Decoding Topological Subsystem Color Codes Over the Erasure Channel Using Gauge Fixing

Hiteshvi Manish Solanki and Pradeep Kiran Sarvepalli

Abstract—Topological subsystem color codes (TSCCs) are an important class of topological subsystem codes that allow for syndrome measurement with only 2-body measurements. It is expected that such low complexity measurements can help in fault tolerance. While TSCCs have been studied over depolarizing noise model, their performance over the erasure channel has not been studied as much. Recently, we proposed erasure decoders for TSCCs and reported a threshold of 9.7%. In this paper, we continue our study of TSCCS over the erasure channel. We propose two erasure decoders for topological subsystem color codes. These decoders employ a mapping of the TSCCs to topological color codes (TCCs). In addition, these decoders use the technique of gauge fixing, where some of the gauge operators of the subsystem code are promoted to stabilizers. We perform gauge fixing using 4-body and 8-body gauge operators. With partial gauge fixing, we obtained a threshold of 17.7% on a TSCC derived from the square octagon lattice. Using an order maximal gauge fixing decoder we were able to improve the threshold to 44%. The performance of the order maximal gauge fixing decoder can be further improved to close to 50% in conjunction with an optimal erasure decoder for topological color codes. We also study the correctability of erasures on the subsystem codes.

Index Terms—Topological subsystem color codes, decoding, gauge fixing, topological codes, subsystem codes, quantum erasure channel.

I. INTRODUCTION

Subsystem codes are an important generalization of stabilizer codes [1], [2], [3]. One of the motivations for this generalization was to simplify the error recovery process of quantum codes. Specifically, they can simplify the syndrome computation which involves the measurement of the stabilizer generators. These codes are defined by a (non-Abelian) subgroup of the Pauli group called the gauge group. Of particular interest are the codes with local gauge group. Topological subsystem color codes are a class of subsystem codes where generators of the gauge group as well as the stabilizer of the code are local.

The advantages of topological subsystem codes (TSCs) motivated many researchers to look at their performance under different noise models. Most of the studies on the performance of TSCs [4], [5], [6], [7] focused on the depolarizing channel. The error correcting capabilities of TSCCs for handling leakage errors were explored in [8]. In contrast, the performance of TSCCs over the quantum erasure channel [9], which can be used to model qubit loss, has not been studied as much. Motivated by this gap, we began the study of topological subsystem color codes over the erasure channel and reported some preliminary results in [10]. Therein we proposed multiple two-stage erasure decoders for the TSCC derived from the square octagon lattice. Using a combination of various techniques like peeling and clustering, we were able to achieve a threshold of 9.7% in [10] for the TSCC derived from the square octagon lattice. However, this threshold was not optimal and there was room for improvement.

In this paper, we continue our study of TSCCs over the erasure channel and present new decoding algorithms that improve upon the decoders in [10]. A driving force for this work is to improve the threshold of the TSCCs for erasure noise. Topological subsystem color codes have fewer stabilizer generators than a comparable color code or a surface code. This will lead to lower thresholds [4], [5], [11]. To address this, we use the technique of gauge fixing, wherein we promote some of the gauge operators to checks. Gauge fixing in effect leads to a larger stabilizer. This technique of promoting some gauge operators to stabilizers was used for analyzing the structure of subsystem codes in [4], and for quantum computation in [12] and [13]. More recently, it was also employed for decoding subsystem codes [7].

An immediate question for the gauge fixing decoders is to determine which gauge operators to promote to stabilizers. In the context of decoding, we choose to promote gauge operators that are local. This will simplify the complexity of syndrome measurement and be more amenable to fault tolerant implementation. We propose two decoding algorithms each of which makes a different choice of the operators for gauge fixing. Both of them rely on the fact that a TSCC can be mapped to color codes. Furthermore, the stabilizers of these color codes are in the gauge group of the TSCC. Another question in the context of gauge fixing decoders is related to the sequence of measurement of the gauge operators. In our
case, all the gauge operators that we promote to stabilizers anticommute with some 2-body gauge operator of the TSCC. While we were able to obtain efficient decompositions of these additional stabilizers individually, these decompositions were not amenable to a joint measurement using 2-body measurements. Therefore our decoding algorithms require direct measurement of these stabilizers. Then employing the mapping of the TSCC to color codes, we proceed to decode the TSCC. We briefly summarize our main contributions below.

(i) First, we propose a decoder that uses partial gauge fixing for TSCC. We obtain a threshold of 17.7% for TSCC derived from the square octagon lattice.

(ii) We then propose an alternate "order" maximal gauge fixing decoder to improve the threshold. This decoder leads to a threshold of 44%. This improvement is attained at the expense of a slight increase in complexity. This threshold can be improved further by using optimal erasure decoders for the TCCs. In particular, in conjunction with the Gaussian elimination based erasure decoding of TCCs, we are able to obtain a threshold close to 50%.

(iii) We study the correctability of erasures on a subsystem code. Specifically, we provide a necessary and sufficient condition for an erasure pattern to be correctable on a subsystem code (without gauge fixing). We also study correctability of erasures under the order maximal gauge fixing decoder.

There are two related works in addition to our previous work [10]. In [8], the authors studied topological subsystem codes for correcting leakage errors, a more severe form of noise than erasure noise. While somewhat related, this noise model is different from ours and assumes that the locations of the erased qubits are unknown. With respect to the technique of gauge fixing, the closest work is that of [7] where the authors also use gauge fixing for decoding subsystem codes. They studied subsystem surface codes and their generalizations over hyperbolic surfaces under depolarizing noise and biased noise models.

We organize the paper as follows. In Section II we review the background for our proposed decoders. In Section III we give an overview of the proposed decoding algorithms. Section IV elaborates on syndrome measurement and preprocessing techniques. In Section V we discuss the first stage for $X$ error correction and in Section VI the second stage for $Z$ error correction. Then we study conditions for correctability of erasures on subsystem codes in Section VII. We report the simulation results in Section VIII and finally conclude with a brief summary in Section IX.

II. BACKGROUND

In this section, we briefly review some background material. We assume that the reader is familiar with stabilizer codes, see [14], [15] for an introduction.

A. Subsystem Codes

We briefly review subsystem codes. For more details, we refer the reader to [15]. Subsystem codes are obtained from stabilizer codes by not encoding information in some of the logical qubits. These qubits are called the gauge qubits. Any error on the gauge qubits does not affect the codespace. An $[[n, k, r, d]]$ subsystem code encodes $k$ logical qubits and $r$ gauge qubits into $n$ qubits. It can detect errors up to $d−1$ qubits, where $d$ is the distance of the code. The distance also signifies the smallest weight of non-trivial logical operator.

