Disparity Surface Code: Optimizing Error Correction in Quantum Networks

Tianjie Hu, Jindi Wu, and Qun Li
Department of Computer Science, William & Mary, Williamsburg, VA 23185, USA

***Abstract*— Practical quantum computing requires error rates much lower than those achievable with physical qubits. Surface codes provide a reliable method to implement logical qubits using physical qubits. However, when these physical qubits are transmitted within quantum networks, their error rates can vary due to differences in the routing paths. Our experimental results indicate that the geometric arrangement of these physical qubits within a surface code significantly impacts its logical error rate. In this paper, we address this issue by proposing disparity surface codes, which are a group of geometric configurations for surface codes that adapt them for transmission within quantum networks. For a distance-d disparity surface code, only d physical qubits need to be transmitted with low error rates, and we identify multiple equivalent choices for selecting these d qubits. Additionally, we propose Di-MWPM, a new decoder designed to exploit physical error rate disparity for enhanced error correction. Results demonstrate that disparity surface codes maintain favorable performance in different network scenarios, even under high noise conditions. Our Di-MWPM decoder achieves improved Pauli-error thresholds.**

***Index Terms*— Quantum Network; Surface Code; Error Correction; Error Correction Decoder**

I. Introduction

Quantum networks offer a promising framework for addressing large-scale problems such as distributed quantum machine learning, quantum key distribution, and clock synchronization [1]–[12]. However, due to inherent noise during transmission within networks, achieving practical quantum networks requires error rates much lower than those achievable with physical qubits. Thus, an effective and efficient way of encoding quantum messages as logical qubits becomes demanding.

Surface codes are a class of quantum error correction codes that represent each logical qubit as a $d \times d$ square lattice of physical qubits. These d^2 physical qubits, referred to as data qubits, collectively encode the logical quantum state of the surface code. Additionally, each surface code contains $d^2 - 1$ measurement qubits, which facilitate error detection and correction. One of the key advantages of surface codes is their high tolerance to noise [13]–[16], making them a promising encoding method for quantum communication. However, due to limited quantum computing resources within contemporary quantum networks [17], transmitting an entire surface code through a single routing path is often infeasible. Instead, data qubits within each surface code are typically transmitted separately through multiple routing paths. Due to variations in the length and transmissivity of these paths, the resulting physical error rates of the data qubits exhibit disparity [18]. As discussed later in Section V, these disparities can have a profound impact on the logical error rate of each surface code, depending on how these data qubits are geometrically arranged. This leads to an assignment problem of determining optimal routing path for each data qubit. In the meantime, identifying the minimal subset of data qubits that must be transmitted through routes with low error rates arises as another problem, as it is critical to minimizing the cost of quantum communication resources.

In this paper, we address these two problems by proposing **disparity surface codes**, a group of geometric configurations for surface codes that adapt them for transmission within quantum networks. Disparity surface codes significantly reduce logical error rates while requiring only a small subset of data qubits to be transmitted with low physical error rates. Within each disparity surface code, certain data qubits form the "core" of the code, which needs to be transmitted with low error rates to ensure fault tolerance of the code. The remaining data qubits can be transmitted with slightly higher error rates without significantly impacting the overall performance of the code. The size of the core is deliberately kept small, and is equal to the minimal length of the logical operator of the code. Thus, for a distance-d disparity surface code, only d data qubits need to be transmitted with low error rates. Furthermore, we also identify multiple geometries for the core, allowing disparity surface codes to better adapt to quantum processors with varying connectivity and topological constraints.

We also propose a specialized error correction decoder tailored for disparity surface codes. While conventional surface code decoders can be directly applied to disparity surface codes, they either overlook or fail to fully exploit the error rate disparity between the data qubits in cores and the remaining data qubits. To address this, we propose the **Di-MWPM decoder**, designed to efficiently and effectively leverage the error rate disparity, optimizing the performance of disparity surface codes. For this paper, we adopt depolarization channels as the noise model, with each qubit subject to single-qubit Pauli errors. Although our approach is demonstrated under minimal noise assumptions to emphasize its core contribution, it can be easily extended to more realistic network environments.

Our main contributions are summarized as follows:

- We propose disparity surface codes, a group of geometric layouts for surface codes that leverage error rate disparity observed among data qubits. Disparity surface codes demonstrate favorable scalability with code size.

979-8-3315-5736-2/25/$31.00 ©2025 IEEE
DOI 10.1109/QCE65121.2025.00270

- We provide an extensive analysis on the size and geometry of the cores in disparity surface codes. For each distance-d disparity surface code, we find the size of its core to be d, and we identify multiple alternative choices for the geometry of the core, allowing disparity surface codes to be implemented on quantum processors with varying connectivity and topology.
- We present Di-MWPM, a novel decoder that efficiently utilize the error rate disparity among data qubits for error correction. It surpasses its baselines decoder in error thresholds, facilitating real implementation of disparity surface codes.

II. Background

A. Surface code

Surface code was developed to effectively detect and correct quantum errors. Among various quantum error correction codes, it stands out for its compact size and high error threshold. Each surface code consists of two groups of physical qubits: data qubits and measurement qubits. The quantum information of each logical qubit is encoded and stored within the data qubits, while the measurement qubits protect the quantum information from random errors. Illustrated in Fig. 1(a), each data qubit is connected to its neighboring measurement qubits, comprising two measure-X qubits and two measure-Z qubits, except on the boundaries. If a random error occurs in a specific data qubit, all of its neighboring measurement qubits would indicate the error: its Y or Z types errors are captured by the measure-X qubits reflect, and its X or Y types errors are captured by the measure-Z qubits. Hence, measurement qubits are also known as syndrome qubits.

Each surface code encodes a single logical qubit. There exist variants on surface codes such as X-cut, Z-cut, or multi-cuts surface codes [13] that are encoded with a multi-qubits state. In this paper, we use surface codes as logical qubits that carry single-qubit states. Their logical states can be modified by logical operations. Each logical operation on surface codes is defined as the tensor product of physical operations on a string of an odd number of data qubits connecting two boundaries. For example, a logical Z operator is typically defined as a sequence of single-qubit Pauli-Z gates acting on an arbitrary row of data qubits. Surface codes are subject to logical errors, which arise when logical Pauli operators are unintentionally applied. However, as will be demonstrated in the next subsection, physical qubit errors in a surface code can be corrected in a timely manner before they accumulate to be logical errors.

B. Error correction

Error correction can be performed on surface codes for correcting physical errors in individual data qubits. To initialize a surface code, it begins with only the data qubits. Each surface code is commonly prepared in its logical $|0\rangle$ state, so all data qubits are initialized in the $|0\rangle$ state. Next, we put on the measurement qubits and connect them to their neighboring data qubits using quantum CNOT gates, as depicted in Fig.1(b) and (c). For the ease of syndrome readout, all measurement qubits are commonly prepared in their $|0\rangle$ states. After all the CNOT gates are established, we need to perform at least one round of error correction cycle to complete the setup of surface code. As in Fig.1(b) and (c), each cycle involves a sequence of CNOT gates and a final measurement on each measurement qubit. Each measurement forces its neighboring data qubits to be in a simultaneous eigenstate of this initial measurement outcome.

