though degenerate codes may slightly exceed it at low physical error rates^{68,95}. However, near-hashing-bound decoding has emerged only recently⁹⁶. By contrast, although the capacity of the erasure channel is known¹², comparable decoding results for erasure noise models remain largely unexplored. Our work closes this gap by providing decoders simultaneously achieving near-erasure-capacity accuracy and linear-time complexity.

A recent review⁹⁷ surveys decoding algorithms for surface codes, including erasure decoding, although focusing on the union-find (peeling) decoder. We recall that peeling decoder + clusters decomposition with unconstrained size ($O(n^3)$ complexity) achieves MLD on GHP codes⁴². More recently, a quantum Maxwell erasure decoder for CSS QLDPC codes⁹⁸ extends peeling with guessing, which interpolates between linear-time decoding (constant guessing budget) and MLD (unconstrained budget). Its asymptotic performance has been analyzed and simulated for bivariate bicycle and quantum Tanner codes.

Circuit-level decoding plays a key role in fault-tolerant quantum computation⁴⁵, with substantial progress focused mainly on depolarizing and gate noise with unknown locations, notably for topological codes^{99–102} and bivariate bicycle codes¹⁷. Implementing circuit-level simulations can be assisted by the stabilizer-circuit tool Stim¹⁰³, following standard practice in circuit-level decoding under unknown error locations^{104–107} or erasures^{108–111}.

Our approach can be generalized similarly using Stim. Alternatively, to retain the benefits of non-binary (A)MBP decoding, our approach

extends to phenomenological or circuit-level noise by representing the syndrome, data, and two-qubit gate errors with mixed-alphabet variables in BP^{69,70}. While circuit-level errors with unknown locations are more challenging, prior results demonstrate the viability of this mixed-alphabet approach^{69,70}.

A detailed treatment of circuit-level erasures, which requires additional erasure-conversion circuits^{108–111}, lies beyond the scope of this paper.

Methods

BP update rules and the relation with GD

For BP₂ syndrome decoding in the erasure channel, we have to satisfy two constraints, $\mathbb{r}$ and $\mathbb{s}$ (i.e., the erasure locations and the syndrome). The erasure information $\mathbb{r}$ is embedded in the initial distributions (14). The syndrome information $\mathbb{s}$ is included in BP₂ to update the initial LLR $\Lambda_j = \ln \frac{\Pr(E_j=0)}{\Pr(E_j=1)}$ to $\Gamma_j = \ln \frac{\Pr(E_j=0 | \text{syndrome is } \mathbb{s})}{\Pr(E_j=1 | \text{syndrome is } \mathbb{s})}$ for each binary error E_j for $j = 1, 2, \dots, 2n$. It is well-known that normalized BP₂^{76,77} provides an efficient computation

$$\Gamma_j = \Lambda_j + \frac{1}{\alpha} \sum_{i \in \mathcal{M}(j)} (-1)^{s_i} 2 \tanh^{-1} \left(\prod_{j' \in \mathcal{N}(i) \setminus \{j\}} \tanh \frac{\Gamma_{j' \rightarrow i}}{2} \right) \quad (22)$$

$$= \Lambda_j + \frac{1}{\alpha} \sum_{i \in \mathcal{M}(j)} \Delta_{i \rightarrow j} \quad (23)$$

like Algorithm 2. We justify this computation for the first iteration, with $\alpha = 1$, in Supplementary Note 5 (part a).

To determine the value of α in Algorithm 2, we relate BP to GD^{99,112}. Define the satisfaction at check node i as $(-1)^{s_i} 2 \tanh^{-1} \left(\prod_{j \in \mathcal{N}(i)} \tanh \frac{\Gamma_j}{2} \right)$. Summing this for all check nodes, we get the reward function (18), which is:

$$S(\Gamma_1, \dots, \Gamma_{2n}) = \sum_{i=1}^m (-1)^{s_i} 2 \tanh^{-1} \left(\prod_{j \in \mathcal{N}(i)} \tanh \frac{\Gamma_j}{2} \right). \quad (24)$$

Then

$$\frac{\partial S}{\partial \Gamma_j} = \sum_{i \in \mathcal{M}(j)} g_{ij} (-1)^{s_i} \prod_{j' \in \mathcal{N}(i) \setminus \{j\}} \tanh \frac{\Gamma_{j'}}{2}, \quad (25)$$

where g_{ij} is a positive term, as a function of $(\Gamma_1, \dots, \Gamma_{2n})$, as

$$g_{ij}(\Gamma_1, \dots, \Gamma_{2n}) = \frac{1 - \tanh^2 \frac{\Gamma_j}{2}}{1 - \left(\prod_{j' \in \mathcal{N}(i)} \tanh \frac{\Gamma_{j'}}{2} \right)^2} > 0. \quad (26)$$

Notice that $|\Gamma_{j \rightarrow i}| < \infty$ due to the use of $\text{soft}(\cdot)$ in (15). For non-erased part, $\Lambda_j = +\infty$, enforcing $\Gamma_j = \Lambda_j$ in (22) and fixing bit j decision as 0. For erased part, $\Lambda_j = 0$ in (22), so the other term determines the decision of bit j . Comparing this term in (22) and (25), and noting that $\tanh^{-1}(\cdot)$ and $g_{ij} > 0$ (as a multiplier) preserve the sign, we conclude that for BP, we should choose

an α value such that

$$\frac{1}{\alpha} 2 \tanh^{-1} \left(\prod_{\substack{j' \in \mathcal{N}(i) \\ j' \neq j}} \tanh \frac{\Gamma_{j' \rightarrow i}}{2} \right) \propto g_{ij} \prod_{\substack{j' \in \mathcal{N}(i) \\ j' \neq j}} \tanh \frac{\Gamma_{j'}}{2} \quad (27)$$

to minimize (24). Using (27), we can express α as a locally linear function of p . A proper initial α value from p notably reduces AMBP iteration count (see Table 2).

