applied sciences

MDPI

Article

Fault-Tolerant Hybrid Decoder for Quantum Surface Codes on Probabilistic Inference and Topological Clustering

Xingyu Qiao [1], Xiaoxuan Xu [2], Hongyang Ma [1] and Tianhui Qiu [1,*]

[1] School of Science, Qingdao University of Technology, Qingdao 266033, China
[2] School of Information and Control Engineering, Qingdao University of Technology, Qingdao 266033, China
* Correspondence: qiutianhui@qut.edu.cn

Abstract

Quantum error correction is a prerequisite for quantum computing; however, the performance critically depends on the accuracy of the decoding algorithm. To address these challenges, we propose a hybrid decoding architecture, BP + UF + BP. The protocol initiates with a truncated global BP stage to extract probabilistic gradients without requiring full convergence. This soft information guides a reliability-based Union-Find (UF) algorithm to prioritize high-likelihood error mechanisms. Finally, a local subgraph BP refinement maximizes correction accuracy. Numerical simulations on rotated surface codes under circuit-level depolarizing noise demonstrate a fault-tolerance threshold of approximately 0.72%. This significantly outperforms standard Minimum Weight Perfect Matching (MWPM) and Union-Find (UF) baselines. Notably, our method significantly reduces the logical error rate compared to the conventional decoders. With its empirically near-linear scaling under fixed iteration, the proposed architecture presents a scalable solution for real-time fault-tolerant quantum computing.

Keywords: quantum error correction; surface code; fault tolerance; belief propagation; Union-Find

1. Introduction

As conventional computing technologies gradually reach their performance limits, quantum computing has become a focal point for handling large-scale data and other problems, achieving exponential improvements in computation time compared to conventional computing techniques. These unique advantages of quantum computers are rooted in the fundamental core principles of quantum mechanical theory, which embraces essential inherent properties such as quantum superposition, quantum entanglement, and quantum interference [1–3]. The principle of quantum superposition allows qubits to reside in a superposition of several distinct states simultaneously, significantly enhancing computational capability and speed; quantum entanglement allows for highly parallel computation between qubits, further improving information processing efficiency [4–6]. Recent advancements have demonstrated the profound potential of these properties in data-intensive domains [7,8]. However, qubits are highly susceptible to changes under the influence of noise and environmental interference, making quantum error correction (QEC) an essential technical step toward implementing reliable quantum algorithms [9,10]. Quantum error correction is founded on the key concept of introducing redundancy to form a stabilizer-based error correction code framework. In quantum error correction coding,

Check for updates

Received: 3 February 2026
Revised: 26 February 2026
Accepted: 6 March 2026
Published: 8 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

redundancy is implemented via encoding physical qubits onto logical qubits to improve the robustness of quantum information in quantum systems [11].

Among topological quantum error correction schemes, the Surface Code is one of the most widely studied and utilized options for quantum error correction tasks, distinguished by its high thresholds, local measurement attributes, and effective error mitigation capacity [11,12]. One of the crucial processing components in quantum error correction codes is the decoder, which extracts error syndromes obtained from stabilizer measurements and then analyzes the locations where data bits have errors [13–15]. An algorithm that can quickly and accurately infer error locations based on error syndromes is a key component in building decoders [16,17]. In the span of recent research endeavors, the continuous evolution of quantum computing systems has driven iterative upgrades to surface code decoding techniques. Several decoders stand out for their remarkable performance gains: these include the Minimum Weight Perfect Matching (MWPM) algorithm, Belief Propagation (BP) algorithm, Union-Find (UF) algorithm, BP + OSD, and advanced decoders developed on the foundation of neural network innovations [18,19]. Ref. [20] introduces MBP, which incorporates memory effects into BP for surface codes, achieving a threshold significantly higher than standard BP. In ref. [21], as a classical iterative decoding approach, BP was employed to optimize the key input error model associated with conventional surface code decoder schemes (including MWPM and UF). To enhance performance, some decoders invoke post-processing algorithms to resolve decoding issues when the primary algorithm fails. Ref. [22] proposes a synergistic decoding scheme that combines the BP algorithm with the ordered statistics algorithm, achieving a performance boost of several orders of magnitude. Ref. [23] employs the localized statistics decoding algorithm and adopts matrix inversion approaches to address the decoding obstacles existing in the post-processing stage. However, a key limitation of these high-precision decoders is their computational cost; for instance, BP-OSD and standard MWPM often exhibit cubic worst-case complexity $O(n^3)$. In contrast, the Union-Find (UF) decoder is designed for speed, with a worst-case complexity of $O(n)$ [23–25]. Therefore, the UF decoder can be combined with other high-precision but higher-complexity decoders in a hybrid decoding scheme to improve overall decoding performance.

In this study, we designed a decoder based on the fusion of BP and UF, which we call BP + UF + BP. First, the initial round of the BP algorithm is applied to the rotated surface code. The soft information output from BP is mapped to the input of UF for use in the syndrome verification phase of the UF [26,27]. The second round of the BP algorithm operates on the spanning forest during the syndrome verification phase of the UF decoder, inferring potentially erroneous qubits from the edge probabilities of computed nodes and correcting them. This decoding strategy combines the characteristics of BP and UF. It effectively handles errors in qubits. Specifically, it reduces the logical error rate when decoding rotated surface codes. This leads to improved decoding accuracy.

2. Background

2.1. Rotated Surface Code

Stabilizer codes are a pivotal class of quantum error correction codes defined by an Abelian group $S = \langle S_1, \ldots, S_m \rangle$ of mutually commuting, Hermitian Pauli operators [28]. The code space $\mathcal{C}$ consists of quantum states invariant under S:

$$\mathcal{C} = \{|\psi\rangle \mid S_i |\psi\rangle = +1 |\psi\rangle, \forall i = 1, 2, \ldots, m\} \quad (1)$$

Measuring the stabilizer operators yields a syndrome vector. Outcomes of $+1$ indicate that the system resides within the code space, while -1 signals errors requiring correction [12].

Surface codes topologically encode information across a two-dimensional lattice [28–31]. Data qubits occupy lattice vertices, while ancilla qubits measure stabilizer parity, utilizing global topological properties to suppress local physical errors [32–35]. Our study focuses on the rotated surface code architecture (Figure 1), composed of X-type and Z-type stabilizers defined as

$$X_v = \prod_{i \in v} \sigma_i^x, \quad Z_m = \prod_{j \in m} \sigma_j^z \tag{2}$$

where σ_i^x and σ_j^z are Pauli operators acting on data qubits. As shown in Figure 1, X-stabilizers (red) detect Z errors, while Z-stabilizers (blue) detect X errors [36–39]. Physical errors are modeled as Pauli operators that disrupt the system via anticommutation. Specifically, a Pauli-X error flips adjacent Z-stabilizer outcomes to -1. Due to the lattice topology, individual errors can form continuous chains where stabilizers along the path flip twice, leaving only the stabilizers at the chain's boundaries flipped. These isolated defects define the decoding problem, identifying the most probable error chains that pair these defects to neutralize the logical noise.

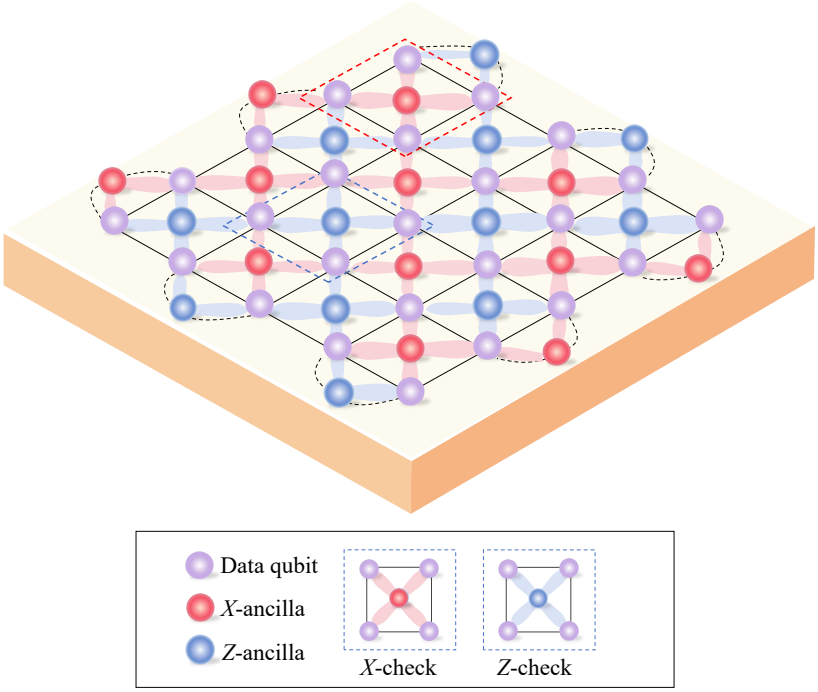


Figure 1. Topology of a rotated surface code with $d = 5$. Data qubits (purple circles) are positioned at lattice vertices. Stabilizer measurement units are represented by squares. X-stabilizers (red squares) denote X-type stabilizers (composed of products of X-operators), which measure the X-parity of adjacent data qubits and detect Z errors; Z-stabilizers (blue squares) denote Z-type stabilizers (composed of products of Z-operators), which measure the Z-parity of adjacent data qubits and detect X errors.

