Resource Allocation for Surface Code Teleportation in Distillation-Based Quantum Networks

Tianjie Hu[1], Jindi Wu[1,2], and Qun Li[1]
[1]William & Mary, Williamsburg, VA 23185, USA
[2]DePaul University, Chicago, IL 60614, USA

2026 IEEE 46th International Conference on Distributed Computing Systems (ICDCS) | 979-8-3195-2979-4/26/$31.00 ©2026 IEEE | DOI: 10.1109/2575-8411.2026.00098

***Abstract*—We study the problem of teleporting a surface-code logical qubit using a limited number of noisy entangled pairs, each of which must be distilled prior to use. Given n entangled pairs and a $d \times d$ surface-code patch, the task is to partition the n pairs into d^2 subsets for distillation and subsequently assign the resulting high-fidelity pairs to data qubits so as to minimize the overall logical error rate. We formalize this task as the entanglement allocation problem and decompose it into two sequential subproblems: the *distillation resource partition* problem and the *qubit mapping* problem. Both subproblems are guided by our key observation that data qubits within a surface-code patch exhibit heterogeneous and correlated importance with respect to the logical error rate. Leveraging this structure, we develop correlated-importance-aware algorithms that jointly optimize resource partitioning and qubit assignment according to these importance relations. Through numerical simulations, we demonstrate that the proposed strategies significantly reduce logical error rates compared to baseline allocation methods under the same entanglement budget.**

***Index Terms*— Quantum Network, Quantum Teleportation, Quantum Error Correction, Surface Code**

I. Introduction

Quantum teleportation is a fundamental technique in quantum information science, enabling the transfer of quantum states via pre-shared entanglement and classical communication. It plays a central role in fault-tolerant quantum communication, quantum networks, and distributed quantum computing [1]–[8]. In practice, quantum teleportation is constrained by both the quantity and fidelity of shared entangled pairs between nodes. Due to environmental decoherence and imperfect state preparation, these entangled pairs are typically noisy, which directly degrades teleportation fidelity.

To address this issue, entanglement distillation is employed to convert multiple low-fidelity entangled pairs into fewer high-fidelity ones using only local quantum operations and classical communication. While effective, distillation is resource-intensive, as producing a single high-fidelity pair may require many input pairs. Moreover, entanglement generation in quantum networks is inherently probabilistic, leading to a limited and fluctuating supply of entangled pairs. At the same time, multiple teleportation requests may arrive concurrently, each requiring a significant number of entangled pairs. This motivates the central challenge of efficiently allocating limited entanglement resources across multiple teleportation tasks.

This challenge becomes particularly critical when teleporting encoded logical qubits, such as those implemented using surface codes, which are a leading architecture for fault-tolerant quantum computing. A surface-code logical qubit is encoded on a $d \times d$ patch of data qubits, where all d^2 physical qubits must be teleported with sufficiently high fidelity to preserve the encoded logical state. Through experimental observations, we find that different allocation strategies for teleporting these d^2 data qubits can result in significantly different logical error rates of the surface-code patch. For example, as illustrated in Fig. 1, consider teleporting a patch of nine data qubits using four high-fidelity (white) and five low-fidelity (gray) entangled pairs. The two illustrated allocation strategies lead to markedly different logical error rates, despite using identical resource counts. This difference arises from the fact that data qubits within a surface-code patch contribute unequally to the logical error rate, a structural property identified in this work that has not been previously emphasized. Based on this finding, an effective teleportation protocol must allocate entangled pairs heterogeneously across data qubits, assigning higher-fidelity resources where they yield greater logical protection, thereby minimizing logical error under a fixed entanglement budget.

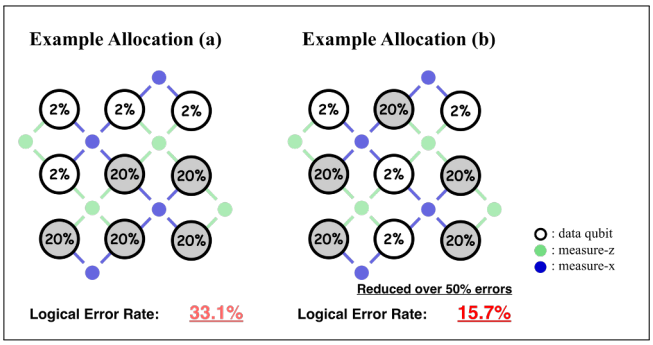

Fig. 1. Two example allocations for teleporting a surface-code patch of 9 data qubits. Each data qubit is denoted with its physical error rate, and listed below each patch is its corresponding logical error rate.

In this paper, we address the problem of allocating a fixed budget of n noisy entangled pairs for teleporting a $d \times d$ surface-code patch. This problem is both theoretically interesting and practically relevant. From a theoretical standpoint, it lies at the intersection of quantum information theory, combinatorial optimization, and quantum error correction. It combines discrete resource allocation (partitioning of entangled pairs) with a complex, non-linear objective function (logical

979-8-3195-2979-4/26/$31.00 ©2026 IEEE
DOI 10.1109/2575-8411.2026.00098

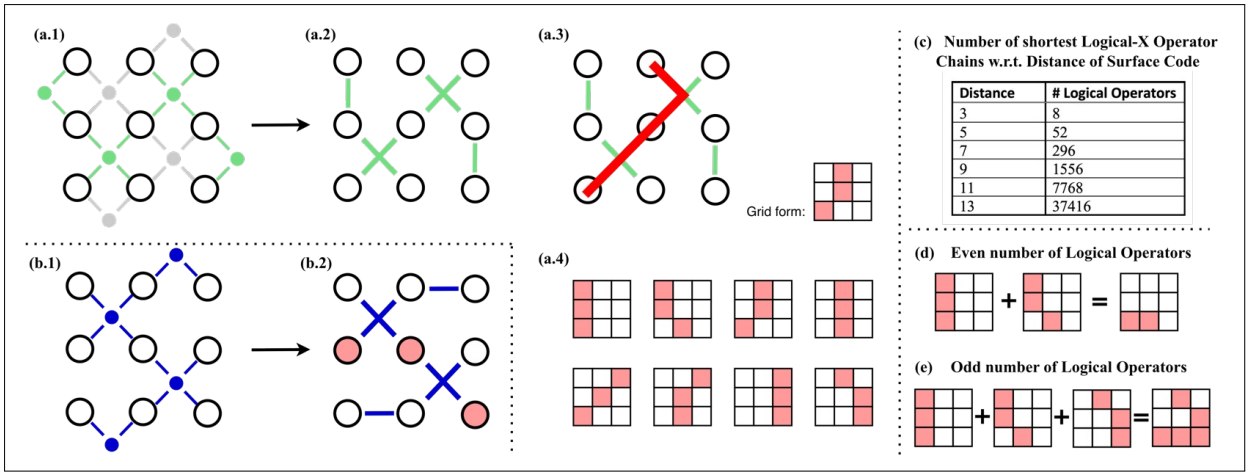

Fig. 2. **(a.1–2)** A distance-3 surface-code patch consisting of data qubits and Z-type measurement qubits. **(a.3)** An example logical-X operator in a distance-3 surface-code patch, shown on the data-qubit lattice. **(a.4)** All shortest (weight-3) logical-X operators in a distance-3 surface code. **(b.1–2)** A distance-3 surface-code patch consisting of data qubits and X-type measurement qubits; red data qubits illustrate an example logical-Z operator. **(c)** The number of shortest logical-X operator chains as a function of surface-code distance. **(d)** Example of two logical operators acting on the code. **(e)** Example of three logical operators acting on the code.

error rate) that is difficult to model and typically evaluated through simulation. From a practical perspective, as quantum networks and modular architectures evolve, there will be an increasing need to teleport fault-tolerant logical qubits across spatially separated subsystems. Optimizing the use of limited entanglement resources is crucial for making such operations reliable and scalable.