We determine α by associating Γ_j with the physical error rate p . This step is crucial to prevent wrong decisions in the first iteration that would propagate incorrect information in subsequent iterations. As all functions in (27) are continuous, α is locally linear in Γ_j . Associating Γ_j with the channel error rate p , we express α as a locally linear function of p . We also associate α with the GD parameters in Algorithm 2 and obtain $T_{\text{GD}} = 5$ and $|\Lambda_{\text{GD}}| = 0.25$ for enough gradients for algorithm updates. See Supplementary Note 5 (part c).

Simulation setup

We conduct computer simulations for HP⁵⁰ and GHP codes⁵⁸, bicycle codes¹⁸, LP codes¹⁹, and topological toric^{44–46} and XZZX codes^{47–49}. Supplementary Note 7 provides more information about the codes. We test BP and MLD on these codes.

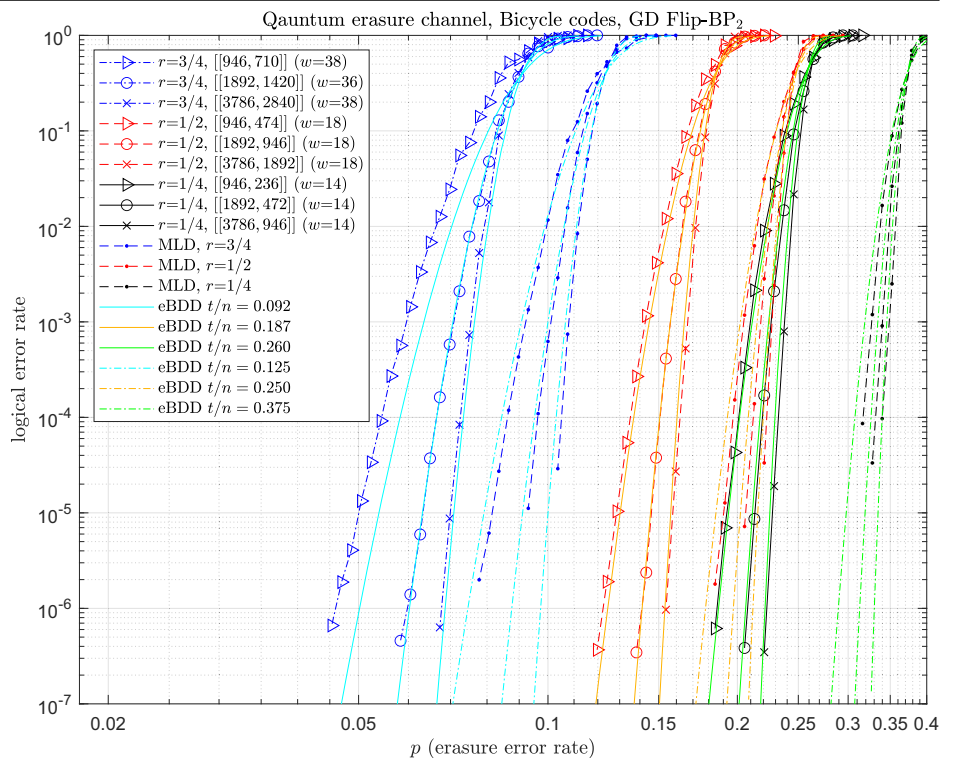
For our BP decoders, we set $T_{\text{max}} = 100$. The average iteration count scales as $O(\log \log n)$ at relevant error rates, as discussed later. For (A) MBP, unless otherwise stated, we use the group random schedule (Supplementary Note 8), and select α^* from 1.2, 1.19, ..., 0.3 using adaptive scheme.

Table 2 | Average BP iteration counts for the [[882, 48, 16]] GHP code

average iteration counts	$p=0.255$	$p=0.328$	$p=0.392$
AMBP ₄ , $\alpha_1 = 1.2$	3.28	10.54	4628.89
AMBP ₄ , $\alpha_1 = \text{func}(p)$	3.28	8.33	90.27
MBP ₄ , $\alpha = 1.2$	3.27	6.14	80.12
GD AMBP ₂ , $\alpha_1 = 1.2$	3.30	11.98	5330.62
GD AMBP ₂ , $\alpha_1 = \text{func}(p)$	3.30	10.67	94.75
GD MBP ₂ , $\alpha = 1.2$	3.30	6.80	83.55
GD Flip-BP ₂	2.99	5.18	51.51

We include more decoders for reference than in Fig. 11. All BPs remain efficient at low error rates. For AMBP, two strategies (31) (fixed α_1) and (32) ($\alpha_1 = \text{func}(p)$) are compared, where (32) provides much faster runtime at high error rates.

Fig. 6 | Accuracy of GD Flip-BP₂ and MLD on bicycle codes. Decoders (A) MBP_q have accuracy comparable to GD Flip-BP₂ on these codes. BP threshold for rate 1/4 can be improved from 0.26 to 0.285 by using row-weight $w = 12$ (as shown in Supplementary Note 7 (part c)).



A sphere-packing bound for accuracy evaluation

We aim for correcting t erasures with $t \propto n$. If a qubit is erased, it encounters I, X, Y , or Z with equal probability $1/4$. Using (2), a qubit encounters I with probability $1 - p + \frac{p}{4} = 1 - \frac{3}{4}p$, and encounters X, Y , or Z with equal probability $\frac{p}{4}$. Thus, before specifying $\mathbb{r}$, we have

$$(p^I, p^X, p^Y, p^Z) = \left(1 - \frac{3p}{4}, \frac{p}{4}, \frac{p}{4}, \frac{p}{4}\right). \quad (28)$$

When $t \propto n$ are both large, we expect most occurred errors $E \in \{I, X, Y, Z\}^n$ to have approximately $\frac{3}{4}t$ components as X, Y , or Z . The probability that the number of such components deviates from $\frac{3}{4}t$ by δt for any $\delta > 0$ is exponentially suppressed in $t \propto n$.