We define a subsystem code by a subgroup $\mathcal{G} \subseteq \mathcal{P}_n$, where $\mathcal{P}_n$ is the Pauli group on $n$ qubits. Elements of $\mathcal{G}$ are called the gauge operators. Elements which generate the gauge group are called the gauge generators. Recall that the centralizer of a subgroup $\mathcal{G} \subseteq \mathcal{P}_n$ is defined as

\[
\mathcal{C}(\mathcal{G}) = \{ g \in \mathcal{P}_n \mid gh = hg \text{ for all } h \in \mathcal{G} \}
\]

The stabilizer $S$ of the subsystem code is a subgroup of $\mathcal{G}$ such that $(\mathcal{I}, S) = \mathcal{G} \cap \mathcal{C}(\mathcal{G})$, where $\mathcal{C}(\mathcal{G})$ is the centralizer of $\mathcal{G}$. Elements of $S$ act trivially on the code space. It is convenient to ignore the phases and write the stabilizer up to a phase as $S = \mathcal{G} \cap \mathcal{C}(\mathcal{G})$. Since $S \subseteq \mathcal{G}$, it follows that $\mathcal{C}(\mathcal{G}) \subseteq \mathcal{C}(S)$.

For an $[[n, k, r, d]]$ subsystem code, ignoring phases, there are $2r + s$ independent generators for $\mathcal{G}$ and $s$ independent generators for $S$, where $n = k + r + s$, see [3]. Operators in $\mathcal{C}(\mathcal{G})$ are called bare logical operators while those in $\mathcal{C}(\mathcal{G}) \setminus S$ are nontrivial bare logical operators. Operators in $\mathcal{C}(S)$, obtained by appending gauge operators to the bare logical operators, are called dressed logical operators. The nontrivial dressed logical operators are in $\mathcal{C}(S) \setminus \mathcal{G}$ [5].

B. Topological Color Codes

Topological color codes (TCCs) in 2D are defined on lattices where the vertices are trivalent and faces are 3-colorable [16]. These lattices are also called 2-colexes. A quantum code can be defined on 2-colex by placing qubits on the vertices, and checks on faces. (We shall refer to a 2-colex as a color code for this reason). For any face $f$, a pair of stabilizers are defined as:

\[
B^X_f = \prod_{v \in f} X_v \quad \text{and} \quad B^Z_f = \prod_{v \in f} Z_v
\]

where $X$ and $Z$ are the Pauli matrices. Let $F_{\gamma}$ be set of faces of $\gamma$ color, $\gamma \in \{r, g, b\}$. These checks (stabilizers) satisfy the following relations:

\[
\prod_{f \in F_{\gamma}} B^X_f = \prod_{f \in F_{\gamma}} B^Z_f = \prod_{f \in F_{\gamma}} B^\sigma_f, \quad \text{for } \sigma \in \{X, Z\}
\]

C. Topological Subsystem Color Codes

Topological subsystem color codes (TSCCs) are a class of subsystem codes obtained from TCCs [17]. To construct a TSCC, we construct a hypergraph from the 2-colex using vertex expansion, refer [5], [17] for more details. Fig. 1 shows the constructed hypergraph on the right. This hypergraph has two kinds of edges: a rank-2 edge of the form $(u, v)$ and a rank-3 edge of the form $(u, v, w)$. As this hypergraph is 3-edge-colorable, in the figures, the rank-2 edges are either dashed or solid while the rank-3 edges are represented by
an operator $K$ face, of the same color. We denote these faces by (gray) triangles. Each color code face generates a hypergraph face, of the same color. We denote these faces by $F$. The $c$ colored faces are denoted $F_c$. Note that $F = F_c \cup F_d \cup F_b$.

We place qubits on the vertices of the hypergraph. We define an operator $K_{(u,v)}$ for any pair of vertices $(u,v)$ which are adjacent or which belong to the same hyperedge, as

$$K_{(u,v)} = \begin{cases} 
X_uX_v & (u,v) \text{ is dashed edge} \\
Y_uY_v & (u,v) \text{ is solid edge} \\
Z_uZ_v & (u,v) \text{ is in some hyperedge}
\end{cases} \tag{4}$$

We can then define the gauge group of the subsystem code as: $G = \langle K_{(u,v)} \rangle$ $(u,v)$ is a rank-2 edge or $u,v$ belong to the same hyperedge).

We also define edge operators for the edges of the hypergraph which are needed to define the stabilizers and logical operators. The edge operator for a rank-2 edge $e = (u,v)$ is $K_e = K_{(u,v)}$. Every rank-3 edge (hyperedge) $e = (u,v,w)$ is also assigned an operator $K_e = Z_uZ_vZ_w$. (Note that the operator $Z_uZ_vZ_w$ is not a gauge operator.) Each hyperedge $(u,v,w)$, gives rise to three $ZZ$ gauge operators, namely, $Z_uZ_vZ_w$, $Z_uZ_vZ_w$, and $Z_uZ_vZ_w$, one for each pair of vertices. Of these three $ZZ$ operators, only two are independent.

The stabilizers and the logical operators of the TSCC are completely determined by the gauge group. In case of TSCCs, they can be characterized in terms of cycles of the hypergraph and they can be defined in terms of the edge operators.

A cycle in a hypergraph is a collection of edges such that every vertex has an even degree with respect to these edges. A rank-2 cycle involves only rank-2 edges. A hypercycle involves both rank-2 edges and hyperedges (rank-3 edges). To every cycle $\sigma$ we can associate an operator as: $W(\sigma) = \prod_{e \in \sigma} K_e$. The importance of these operators is that they are precisely the operators in $C(G)$, the centralizer of $G$. A generating set for $C(G)$ can be given by considering all the cycles. Cycles of trivial homology$^2$ generate the stabilizer of the TSCC, as shown in Fig. 2. Cycles of nontrivial homology give rise to the logical operators of the TSCC.

For every face $f \in F$, we associate two independent cycles:

i) A rank-2 cycle denoted $\sigma^f_1$. This is formed by the rank-2 edges in the boundary of $f$.

ii) A hypercycle denoted $\sigma^f_2$. This cycle consists of an alternating set of inner rank-2 edges (only dashed edges) in the boundary of $f$, all the hyperedges incident on $f$, and outer rank-2 edges connecting these hyperedges.

(A dependent hypercycle can be generated by $\sigma^f_1$ and $\sigma^f_2$.) The hypercycle $\sigma^f_2$ is chosen so that the edges corresponding to the $XX$ type gauge operators in the boundary of $f$ are chosen to be in the hypercycle, see Fig. 2. We denote the stabilizer from rank-2 cycles as $W^f_1$ and the stabilizer from the hypercycle as $W^f_2$,

$$W^f_1 = \prod_{e \in \sigma^f_1} K_e = \prod_{v \in f} Z_v \quad W^f_2 = \prod_{e \in \sigma^f_2} K_e \tag{5}$$

where $K_e$ is the edge operator associated to the edge $e$. Fig. 2 shows these stabilizers: $W_1$ on faces $f_1$ and $f_2$ and $W_2$ on faces $f_3$ and $f_4$. The following dependencies exist among the stabilizer generators of the topological subsystem color codes.

$^1$Our notation is different from [5]. The operators in Eq. (4) are called link operators therein.

$^2$A hypercycle is of trivial homology if on contracting all the rank-3 edges and replacing all parallel edges by a single edge, we obtain a cycle of trivial homology. It is of nontrivial homology otherwise.
D. Mapping a TSCC Onto Color Codes

Reference [4] proposed a mapping of subsystem codes onto three copies of color codes. We illustrate this in Fig. 3, for the TSCC derived from the square octagon lattice. The faces of the TSCC are colored with three different colors. Qubits belonging to faces of the same color are grouped together and all of them are colored the same. Each of this stack (group) of qubits can be reinterpreted as a color code, see Fig. 3. This mapping could be viewed as a reinterpretation of the operators on the subsystem code in terms of these copies of color codes.