We address this allocation problem in two stages, corresponding to two coupled subproblems: the *distillation resource partition* problem and the *qubit mapping* problem. A distillation resource partition divides the n entangled pairs into d^2 disjoint subsets, each used to produce one distilled entangled pair. Given a partition, a qubit mapping then assigns the resulting d^2 distilled pairs to the d^2 data qubits of the surface-code patch, and the resulting configuration is evaluated via its logical error rate. Our objective is to identify the joint partition–mapping configuration that minimizes the logical error rate of the teleported logical qubit. To this end, we first construct a set of representative distillation partition strategies. For each candidate partition, we solve the corresponding qubit mapping problem by incorporating the heterogeneous importance of data qubits within the code patch. Finally, we select the partition whose optimal mapping achieves the lowest logical error rate.

We are the first to observe that individual data qubits within a surface-code patch have differing importance with respect to the logical error rate, and that these importance values are correlated. This finding has profound implications for the future design of surface codes in quantum computing architectures and for algorithm development. In the context of this paper, it significantly advances our understanding of how to transmit surface-code patches over a quantum network.

Building on this insight, we present the first systematic analysis of how entanglement resources should be allocated across data qubits in a surface-code patch. We further provide the first integration of distillation-based teleportation with surface-code transmission to enable optimized quantum communication.

Our evaluation demonstrates that intelligent resource allocation for teleporting logical qubits can significantly reduce transmission-induced logical errors. As quantum technologies progress toward practical deployment, such approaches may become essential for realizing robust and efficient long-range quantum communication systems.

II. Background

A. Surface codes

Surface codes are designed to detect and correct errors arising during quantum computation. Each surface-code patch consists of two types of physical qubits: data qubits and measurement qubits. The data qubits store the encoded quantum information, while the measurement qubits continuously probe the stabilizers to detect potential errors. As illustrated in Fig. 1, each data qubit is coupled to its neighboring measurement qubits, comprising two X-type and two Z-type stabilizer qubits, except at the boundaries. A local error on a data qubit triggers correlated syndromes in its adjacent measurement qubits, enabling error identification. Each patch thus encodes a logical qubit and supports the implementation of logical operations.

Common logical single-qubit operations include logical-X and logical-Z operators. A logical-X operator can be realized in multiple equivalent ways, typically as a chain of d data qubits (where d is the code distance) extending from top to bottom, with each qubit acted on by a physical Pauli-X gate. As shown in Fig. 2(a.1–a.3), the data-qubit lattice forms the graph vertices, while Z-type stabilizer measurements define the edges, providing a natural representation of logical-X paths. Fig. 2(a.4) enumerates all minimum-weight logical-X

operators for a distance-3 code, and Fig. 2(c) shows their count as a function of code distance.

The composition of multiple logical-X operators follows Pauli parity rules: an even number of identical operators cancels, while an odd number yields an effective logical operation corresponding to a longer chain, as illustrated in Fig. 2(d–e). Logical-Z operators are defined analogously, with chains running horizontally across the lattice and implemented using Pauli-Z gates, as shown in Fig. 2(b.1–b.2). Logical-Y operators are given by $Y = iXZ$, and can be interpreted as the combined action of logical-X and logical-Z operators.

B. Logical errors in surface codes

While surface codes can detect individual physical errors using measurement qubits, undetected logical errors can occur when multiple errors form a chain matching a logical operator. Thus, logical errors are also described as unintended logical operators. Shortest logical errors (length d) are the most probable, as longer chains require more accumulated errors. Among logical errors of the same length, rates may also vary due to differing physical error rates across data qubits. The precise formula for calculating the error rate of each logical error chain is complex and depends on the chosen error correction decoder, but it can generally be approximated by multiplying the individual error rates of each data qubit along the chain:

$$E_L = \prod_{q_i \in L} e_{q_i} \tag{1}$$

where L is the logical error chain, and e_{q_i} is the error rate of each data qubit q_i traversed by L. The logical error rate of a surface-code patch can then be approximated by summing the error rates of all logical error chains:

$$E_{code} = \sum_{n=d}^{d^2} \sum_{L_s \in \{L_n\}} \prod_{q_i \in L_s} e_{q_i} \tag{2}$$

where $\{L_n\}$ denotes the set of all logical error chains of an odd length n, and d is the distance of the surface-code patch.

C. Entanglement distillation

Transmitting physical qubits within networks typically requires consuming entangled pairs for teleportation. The quality of an entangled pair is characterized by *fidelity*, which measures the closeness of a shared quantum state to an ideal maximally entangled state. Entanglement distillation is a technique for improving the fidelity of entangled pairs shared over noisy quantum channels. Among various distillation methods, the recurrence protocol, introduced by Bennett et al. [9], is one of the simplest and most practical. The recurrence protocol operates on two noisy entangled pairs. In each round, two noisy entangled pairs are processed using local operations and classical communication. Specifically, bilateral CNOT gates are performed on the two pairs, then one of the pairs are measured in the computational basis. Depending on the measurement outcomes, the remaining pair may be retained or discarded. For example, taking two entangled pairs of fidelity F, the result fidelity F' after one round of distillation will be:

$$F' = \frac{F^2 + \frac{1}{9}(1-F)^2}{F^2 + \frac{2}{3}F(1-F) + \frac{5}{9}(1-F)^2} \tag{3}$$

if the measurement outcomes indicate retaining the remaining pair. The probability of retaining the remaining pair, which is the success rate for each round of distillation, is exactly the value of the denominator in Eq. 3. Depending on the target fidelity, multiple rounds of distillation can be performed iteratively until the result fidelity meets the requirement.

III. Related Work

Entanglement distillation, as a fundamental technique in quantum communication and distributed quantum computing, enables the purification of noisy entangled pairs into higher-fidelity states [10]–[12]. Various distillation protocols have been proposed, including hashing methods [9], recurrence protocols [10], and measurement-based approaches [13], with recent work focusing on their experimental implementation under resource constraints [14]–[16].

Parallel to advances in distillation, quantum error-correcting codes, particularly surface codes [17]–[22], have become a leading architecture due to their high threshold and simple layouts. Prior research has studied how physical error distributions affect logical error rates [23]–[27]. Recent work has also considered optimizing error correction under hardware noise asymmetries, including biased-noise decoders [27]–[30] and nonuniform error rate modeling [31]–[37].

However, few studies have examined how to jointly optimize distillation resource distribution and qubit mapping for a specific code layout, under a limited entanglement budget. Prior work in this direction includes network routing on logical qubits [38]–[40], and local logical qubit mapping [41]–[43]. However, existing work does not consider the joint interplay between distillation depth, error-rate thresholding, and the spatial importance of qubits within a surface-code patch. Our approach differs by formulating entanglement-distillation allocation as a constrained optimization problem that incorporates both physical fidelity and logical code structure. By combining partitioning strategies with qubit assignment and error-rate-aware decoders, we provide a practical framework that reduces transmission-induced logical error rates.

IV. Transmitting surface codes in quantum networks

In quantum networks, surface codes are widely used to encode quantum information for fault-tolerant communication. Transmitting a surface-code patch of size d^2 through quantum teleportation typically requires d^2 entangled pairs, with one pair assigned to each physical qubit. However, in long-distance quantum communication, channel noise and environmental decoherence often degrade entangled-pair fidelity below the threshold necessary for reliable surface-code performance. Teleportation using low-fidelity entanglement increases physical error rates, which can substantially elevate the logical error

probability of the transmitted patch. To mitigate this issue, entanglement distillation protocols are commonly employed to improve the fidelity of entangled pairs prior to teleportation, at the cost of additional entanglement consumption. In practical quantum networking scenarios, where entanglement resources are constrained, routing and resource-allocation protocols must carefully determine the number of entangled pairs allocated to each data qubit transmission within a surface-code patch in order to balance communication reliability against overall resource efficiency.

