Constant-Overhead Addressable Gates via Single-Shot Code Switching

Louis Golowich
UC Berkeley & IBM Quantum
lgolowich@berkeley.edu

Kathleen (Katie) Chang
Yale University & IBM Quantum
katie.chang@yale.edu

Guanyu Zhu
IBM Quantum
Guanyu.Zhu@ibm.com

October 9, 2025

Abstract

It is a major challenge to perform addressable and parallel logical operations on constant-rate quantum LDPC (qLDPC) codes. Indeed, the overhead of targeting specific logical qubits represents a crucial bottleneck in many quantum fault-tolerance schemes.

We introduce fault-tolerant protocols for performing various addressable as well as parallel logical operations with constant space-time overhead, on a family of constant-rate and polynomial-distance qLDPC codes. Specifically, we construct gadgets for a large class of permutations of logical qubits. We apply these logical permutations to construct gadgets for applying a targeted Hadamard (or $CNOT$) gate on any chosen logical qubit (pair). We also construct gadgets for preparing logical code states, and for applying Hadamard gates on all logical qubits in a codeblock. All of our gadgets use constant quantum space-time overhead along with polynomially bounded classical computation. Prior protocols for such operations required larger overhead, or else relied on codes with certain symmetries that lack known asymptotic constructions.

Our codes are given by tensor (i.e. hypergraph) products of classical codes constructed from lossless expander graphs. Our core technical contribution is a constant-overhead code-switching procedure between 2- and 3-dimensional product codes, which generalizes Bombín's dimensional jump (arXiv:1412.5079). We provide rigorous fault-tolerance proofs for our gadgets, and specifically prove a constant threshold under locally stochastic noise. Along the way, we develop a small-set flip decoder for high-dimensional product codes from lossless expanders. Our techniques yield additional interesting consequences, such as single-shot state preparation of 2-dimensional product codes with constant space-time overhead. We also propose a method for performing parallel non-Clifford gates by extending our techniques to codes supporting transversal application of such gates.

Contents

1 Introduction

Constant-rate quantum LDPC (qLDPC) codes provide fault-tolerant quantum memories with just constant space overhead. However, it remains a major challenge to efficiently perform logical computations on the encoded data. Existing approaches typically incur large (growing in the block length) space-time overheads for such gates that perform highly targeted or parallel operations on specific logical qubits within a codeblock.

In this work, we construct protocols for performing both addressable (i.e. targeted) and parallel fault-tolerant logical operations with constant space-time overhead, on constant-rate codes. Constant overhead here means that the number of physical qubits and physical timesteps are within a constant factor of the number of logical qubits and logical timesteps, respectively. Specifically, we show the following:

Theorem 1.1 (Informal)**.** [1] *There exists an infinite family of*[2] $[[n,\ k = \Theta(n),\ d = \mathrm{poly}(n)]]$ *qLDPC codes for which the following logical operations can be performed in constant quantum space-time overhead, while using polynomially bounded noiseless classical computation. These constructions exhibit a threshold under locally stochastic noise.*

1. *Initialize a new codeblock with all logical qubits in the* $|0\rangle$ *or* $|+\rangle$ *state.*
2. *Permute the logical qubits in a codeblock according to an appropriate permutation* $\pi : [k] \to [k]$*, meaning that logical qubit* j *is moved to* $\pi(j)$*. For instance:*
   (a) *For every* $s \in [k]$*, we can choose* π *be the cyclic shift* $\pi(j) \equiv j + s \pmod{k}$*.*
   (b) *For every constant* $r \geq 3$*, we can set* $k = \ell^r$ *for* $\ell \in \mathbb{N}$*. Then for every* $h \in [r]$ *and every permutation* $\sigma : [\ell] \to [\ell]$ *we can choose* $\pi = I^{\otimes h-1} \otimes \sigma \otimes I^{r-h}$ *to be the permutation on* $[\ell]^r \cong [k]$ *that applies* σ *to the* h^{th} *coordinate.*
3. *Perform logical Hadamard gates on all logical qubits in a codeblock.*
4. *Perform a targeted logical Hadamard gate on any single logical qubit.*
5. *Perform a targeted logical CNOT gate between any pair of logical qubits within a codeblock, or across two codeblocks.*

Remark 1.2. *Our qLDPC codes in Theorem 1.1 are actually capable of encoding a larger number* $k' = O(k)$ *of logical qubits. However, due to the structure of our fault-tolerance scheme, we only encode our message into a fixed subset containing* k *of these possible* k' *logical qubits.*

To our knowledge, our construction is the first achieving constant space-time overhead fault-tolerant (i.e. exhibiting a threshold) gadgets for items 1–5 in Theorem 1.1. Many prior proposals for performing the fault-tolerant logical operations described in Theorem 1.1 require space-time overhead growing polynomially in the code distance (or the protocol's fault distance) d, or else require specific code properties that are difficult to achieve with good parameters.

