Single-shot preparation of hypergraph product codes via dimension jump

Yifan Hong

Joint Quantum Institute & Joint Center for Quantum Information and Computer Science, NIST/University of Maryland, College Park, MD 20742, USA

Quantum error correction is a fundamental primitive of fault-tolerant quantum computing. But in order for error correction to proceed, one must first prepare the codespace of the underlying error-correcting code. A popular method for encoding quantum low-density parity-check codes is transversal initialization, where one begins in a product state and measures a set of stabilizer generators. In the presence of measurement errors however, this procedure is generically not fault-tolerant, and so one typically needs to repeat the measurements many times, resulting in a deep initialization circuit. We present a protocol that prepares the codespace of constant-rate hypergraph product codes in constant depth with $O(\sqrt{n})$ spatial overhead, and we show that the protocol is robust even in the presence of measurement errors. Our construction is inspired by dimension-jumping in topological codes and leverages two properties that arise from the homological product of codes. We provide some improvements to lower the spatial overhead and discuss applications to fault-tolerant architectures.

1 Introduction

The reliability of large-scale quantum algorithms hedges on the creation and operation of fault-tolerant logical qubits. At the heart of a fault-tolerant architecture lies a quantum error-correcting code, whose role is to reverse the unwanted effect of noise introduced by the environment.

There has been immense development in fault-tolerant architectures using 2D topological codes, both on the theoretical [1–8] and experimental fronts [9–12]. Equipped with competitive thresholds, fault-tolerant gadgets and local implementation in planar geometries, 2D topological codes are the leading candidates for fault tolerance in near-term platforms. However, geometric locality in two spatial dimensions places fundamental limits on the physical overhead [13, 14] and capability [15] of these codes. By relaxing geometrical constraints, such as going to higher dimensions, one is able to construct codes that support more complex gadgets such as transversal non-Clifford gates [16, 17] and single-shot decoding [18–20]. Single-shot codes are able to suppress errors with just one round of noisy syndrome measurement and are thus attractive from the perspective of decoding complexity and computational clock speeds.

Quantum low-density parity-check (LDPC) codes are a broad class of quantum stabilizer codes [21] whose connectivity is few-body: every qubit is involved in a constant number of parity checks, and every parity check acts on a constant number of qubits [22]. This class encompasses the topological codes mentioned above but also includes families of codes with asymptotically optimal parameters such as constant rate [23, 24] and/or linear distance [25–27]. These more exotic code families cannot be embedded in any finite spatial dimension without long-range interactions [28–30]. Despite their geometrical nonlocality, certain LDPC codes show feasible promise in platforms capable of long-range connectivity such as those based on trapped ions [31–33] and neutral atoms [11, 34–36], with recent experimental demonstrations showcasing their performance [37–39]. In this work, we will focus on one particular code family called hypergraph product (HGP) codes [24]. Various HGP code instances can achieve constant rate (but sublinear distance), single-shot decoding [40], and passive error correction [41]. HGP codes also possess distance-preserving stabilizer measurements [42] as well as an arsenal of fault-tolerant gadgets [43–46] that can enable fault-tolerant quantum computation with constant overhead [40, 47, 48].

Although constant-rate HGP codes support single-shot and passive error correction, these features crucially rely on the assumption that the codespace has already been prepared. The standard procedure for preparing these HGP codes from an unentangled product state is to perform $\Theta(d) = \Theta(\sqrt{n})$ rounds of repeated syndrome measurement in order to reliably obtain an initial syndrome baseline or Pauli frame [3]. Since HGP codes can be decoded in constant depth after initialization using single-shot decoding, this initial $\Theta(d)$ depth presents a temporal bottleneck in their implementation.

In this work, we introduce a single-shot protocol that prepares the codespace of constant-rate HGP

Yifan Hong: yhong137@umd.edu

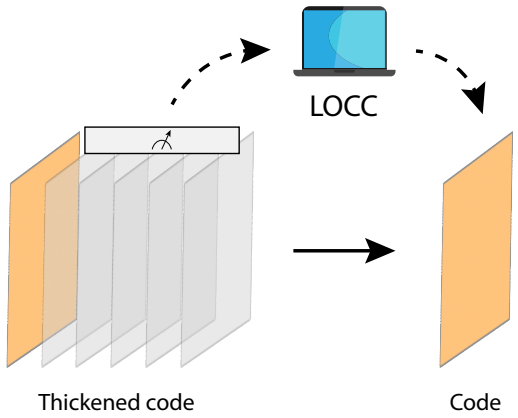


Figure 1: A high-level schematic of the state-preparation protocol is illustrated. The protocol involves fault-tolerant code-switching between the target HGP code and a "thickened" code with enhanced error-correcting properties. Local operations and classical communication (LOCC) in the form of measurements and feedback will be used to correct errors.

codes in $O(1)$ depth and $O(d)$ spatial overhead; see Fig. 1 for a schematic. Our construction generalizes dimension-jumping [49], a specific form of code-switching, from topological codes to more general quantum LDPC codes. 2D $\leftrightarrow$ 3D dimension-jumping in the surface code has been previously studied as a way to perform non-Clifford gates [50, 51] and generate long-range localizable entanglement [52, 53], but doing so in a single shot sacrifices performance [54]. In contrast, since HGP codes support single-shot error correction, we can easily show that single-shot dimension-jumping is robust, and we can furthermore use existing single-shot HGP decoders [40, 55] to correct errors during the process.

The paper is organized as follows. In Section 2, we review some preliminary mathematical background regarding HGP codes and the necessary algebraic machinery needed to describe our protocol. In Section 3, we introduce the state-preparation protocol, provide a decoding algorithm, and analyze its fault tolerance against adversarial noise. In Section 4, we discuss generalizations and introduce a modification that reduces the spatial overhead. In Section 5, we conduct numerical simulations of the protocol under an independent noise model. In Section 6, we provide some concluding remarks and discuss applications to fault-tolerant architectures based on HGP codes. The appendices provide formal statements and proofs.

2 Background

2.1 Hypergraph product codes

In this section, we briefly review the construction of quantum codes from classical codes via the hypergraph product and introduce some necessary machinery to describe the state preparation protocol.

Given n bits, each taking on a value of 0 or 1, the set of all 2^n possible bitstrings forms an n-dimensional binary vector space $\mathbb{F}_2^n$, where addition is performed modulo 2. A classical $[n, k, d]$ linear code encoding k logical bits is formally a k-dimensional subspace of logical codewords within this n-dimensional space. We can choose k basis vectors in this subspace to define a $k \times n$ generator matrix $G \in \mathbb{F}_2^{k\times n}$, which is a compact representation of the code. The code distance d is defined as the minimum number of bit flips to transition between any two codewords. Since in a linear code, the difference between two codewords is also a codeword, the code distance is simply the minimum Hamming weight amongst all $2^k - 1$ nonzero codewords. It is often useful to alternatively present this linear code using an $m \times n$ parity-check matrix $H \in \mathbb{F}_2^{m\times n}$ which satisfies $HG^\mathsf{T} = 0$ and so $\mathrm{rs}(G) \subseteq \ker(H)$; in other words, H encodes parity constraints that every codeword must satisfy. Note that there exist many different presentations of H, similar to our freedom in choosing basis vectors for G. We say that H is $(\Delta_\mathrm{c}, \Delta_\mathrm{r})$-LDPC if the nonzero weight of every column and row is bounded by constants $\Delta_\mathrm{c}, \Delta_\mathrm{r} = O(1)$ respectively [56]. From an algebraic perspective, a classical linear code can be described by the 2-term chain, or 1-complex

$$B \xrightarrow{H} S\,, \tag{1}$$

where B is the space of physical bitstrings or errors, and S is the space of violated parity checks or error syndromes [57].

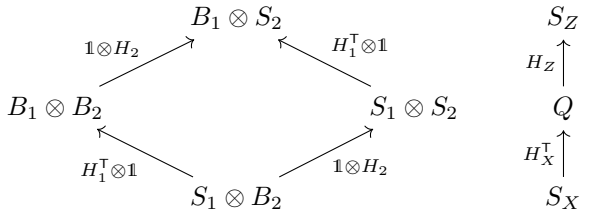
Similar to the classical story, a quantum $[\![n, k, d]\!]$ stabilizer code is a 2^k-dimensional subspace of the 2^n-dimensional Hilbert space on n qubits [21]. In analogy to the classical parity constraints, the quantum codespace is defined as the simultaneous $+1$ eigenspace of a set of commuting Pauli strings (i.e. tensor products of I, X, Y, Z), which generates the code's stabilizer group. The code distance is similarly defined as the minimum number of local operators needed to transition between different codewords. A Calderbank-Shor-Steane (CSS) code is a special type of stabilizer code whose Pauli checks are purely X-type or Z-type [58, 59]. As such, they can be packaged into two binary parity-check matrices H_X and H_Z, where a nonzero entry in a row of H_X (H_Z) denotes the location of a Pauli X (Z) operator. The commutativity of the Pauli checks then translates into the orthogonality condition $H_X H_Z^\mathsf{T} = 0$ between the two parity-check matrices. We say a CSS code is $(\Delta_\mathrm{c}, \Delta_\mathrm{r})$-LDPC if both X-type and Z-type parity-check matrices are individually $(\Delta_\mathrm{c}, \Delta_\mathrm{r})$-LDPC. From an algebraic standpoint, a CSS code can be described by the 3-term chain

$$S_X \xrightarrow{H_X^\mathsf{T}} Q \xrightarrow{H_Z} S_Z\,, \tag{2}$$

where S_X (S_Z) is the space of error syndromes according to H_X (H_Z), Q is the space of physical qubit

errors, and the orthogonality condition is explicitly expressed in terms of the composition of consecutive maps being zero in a chain complex. Logical $\bar{X}$ operators are classified according to the first homology group $\mathcal{H}_1 \equiv \ker H_Z / \operatorname{im} H_X^\mathsf{T}$, which consists of Pauli X strings that satisfy all Z-checks modded out by the X-type stabilizer subgroup. Logical $\bar{Z}$ operators are classified according to the first cohomology group $\mathcal{H}^1 \equiv \ker H_X / \operatorname{im} H_Z^\mathsf{T}$.

Given two classical linear codes with parity-check matrices H_1 and H_2, we can take a (homological) tensor product [60, 61] of their associated 2-term chains (1) to obtain the 2-dimensional complex



(3)

which we can then "collapse" into a one-dimensional 3-term chain by combining the spaces $B_1 \otimes B_2$ and $S_1 \otimes S_2$. The HGP code is then defined as the quantum CSS code associated with this 3-term chain [24]. Explicitly, the CSS parity-check matrices take the form

$$H_X = \left(H_1 \otimes \mathbb{1}_{n_2} \,\middle|\, \mathbb{1}_{m_1} \otimes H_2^\mathsf{T} \right) \tag{4a}$$

$$H_Z = \left(\mathbb{1}_{n_1} \otimes H_2 \,\middle|\, H_1^\mathsf{T} \otimes \mathbb{1}_{m_2} \right), \tag{4b}$$

and we can verify that $H_X H_Z^\mathsf{T} = 2(H_1 \otimes H_2^\mathsf{T}) = 0$ over $\mathbb{F}_2$. Geometrically, the tensor products in (4) bestow a certain "two-dimensional" structure to the HGP code, where the connectivity of one classical code is "copied" along one dimension and the other along the second dimension; see Fig. 2 for an illustration. This structure makes HGP codes especially appealing for hardware which can connect distant rows and columns of qubits in parallel such as the neutral-atom platform [48]. Denoting transposed code parameters with the transpose symbol, the $[\![n, k, d]\!]$ parameters of the HGP code are given by

$$n = n_1 n_2 + m_1 m_2 \tag{5a}$$

$$k = k_1 k_2 + k_1^\mathsf{T} k_2^\mathsf{T} \tag{5b}$$

$$d = \min\left(d_1, d_2, d_1^\mathsf{T}, d_2^\mathsf{T}\right), \tag{5c}$$

which can be obtained from properties of the homological tensor product. Many properties of a hypergraph product are inherited from its input classical codes. For example, if the input codes are LDPC, then so is the HGP code. If the input codes have constant rate (i.e. $k = \Theta(n)$), then the HGP code will as well. However, the code distance can only achieve an asymptotic scaling of $d = O(\sqrt{n})$.