This gives rise to a resource-allocation problem under a limited supply of entangled pairs for transmitting a surface-code patch. Consider a simplified scenario in which all available entangled pairs are used to distill the teleportation channel for only one of the d^2 data qubits in Fig. 2(a.2). As discussed in Section II, each data qubit participates in multiple logical error chains, and its impact on the overall logical error rate depends on both its physical error rate and the number of logical error chains that traverse it. Thus, allocating distillation resources to the central qubit is generally more effective, since it is traversed by four logical error chains, the largest among all qubits in the patch. In contrast, prioritizing the upper-right qubit, which is traversed by only two logical error chains, suppresses fewer logical error pathways. These different allocation choices therefore produce different overall logical error rates.

Many other allocation strategies are possible, including uniform allocation, where each data qubit receives the same distillation resources. Formally, consider transmitting a surface-code patch of size d^2 through a single path with fixed attenuation using n entangled pairs of fidelity F. The problem is to determine an allocation $\{N_i \,|\, i \in [1, d^2],\ \ N_i \in \mathbb{Z}^+,\ \sum_{i=1}^{d^2} N_i = n\}$, where each N_i denotes the number of entangled pairs consumed to distill the teleportation channel for the i-th data qubit. The objective is to identify the allocation that minimizes the logical error rate of the transmitted patch. An exhaustive search over all possible allocations requires enumerating every partition of n entangled pairs into d^2 positive integer subsets, corresponding to the integer composition problem with a fixed number of parts. The total number of possible allocations is $\binom{n-1}{d^2-1}$. Since practical settings often involve large entanglement budgets (e.g., $n > 1000$), exhaustive search quickly becomes computationally intractable. To address this challenge, we decompose the allocation problem into two stages: the *distillation resource partition* stage and the *qubit mapping* stage.

In the distillation resource partition stage, the n available entangled pairs are divided into d^2 subsets, where each subset is used to distill the teleportation channel for one data qubit. The size of each subset determines the number of distillation rounds applied and, consequently, the resulting fidelity of the distilled entangled pair. At this stage, the focus is solely on how to partition resources among the d^2 transmissions, without yet assigning the resulting fidelities to specific qubits in the surface-code patch.

The qubit mapping stage then determines how these d^2 fidelities should be assigned to the d^2 data qubits. We characterize each qubit's influence on the patch logical error rate by its *importance*, as discussed later in Section V. Intuitively, higher-fidelity entangled pairs should be allocated to more important qubits, while lower-fidelity pairs are assigned to less critical ones. However, this mapping is nontrivial because qubit importance is interdependent: changing the physical error rate of one qubit can alter the relative importance of others.

In the following sections, we first address the qubit mapping stage (Section V), since evaluating any resource partition strategy requires an effective mapping of distilled fidelities to data qubits. We then build upon this result to optimize the distillation resource partition stage (Sections VI–VII).

V. Qubit mapping problem for surface codes in quantum networks

In this section, we present our approach for identifying an optimal qubit mapping. For clarity, we use *error rate*, defined as one minus fidelity, to characterize both entanglement quality and physical qubit quality throughout the paper. The qubit mapping problem is formally stated as follows: given a distance-d surface-code patch and a set of d^2 error rates $\{e_x : x \in [1, d^2]\}$, assign each error rate to a distinct data qubit such that the resulting logical error rate of the patch is minimized. The logical error rate of each candidate mapping is estimated via Monte Carlo simulation: for each mapping, physical error configurations are sampled according to the associated error rates, and the resulting syndromes are decoded using [27]. The logical error rate is computed as the fraction of trials in which the decoded logical outcome differs from the known initial state.

A brute-force search for the optimal qubit mapping requires enumerating all permutations of assigning d^2 error rates to d^2 data qubits, yielding a search space of size $(d^2)!$. This becomes intractable even for moderate code distances (e.g., $d \geq 15$), where the number of possible mappings grows exponentially. Thus, efficient approximation strategies are necessary to identify high-quality mappings without exhaustive search. In the following, we present our proposed method together with trimmed variants that reduce computational overhead while maintaining performance.

A. Correlated-importance–aware mapping

Our approach is based on the *importance* of each data qubit within a surface-code patch. As discussed in Section IV, each data qubit is traversed by multiple logical error chains. If all chains had identical error rates, a qubit traversed by more chains would be inherently more important, since perturbations to its error rate would affect a larger number of logical error events. In practice, however, logical error chains exhibit heterogeneous error rates due to nonuniform physical error rates across data qubits. Accordingly, instead of counting the number of traversing chains, we define the importance of a data qubit as the sum of the error rates of all logical error

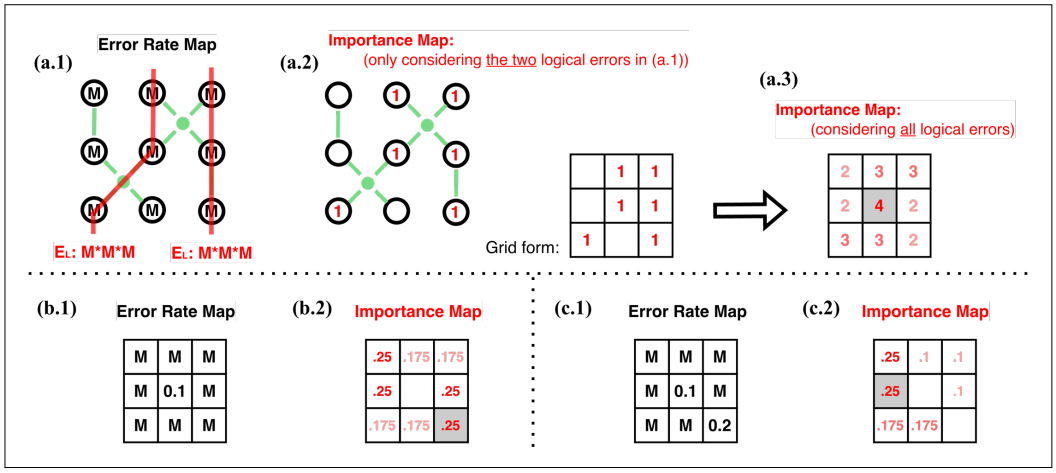

Fig. 3. **(a.1)** Two example logical error chains in a distance-3 surface-code patch. The logical error rate (E_L) of each chain is computed as $M*M*M = M^3$, where M is the assigned default error rate of each data qubit. **(a.2)** An importance map considering only the two logical error chains, where a unit of M^3 is omitted. **(a.3)** An importance map considering all shortest logical error chains. **(b.1-2) (c.1-2)** Importance maps for the corresponding error rate maps, assuming $M = 0.5$ for computation.

chains that pass through it. This definition naturally captures the unequal contribution of different chains.

In the meantime, since the error rate of each logical chain is jointly determined by all qubits along the chain, modifying the error rate of one qubit influences the importance of others. For example, consider three qubits q_1, q_2, q_3, and assume there exists a logical error chain traversing q_1 and q_2, but no chain connecting q_1 and q_3. In this case, decreasing the error rate of q_1 reduces the importance of q_2, while increasing it has the opposite effect. In contrast, the importance of q_3 remains unchanged, as it is independent of q_1. This *correlated importance* among data qubits motivates our correlated-importance-aware (CIA) mapping strategy. In our approach, we iteratively update two maps: an *error-rate map* $G = \{g_{i,j} : i, j \in [1, d]\}$ and an *importance map* $H = \{h_{i,j} : i, j \in [1, d]\}$, which respectively record the assigned error rates and the corresponding importance values of data qubits located at row i and column j in a distance-d surface-code patch.