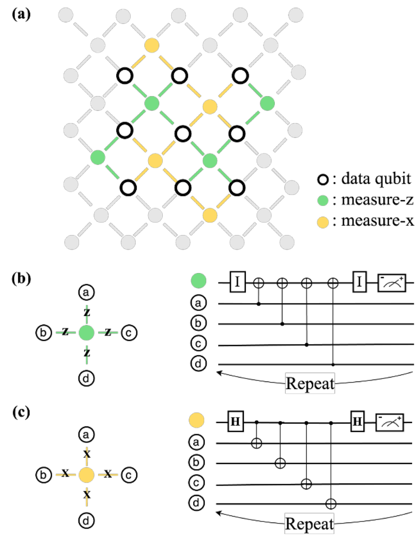


Fig. 1. **(a)** Distance-3 surface code. Distance refers to the minimum number of data qubits for a logical operation. Open dots represent data qubits, and filled dots represent measurement qubits (green for measure-z, yellow for measure-x). **(b)** Quantum circuit for measure-z qubits. **(c)** Quantum circuit for measure-x qubits.

At this point, we successfully initialize a surface code, with all data qubits and measurement qubits becoming highly entangled. And we name the first round of measurement result of all measurement qubits as the quiescent state. If no error occurs in the future, all measurement results in subsequent error correction cycles would stay in this quiescent state. And if the result of any measurement qubit changes, then at least one of its neighboring data qubits has an error, and we mark this measurement qubit as a “syndrome”. In each error correction cycle, all syndromes collectively form a syndrome pattern. Next, an error correction decoder is applied to the syndrome pattern to find all data qubits with errors. The decoder aims to connect all syndromes either into pairs or to the boundary, with each syndrome must be connected and connected exactly once. Any data qubit along the connection are determined with errors. The decoding result does not need to match the actual erroneous data qubits, and only needs to be equivalent upon stabilizers [13]. However, if the decoding result differs from the actual erroneous data qubits by a logical operator, then a logical error will be raised. Notably, logical errors in surface codes do not induce any syndrome, and thus, will not be caught by the measurement qubits.

III. Related Work

Due to the non-cloning theorem within quantum mechanics, most classical error correction code [19]–[21] cannot be directly applied in quantum networks. Most quantum error correction codes are stabilizer codes. Calderbank-Shor-Steane (CSS) codes have been the mainstream stabilizer codes, including Shor codes [22]–[24], Steane codes [25]–[27], and more [28]–[30]. Another main branch from the stabilizer codes is the quantum LDPC codes [31]–[33] known for their fast and parallelizable decoding process. Recent works on non-additive quantum codes [34]–[36] that are not associated with stabilizers also find potentials in computations. An extensive list of error correction codes is available in [37].

Surface codes, which arise as a promising stabilizer code, was firstly proposed in [38]–[40] as toric codes. Later the torus geometry was proven to be unnecessary, and a planar version of surface codes had been developed [41]–[43]. Many efforts have been made in the analysis on surface codes about their performance [13], [44], [45], and their applications [46]–[48]. Various decoders [14], [49]–[54] have been proposed to enhance the error threshold of surface codes. To the best of our knowledge, this work is the first to investigate the influence of assigning low error rates to a specific subset of data qubits in surface codes.

IV. Surface Code in Quantum Networks

In quantum networks, different routing paths often exhibit varying error rates for qubit transmission. This **error rate disparity** arises primarily from differences in path length and the transmissivity of optical fibers, as illustrated in Fig. 2(a). It may also result from a differing number of entanglement purification or error correction rounds applied along each path, depending on the routing protocol employed.

In an ideal scenario, all data qubits would be transmitted along the route with the lowest error rate, such as Route 1 in Fig. 2(a). However, practical constraints make this infeasible. The limited capacity of each optical fiber, along with the simultaneous communication demands of multiple user pairs, prevents all qubits from being routed through the most reliable path. As a result, in a quantum network, different qubits may be transmitted through different paths, each with a different error rate.

Now, consider a surface code composed of n data qubits, each positioned differently on the surface code and routed through a distinct path with its own error rate. A natural question arises: Are all data qubits in the surface code equally sensitive to transmission errors? In other words, does assigning higher-quality (i.e., lower-error-rate) paths to certain qubits have a more significant impact on the overall logical error rate of the surface code?

Our observations indicate that not all physical qubits in a surface code contribute equally to its logical error performance — some are more critical than others. Therefore, if we have the opportunity to assign lower-error-rate paths to only a subset of the qubits, the choice of which qubits to prioritize can make a substantial difference in the resulting logical error rate.

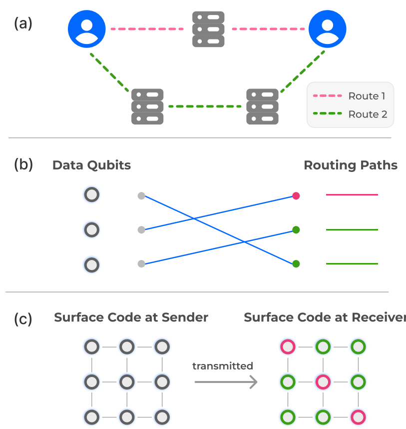


Fig. 2. **(a)** Routing paths between two users in quantum networks. Route 1 in red represents a path with lower error rate, and Route 2 in green represent a path with higher error rate. **(b)** Assignment for data qubits to routing paths. In this example, three data qubits need to be transmitted, within which one can be transmitted through Route 1, and the other two need to be transmitted through Route 2. **(c)** Assignment for a complete surface code to routing paths. In this example, the three data qubits in the diagonal are transmitted through Route 1, and the other six data qubits are transmitted through Route 2.

To address this, we ask: How should we select a subset of qubits for transmission over higher-quality channels in order to maximize the improvement in logical error rate? For instance, consider the transmission of data qubits in the surface code shown in Fig. 2(c). Suppose three qubits can be transmitted through the high-quality Route 1 and the remaining six through the lower-quality Route 2. An effective strategy is to assign the three qubits located along the diagonal of the surface code, where qubits tend to have higher importance, with the more reliable path (Route 1), while the remaining qubits are transmitted through Route 2. As we will show, such geometry-aware selection leads to significant improvements in logical performance.

V. Impact of Low-Error-Rate Data Qubits on Surface Code

In this section, we investigate which subsets of data qubits in a surface code, specifically their number and positions, most significantly influence the logical error rate when transmitted with lower error rates. We identify those as critical data qubits whose improved transmission quality results in the greatest reduction in logical errors. Furthermore, we examine how the size of the surface code impacts the minimum number of low-error-rate qubits required to achieve a substantial improvement in logical error performance.