Since the hypergraph product works for any two classical codes, optimizing the quantum code parameters reduces to a problem of optimizing the classical code parameters. Currently, the best classical code parameters are achieved by probabilistic methods. For instance, random classical LDPC codes are known to approach the Gilbert–Varshamov bound, a general rate-distance tradeoff, for linear codes [62].

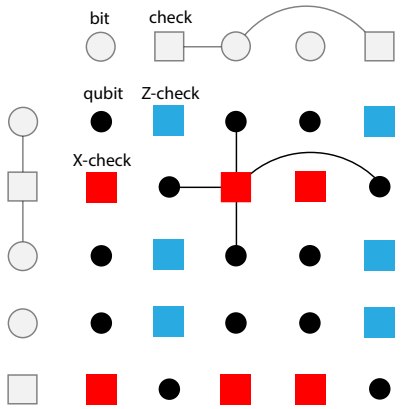


Figure 2: The hypergraph product of two classical linear codes is depicted in a 2D layout. Gray circles and squares denote classical bits and parity checks respectively. Solid black circles, red squares, and blue squares denote qubits, X-checks, and Z-checks respectively. The connectivity of the resulting HGP code is inherited from its classical codes, as shown for a subset of edges.

2.2 Single-shot error correction

For stabilizer codes, error correction typically proceeds by (i) measuring the set of Pauli check operators to obtain the error syndrome and then (ii) decoding the error syndrome to obtain a correction operator to be applied back to the code. The measurement step will collapse any arbitrary error into a Pauli string, a phenomenon known as error digitization, and so it suffices to focus on only decoding Pauli errors [63]. For CSS codes, the analysis further factorizes into independently correcting X-type and Z-type errors. Since our focus is on HGP codes, which are CSS codes, we will hence drop the X/Z subscript for brevity unless otherwise noted.

Given a physical error $\mathbf{e} \in Q$, its error syndrome is defined as $\mathbf{s}(\mathbf{e}) \equiv H\mathbf{e} \in S$, where H is either H_X or H_Z. The primary objective of (maximum likelihood) decoding can be summarized as: given a syndrome $\mathbf{s}$, compute the most probable error $\hat{\mathbf{e}}$. Note that there can be many different errors, differing by elements of $\ker(H)$, that all give the exact same syndrome. In order to know which error in this set is the most probable, we need some information about the structure of the underlying noise. A popular noise model to test worst-case scenarios is adversarial noise, where one examines errors below some cutoff weight, which typically increases with the code distance. In this simple error model, the most probable error is usually the one with the lowest weight, and so one usually designs a

decoder which tries to output a minimal-weight correction. Decoding is successful if the combination $\mathbf{e}+\hat{\mathbf{e}}$ leaves the codespace unchanged, i.e. it is an element of the code's stabilizer group. In this paper, we will focus on the adversarial noise model. In the outlook (Sec. 6), we comment on the more realistic scenario of local stochastic noise.

In real hardware, syndrome extraction is a physical operation and will be subject to errors itself. As a consequence, the obtained error syndrome will not always be faithful, and if we naively decode the corrupted syndrome, our inferred correction may unintentionally introduce even more errors than before. As a example, suppose we have a large surface code with no errors on the data qubits but a single ancilla error causing a flipped check in the bulk. If we input this weight-1 syndrome into a matching decoder, it will match the flipped check to a boundary, resulting in an unintentional, residual error of potentially $O(d)$ weight that may overwhelm the code in the next QEC cycle. The typical approach to counteract measurement errors in the surface code is to perform $\Theta(d)$ rounds of syndrome measurements and then decode over this entire history of syndromes [3]. Remarkably, there exist families of codes and decoders where one can keep residual errors under control with only a single round of noisy syndrome measurement; we call such codes single-shot codes [18]. These codes have additional structure that allows a decoder to detect and correct for measurement errors.

We now briefly review two conditions which are sufficient for single-shot error correction against adversarial noise [64, 65], and they encompass all known single-shot codes to date. These two conditions share some similarity, but their decoder implementations can be quite different, as we will explain below.

The first condition comes from a low-error property of the parity-check matrix called confinement, and it roughly stipulates that the number of violated checks, or energy penalty, grows with the error weight, up to some cutoff. For notation, we use $|\cdot|$ to denote the Hamming weight. Since in a stabilizer code, the weight of an error can be arbitrarily increased by appending stabilizer elements, we also define a reduced weight $\|\cdot\|$, which is the minimum weight of an error over its stabilizer group orbit. Formally, confinement is defined as follows.

Definition 2.1 (Confinement [65])**.** *Let $t > 0$ be an integer and $f : \mathbb{Z} \to \mathbb{R}$ an increasing function. For a parity-check matrix $H \in \mathbb{F}_2^{m\times n}$, we say it is (t, f)-confined if for any error $\mathbf{e} \in \mathbb{F}_2^n$ with reduced weight $\|\mathbf{e}\| \le t$, its syndrome $\mathbf{s}(\mathbf{e}) = H\mathbf{e}$ obeys*

$$f\big(|\mathbf{s}(\mathbf{e})|\big) \ge \|\mathbf{e}\| \,. \tag{6}$$

If the confinement function f is an increasing function independent of n, and the confinement cutoff t grows polynomially with n, then the code is single-shot against adversarial noise [65]. For confined codes, decoding typically proceeds by including the parity checks themselves as noisy variables during the weight minimization. Confinement then guarantees that the low-energy state space is partitioned into clusters that are well-separated in Hamming distance [66]. This cluster separation allows a minimum-weight decoder to stay within its starting cluster, i.e. not fail too badly, and thus any residual error will be bounded by the size of the cluster (governed by t and f).

The second condition for single-shot error correction comes from a low-syndrome property called soundness, and it stipulates that low-weight syndromes can always be produced by low-weight errors. Formally, it is defined as follows.

Definition 2.2 (Soundness [64])**.** *Let $t > 0$ be an integer and $f : \mathbb{Z} \to \mathbb{R}$ an increasing function. For a parity-check matrix $H \in \mathbb{F}_2^{m\times n}$, we say it is (t, f)-sound if for any valid syndrome $\mathbf{s} \in \mathrm{im}(H)$ with $|\mathbf{s}| \le t$, there exists an error $\mathbf{e} \in \mathbb{F}_2^n$ satisfying*

$$\|\mathbf{e}\| \le f\big(|\mathbf{s}(\mathbf{e})|\big) \,. \tag{7}$$

Similarly to confinement, if the soundness function f is an increasing function independent of n, and the soundness cutoff t grows polynomially with n, then the code is single-shot against adversarial noise [64]. In fact, for LDPC codes, soundness (Def. 2.2) implies confinement (Def. 2.1), with the two cutoffs being related by the $O(1)$ column weight of H [65]. Nonetheless, soundness is a stricter condition because it entails redundancies amongst the check operators [67], which is not a requirement for confinement. The existence of linearly dependent parity checks means that H is rank deficient and has nontrivial left null vectors. In particular, we can collect these left null vectors into rows of a "metacheck" matrix M which then satisfies $MH = 0$. From the algebraic perspective, we can then extend the original 2-term chain ((1) or either side of (2)) to

$$B \xrightarrow{H} S \xrightarrow{M} R \,, \tag{8}$$

where R is the space of check redundancies amongst the rows of H. Since soundness implies confinement in LDPC codes, we can employ the same decoding strategy as previously mentioned. However, the structure (8) enables an alternative decoding strategy. Notice that $\mathrm{im}(H) \subseteq \ker(M)$, i.e. valid syndromes are codewords of the code associated with the $S \to R$ chain. So given a corrupted syndrome $\mathbf{s}$, we can construct a "metasyndrome" $\mathbf{m}(\mathbf{s}) = M\mathbf{s} \in R$. We can then perform the following two-stage decoding procedure:

1. Decode the metasyndrome $\mathbf{m}$ using M to obtain a repaired syndrome $\hat{\mathbf{s}}$.

2. Decode the repaired syndrome $\hat{\mathbf{s}}$ using H to obtain a correction $\hat{\mathbf{e}}$.

The soundness property (Def. 2.2) then guarantees that the residual error $\mathbf{e} + \hat{\mathbf{e}}$ has bounded size [64].

Examples of single-shot stabilizer codes include the 4D surface code [3], whose CSS parity checks possess the soundness property, and constant-rate HGP codes [68], whose CSS parity checks possess the confinement property.

3 Single-shot codespace preparation

In the previous section, we reviewed two sufficient conditions for single-shot error correction as well as two decoding strategies that leverage their corresponding properties. An important assumption that went into those results was that we started in the codespace of our error-correcting code. The reason is that the notion of "small" for errors and syndromes is only well-defined with respect to the codespace (no errors and zero syndrome), and so in order for those results to hold, we need to assume our codespace has been reliably prepared. In practice, the codespace needs to be prepared from an unentangled product state, and designing such a protocol to be fault-tolerant can be nontrivial.

For large-block CSS codes, a popular method of preparing $|\bar{\mathbf{0}}\rangle$ or $|\overline{+}\rangle$ codeword states is transversal initialization [3]: to prepare logical $|\overline{+}\rangle$,

1. Initialize all physical qubits in $|+\rangle^{\otimes n}$.
2. Measure all Z-type check operators to obtain an initial Z-syndrome.
3. Decode the Z-syndrome and apply the resulting X-correction.

The procedure for logical $|\bar{\mathbf{0}}\rangle$ follows upon switching the roles of X and Z. In the absence of errors, the first step guarantees that we are in the simultaneous $+1$ eigenstate of all X-checks and logical $\bar{X}$ operators. Since Z-checks commute with X-checks, the second step maintains the first property while projecting our product state into random[1] ± 1 eigenstates of the Z-checks. At this point, our state is effectively $|\overline{+}\rangle$ up to a (potentially large) Z error. The third and last step returns us to the simultaneous $+1$ eigenstate of all the Z-checks. Note that any suitable correction in the last step suffices because we are not worried about logical $\bar{X}$ errors on the $|\overline{+}\rangle$ state. In practice, the correction in step 3 is not actually performed, but rather the Z-syndrome from the second step is kept in software as a baseline (i.e. Pauli frame) from which future syndromes are subtracted from.