In mapping a TSCC to three copies of the parent TSCC, we extend the stabilizer of the TSCC to $S_c \subseteq G$, where $S_c$ is generated by all the stabilizers of the three copies of the parent TCC. The general relation between the parameters of the color code copies and TSCC is given in [18]. An $[[n, 4]]$ parent TCC on a torus gives rise to a TSCC with $[[3n, 2]]$ subsystem code with $r = 2n$ gauge qubits.

E. Error Model

We assume the erasures occur independently on each qubit with a probability $\varepsilon$ and each qubit is erased. Once the loss is detected, the erased qubit is replaced with a completely mixed state. For the purposes of decoding, we can measure the syndrome with the replaced qubit [19], [20], [21], [22]. Since the completely mixed state can be written as $\rho = \frac{1}{4} I + \frac{1}{4} X \rho X + \frac{1}{4} Y \rho Y + \frac{1}{4} Z \rho Z$, this is equivalent to placing one of the Pauli errors, $I, X, Y, Z$ with equal probability on the erased qubit. These errors are induced by the erasure, and we refer to them as erasure induced Pauli errors. Decoding over an erasure channel requires us to correct the erasure induced Pauli errors.

III. OVERVIEW OF THE GAUGE FIXING DECODERS

In this section we give an overview of proposed erasure decoders for TSCCs. As discussed in the previous section, for correcting the erasure errors, we can map them to Pauli errors on the erased qubits where each Pauli error occurs with equal probability. Although TSCCs are not CSS type codes, we can still correct these induced $X$ and $Z$ errors separately as in the case of the decoders proposed for the depolarizing channel [5], [6]. The proposed decoders have two stages: one for correcting $X$ errors and one for correcting $Z$ errors. We use the terms bit flip errors and $X$ error interchangeably. Similarly, for phase flip errors and $Z$ errors.

For the proposed decoders we map TSCC to three color code copies (refer Section II-D) and decode over the color codes instead of decoding on the TSCC directly.

While some of stabilizer generators of the color codes are also stabilizers of the TSCC, many of them are not. However, all the stabilizers of the color codes belong to the gauge group of the TSCC. In the proposed decoding algorithms we extend the stabilizer of the TSCC by augmenting it with some of the stabilizers of the color codes. We call this gauge fixing. (Note that the 2-body gauge generators are not treated as stabilizers.) We propose two decoders which differ in how the gauge fixing is implemented. The two decoders differ in how the phase flip errors are corrected.

For the first decoder, bit flip errors are corrected by mapping the TSCC onto three copies of color codes. This requires additional gauge operators corresponding to $Z$ type stabilizers of the color codes to be measured. Some of the $Z$ type stabilizers on the color codes are the rank-2 stabilizers $W^f_i$ of the TSCC. The $Z$ type stabilizers of the color codes are all promoted as stabilizers. However, the $X$ type stabilizers of the color codes are not promoted. These promoted stabilizers are used only for correcting the bit flip errors. The bit flip errors are corrected on the color codes and the estimate is lifted back to the TSCC. After the bit flip ($X$) errors are corrected, the phase flip ($Z$) errors are corrected. The $Z$ errors are corrected using the stabilizers of the subsystem code.
(hypercycle stabilizers) by mapping TSCC to the parent color code from which the subsystem color code was derived. The error estimate on the parent color code is then pulled back to the TSCC. We refer to this decoder as the partial gauge fixing decoder.

For the second decoder, both bit flip errors and phase flip errors are corrected by mapping the TSCC to copies of color codes. All the stabilizers of the color codes are promoted in this case. We use $X$ ($Z$) type stabilizers of the color codes for phase (bit) flip correction. The stabilizers of the subsystem code corresponding to the hypercycles are not used directly in this decoder. The total number of gauge operators that are promoted to stabilizers is almost close to the maximal set possible. We refer to this as order maximal gauge fixing decoder. After the errors are decoded on the color codes, the estimates are lifted back to the TSCC. Although the order maximal gauge fixing decoder has higher complexity, it leads to an improvement in performance as will be seen later in Section VIII.

In order to promote the color code stabilizers, we need to show that they are in the gauge group of the TSCC. This was already known from [4] but explicit details were not given. So, in the next section we give a simple proof of this fact by giving an explicit decomposition of the color stabilizers in terms of the 2-body gauge generators of the TSCC. These 2-body decompositions of the color code stabilizers may be of independent interest.

IV. SYNDROME MEASUREMENT AND PREPROCESSING

A popular approach for decoding topological codes is to map the original code into some other quantum code for which efficient decoders are known [4], [23], [24], [25], [26]. For our decoding algorithm, we use the map discussed in Sec. II-D. Instead of correcting errors on the TSCC, we correct the copies of color codes and then lift the error estimate to the TSCC. In Sec. IV-A we discuss the measurement of TCC and TSCC stabilizers. In Sec. IV-B we discuss the preprocessing techniques to improve the decoding performance.

A. Measurement of TCC and TSCC Stabilizers

We map the TSCC onto three (identical) copies of color codes. Our plan is to decode on the color codes instead of decoding on the TSCC. For this we need to measure the color code stabilizers on the TSCC.

One idea to reduce the complexity of the measurements of TCC stabilizers is to break them down to smaller body measurements (product of gauge generators). This is similar to how a TSCC stabilizer is measured using measurements of commuting gauge generators and then combining their outcome. Fig. 4a illustrates how the $Z$ type stabilizers on the red stack can be decomposed as product of gauge generators. We can obtain similar map for $X$ type stabilizers. Measurement of all the TCC stabilizers using only 2-body gauge generators is not possible since some of the TCC stabilizers anticommute with the gauge generators.

Therefore, instead of measuring the TCC stabilizers by classically combining the measurement outcomes of individual gauge generators, we measure 4-body or 8-body TCC stabilizer directly on TSCC. Fig. 4b shows how these measurements are done on the TSCC to obtain the $Z$ type TCC stabilizers on red stack. Similarly, we can directly measure both $X$ and $Y$ type stabilizers, for all three stacks, on the TSCC using 4 and 8-body measurements. Note that while measuring stabilizers for $c$ stack, where $c \in \{r, g, b\}$, only the qubits on $c$ faces of TSCC are involved.

Remark 2: Note that all the stabilizer generators of the color codes on each of the three stacks are generated as elements of the gauge group of the TSCC. The dependencies among the individual color codes are respected by this decomposition. So for instance on the color code on each stack the dependencies corresponding to Eq. (3) are also respected.

The above remark makes sure that we get the correct measurement outcome for all the stabilizers after the syndrome measurement process. The syndrome measurement is given in Algorithm 1.