2.2. Graph Construction

In the context of circuit-level noise, a single round of stabilizer measurement is inherently unreliable due to measurement and idle errors [40–42]. To address this, the decoding problem is formulated in terms of detectors rather than raw stabilizer outcomes. A detector represents a temporal parity mismatch, defined as the difference between consecutive measurements of the same stabilizer. Consequently, each detector acts as a unique space-time coordinate within the decoding volume [43–46]. Let $\mathcal{D}$ denote the set of all such measurement events. The decoder's input is the syndrome vector $s = \{s_d\}_{d \in \mathcal{D}}$, where $s_d \in \{0, 1\}$. Here, $s_d = 1$ signifies that the detector d has been triggered.

To rigorously bridge the Belief Propagation and Union-Find decoders based on this syndrome, we distinguish between two graph representations: the Tanner Graph and the Soft Decoding Graph used for topological clustering [47,48]. We first define the Tanner graph, denoted as $G_{\text{Tanner}} = (\mathcal{D} \cup \mathcal{F}, E)$. Here, the vertex set is exactly the set of detectors $\mathcal{D}$ defined above. $\mathcal{F}$ represents the set of elementary fault mechanisms, such as Pauli errors on data qubits, measurement flips, or correlated circuit faults. In this bipartite graph, an edge connects a detector $d \in \mathcal{D}$ to a fault $f \in \mathcal{F}$ if the fault triggers that detector. The BP decoder operates directly on this structure to compute the marginal posterior probability, denoted as $q_f = \Pr(f = 1|s)$, for each fault mechanism. Crucially, this representation preserves hyperedges where a single fault triggers more than two detectors, thereby avoiding the information loss typical of standard matching graphs.

In contrast, the Union-Find algorithm requires a graph-like structure. To accommodate the surface code topology, we construct two separate soft decoding graphs $G_{\text{soft}} = (\mathcal{D} \cup \{v_B\}, E_{\text{soft}})$. The vertex set includes the same detectors $\mathcal{D}$ plus a virtual boundary node v_B to handle open boundary conditions. The edge set E_{soft} consists of graph-like error events, where each edge connects either two detectors or one detector to the boundary [49,50].

To project the high-fidelity information from BP onto G_{soft}, we employ a specific mapping strategy. For each edge e in the soft decoding graph, we define a subset $\mathcal{F}(e) \subseteq \mathcal{F}$. This subset contains all fault mechanisms from the Tanner graph that decompose into edge e. The effective error probability $\tilde{p}_e$ for edge e corresponds to the likelihood that an odd number of these constituent faults have occurred. Assuming the posteriors approximated by BP are independent, this is derived using the Piling-up Lemma:

$$\tilde{p}_e = \frac{1}{2}\left(1 - \prod_{f \in \mathcal{F}(e)} (1 - 2q_f)\right) \tag{3}$$

The quantitie q_f in Equation (3) denotes the approximate marginal posterior belief produced by the first-stage sum-product BP on the Tanner graph under the assumed circuit-level depolarizing noise model, rather than a parameter introduced through post-hoc calibration. Equation (3) then provides a fixed, model-based transformation, under the stated independence approximation, from these BP beliefs to effective edge probabilities $\tilde{p}_e$ for the subsequent topological clustering and local refinement stages. Although this independence approximation is not exact under circuit-level noise, it provides a consistent relative weighting of candidate edges for downstream processing. The same mapping is applied uniformly across all reported code distances and physical error rates. Finally, the edge weight for the Union-Find decoder is set as $w_e = -\ln(\tilde{p}_e)$. This ensures that the topological cluster growth is guided by the accurate probabilistic landscape extracted from the full circuit model [51].

3. Decoding the Rotated Surface Code

In this section, we propose a hybrid decoding architecture, referred to as the BP + UF + BP decoder, which integrates the granular probabilistic inference of Belief Propagation (BP) with the topological robustness of the Union-Find (UF) decoder. As discussed in Section 2, circuit-level noise introduces complex spacetime correlations and hyperedge structures into the Tanner graph. These features pose significant hurdles for conventional decoders. Traditional decoders such as MWPM or Union-Find (UF) typically rely on a matching graph that decomposes these hyperedges, leading to unavoidable information loss. Conversely, while Belief Propagation (BP) can effectively model such intricate correlations, it suffers from intractable convergence issues on loopy graphs. To address these dual limitations, our BP + UF + BP decoding pipeline is divided into three sequential stages:

(1) overall probabilistic inference via Belief Propagation, (2) clustering and subgraph extraction, and (3) local refinement via subgraph Belief Propagation. Below, we describe the mathematical formulation and implementation details of each stage.

3.1. Belief Propagation

Belief Propagation (BP), alternatively termed the sum-product algorithm, is an iterative message-passing approach engineered to enable highly efficient inferential computations over graphical model structures, including factor graphs [52]. For quantum error correction scenarios, BP is used to approximate the maximum a posteriori (MAP) estimation of the error vector e. This estimation is based on the observed syndrome s. The algorithm operates on the Tanner graph $G_{\text{Tanner}} = (V_v \cup V_c, E)$ constructed in Section 2.2. In this graph, variable nodes represent physical error mechanisms. Check nodes represent the stabilizer constraints. The fundamental principle of BP involves the exchange of beliefs between neighboring nodes along the edges of the graph. These beliefs are probabilistic messages. A message sent from a variable node to a check node reflects the local probability that the qubit is erroneous [53]. This probability is based on information received from other parity checks. Conversely, a message from a check node to a variable node expresses the likelihood of an error on that qubit. This likelihood ensures that the parity constraint is satisfied given the states of other connected qubits. BP is proven to converge to exact marginal probabilities on cycle-free graphs. However, the Tanner graphs for surface codes contain numerous short cycles. They also exhibit high degeneracy. Degeneracy means that multiple distinct error patterns result in the same syndrome. Consequently, standard BP acts as a heuristic approximation when applied to quantum codes. Specifically, rather than providing a true global MAP solution, the first-stage sum-product BP outputs approximate marginal posterior probabilities to serve as soft information for the subsequent decoding stages. Short loops introduce correlations that violate the independence assumption of the message updates. This violation can lead to convergence failures, oscillations, or the split belief phenomenon. In a split belief state, the decoder is indecisive between two or more logically equivalent error chains.

As illustrated in Figure 2, this causes the marginal probabilities to stagnate around 0.5 (maximum entropy). Despite these challenges, BP effectively captures high-order correlations and soft information regarding circuit-level noise. Hard-decision decoders like MWPM often discard this information. Therefore, BP is an ideal precursor for extracting reliability information to guide subsequent decoding stages.

3.2. Overall Probabilistic Inference via Belief Propagation

To effectively mitigate the complex correlations introduced by circuit-level noise, we first execute the Belief Propagation algorithm on the Tanner graph constructed in Section 2.2. In contrast to conventional matching-based decoders that rely solely on binary syndrome information, BP exploits the prior probability distribution provided by the physical noise model [52]. Through iterative message passing, it infers the marginal posterior probability for each fault mechanism. This soft information constitutes the critical foundation for the subsequent topological clustering stage, and the input to the decoder includes the parity check matrix H, the observed detector syndrome vector s, and the vector of prior error probabilities p determined by the physical noise model [54]. To enhance numerical stability and simplify computations, probability information is transformed into the Log-Likelihood Ratio (LLR) domain. For each variable node $f \in \mathcal{F}$ representing an independent elementary fault mechanism, its initial LLR is defined as the natural logarithm of the ratio between the probability of the fault not occurring and the probability of it occurring.