Like single-shot error correction, the goal of single-shot state preparation is to initialize a desired quantum state in a fault-tolerant way such that residual errors are controlled. Examining the transversal initialization strategy, we see that the first step that initializes $|+\rangle^{\otimes n}$ consists of n single-qubit operations and so is inherently fault-tolerant. If our code is LDPC, then the second step can be implemented with a constant-depth circuit of 2-qubit gates; the propagation of correlated errors is bounded, and hence this step is also inherently fault-tolerant. The third step is where dangerous errors can occur. Suppose we obtain an unfaithful syndrome in step 2 due to measurement noise. If our code is sound and contains metachecks, we can detect and repair these syndrome errors using the method described in the previous section. We then set our repaired syndrome as the baseline syndrome for step 3, and the soundness of the code guarantees that any residual error remains small; this procedure has been recently emphasized in [69]. Indeed, for sound codes, there is not really a distinction between initialization and subsequent QEC cycles.

On the other hand, if our code does not contain any metachecks, then every parity check is an independent constraint and hence has equal probability to be measured as either $+1$ or -1 during transversal initialization. As a consequence, all 2^m error syndromes are equally likely (for m checks), and we will have no information at our disposal to verify whether our obtained syndrome is faithful. If we naively pretend that this syndrome is faithful, then we risk encountering the same issue that plagued single-shot decoding of the surface code: a single flipped check may propagate through the decoder to a large physical error. Because constant-rate HGP codes can exhibit confinement without any metachecks[2] [68], they fall into a niche class of codes which possess single-shot QEC but *not* single-shot initialization.

3.1 Overview of the protocol

We now describe our single-shot protocol for preparing the logical $|\overline{+}\rangle$ state of constant-rate HGP codes [68], which support single-shot error correction due to confinement but are not sound due to the lack of metachecks. The protocol will consist of two stages, which we will independently describe and analyze in the following two subsections. The main idea will be to first construct a different, but related, "thickened" code that possesses soundness and hence can be prepared in a single shot via transversal initialization. Then we perform fault-tolerant code-switching to return to our original HGP code. The preparation of $|\bar{\mathbf{0}}\rangle$ is analogous.

We will use tildes to denote quantities related to the thickened code — e.g. $[\![\tilde{n}, \tilde{k}, \tilde{d}]\!]$. The first stage of the protocol to prepare $|\overline{+}\rangle$ can be summarized as:

1. Initialize all physical qubits in $|+\rangle^{\otimes \tilde{n}}$.
2. Measure all Z-type check operators to obtain

[1] up to metacheck constraints

[2] Random classical LDPC codes have full-rank parity-check matrices and good parameters with high probability [70].

an initial Z-syndrome $\mathbf{s}_0$ and Z-metasyndrome $\mathbf{m}_0 = \tilde{M}_Z \mathbf{s}_0$.

3. Decode the Z-metasyndrome using $\tilde{M}_Z$ (11) to obtain a repaired syndrome $\hat{\mathbf{s}}_0$.

4. Decode the repaired syndrome $\hat{\mathbf{s}}_0$ using $\tilde{H}_Z$ (10b) to return to $|\overline{+}\rangle_{\text{th}}$, up to a small residual X error.

The second stage can be summarized as:

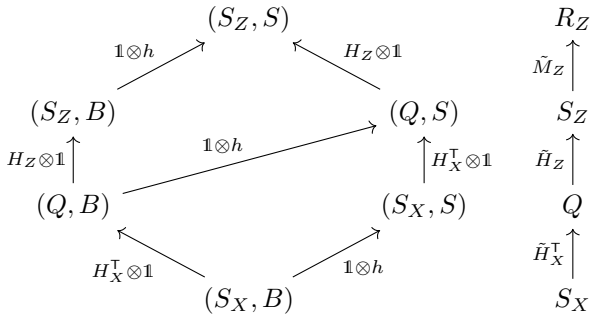
5. Measure X on all physical qubits except on one of the boundary HGP codes and reconstruct a bulk X-syndrome.

6. Input the above bulk X-measurement outcomes, X-syndrome, and a suitable single-shot decoder into Algorithm 1 to obtain a final boundary Z-correction.

Note that only the first two steps of stage 1 and the first step of stage 2 are physical operations and hence prone to noise. Since we begin in the product $|+\rangle^{\otimes \tilde{n}}$ state, the only troublesome noise during stage 1 are physical Z errors and syndrome measurement errors. Syndrome errors are accounted for during the syndrome repair step, and physical Z errors are corrected during the second stage. Since all physical operations are either transversal or involve a finite-depth circuit (e.g. for single-ancilla measurement schemes), error propagation is also bounded. Thus, the entire protocol will be fault-tolerant as long as we are able to reliably correct the errors mentioned above.

3.2 Stage 1: Homological dimension jump

In the first stage, we will transversally initialize the logical $|\overline{+}\rangle$ state of a thickened code, which is built from the homological product of our original $[\![n, k, d]\!]$ HGP code, described by a 3-term chain (2), with a suitable classical code of distance d, described by a 2-term chain (1) [71, 72]. To make the analysis easy to follow, we will first use a 1D repetition code of length d for thickening. Later in Sec. 4, we will discuss alternative classical codes which go beyond repetition. Since the original HGP code can be described by a two-dimensional complex, we can interpret this step as a dimension jump from a two-dimensional complex to a three-dimensional one. Denoting H_X, H_Z and h as the CSS parity-check matrices of the HGP code and the parity-check matrix of the repetition code respectively, the product complex is given by



(9)

which we can collapse into a 4-term chain as shown on the right. We thus obtain a new CSS code with parity-check matrices

$$\tilde{H}_X = \left(H_X \otimes \mathbb{1} \;\middle|\; \mathbb{1} \otimes h^\mathsf{T} \right) \tag{10a}$$

$$\tilde{H}_Z = \left(\begin{array}{c|c} H_Z \otimes \mathbb{1} & \mathbf{0} \\ \mathbb{1} \otimes h & H_X^\mathsf{T} \otimes \mathbb{1} \end{array} \right) \tag{10b}$$

and Z-metacheck matrix

$$\tilde{M}_Z = \left(\mathbb{1} \otimes h \;\middle|\; H_Z \otimes \mathbb{1} \right), \tag{11}$$

from which we can verify that $\tilde{H}_X \tilde{H}_Z^\mathsf{T} = H_X H_Z^\mathsf{T} \otimes \mathbb{1} + 2\big(H_X \otimes h^\mathsf{T}\big) = 0$ and $\tilde{M}_Z \tilde{H}_Z = 2\big(H_Z \otimes h\big) + H_Z H_X^\mathsf{T} \otimes \mathbb{1} = 0$ as required. Geometrically, the Tanner graph of this thickened code consists of "sheets" of the original HGP code interleaved with intermediate qubits and Z-checks; see Fig. 3 for an illustration. The sheets are labeled by the left block in (10) and the intermediate qubits by the right block. For simplicity, we will assume that the two classical input codes for the HGP code are the same, so that H_X and H_Z both have m rows. The quantum code parameters of the thickened code are then given by [73]

$$\tilde{n} = nd + m(d-1) = O(n^{3/2}) \tag{12a}$$

$$\tilde{k} = k = O(\tilde{n}^{2/3}) \tag{12b}$$

$$\tilde{d}_X = d^2 = O(\tilde{n}^{2/3}) \tag{12c}$$

$$\tilde{d}_Z = d = O(\tilde{n}^{1/3}), \tag{12d}$$

where we used the fact that $d = O(\sqrt{n})$ for HGP codes.

Importantly, we can utilize the metachecks (11) to catch and repair any measurement errors during the transversal initialization. The end result is the logical $|\overline{+}\rangle$ state of the thickened code up to a small residual X error. We provide a bound on the residual error in Appendix A, but we comment that it is essentially the same proof used to demonstrate single-shot error correction in sound codes [64]; see also [69]. The correction for Z errors is performed in the next stage.

3.3 Stage 2: Dimensional collapse

After the completion of stage 1, we can assume, up to a small residual error, that we are in the codespace

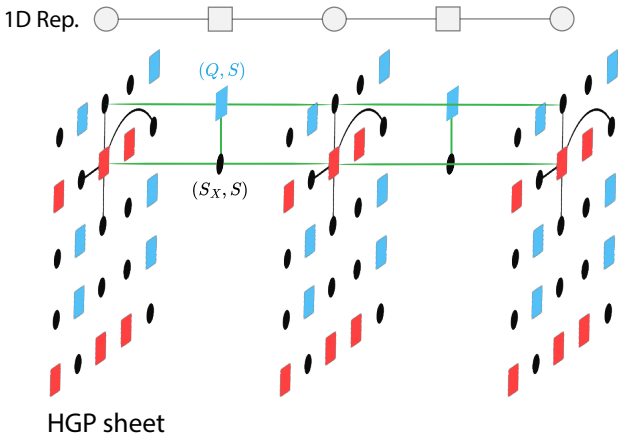


Figure 3: The thickened code (10) for the HGP code in Fig. 2 with $z = 3$ is illustrated. Three copies of the original HGP code are connected (green) by intermediate qubits and Z-checks, corresponding to the (Q, S) and (S_X, X) spaces of (9). Intermediate Z-checks connect pairs of identified qubits along adjacent HGP sheets, and intermediate qubits connect pairs of identified HGP X-checks. The 1D repetition code is overlayed above for reference. To reduce clutter, we only display a subset of intermediate qubits and Z-checks.

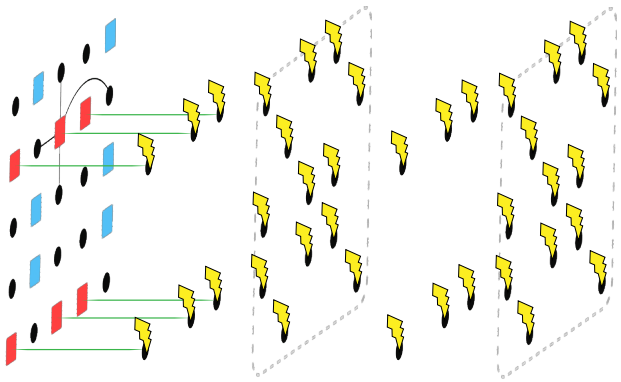


Figure 4: The dimensional collapse in stage 2 is depicted. All qubits other than a boundary HGP sheet are measured in the X basis (denoted by lightning bolts). The intermediate qubits adjacent to the boundary sheet are connected by green edges to the boundary X-checks.

of the thickened code. In the second stage, we begin by measuring all qubits in the X basis except for one of the "boundary" HGP codes, which corresponds to a boundary bit of the 1D repetition code; see Fig. 4 for an illustration. As a result, we effectively collapse the extra "dimension" introduced by the homological product, and we will be left with our original HGP code up to a (potentially large) Z error.