The procedure of our approach on a distance-3 surface-code patch is illustrated in Fig. 3. Here, we focus on logical-X errors, since a surface-code patch can be regarded as a means to protect against X and Z errors as two separate (yet overlapped) tasks; later in this section, we show how our approach extends naturally to a comprehensive protection against both. In addition, as discussed in Section II, distance-d surface codes have both shortest logical error chains of length d and longer chains of length greater than d. As shown in Fig. 2, all error chains have odd length, implying that any long chain has length at least $d + 2$. Moreover, for practical network channels the physical error rate is typically below 1%, making these longer chains at least $0.01^2 = 10^{-4}$ less likely than the shortest ones. Thus, when estimating the logical error rate of a surface-code patch, it suffices to consider only the shortest logical error chains. For the rest of this paper, we therefore focus primarily on these shortest chains.

As shown in Fig. 3(a.1), we first construct a graph in which data qubits correspond to nodes and Z-type measurement qubits correspond to edges. Based on this graph, we identify all shortest logical-X error chains via depth-first search (DFS), where each valid chain connects a data qubit on the top boundary to one on the bottom boundary. We denote the resulting set of shortest logical-X chains as $\{L_s\}$. In addition, we construct the two maps: an *error-rate map* G and an *importance map* H, both represented as $d \times d$ grids indexed by the row and column of the surface-code patch. For the error-rate map G, each entry $g_{i,j}$ is initialized to a constant value M, chosen as the maximum among the available error rates to be assigned, i.e., $\{e_x : x \in [1, d^2]\}$.

For the importance map H, Fig. 3(a.2) shows an example computed using only the two highlighted logical error chains in Fig. 3(a.1), both with error rate M^3. Since the two chains are disjoint, we assign unit contributions to the data qubits they traverse, omitting the common factor M^3. Fig. 3(a.3) shows the full importance map computed over the complete set $\{L_s\}$, where all shortest logical-X error chains (as illustrated in Fig. 2) are included. For each data qubit, we record the number of logical chains that traverse it. For example, the central qubit is traversed by four chains, and thus is assigned a value of 4 at coordinate $(2, 2)$. Based on this importance map, the central qubit is identified as having the highest importance and is therefore assigned the lowest available error rate. Subsequent iterations of the procedure are illustrated in Fig. 3(b.1–b.2) and (c.1–c.2), with $M = 0.5$ assumed in the computation of intermediate importance maps.

The detailed algorithm is presented in Algorithm 1, which consists of three stages. In the first stage (lines 1–3), all logical-X error chains in the surface-code patch are identified via depth-first search, and each data qubit in the error-rate map G is initialized to a default value M. In the second stage (lines 6–14), the importance map H is computed based on the set of

Algorithm 1 Correlated-Importance–Aware (CIA) Mapping

Input: code distance d, set of physical error rates $\{e_x\}$,
Output: error rate map G

1: Construct a $d \times d$ error rate map G as in Fig. 3
2: Initialize each node $g_{(i,j)}$ in G as $M = \max\{e_x\}$
3: Find all logical-X error chains $\{L_s\}$ of length d
4: Sort $\{e_x\}$ in ascending order
5: **for** $e_x \in \{e_x\}$ **do**
6: Construct a $d \times d$ importance map H
7: Initialize each cell $h_{(i,j)}$ in H as 0
8: **for** $L_s \in \{L_s\}$ **do**
9: Find nodes $\{g_{(a,b)}, g_{(c,d)}, \dots\}$ traversed by L_s
10: Compute $E_L = g_{(a,b)} * g_{(c,d)} * \dots$
11: **for** $g_{(i,j)} \in \{g_{(a,b)}, g_{(c,d)}, \dots\}$ **do**
12: $h_{(i,j)} += E_L$
13: Find cell $h_{(i,j)}$ with largest number in H
14: **if** $g_{(i,j)} == M$ **then**
15: Update $g_{(i,j)} = e_x$
16: **else**
17: Set $h_{(i,j)} = -1$, then return to step 15
18: **return** G

logical error chains $\{L_s\}$. In the third stage (lines 15–20), the most important unassigned data qubit is selected and mapped to the lowest available error rate. Then the algorithm iterates through the second and third stages until all data qubits are assigned. Notably, though we only consider logical-X errors in the above illustration, Algorithm 1 can be easily adapted to a comprehensive protection against both logical errors by also adding logical-Z errors into the set of logical errors $\{L_s\}$.

The time complexity of Algorithm 1 is $\mathcal{O}(d^4 2^d)$ due to: graph initialization $\mathcal{O}(d^2)$, DFS to find logical chains $\mathcal{O}(d \cdot 2^d)$, importance map construction $\mathcal{O}(d^2 2^d)$, and $\mathcal{O}(d^2)$ iterations. In the following, we discuss how trimming methods can be applied to Algorithm 1 to reduce its time complexity.

B. Trimmed version

Trimming can be applied in two directions to reduce the computational complexity of Algorithm 1. First, we reduce the number of logical error chains in $\{L_s\}$ by modifying the depth-first search procedure. Since the number of logical-X operators grows exponentially with code distance ($\sim 2^d$), we restrict $\{L_s\}$ to a representative subset. For instance, one may retain only the diagonal operator together with the d vertical logical operators. This reduces the overall time complexity to $\mathcal{O}(d^4)$, primarily due to the simplified construction of logical chains and the corresponding importance maps. However, as shown later in Section VIII, this reduced $(d+1)$-operator subset leads to degraded performance in logical error suppression compared to the full (untrimmed) formulation, although it still significantly outperforms baseline random mappings. Therefore, restricting the set of logical operators introduces a trade-off between computational efficiency and mapping quality, and identifying alternative representative subsets remains an open direction for future work.

Algorithm 2 CIA Mapping with Trimming

Input: code distance d, set of physical error rates $\{e_x\}$,
Output: error rate mapping G

1: Construct a $d \times d$ grid graph G as in Fig. 2(a.2)
2: Initialize each node $g_{(i,j)}$ in G as $M = \max\{e_x\}$
3: Find all logical-X operators $\{L_s\}$ of length d in G
4: Sort $\{e_x\}$ in ascending order
5: **for** $e_x \in \{e_1, e_2, ..., e_n\}$ **do**
6: *... (same as steps 6-20 in Algorithm 1)*
7: **for** $e_x \in \{e_{n+1}, e_{n+2}, ..., e_{d^2}\}$ **do**
8: randomly select a $g_{(i,j)}$ that is equal to M
9: $g_{(i,j)} = e_x$
10: **return** G

Second, we reduce the number of iterations in Algorithm 1 by applying it only to identify the n most important physical qubits, while assigning the remaining qubits randomly, as formalized in Algorithm 2. Depending on n, this strategy effectively reduces the time complexity to $\mathcal{O}(d^2 2^d)$, due to the reduced number of iterations. In practice, n can be chosen as a constant in networking scenarios with only two available routing options, where only a small subset m of qubits can be transmitted through a lower-error channel. In such cases, setting $n = m$ is sufficient, as the remaining qubits experience uniform error rates. Alternatively, n may scale with the code distance in more general multi-route settings with heterogeneous error characteristics. As shown later in Section VIII, this trimming strategy enables a flexible trade-off between runtime and logical error suppression.