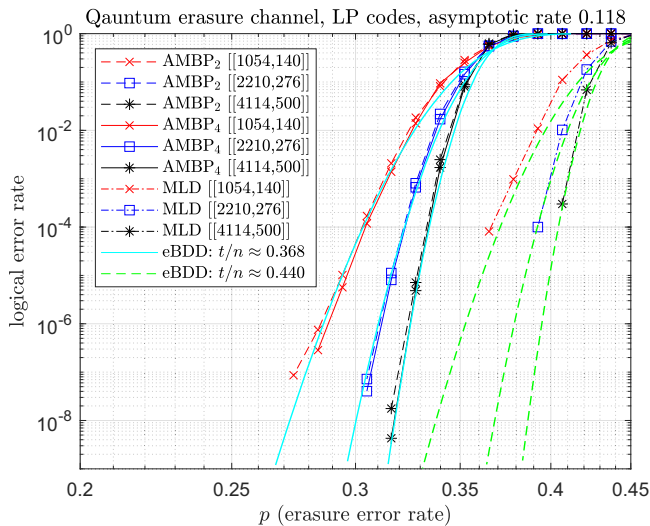


Fig. 7 | Accuracy of AMBP and MLD on rate 0.118 LP codes. Our AMBP_q exhibit a threshold of $p_{\text{th}} = 0.368$. MLD curves suggest a higher threshold of $p_{\text{th}} = 0.44$. (Results for rate 0.04 LP codes are in Supplementary Note 7 (part d)).

Definition 4. The logical error rate of *erasure bounded-distance-decoding* (eBDD) with parameters n, t , and p is

$$P_{\text{eBDD}}(n, t, p) := \sum_{j=\lceil \frac{3}{4}t \rceil + 1}^n \binom{n}{j} \left(\frac{3}{4}p\right)^j \left(1 - \frac{3}{4}p\right)^{n-j}. \quad (29)$$

If a code family has decoding accuracy closely matching $P_{\text{eBDD}}(n, t, p)$ with a constant $\frac{t}{n} = p_{\text{th}}$ as n grows, then p_{th} serves an estimated threshold for the code family. With a fixed $\frac{t}{n} = p_{\text{th}}$ in (29), at any physical error rate $p < p_{\text{th}}$, the function $P_{\text{eBDD}}(n, t, p)$ decays exponentially as n increases (see Supplementary Note 6). The accuracy estimated from eBDD is more reliable than the intersection of performance curves, particularly at low error rates (see Fig. 8).

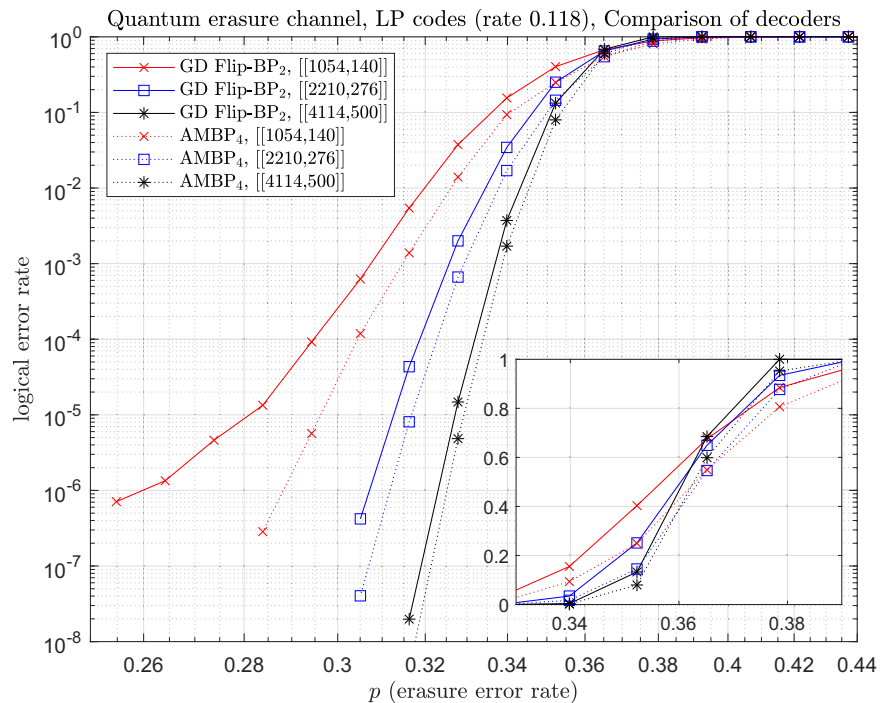
Performance of capacity-achieving codes

We consider moderate-to-high-rate bicycle codes, low-rate LP codes, and vanishing-rate 2D topological codes. For bicycle and LP code families, we estimate the threshold values using eBDD, by comparing the decoder accuracy with (29) given fixed t/n .

First, we construct bicycle codes with rates $1/4, 1/2$, and $3/4$. A key step in bicycle construction is row-deletion, which requires heuristics¹⁸, discussed in Supplementary Note 7 (part c). GD Flip-BP₂ suffices for decoding bicycle codes, with accuracy shown in Fig. 6, including MLD for comparison. Soft decoders (A)MBP have accuracy comparable to GD Flip-BP₂ on these codes. Using eBDD, we estimate the threshold values, as shown in the legend of Fig. 6. Bicycle codes exhibit capacity accuracy ($r = 1 - 2p$) with MLD threshold values at $p = 0.125, 0.25, 0.375$ for rates $r = 3/4, 1/2, 1/4$, respectively. The BP threshold follows closely, with a small gap at $p = 0.092, 0.187, 0.26$ for these rates. BP threshold for rate $1/4$ can be improved from 0.26 to 0.285 by using row-weight $w = 12$, by tolerating some error floor (see Supplementary Note 7 (part c)).