To see how the dimensional collapse performs the desired code-switching, first observe that the HGP Z-checks on the left boundary do not extend past the boundary, and so these checks remain invariant during the bulk measurement. On the other hand, the boundary X-checks take the form

$$\tilde{H}_{X,\partial} = \left(H_X \,\middle|\, \mathbb{1} \right) , \tag{13}$$

where the left block denotes qubits on the boundary, and the right block denotes qubits in the adjacent intermediate sheet (green lines in Fig. 4). There is thus a one-to-one correspondence between each intermediate qubit and each X-check on the boundary. The random measurement outcomes of these intermediate qubits will determine the eigenvalues of the boundary X-checks. It follows that the boundary qubits are in the simultaneous eigenspace of all HGP Z-checks and X-checks, up to a potentially large Z error due to the random X-check eigenvalues. It remains to show that the logical operators reduce to that of the HGP code on the boundary. Denote G_X and G_Z as the X-type and Z-type generator matrices for the HGP code, and denote g as the generator matrix of the repetition code. Using the Künneth formula for product complexes, one can construct X-type and Z-type generator matrices for the logical operators of the thickened code [61]:

$$\tilde{G}_Z = \left(G_Z \otimes \mathbb{1}_{d,1} \,\middle|\, \mathbf{0} \right) \tag{14a}$$

$$\tilde{G}_X = \left(G_X \otimes g \,\middle|\, \mathbf{0} \right) , \tag{14b}$$

where $\mathbb{1}_{d,1}$ is a (truncated) $d \times d$ identity matrix with only the first column being nonzero. One can quickly verify that $\tilde{H}_X \tilde{G}_Z^\mathsf{T} = \tilde{H}_Z \tilde{G}_X^\mathsf{T} = 0$ as required. (14a) tells us that the logical $\bar{Z}$ operators of the thickened code are exactly the same as those of the HGP code, and since they live on the boundary, they survive the dimensional collapse. (14b) tells us that the logical $\bar{X}$ operators of the thickened code are extended in the "third dimension", but their intersections with the boundary are exactly the HGP logical $\bar{X}$ operators. Thus, up to a physical Z error, the dimensional collapse successfully code-switches between our thickened code and our HGP code.

At this point, in terms of fault tolerance, it seems like we have just returned to the same scenario in which we were hoping to avoid, this time with the X-checks. The X-checks of our HGP code post-collapse have random ± 1 eigenvalues, and it is not immediately clear how to return to the codespace without inflicting a logical $\bar{Z}$ error. Fortunately, there is a way to correct these errors by utilizing the measurement outcomes from the bulk qubits, as we describe below.

In the state-vector picture, the logical $|\overline{+}\rangle$ state of the thickened code has the form

$$\begin{aligned} |\overline{+}\rangle_{\text{th}} &\propto \left(\prod_{\{\tilde{C}_Z\}} \frac{\mathbb{1} + \tilde{C}_Z}{2} \right) |+\rangle^{\otimes \tilde{n}} \\ &\propto \sum_{\tilde{O}_Z \in \tilde{\mathcal{S}}_Z} \tilde{O}_Z \, |+\rangle^{\otimes \tilde{n}} , \end{aligned} \tag{15}$$

where $\{\tilde{C}_Z\}$ is the collection of all Z-checks, and $\tilde{\mathcal{S}}_Z$ is the Z-type stabilizer subgroup of the thickened code. So we see that in the X-basis, the logical $|\overline{+}\rangle$ state is

a uniform superposition over all Z-stabilizer elements applied to $|+\rangle^{\otimes \tilde{n}}$. If we measure X on all $\tilde{n}$ qubits, then we collapse the superposition onto a random basis state, which is equivalent to applying a random Z-stabilizer element to $|+\rangle^{\otimes \tilde{n}}$. Since we only measure a subset of all the qubits, the superposition (15) instead only partially collapses. Recall that the Z-stabilizer subgroup generated by Z-checks supported only on the boundary is precisely $\mathcal{S}_Z$ of our HGP code. Our post-collapse state then takes the form

$$\begin{aligned} \mathcal{M}_{X,\text{bulk}} |\overline{+}\rangle_{\text{th}} &\propto \tilde{O}'_Z \sum_{O_Z \in \mathcal{S}_Z} O_Z \, |+\rangle_{\partial}^{\otimes n} \otimes |+\rangle_{\partial^c}^{\otimes(\tilde{n}-n)} \\ &\propto \tilde{O}'_Z \, |\overline{+}\rangle_{\partial} \otimes |+\rangle_{\partial^c}^{\otimes(\tilde{n}-n)} \, , \end{aligned} \tag{16}$$

where $\tilde{O}'_Z \in \tilde{\mathcal{S}}_Z \setminus \mathcal{S}_Z$. In particular, the only component of $\tilde{O}'_Z$ which acts nontrivially on $|\overline{+}\rangle_{\partial}$ consists of intermediate Z-checks with (partial) support on the boundary. Since $\tilde{O}'_Z$ arises from the randomness of measurement, we call it an *intrinsic* Z error.

To return to the codespace of our boundary HGP code, we need to undo the boundary action of the intrinsic Z error $\tilde{O}'_Z$. We will leverage a causal structure induced by the 1D repetition code to perform this correction, generalizing a similar procedure for the 3D surface code [54]. Starting from the opposite boundary, we locate the qubits in the $|-\rangle$ state and apply Z to its identified qubit in the next sheet, which effectively pushes these Z "errors" onto the next HGP sheet. We then iterate this procedure starting from this next sheet until we terminate at our boundary HGP code. To see why this algorithm undoes $\tilde{O}'_Z$ up to an element of $\mathcal{S}_Z$, we will analyze the effect of this procedure on the two types of Z-checks: intermediate checks and sheet checks. Since the algorithm is linear, our analysis will then also apply to arbitrary elements of $\tilde{\mathbf{S}}_Z$. Recall that an intermediate Z-check connects a pair of identified qubits supported on adjacent HGP sheets. So when we push our Z errors between two neighboring sheets, we will cancel out its effect on the succeeding sheet. For a sheet Z-check located strictly on a single HGP sheet, we will push it all the way to the boundary, but since it is in the HGP stabilizer group, its effect will be trivial. Therefore, by linearity, the only nontrivial Z errors that we push onto the boundary precisely correspond to intermediate Z-checks that touch the boundary. One can interpret the above procedure as computing the feedback corrections in a teleportation scheme [74, 75].

The aforementioned algorithm only works in the absence of external errors. If we have external Z errors such that the bulk pattern of $|-\rangle$ does not correspond to any Z-stabilizer element, we need to first correct these errors before we can proceed. Using the bulk measurement outcomes, we can infer the values of the bulk X-checks and reconstruct an X-syndrome in an attempt to correct for external Z errors. However, since we did not measure our boundary sheet, what we have is an incomplete X-syndrome, and so we need to be careful with how to proceed with decoding. Fortunately, we can leverage the fact that our HGP code support single-shot decoding to reliably correct these Z errors even with an incomplete X-syndrome. The crucial observation is that the bulk X-check Tanner graph is precisely the spacetime decoding graph for $d-1$ noisy rounds of X-syndrome measurement of the HGP code; the intermediate qubits label the location of syndrome measurement errors in the corresponding spacetime code [76–78]; see Fig. 5 for an illustration. Using this spacetime mapping, we can reinterpret our bulk X-syndrome as $d-1$ rounds of noisy HGP X-syndromes, with the asymmetry of the bulk Tanner graph (missing boundary sheet on one side) providing us with an arrow of time. In this manner, we can employ a single-shot decoder for our HGP code to decode the bulk X-syndrome one sheet at a time, starting from the opposite HGP boundary and ending on our desired boundary. Furthermore, we can bake in this decoder during each step of the algorithm. The combined algorithm is described below in pseudocode.

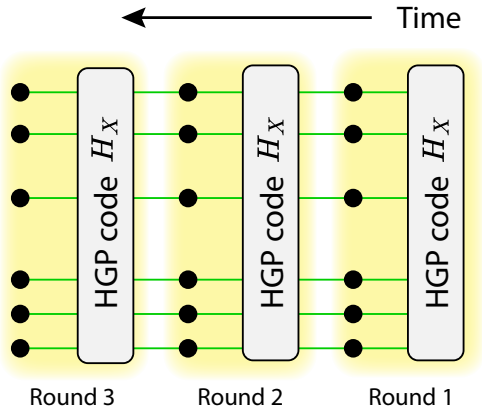


Figure 5: The interpretation of the bulk X-check Tanner graph as a spacetime code is depicted. Gray rectangles denote HGP sheets, and black circles denote the intermediate qubits connected to each X-check in its neighboring HGP sheets. The arrow of time runs from the opposite HGP boundary to the unmeasured one. The combination of each HGP sheet with its subsequent intermediate qubits can be interpreted as a round of noisy syndrome measurement.

As long as residual errors remain small during each round of single-shot decoding, then the final residual error that is pushed onto the boundary should also be small. A quantitative bound on the residual error is presented in Appendix B. We note that one can also simply decode the entire bulk syndrome at once, which should further decrease any residual Z error on the boundary at the cost of increased decoding complexity.

The fault tolerance of the entire protocol is analyzed in Appendix C, combining the results of Sections 3.2 and 3.3.

Algorithm 1: Z-correction for stage 2

Input : bulk X-measurement outcomes $\{\mathbf{m}_i\}$ arranged by sheet, bulk X-syndromes $\{\mathbf{s}_i\}$ arranged by sheet, and a single-shot HGP decoder

Return : boundary Z-correction vector $\mathbf{z} \in \mathbb{F}_2^n$

$\mathbf{z} = \text{zeros(n)}$ // stored Z errors

$\hat{\mathbf{s}} = \text{zeros(m)}$ // inferred syndrome error

for $\tau = 1$ **to** $d-1$ **do**

  $\mathbf{s}' = \mathbf{s}_\tau + \hat{\mathbf{s}}$ // $\tau = 1$ is the opposite boundary

  Decode $\mathbf{s}'$ to obtain an inferred syndrome error $\hat{\mathbf{s}}$ and an inferred Z error $\hat{\mathbf{z}}$

  $\mathbf{z} \mathrel{+}= \mathbf{m}_\tau + \hat{\mathbf{z}}$

end

4 Beyond repetition

In this section, we present a simple modification to the protocol which reduces the spatial overhead by a factor of at most 2, but at the cost of needing to initialize multiple HGP codes in parallel. Recall that for stage 1 to be fault-tolerant against syndrome measurement errors, we needed our classical code in the homological product to have the same distance as our quantum HGP code. We employed a $[d, 1, d]$ repetition code and saw how its induced causality structure allowed us to formulate a decoding algorithm (Alg. 1) for stage 2. The spatial overhead of the protocol essentially comes from the ratio n/k, which is equal to d for the repetition code.

From the perspective of stage 2, we did not need the thickness to be $\ell = d$ because our HGP code was single-shot; $\ell = O(1)$ is sufficient for stage 2 to be fault-tolerant. It may be tempting to try and isolate more HGP sheets during stage 2, either by using a different classical code or by not measuring some HGP sheets, in order to lower the spatial overhead *per HGP sheet*. However, naively doing so destroys the causality structure that we crucially leveraged to correct the intrinsic Z error (16) on the boundary during stage 2. In particular, we may encounter a "causal split" in the classical Tanner graph, where there is more than one direction (check) that we can choose; see Fig. 6. When this happens, we will encounter a $|-\rangle$ on one of our bulk HGP sheets (corresponding to a classical bit in the homological product) that could have been caused by multiple intermediate Z-checks (corresponding to a classical check), and without additional information it is unclear which Z-check we should pick.