VI. Distillation resource partition problem for surface codes in quantum networks

Building upon the qubit mapping algorithms developed in the previous section, we can now evaluate different distillation resource partition strategies. In this section, we propose several empirical partition strategies and assess their effectiveness. Section VII then discusses how these insights can be used to construct optimized partition schemes.

In each partition strategy, the available entangled pairs are partitioned into $d^2 = 25$ subsets corresponding to the data qubits of a distance-5 surface-code patch. Each subset undergoes entanglement distillation, and the resulting error rates are assigned to data qubits using the CIA mapping algorithm described in Section V. The logical error rate of each transmitted patch is then estimated by simulating depolarizing noise using Stim [44] and decoding with Sparse Blossom [27] over 10,000 trials. For ease of discussion, all experiments in this section focus on distance-5 surface codes. We consider three resource scenarios — insufficient, sufficient, and abundant — distinguished by the total number of available entangled pairs n. Within each scenario, we vary the physical error rate of the initial entangled pairs and measure the resulting logical

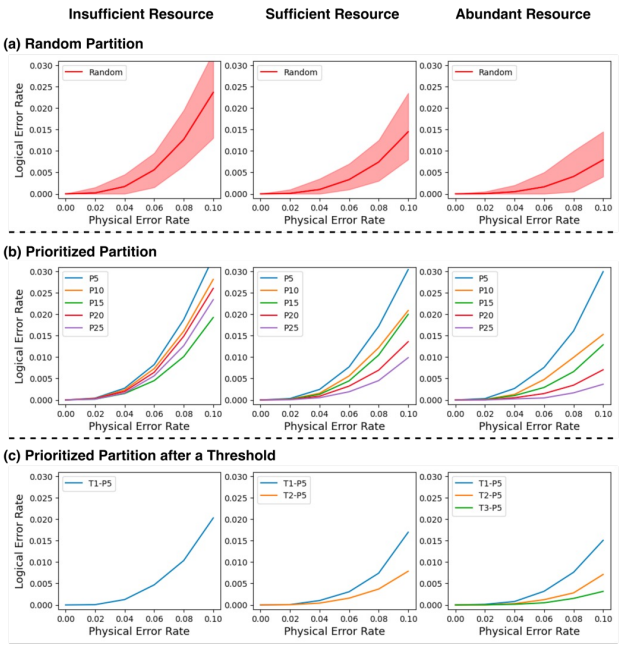

Fig. 4. Performance of **(a)** random partition, **(b)** prioritized partition, and **(c)** prioritized partition after a threshold strategies under insufficient, sufficient, or abundant resource.

error rate of the transmitted surface-code patch to evaluate the performance of each partition strategy.

A. Random partition

We use random partition as a baseline to evaluate the performance of alternative strategies. As shown in Fig. 4(a), we test the random partition strategy multiple times at each scenario: the average logical error rate is reported as a curve, while the shaded region indicates the full range of observed outcomes. Notably, the random strategy exhibits substantial variance, rendering it unsuitable for reliable deployment.

B. Prioritized partition

As discussed in Section V, data qubits contribute unequally to logical errors. Motivated by this observation, we propose a prioritized partition strategy that allocates more distillation resources to higher-priority qubits. For a distance-5 surface code, we consider five strategies $(P5, P10, \ldots, P25)$, where Px allocates entangled pairs preferentially to the x most important qubits, while the remaining $25 - x$ qubits are teleported without distillation. More generally, for a distance-d surface code, this defines a family of d prioritized strategies $Pd, P2d, \ldots, Pd^2$.

The performance of these strategies is shown in Fig. 4(b). Different strategies perform best under different resource scenarios: for example, $P15$ achieves the lowest logical error rate when resources are limited. However, no single prioritized strategy consistently dominates across all scenarios, indicating that further refinement of priority-based resource allocation is necessary.

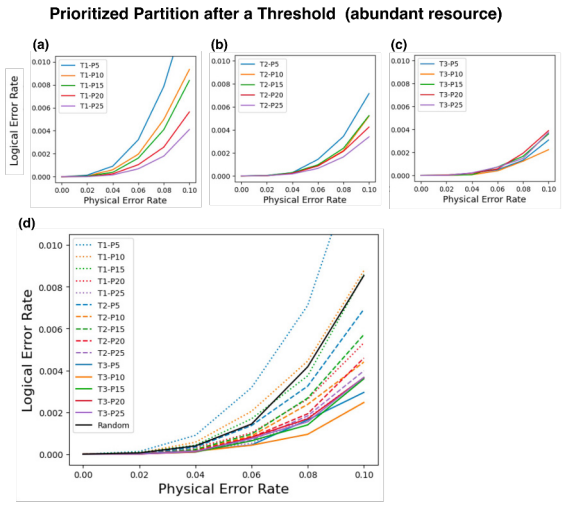

Fig. 5. Performance of different prioritized partition strategies under abundant resource settings. **(a) (b) (c)** Performance when prioritizing a different number of qubits after a threshold of 1/2/3 distillation rounds is applied, respectively. **(d)** Aggregate comparison of the results shown in (a–c).

C. Prioritized partition after a threshold

After analyzing the decoding outcomes under prioritized partitioning, we find that many logical errors arise from excessively high error rates on non-prioritized data qubits. To mitigate this effect, we introduce the *prioritized partition after a threshold* strategy. This approach first reduces the error rate of all data qubits to a predefined threshold, and then allocates any remaining entangled pairs to prioritized qubits. The threshold is selected from discrete levels corresponding to the number of distillation rounds and is constrained by resource availability. For example, in Fig. 4, at most one round of distillation is feasible in the insufficient-resource scenario, whereas multiple rounds can be applied under abundant resources.

In Fig. 4(c), $T1$-$P5$ denotes one round of distillation applied to all qubits followed by prioritization of five qubits, while $T3$-$P5$ denotes three rounds followed by the same prioritization. Compared with Fig. 4(b), thresholding significantly improves performance across all resource regimes.

A more detailed analysis in the abundant-resource scenario is shown in Fig. 5, where (a)–(c) correspond to one, two, and three distillation rounds, respectively. Notably, $T1$-$P25$, $T2$-$P25$, and $T3$-$P25$ represent the same uniform allocation strategy. We observe that $T1$-$P25$ and $T2$-$P25$ perform best in (a) and (b), indicating the benefit of uniformly reducing error rates, while $T3$-$P10$ achieves the best overall performance, demonstrating the advantage of combining thresholding with prioritization. Fig. 5(d) summarizes all strategies under abundant resources, showing that threshold-based prioritized strategies consistently outperform random partitioning. In particular, a three-round threshold $(T3)$ yields the best overall performance. The selection of threshold and prioritization levels in different scenarios is further discussed in Section VII.

D. Prioritized partition after multiple thresholds

To further refine priority-based allocation, we also consider applying multiple thresholds prior to prioritization to better capture the heterogeneous importance of data qubits (Section V). For example, we denote a strategy $T1$-$T2$-$T3$-$P5$ as applying one, two, and three rounds of distillation to three disjoint groups of five, five, and fifteen qubits, respectively, followed by prioritization over five qubits in the final group. More generally, a Ta-Tb-Tc-Pk strategy on a distance-d code assigns a, b, and c rounds of distillation to three groups of d, d, and $d(d-2)$ qubits, respectively, with remaining resources distributed among k qubits in the last group.

However, as shown in Fig. 6(a), multi-threshold strategies do not consistently improve performance and may even degrade it. Similar trends are observed in Fig. 6(b) under varying resource levels. This behavior is partly due to reduced flexibility in prioritization after multiple thresholds. For example, after applying the $T1 - T2 - T3 - T4$ thresholds, only ten qubits remain eligible for prioritization, offering significantly less flexibility than selecting from all 25 qubits after applying a single $T4$ threshold. However, since our evaluation does not cover all possible scenarios, multi-threshold strategies remain a promising direction for future exploration.