Second, we construct quantum LP codes [19, Example 3] from classical quasi-cyclic (QC) matrices^{27,113,114}. We construct LP code families with rates of 0.118 and 0.04 . For rate 0.118 , Figure 7 show the accuracy of AMBP and MLD, along with eBDD curves. LP codes exhibit strong accuracy with MLD,

Fig. 8 | GD Flip-BP₂ vs AMBP₄ on rate 0.118 LP codes. AMBP_4 has a threshold of 0.368 (Fig. 7). GD Flip-BP₂ closely matches AMBP_4 as n increases. Also note that good curve intersection does not guarantee accuracy at low error rates (red line).



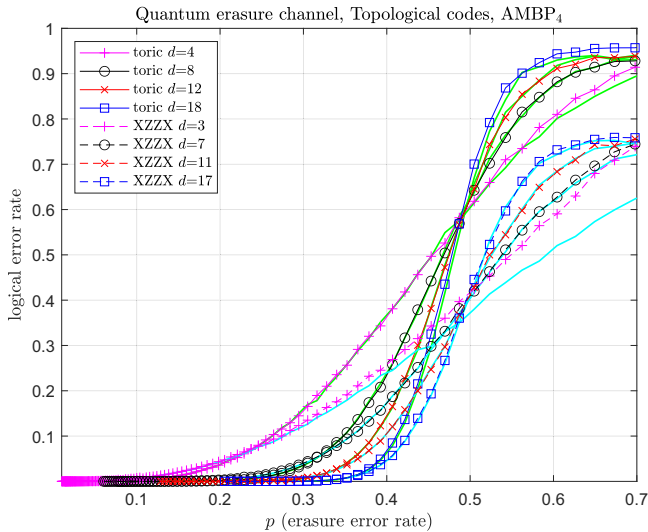


Fig. 9 | Accuracy of AMBP₄ and MLD on toric and XZZX codes. Both decoders achieve a threshold of 0.5. MLD curves are shown in green and cyan for the toric and XZZX codes, respectively. Both decoders have the same accuracy at low error rates (as a log-scale figure in Supplementary Note 7 (part e)).

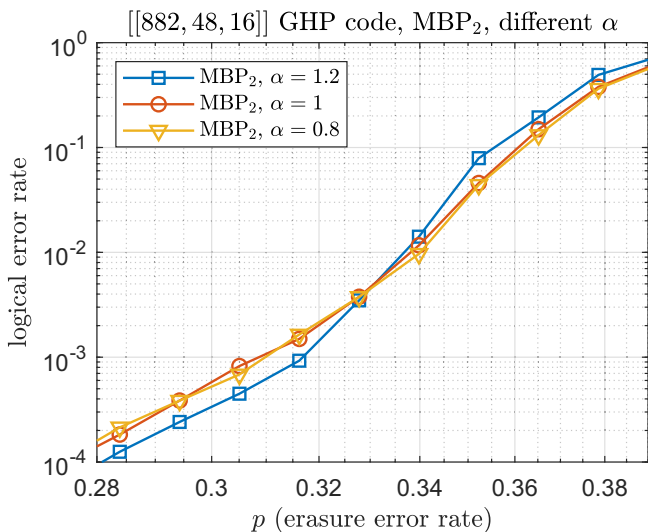


Fig. 10 | Accuracy of MBP₂ on the [[882, 48, 16]] GHP code. Three transition points occur around $p = 0.32, 0.33, 0.34$, for $\alpha = 1.2, 1.0, 0.8$, respectively, establishing a linear relation.

achieving a threshold of $p_{th}^{(MLD)} = 0.44$ at rate $r = 0.118$, closely matching the capacity $r = 1 - 2p$. AMBP decoders also achieve a large threshold of $p_{th}^{(BP)} = 0.368$. For this code family, GD Flip-BP₂ performs almost as well as AMBP₄ when the code length increases, as shown in Fig. 8. This figure also indicate that good curve intersection does not guarantee accuracy at low error rates; see the red line.

Estimated thresholds for rate 0.04 LP codes are $p_{th}^{(MLD)} = 0.48$ and $p_{th}^{(BP)} = 0.405$ (Supplementary Note 7 (part d)).

GD Flip-BP₂ has very low implementation complexity, though its accuracy may depend on the code construction. Figures 6 and 8 show that it can achieve high accuracy with bicycle and LP constructions.

Third, for 2D topological codes, we consider (rotated) toric codes $[[L^2, 2, L]]$ for even L ^{44,46} and (twisted) XZZX codes $[[((d^2 + 1)/2, 1, d)]]$ for odd d ⁴⁷ (see a summary of code structures in [15, Table I and figures]). For 2D codes, the intersection of performance curves, known as the Nishimori point,

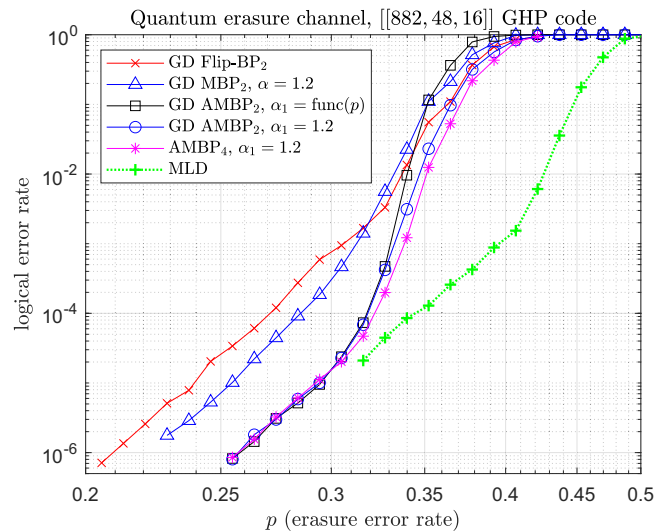


Fig. 11 | Accuracy of several decoders on the [[882, 48, 16]] GHP code. For BP decoders, AMBP-type decoders achieve the best accuracy, and can maintain a low iteration count when using $\alpha_1 = \text{func}(p)$ as in (32). The runtime is compared in Table 2. More HP and GHP results are in Supplementary Note 7 (part a).