The modification we propose is to replace the 1D repetition code with a $[n, n-1, 2]$ single-parity code, whose Tanner graph resembles a star. To increase the code distance, we can repeat the physical bits (i.e. concatenate with a repetition code) so that the final

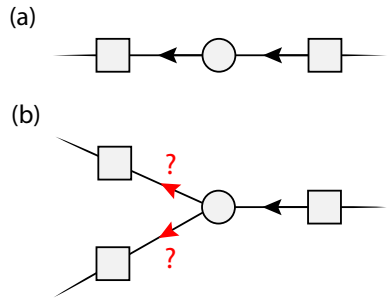


Figure 6: A causal structure is overlayed on top of two classical Tanner graphs. (a) In the repetition code, there is a clear arrow of time from right to left, which is used to correct the intrinsic Z error on the left boundary during stage 2. (b) On a general Tanner graph, we may encounter a causal split, where there is no definitive path to take.

Code	n	k	$\frac{n}{kd}$	Weight
1D Rep.	d	1	1	2
Star	$zd/2$	$z-1$	$\frac{1}{2} \cdot \frac{z}{z-1}$	z

Table 1: Parameters of the 1D repetition code and the degree-z star code are tabulated. k is also the number of HGP codes produced at the end of the protocol. The penultimate column displays the overhead reduction compared to the repetition code, and the last column displays the maximum check weight.

"star" code has parameters $[n, z-1, 2n/z]$, where z is the degree of the central check, and n is a multiple of z. Geometrically, codewords have support on paths which begin and end on distinct branches of the star. Importantly, each physical bit is connected to at most two checks, and so we preserve the same causality structure that emerged in the repetition code. Specifically, we choose any branch as our starting point and call it the *incoming* branch. We then label all other branches as *outgoing* branches; see Fig. 7a for an illustration. When we perform the dimensional collapse in stage 2, we will now refrain from measuring the HGP code sheets located at the endpoints of all $z-1$ outgoing branches.

Now that the arrows of time have been established, we can proceed with stage 2 but with the following modifications to Algorithm 1. The rule for pushing Z-corrections between sheets now follows the Tanner graph of the star code. In particular, when we arrive at the center of the star along the incoming branch, we will need to push our Z-correction onto each of the $z-1$ outgoing branches. Afterwards, the rest of the algorithm is the same as before within each outgoing branch. The correctness of this modified algorithm follows the same linearity argument given in Sec. 3.3.

We can now compute the overhead reduction of the star code over the repetition code with fixed distance d. We use a heuristic $n/(kd)$, which equals one for the repetition code and is less than one for more efficient

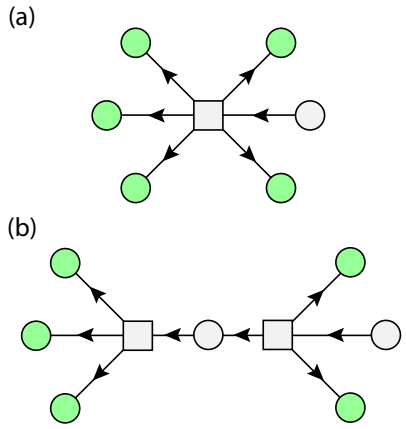


Figure 7: (a) The Tanner graph of the star code is depicted (repeated bits not shown). Green circles denote the bits which will host the final unmeasured HGP codes, and black arrows label the causal structure. (b) The same illustration is shown for a weight-reduced star code with one auxiliary bit in the center.

encodings. For our $[n, z-1, 2n/z]$ star code, we have

$$\frac{n}{kd} = \frac{1}{2} \cdot \frac{z}{z-1} > \frac{1}{2} \,. \tag{17}$$

In particular, the overhead reduction per HGP code approaches $1/2$ for increasing z. A comparison between the star code and the repetition code is catalogued in Table 1. Although stage 1 fault tolerance may begin to suffer at large z due to deeper syndrome extraction circuits, one can perform weight-reduction [46, 79] on the star code to evenly distribute the central check weight amongst multiple checks, at the cost of extra auxiliary bits; see Fig. 7b for an example. Since the weight-reduced code resembles a "stretched" version of the original code, with auxiliary bits connected in a chain-like fashion, it inherits the causal structure as well.

Lastly, we need to ensure that each HGP code is indeed in its logical $|\overline{+}\rangle$ state at the end of the protocol. We can show this result by examining the codewords of the star code. We can write the generator matrix of the $[n, n-1, 2]$ code in the form

$$g_\star = \left(\mathbb{1}_z \,\middle|\, \mathbf{1}_{z\times 1} \right) , \tag{18}$$

where $\mathbb{1}_z$ is the $z \times z$ identity matrix, and $\mathbf{1}_{z\times 1}$ is a column of all ones corresponding to the starting branch. For the thickened code, we can write its X-type generator matrix as

$$\tilde{G}_X = \left(G_X \otimes g_\star \,\middle|\, \mathbf{0} \right) , \tag{19}$$

where G_X is the X-type generator matrix for the HGP code, analogous to (14b). From (18) and (19), we see that for each HGP logical $\bar{X}$ operator, there exist $z-1$ distinct logical $\bar{X}$ operators in the thickened code supported on HGP sheets labeled by the incoming and outgoing branches of the star code. Since each branch hosts a distinct logical operator, and our initial logical state of the thickened code was the product $|\overline{+}\rangle$ state, the resulting state on the $z-1$ HGP codes is $|\overline{+}\rangle^{\otimes(z-1)}$.

Although the overhead reduction with the star code is only a constant, it applies at all distances (not just an asymptotic result). So if one desires to prepare multiple HGP code blocks in parallel, the star code provides an immediate spatial reduction, which may be beneficial for moderate system sizes. From Fig. 7, we can see that the causality structure and arrows of time can be interpreted as a type of message-passing schedule on the Tanner graph. For tree-like Tanner graphs such as that of the star code, it is quite easy to establish a schedule as we have seen. However, the code parameters of Tanner graphs without loops is extremely limited [80]. For more general Tanner graphs, one may need to worry about scheduling around closed loops and choosing endpoints carefully. In particular, an endpoint vertex cannot have any outgoing arrows since we do not measure its corresponding HGP sheet.

5 Numerical simulations

In the fault-tolerance analysis of the previous section, we had assumed both an adversarial error model as well as theoretically optimal decoders. In practice, we will mostly likely encounter a local stochastic error model, such as independent single-qubit errors. We will also need an efficient decoder since it is known that maximum-likelihood decoding of linear codes is generically NP hard [81].

We numerically benchmark the performance of the protocol under a phenomenological noise model: each physical qubit has an independent probability p to incur a Pauli X or Z error, and each Z-check in the first stage is incorrectly measured with probability p. For syndrome decoding, we employ an open-source message-passing decoder [82] based on belief propagation with ordered-statistics postprocessing (BP+OSD) [83], which displays good performance as a general-purpose quantum LDPC decoder [55, 83, 84]. The time complexity of BP is $O(n)$ while that of OSD is $O(n^3)$. We configure BP+OSD with the "minimum-sum" strategy (20 iterations) for BP and the "combination-sweep" strategy (search depth 20) for OS.

For the first simulation, we construct a $[\![549, 9, 9]\!]$ HGP code by taking the hypergraph product of a classical $[18, 3, 9]$, $(5, 6)$-LDPC code with itself; the classical parity-check matrix is randomly selected using the bipartite configuration model. We then construct the thickened code (10) by taking a homological product of this HGP code with a $[\ell, 1, \ell]$ repetition code, where ℓ is a tunable thickness parameter. Without loss of generality, we assume that the true initial Z-syndrome is the all-zero vector. We then flip the value

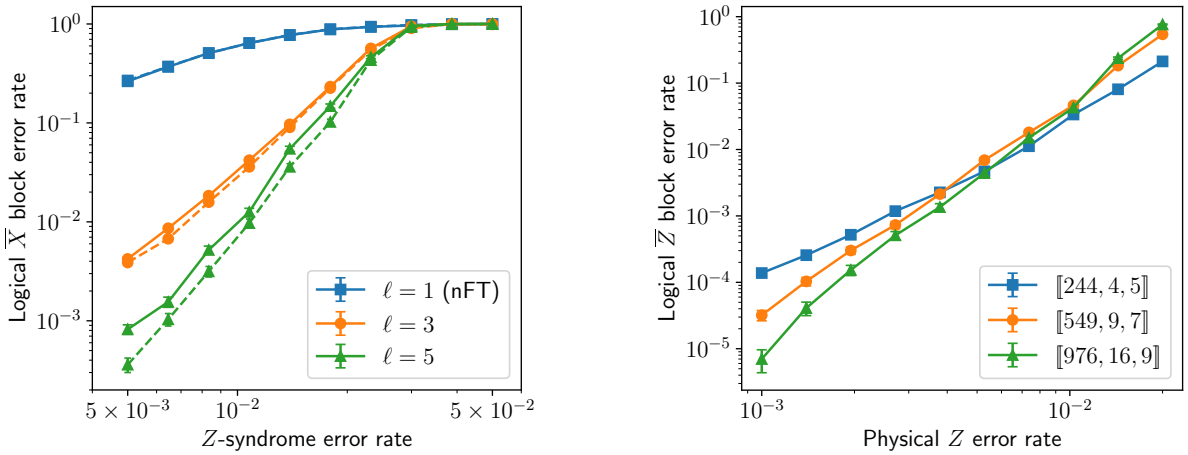


Figure 8: **Left:** The logical $\bar{X}$ error rate as a function of the syndrome error rate is plotted for the first simulation of a $[\![549, 9, 9]\!]$ HGP code with varying thickness ℓ (solid lines). The same logical error rate for ℓ rounds of repeated syndrome measurements is also displayed for comparison (dashed lines). **Right:** The logical $\bar{Z}$ error rate as a function of the physical Z error rate is plotted for the second simulation of three HGP codes derived from the $(5, 6)$-LDPC ensemble. Error bars denote the standard error $\sigma = \sqrt{p(1-p)/N}$ for N samples.

of each Z-check in the syndrome with probability p to obtain a corrupted syndrome. We repair this corrupted syndrome using BP+OSD with respect to the Z-metacheck matrix (11), and then decode the repaired syndrome using BP+OSD with respect to $\tilde{H}_Z$ (10b) to obtain an X-correction. Since our true initial syndrome was zero by assumption, this X-correction hence becomes the residual X error. To account for the second stage of the protocol, we take the boundary restriction of this residual X error so that what remains is our original HGP code with a residual X error. To verify whether this residual error is dangerous, we apply a new round of independent X errors (with probability p per qubit), add it to the residual error, and finally decode this combined error using BP+OSD with respect to H_Z (4b). We declare success if no logical $\bar{X}$ is applied at the end and failure otherwise. The simulation results are showcased on the left plot of Fig. 8. Note that $\ell = 1$ corresponds to non-fault-tolerant (nFT) transversal initialization of the original HGP code with no syndrome repair. Below $p \lesssim 3 \times 10^{-2}$, we observe significant suppression of logical errors upon increasing the thickness ℓ, providing evidence that the metachecks are successfully controlling residual errors. We also compare our protocol with the traditional method of repeated syndrome measurements and observe nearly identical performance.