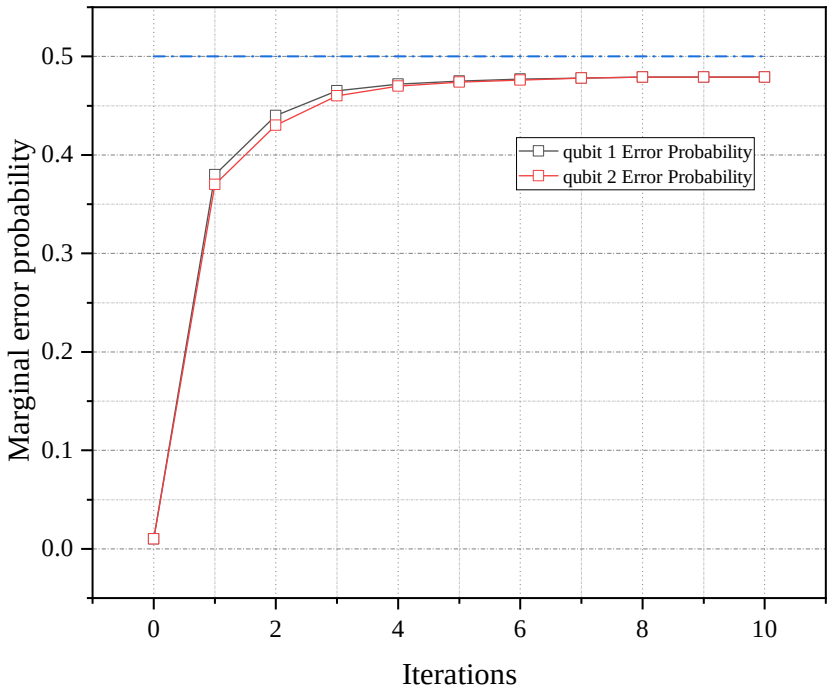


Figure 2. Convergence behavior of marginal error probabilities during BP iterations. The curves illustrate that beliefs for qubits involved in degenerate error patterns tend to converge to 0.5. This stagnation indicates the decoder's inability to distinguish between logically equivalent error classes solely through message passing.

$$L_f^{(0)} = \ln\left(\frac{1-p_f}{p_f}\right) \tag{4}$$

Here, p_f is the physical probability of the fault mechanism. A larger magnitude of $L_f^{(0)}$ indicates a lower degree of confidence that an error has occurred at this node. The Belief Propagation algorithm updates beliefs by iteratively exchanging information between variable nodes and check nodes [43]. Let $M_{f\to d}^{(t)}$ denote the message sent from variable node f to check node d during the t-th iteration. In the initial iteration, messages are initialized to the prior values. In subsequent iterations, adhering to the extrinsic information principle, the message sent from variable node f to check node d is composed of its initial LLR summed with the messages received from all neighboring check nodes, excluding d [21]:

$$M_{f\to d}^{(t)} = L_f^{(0)} + \sum_{d'\in\mathcal{N}(f)\setminus\{d\}} M_{d'\to f}^{(t-1)} \tag{5}$$

where $\mathcal{N}(f)$ represents the set of check nodes neighboring variable node f. Conversely, check node d is responsible for enforcing parity check constraints. The feedback message $M_{d\to f}^{(t)}$ sent to variable node f is computed using the hyperbolic tangent rule:

$$M_{d\to f}^{(t)} = (-1)^{s_d}\cdot 2\tanh^{-1}\left(\prod_{f'\in\mathcal{N}(d)\setminus\{f\}} \tanh\left(\frac{M_{f'\to d}^{(t)}}{2}\right)\right) \tag{6}$$

Here, $s_d \in \{0,1\}$ represents the observed value of detector d. The factor $(-1)^{s_d}$ is crucial for injecting syndrome information into the decoding process. When a detector is triggered, this term flips the sign of the message. This mechanism reduces the confidence that the associated variable nodes are in the no-error state and drives the belief toward the error state.

While Belief Propagation is exact on acyclic graphs, standard algorithms face severe convergence challenges on surface code Tanner graphs. These graphs contain numerous short loops and degenerate errors. Short loops cause information to echo infinitely within the cycle and violate the independence assumption of probability estimation. Furthermore, highly symmetric degenerate errors often lead to periodic oscillations of beliefs between multiple equivalent solutions. These are known as trapping sets and they prevent convergence to a deterministic hard decision. To address these issues, we do not aim for the complete convergence of the algorithm. Instead, we adopt an early stopping strategy. We constrain the algorithm to a fixed small number of iterations T_{global}. Although the algorithm may not have fully converged at this stage, the output posterior LLRs are sufficient to break the symmetry of the initial distribution. This establishes a probabilistic gradient that reflects the topological structure of the errors. At the end of the iterations, the posterior LLR for each variable node is computed as follows:

$$L_f^{(\text{post})} = L_f^{(0)} + \sum_{d \in \mathcal{N}(f)} M_{d \to f}^{(T_{\text{global}})} \tag{7}$$

Nodes located on the actual error chain typically exhibit smaller $L_f^{(\text{post})}$ values. This probabilistic gradient provides a high-fidelity reliability metric for the subsequent weighted topological clustering stage. It enables the clustering process to prioritize covering the most probable error regions and effectively circumvents the high logical error rates associated with direct Belief Propagation hard decisions.

3.3. Clustering and Subgraph Extraction

In this section, we project the information from the Tanner graph onto the Soft Decoding Graph [55]. We denote this graph as G_{soft}. The edges in this graph represent graph-like error events. However, a single edge e in the Soft Decoding Graph often corresponds to a collection of multiple circuit-level fault mechanisms. We define e as the set of all faults in the Tanner graph that map to edge e [56].

We need to calculate the aggregate probability that edge e is in an error state. This occurs if an odd number of the constituent faults in e have triggered. We use the posterior probabilities q_f obtained from the first Belief Propagation stage. By applying the mapping strategy established in Section 2.2, the effective edge probability $\tilde{p}_e$ is derived using the Piling-up Lemma as defined in Equation (3). This formula provides a method to compress the soft information. We then convert this probability into a scalar weight w_e for the Union-Find decoder. We use a negative logarithmic mapping to ensure that higher error probabilities correspond to lower edge weights:

$$w_e = -\ln(\tilde{p}_e + \epsilon) \tag{8}$$

where ϵ is a small constant to avoid numerical singularities. This weighting scheme ensures that the clustering algorithm preferentially grows along edges that are most likely to be erroneous.

At this stage, we employ a modified weighted Union-Find algorithm to identify the topological structure of errors. An example of cluster growth is illustrated in Figure 3. Instead of standard Union-Find decoders that strictly require a cycle-free forest structure, our method introduces an augmentation strategy. This strategy retains not only the minimum spanning forest as the major structure but also the additional high-probability augmentation edges that form local cycles. This design preserves competing error paths, allowing the subsequent local Belief Propagation stage to resolve problems through probabilistic inference rather than arbitrary topological breaking.

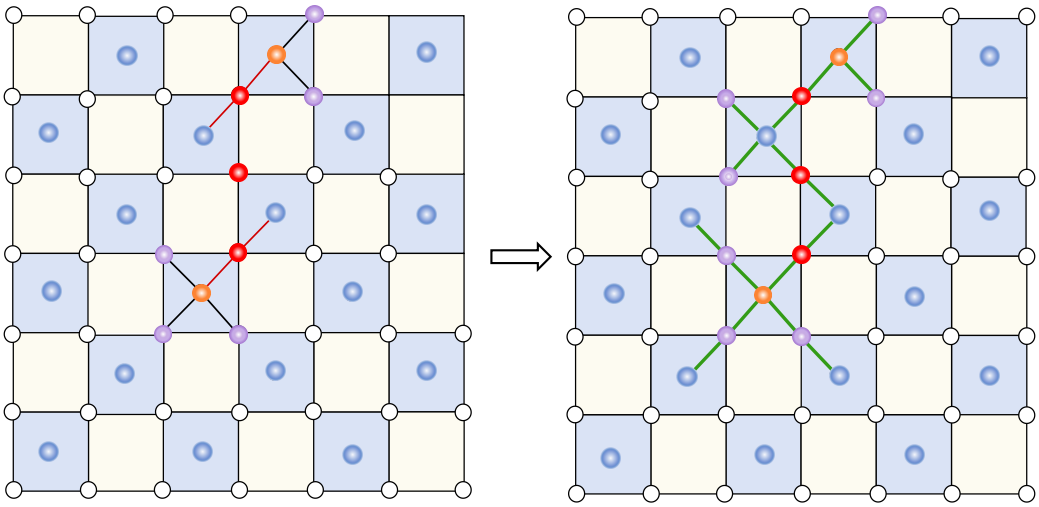


Figure 3. The clustering growth process is guided based on the soft information output from the first Belief Propagation.