provides an estimate of the critical error probability (threshold) for MLD⁴⁵. Among our decoders, AMBP₄ closely matches MLD accuracy for these codes, as shown in Fig. 9. Both decoders have a threshold of 0.5, consistent with^{14–16}. For AMBP, we use a smaller $\alpha_1 = 0.95$ to increase the step-size, handling more erasures while maintaining decoder accuracy. To examine accuracy at low error rates, we replot Fig. 9 on a log scale, showing that AMBP₄ matches MLD on these 2D codes (see Supplementary Note 7 (part e)).

Selection of the α value

We demonstrate how to select a proper α value for MBP or an appropriate initial value α_1 for AMBP. A code with a transition point in the performance curve is convenient for this analysis. Consider the $[[882, 48, 16]]$ GHP code from⁵⁸. The accuracy of MBP₂ with different α values is shown in Fig. 10. Three transition points occur around $p = 0.32, 0.33, 0.34$, for $\alpha = 1.2, 1.0, 0.8$, respectively. This locally linear trend aligns with our expectation (see Supplementary Note 5). We establish a linear relation by $(p, \alpha) = (0.32, 1.2)$ and $(0.34, 0.8 + \text{margin})$, where the margin is 0.1 to prevent α from being too small. The obtained relation is: $\alpha = -15p + 6$. Hence, we propose a function:

$$\alpha = \text{func}(p) := \max(\min(-15p + 6, 1.2), 0.3), \quad (30)$$

where 1.2 and 0.3 are the upper and lower bounds, respectively. Additional simulations show that a wider range of α values does not lead to further improvement. With this α selection strategy (30), MBP achieves accuracy as the lower envelope of the curves in Fig. 10.

For AMBP-type decoders, we evaluate two strategies:

$$\alpha^* \in \{1.2, 1.19, \dots, 0.3\}, \text{ or} \quad (31)$$

$$\alpha^* \in \{\text{func}(p), \text{func}(p) - 0.01, \dots, 0.3\}. \quad (32)$$

We present the decoder accuracy in Fig. 11. As shown, the two strategies exhibit similar accuracy. Both strategies show similar average iteration counts when $p \leq 0.328$ (logical error rate $< 10^{-4}$), as detailed in Table 2. A significant difference arises at higher error rates, such as $p = 0.392$. The strategy in (32) selects a proper α_1 for a given physical error rate p , making AMBP as efficient as MBP. An illustration of the two strategies is shown in Supplementary Note 5 (see S Fig. 5 therein).

Data availability

We provide a software implementation package available at https://github.com/kywukuo/qLDPC_dec.

Received: 31 May 2025; Accepted: 23 February 2026;