Observe that every $W_1^f$ stabilizer corresponds to a color code stabilizer. In fact, $(B_2^f)_c = W_1^f$ where $f \in F_c$. On the other hand, none of the color code stabilizer directly corresponds to the hypercycle stabilizer $W_2^f$ of TSCC since $W_2^f$ have support on all three stacks. However, the hypercycle stabilizer $W_2^f$ can be decomposed into three stabilizers each supported on a stack of different color, see Fig. 5. The figure shows how $X$ type stabilizer on the red stack and $Y$ type stabilizers on the blue and green stacks can be combined to form the hypercycle stabilizer on green face of TSCC. Using these decompositions, we can also measure the syndromes of the TSCCs using the color code stabilizer outcomes as given in equations below.

\[
\begin{align}
W_2^r &= (B_2^X)_{f_b} (B_2^Y)_{f_g} (B_2^X)_{f_g} \\
W_2^g &= (B_2^X)_{f_r} (B_2^Y)_{f_b} (B_2^Y)_{f_g} \\
W_2^b &= (B_2^X)_{f_r} (B_2^X)_{f_g} (B_2^Y)_{f_g}
\end{align}
\]

B. Preprocessing the TSCC

In this section we state how the TSCC syndrome can be used to do preprocessing on the TSCC. We use two preprocessing
techniques, clustering and peeling for our algorithms. TSCC syndromes are used during the peeling stage. First, we apply the preprocessing techniques to remove some simple erasure patterns. We update the TCC syndromes when any correction is done during peeling. After peeling, we perform clustering of the remaining erasures. For more details, refer [10].

V. FIRST STAGE: X ERROR CORRECTION USING GAUGE FIXING

In this section we study the first stage-bit flip error correction. As shown in Fig. 3, we map the TSCC to three copies of color codes. Since there is one to one correspondence between the qubits of the color codes and the TSCC, we can directly map the erasures on TSCC onto the color codes. Next we obtain the X error syndrome, by measuring Z type stabilizers of the color codes as shown in Fig. 4b. Once the syndrome on each stack is obtained, we decode the color code on that stack using a color code erasure decoder and get an error estimate. We adapt the color code erasure decoder proposed in [24]. The final step is to lift the estimate from all the color codes to the TSCC. Since every qubit of the color code is a dedicated TSCC qubit (see Sec. II-D), we lift the error estimate at the same location on the TSCC. Note that since no two stacks share any qubit, all the three color codes can be decoded parallelly and independently.

Since the syndrome corresponding to the $W^f_2$ stabilizer was measured prior to the correction of the $X$ errors it needs to be updated following the correction. The correction of $X$ errors clears the syndrome corresponding to $B^f_Z$ stabilizers i.e., if we measure them now we obtain zero syndrome. Similarly the syndrome corresponding to $B^f_Y$ stabilizers also needs to be updated. As shown in Eq. 7, the hypercycle stabilizer directly depends on both $X$ and $Y$ type color code stabilizers. Hence, any modification in syndromes of these stabilizers also affects syndrome of the hypercycle.

The complete decoding procedure for the first stage is given in Algorithm 2 with $\sigma = X$. After the $X$ errors are corrected, the syndromes of the $Z$ type stabilizers of the color codes are cleared. Note that only a subset of these stabilizers are also stabilizers on the TSCC. (Recall that $(B^f_Z)_c = W^f_f$ where $f \in F_c$.) Therefore while correcting the bit flip errors additional TSCC gauge operators are fixed.

A. Residual Errors After Correcting Bit Flip Errors

As can be seen from Algorithm 2, we decode the erasure induced bit flip errors by means of a color code erasure decoder. In this paper we used the erasure color code decoder proposed in [24]. This decoder maps the erasures on the color code to erasures on two copies of surface code. In this process...
Algorithm 2 TSCC Erasure Decoder via Mapping to TCCs: Bit Flip (Phase Flip) Error Correction

Require: TSCC hypergraph \( H \), set of erasure positions \( \mathcal{E} \) and \( \sigma \) error syndrome \( s_\sigma^c \), \( c \in \{r,g,b\} \). (Note \( \sigma \) can be either \( X \) or \( Z \))

Ensure: \( \sigma \) error estimate \( \hat{E}_\sigma \)

1: Group the qubits according to color of faces on TSCC. (Initially, \( \hat{E}_\sigma = I \).)
2: for every \( c \)-stack, \( c \in \{r,g,b\} \) do
3: Let \( \hat{E}_c = I \) and map erasure locations directly as per the color.
4: Given \( s_\sigma^c \), decode color code using any color code erasure decoder to get estimate \( \hat{E}_c \).
5: Lift the estimate \( \hat{E}_c \) to the subsystem code and update \( W_2^c \) syndrome.
6: Update \( \hat{E}_\sigma = \hat{E}_\sigma \hat{E}_c \).
7: end for
8: Return \( \hat{E}_\sigma \) as the final \( \sigma \) error estimate.

it creates additional erasures on the surface codes. It decodes on the surface codes and lifts the estimate back to the color code. This estimate can have support on the unerased qubits of the color code and therefore the TSCC. When it succeeds the error estimate is up to a stabilizer. Therefore, any support on the nonerased qubits can be ignored, see [24] for more details. When the estimate is up to a stabilizer on the color code, on the TSCC the error is up to an \( X \) type gauge operator because \( X \) type stabilizers on the color code are gauge operators on the TSCC. When the estimate is up to a logical operator on the color code, on the TSCC the error can either be a logical operator or an \( X \) type gauge operator.

Suppose a part of the error pattern is as shown in Fig. 6a. Note that support of the error lies only on the red color code, see Fig. 6b. We measure the color code stabilizers directly on the TSCC to obtain the syndrome shown in Fig. 6b. With the syndrome and erased locations, we decode the color code using color code erasure decoder and obtain error estimate as shown in Fig. 6c. Note that the estimate is not the same as the original error, but it results in the same syndrome. When we apply this estimate to the original error on the color code, it results in a stabilizer as shown in Fig. 6d. (Therefore, the correction is up to a stabilizer.) When the estimate is lifted to the TSCC, we get a resultant \( X \) type gauge operator, shown in Fig. 6e. This example shows that the final outcome can induce an \( X \) type gauge error but not any additional \( Z \) errors on the unerased qubits. The following lemma proves this result formally.

Lemma 1: On performing \( X \) error correction on a TSCC by decoding on three copies of color codes through Algorithm 2, the lifted error estimate can either be an \( X \) type gauge operator or a logical operator (bare or dressed). If the estimate on the color codes is up to a stabilizer, then the lifted error estimate does not contain any \( Z \) error on the unerased qubits.

Proof: Suppose that the \( X \) part of the error on the TSCC is denoted \( E \) and its restriction to the color code on \( c \)-stack be \( E_c \). Let the error estimate returned by Algorithm 2 be \( \hat{E}_c \) on the color code on the \( c \)-stack. This estimate can be decomposed as \( \hat{E}_c = E_cL \), where \( L \) is a logical operator and \( S \) is a stabilizer of the color code. This estimate will be a pure \( X \) type operator on the color code as it is a CSS code. When \( \hat{E}_c \) is equal to \( E \), we get a perfect correction step and there is no residual error on the color code or the TSCC. If \( \hat{E}_c \) is not equal to \( E \), the estimate is up to a \( X \) stabilizer or a logical operator. When this estimate is lifted to the TSCC, it can either be a logical operator or a pure \( X \) type gauge operator. If the estimate is up to a (nontrivial) logical operator on the color codes, then we see that the lifted error estimate on the subsystem code, is an \( X \) type gauge operator or a dressed logical operator of the subsystem code. Recall that a dressed logical operator is a logical operator of the TSCC augmented with additional gauge generators. In case the estimate is up to a stabilizer on the color codes, then on the subsystem code, the estimate is up to an \( X \) type gauge operator that does not contain any \( Z \) errors on the unerased qubits. This is because the \( (X) \) stabilizers of the color codes on each stack are in the gauge group.