A. *Finding the most important data qubit*

Our analysis begins by investigating the single most important data qubit in a surface code. In Fig. 3(a), we analyze a distance-3 surface code. This code contains nine data qubits, so we have nine choices, protecting one data qubit with a low error rate in each choice while assigning the other eight data

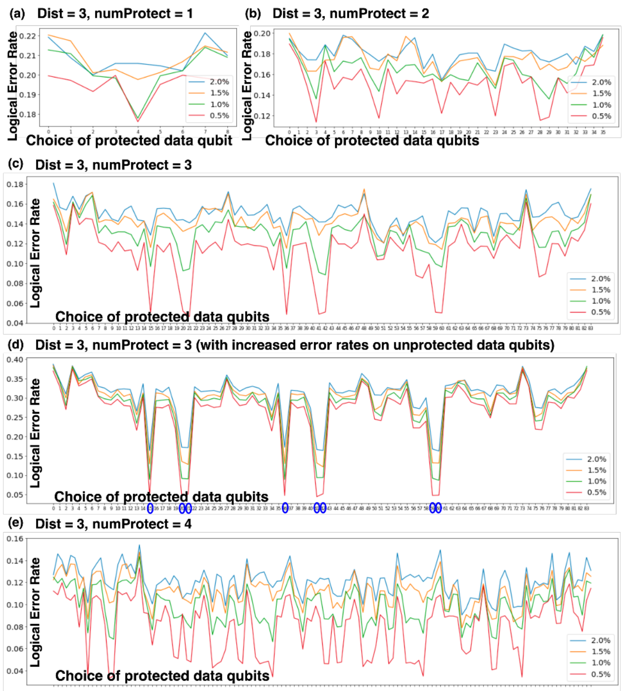


Fig. 3. **(a)** Logical error rates of a distance-3 surface code with a random qubit protected with a lower error rate. **(b)** Distance-3 surface codes with two data qubits being protected. **(c)** Distance-3 surface codes with three data qubits being protected. **(b)** Distance-3 surface codes with three data qubits being protected, and the other data qubits are with a 20% error rates. **(e)** Distance-3 surface codes with four data qubits being protected. All logical error rates are evaluated using logical X measurements.

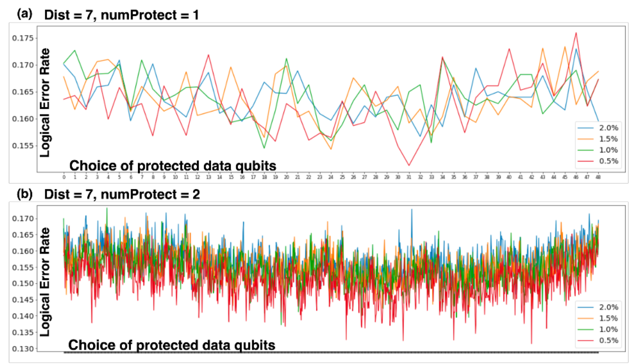


Fig. 4. **(a)** Logical error rates of a distance-7 surface code with a random qubit protected with a lower error rate. **(b)** Distance-7 surface codes with two data qubits being protected. All logical error rates are evaluated using logical X measurements.

qubits a higher error rate. The x-axis of Fig. 3(a) represents these nine choices: for example, "choice 0" represents only protecting the first data qubit (counting from left to right, top to bottom on the lattice), and "choice 1" represents only protecting the second data qubit.

We conduct four sets of experiments, varying the error rate of the protected data qubit (0.5%, 1%, 1.5%, and 2%) in each experiment set, while keeping the error rate of the unprotected data qubits constant at 10%. Across all four sets of experiments, we observe that the data qubit positioned at the center of the distance-3 surface code (labeled as 4 on the x-axis) has the greatest influence on the logical error rate. However, a distance-7 surface code analyzed in Fig. 4(a) exhibits a different result, with no clear indication of a most important data qubit. This is primarily due to the different sizes of these two surface codes, since a distance-7 surface code consists of 49 data qubits, and merely protecting one data qubit among them is clearly not enough for an obvious reduction on the logical error rate.

B. Finding the most important data qubit pair

Building upon the previous analysis, we now investigate the most important subset of two data qubits in a surface code. In Fig. 3(b), we examine a distance-3 surface code by conducting experiments in which two data qubits are randomly selected to be protected with low error rates, while the remaining seven data qubits are assigned higher error rates. Thus, in total we have $\binom{9}{2} = 36$ choices. The x-axis represents these 36 choices, each corresponding to a unique pair of protected data qubits. For example, "choice 0" represents protecting the first and second data qubit, and "choice 1" represents protecting the first and third data qubits. And Fig. 4(b) analyzes a distance-7 surface code following the same approach, with two data qubits protected with low error rates and the remaining 47 assigned higher error rates. Similarly, we observe several assignments for distance-3 surface codes with relatively lower logical error rates, while the variation becomes less obvious in distance-7 surface codes.

C. Finding the most important subset of data qubits

When investigating the most important subset of three data qubits in a distance-3 surface code, the logical error rate differences between choices become the most obvious, as shown in Fig. 3(c). These differences become even more obvious when we increase the error rates of the unprotected data qubits from 10% to 20%, as illustrated in Fig. 3(d). The eight choices with the lowest logical error rates are marked in blue and will be further analyzed in Section VI-A. Another observation here is that in both Fig. 3(c) and (d), the eight well-performing choices demonstrate similar logical error rates when the three protected qubits are assigned a low error rate of 0.5% (indicated by the red plots). In contrast, the other choices exhibit logical error rates of approximately 13% in Fig. 3(c), which increase dramatically to 30% in Fig. 3(d) when the unprotected data qubits experience higher error rates. These findings highlight the potential of surface codes to maintain functionality even under extremely poor network quality of service (QoS), as long as a carefully selected subset of data qubits is well-protected. We will further analyze this property of surface codes in Section VI-D.

Notably, when we investigate the most important subset of four data qubits in a distance-3 surface code, the difference

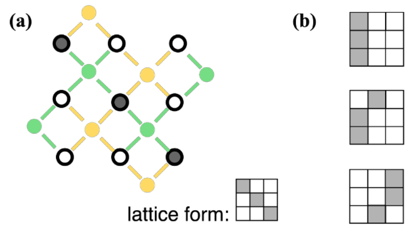


Fig. 5. **(a)** Disparity Surface Code of distance 3. Green and yellow dots represent measurement qubits, respectively for measure-z and measure-x. Black dots (solid: core, empty: the remaining) represent data qubits. **(b)** Alternative choices of cores for distance 3 disparity surface codes. When represented in the lattice form, each square represents a data qubit, and the filled squares represent the core.

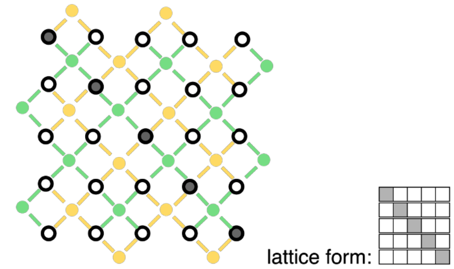


Fig. 6. Disparity Surface Code of distance 5. Green and yellow dots represent measurement qubits, respectively for measure-z and measure-x. Black dots (solid: core, empty: the remaining) represent data qubits.

in logical error rates between choices becomes less obvious, as shown in Fig. 3(e). Specifically, most choices exhibit moderately low logical error rates, suggesting that protecting four out of the nine data qubits is generally sufficient to achieve a noticeable reduction in the logical error rate. Another possible explanation is that as in Fig. 3(c), there are eight identified subsets of three data qubits with the lowest logical error rates. Adding one additional data qubit to any of these subsets qualifies it as a potential candidate for the most important subset of four data qubits, thereby diminishing the differences in logical error rates between choices.