Published online: 13 March 2026

References

- Bluvstein, D. et al. Logical quantum processor based on reconfigurable atom arrays. *Nature* **626**, 58 (2024).
- Scholl, P. et al. Erasure conversion in a high-fidelity Rydberg quantum simulator. *Nature* **622**, 273 (2023).
- Ma, S. et al. High-fidelity gates and mid-circuit erasure conversion in an atomic qubit. *Nature* **622**, 279 (2023).
- Wu, Y., Kolkowitz, S., Puri, S. & Thompson, J. D. Erasure conversion for fault-tolerant quantum computing in alkaline earth Rydberg atom arrays. *Nat. Commun.* **13**, 4657 (2022).
- Cong, I. et al. Hardware-efficient, fault-tolerant quantum computation with Rydberg atoms. *Phys. Rev. X* **12**, 021049 (2022).
- Kubica, A. et al. Erasure qubits: Overcoming the T_1 limit in superconducting circuits. *Phys. Rev. X* **13**, 041022 (2023).
- Teoh, J. D. et al. Dual-rail encoding with superconducting cavities. *Proc. Natl. Acad. Sci. (PNAS)* **120**, e2221736120 (2023).
- Levine, H. et al. Demonstrating a long-coherence dual-rail erasure qubit using tunable transmons. *Phys. Rev. X* **14**, 011051 (2024).
- Kang, M., Campbell, W. C. & Brown, K. R. Quantum error correction with metastable states of trapped ions using erasure conversion. *PRX Quantum* **4**, 020358 (2023).
- Lu, C.-Y. et al. Experimental quantum coding against qubit loss error. *Proc. Natl. Acad. Sci. (PNAS)* **105**, 11050 (2008).
- Grassl, M., Beth, T. & Pellizzari, T. Codes for the quantum erasure channel. *Phys. Rev. A* **56**, 33 (1997).
- Bennett, C. H., DiVincenzo, D. P. & Smolin, J. A. Capacities of quantum erasure channels. *Phys. Rev. Lett.* **78**, 3217 (1997).
- Delfosse, N. & Zémor, G. Upper bounds on the rate of low density stabilizer codes for the quantum erasure channel. *Quantum Info Comput.* **13**, 793–826 (2013).
- Stace, T. M., Barrett, S. D. & Doherty, A. C. Thresholds for topological codes in the presence of loss. *Phys. Rev. Lett.* **102**, 200501 (2009).
- Ohzeki, M. Error threshold estimates for surface code with loss of qubits. *Phys. Rev. A* **85**, 060301 (2012).
- Delfosse, N. & Zémor, G. Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel. *Phys. Rev. Res.* **2**, 033042 (2020).
- Bravyi, S. et al. High-threshold and low-overhead fault-tolerant quantum memory. *Nature* **627**, 778 (2024).
- MacKay, D. J. C., Mitchison, G. & McFadden, P. L. Sparse-graph codes for quantum error correction. *IEEE Trans. Inf. Theory* **50**, 2315 (2004).
- Panteleev, P. & Kalachev, G. Quantum LDPC codes with almost linear minimum distance. *IEEE Trans. Inf. Theory* **68**, 213 (2022).
- Williams, V. V. Multiplying matrices faster than Coppersmith–Winograd, in *Proc. Annu. ACM Symp. Theory Comput. (STOC)* 887–898 (2012).
- Le Gall, F. Powers of tensors and fast matrix multiplication, in *Proc. Int. Symp. Symb. Algebraic Comput.* 296–303 (2014).
- Sipser, M. & Spielman, D. A. Expander codes. *IEEE Trans. Inf. Theory* **42**, 1710 (1996).
- Luby, M. G., Mitzenmacher, M., Shokrollahi, M. A. & Spielman, D. A. Efficient erasure correcting codes. *IEEE Trans. Inf. Theory* **47**, 569 (2001).
- Di, C., Proietti, D., Telatar, I. E., Richardson, T. J. & Urbanke, R. L. Finite-length analysis of low-density parity-check codes on the binary erasure channel. *IEEE Trans. Inf. Theory* **48**, 1570 (2002).
- Richardson, T. J. & Urbanke, R. L. Efficient encoding of low-density parity-check codes. *IEEE Trans. Inf. Theory* **47**, 638 (2001).
- MacKay, D. J. C. & Postol, M. S. Weaknesses of Margulis and Ramanujan–Margulis low-density parity-check codes. *Electron. Notes Theor. Comput. Sci.* **74**, 97 (2003).
- Tanner, R. M., Sridhara, D., Sridharan, A., Fuja, T. E. & Costello, D. J. LDPC block and convolutional codes based on circulant matrices. *IEEE Trans. Inf. Theory* **50**, 2966 (2004).
- Shokrollahi, A. Raptor codes. *IEEE Trans. Inf. Theory* **52**, 2551 (2006).
- Popovski, P., Trillingsgaard, K. F., Simeone, O. & Durisi, G. 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. *IEEE Access* **6**, 55765 (2018).
- Zaidi, A. et al. *5G Physical Layer: principles, models and technology components* (Academic Press, 2018).
- Arora, K., Singh, J. & Randhawa, Y. S. A survey on channel coding techniques for 5G wireless networks. *Telecommun. Syst.* **73**, 637 (2020).
- Richardson, T. Error floors of LDPC codes, in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, Vol. 41 1426–1435 (2003).
- Orlitsky, A., Urbanke, R., Viswanathan, K. & Zhang, J. Stopping sets and the girth of Tanner graphs, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)* 2 (2002).
- Aloshious, A. B. & Sarvepalli, P. K. Erasure decoding of two-dimensional color codes. *Phys. Rev. A* **100**, 042312 (2019).
- Solanki, H. M. & Sarvepalli, P. K. Decoding topological subsystem color codes over the erasure channel using gauge fixing. *IEEE Trans. Commun.* **71**, 4181 (2023).
- Lee, S., Mhalla, M. & Savin, V. Trimming decoding of color codes over the quantum erasure channel, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)* 1886–1890 (2020).
- Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. *Introduction to algorithms* (MIT press, Cambridge, MA, 2009).
- Connolly, N., Londe, V., Leverrier, A. & Delfosse, N. Fast erasure decoder for hypergraph product codes. *Quantum* **8**, 1450 (2024).
- Sahay, K., Jin, J., Claes, J., Thompson, J. D. & Puri, S. High-threshold codes for neutral-atom qubits with biased erasure errors. *Phys. Rev. X* **13**, 041013 (2023).
- Edmonds, J. Paths, trees, and flowers. *Can. J. Math.* **17**, 449 (1965).
- Gökdoğan, M., Yao, H. & Pfister, H. Erasure decoding for quantum LDPC codes via belief propagation with guided decimation, in *Proc. Annu. Allerton Conf. Commun., Control, Comput.* 1–8 (2024).
- Yao, H., Gökdoğan, M. & Pfister, H. D. Cluster decomposition for improved erasure decoding of quantum LDPC codes, *IEEE J. Sel. Areas Inf. Theory* <https://doi.org/10.1109/JSait.2025.3578597> (2025).
- Bravyi, S. B. & Kitaev, A. Y. Quantum codes on a lattice with boundary. arXiv preprint quant-ph/9811052 (1998).
- Kitaev, A. Y. Fault-tolerant quantum computation by anyons. *Ann. Phys.* **303**, 2 (2003). preprint arXiv: quant-ph/9707021 (1997).
- Dennis, E., Kitaev, A., Landahl, A. & Preskill, J. Topological quantum memory. *J. Math. Phys.* **43**, 4452 (2002).
- Bombin, H. & Martin-Delgado, M. A. Optimal resources for topological two-dimensional stabilizer codes: Comparative study. *Phys. Rev. A* **76**, 012305 (2007).
- Kovalev, A. A., Dumer, I. & Pryadko, L. P. Design of additive quantum codes via the code-word-stabilized framework. *Phys. Rev. A* **84**, 062319 (2011).
- Terhal, B. M., Hassler, F. & DiVincenzo, D. P. From Majorana fermions to topological order. *Phys. Rev. Lett.* **108**, 260504 (2012).
- Bonilla Ataides, J., Tuckett, D., Bartlett, S., Flammia, S. & Brown, B. The XZZX surface code. *Nat. Commun.* **12**, 1 (2021).
- Tillich, J.-P. & Zémor, G. Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Trans. Inf. Theory* **60**, 1193 (2014).