We operate on the Soft Decoding Graph G_{soft}. We initialize a cluster C for every detector node d with a non-trivial syndrome ($s_d = 1$). The virtual boundary node v_B is treated as a pre-existing valid cluster. For each cluster C, we maintain its syndrome parity $P(C)$ and boundary connectivity status $B(C)$. The clustering process handles edges in ascending order of their weights w_e, which corresponds to the descending order of error probability. Let E_{back} denote the set of major edges that merge disjoint clusters, and let E_{aug} denote the set of augmentation edges that connect vertices within the same cluster. For each edge $e = (u, v)$, let C_u and C_v represent the clusters that contain vertices u and v, respectively. We first execute the merge step. When $C_u \neq C_v$, the edge e is classified as a major edge ($e \in E_{\text{back}}$).

We then merge the two clusters, updating the parity as $P(C_{new}) = P(C_u) \oplus P(C_v)$ and the boundary status as $B(C_{new}) = B(C_u) \vee B(C_v)$. Subsequently, we perform the augmentation step. When $C_u = C_v$, the edge forms a cycle. Instead of discarding it immediately, we check its effective error probability $\tilde{p}_e$. Given that $\tau \in (0, 1)$ is a tunable threshold, we retain this edge as an augmentation edge ($e \in E_{\text{aug}}$) if $\tilde{p}_e \geq \tau$. Here, τ is a heuristic threshold (hyperparameter) that controls the sparsity of the UF-augmented local refinement subgraph and therefore governs a complexity–accuracy trade-off. Setting τ closer to 1 yields a sparser, more tree-like subgraph (reducing the computational overhead of local BP but potentially discarding valid candidate paths), whereas setting τ closer to 0 retains more redundant edges (increasing the local search space and refinement cost). In addition, the choice of τ affects support selection for local refinement but does not alter the definition of the first-stage BP posterior beliefs or the topological validity criterion of the Union-Find clustering. Accordingly, τ acts as a structural parameter for support selection rather than a probabilistic calibration parameter. This heuristic strategy incorporates high-probability error mechanisms into the subgraph, even if they are topologically redundant, thereby preserving potentially relevant local correlations for the subsequent BP refinement. This heuristic strategy will incorporate high-probability error mechanisms into the subgraph, even if they are topologically redundant, and provide a more sufficient search space for local refinement. The process terminates when all active clusters meet the validity criteria, which is defined as follows:

$$\Phi(C) = 1 \iff (P(C) = 0) \vee (B(C) = 1) \tag{9}$$

The final candidate edge set is given by $E_{\text{cand}} = E_{\text{back}} \cup E_{\text{aug}}$. Algorithm 1 summarizes the entire procedure presented above.

Algorithm 1: Clustering growth with soft information

Input: Soft Graph G_{soft}, Syndrome s, Weights $\{w_e\}$, Probabilities $\{\tilde{p}_e\}$, Threshold τ.

Output: Candidate edge set E_{cand}.

Initialize: ;
  For each detector d with $s_d = 1$: Create cluster C, set $P(C) = 1$, $B(C) = 0$. ;
  For boundary node v_B: Create cluster C, set $P(C) = 0$, $B(C) = 1$. ;
  $N_{\text{invalid}} \leftarrow$ count of clusters where $(P(C) = 1 \wedge B(C) = 0)$. ;
$Q \leftarrow$ Sort all edges in E_{soft} by w_e ascending;
$E_{\text{back}} \leftarrow \emptyset$, $E_{\text{aug}} \leftarrow \emptyset$;
while $N_{\text{invalid}} > 0$ ***and*** Q *is not empty* **do**
  $e = (u, v) \leftarrow Q.\text{pop}()$;
  $r_u \leftarrow \text{Find}(u)$, $r_v \leftarrow \text{Find}(v)$;
  if $r_u \neq r_v$ **then**
    $Valid_u \leftarrow (P(r_u) = 0) \vee (B(r_u) = 1)$;
    $Valid_v \leftarrow (P(r_v) = 0) \vee (B(r_v) = 1)$;
    if not $Valid_u$ **then** $N_{\text{invalid}} \leftarrow N_{\text{invalid}} - 1$;
    if not $Valid_v$ **then** $N_{\text{invalid}} \leftarrow N_{\text{invalid}} - 1$;
    $E_{\text{back}} \leftarrow E_{\text{back}} \cup \{e\}$;
    $r_{\text{new}} \leftarrow \text{Union}(r_u, r_v)$;
    $P(r_{\text{new}}) \leftarrow P(r_u) \oplus P(r_v)$;
    $B(r_{\text{new}}) \leftarrow B(r_u) \vee B(r_v)$;
    $Valid_{\text{new}} \leftarrow (P(r_{\text{new}}) = 0) \vee (B(r_{\text{new}}) = 1)$;
    if not $Valid_{\text{new}}$ **then** $N_{\text{invalid}} \leftarrow N_{\text{invalid}} + 1$;
  else
    if $\tilde{p}_e \geq \tau$ **then**
      $E_{\text{aug}} \leftarrow E_{\text{aug}} \cup \{e\}$;
return $E_{\text{cand}} \leftarrow E_{\text{back}} \cup E_{\text{aug}}$;

3.4. Local Refinement via Subgraph Belief Propagation

The final stage performs inference on the subgraph identified by the augmented Union-Find step. The major edge set ensures topological connectivity; the augmentation edge set captures ambiguities in errors by introducing local loops. The Subgraph Belief Propagation algorithm resolves these ambiguities to identify the most probable error pattern.

We first map the edges from the soft decoding graph back to their corresponding fault mechanisms. The candidate fault set is defined as $\mathcal{F}_{\text{sub}} = \bigcup_{e \in E_{\text{cand}}} \mathcal{F}(e)$. Next, we collect all detectors adjacent to these faults: $\mathcal{D}_{\text{sub}} = \bigcup_{f \in \mathcal{F}_{\text{sub}}} N(f)$. We then build the Tanner graph $G^{\text{sub}}_{\text{Tanner}} = (\mathcal{D}_{\text{sub}} \cup \mathcal{F}_{\text{sub}}, E_{\text{sub}})$, where the edge set $E_{\text{sub}} = \{(d, f) \in E : d \in \mathcal{D}_{\text{sub}}, f \in \mathcal{F}_{\text{sub}}\}$ connects detectors to their associated faults. Then, we reinitialize the Belief Propagation algorithm on this restricted graph. We reset the prior probability p_f for each fault $f \in \mathcal{F}_{\text{sub}}$ to its initial value from the physical noise model. We apply numerical clipping to the prior probabilities to ensure numerical stability. We clamp p_f within the interval $[\delta, 1 - \delta]$. Here δ is a small constant such as 10^{-4}. This prevents the Log-Likelihood Ratios from diverging to infinity. The initial LLR is computed as follows Equation (4). We initialize all messages between variable nodes and check nodes to zero. We then execute the sum-product algorithm on the restricted subgraph [57]. From a probabilistic perspective, since

this second-stage BP operates on a restricted subgraph determined by the UF clustering, it should be interpreted as approximate conditional posterior inference on a UF-restricted support, i.e., a local posterior refinement within a reduced feasible region, rather than an exact MAP solver. The message update rules follow the standard equations defined in Section 3.1. However, we introduce a damping factor $\eta \in [0,1]$ to suppress oscillations caused by the local short loops. Let m_{new} be the computed message value for the current iteration. Let m_{old} be the message value from the previous iteration.

$$m_{\text{update}} = \eta \cdot m_{\text{new}} + (1-\eta) \cdot m_{\text{old}} \tag{10}$$

This damping mechanism improves the convergence rate on the augmented subgraph. We run the algorithm for a fixed number of iterations T_{local}. This number is typically proportional to the diameter of the clusters.