VII. Finding the Optimal Distillation Resource Partition for Surface Codes

In the previous section, we showed that although the proposed strategies outperform random allocation, no single strategy is optimal across all scenarios. In practice, we therefore enumerate a set of *representative* distillation resource partition strategies and select the one that minimizes the logical error rate. This reduces the distillation resource partition problem to efficiently identifying such representative strategies, each parameterized by (i) the number of distillation rounds applied before prioritization and (ii) the number of prioritized qubits.

A. Determining the number of distillation rounds before prioritization

We define a *threshold* T as the number of distillation rounds applied prior to prioritization. Under the recurrence protocol [9], each round consumes two entangled pairs and succeeds probabilistically. Given n entangled pairs with fidelity F, the maximum feasible number of rounds per data qubit can be computed using the formulas in Section II. For example, with 2500 entangled pairs of fidelity 0.9 for a distance-5 surface code, each of the 25 data qubits can use 100 pairs, supporting up to four distillation rounds, so we can choose $T \in \{0, 1, 2, 3, 4\}$. If the fidelity decreases to 0.8, the lower success probability reduces the feasible rounds to three, yielding $T \in \{0, 1, 2, 3\}$.

B. Choosing the number of prioritized data qubits

Let P denote the number of prioritized data qubits. For a distance-d surface code, empirical results (Fig. 4) suggest selecting P from $\{kd \mid k \in \mathbb{Z},\ 1 \le k \le d\}$. For instance, for $d = 5$, we use $P \in \{5, 10, 15, 20, 25\}$. Although P can in principle take any value in $[1, d^2]$, enumerating all possibilities is computationally expensive. Restricting P to a representative subset provides a practical trade-off between efficiency and solution quality.

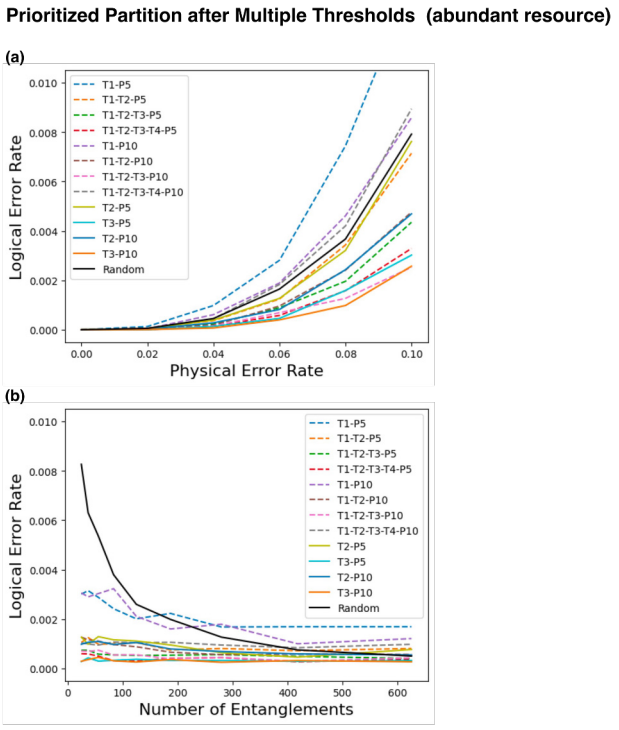

Fig. 6. Performance of different prioritized partition strategies under abundant resource settings with multiple threshold levels. **(a)** Aggregate comparison on prioritizing 5 or 10 qubits under 1/2/3/4 applied thresholds. **(b)** Logical error rates of different partition strategies as a function of the number of available entangled pairs.

C. Algorithm Summary

We summarize the procedure for selecting a distillation resource partition strategy as follows:

1. Determine the feasible set of distillation thresholds: $\mathbf{T} = \{0, 1, 2, \dots\}$, based on available resources;
2. Select a representative set of prioritized strategies: $\mathbf{P} = \{d, 2d, \dots, d^2\}$;
3. For each $(T, P) \in \mathbf{T} \times \mathbf{P}$:
   - 3.1 Apply the CIA mapping algorithm;
   - 3.2 Estimate the logical error rate via simulation;
4. Select the pair (T, P) that minimizes the logical error rate.

Examples are shown in Fig. 5(d), where we identify $\mathbf{P} = \{5, 10, 15, 20, 25\}$ and $\mathbf{T} = \{0, 1, 2, 3\}$.

Notably, this approach evaluates only a representative subset of partition strategies and does not guarantee global optimality. Designing efficient algorithms that directly identify the optimal partition without enumeration remains an open problem and is left for future work.

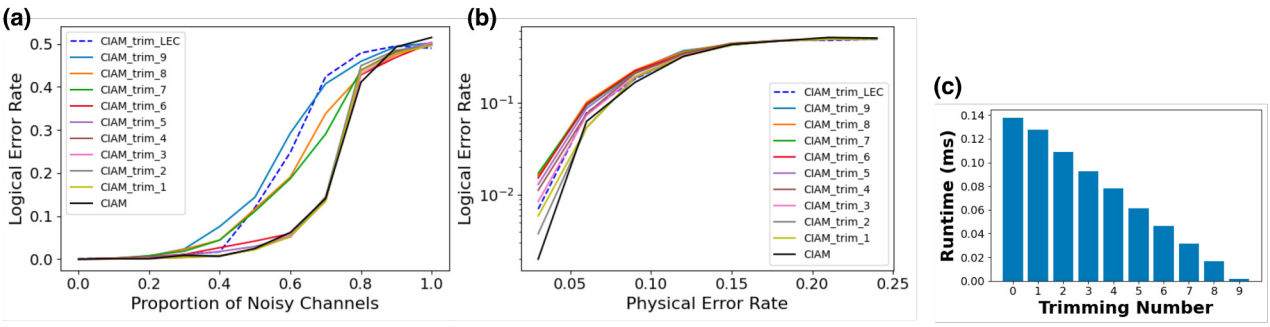

Fig. 7. Logical error rates of a distance-9 surface-code patch with respect to **(a)** the proportion of noisy channels and **(b)** the physical error rates of high-quality channels, under different mapping strategies. **(c)** Computation time for a distance-9 patch as a function of the trimming number used in the CIA mapping.

VIII. Evaluation

In the previous sections, we conducted an extensive evaluation of distillation resource partition strategies. Here, we focus on the performance of the qubit mapping algorithms.

A. Evaluation Methodology

We use a simplified quantum network model based on our priority-based distillation resource partition strategies. The network has two types of channels: *high-quality* and *noisy*. High-quality channels correspond to prioritized qubits that undergo more distillation and therefore have lower error rates. In contrast, noisy channels receive fewer resources, resulting in higher error rates for the transmitted qubits.

Each surface-code patch is transmitted by routing some data qubits through high-quality channels and others through noisy channels. The mapping algorithm determines which data qubits are assigned to each channel type. After fixing a mapping, we simulate depolarizing noise using the Stim library [44] and perform syndrome decoding with the Sparse Blossom decoder [27], using 10,000 Monte Carlo trials per surface-code patch. In the following analysis, we study two key factors that affect mapping performance: (i) the proportion of noisy channels used for transmission, and (ii) the physical error rates of the high-quality channels.