A similar pattern emerges in the case of distance-5 surface codes when we try to find the most important subset of five data qubits. Among the 53,130 possible choices, 52 exhibit relatively lower logical error rates. And similarly, the differences between choices become less obvious when searching for the most important subset of four or six data qubits in distance-5 surface codes. However, despite these findings, this exhaustive search approach becomes impractical for surface codes of distance 7 or larger due to the rapid growth of the search space. Specifically, for a distance-d surface code, the number of possible subsets of size d is given by $\binom{d^2}{d}$, which increases exponentially with d.

VI. Disparity Surface Code

As mentioned previously in Section IV, it is often impractical to route all data qubits in a surface code through paths with the lowest error rates. Thus, we aim to identify the minimal subset of data qubits that must be transmitted with low error rates to maintain a low logical error rate. For the rest of this paper, we refer to these minimal subsets as the **core** of each surface code, while the remaining data qubits are referred to as the **remaining** data qubits.

In this section, we build on the observations from Section V, which demonstrates the impact of the number of position of low-error-rate data qubits on the logical error rate of surface codes. Leveraging these observations, we identify the cores in different surface codes and propose an effective strategy for addressing the assignment problem for surface codes in quantum networks. Specifically, our strategy involves assigning all data qubits in the core to routing paths with the lowest error rates, while the remaining data qubits can be transmitted with slightly higher error rates.

To distinguish our approach from traditional surface codes, where all data qubits have roughly equal error rates (assuming no defects induced during fabrication or by cosmic rays), we name our surface codes with a protected core as the **disparity surface code**. This variant of surface code features a well-defined disparity in error rates, with the core data qubits being protected by lower error rates and the remaining data qubits tolerating slightly higher error rates. These specialized configurations provide a practical and efficient means of optimizing surface codes for use in quantum networks.

A. Optimal geometry of cores

As noted earlier in Fig. 3(d), we observe eight trials with significantly lower logical error rates. The geometry of one of these eight trials is shown in Fig. 5(a), where three data qubits (marked with solid dots) along the diagonal are assigned low error rates, while the remaining six data qubits (marked with empty dots) are assigned slightly higher error rates. The geometries for the other seven trials are listed in Fig. 5(c). In Section VI-B, we will demonstrate that the cores for distance-3 surface codes correspond precisely to these eight geometries.

For larger surface codes, performing an exhaustive search to identify cores becomes infeasible due to the exponential growth of the search space. However, we identify certain geometries that consistently yield low logical error rates across all surface code sizes. These geometries include the diagonal core shown in Fig. 5(a) and the column cores illustrated in the first row of Fig. 5(b). For example, Fig. 6 shows the diagonal core in a distance-5 surface code. Thus for each distance-d surface code, we can identify at least $(d+1)$ potential cores. As illustrated in Fig. 3(d), these cores exhibit similar performance in reducing logical error rates. In Section VI-C, we will explain why these geometries consistently serve as cores.

B. Optimal size of cores

We analyze the minimal size of the core in each disparity surface code by experimenting with incomplete cores and evaluating their impact on logical error rates. Each incomplete core is created by selecting only some of the data qubits within our proposed cores and assigning them low error rates, as

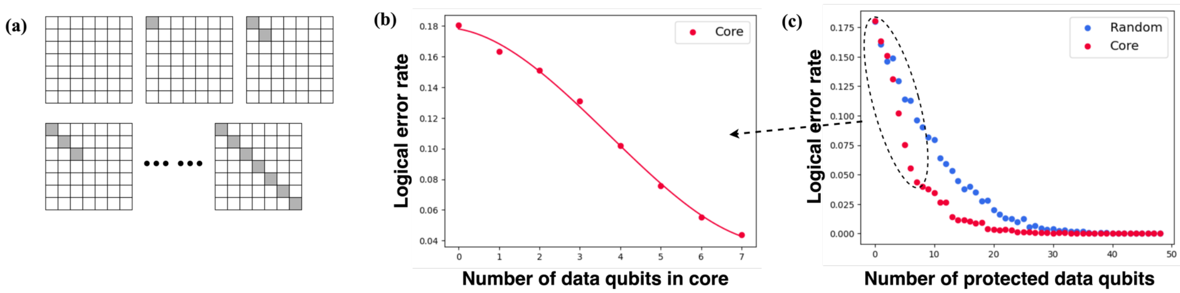


Fig. 7. **(a)** Incomplete cores (except the last one, which is a complete core) in disparity surface codes of distance 7. **(b)** Logical error rates based on number of data qubits in the incomplete cores. **(c)** Comparison of proposed cores and randomly selected qubits, with respect to number of data qubits protected with low error rates. Within plots, *Core* represents our disparity surface code with its core transmitted with comparatively lower error rates, and *Random* represents a basic surface code with randomly selected data qubits transmitted with comparatively lower error rates.

illustrated in Fig. 7(a). The results, shown in Fig. 7(b), indicate that removing the low-error-rate protection of any data qubit within the core leads to an approximate 2% increase in the logical error rate. Thus, when sufficient low-error-rate routing paths are available, users should always implement a complete core by assigning low error rates to all its data qubits.

Additionally, we analyze the effect of adding more data qubits into cores and protecting them with low error rates, as shown by the red scatter plot labeled "Core" in Fig. 7(c). We compare its performance against a random selection from all data qubits in surface codes to be protected with low error rates (labeled "Random" in Fig. 7(c)). The results show that cores consistently outperform random selections in reducing logical error rates. We also observe that the slope of the plot for cores is steepest (highlighted by the dashed circle) as the cores approach completion, indicating the most significant reduction in logical error rates. Beyond core completion, adding more protected data qubits shows diminishing marginal benefit in further reducing logical error rates. Thus, users may consider adding more data qubits to cores only when there are surplus computing resources available within the networks.

It is important to note that in Fig. 7(c), the removal of data qubits from cores is randomly performed among the core data qubits, while the addition of data qubits into cores are randomly selected from the remaining qubits. Thus, the red scatter plot does not necessarily represent the most important subsets of varying sizes identified in Fig. 4. An exhaustive search for these most important subsets becomes computationally infeasible for large surface codes. So far, we have identified fixed geometries for cores of size d in each distance-d surface codes. However, identifying fixed geometries for cores of arbitrary sizes remains an open challenge, assuming such configurations truly exist.

C. Why cores can reduce logical error rates

We analyze why protecting cores with lower error rates significantly reduces the logical error rates in disparity surface codes, using a distance-3 disparity surface code with a diagonal core (Fig. 5(a)). Logical errors in surface codes arise from unintended logical operators, either due to noise or erroneous

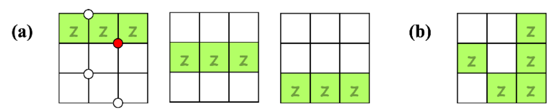


Fig. 8. **(a)** Three shortest logical Z operators for surface codes of distance 3. **(b)** A long logical Z operator for surface codes of distance 3. In the lattice form, each square represents a data qubit, and the green squares represent Pauli-Z gates applied to the corresponding data qubits. Each dot on the vertex represents a measurement qubit, where the empty dot does not have a syndrome and the red dot indicates a syndrome.

decoder results (Section II). In surface codes, a logical Z operator is commonly defined as a sequence of single-qubit Pauli-Z gates acting along a row of data qubits, as shown in Fig. 8(a). Thus, a logical Z operator in a distance-d surface code is typically of length d. Longer logical operators, such as the one in Fig. 8(b), also exist but are rarely used. These long logical operators are less likely to be unintentionally applied, due to their longer lengths. Thus, in the following, we focus our discussion on the shortest logical error chains.