Lemma 1 shows that the lifted estimate on the TSCC does not result in any nonzero syndrome on the TSCC, therefore the second stage can also focus only on the erased qubits and ignore the unerased qubits.

Note that when the decoders on the color codes fail, it does not always lead to logical error on the TSCC as well. We illustrate this using the error pattern shown in Fig. 7. We correct the bit flip errors using Algorithm 2. For simplicity we choose an error pattern which is supported only on the red faces. Fig. 7 shows the error. Fig. 7b shows the error pattern and the corresponding syndrome on the red stack. Suppose the decoder results in the estimate shown in Fig. 7c. On correcting according to this estimate, the resultant operator is a bare logical operator on the color code as shown in Fig. 7d. When the estimate shown in Fig. 7c is lifted to the TSCC, we get an \( X \) type gauge operator as shown in Fig. 7e. Estimate on color code up to a logical error can result to an \( X \) type gauge operator on the TSCC or a dressed logical operator on TSCC. In either case it does not create any additional \( Z \) error syndrome on the TSCC.

VI. SECOND STAGE: Z ERROR CORRECTION

After performing \( X \) error correction, the next step is to correct the \( Z \) errors. We corrected the \( X \) errors with gauge fixing since we also fixed additional TCC stabilizers apart from TSCC stabilizers. In this section, we propose two algorithms to correct the \( Z \) errors. One of them uses only the stabilizers of the TSCC while the other uses the TCC \( X \) type stabilizers.

A. Correcting \( Z \) Errors Without Gauge Fixing

Recall that the TSCC is constructed from a 2-colex using vertex expansion. For \( Z \) error correction, we take the hypercycle syndromes of the TSCC and map them onto the parent 2-colex \( \Gamma \) from which the TSCC is derived. For face \( f \), the hypercycle syndrome \( W_2^f \) corresponds to the \( X \) type stabilizer syndrome of the corresponding face on the 2-colex \( \Gamma \). Since we have already cleared the \( X \) errors in the first stage, the residual
 syndrome on the lattice due to hypercycle stabilizers can be explained purely in terms of $Z$ errors alone. Next we also map the erasures from the TSCC to the parent 2-colex. A vertex of the 2-colex is erased if any one or more qubits of the hyper-edge (inflated triangle) in the TSCC are erased. The syndromes of the hypercycle stabilizer are projected to the 2-colex. (Note that the hypercycle stabilizer syndromes were updated after performing $X$ error correction.) Once the erased locations and syndromes are obtained on the underlying color code, we can adapt any color code erasure decoder to decode it.

After decoding, we lift the error estimate on the TSCC. A $Z$ error estimate on a color code qubit is lifted to $Z$ error on any one of the qubits on the corresponding hyperedge in the TSCC. The complete procedure for $Z$ error correction is given in Algorithm 3.

**Algorithm 3 Correcting $Z$ Errors Without Gauge Fixing**

**Require:** TSCC hypergraph $\mathcal{H}_T$, erasure positions $\mathcal{E}$ and syndrome of hypercycle stabilizers ($W^I_2$).

**Ensure:** $Z$ Error estimate $E^Z$

1. Project the updated $W^I_2$ syndromes (from Algorithm 2) to the underlying parent color code of the TSCC.
2. Map the erasure locations.
3. Adapt any color code erasure decoder to obtain an error estimate $E$ (on the color code).
4. Lift the error estimate $E$ to the TSCC. Denote the lifted estimate $E^Z$.
5. Return $E^Z$ as the error estimate for $Z$ errors.

We summarize the entire decoding algorithm. We use Algorithm 1 to compute the syndrome for decoding. We incorporate preprocessing techniques like peeling and clustering for better performance as in [10]. After performing the preprocessing steps, we decode the bit flip errors using Algorithm 2, for the first stage of decoding. In the next stage we decode the phase flip errors using Algorithm 3. The performance of this decoder is shown in Fig. 8a. We obtain a threshold of about $17.7\%$.

**Remark 3:** For correcting the $X$ errors we fix, in effect, $3|F| - 6$ independent commuting operators from the gauge group and $|F| - 2$ commuting operators for correcting the $Z$ errors. Thus a total of $4|F| - 8$ commuting gauge operators are fixed. The TSCC would have measured only $2|F| - 2$ operators. Thus by partial gauge fixing we are fixing an additional $2|F| - 6$ gauge operators as checks.

Recall, from Remark 1, that the gauge group of the TSCC is generated by $2r + s = 10|F| - 2$ operators where $s = 2|F| - 2$ is the number of independent stabilizer generators of the subsystem code and $r = 4|F|$ is the number of gauge qubits. The maximal commuting subgroup in the gauge group is of size $r + s = 6|F| - 2$. Since we fixed only $4|F| - 8$ gauge operators we see that this decoder is a partial gauge fixing algorithm.

Observe that in this decoder, the erasure induced $X$ errors are corrected using gauge fixing while the $Z$ errors are not. This naturally suggests the design of a decoder that fixes more gauge operators. We take this up in the next section.

**B. Correcting $Z$ Errors With Gauge Fixing**

In Section VI-A we discussed an algorithm which uses partial gauge fixing to decode TSCC. In partial gauge fixing we perform gauge fixing for clearing only bit flip errors. The phase flip errors are cleared by decoding the parent color code. The improvement in the threshold performance prompts us to fix additional gauge operators. We now explore
the correction of $Z$ errors by gauge fixing. In partial gauge fixing, we fixed $4|F|-8$ gauge operators out of the maximal commuting subgroup of $6|F|-2$. Hence there is a scope of fixing $2|F|+6$ additional gauge operators on TSCC. In this section we correct phase flip errors with gauge fixing, similar to bit flip error correction. We map the TSCC on to three copies of color codes and decode them for correcting phase flip errors.

We obtain syndromes for $X$ type stabilizers of the color codes using Algorithm 1. Once the color code syndromes are obtained, we decode the color codes by adapting a color code erasure decoder. We then lift the joint estimate on all the stacks back to the TSCC. We lift the error estimate back to the TSCC according to the color of the color code. Considering we clear both $X$ and $Z$ error on the copies of color codes, we perform gauge fixing for correcting both the errors. We use Algorithm 2, with $\sigma = Z$, to correct phase flip errors with gauge fixing.

Remark 4: We can see that in effect we are fixing $6|F|-12$ independent stabilizers. On TSCC it is possible to gauge fix at most $r+s=6|F|-2$ stabilizers. Compared to the TSCC where we measure $2|F|-2$ stabilizers we are measuring in addition $4|F|-10$ gauge operators. Thus we are order maximal with respect to the number of gauge operators that are being fixed.