After the iterations complete, we compute the local posterior LLR $L_f^{(\text{post})}$ for each fault node. We derive the local posterior probability $q_f^{(\text{local})}$.

$$q_f^{(\text{local})} \approx \frac{1}{1+\exp\left(L_f^{(\text{post})}\right)} \tag{11}$$

We perform a hard decision to estimate the error vector $\hat{e}$. We assign $\hat{e}_f = 1$ if the posterior probability $q_f^{(\text{local})}$ exceeds 0.5. Otherwise, we assign $\hat{e}_f = 0$. This operation is equivalent to minimizing the linear cost defined by the posterior LLRs. Finally, we perform a global consistency check. We verify if the estimated error explains the syndrome s.

$$H \cdot \hat{e} \equiv s \tag{12}$$

In most cases, the Subgraph BP converges to a valid solution. However, residual loops or numerical errors may strictly prevent convergence. In such rare events, we trigger a fallback mechanism. We discard the augmentation edges and execute a restricted Peeling Decoder on the spanning forest defined by E_{back}. This guarantees that the decoder always returns a valid correction.

Algorithm 2 outlines the complete execution flow of the proposed decoder, which is designed to progressively optimize the error estimation through three distinct stages.

Algorithm 2: BP + UF + BP Hybrid Decoder

Input: Syndrome s, Matrix H, Priors p, Soft Graph G_{soft}.
Output: Estimated error $\hat{e}$.
$(\hat{e}, q) \leftarrow \text{Global_BP}(H, s, p, T_{\text{global}})$;
if $H \cdot \hat{e} = s$ **then return** $\hat{e}$;
Compute edge probabilities $\tilde{p}$ from q using Equation (3);
Compute weights w: $w_e \leftarrow -\ln(\tilde{p}_e + \epsilon)$ for all $e \in E_{\text{soft}}$;
$E_{\text{cand}} \leftarrow \text{Augmented_Union_Find}(G_{\text{soft}}, s, w)$;
Construct subgraph H_{sub} from edges E_{cand};
$(\hat{e}_{\text{local}}, _) \leftarrow \text{Subgraph_BP}(H_{\text{sub}}, s, p, T_{\text{local}})$;
return $\hat{e}_{\text{local}}$;

4. Experiment Results

4.1. Simulation Setup

We performed experiments using Monte Carlo sampling on rotated surface-code patches with code distances 5, 7, 9, and 11. The simulations were performed using Stim, employing a conventional circuit-level depolarizing noise model. We used Stim as the circuit-level stabilizer simulator because it efficiently generated detection-event (syndrome) samples under the circuit-level depolarizing noise model considered in this work. Importantly, Stim was used only for data generation: all evaluated decoders were implemented independently of Stim and processed the exact same shared datasets under an identical noise model and circuit schedule, ensuring an internally consistent and fair comparison across decoders. We additionally performed a limited cross-check on selected code distances ($d \in \{5, 7\}$) using an independent Clifford-circuit simulator and found logical error rates consistent with the Stim-based results within statistical uncertainty. To accurately reflect spatiotemporal error spreading, we executed cycles of noisy syndrome extraction, concluding with one round of ideal measurement [58]. The noise strength was defined by a physical error parameter, distributed across operations as follows:

- Single-Qubit Operations: Every single-qubit gate underwent a depolarizing channel where Pauli errors X, Y, Z occurred with equal probability $p/3$.
- Two-Qubit Operations: Entangling gates were succeeded by a two-qubit depolarizing channel. An error was selected uniformly from the 15 non-identity Pauli combinations, each assigned a probability of $p/15$.
- SPAM: Both state preparation and measurement steps were subject to bit-flip errors with rate p.
- Idle Qubits: Qubits typically idled during operations on others; this was modeled by a depolarizing channel with probability $p/3$.

To evaluate performance, we compared our method against three baseline decoders: Union-Find (UF), MWPM, and Belief Propagation with Ordered Statistics Decoding (BP + OSD-0). It is important to state explicitly that the MWPM baseline evaluated in this work corresponded to a standard pairwise matching decoder without explicit correlated-noise modeling. Under circuit-level noise, a single physical fault may induce correlated detection events involving more than two detectors due to fault propagation through entangling gates. Standard MWPM operates on a pairwise matching graph and therefore represents such events through a pairwise graph approximation, which may not fully preserve the original correlation structure. All decoders processed the same set of syndrome data to ensure a fair comparison. Unlike conventional BP + OSD architectures that require extensive iterations for convergence, our approach utilized a lightweight Union-Find post-processor. This structural advantage allowed us to relax the convergence criteria. We limited the initial global BP stage to a fixed schedule of $T_{\text{global}} = 30$ iterations. This depth was sufficient to extract high-fidelity soft information without becoming trapped in short cycles. Edge weights for the subsequent UF stage were derived via Equation (8). Finally, the local subgraph BP stage was limited to $T_{\text{local}} = 15$ iterations. This configuration effectively balanced high decoding accuracy with low runtime complexity.

4.2. Threshold Analysis

The error correction threshold is a pivotal metric for assessing quantum error-correcting code efficacy, referring to the critical physical error probability level at which the coding scheme can reliably suppress error propagation under defined physical error condi-

tions. The correlation between the logical error probability p_L, physical error probability p, and code distance d is often described by the following formula:

$$p_L(p,d) \approx C\left(\frac{p}{p_{th}}\right)^{\frac{d+1}{2}} \tag{13}$$

Consequently, the logical error rate p_L shows an exponential decrease with the code distance d. This signifies effective error suppression. In contrast, once p exceeds p_{th}, the error correction capability collapses.

Figure 4 illustrates the relationship between the logical error rate p_L and the physical error rate p, where the code distances were set as $d \in \{5,7,9,11\}$. As shown in the figure, the proposed BP + UF + BP decoder achieved a fault-tolerance threshold equivalent to approximately 0.72%. Under ideal conditions, if the physical error probability fell beneath this threshold value, elevating the code distance yielded a measurable reduction in the logical error rate [13]. For comparison, under the same simulation conditions of the circuit-level depolarizing noise, the threshold of the standard MWPM decoder was generally around 0.6%. These results demonstrate that the proposed hybrid decoding architecture can effectively improve decoder performance, thus yielding a higher fault-tolerance threshold.

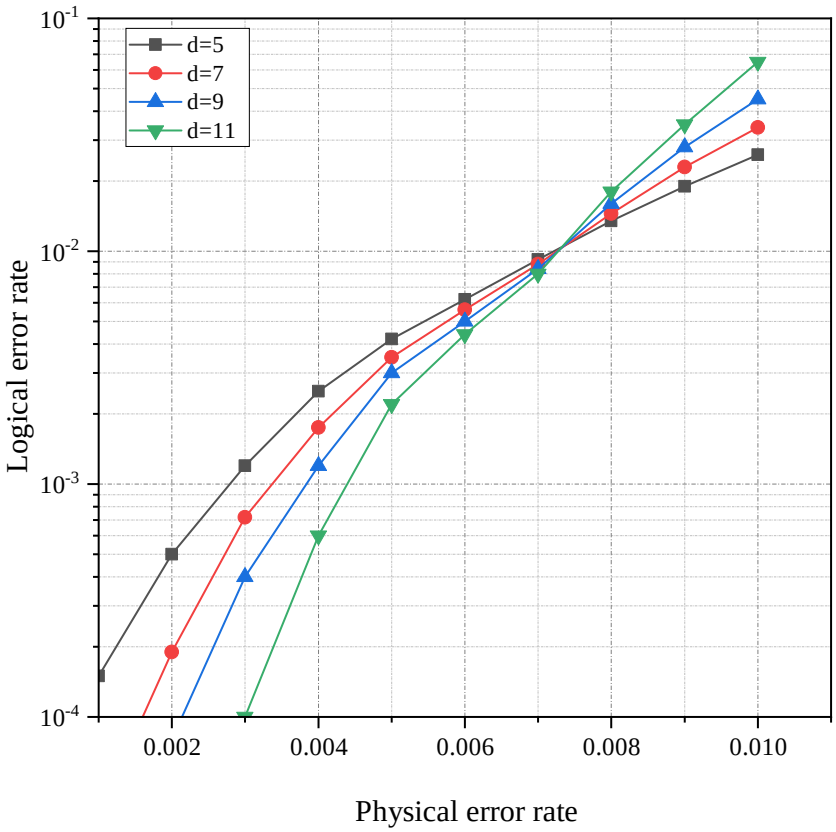


Figure 4. The figure displays the variation in the logical error rate of the decoder for the rotated surface code at different code distances.