B. Benchmarks

- **CIAM**: The proposed correlated-importance-aware mapping strategy, which assigns d^2 error rates to data qubits in a distance-d surface-code patch using Algorithm 1.
- **CIAM_trim_n**: A trimmed variant of CIAM that skips the final n iterations of the mapping procedure. In the extreme case $n = d$, it reduces to random mapping.
- **CIAM_trim_LEC**: A trimmed variant of CIAM that computes importance using only $(d + 1)$ logical error chains, consisting of d vertical chains and one diagonal chain.
- **CIAM_ALL**: An extended version of CIAM that incorporates both logical-X and logical-Z error chains in the computation of qubit importance.
- **Count**: A baseline importance estimator that assigns importance to each data qubit by counting the number of logical error chains traversing it.

C. Logical error rates based on proportion of noisy channels

Firstly, we analyze how the proportion of noisy channels affects the performance of *CIAM* and its trimmed variants. In our setup, noisy channels have a physical error rate of approximately 20%, while high-quality channels have an error rate of about 2%. Fig. 7(a) presents results for a distance-9 surface-code patch.

We observe that when the trimming parameter $n \leq 6$, the trimmed variants perform comparably to *CIAM*. This indicates that mapping only the top $(9 - 6) \times 9 = 27$ most important qubits to low-error channels is sufficient to retain near-optimal performance. In this setting, $n = 6$ provides the best trade-off between logical error rate and computational cost. However, the optimal value of n may change under different ratios between noisy and high-quality channel error rates (here 20% vs. 2%).

In addition, *CIAM_trim_LEC* shows the effect of reducing the number of logical error chains used in importance estimation. Although this strategy significantly reduces computational cost, its performance is only marginally better than random mapping (*CIAM_trim_9*). This suggests that trimming logical error chains is generally not beneficial.

D. Logical error rates based on quality of channels

Next, we analyze how the average error rate of high-quality channels affects the performance of *CIAM* and its trimmed variants. In these experiments, we fix the noisy-channel error rate above 20% and set the proportion of noisy channels to 0.2 (with similar trends observed for proportions below 0.6). We again use a distance-9 surface-code patch, as shown in Fig. 7(b).

In contrast to previous results, reducing the number of iterations consistently degrades performance in this setting, and no optimal trimming level is observed. This indicates that when the noisy-channel proportion is fixed, trimming serves as a controllable trade-off between computational cost and logical error rate, rather than yielding an optimal configuration.

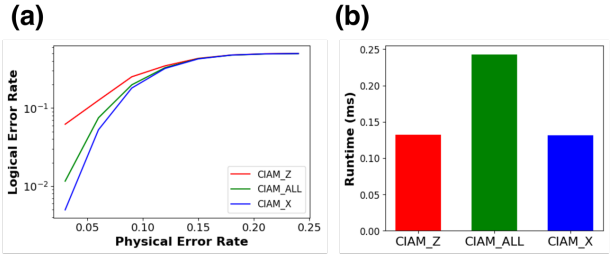

Fig. 8. Comparison between the proposed *CIAM* method (prioritized for logical-X/Z measurements) and its comprehensive-protection version (*CIAM_ALL*). **(a)** Logical error rates with respect to the physical error rate of the high-quality channels. **(b)** Average runtime of each method.

Fig. 7(c) further shows the runtime (in milliseconds) for a distance-9 code, which also decreases consistently with the trimming number applied.

E. Protection against both X and Z logical errors

Fig. 8(a) compares *CIAM* under different choices of logical error chains used for importance estimation. Here, *CIAM_X* and *CIAM_Z* use only logical-X or logical-Z error chains, respectively, while *CIAM_ALL* incorporates both. Here, we evaluate logical error rates using logical-X measurements. Thus, *CIAM_X* achieves the best performance, as it is directly optimized for the relevant error structure. In contrast, *CIAM_ALL* achieves intermediate performance due to balancing both types of chains, while *CIAM_Z* performs worst because it does not account for logical-X error chains. In addition, Fig. 8(b) shows the corresponding runtime. Both *CIAM_X* and *CIAM_Z* require significantly less computation time than *CIAM_ALL*, since they consider only a subset of logical error chains.

Therefore, if the surface-code patch is dedicated to either logical-X or logical-Z operations, a standard *CIAM_X* or *CIAM_Z* is sufficient and more efficient. However, for general-purpose use where both logical operators are relevant, *CIAM_ALL* provides a more robust and balanced solution.

F. Validation of importance estimation

We further evaluate the effectiveness and computational cost of the importance estimation method used in *CIAM*, where the importance of a data qubit is defined as the sum of the error rates of all logical error chains passing through it. We compare this against a baseline estimator, *Count*, which assigns importance by simply counting the number of logical error chains traversing each data qubit. This provides a purely combinatorial heuristic that ignores chain-dependent error rates and global structure.

We evaluate both methods in terms of logical error rate and runtime. Logical performance is measured across increasing code distances, while runtime reflects the average execution time of the importance estimation procedure at each scale. This enables a joint assessment of error suppression capability and computational cost.

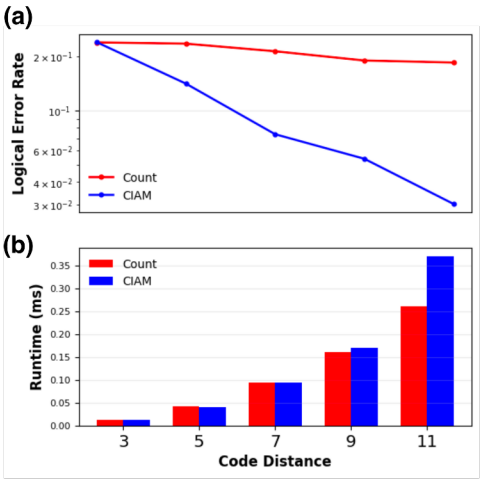

Fig. 9. Comparison of the proposed *CIAM* method against a naive *Count*-based baseline under different metrics. **(a)** Logical error rates across code distances (log scale), showing that *CIAM* consistently outperforms the baseline. **(b)** Average runtime, illustrating the additional computational overhead of *CIAM* compared to the baseline.

As shown in Fig. 9(a), *CIAM* consistently outperforms the *Count* baseline across all tested code distances, with the performance gap increasing as the code distance grows. This indicates that incorporating error-rate-weighted chain information yields a more accurate importance estimate, particularly in larger surface codes. Fig. 9(b) shows that this improvement comes at the cost of higher runtime, especially at larger code distances. However, the trade-off is favorable in common networking scenarios where logical error suppression is the primary objective.

IX. Conclusion

We address the problem of teleporting a $d \times d$ surface code using a fixed budget of n noisy entangled pairs. We formalize this problem as an entanglement allocation problem, and decompose it into two subproblems: the distillation resource partition problem and the qubit mapping problem. We solve the the distillation resource partition problem by developing representative partition strategies using different prioritized numbers and thresholds; and we solve the qubit mapping problem by developing a correlated-importance-aware algorithm that iteratively finds and updates the most important data qubits that is not yet mapped. Through simulations, we show that our proposed prioritized partition after a threshold strategies outperform the baseline random partition strategies under all scenarios. However, there is no single strategy that consistently yields the best performance. Thus, given different numbers of entangled pairs or different fidelity, one needs to compare all representative partition strategies to find the one with the best performance. In addition, we also show that our proposed correlated-importance-aware mapping algorithm, along with its trimmed variant, is effective in reducing logical error rates under different scenarios.

REFERENCES

[1] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*. Cambridge University Press, 2002.

[2] Z. Xiao, J. Li, K. Xue, N. Yu, R. Li, Q. Sun, and J. Lu, "Purification scheduling control for throughput maximization in quantum networks," *Communications Physics*, vol. 7, no. 1, p. 307, 2024.

[3] Y. Zeng, J. Zhang, J. Liu, Z. Liu, and Y. Yang, "Multi-entanglement routing design over quantum networks," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pp. 510–519, IEEE, 2022.