For the second simulation, we randomly select three codes of increasing length from the $(5, 6)$-LDPC ensemble and take the hypergraph product of each code with itself; each resulting HGP code has rate $k/n = 1/61 \approx 1.64\%$. For each HGP code, we construct its corresponding thickened code by taking the homological product of the HGP code with a classical repetition code whose length is equal to the distance of the HGP code. To simulate the decoding process for each thickened HGP code, we apply a Z error to each physical qubit with probability p, compute the bulk error syndrome, and then run the decoding strategy given in Algorithm 1, which will produce a final residual error on our boundary HGP code. For the single-shot BP+OSD decoder, we use the augmented parity-check matrix [55]

$$H_X^{ss} = \left(H_X \mid \mathbb{1} \right) , \tag{20}$$

where we connect an auxiliary qubit to each X-check. Corrections on the auxiliary qubits denote inferred syndrome errors. Note the similarity to (13). After Algorithm 1 has terminated, we probe the residual error with a final round of perfect syndrome decoding on the HGP code and verify whether a logical error has occurred. These simulation results are showcased on the right plot of Fig. 8. For the family of HGP codes built from the $(5, 6)$-LDPC ensemble, we observe threshold-like behavior around a physical Z error rate of $p \approx 6 \times 10^{-3}$. We did not observe any definitive evidence of a threshold when using input LDPC ensembles with degrees less than $(5, 6)$ and similar system sizes.

To conclude this section, we briefly comment on the classical decoding complexity of the entire protocol under BP+OSD. Recall that the thickened code has $\tilde{n} = \Theta(n^{3/2})$ (12a) and $\dim \tilde{M}_Z = \Theta(\tilde{n})$ (11). Hence the syndrome repair step during stage 1 has time complexity $O(n^{9/2})$ from OSD. If we run the sequential decoder of Algorithm 1 for the second stage, then the decoding complexity is $O(n^3\sqrt{n}) = O(n^{7/2})$. If we decoded the entire bulk syndrome at once, then the decoding complexity is $O(n^{9/2})$. Parallelized algorithms leveraging LDPC properties [85–87] may significantly reduce the decoding complexity for low error rates.

6 Outlook

We have designed a single-shot $|\overline{\mathbf{0}}\rangle$ / $|\overline{+}\rangle$ state-preparation protocol for constant-rate HGP codes and demonstrated its fault tolerance against adversarial noise. The construction is inspired by the notion of dimension-jumping in 2D topological codes [49, 54], and we provide a direct generalization to HGP codes using the homological product. The key ingredient is a mapping between a thickened code and a spacetime code, induced by the homological product with a structured classical code. This mapping allows us to import existing single-shot HGP decoders [40, 55] to perform fault-tolerant code-switching during the second stage of the protocol.

Since the HGP codes we study are LDPC, the circuit depth of our protocol (with enough ancillas) is $O(1)$, which may provide significant improvements to logical clock speeds in fault-tolerant architectures where state preparation is a key subroutine [69]. Unfortunately, the protocol does not preserve the constant-rate property of HGP codes, which is often desirable from a physical resource standpoint [47, 48]. In terms of a combined spacetime overhead, our protocol essentially trades the original $O(d)$ time overhead with an $O(d)$ spatial overhead, and so we only expect the protocol to be practical in the regime where computational time becomes more valuable than physical qubit number. While not a completely satisfactory solution, the protocol is a first step in the direction of efficiently preparing HGP codewords, and we hope that the ideas set forth in this paper accelerate progress along this research thrust. In particular, it would be interesting to explore generalizations of Sec. 4, such as whether other graphs could further lower the spatial overhead, perhaps even to a constant.

We also emphasize that our protocol is not limited to just HGP codes. Although the thickening procedure (9) applies to any CSS code, we require a special structure in its 3-term chain (2) for fault tolerance. For stage 1 to be fault-tolerant against syndrome measurement errors, we required a parity-check matrix of the thickened code to exhibit soundness. Since HGP codes are formed from the homological product, their classical "transpose" codes (by relabeling the CSS 3-term chain (2) as (8), which can also be interpreted as a gauging procedure [88]) are sound. This soundness is then inherited by the thickened code upon taking an additional homological product [64]. So in order for us to use this protocol for some generic CSS code, we need its classical transpose code to be sound. Families of good quantum LDPC codes satisfy this property since their classical transpose codes are locally testable with linear soundness functions [25–27, 89]. The second important ingredient is a single-shot decoder for stage 2, which has been demonstrated with the quantum Tanner codes [90]. Alternatively, we could have simply just used a constant-rate, four-dimensional homological product code which provides both $O(1)$ spatial and temporal overheads [69]. However, these $O(1)$ constants are rather large in practice (e.g. rate gets decreased by a power of 4), and it remains to be seen whether higher-dimensional homological product codes can rival their two-dimensional cousins in finite-length performance.

It would also be interesting to see whether a sustainable threshold for the protocol can be proven for local stochastic noise. Even though for any local error rate there are $O(n)$ errors on average, because the code is LDPC, typical error configurations form separated clusters that can be decoded independently [91]. Since HGP codes can have linear confinement [68], they do exhibit thresholds under local stochastic noise [65]. As a result, we already know that stage 2 has a threshold under local stochastic noise due to the spacetime mapping, and it is precisely the threshold for single-shot error correction. It remains to be seen whether a sustainable threshold can be proven for stage 1. Our numerical simulations in Sec. 5 suggest that a sustainable threshold should exist. It has also recently been shown that higher-dimensional hypergraph product codes (homological products of 1-complexes) support distance-preserving syndrome extraction circuits [92], and so we further expect competitive performance under circuit-level noise.

Lastly, we speculate that our protocol could potentially provide an avenue to non-Clifford gates in HGP codes. Since the logical operators of HGP codes essentially mimick their classical codewords along horizontal and vertical directions (in a 2D layout), they can be chosen such that logical $\bar{X}$ and $\bar{Z}$ only intersect on at most one physical qubit. As a result, transversal gates will be limited to the Clifford group [93] just like for 2D codes [15]. However, for the surface code, there exist constructions in 3D which enable transversal non-Clifford gates [17, 94], and one can use dimension-jumping to apply these ideas to 2D [50, 51, 54], thereby completing a universal gate set. It would interesting to see if the thickening procedure can be modified such that the thickened code admits transversal non-Clifford gates, which combined with single-shot dimensional collapse, may provide a route to circumvent magic state distillation in constant-rate HGP codes.

Acknowledgements

We thank Victor Albert for pointing out the connection to spacetime codes. We also thank Daniel Gottesman and Michael Gullans for valuable discussions, and especially Noah Berthusen and Shi Jie Samuel Tan for feedback on the draft. This work is sponsored by the Department of Energy Office of Advanced Scientific Computing Research's (DoE ASCR) Quantum Testbed Pathfinder program via Award No.

DE-SC0019040 and No. DE-SC0024220. We also acknowledge support from the DARPA MeasQuIT program.

Data availability

All simulation code and data are available at this GitHub repository.

A Fault-tolerance analysis of stage 1

We first need to introduce some extra notation. Recall the 4-term chain of (9):

$$S_X \xrightarrow{\tilde{H}_X^\mathsf{T}} Q \xrightarrow{\tilde{H}_Z} S_Z \xrightarrow{\tilde{M}_Z} R_Z\,. \tag{21}$$

Define the single-shot distance d_{ss} as the minimum weight of nontrivial elements in the second homology group $\mathcal{H}_2 \equiv \ker \tilde{M}_Z / \operatorname{im} \tilde{H}_Z$, comprised of Z-syndromes that satisfy all Z-metachecks but cannot be produced by any physical X error. By convention, we set $d_{\text{ss}} = \infty$ if $\mathcal{H}_2$ is trivial.

Denote $\tilde{m} = md + n(d-1)$ as the number of Z-checks in the thickened code (i.e. rows of $\tilde{H}_Z$). We now prove the following Lemma.

Lemma A.1 (Stage 1 residual X error)**.** *For any Z-syndrome error $\mathbf{s}_\text{e} \in \mathbb{F}_2^{\tilde{m}}$ during the stage 1 protocol that satisfies $|\mathbf{s}_\text{e}| \leq d/2$, the residual X error $\mathbf{e} + \hat{\mathbf{e}} \in \mathbb{F}_2^{\tilde{n}}$ satisfies*

$$\|\mathbf{e} + \hat{\mathbf{e}}\| \leq f\big(2|\mathbf{s}_\text{e}|\big)\,, \tag{22}$$

where $f(x) = x^3/4$.

Proof. Lemma 5 and Lemma 6[3] of [64] tell us that $\tilde{H}_Z$ of our thickened code (10b) is $(d, x^3/4)$-sound with $d_{\text{ss}} = \infty$ according to Def. 2.2. In the first step of decoding, we find a minimum-weight syndrome correction $\hat{\mathbf{s}}$ such that the total syndrome $\mathbf{s}+\mathbf{s}_\text{e}+\hat{\mathbf{s}} \in \ker \tilde{M}_z$, i.e. it satisfies all metachecks. Since $\mathbf{s} \in \operatorname{im} \tilde{H}_Z \subseteq \ker \tilde{M}_z$, by linearity we also have that $\mathbf{s}_\text{e}+\hat{\mathbf{s}} \in \ker \tilde{M}_z$. Furthermore, because $\mathcal{H}_2 = \ker \tilde{M}_Z / \operatorname{im} \tilde{H}_Z$ is trivial ($d_{\text{ss}} = \infty$), $\mathbf{s}_\text{e} + \hat{\mathbf{s}} \in \operatorname{im} \tilde{H}_Z$, i.e. it is a physical syndrome.

Having established that $\mathbf{s} + \mathbf{s}_\text{e} + \hat{\mathbf{s}}$ is a physical Z-syndrome, we now find a minimum-weight X-correction $\hat{\mathbf{e}}$ such that $\tilde{H}_Z\hat{\mathbf{e}} = \mathbf{s} + \mathbf{s}_\text{e} + \hat{\mathbf{s}}$. Using the fact that $\mathbf{s} = \tilde{H}_Z\mathbf{e}$, we can write

$$\begin{aligned}\mathbf{s}_\text{e} + \hat{\mathbf{s}} &= \mathbf{s} + \tilde{H}_Z\hat{\mathbf{e}} \\ &= \tilde{H}_Z\mathbf{e} + \tilde{H}_Z\hat{\mathbf{e}} \\ &= \tilde{H}_Z\left(\mathbf{e} + \hat{\mathbf{e}}\right)\,.\end{aligned} \tag{23}$$

We will now use the soundness property to upper bound the minimum weight of $\mathbf{e}+\hat{\mathbf{e}}$. Note that since $\hat{\mathbf{s}}$ is a minimum-weight syndrome correction, its weight is at most that of $\mathbf{s}_\text{e}$ because $\hat{\mathbf{s}} = \mathbf{s}_\text{e}$ is a possible solution. Thus, we have

$$|\mathbf{s}_\text{e} + \hat{\mathbf{s}}| \leq |\mathbf{s}_\text{e}| + |\hat{\mathbf{s}}| \leq 2|\mathbf{s}_\text{e}| \leq d\,, \tag{24}$$

[3] Technically, Lemma 6 of [64] involves the homological product of two 3-term chains, whereas we take the product of a 3-term chain with a 2-term chain. However, the one-sided version of the statement (and Lemma 7) still holds upon setting one of the boundary maps of one of the chains (e.g. $\tilde{\delta}_0$) as the zero map, which results in $r_c = 0$ in the proof.

and so $\mathbf{s}_e + \hat{\mathbf{s}}$ lies within the soundness radius of the code. From the definition of soundness (Def. 2.2), there then exists an $\mathbf{e} + \hat{\mathbf{e}}$ that satisfies

$$\|\mathbf{e} + \hat{\mathbf{e}}\| \leq f\big(|\mathbf{s}_e + \hat{\mathbf{s}}|\big) \leq f\big(2|\mathbf{s}_e|\big)\,, \tag{25}$$

where in the last inequality we used the fact that $f(x) = x^3/4$ is convex. □

B Fault-tolerance analysis of stage 2

The idea will be to use the confinement property of constant-rate HGP codes [68] to show that the residual error in each iteration of Algorithm 1 is bounded. The proof utilizes the single-shot capability of confined codes, which has been shown in [65]; we recite the relevant results for completeness.