To assess whether the reported fault-tolerance threshold of approximately 0.72% could be influenced by hyperparameter choices, we evaluated five independent parameter configurations by varying the number of global BP iterations T_{global}, the number of local BP iterations T_{local}, and the augmentation threshold τ, while keeping all other experimental conditions identical to those described in Section 4.1. For each configuration, the threshold p_{th} was extracted using the same finite-size scaling procedure (Equation (13)) via least-squares regression, and the statistical uncertainty was estimated by bootstrap resampling of

the Monte Carlo trials. The results are summarized in Table 1. Across all five configurations, the estimated threshold remained within the range [0.70%, 0.74%], with overlapping uncertainty intervals, and differed from the baseline value by at most 0.02 percentage points. These observations support the interpretation that the reported ∼0.72% threshold is robust to moderate variations in the iteration budgets and the augmentation threshold across the tested configurations. The baseline configuration was retained as a fixed operating point for consistency (and to balance decoding cost and performance), rather than being selected through post-hoc maximization of p_{th}.

Table 1. Hyperparameter sensitivity analysis of the BP + UF + BP decoder. The threshold p_{th} was estimated under five independent parameter configurations. The baseline configuration used throughout this study is highlighted in gray.

Configuration	T_{global}	T_{local}	τ	p_{th} (%)
Baseline (this work)	30	15	0.5	0.72 ± 0.02
Group A	20	10	0.5	0.70 ± 0.02
Group B	40	20	0.5	0.72 ± 0.02
Group C	30	15	0.25	0.71 ± 0.02
Group D	30	15	0.75	0.70 ± 0.02

We performed a comparative analysis of logical error rates against the baseline decoders. Figure 5 presents the results. To further assess efficacy, we benchmarked our decoder against three baselines: Union-Find (UF), standard MWPM, and BP + OSD-0. The comprehensive logical error rate comparison for code distances $d \in \{5, 7, 9, 11\}$ is presented in Figure 5. Our decoder consistently outperformed both UF and MWPM across all distances. This advantage widened significantly at lower physical error rates. For example, at $d = 11$ and $p = 2 \times 10^{-3}$, our method yielded a significant reduction in logical error propagation, exhibiting considerably better performance than both the UF and MWPM schemes. This gain stemmed from the effective utilization of soft information, which helps resolve hyperedge-like error mechanisms that limit matching-based methods. Furthermore, the proposed architecture achieved logical error rates on par with, or slightly better than, the computationally intensive BP + OSD decoder. This demonstrates that reliability-guided clustering, when refined by a localized secondary BP, yields correction capabilities equivalent to global matrix inversion.

4.3. Decoding Runtime

To evaluate the scalability and time potential of the decoder, we measured the average decoding time per shot as a function of the code distance d. For each code distance $d \in \{5, 7, 9, 11\}$, we constructed a rotated surface code circuit with $T = d$ rounds of syndrome extraction. A fixed physical error rate of $p = 10^{-3}$ was used to generate a consistent set of detector syndromes via Stim. To ensure a rigorous comparison, we strictly measured the decode time only, excluding the overhead of syndrome generation and graph initialization. To establish a comprehensive performance benchmark, we evaluated four distinct decoders under identical conditions. As a baseline for topological decoding efficiency, we implemented the standard Union-Find (UF) decoder using an unweighted growth strategy. For a rigorous comparison, we employed the MWPM decoder. To represent high-precision probabilistic decoding, we included the BP + OSD-0 decoder, configuring the BP stage with a maximum of 100 iterations to ensure convergence.

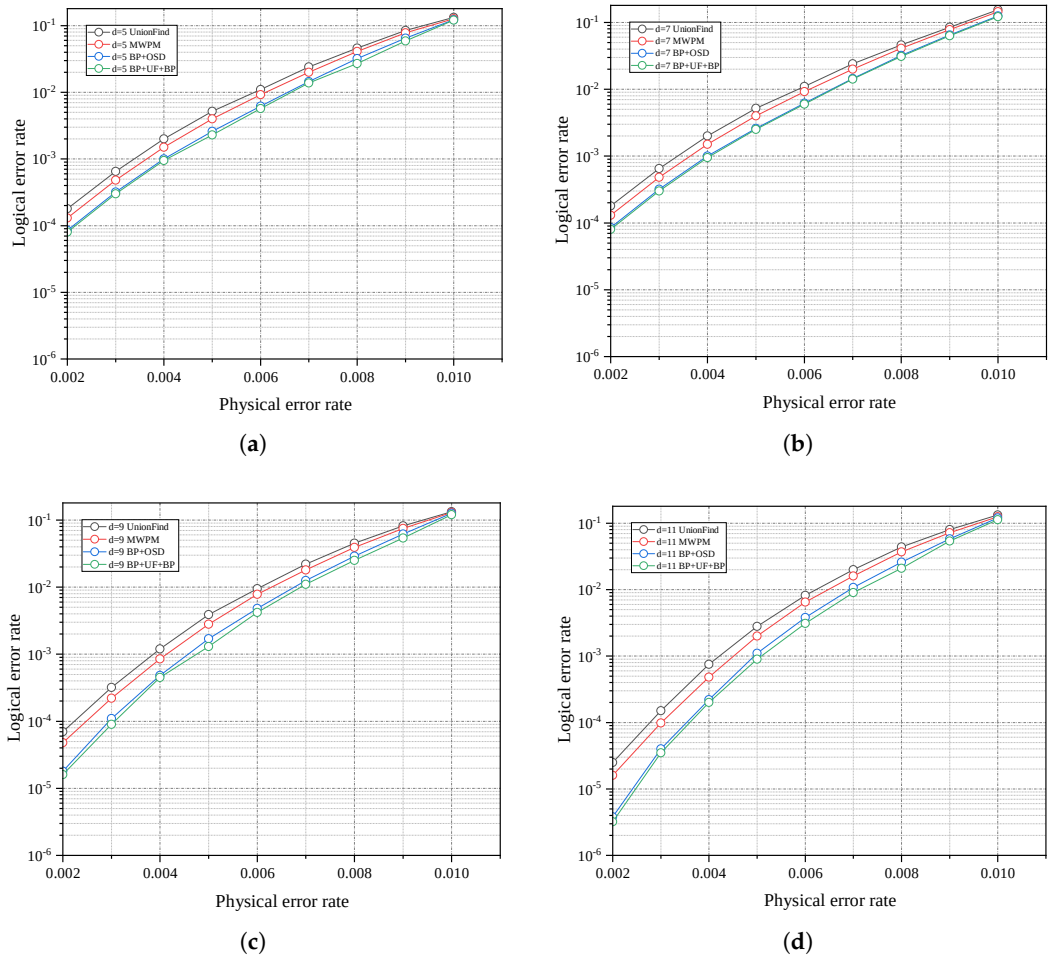


Figure 5. Simulated syndrome rounds with the number equal to the distance of each code. (**a**) Performance comparison of the Union-Find decoder (black), MWPM decoder (red), BP + OSD decoder (blue), and the decoder proposed in this work (green) under standard circuit-level depolarizing noise (d = 5). (**b**–**d**) Performance comparisons respectively for code distances of 7, 9, and 11.

As shown in Figure 6, the runtime of our BP + UF + BP decoder (in blue curve) exhibited a polynomial scaling slope nearly parallel to that of the UF and MWPM decoders. This confirms that our algorithm ensured that the secondary BP stage added only a constant-factor overhead rather than increasing the complexity class.

Our proposed architecture adopted the Union-Find (UF) decoder for topological clustering. This decoder has extremely low computational overhead and empirically near-linear scaling under a fixed iteration schedule. This structural advantage allowed us to significantly reduce the computational load of the probabilistic inference module. In contrast, the BP + OSD decoder (green curve) showed a steep increase in runtime, reflecting the cubic complexity $\mathcal{O}(n^3)$ of the matrix inversion step. At $d = 11$, while BP + OSD required nearly 100 ms per shot, our method completed decoding in less than 1 ms. Runtime experiments verified that the proposed method effectively bridges the performance gap between the speed advantage of topological decoders and the accuracy advantage of ordered-statistics decoding methods.