For logical Z errors induced by noise, every data qubit along a logical operator must be flipped along the Z axis by noise. This chain of errors does not trigger syndromes in measurement qubits, making such errors undetectable by decoders. In disparity surface codes, the presence of a core with low error rates significantly reduces the likelihood of these error chains. Since every logical Z operator intersects with the diagonal core at least once, and thus, the error rate for logical Z errors decreases sharply. Assuming core error rates of 0.05 and remaining qubit error rates of 0.3:

- Without the core, the probability of a noise-induced logical error is $0.3^3 = 2.7\%$.
- With the core, the probability drops to $0.3^2 \times 0.05 = 0.45\%$, and is even lower if multiple intersections exist between the error chain and the core.

For logical Z errors induced by erroneous decoding, the noise needs not flip all data qubits in a logical operator. Instead, the noise flips only some of them, which induce syndromes among measurement qubits. A decoder may misinterpret these syndromes, flipping data qubits incorrectly and

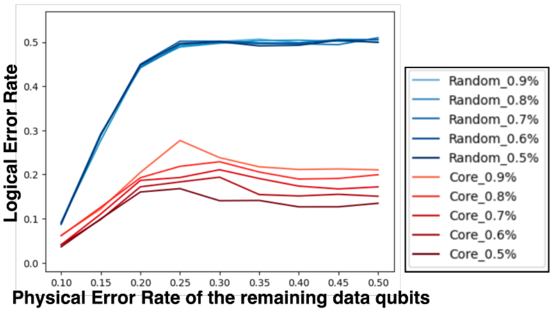


Fig. 9. Logical error rates of disparity surface codes (denoted as *Core*) and random assignment (denoted as *Random*) with respect to physical error rates of the remaining data qubits. Experiments are conducted on distance-9 surface codes. Physical error rates of data qubits in cores (or randomly selected when in *Random*) are varied for each set of experiments, denoted in the legend by the side of plots.

completing a logical error chain. Consider the first logical Z operator in Fig. 8(a), and suppose that noise flips the first two qubits in a row, inducing a syndrome at the red dot. The decoder can resolve this syndrome by either: flipping the first two qubits back, or flipping the third qubit instead. If the decoder chooses the second option, which requires fewer operations to resolve the syndrome, it inadvertently completes the logical Z error chain. Again, cores in disparity surface codes reduce the likelihood of such errors by intersecting the corresponding logical error chain. Assuming core error rates of 0.05 and remaining qubit error rates of 0.3:

- Without the core, the probability of a logical error due to decoder mistakes is $0.3 \times 0.3 = 9\%$.
- With a core, this probability drops to $0.3 \times 0.05 = 1.5\%$, and is even lower if multiple intersections exist between the error chain and the core.

Thus, the core in each disparity surface code acts as a "wall," impeding logical error chains from forming across the code. It effectively mitigates errors by serving as a minimal subset of data qubits that intersects all possible logical error paths. Therefore, as long as the data qubits in the core are transmitted with low physical error rates, the logical error rate of the disparity surface code remains low.

D. Disparity surface code in low QoS

Here, we analyze the performance of our disparity surface code in networks with poor quality of service, where most routing paths exhibit high error rates. Notably, in such scenarios, successful transmission of quantum messages becomes infeasible if all routing paths are subjected to high error rates. Thus, we assume that the core of each disparity surface code can still be transmitted with low error rates, while the remaining data qubits are transmitted through paths with significantly higher error rates. As shown in Fig. 9, under these conditions, our disparity surface codes consistently achieve much lower logical error rates compared to random assignment. Notably, when the error rates of the remaining data qubits exceed 0.2, the random assignments (denoted as $Random\_0.9\%$ to $Random\_0.5\%$) reach a logical error rate of approximately 50%, which is the highest logical error rate for a surface code (logical error rates above 50% typically indicate an issue with the error correction decoder). In contrast, the logical error rates of our disparity surface codes remain stable around 10%. Thus, disparity surface codes are significantly more robust under low QoS conditions.

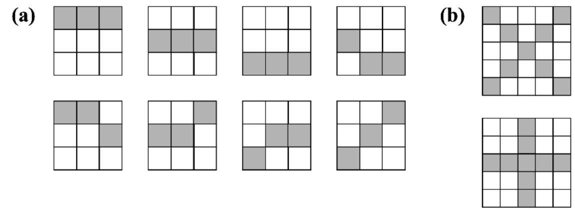


Fig. 10. **(a)** Cores for disparity surface codes of distance 3 protected along logical Z. **(b)** Two example cores for disparity surface codes of distance 5 with full protection.

E. Disparity surface code with measurement X, Z

All experiments above focus on logical X measurements for disparity surface codes, with the proposed cores effectively reducing logical error rates along the logical X axis. These cores can effectively prevent logical Z and Y errors. In the meantime, a different set of cores can be employed to protect disparity surface codes along their logical Z axis, as shown in Fig. 10(a). These cores, of size equal to the code distance d, act as "walls" preventing logical X errors from traversing vertically across the code.

Since the two sets of cores (one protecting along the logical X axis and the other along the logical Z axis) are entirely distinct, no core of size d exists that can simultaneously protect disparity surface codes along both logical X and Z axes. Thus, to implement a full protection along both logical axes, the minimal core size increases to at least $2d-1$ data qubits, as illustrated in Fig. 10(b). However, contemporary quantum computation rarely requires such full protection, as measurements are typically performed along one logical axis (X or Z) for each surface code, and logical non-demolition measurements are not yet fully developed.

VII. Disparity Surface Code Decoder

To further reduce the logical error rates of our disparity surface codes, we propose a novel decoder that leverages the unique error rate disparity between the core and the remaining data qubits. Conventional decoders, such as the PyMatching [49] library for generating all results in the above section, can be directly applied to disparity surface codes. However, most of them often fail to fully exploit this disparity. As a result, they may lead to suboptimal decoding outcomes. Consider the decoding scenarios illustrated in Fig. 11. Assume the core is implemented as in (a1), and (b) is the decoding graph $G = \{V, E, W\}$ observed by the decoder, where the weights W are the error rates of data qubits. There are two potential decoding results given by the decoder:

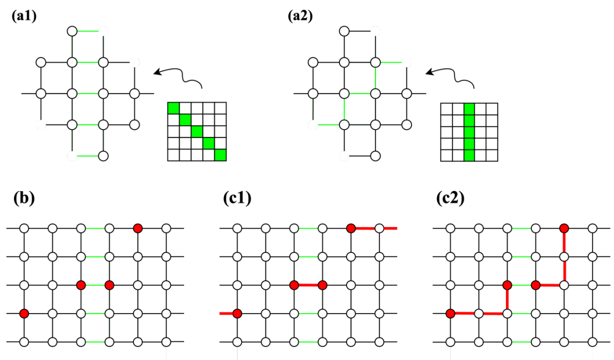


Fig. 11. **(a1)** and **(a2)** Decoding graphs for disparity surface codes. In each decoding graph, each vertex dot is a measurement qubit, and each edge is a data qubit. The five edges in green represent the core, with their corresponding lattice forms shown aside. **(b)** Decoding graph with syndromes. The four vertices in red represent four example syndromes. **(c1)** Decoding results without considering the core. **(c2)** Decoding results considering the core. The red edges in c1 and c2 represent the decoding results.