[4] Y. Sun, H. Ochiai, and H. Esaki, "Decentralized deep learning for multi-access edge computing: A survey on communication efficiency and trustworthiness," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 6, pp. 963–972, 2021.

[5] Y. Sahni, J. Cao, L. Yang, and Y. Ji, "Multi-hop multi-task partial computation offloading in collaborative edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1133–1145, 2020.

[6] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang, "Multiresource allocation: Fairness–efficiency tradeoffs in a unifying framework," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1785–1798, 2013.

[7] R. Tourani, S. Misra, T. Mick, and G. Panwar, "Security, privacy, and access control in information-centric networking: A survey," *IEEE communications surveys & tutorials*, vol. 20, no. 1, pp. 566–600, 2017.

[8] Y. Chen, W. Trappe, and R. P. Martin, "Detecting and localizing wireless spoofing attacks," in *2007 4th Annual IEEE Communications Society Conference on sensor, mesh and ad hoc communications and networks*, pp. 193–202, IEEE, 2007.

[9] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, "Purification of noisy entanglement and faithful teleportation via noisy channels," *Physical review letters*, vol. 76, no. 5, p. 722, 1996.

[10] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Physical Review A*, vol. 54, no. 5, pp. 3824–3851, 1996.

[11] W. Dür and H. J. Briegel, "Entanglement purification and quantum error correction," *arXiv preprint arXiv:0705.4165*, 2007.

[12] J. Li, M. Wang, K. Xue, R. Li, N. Yu, Q. Sun, and J. Lu, "Fidelity-guaranteed entanglement routing in quantum networks," *IEEE Transactions on Communications*, vol. 70, no. 10, pp. 6748–6763, 2022.

[13] H. J. Briegel and R. Raussendorf, "Persistent entanglement in arrays of interacting particles," *Physical Review Letters*, vol. 86, no. 5, pp. 910–913, 2001.

[14] E. T. Campbell, "Purification of faulty ancilla qubits by quantum error correction," *Physical Review A*, vol. 76, no. 4, p. 040302, 2007.

[15] M. Zwerger, W. Dür, and H. J. Briegel, "Measurement-based quantum repeaters," *Physical Review A*, vol. 85, no. 6, p. 062326, 2012.

[16] J. Hannegan, J. D. Siverns, J. Cassell, and Q. Quraishi, "Improving entanglement generation rates in trapped-ion quantum networks using nondestructive photon measurement and storage," *Physical Review A*, vol. 103, no. 5, p. 052433, 2021.

[17] A. Y. Kitaev, "Quantum computations: algorithms and error correction," *Russian Mathematical Surveys*, vol. 52, no. 6, p. 1191, 1997.

[18] A. Y. Kitaev, "Quantum error correction with imperfect gates," in *Quantum communication, computing, and measurement*, pp. 181–188, Springer, 1997.

[19] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of physics*, vol. 303, no. 1, pp. 2–30, 2003.

[20] S. B. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," *arXiv preprint quant-ph/9811052*, 1998.

[21] M. H. Freedman and D. A. Meyer, "Projective plane and planar quantum codes," *Foundations of Computational Mathematics*, vol. 1, pp. 325–332, 2001.

[22] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4452–4505, 2002.

[23] A. G. Fowler, "Optimal complexity correction of correlated errors in the surface code," *arXiv preprint arXiv:1310.0863*, 2013.

[24] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A*, vol. 86, no. 3, p. 032324, 2012.

[25] J. R. Wootton, A. Peter, J. R. Winkler, and D. Loss, "Proposal for a minimal surface code experiment," *Physical Review A*, vol. 96, no. 3, p. 032338, 2017.

[26] O. Higgott, T. C. Bohdanowicz, A. Kubica, S. T. Flammia, and E. T. Campbell, "Fragile boundaries of tailored surface codes and improved decoding of circuit-level noise," *arXiv preprint arXiv:2203.04948*, 2022.

[27] O. Higgott and C. Gidney, "Sparse blossom: correcting a million errors per core second with minimum-weight matching," *arXiv preprint arXiv:2303.15933*, 2023.

[28] A. G. Fowler, "Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $o(1)$ parallel time," *arXiv preprint arXiv:1307.1740*, 2013.

[29] N. Delfosse and N. H. Nickerson, "Almost-linear time decoding algorithm for topological codes," *Quantum*, vol. 5, p. 595, 2021.

[30] C. T. Chubb and S. T. Flammia, "Statistical mechanical models for quantum codes with correlated noise," *Annales de l'Institut Henri Poincaré D*, vol. 8, no. 2, pp. 269–321, 2021.

[31] M. McEwen, L. Faoro, K. Arya, A. Dunsworth, T. Huang, S. Kim, B. Burkett, A. Fowler, F. Arute, J. C. Bardin, *et al.*, "Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits," *Nature Physics*, vol. 18, no. 1, pp. 107–111, 2022.

[32] A. Strikis, S. C. Benjamin, and B. J. Brown, "Quantum computing is scalable on a planar array of qubits with fabrication defects," *Physical Review Applied*, vol. 19, no. 6, p. 064081, 2023.

[33] S. F. Lin, J. Viszlai, K. N. Smith, G. S. Ravi, C. Yuan, F. T. Chong, and B. J. Brown, "Codesign of quantum error-correcting codes and modular chiplets in the presence of defects," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, pp. 216–231, 2024.

[34] K. Yin, H. Zhang, Y. Shi, T. Humble, A. Li, and Y. Ding, "Flexiscd: Flexible surface code deformer for dynamic defects," *arXiv preprint arXiv:2405.06941*, 2024.

[35] K. Tiurev, P.-J. H. Derks, J. Roffe, J. Eisert, and J.-M. Reiner, "Correcting non-independent and non-identically distributed errors with surface codes," *Quantum*, vol. 7, p. 1123, 2023.

[36] Y. Lensky, V. Sivak, K. Kechedzhi, and I. Aleiner, "Modeling the performance of the surface code with non-uniform error distribution: Part 1," in *APS March Meeting Abstracts*, vol. 2024, pp. N51–011, 2024.

[37] V. Sivak, M. Newman, C. Jones, H. Schurkus, D. Kafri, Y. Lensky, P. Klimov, K. Kechedzhi, and V. Smelyanskiy, "Modeling the performance of the surface code with non-uniform error distribution: Part 2," in *APS March Meeting Abstracts*, vol. 2024, pp. N51–012, 2024.

[38] A. G. Fowler, D. S. Wang, C. D. Hill, T. D. Ladd, R. Van Meter, and L. C. Hollenberg, "Surface code quantum communication," *Physical review letters*, vol. 104, no. 18, p. 180503, 2010.

[39] S. Muralidharan, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang, "Ultrafast and fault-tolerant quantum communication across long distances," *Physical review letters*, vol. 112, no. 25, p. 250501, 2014.

[40] T. Hu, J. Wu, and Q. Li, "Quantum network routing based on surface code error correction," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1236–1247, IEEE, 2024.

[41] A. Paler, S. J. Devitt, K. Nemoto, and I. Polian, "Mapping of topological quantum circuits to physical hardware," *Scientific reports*, vol. 4, no. 1, p. 4657, 2014.

[42] A. Wu, G. Li, H. Zhang, G. G. Guerreschi, Y. Ding, and Y. Xie, "Mapping surface code to superconducting quantum processors," *arXiv preprint arXiv:2111.13729*, 2021.

[43] T. Hu, J. Wu, and Q. Li, "Disparity surface code: Optimizing error correction in quantum networks," in *2025 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1, pp. 2493–2504, IEEE, 2025.

[44] C. Gidney, "Stim: a fast stabilizer circuit simulator," *Quantum*, vol. 5, p. 497, 2021.