We employ the Shadow decoder [65] which takes in a parameter $t > 0$ and operates in two steps: given a corrupted syndrome $\mathbf{s} + \mathbf{s}_e$,

1. Find a minimum-weight syndrome correction $\hat{\mathbf{s}}$ such that for the repaired syndrome $\mathbf{s} + \mathbf{s}_e + \hat{\mathbf{s}}$, there exists an error $\mathbf{e}'$ with $\|\mathbf{e}'\| \leq t$ satisfying $H\mathbf{e}' = \mathbf{s} + \mathbf{s}_e + \hat{\mathbf{s}}$.

2. Find a minimum-weight correction $\hat{\mathbf{e}}$ such that $H\hat{\mathbf{e}} = \mathbf{s} + \mathbf{s}_e + \hat{\mathbf{s}}$.

If H exhibits confinement, then the weight of the residual error under the Shadow decoder is bounded, as formalized in the following lemma.

Lemma B.1 (Lemma 1 of [65]). *Consider a parity-check matrix H that is (t, f)-confined. For any error $\mathbf{e}$ with $\|\mathbf{e}\| \leq t/2$ and syndrome error $\mathbf{s}_e$, the residual error $\mathbf{e} + \hat{\mathbf{e}}$ left by the shadow decoder with parameter $t/2$ is bounded by*

$$\|\mathbf{e} + \hat{\mathbf{e}}\| \leq f\big(2|\mathbf{s}_e|\big)\,. \tag{26}$$

The proof of Lemma B.1 is of the same flavor as the one for sound codes. Recall that for stage 2, we reinterpret the bulk X-check Tanner graph as a space-time decoding graph over $d-1$ QEC rounds of the HGP code. In particular, Z errors on sheet qubits denote physical Z errors, and Z errors on intermediate qubits denote syndrome errors. To make this mapping explicit, for a bulk error $\mathbf{e} \in \mathbb{F}_2^{(\tilde{n}-n)}$, denote the partitions

$$E \equiv \mathbf{e}|_{\text{sheets}} \in \mathbb{F}_2^{(d-1)\times n} \quad , \quad S_e \equiv \mathbf{e}|_{\text{int}} \in \mathbb{F}_2^{(d-1)\times m} \tag{27}$$

as matrices for the physical and syndrome errors where the first index runs over the $d-1$ sheets in the bulk. We also construct the matrix $S \in \mathbb{F}_2^{(d-1)\times n}$ similarly but with the X-syndromes. With slight abuse of notation, we will denote $\|E\|$ as the Hamming weight of E when E is the restriction of the reduced error (i.e. minimum-weight representative of $\mathbf{e} + \mathcal{S}_Z$).

A key subroutine of Algorithm 1 involves single-shot decoding of the original HGP code. We will use the fact that constant-rate HGP codes exhibit linear confinement, i.e. $f(x) = \alpha x$ and $t = \gamma\sqrt{n}$, when their underlying Tanner graphs have sufficient expansion [68]. As a result, Lemma B.1 guarantees their single-shot performance under the Shadow decoder. The next lemma says that, if both the physical and syndrome errors in the bulk are sufficiently small, then the residual error on the boundary at the end of stage 2 is bounded.

Lemma B.2 (Stage 2 residual Z error). *Suppose H_X of the HGP code is (t, f)-confined. Given a Z error $\mathbf{e} \in \mathbb{F}_2^{\tilde{n}-n}$ in the bulk whose partitions (27) satisfy*

$$\|E\| \leq \frac{t}{4} \quad , \quad f\big(2|S_e|\big) \leq \frac{t}{4}\,, \tag{28}$$

the residual Z error on the boundary $\mathbf{e}_\partial \in \mathbb{F}_2^n$ at the end of the stage 2 protocol satisfies

$$\|\mathbf{e}_\partial\| \leq t/4\,. \tag{29}$$

Proof. Following Algorithm 1, we define

$$\mathbf{e}^\tau \equiv E_\tau \in \mathbb{F}_2^n\,,\ \ \mathbf{s}^\tau \equiv S_\tau \in \mathbb{F}_2^m\,,\ \ \mathbf{s}_e^\tau \equiv S_{e,\tau} \in \mathbb{F}_2^m \tag{30}$$

as the restrictions of E, S, and S_e to the sheet indexed by $\tau = 1, \ldots, d-1$, where $\tau = 1$ denotes the opposite boundary. Since $\mathbf{e}^\tau$ and $\mathbf{s}^\tau$ are restrictions, we have $\|\mathbf{e}^\tau\| \leq \|E\| \leq t/4$ and $f(|\mathbf{s}_e^\tau|) \leq f(|S_e|) \leq t/4$. Denote $\bar{\mathbf{e}}^\tau$ and $\bar{\mathbf{s}}^\tau$ as the physical error and observed syndrome respectively prior to decoding in step τ of Algorithm 1. The Shadow decoder takes in $\bar{\mathbf{s}}^\tau$ and outputs a syndrome correction $\hat{\mathbf{s}}^\tau \in \mathbb{F}_2^m$. In the next step of the algorithm, this syndrome correction $\hat{\mathbf{s}}^\tau$ is then added to the next syndrome $\mathbf{s}^{\tau+1}$ and its syndrome error $\mathbf{s}_e^{\tau+1}$ to obtain the next observed syndrome $\bar{\mathbf{s}}^{\tau+1} = \hat{\mathbf{s}}^\tau + \mathbf{s}^{\tau+1} + \mathbf{s}_e^{\tau+1}$.

We now proceed to prove (29) by induction. Suppose the residual error at step τ satisfies $\|\bar{\mathbf{e}}^\tau + \hat{\mathbf{e}}^\tau\| \leq t/4$. Then the combined physical error in the next step $\bar{\mathbf{e}}^{\tau+1}$ obeys

$$\begin{aligned}\big\|\bar{\mathbf{e}}^{\tau+1}\big\| &= \big\|\mathbf{e}^{\tau+1} + \bar{\mathbf{e}}^\tau + \hat{\mathbf{e}}^\tau\big\| \\ &\leq \big\|\mathbf{e}^{\tau+1}\big\| + \|\bar{\mathbf{e}}^\tau + \hat{\mathbf{e}}^\tau\| \\ &\leq t/2\,,\end{aligned} \tag{31}$$

which satisfies the assumption of Lemma B.1. Plugging in the result of the lemma (26), we arrive at

$$\big\|\bar{\mathbf{e}}^{\tau+1} + \hat{\mathbf{e}}^{\tau+1}\big\| \leq f\big(2|\mathbf{s}_e^{\tau+1}|\big) \leq f\big(2|S_e|\big) \leq t/4\,. \tag{32}$$

Since the physical error in the initial step ($\tau = 1$ sheet) obeys $\big\|\bar{\mathbf{e}}^1\big\| = \big\|\mathbf{e}^1\big\| \leq \|E\| \leq t/4$, (31) holds for $\tau = 1$, and thus by induction we conclude that

$$\|\mathbf{e}_\partial\| = \big\|\bar{\mathbf{e}}^{d-1} + \hat{\mathbf{e}}^{d-1}\big\| \leq t/4\,. \tag{33}$$

□

C Fault-tolerance analysis of the full protocol

We now combine the results of Appendix A and Appendix B in the following theorem which demonstrates the single-shot capability of the entire protocol.

Theorem C.1 (Full protocol residual error)**.** *Suppose we have a HGP code with (t, f)-confinement and distance $d > t$. Let $\mathbf{s}_e \in \mathbb{F}_2^{\tilde{m}}$ be a Z-syndrome error during stage 1, and let $\tilde{\mathbf{e}} \in \mathbb{F}_2^{\tilde{n}-n}$ be a bulk Z error during stage 2 with partitions (27). If*

$$|\mathbf{s}_e| \leq \frac{d}{2} \quad , \quad \|E\| \leq \frac{t}{4} \quad , \quad f\big(2|S_e|\big) \leq \frac{t}{4} , \tag{34}$$

then the residual X and Z errors $\mathbf{e}_X$ and $\mathbf{e}_Z$ on the HGP code at the end of the protocol satisfy

$$\|\mathbf{e}_X\| \leq 2|\mathbf{s}_e|^3 \quad , \quad \|\mathbf{e}_Z\| \leq \frac{d}{4} . \tag{35}$$

Proof. Since $|\mathbf{s}_e| \leq d/2$, Lemma A.1 tells us that the bulk residual X error $\tilde{\mathbf{e}}_X \in \mathbb{F}_2^{\tilde{n}-n}$ satisfies

$$\|\tilde{\mathbf{e}}_X\| \leq \frac{1}{4}\left(2|\mathbf{s}_e|\right)^3 = 2|\mathbf{s}_e|^3 . \tag{36}$$

Because $\mathbf{e}_X \subset \tilde{\mathbf{e}}_X$ is a restriction onto the boundary, we arrive at the first equation in (35). Since $\|E\|, f\big(2|S_e|\big) \leq t/4$, the second equation in (35) follows from Lemma B.2 and the fact that $d > t$ from the definition of confinement (Def. 2.1). □

In particular, if $2|\mathbf{s}_e|^3 < d/2$, then the residual errors on the boundary are correctable (by minimum-weight decoding).

References

[1] A. Kitaev, Annals of Physics **303**, 2–30 (2003).

[2] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **97**, 180501 (2006).

[3] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Journal of Mathematical Physics **43**, 4452–4505 (2002).

[4] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, New Journal of Physics **14**, 123011 (2012).

[5] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).

[6] A. J. Landahl and C. Ryan-Anderson, Quantum computing by color-code lattice surgery (2014), arXiv:1407.5103 [quant-ph] .

[7] B. J. Brown, K. Laubscher, M. S. Kesselring, and J. R. Wootton, Phys. Rev. X **7**, 021029 (2017).

[8] H. Zhou, C. Zhao, M. Cain, D. Bluvstein, N. Maskara, C. Duckering, H.-Y. Hu, S.-T. Wang, A. Kubica, and M. D. Lukin, Low-overhead transversal fault tolerance for universal quantum computation (2025), arXiv:2406.17653 [quant-ph] .

[9] C. Ryan-Anderson et al., Implementing fault-tolerant entangling gates on the five-qubit code and the color code (2022), arXiv:2208.01863 [quant-ph] .

[10] Google Quantum AI, Nature **614**, 676 (2023).