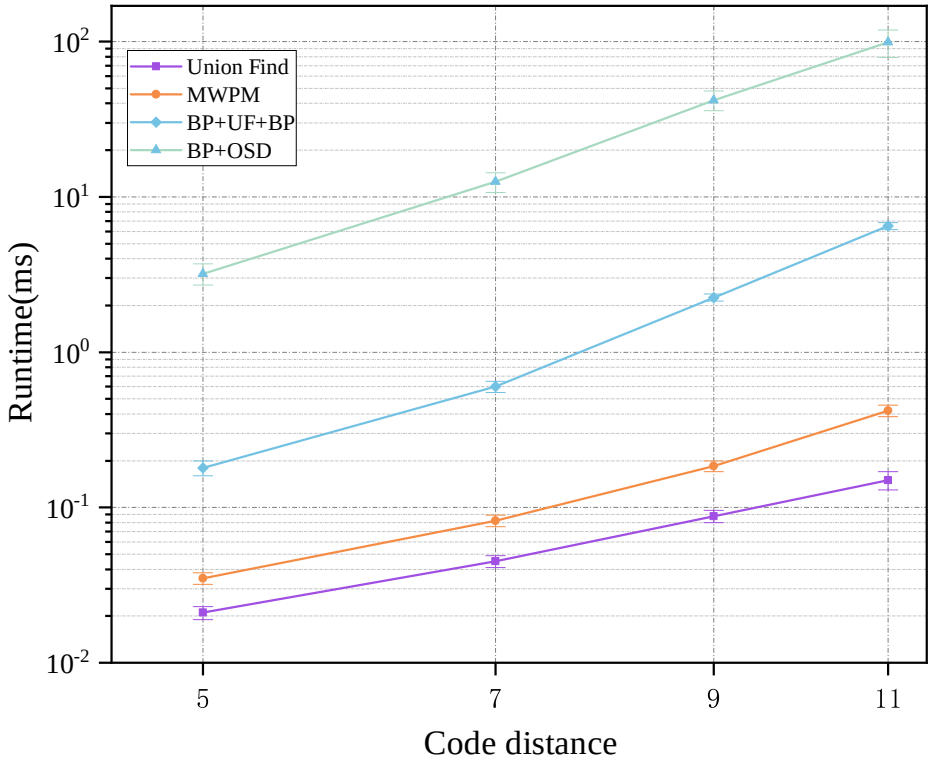


Figure 6. Average decoding time per sample versus code distance. The proposed BP + UF + BP decoder (in blue) exhibited an empirically near-linear scaling under a fixed iteration schedule that was comparable to that of the Union-Find decoder (in purple) and the Minimum Weight Perfect Matching (MWPM) decoder (in orange). In contrast, the runtime of the BP + OSD decoder (in green) increased sharply as the code distance grew. Compared with the BP + OSD decoder, our method achieved a higher threshold as well as a faster decoding speed.

5. Conclusions

In this work, we proposed a hybrid decoding architecture, BP + UF + BP. By using probabilistic soft information from a global Belief Propagation stage to guide the topological growth of Union-Find (UF) clusters and then refining these clusters with a local secondary BP, this method effectively balances high error correction accuracy and low computational latency in fault-tolerant quantum computing. Numerical simulations on rotated surface codes with distances d = 5, 7, 9, and 11 show that this architecture achieves a fault-tolerance threshold of approximately 0.72%, outperforming traditional matching-based decoders. Despite these promising results, several limitations need further investigation. First, although the algorithm complexity is low, processing and storing high-precision soft information incurs higher memory and bandwidth overheads than hard-decision decoders such as standard Union-Find. Second, for surface codes with very large distances, trapping sets and cycles remain unavoidable challenges for the truncated iteration strategy. In the next step, future research will focus on improving the performance of both the probabilistic and topological components of the decoder. For the probabilistic part, we plan to integrate advanced BP variants, such as neural belief propagation or decoders with memory effects, to extract higher-confidence priors with fewer iterations. For the topological part, optimizing clustering algorithms will enable fully parallel cluster growth.

Author Contributions: Writing—original draft, X.Q.; data curation, X.Q.; formal analysis, X.X., T.Q. and H.M.; simulation computation, X.Q.; methodology, X.Q. and H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Shandong Provincial Natural Science Foundation (grant nos. ZR2024LLZ004 and ZR2025LLZ007) and the Key R&D Program of Shandong Province (grant nos. 2025CXGC010901 and 2023CXGC010901).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Erhard, M.; Krenn, M.; Zeilinger, A. Advances in high-dimensional quantum entanglement. *Nat. Rev. Phys.* **2020**, *2*, 365–381. [CrossRef]
2. He, X.; Gong, L.; Zhou, N. Image Denoising with Hybrid Classical-Quantum Convolutional Neural Network. *Memetic Comput.* **2025**, *17*, 23. [CrossRef]
3. Gong, L.H.; Chen, Y.Q.; Zhou, S.; Zeng, Q.W. Dual discriminators quantum generation adversarial network based on quantum convolutional neural network. *Adv. Quantum Technol.* **2025**, *8*, e2500224. [CrossRef]
4. deMarti iOlius, A.; Fuentes, P.; Orús, R.; Crespo, P.M.; Etxezarreta Martinez, J. Decoding algorithms for surface codes. *Quantum* **2024**, *8*, 1498. [CrossRef]
5. Jayashankar, A.; Mandayam, P. Quantum error correction: Noise-adapted techniques and applications. *J. Indian Inst. Sci.* **2023**, *103*, 497–512. [CrossRef]
6. Ding, Y.F.; Li, Z.; Zhou, N.R. Quantum generative adversarial network based on the quantum Born machine. *Adv. Eng. Inform.* **2025**, *68*, 103622. [CrossRef]
7. Gong, C.; Wen, Z.Y.; Deng, Y.W.; Zhou, N.R.; Zeng, Q.W. Unrolled generative adversarial network for continuous distributions under hybrid quantum-classical model. *Laser Phys. Lett.* **2024**, *21*, 125207. [CrossRef]
8. Njoku, T.K. Quantum software engineering: Algorithm design, error mitigation, and compiler optimization for fault-tolerant quantum computing. *Int. J. Comput. Appl. Technol. Res.* **2025**, *14*, 30–42.
9. Dür, W.; Skotiniotis, M.; Froewis, F.; Kraus, B. Improved Quantum Metrology Using Quantum Error Correction. *Phys. Rev. Lett.* **2014**, *112*, 080801. [CrossRef]
10. Huang, S.Y.; An, W.J.; Zhang, D.S. Image classification and adversarial robustness analysis based on hybrid quantum–classical convolutional neural network. *Opt. Commun.* **2023**, *533*, 129287. [CrossRef]
11. Cai, W.; Mu, X.; Wang, W.; Zhou, J.; Ma, Y.; Pan, X.; Sun, L. Protecting entanglement between logical qubits via quantum error correction. *Nat. Phys.* **2024**, *20*, 1022–1026. [CrossRef]
12. Stephens, A.M. Fault-tolerant thresholds for quantum error correction with the surface code. *Phys. Rev. A* **2014**, *89*, 022321. [CrossRef]
13. Fan, Z.; Tian, C.; Xiao, B. A Hybrid Architecture Decoder Integrating Kolmogorov-Arnold Network and Transformer for Decoding Rotating Surface Codes. *Int. J. Theor. Phys.* **2025**, *64*, 313. [CrossRef]
14. Gottesman, D. Stabilizer Codes and Quantum Error Correction. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 1997.
15. Wang, H.W.; Xue, Y.J.; Ma, Y.L.; Hua, N.; Ma, H.Y. Determination of quantum toric error correction code threshold using convolutional neural network decoders. *Chin. Phys. B* **2025**, *31*, 010303. [CrossRef]
16. Chamberland, C.; Goncalves, L.; Sivarajah, P.; Peterson, E.; Grimberg, S. Techniques for combining fast local decoders with global decoders under circuit-level noise. *Quantum Sci. Technol.* **2023**, *8*, 045011. [CrossRef]
17. Li, A.; Li, F.; Gan, Q.; Ma, H. Convolutional-Neural-Network-Based Hexagonal Quantum Error Correction Decoder. *Appl. Sci.* **2023**, *13*, 9689. [CrossRef]
18. Fu, M.; Tian, C.; Fan, Z.; Ma, H. Resource-Efficient Decoding of Topological Color Codes via Neural-Guided Union-Find Optimization. *Appl. Sci.* **2025**, *15*, 8937. [CrossRef]
19. Varsamopoulos, S.; Bertels, K.; Almudever, C.G. Comparing neural network-based decoders for the surface code. *IEEE Trans. Comput.* **2019**, *69*, 300–311. [CrossRef]
20. Kuo, K.Y.; Lai, C.Y. Exploiting degeneracy in belief propagation decoding of quantum codes. *npj Quantum Inf.* **2022**, *8*, 111. [CrossRef]
21. Higgott, O.; Breuckmann, N.P. Improved single-shot decoding of higher-dimensional hypergraph-product codes. *PRX Quantum* **2023**, *4*, 020332. [CrossRef]
22. Panteleev, P.; Kalachev, G. Degenerate quantum LDPC codes with good finite length performance. *Quantum* **2021**, *5*, 585. [CrossRef]