From Eq. (7), we have that the hypercycle stabilizer of TSCC can be decomposed to a sum of color code stabilizers. Therefore, clearing the syndromes on the color codes also clears the syndrome on the TSCC. This makes sure that correction restores the state to the codespace, namely, the joint +1-eigenspace of the subsystem code stabilizers. (Since color code is a CSS code, we can perform $X$ error correction and $Z$ error correction in any order.)

C. Complete Algorithm

We now summarize $X$ and $Z$ correction for both approaches. The first step is to obtain the necessary syndromes for decoding the TSCC. Along with the syndromes of the color codes, we also compute the syndrome of the TSCC which helps in the preprocessing using peeling.

The second step is to preprocess using peeling, see refer Algorithm 2 in [10] for further details on peeling. During peeling we correct single isolated erasures using TSCC stabilizer measurements. Peeling results in removal of some erasures and also updates syndromes of some of the stabilizers of the TSCC and the color codes. Based on the location of the erasure where the correction is applied, we modify the color code syndromes as per the stack it belongs to. An erased qubit belonging to a face $f \in F_c$ of TSCC modifies only syndromes of color code on $c$-stack. We also update the hypercycle stabilizer syndromes. Every qubit participates in three hypercycles. Hence syndrome of these hypercycle stabilizers must be modified based on the error estimate. For simulation simplicity we re-compute hypercycle stabilizers after every correction step during peeling. They are used for phase flip error correction during partial gauge fixing decoding. We do need not to keep track of the hypercycle syndromes for the order maximal gauge fixing decoder since we do not utilize them for further error correction.

After peeling, some of the erasures are corrected and the syndromes appropriately modified. Peeling is followed by clustering of erasures. We scan the lattice from left to right and top to bottom and search for a stabilizer which contains at least one erased qubit. We group the erased qubits of the stabilizer to form a cluster. (This is true even if there is a single erasure in that stabilizer.) For each of the erased qubit we check if it is already part of a previously formed cluster. If so, we merge these clusters. We repeat this procedure for every stabilizer. At the end, erasures of two distinct clusters do not participate in a common stabilizer. For more details on clustering, see [10].

Once the clusters are formed, the next step is to decode every cluster independently. For every cluster, we perform bit flip error correction using Algorithm 1. We then correct the phase flip errors. If decoding with partial gauge fixing decoder, we use Algorithm 3 and for order maximal gauge fixing decoder, we use Algorithm 1. All error estimates for each of the clusters are combined and returned at the end of the algorithm. Note that for order maximal gauge fixing since both bit flip error syndrome and phase flip error syndrome do not affect each other, we can decode both these errors in parallel.

We obtained a threshold of 44% for the order maximal gauge fixing decoder, shown in Fig. 8b. Note that for both the decoding algorithms we are measuring all the color code stabilizers. The main difference between the two algorithms is that gauge fixing is used only for correcting bit flip errors in the first decoder whereas it is used for correcting both bit flip and phase flip errors for the second decoder. We discuss the simulation results in Section VIII. Before that, we study the correctability of erasure patterns on TSCCs in the next section.

VII. Correctability Condition on Erasure Pattern

In this section we propose correctability conditions for an erasure pattern on subsystem codes. Using the correctability condition we can test, prior to decoding, whether an erasure pattern can be corrected or not. In Theorem 1, we propose the condition for a general subsystem code.

Before we begin, we introduce a few notations necessary for this section. We denote the stabilizer matrix of the subsystem code by $H$ and gauge group by $G$, over $\mathbb{F}_2$. The stabilizer matrix $H$ involves all the (independent) stabilizer generators of the subsystem code while $G$ consists of the gauge generators. Let $E$ be the set of qubits which are erased on the TSCC. Let $H_E$ be a submatrix which is a restriction of $H$ to the subset of columns corresponding to locations of erased qubits. Similarly, we denote by $G_E$ and $E_E$, the submatrices of $G$ restricted to the qubits in $E$ and $\bar{E}$ respectively. The submatrix $H_E$ consists of the remaining columns of $H$. We denote the stabilizer matrix of the parent color code by $H_c$, where $c \in \{r, g, b\}$.

An erasure pattern $E$ is correctable if all the errors supported in $E$ are correctable and non-correctable otherwise. Equivalently, if $E$ supports a logical error, then it is not correctable and correctable otherwise.
We now propose a condition for correctable erasures applicable to general subsystem codes decoded without gauge fixing. This condition also accounts for the gauge group.

**Theorem 1 (Correctable Erasures on a Subsystem Code Without Gauge Fixing):** An erasure pattern \( \mathcal{E} \) on a TSCC is correctable if and only if \( 2|\mathcal{E}| = \text{rank}(H_E) + \text{rank}(G) - \text{rank}(G_E) \).

**Proof:** Up to a phase, there are \( 4|\mathcal{E}| \) Pauli errors that can be supported in \( \mathcal{E} \). There are \( 2^{\text{rank}(H_E)} \) syndromes possible. For each syndrome there exist \( 2^{2|\mathcal{E}| - \text{rank}(H_E)} \) error patterns. Consider the map \( \pi_E : G \to G_E \). The operators in the kernel of \( \pi_E \) are exactly the operators in \( G \) whose support is entirely in \( \mathcal{E} \). The dimension of \( \ker \pi_E \) is given by \( \dim \ker(\pi_E) = \text{rank}(G) - \text{rank}(G_E) \). All errors in \( \ker(\pi_E) \) have support only in \( \mathcal{E} \) and have zero syndrome. First observe that such errors cannot be more than the total number of errors supported in \( \mathcal{E} \) which have zero syndrome. Therefore, \( \text{rank}(G) - \text{rank}(G_E) \leq 2|\mathcal{E}| - \text{rank}(H_E) \). If errors in \( \ker(\pi_E) \) are fewer than the number of distinct errors with the same syndrome, then \( \mathcal{E} \) supports nontrivial logical error(s). Therefore, if \( \text{rank}(G) - \text{rank}(G_E) < 2|\mathcal{E}| - \text{rank}(H_E) \), then \( \mathcal{E} \) is not correctable. If \( 2|\mathcal{E}| - \text{rank}(H_E) = \text{rank}(G) - \text{rank}(G_E) \), all such errors are in the gauge group. So \( \mathcal{E} \) does not support a (nontrivial) logical operator and \( \mathcal{E} \) is a correctable erasure pattern.

We performed simulations based on the condition in Theorem 1. We generate an erasure pattern \( \mathcal{E} \) where qubit is erased with a probability \( \varepsilon \) on the subsystem code. As per the locations of \( \mathcal{E} \), we compute the rank of the matrices \( H_E \) and \( G_E \). Rank of \( G \) remains constant throughout the simulations since it is independent of \( \mathcal{E} \). If the condition in Theorem 1 gets violated, we flag it as an uncorrectable error. We repeated the experiment for 10000 trials for various \( \varepsilon \) for code distances \( d = 4, 8, 16 \). We obtained a threshold of approximately 16.5%. For space constraints, the figure is not shown. Interested readers can find the plot in the extended version [27].