- Decoding Result (c1): This approach does not account for the low error rates of the core. It pairs up syndromes or connects them to boundaries using the shortest paths available, resulting in a decoding outcome with a minimal path length of four edges.
- Decoding Result (c2): This approach incorporates the low error rates of the core into the decoding process. Although the resulting path has a greater length of six edges, it avoids intersecting with the core, reflecting the lower likelihood of errors within the core.

In both cases, the decoder can resolve the syndromes by applying Pauli gates to the indicated data qubits. However, the two decoding outcomes differ by a logical error chain that traverse across the code horizontally: if the actual error pattern corresponds to (c2), the correction based on (c1) introduces a logical error, and vice versa. Although the actual error pattern may not match either (c1) or (c2) exactly, it is equivalent to one of them under stabilizer transformations. Theoretically, either (c1) or (c2) can be the actual error pattern, but given the error rate disparity between the core and the remaining data qubits, they are with different likelihoods.

Thus, by explicitly accounting for the error rate disparity, a core-aware decoder can better prioritize decoding outcomes that align with the lower error rates of the core. This reduces the probability of logical errors, particularly in scenarios where the shortest-path decoding strategy used in conventional methods is less likely to correspond to the actual error pattern. In the following, we detail the design of our core-aware decoder, the Di-MWPM decoder, and demonstrate its effective and efficient use of the error rate disparity between the core and the remaining data qubits.

The routine of our **Di-MWPM decoder** is adapted from the Blossom decoder [55], which is a minimum weight perfect matching (MWPM) decoder for surface codes. The Di-MWPM decoder follows the minimum weight perfect matching de-

Algorithm 1 Di-MWPM Decoder

Input: set of syndromes σ, data qubits in core ρ,
error rate of the core α, error rate of the remaining β
Output: estimated error pattern Γ

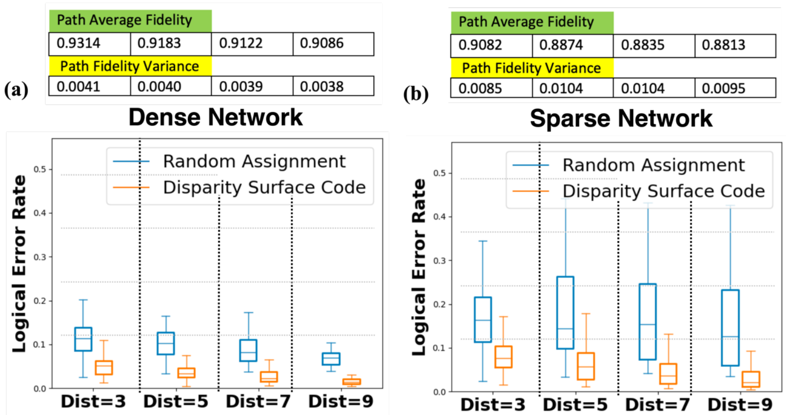
Create decoding graph $G = \{V, E, W\}$
Divide V into subsets L, M, R, where M covers ρ
$L = L \cup M,\ R = R \cup M$
Initialize path graph $G' = \{\sigma, \{\}, \{\}\}$
for $H = L, R$ **do**
  Let $\mathbb{H}$ be the induced graph by H in G
  for $u, v \in (\sigma \cap H)$ with $u \neq v$ **do**
    find shortest path w between u, v in $\mathbb{H}$
    compute weight for w as:
      $\sum_{(i,j)\in w}(-\ln(\alpha)\ \textbf{if}\ (i,j) \in \rho\ \textbf{else}\ -\ln(\beta))$
    add $w = (u, v)$ as a weighted edge into G'
  end for
end for
Apply Blossom to G' to get a perfect matching P
initialize Γ as empty
for (u, v) in P **do**
  break (u, v) into edges $(u, i_1), (i_1, i_2), \ldots, (i_k, v)$
  add these edges to Γ
end for

coder paradigm but incorporates specific modifications to account for the core structure of disparity surface codes. As in Alg. 1, the Di-MWPM decoder partitions the decoding graph G into two subgraphs, positioned on opposite sides of the core. Depending on the actual error rates of the core, these subgraphs can be either separated or overlapped.

First we talk about the ideal case, when the error rate of the core is extremely low ($\alpha \approx 0$). In this case, as discussed previously in Section VI-C, the core acts as a solid wall, completely preventing error paths from traversing across the core. Thus, the decoding graph G can be completely separated into two subgraphs:

- Left subgraph L, which contains all vertices and edges to the left of the core;
- Right subgraph R, which contains all vertices and edges to the right of the core.

And the middle subgraph M contains only the core, which serves as a mutual boundary between L and R. Next, a path graph G' is created by computing the shortest paths between every pair of syndromes within each subgraph L and R. These paths are efficiently computed using Dijkstra's algorithm. Any path attempting to cross the core is disallowed, ensuring no error chains traverse through M. Then the Blossom algorithm [55] is applied to G' to achieve a minimal weight perfect matching P. In the Blossom algorithm, syndromes are treated as circles with radii, and blossoms are created when circles collide. Syndromes within a blossom are paired to minimize the total weight. Finally, the shortest paths in P are reverted back to edges in the original graph G. These edges correspond to the data qubits requiring correction.



(c) Logical Error Rate in Dense Network

	Distance = 5	Distance = 7	Distance = 9
disp_xz: X	0.060	0.041	0.027
disp_x: X	0.038	0.025	0.014
disp_z: X	0.087	0.060	0.039
disp_xz: Z	0.080	0.065	0.055
disp_x: Z	0.129	0.139	0.141
disp_z: Z	0.039	0.038	0.031

(d) Logical Error Rate in Sparse Network

	Distance = 5	Distance = 7	Distance = 9
disp_xz: X	0.102	0.081	0.072
disp_x: X	0.065	0.049	0.039
disp_z: X	0.156	0.114	0.097
disp_xz: Z	0.173	0.175	0.176
disp_x: Z	0.264	0.282	0.313
disp_z: Z	0.107	0.115	0.125

Fig. 12. Logical error rates of disparity surface code versus random assignment in **(a)** dense networks and **(b)** sparse networks, both evaluated using logical X measurements. Average fidelity (fidelity$= 1-$error rate) and fidelity variance of all routing paths in each network scenario are included in the tables above. Tables **(c)** and **(d)** include logical error rates of different versions of disparity surface codes, evaluated using both logical X and logical Z measurements. The lowest logical error rate across each column is marked red.