[11] Dolev Bluvstein et al., Nature **626**, 58–65 (2023).

[12] Google Quantum AI et al., Quantum error correction below the surface code threshold (2024), arXiv:2408.13687 [quant-ph] .

[13] S. Bravyi and B. Terhal, New Journal of Physics **11**, 043029 (2009).

[14] S. Bravyi, D. Poulin, and B. Terhal, Phys. Rev. Lett. **104**, 050503 (2010).

[15] S. Bravyi and R. König, Phys. Rev. Lett. **110**, 170503 (2013).

[16] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **98**, 160502 (2007).

[17] A. Kubica, B. Yoshida, and F. Pastawski, New Journal of Physics **17**, 083026 (2015).

[18] H. Bombín, Phys. Rev. X **5**, 031043 (2015).

[19] A. Kubica and M. Vasmer, Nature Communications **13**, 6272 (2022).

[20] C. Stahl, Phys. Rev. B **110**, 075143 (2024).

[21] D. Gottesman, Stabilizer codes and quantum error correction (1997), arXiv:quant-ph/9705052 [quant-ph] .

[22] N. P. Breuckmann and J. N. Eberhardt, PRX Quantum **2**, 040101 (2021).

[23] M. Freedman, D. Meyer, and F. Luo, Z2-systolic freedom and quantum codes, in *Mathematics of Quantum Computation* (CRC Press, 2002) pp. 287–320.

[24] J.-P. Tillich and G. Zemor, IEEE Transactions on Information Theory **60**, 1193–1202 (2014).

[25] P. Panteleev and G. Kalachev, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022 (Association for Computing Machinery, New York, NY, USA, 2022) p. 375–388.

[26] A. Leverrier and G. Zemor, in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE Computer Society, Los Alamitos, CA, USA, 2022) pp. 872–883.

[27] I. Dinur, M.-H. Hsieh, T.-C. Lin, and T. Vidick, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023 (Association for Computing Machinery, New York, NY, USA, 2023) p. 905–918.

[28] N. Baspin and A. Krishna, Quantum **6**, 711 (2022).

[29] N. Baspin, V. Guruswami, A. Krishna, and R. Li, Quantum Science and Technology **10**, 015021 (2024).

[30] S. Dai and R. Li, in *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25 (Association for Computing Machinery, New York, NY, USA, 2025) p. 677–688.

[31] J. I. Cirac and P. Zoller, Phys. Rev. Lett. **74**, 4091 (1995).

[32] J.-S. Chen, E. Nielsen, M. Ebert, V. Inlek, K. Wright, V. Chaplin, A. Maksymov, E. Páez, A. Poudel, P. Maunz, and J. Gamble, Quantum **8**, 1516 (2024).

[33] S. A. M. et al., Phys. Rev. X **13**, 041052 (2023).

[34] M. Saffman, Journal of Physics B: Atomic, Molecular and Optical Physics **49**, 202001 (2016).

[35] A. Jenkins, J. W. Lis, A. Senoo, W. F. McGrew, and A. M. Kaufman, Phys. Rev. X **12**, 021027 (2022).

[36] S. J. Evered, D. Bluvstein, M. Kalinowski, S. Ebadi, T. Manovitz, H. Zhou, S. H. Li, A. A. Geim, T. T. Wang, N. Maskara, H. Levine, G. Semeghini, M. Greiner, V. Vuletić, and M. D. Lukin, Nature **622**, 268–272 (2023).

[37] Y. Hong, E. Durso-Sabina, D. Hayes, and A. Lucas, Phys. Rev. Lett. **133**, 180601 (2024).

[38] N. Berthusen, J. Dreiling, C. Foltz, J. P. Gaebler, T. M. Gatterman, D. Gresh, N. Hewitt, M. Mills, S. A. Moses, B. Neyenhuis, P. Siegfried, and D. Hayes, Phys. Rev. A **110**, 062413 (2024).

[39] B. W. Reichardt, D. Aasen, R. Chao, A. Chernoguzov, W. van Dam, J. P. Gaebler, D. Gresh, D. Lucchetti, M. Mills, S. A. Moses, B. Neyenhuis, A. Paetznick, A. Paz, P. E. Siegfried, M. P. da Silva, K. M. Svore, Z. Wang, and M. Zanner, Demonstration of quantum computation and error correction with a tesseract code (2024), arXiv:2409.04628 [quant-ph] .

[40] O. Fawzi, A. Grospellier, and A. Leverrier, in *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, Vol. 14 (IEEE, 2018) p. 743–754.

[41] Y. Hong, J. Guo, and A. Lucas, Nature Communications **16**, 10.1038/s41467-024-55570-7 (2025).

[42] A. G. Manes and J. Claes, Quantum **9**, 1618 (2025).

[43] A. Krishna and D. Poulin, Phys. Rev. X **11**, 011023 (2021).

[44] N. P. Breuckmann and S. Burton, Quantum **8**, 1372 (2024).

[45] A. O. Quintavalle, P. Webster, and M. Vasmer, Quantum **7**, 1153 (2023).

[46] Y. Hong, M. Marinelli, A. M. Kaufman, and A. Lucas, Phys. Rev. A **110**, 022607 (2024).

[47] D. Gottesman, Quantum Info. Comput. **14**, 1338–1372 (2014).

[48] Q. Xu, J. P. Bonilla Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasić, M. D. Lukin, L. Jiang, and H. Zhou, Nature Physics **20**, 1084 (2024).

[49] H. Bombín, New Journal of Physics **18**, 043038 (2016).

[50] B. J. Brown, Science Advances **6** (2020).

[51] T. R. Scruby, D. E. Browne, P. Webster, and M. Vasmer, Quantum **6**, 721 (2022).

[52] R. Raussendorf, S. Bravyi, and J. Harrington, Phys. Rev. A **71**, 062313 (2005).

[53] S. Bravyi, D. Gosset, R. König, and M. Tomamichel, Nature Physics **16**, 1040–1045 (2020).

[54] T. R. Scruby, M. Vasmer, and D. E. Browne, Phys. Rev. Res. **4**, 043052 (2022).

[55] O. Higgott and N. P. Breuckmann, PRX Quantum **4**, 020332 (2023).

[56] R. Gallager, IRE Transactions on Information Theory **8**, 21 (1962).

[57] H. Bombin and M. A. Martin-Delgado, Journal of Mathematical Physics **48** (2007).

[58] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).

[59] A. Steane, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452**, 2551 (1996).

[60] M. H. Freedman and M. B. Hastings, Quantum Info. Comput. **14**, 144–180 (2014).

[61] S. Bravyi and M. B. Hastings, in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14 (Association for Computing Machinery, New York, NY, USA, 2014) p. 273–282.

[62] J. Mosheiff, N. Resch, N. Ron-Zewi, S. Silas, and M. Wootters, SIAM Journal on Computing , FOCS20 (2021).

[63] P. W. Shor, Phys. Rev. A **52**, R2493 (1995).

[64] E. T. Campbell, Quantum Science and Technology **4**, 025006 (2019).

[65] A. O. Quintavalle, M. Vasmer, J. Roffe, and E. T. Campbell, PRX Quantum **2**, 020340 (2021).

[66] A. Anshu, N. P. Breuckmann, and C. Nirkhe, in *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC '23, Vol. 5 (ACM, 2023) p. 1090–1096.

[67] E. Ben-Sasson, V. Guruswami, T. Kaufman, M. Sudan, and M. Viderman, in *2009 24th Annual IEEE Conference on Computational Complexity* (2009) pp. 52–61.

[68] A. Leverrier, J.-P. Tillich, and G. Zemor, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (IEEE, 2015).

[69] Q. Xu, H. Zhou, G. Zheng, D. Bluvstein, J. P. B. Ataides, M. D. Lukin, and L. Jiang, Phys. Rev. X **15**, 021065 (2025).

[70] T. Richardson and R. Urbanke, *Modern coding theory* (Cambridge University Press, 2008).

[71] M. B. Hastings, Quantum Info. Comput. **17**, 1307–1334 (2017).

[72] S. Evra, T. Kaufman, and G. Zémor, 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS) , 218 (2020).

[73] W. Zeng and L. P. Pryadko, Phys. Rev. Lett. **122**, 230501 (2019).

[74] A. Bolt, G. Duclos-Cianci, D. Poulin, and T. M. Stace, Phys. Rev. Lett. **117**, 070501 (2016).

[75] A. Bolt, D. Poulin, and T. M. Stace, Phys. Rev. A **98**, 062302 (2018).

[76] D. Bacon, S. T. Flammia, A. W. Harrow, and J. Shi, in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15 (Association for Computing Machinery, New York, NY, USA, 2015) p. 327–334.

[77] D. Gottesman, Opportunities and challenges in fault-tolerant quantum computation (2022), arXiv:2210.15844 [quant-ph] .

[78] N. Delfosse and A. Paetznick, Spacetime codes of clifford circuits (2023), arXiv:2304.05943 [quant-ph] .

[79] E. Sabo, L. G. Gunderman, B. Ide, M. Vasmer, and G. Dauphinais, PRX Quantum **5**, 040302 (2024).

[80] T. Etzion, A. Trachtenberg, and A. Vardy, IEEE Transactions on Information Theory **45**, 2173 (1999).

[81] E. Berlekamp, R. McEliece, and H. van Tilborg, IEEE Transactions on Information Theory **24**, 384 (1978).

[82] J. Roffe, LDPC: Python tools for low density parity check codes (2022).

[83] P. Panteleev and G. Kalachev, Quantum **5**, 585 (2021).

[84] J. Roffe, D. R. White, S. Burton, and E. Campbell, Phys. Rev. Res. **2**, 043423 (2020).

[85] S. Wolanski and B. Barber, Ambiguity clustering: an accurate and efficient decoder for qldpc codes (2024), arXiv:2406.14527 [quant-ph] .

[86] T. Hillmann, L. Berent, A. O. Quintavalle, J. Eisert, R. Wille, and J. Roffe, Nature Communications **16**, 10.1038/s41467-025-63214-7 (2025).

[87] A. deMarti iOlius, I. E. Martinez, J. Roffe, and J. E. Martinez, An almost-linear time decoding algorithm for quantum ldpc codes under circuit-level noise (2024), arXiv:2409.01440 [quant-ph] .

[88] T. Rakovszky and V. Khemani, The physics of (good) ldpc codes i. gauging and dualities (2023), arXiv:2310.16032 [quant-ph] .

[89] I. Dinur, S. Evra, R. Livne, A. Lubotzky, and S. Mozes, Locally testable codes with constant rate, distance, and locality, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, USA, 2022) p. 357–374.

[90] S. Gu, E. Tang, L. Caha, S. H. Choe, Z. He, and A. Kubica, Communications in Mathematical Physics **405**, 85 (2024).

[91] A. A. Kovalev and L. P. Pryadko, Phys. Rev. A **87**, 020304 (2013).

[92] S. J. S. Tan and L. Stambler, Effective distance of higher dimensional hgps and weight-reduced quantum ldpc codes (2024), arXiv:2409.02193 [quant-ph] .

[93] S. Burton and D. Browne, IEEE Transactions on Information Theory **68**, 1772 (2022).

[94] M. Vasmer and D. E. Browne, Phys. Rev. A **100**, 012312 (2019).