When Theorem 6 is specialized to stabilizer codes, in other words \( G \) is abelian, we obtain the following corollary. This result was originally shown in [22] in a slightly different but equivalent form, see Eq. (4) therein.

**Corollary 1:** For stabilizer codes, an erasure pattern \( \mathcal{E} \) is correctable if and only if \( 2|\mathcal{E}| = \text{rank}(H) + \text{rank}(H_E) - \text{rank}(H_E) \).

**Proof:** Consider the map \( \pi_E : G \to G_E \). The operators in the kernel of \( \pi_E \) are exactly the operators in \( G \) whose support is entirely in \( \mathcal{E} \). The dimension of \( \ker \pi_E \) is given by \( \dim \ker(\pi_E) = \text{rank}(G) - \text{rank}(G_E) \). All errors in \( \ker(\pi_E) \) have support only in \( \mathcal{E} \) and have zero syndrome. First observe that such errors cannot be more than the total number of errors supported in \( \mathcal{E} \) which have zero syndrome. Therefore, \( \text{rank}(G) - \text{rank}(G_E) \leq 2|\mathcal{E}| - \text{rank}(H_E) \). If errors in \( \ker(\pi_E) \) are fewer than the number of distinct errors with the same syndrome, then \( \mathcal{E} \) supports nontrivial logical error(s). Therefore, if \( \text{rank}(G) - \text{rank}(G_E) < 2|\mathcal{E}| - \text{rank}(H_E) \), then \( \mathcal{E} \) is not correctable. If \( 2|\mathcal{E}| - \text{rank}(H_E) = \text{rank}(G) - \text{rank}(G_E) \), all such errors are in the gauge group. So \( \mathcal{E} \) does not support a (nontrivial) logical operator and \( \mathcal{E} \) is a correctable erasure pattern.

We performed simulations based on the condition in Theorem 1. We generate an erasure pattern \( \mathcal{E} \) where qubit is erased with a probability \( \varepsilon \) on the subsystem code. As per the locations of \( \mathcal{E} \), we compute the rank of the matrices \( H_E \) and \( G_E \). Rank of \( G \) remains constant throughout the simulations since it is independent of \( \mathcal{E} \). If the condition in Theorem 1 gets violated, we flag it as an uncorrectable error. We repeated the experiment for 10000 trials for various \( \varepsilon \) for code distances \( d = 4, 8, 16 \). We obtained a threshold of approximately 16.5%. For space constraints, the figure is not shown. Interested readers can find the plot in the extended version [27].

When Theorem 6 is specialized to stabilizer codes, in other words \( G \) is abelian, we obtain the following corollary. This result was originally shown in [22] in a slightly different but equivalent form, see Eq. (4) therein.3

**Corollary 1:** For stabilizer codes, an erasure pattern \( \mathcal{E} \) is correctable if and only if \( 2|\mathcal{E}| = \text{rank}(H) + \text{rank}(H_E) - \text{rank}(H_E) \).

Corollary 1 immediately gives a sufficient condition for a correctable erasure pattern decoded via order maximal gauge fixing decoder. Specifically, if the condition is satisfied on all the three copies of color code i.e., \( 2|\mathcal{E}| = \text{rank}(H_E) + \text{rank}(H_E) - \text{rank}(H_E) \), for every \( c \), where \( c \in \{r, g, b\} \), then the erasure is correctable by the order maximal gauge fixing decoder. Note that this is only a sufficient condition for the correctness of an erasure pattern and not a necessary condition, see Fig. 7 and the related discussion in Section V-A. The reason this is not a necessary condition is that even if two or more color code decoders estimate incorrectly, when lifted the joint estimate might be correct.

**VIII. SIMULATION RESULTS**

In this section we present the simulation results for both the decoding algorithms. An erasure pattern is generated on

---

3The condition in Corollary 1 can also be modified as \( 2|\mathcal{E}| \leq \text{rank}(H) + \text{rank}(H_E) - \text{rank}(H_E) \).
the subsystem codes according to the probability of erasure \(\varepsilon\). Noise on each qubit is assumed to be identical and independent. Every erased location can undergo any of the Pauli errors with equal probability as described in Section II-E. Then we correct these induced Pauli errors on the subsystem code. In practice, while decoding a subsystem code, its stabilizers are measured indirectly via measuring the gauge generators and classically combining the outcomes appropriately. The proposed decoders map the TSCCs to multiple copies of color codes in order to decode them. We measure the multi-body color code stabilizers directly on the TSCC. Once the color code stabilizers are measured, we decode the three copies of color code using the color code erasure decoder [24]. This decoder only needs the erasure locations as input and generates its own syndrome. We adapted it so that it takes as input the previously measured syndrome and the erasure locations. After decoding the color code copies, we lift the estimate to TSCC. After lifting we check if any logical error has occurred or not. If the product of original error and the error estimate anticommutes with any one of the bare logical operators, we declare a logical error.

We present the simulation results for the TSCC derived from square octagon lattice. The TSCCs derived from the square octagon lattices have the code parameters \([3d^2, 2, 2d^2, d]\), where \(d\) is the distance of the code [17], [18]. We have simulated the proposed decoders for these codes for various erasure probabilities and lattice sizes. The plots shown in Fig. 8a and Fig. 8b show the variation of logical error rate with respect to the probability of erasure errors. A logical error occurs when the error and error estimate differ by a nontrivial logical operator of the TSCC. Every data point has been obtained by accumulating 2000 logical errors or 10000 runs, whichever occurs earlier.

Fig. 8a shows the plot for performance of the partial gauge fixing decoder. We report a threshold of approximately 17.7\%. This suggests that gauge fixing improves the performance in comparison to the highest threshold of 9.7\% given in [10]. When extending to almost full gauge fixing, we observe a threshold of 44\% as shown in Fig. 8b. The performance of our decoder is close to 50\% which is set by the no-cloning theorem. Using color code decoders proposed in [28] and [29] we can potentially improve the performance of our decoding algorithms as they have slightly higher thresholds. Using a maximum-likelihood erasure decoding of the TCCs, the threshold approaches 50\%, as shown in Fig. 9. Here, the maximum-likelihood decoding is implemented by Gaussian elimination.\(^4\) This led an increase in complexity compared to the TCC decoder used from [24]. However, efficient, low-complexity implementations of the maximum-likelihood decoding for the TCCs exist [29].

IX. CONCLUSION

In this paper, we proposed two algorithms for decoding subsystem codes over erasure channel. By using the technique of gauge fixing in combination with other preprocessing techniques we were able to significantly improve the threshold of TSCCs for the erasure noise with respect to our previous work [10]. Our work draws upon the mapping of TSCC to multiple copies of color codes proposed in [4]. We decode on these color codes instead of decoding directly on TSCC which motivates in a sense the need for gauge fixing. Our first decoder uses partial gauge fixing where gauge fixing is used only to correct erasure induced bit flip errors. The second decoder uses maximal gauge fixing where both bit flip and phase flip errors are corrected via gauge fixing. The later decoder gives us a threshold of 44\%. To the best of our knowledge, this is the highest threshold to date for a TSCC for erasure noise. Syndrome measurement for both the decoders requires us to measure 4-body and 8-body measurements. We also formulated conditions for correctability of erasures on the subsystem codes. There remain many other interesting open questions in this context. Developing an optimal decoder without gauge fixing, improving the performance of the proposed decoders, reducing the complexity of syndrome measurement are some natural problems for further study.