51. Kovalev, A. A. & Pryadko, L. P. Quantum Kronecker sum-product low-density parity-check codes with finite rate. *Phys. Rev. A* **88**, 012311 (2013).
52. Calderbank, A. R. & Shor, P. W. Good quantum error-correcting codes exist. *Phys. Rev. A* **54**, 1098 (1996).
53. Steane, A. M. Error correcting codes in quantum theory. *Phys. Rev. Lett.* **77**, 793 (1996).
54. Gottesman, D. Fault-tolerant quantum computation with constant overhead. *Quantum Inf. Comput.* **14**, 1338 (2014).
55. Breuckmann, N. P. & Eberhardt, J. N. Quantum low-density parity-check codes. *PRX Quantum* **2**, 040101 (2021).
56. Lai, C.-Y. & Kuo, K.-Y. Harnessing coding theory for reliable network quantum communication. *IEEE Wirel. Commun.* **31**, 82 (2024).
57. Poulin, D. & Chung, Y. On the iterative decoding of sparse quantum codes. *Quantum Inf. Comput.* **8**, 987 (2008).
58. Panteleev, P. & Kalachev, G. Degenerate quantum LDPC codes with good finite length performance. *Quantum* **5**, 585 (2021).
59. Kuo, K.-Y. & Lai, C.-Y. Exploiting degeneracy in belief propagation decoding of quantum codes. *npj Quantum Inf.* **8**, art. no. 111 (2022a).
60. Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci. (PNAS)* **81**, 3088 (1984).
61. Hopfield, J. J. & Tank, D. W. "neural" computation of decisions in optimization problems. *Biol. Cybern.* **52**, 141 (1985).
62. Hopfield, J. J. & Tank, D. W. Computing with neural circuits: A model. *Science* **233**, 625 (1986).
63. Kuo, K.-Y. & Lai, C.-Y. Refined belief propagation decoding of sparse-graph quantum codes. *IEEE J. Sel. Areas Inf. Theory* **1**, 487 (2020).
64. Ouyang, Y. Permutation-invariant quantum codes. *Phys. Rev. A* **90**, 062317 (2014).
65. Ouyang, Y. Permutation-invariant quantum coding for quantum deletion channels, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)* (2021) pp. 1499–1503.
66. Bluvstein, D. et al. A quantum processor based on coherent transport of entangled atom arrays. *Nature* **604**, 451 (2022).
67. Calderbank, A. R., Rains, E. M., Shor, P. W. & Sloane, N. J. A. Quantum error correction via codes over GF(4). *IEEE Trans. Inf. Theory* **44**, 1369 (1998).
68. Gottesman, D. *Stabilizer codes and quantum error correction*, Ph.D. thesis, California Institute of Technology, Pasadena, CA USA (1997).
69. Kuo, K.-Y. & Lai, C.-Y. Generalized quantum data-syndrome codes and belief propagation decoding for phenomenological noise. *IEEE Trans. Inf. Theory* **71**, 1824 (2025).
70. Kuo, K.-Y. & Lai, C.-Y. Fault-tolerant belief propagation for practical quantum memory. arXiv preprint arXiv:2409.18689v1 (2024).
71. Gallager, R. G. *Low-Density Parity-Check Codes*, no. 21 in Research Monograph Series (MIT Press, 1963).
72. Tanner, R. A recursive approach to low complexity codes. *IEEE Trans. Inf. Theory* **27**, 533 (1981).
73. Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference* (Morgan Kaufmann, 1988).
74. MacKay, D. J. C. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inf. Theory* **45**, 399 (1999).
75. Kschischang, F. R., Frey, B. J. & Loeliger, H.-A. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* **47**, 498 (2001).
76. Chen, J. & Fossorier, M. P. C. Density evolution for two improved BP-based decoding algorithms of LDPC codes. *IEEE Commun. Lett.* **6**, 208 (2002a).
77. Chen, J., Dholakia, A., Eleftheriou, E., Fossorier, M. P. C. & Hu, X.-Y. Reduced-complexity decoding of LDPC codes. *IEEE Trans. Commun.* **53**, 1288 (2005).
78. Davey, M. & MacKay, D. Low-density parity check codes over GF(q). *IEEE Commun. Lett.* **2**, 165 (1998).
79. McEliece, R. J., MacKay, D. J. C. & Cheng, J.-F. Turbo decoding as an instance of Pearl's "belief propagation" algorithm. *IEEE J. Sel. Areas Commun.* **16**, 140 (1998).
80. Lai, C.-Y. & Kuo, K.-Y. Log-domain decoding of quantum LDPC codes over binary finite fields. *IEEE Trans. Quantum Eng.* **2**, art. no. 2103615 (2021).
81. Fossorier, M. & Lin, S. Soft-decision decoding of linear block codes based on ordered statistics. *IEEE Trans. Inf. Theory* **41**, 1379 (1995).
82. Roffe, J., White, D. R., Burton, S. & Campbell, E. T. Decoding across the quantum low-density parity-check code landscape. *Phys. Rev. Res.* **2**, 043423 (2020).
83. Ruskai, M. B. Pauli exchange errors in quantum computation. *Phys. Rev. Lett.* **85**, 194 (2000).
84. Pollatsek, H. & Ruskai, M. B. Permutationally invariant codes for quantum error correction. *Linear Algebra Appl.* **392**, 255 (2004).
85. Ouyang, Y. & Fitzsimons, J. Permutation-invariant codes encoding more than one qubit. *Phys. Rev. A* **93**, 042340 (2016).
86. Ouyang, Y. Permutation-invariant qudit codes from polynomials. *Linear Algebra Appl.* **532**, 43 (2017).
87. Aydin, A., Alekseyev, M. A. & Barg, A. A family of permutationally invariant quantum codes. *Quantum* **8**, 1321 (2024).
88. Shibayama, T. & Hagiwara, M. Permutation-invariant quantum codes for deletion errors, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)* 1493–1498 (2021).
89. Nakayama, A. & Hagiwara, M. The first quantum error-correcting code for single deletion errors. *IEICE Commun. Express* **9**, 100 (2020).
90. Hagiwara, M. & Nakayama, A. A four-qubits code that is a quantum deletion error-correcting code with the optimal length, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)* 1870–1874 (2020).
91. Delfosse, N. & Nickerson, N. H. Almost-linear time decoding algorithm for topological codes. *Quantum* **5**, 595 (2021).
92. Hagiwara, M., Fossorier, M. P. C. & Imai, H. Fixed initialization decoding of LDPC codes over a binary symmetric channel. *IEEE Trans. Inf. Theory* **58**, 2321 (2012).
93. Chen, J. & Fossorier, M. P. C. Near optimum universal belief propagation based decoding of low-density parity check codes. *IEEE Trans. Commun.* **50**, 406 (2002b).
94. Valls, J., Garcia-Herrero, F., Raveendran, N. & Vasić, B. Syndrome-based min-sum vs OSD-0 decoders: FPGA implementation and analysis for quantum LDPC codes. *IEEE Access* **9**, 138734 (2021).
95. Bennett, C. H., DiVincenzo, D. P., Smolin, J. A. & Wootters, W. K. Mixed-state entanglement and quantum error correction. *Phys. Rev. A* **54**, 3824 (1996).
96. Komoto, D. & Kasai, K. Quantum error correction near the coding theoretical bound. *npj Quantum Inf.* **11**, 154 (2025).
97. deMarti iOlius, A., Fuentes, P., Orús, R., Crespo, P. M. & Martinez, J. E. Decoding algorithms for surface codes. *Quantum* **8**, 1498 (2024).
98. Freire, B. C. A., Régent, F.-M. L. & Leverrier, A. Quantum Maxwell erasure decoder for qLDPC codes. arXiv preprint arXiv:2601.10713 <https://doi.org/10.48550/arXiv.2601.10713> (2026).
99. Wang, C., Harrington, J. & Preskill, J. Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Ann. Phys.* **303**, 31 (2003).
100. Raussendorf, R., Harrington, J. & Goyal, K. Topological fault-tolerance in cluster state quantum computation. *N. J. Phys.* **9**, 199 (2007).
101. Wang, D. S., Fowler, A. G., Stephens, A. M. & Hollenberg, L. C. L. Threshold error rates for the toric and planar codes. *Quantum Inf. Comput.* **10**, 456 (2010).