However, in practical scenarios, the core is not perfectly reliable, and error paths crossing it are possible. Thus, in this case, the middle subgraph M includes not only the core but also its nearby vertices and edges, depending on α, the error rate of the core: as α, increases, M grows, resulting in a larger overlap between the left (L) and right (R) subgraphs. This overlap region enables syndromes to interact across the core. A path graph G' is then constructed as in the ideal case, but with key differences:

- Shortest paths are firstly computed locally in L and R;
- Syndromes in the overlap region M are considered part of both subgraphs, and thus, can be paired with syndromes in both L and R, or within itself.
- Paths crossing the M are still not allowed, meaning that a syndrome in L cannot be paired with a syndrome in R if neither of them is in M.

The rest procedure is the same as in the ideal case: the blossom algorithm is applied to G' to derive the data qubits requiring correction. Thus, in this practical scenario, the middle graph M, instead of the core, preserves its function as a logical wall that prevents decoding path made across the disparity surface code. This is also the reason why we make the size of M dependent on α: when α is low, the core remains a strong wall, with minimal overlap between L and R, and only a few decoding paths can be made across the core. Notably, in the worst case when α is the same as the error rate of the remaining data qubits, M grows to span the entire decoding graph G, and the Di-MWPM decoder converges to a standard MWPM decoder.

VIII. Evaluation

In this section, we evaluate the performance of disparity surface code by implementing it in different network scenarios. We compare its performance against baseline surface codes that randomly assign data qubits to scheduled routing paths.

A. Evaluation methodology

Considering the variety in real-world network structures, we randomly generate networks without a predefined topology. Among these, two representative networks are chosen: a dense network where each pair of users are connected with multiple possible paths, and a comparatively sparse network where only a few possible paths exist between each pair of users. Next, we assign capacity and error rate to each edge in the two networks. Sum of capacity of all edges within each networks is identical, and within each network, each edge has the same capacity. Thus, each edge in the dense network has a lower capacity than each edge in the sparse network. Error rate of each edge in both networks is randomly assigned, but is mostly controlled under 5% to make surface codes function properly. In each experiment, communication requests are randomly generated between user pairs in both networks. We utilize a routing protocol in to schedule all requests into routing paths. For each communication of a distance d surface code, d^2 routing paths will be scheduled to transmit the d^2 data qubits, and the error rate of each routing path is determined by the routing protocol. To implement our proposed **disparity surface code**, we transmit all its core data qubits through the ones with the lowest error rates among all scheduled routing paths. We compare the performance of disparity surface codes against a baseline strategy denoted as **random assignment**, which randomly assigns data qubits to all scheduled routing paths.

Networks are generated using the NetworkX [56] library, and the routing protocol in [48] is utilized for scheduling communication requests. Quantum circuits for disparity surface codes are simulated using the Pymatching [49] library, and run locally on a classical machine.

B. Disparity surface code evaluation

From Fig. 12, we observe that in both dense and sparse networks, the disparity surface code significantly outperforms the random assignment strategy, achieving much lower logical error rates. In the dense network (a), both disparity surface codes and random assignment exhibit low logical error rates, with both demonstrating favorable scalability: larger code sizes result in lower logical error rates. However, in the sparse network (b), where path average fidelity is lower and varies more, random assignment loses the positive scalability, while the disparity surface code maintains it. Therefore, compared with basic surface codes, our disparity surface codes are more adapted to practical network scenarios.

Figures 12(c) and (d) further compare different versions of disparity surface codes: $disp\_xz$ that protects along both logical X and Z axes, $disp\_x$ that protects along logical X axis, and $disp\_z$ that protects along logical Z axis. In the tables, $disp\_xz : X$ refers to a $disp\_xz$ disparity surface code evaluated using logical X measurements, while $disp\_xz : Z$ refers to evaluations using logical Z measurements.

The results indicate that $disp\_x$ consistently performs best when evaluated with logical X measurements, and $disp\_z$ consistently performs best under logical Z measurements. In the meantime, $disp\_xz$, which aims to protect both axes simultaneously, delivers a more balanced but mediocre performance in both logical X and logical Z evaluations. An intuitive explanation for this observation is that: $disp\_x$ effectively focuses the surface code decoder on logical Z errors, while $disp\_z$ emphasizes logical X errors. Meanwhile, $disp\_xz$, however, distributes the decoder's emphasis between both error types, which weakens its focus on either. Developing an improved approach for achieving full protection along both axes in surface codes remains a potential direction for further exploration.

We also observe in Fig. 12(d) that in sparse networks, when evaluated with logical Z measurements, none of the three versions of disparity surface codes demonstrate positive scalability with respect to code distance. As will be discussed below in Section VIII-C, surface codes only achieve positive scalability when the error rates of their data qubits are kept below a certain threshold. The results here correspond to network scenarios where error rates across all routing paths are excessively high. While such scenarios are rarely considered in practice, they may arise occasionally due to insufficient optical fibers or repeaters implemented within quantum networks. We include these results here for completeness but consider them a potential direction for future exploration.

C. Decoder evaluation

We evaluate the performance of our Di-MWPM decoder against its baseline model [55], focusing on their respective Pauli-error thresholds on disparity surface codes. We generate disparity surface codes using the Pymatching [49] library, with depolarization errors randomly applied to individual data qubits. The depolarization channel used for testing includes the identity gate $\{I\}$ and Pauli errors $\{X, Y, Z\}$. For a given physical error rate p, each Pauli error has a probability of $p/3$, while the identity gate (indicating no error) occurs with a probability of $1-p$. To evaluate the decoders, the syndrome patterns of generated disparity surface codes are input into

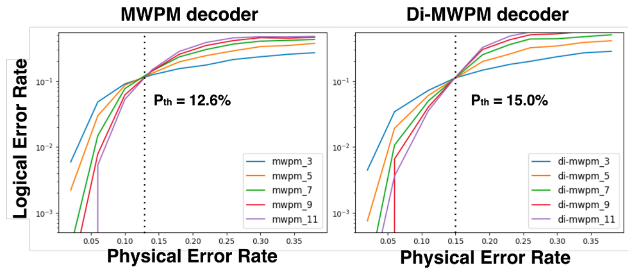


Fig. 13. Pauli error thresholds of disparity surface codes using a MWPM decoder (left) and our Di-MWPM decoder (right). Physical error rates on the x-axis represent the error rates of the remaining data qubits, while the error rates of the core data qubits are set at $1/10$ of that value.

each decoder. Their decoded results are compared with the actual observable of the disparity surface code prior to the introduction of random errors. A logical error is recorded if the decoded result does not match the original observable. In Fig. 13, for each distance-d and physical error rate p, 10,000 random disparity surface codes are generated for decoding. The logical error rate is calculated as the number of occurrence of logical errors divided by 10,000.

As shown in Fig. 13, our Di-MWPM decoder demonstrates significant improvements on Pauli-error thresholds, elucidating its effective utilization of the error rate disparity within disparity surface codes. Higher Pauli-error thresholds reduce the technical requirements for implementing larger disparity surface codes. As illustrated in Fig. 13, larger disparity surface codes exhibit lower logical error rates when the physical error rate is below the threshold. Conversely, when the physical error rate exceeds the threshold, smaller disparity surface codes tend to perform better. Thus, to optimize the scalability of disparity surface codes relative to code size, users should ensure that the physical error rate of individual data qubits remains below the error threshold. And the data qubits within the core must be maintained at an even lower error rate to fully exploit the advantages of the disparity surface code.