ACKNOWLEDGMENT

Hiteshvi Manish Solanki would like to thank Arun B. Aloshious for valuable discussions. The authors would like to thank Oscar Higgott for pointing out an error in a previous version. They would like to thank Nicolas Delfosse for clarification on [22] and also like to thank the anonymous reviewers for their comments to improve the presentation.

REFERENCES

[1] D. Bacon, “Operator quantum error-correcting subsystems for self-correcting quantum memories,” Phys. Rev. A, Gen. Phys., vol. 73, no. 1, Jan. 2006, Art. no. 012340.

[2] D. Kribs, R. Laflamme, and D. Poulin, “Unified and generalized approach to quantum error correction,” Phys. Rev. Lett., vol. 94, no. 18, May 2005, Art. no. 180501.

[3] D. Poulin, “Stabilizer formalism for operator quantum error correction,” Phys. Rev. Lett., vol. 95, no. 23, Dec. 2005, Art. no. 230504.

[4] H. Bombin, G. Duclos-Cianci, and D. Poulin, “Universal topological phase of two-dimensional stabilizer codes,” New J. Phys., vol. 14, no. 7, Jul. 2012, Art. no. 073048.

[5] M. Suchara, S. Bravyi, and B. Terhal, “Constructions and noise threshold of topological subsystem codes,” J. Phys. A, Math. Theor., vol. 44, no. 15, Apr. 2011, Art. no. 155301.

[6] V. V. Gayatri and P. K. Sarvepalli, “Decoding topological subsystem color codes and generalized subsystem surface codes,” in Proc. IEEE Inf. Theory Workshop (ITW), Nov. 2018, pp. 1–5.

[7] O. Higgott and N. P. Breuckmann, “Subsystem codes with high thresholds by gauge fixing and reduced qubit overhead,” Phys. Rev. X, vol. 11, no. 3, Aug. 2021, Art. no. 031039.

[8] N. C. Brown, M. Newman, and K. R. Brown, “Handling leakage with subsystem codes,” New J. Phys., vol. 21, no. 7, Jul. 2019, Art. no. 073055.

[9] M. Grassl, T. Beth, and T. Pellizzari, “Codes for the quantum erasure channel,” Phys. Rev. A, Gen. Phys., vol. 56, no. 1, pp. 33–38, Jul. 1997.

[10] H. M. Solanki and P. K. Sarvepalli, “Correcting erasures with topological subsystem color codes,” in Proc. IEEE Inf. Theory Workshop (ITW), Apr. 2021, pp. 1–5.

[11] R. S. Andrist, H. Bombin, H. G. Katzgraber, and M. A. Martin-Delgado, “Optimal error correction in topological subsystem codes,” Phys. Rev. A, Gen. Phys., vol. 85, no. 5, May 2012, Art. no. 050302.

[12] A. Paetznick and B. W. Reichardt, “Universal fault-tolerant quantum computation with only transversal gates and error correction,” Phys. Rev. Lett., vol. 111, no. 9, Aug. 2013, Art. no. 090505.

[13] H. Bombin, “Gauge color codes: Optimal transversal gates and gauge fixing in topological stabilizer codes,” New J. Phys., vol. 17, no. 8, Aug. 2015, Art. no. 083002.
[14] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[15] Daniel A. Lidar and Todd A. Brun, *Quantum Error Correction*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[16] H. Bombin and M. A. Martin-Delgado, “Topological quantum distillation,” *Phys. Rev. Lett.*, vol. 97, no. 18, Oct. 2006, Art. no. 180501.

[17] H. Bombin, “Topological subsystem codes,” *Phys. Rev. A, Gen. Phys.*, vol. 81, no. 3, Mar. 2010, Art. no. 032301.

[18] P. Sarvepalli and K. R. Brown, “Topological subsystem codes from graphs and hypergraphs,” *Phys. Rev. A, Gen. Phys.*, vol. 86, no. 4, Oct. 2012, Art. no. 042336.

[19] A. J. Moncy and P. K. Sarvepalli, “Performance of nonbinary cubic codes,” in *Proc. Int. Symp. Inf. Theory Appl. (ISITA)*, Oct. 2018, pp. 334–338.

[20] A. Kulkarni and P. K. Sarvepalli, “Decoding the three-dimensional toric codes and welded codes on cubic lattices,” *Phys. Rev. A, Gen. Phys.*, vol. 100, no. 1, Jul. 2019, Art. no. 012311.

[21] N. Delfosse and G. Zémor, “Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel,” *Phys. Rev. Res.*, vol. 2, no. 3, Jul. 2020, Art. no. 033042.

[22] N. Delfosse and G. Zémor, “Upper bounds on the rate of low density stabilizer codes for the quantum erasure channel,” 2012, arXiv:1205.7036.

[23] H. P. Nautrup, N. Friis, and H. J. Briegel, “Fault-tolerant interface between quantum memories and quantum processors,” *Nature Commun.*, vol. 8, no. 1, pp. 1–8, Nov. 2017.

[24] A. B. Aloshious and P. K. Sarvepalli, “Erasure decoding of two-dimensional color codes,” *Phys. Rev. A, Gen. Phys.*, vol. 100, no. 4, Oct. 2019, Art. no. 042312.

[25] J. Haah, “Algebraic methods for quantum codes on lattices,” *Rev. Colomb. de Mat.*, vol. 50, no. 2, pp. 299–349, 2016.

[26] N. Delfosse, “Decoding color codes by projection onto surface codes,” *Phys. Rev. A, Gen. Phys.*, vol. 89, no. 1, Jan. 2014, Art. no. 012317.

[27] H. M. Solanki and P. K. Sarvepalli, “Decoding topological subsystem color codes over the erasure channel using gauge fixing,” 2021, arXiv:2111.14594.

[28] D. Vodola, D. Amaro, M. A. Martin-Delgado, and M. Müller, “Twins percolation for qubit losses in topological color codes,” *Phys. Rev. Lett.*, vol. 121, no. 6, Aug. 2018, Art. no. 060501.

[29] S. Lee, M. Mhalla, and V. Savin, “Trimming decoding of color codes over the quantum erasure channel,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1886–1890.

Hiteshvi Manish Solanki received the B.Tech. degree in electronics and communication from DDU, Nadiad, and the M.S. degree in electrical engineering from IIT Madras. She is currently a Wireless Systems Engineer with Qualcomm India Pvt. Ltd., Bengaluru. Her research interests include quantum error correction and wireless communications.

Pradeep Kiran Sarvepalli received the B.Tech. degree in electrical engineering from IIT Madras, and the master’s degree in electrical engineering and the Ph.D. degree in computer science from Texas A&M University. After graduating from IIT Madras, he worked as an IC Design Engineer at Texas Instruments India, Bengaluru. He was a Post-Doctoral Fellow with the University of British Columbia and the Georgia Institute of Technology. He is currently an Associate Professor with the Department of Electrical Engineering, IIT Madras. His research interests include quantum and classical error correcting codes, quantum cryptography, quantum computation, and coding for distributed storage.