102. Fowler, A. G., Whiteside, A. C. & Hollenberg, L. C. Towards practical classical processing for the surface code. *Phys. Rev. Lett.* **108**, 180501 (2012).
103. Gidney, C. Stim: a fast stabilizer circuit simulator. *Quantum* **5**, 497 (2021).
104. Skoric, L., Browne, D. E., Barnes, K. M., Gillespie, N. I. & Campbell, E. T. Parallel window decoding enables scalable fault tolerant quantum computation. *Nat. Commun.* **14**, 7040 (2023).
105. Gong, A., Cammerer, S. & Renes, J. M. Toward low-latency iterative decoding of QLDPC codes under circuit-level noise. arXiv preprint arXiv:2403.18901 <https://doi.org/10.48550/arXiv.2403.18901> (2024).
106. Hillmann, T. et al. Localized statistics decoding for quantum low-density parity-check codes. *Nat. Commun.* **16**, 8214 (2025).
107. Müller, T. et al. Improved belief propagation is sufficient for real-time decoding of quantum memory. arXiv preprint arXiv:2506.01779 <https://doi.org/10.48550/arXiv.2506.01779> (2025).
108. Gu, S., Retzker, A. & Kubica, A. Fault-tolerant quantum architectures based on erasure qubits. *Phys. Rev. Res.* **7**, 013249 (2025a).
109. Gu, S., Vaknin, Y., Retzker, A. & Kubica, A. Optimizing quantum error-correction protocols with erasure qubits. *PRX Quantum* **6**, 040354 (2025b).
110. Perrin, H., Jandura, S. & Pupillo, G. Quantum error correction resilient against atom loss. *Quantum* **9**, 1884 (2025).
111. Baranes, G. et al. Leveraging qubit loss detection in fault-tolerant quantum algorithms. *Phys. Rev. X* **16**, 011002 (2026).
112. Lucas, R., Bossert, M. & Breibach, M. On iterative soft-decision decoding of linear binary block codes and product codes. *IEEE J. Sel. Areas Commun.* **16**, 276 (1998).
113. Fossorier, M. P. C. Quasi-cyclic low-density parity-check codes from circulant permutation matrices. *IEEE Trans. Inf. Theory* **50**, 1788 (2004).
114. Malema, G. A. *Low-density parity-check codes: construction and implementation*, Ph.D. thesis, University of Adelaide, SA, Australia (2007).
115. Kuo, K.-Y. & Lai, C.-Y. Comparison of 2D topological codes and their decoding performances, in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)* 186–191 (2022).
116. Wiedemann, D. Solving sparse linear equations over finite fields. *IEEE Trans. Inf. Theory* **32**, 54 (1986).

Acknowledgements

K.K. and Y.O. acknowledge support from EPSRC (Grant No. EP/W028115/1). Y.O. also acknowledges support from the EPSRC funded QCI3 Hub under Grant No. EP/Z53318X/1.

Author contributions

All authors wrote the manuscript. KY Kuo performed the numerics of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41534-026-01212-3>.

Correspondence and requests for materials should be addressed to Kao-Yueh Kuo or Yingkai Ouyang.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2026