23. Hillmann, T.; Berent, L.; Quintavalle, A.O.; Eisert, J.; Wille, R.; Roffe, J. Localized statistics decoding for quantum low-density parity-check codes. *Nat. Commun.* **2025**, *16*, 8214. [CrossRef] [PubMed]
24. Delfosse, N.; Nickerson, N.H. Almost-linear time decoding algorithm for topological codes. *Quantum* **2021**, *5*, 595. [CrossRef]
25. Wu, Y.; Liyanage, N.; Zhong, L. An interpretation of union-find decoder on weighted graphs. *arXiv* **2022**, arXiv:2211.03288. [CrossRef]
26. Jung, H.; Ha, J. Topological blocking decoder for surface codes. *Phys. Rev. A* **2025**, *111*, 042424. [CrossRef]
27. Higgott, O.; Bohdanowicz, T.C.; Kubica, A.; Flammia, S.T.; Campbell, E.T. Improved decoding of circuit noise and fragile boundaries of tailored surface codes. *Phys. Rev. X* **2023**, *13*, 031007. [CrossRef]
28. Terhal, B.M. Quantum error correction for quantum memories. *Rev. Mod. Phys.* **2015**, *87*, 307–346. [CrossRef]
29. Mondal, A.; Parhi, K.K. Quantum circuits for stabilizer error correcting codes: A tutorial. *IEEE Circuits Syst. Mag.* **2024**, *24*, 33–51. [CrossRef]
30. Nigg, D.; Mueller, M.; Martinez, E.A.; Schindler, P.; Hennrich, M.; Monz, T.; Blatt, R. Quantum computations on a topologically encoded qubit. *Science* **2014**, *345*, 302–305. [CrossRef]
31. Gambetta, J.M.; Chow, J.M.; Steffen, M. Building logical qubits in a superconducting quantum computing system. *npj Quantum Inf.* **2017**, *3*, 2. [CrossRef]
32. Qu, Y.J.; Chen, Z.; Wang, W.J.; Ma, H.Y. Approximate error correction scheme for three-dimensional surface codes based reinforcement learning. *Chin. Phys. B* **2023**, *32*, 100307. [CrossRef]
33. Dennis, E.; Kitaev, A.; Landahl, A.; Preskill, J. Topological quantum memory. *J. Math. Phys.* **2002**, *43*, 4452–4505. [CrossRef]
34. Xue, Y.J.; Wang, H.W.; Tian, Y.B.; Wang, Y.N.; Wang, Y.X.; Wang, S.M. Quantum information protection scheme based on reinforcement learning for periodic surface codes. *Quantum Eng.* **2022**, *2022*, 7643871. [CrossRef]
35. Dangniam, N.; Han, Y.G.; Zhu, H. Optimal verification of stabilizer states. *Phys. Rev. Res.* **2020**, *2*, 043323. [CrossRef]
36. Tuckett, D.K.; Darmawan, A.S.; Chubb, C.T.; Bravyi, S.; Bartlett, S.D.; Flammia, S.T. Tailoring surface codes for highly biased noise. *Phys. Rev. X* **2019**, *9*, 041031. [CrossRef]
37. Kuo, K.Y.; Lai, C.Y. Comparison of 2D topological codes and their decoding performances. In Proceedings of the 2022 IEEE International Symposium on Information Theory (ISIT), Espoo, Finland, 26 June–1 July 2022; pp. 186–191.
38. Huang, E.; Pesah, A.; Chubb, C.T.; Vasmer, M.; Dua, A. Tailoring three-dimensional topological codes for biased noise. *PRX Quantum* **2023**, *4*, 030338. [CrossRef]
39. Chen, Y.; Fan, Z.; Wang, H.; Tian, C.; Ma, H. Probing Threshold Behavior of Adaptive Cascaded Quantum Codes Under Variable Biased Noise for Practical Fault-Tolerant Quantum Computing. *Electronics* **2026**, *15*, 436. [CrossRef]
40. Berent, L.; Burgholzer, L.; Derks, P.J.H.; Eisert, J.; Wille, R. Decoding quantum color codes with MaxSAT. *Quantum* **2024**, *8*, 1506. [CrossRef]
41. Meister, N.; Pattison, C.A.; Preskill, J. Efficient soft-output decoders for the surface code. *arXiv* **2022**, arXiv:2405.07433.
42. Liu, Y.H.; Poulin, D. Neural belief-propagation decoders for quantum error-correcting codes. *Phys. Rev. Lett.* **2019**, *122*, 200501. [CrossRef]
43. Yan, D.D.; Fan, X.K.; Chen, Z.Y.; Ma, H.Y. Low-loss belief propagation decoder with Tanner graph in quantum error-correction codes. *Chin. Phys. B* **2022**, *31*, 010304. [CrossRef]
44. Old, J.; Rispler, M. Generalized belief propagation algorithms for decoding of surface codes. *Quantum* **2023**, *7*, 1037. [CrossRef]
45. Zhang, B.; Yao, H.; Pfister, H.D. Belief Propagation Decoding on a Sparsified Graph Ensemble of the Surface Code. In Proceedings of the 2025 IEEE International Symposium on Information Theory (ISIT), Ann Arbor, MI, USA, 22–27 June 2025; pp. 1–6.
46. Huang, S.; Newman, M.; Brown, K.R. Fault-tolerant weighted union-find decoding on the toric code. *Phys. Rev. A* **2020**, *102*, 012419.
47. Löbl, M.C.; Chen, S.X.; Paesani, S.; Sørensen, A.S. Breadth-first graph traversal union-find decoder. *arXiv* **2024**, arXiv:2407.15988.
48. Zass, R.; Shashua, A. Probabilistic graph and hypergraph matching. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
49. Piveteau, C.; Renes, J.M. Quantum message-passing algorithm for optimal and efficient decoding. *Quantum* **2022**, *6*, 784. [CrossRef]
50. Hu, G.; Ouyang, W.; Lu, C.Y.; Lin, C.; Zhong, H.S. Efficient and universal neural-network decoder for stabilizer-based quantum error correction. *arXiv* **2025**, arXiv:2502.19971.
51. Bravyi, S.; Cross, A.W.; Gambetta, J.M.; Maslov, D.; Rall, P.; Yoder, T.J. High-threshold and low-overhead fault-tolerant quantum memory. *Nature* **2024**, *627*, 778–782.
52. Müller, T.; Alexander, T.; Beverland, M.E.; Bühler, M.; Johnson, B.R.; Maurer, T.; Vandeth, D. Improved belief propagation is sufficient for real-time decoding of quantum memory. *arXiv* **2025**, arXiv:2506.01779. [CrossRef]
53. Google Quantum AI. Exponential suppression of bit or phase errors with cyclic error correction. *Nature* **2021**, *595*, 383–387. [CrossRef] [PubMed]

54. Forlivesi, D.; Valentini, L.; Chiani, M. Bubble Clustering Decoder for Quantum Topological Codes. *IEEE Trans. Commun.* **2025**, *73*, 8470–8483. [CrossRef]
55. Lange, M.; Havström, P.; Srivastava, B.; Bengtsson, I.; Bergentall, V. Data-driven decoding of quantum error correcting codes using graph neural networks. *Phys. Rev. Res.* **2024**, *7*, 023181.
56. Liao, P.; Sanders, B.C.; Feder, D.L. Topological graph states and quantum error-correction codes. *Phys. Rev. A* **2022**, *105*, 042418. [CrossRef]
57. Kschischang, F.R.; Frey, B.J.; Loeliger, H.A. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* **2002**, *47*, 498–519. [CrossRef]
58. Google Quantum AI and Collaborators. Measurement-induced entanglement and teleportation on a noisy quantum processor. *Nature* **2023**, *622*, 481–486. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.