IX. Conclusion

In this paper, we present disparity surface codes, a group of geometric configurations of surface codes for optimized performance in practical quantum networks. Our results demonstrate that disparity surface codes can significantly reduce logical error rates by assigning their cores to low-error-rate routing paths. Maintaining the physical error rates for the core around 0.5% enables disparity surface codes to function properly even within quantum networks with poor quality of service. We determine that the minimal core size required for protection along either the logical X or Z axis corresponds to the code distance d. For simultaneous protection along both axes, a minimum of $2d-1$ data qubits is required. Furthermore, we present Di-MWPM, a specialized error correction decoder for disparity surface codes, which efficiently leverages the error rate disparity among data qubits. Our decoder achieves higher error thresholds compared to baseline models.

REFERENCES

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

[2] M. Tengner, *Photonic Qubits for Quantum Communication: Exploiting photon-pair correlations; from theory to applications*. PhD thesis, KTH, Stockholm, Sweden, 2008.

[3] Y. Mao, Y. Liu, X. Shang, and Y. Yang, "Network topology design for distributed quantum computing," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1213–1223, IEEE, 2024.

[4] Q. Zhu, Y. Zhao, H. Xu, L. Huang, and C. Qiao, "Eps placement and lightweight entanglement routing for quantum data networks," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1190–1201, IEEE, 2024.

[5] Y. Zeng, J. Zhang, X. Shang, J. Liu, Z. Liu, and Y. Yang, "Multi-user entanglement routing design over quantum internets," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pp. 266–276, IEEE, 2024.

[6] O. Amer, W. O. Krawec, M. Z. Hossain, V. U. Manfredi, and B. Wang, "Dynamic routing and post-processing strategies for hybrid quantum key distribution networks," in *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1224–1235, IEEE, 2024.

[7] H. Ushijima-Mwesigwa, C. F. Negre, and S. M. Mniszewski, "Graph partitioning using quantum annealing on the d-wave system," in *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, pp. 22–29, 2017.

[8] D. Herman, C. Googin, X. Liu, Y. Sun, A. Galda, I. Safro, M. Pistoia, and Y. Alexeev, "Quantum computing for finance," *Nature Reviews Physics*, vol. 5, no. 8, pp. 450–465, 2023.

[9] W. Jiang, J. Xiong, and Y. Shi, "A co-design framework of neural networks and quantum circuits towards quantum advantage," *Nature communications*, vol. 12, no. 1, p. 579, 2021.

[10] S. S. Gill, O. Cetinkaya, S. Marrone, D. Claudino, D. Haunschild, L. Schlote, H. Wu, C. Ottaviani, X. Liu, S. P. Machupalli, *et al.*, "Quantum computing: Vision and challenges," *arXiv preprint arXiv:2403.02240*, 2024.

[11] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, "Quantum internet: Networking challenges in distributed quantum computing," *IEEE Network*, vol. 34, no. 1, pp. 137–143, 2019.

[12] O. Bel and M. Kiran, "Simulators for quantum network modelling: A comprehensive review," *Computer Networks*, p. 111204, 2025.

[13] 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.

[14] 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.

[15] Google Quantum AI, "Suppressing quantum errors by scaling a surface code logical qubit," *Nature*, vol. 614, no. 7949, pp. 676–681, 2023.

[16] R. Acharya, L. Aghababaie-Beni, I. Aleiner, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, N. Astrakhantsev, J. Atalaya, *et al.*, "Quantum error correction below the surface code threshold," *arXiv preprint arXiv:2408.13687*, 2024.

[17] S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, p. eaam9288, oct 2018.

[18] 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.

[19] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[20] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[21] J.-M. Goethals and S. Snover, "Nearly perfect binary codes," *Discrete Mathematics*, vol. 3, no. 1-3, pp. 65–88, 1972.

[22] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Physical review A*, vol. 52, no. 4, p. R2493, 1995.

[23] D. Bacon, "Operator quantum error-correcting subsystems for self-correcting quantum memories," *Physical Review A*, vol. 73, no. 1, p. 012340, 2006.

[24] D. M. Debroy, L. Egan, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Cetina, C. Monroe, and K. R. Brown, "Optimizing stabilizer parities for improved logical qubit memories," *Physical Review Letters*, vol. 127, no. 24, p. 240501, 2021.

[25] A. Steane, "Multiple-particle interference and quantum error correction," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 452, no. 1954, pp. 2551–2577, 1996.

[26] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, "Fault-tolerant conversion between the steane and reed-muller quantum codes," *Physical review letters*, vol. 113, no. 8, p. 080501, 2014.

[27] H. Goto, "Minimizing resource overheads for fault-tolerant preparation of encoded states of the steane code," *Scientific reports*, vol. 6, no. 1, p. 19578, 2016.

[28] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, p. 1098, 1996.

[29] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE transactions on information theory*, vol. 44, no. 2, pp. 744–765, 1998.

[30] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information theory*, vol. 45, no. 5, pp. 1456–1467, 1999.

[31] D. J. MacKay, G. Mitchison, and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2315–2330, 2004.

[32] L. Golowich and V. Guruswami, "Quantum locally recoverable codes," *arXiv preprint arXiv:2311.08653*, 2023.

[33] D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *arXiv preprint arXiv:0801.1241*, 2008.

[34] M. Grassl and M. Rotteler, "Non-additive quantum codes from goethals and preparata codes," in *2008 IEEE Information Theory Workshop*, pp. 396–400, IEEE, 2008.

[35] E. Kubischta and I. Teixeira, "Family of quantum codes with exotic transversal gates," *Physical Review Letters*, vol. 131, no. 24, p. 240601, 2023.

[36] P.-N. Chen, H.-Y. Lin, and S. M. Moser, "Nonlinear codes outperform the best linear codes on the binary erasure channel," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 1751–1755, IEEE, 2015.

[37] V. V. Albert and P. Faist, "The error correction zoo," 2024. https://errorcorrectionzoo.org/ [Accessed: 2024].

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

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

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

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

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

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

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] T. Hu, J. Wu, and Q. Li, "SurfaceNet: Fault-tolerant quantum networks with surface codes," *IEEE Network*, 2023.

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

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

[51] 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.
[52] K.-Y. Kuo and C.-Y. Lai, "Exploiting degeneracy in belief propagation decoding of quantum codes," *npj Quantum Information*, vol. 8, no. 1, p. 111, 2022.
[53] M. Dinitz, S. Im, T. Lavastida, B. Moseley, and S. Vassilvitskii, "Faster matchings via learned duals," *Advances in neural information processing systems*, vol. 34, pp. 10393–10406, 2021.
[54] S. Gicev, L. C. Hollenberg, and M. Usman, "A scalable and fast artificial neural network syndrome decoder for surface codes," *Quantum*, vol. 7, p. 1058, 2023.
[55] V. Kolmogorov, "Blossom V: a new implementation of a minimum cost perfect matching algorithm," *Mathematical Programming Computation*, vol. 1, pp. 43–67, 2009.
[56] A. Hagberg, P. J. Swart, and D. A. Schult, "Exploring network structure, dynamics, and function using networkx," tech. rep., Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.