For instance, standard approaches for the state preparation in item 1 incur polynomial overheads from repeated measurements (in space or time) or concatenation; see e.g. [Got14, BL25]. While

[1] This theorem statement combines the gadgets in Proposition 7.1 and Proposition 5.1 (for item 1), Corollary 10.3 (for item 2a), Proposition 10.2 (for item 2b), Proposition 10.4 (for item 3), and Corollary 10.6 (for item 4 and item 5). By Lemma 3.29 and Lemma 3.31, the threshold for these gadgets follows by combining the results listed above with the error-correction gadget in Proposition 8.1.

[2] Recall that an $[[n, k, d]]$ code encodes k logical qubits into n physical qubits with distance d.

it was previously known that such state preparation can be performed with constant overhead using asymptotically good $[[n, \Theta(n), \Theta(n)]]$ quantum locally testable codes (qLTCs) [Pat24], no such codes have yet been constructed. A state preparation scheme with low (though still super-constant) overhead was given by [NP25] using nearly good qLTCs [DLV24, KP25]. In contrast, we achieve the constant-overhead state preparation in item 1 for codes as simple as (the disjoint union of multiple copies of) 2-dimensional hypergraph product codes [TZ14]; see Section 1.1 below.

Prior approaches for the constant-overhead cyclic permutations in item 2a used codes with strong symmetry properties for which good asymptotic families are not known [XZZ+24, Section A.5]. Perhaps surprisingly, we obtain such cyclic permutations, as well as the more general permutations in item 2b, on families of qLDPC codes without any special symmetry properties. Similarly, for product codes with an appropriate symmetry, the fold-transversal Hadamard gate [BB24, QWV23] achieves item 3 up to some logical swaps, which then may require polynomial time to revert (e.g. [XZZ+24, Section A.5]). We avoid this polynomial overhead, as well as the symmetry assumption.

For the targeted logical Clifford gates in item 4 and item 5, prior approaches such as qLDPC-code lattice surgery (e.g. [CKBB22, WY24, CHRY24, SJOY25, HCWY25], which can be viewed as an application of stabilizer-weight reduction [Has23]) or concatenation [Got14] naively incur a $\mathrm{poly}(d)$ time overhead. As described in [Got14], this time overhead can be reduced by encoding the k logical qubits into $\mathrm{poly}(k)$ smaller codeblocks, and then trading off time for space when performing gates on these smaller codeblocks. Such techniques may yield similar results as item 4 and item 5 in Theorem 1.1, though with some caveats. For instance, [Got14] still requires super-constant time overhead for preparing certain logical ancilla states.[3] Meanwhile, to our knowledge it remains an open question to construct an asymptotically efficient decoder for qLDPC-code lattice surgery schemes. In contrast, all of our gadgets use just polynomial-sized classical computation, including the cost of decoding.

[NP25] provide gadgets for targeted logical gates that have a low space-time overhead in an amortized sense, meaning that the cost per gate is lower when many gates are performed in parallel. However, even this amortized overhead is super-constant, while the non-amortized overhead to perform a single targeted gate is at least polynomial in the distance. Furthermore, [NP25] use somewhat involved distillation techniques within nearly good quantum locally testable codes. In contrast, we use more elementary codes; see Section 1.1 below.

While [HVWZ25b, HVWZ25a] construct asymptotically good codes with fully addressable and transversal non-Clifford (as well as Clifford) gates, their codes are not LDPC, but rather have linear-weight stabilizers. Therefore additional techniques such as concatenation will be needed to perform fault-tolerant syndrome extraction and error correction with these codes, which introduces a growing space-time overhead.

1.1 Product Codes with Single-Shot Code Switching

Our codes in Theorem 1.1 are given by tensor products of (co)chain complexes associated to lossless expander graphs, which in turn can be constructed randomly (e.g. [HLW06, Theorem 4.16]) or explicitly [HLM+25] (see Section 3.3). Two-dimensional such product codes are often called hypergraph product codes [TZ14]. The key technical ingredient behind Theorem 1.1 is a *single-shot code switching* procedure that we provide, which allows us to fault-tolerantly switch between product codes of different dimensions in constant space-time overhead, while preserving the encoded logical

[3] It may be possible to reduce this time overhead in [Got14] to constant for the special case of Clifford circuits; it is an interesting direction of future work to further develop such optimizations and compare them to our methods.

qubits. Specifically, for arbitrary fixed $r \geq 3$, the gadgets in Theorem 1.1 are based on (sometimes repeated) switching between an r-dimensional $[[\Theta(n^r), \Theta(n^r), \Omega(n)]]$ product code and a collection of $\Theta(n)$ copies of an $(r-1)$-dimensional $[[\Theta(n^{r-1}), \Theta(n^{r-1}), \Omega(n)]]$ product code. For item 1 in Theorem 1.1, we also develop a gadget for single-shot state preparation, which shares many technical ingredients with the code switching.

Our single-shot code switching procedure, which we prove exhibits a threshold under locally stochastic noise, has many interesting consequences. For instance, Section 1.2 below describes how we can use such code switching to target individual logical qubits. Code switching also allows us to prove Theorem 1.1 for the code given by $\Theta(\sqrt{n})$ disjoint copies of a $[[n, \Theta(n), \Theta(\sqrt{n})]]$ 2-dimensional product code. In particular, we prove item 1 by first preparing logical $|0\rangle$ or $|+\rangle$ states in a 3-dimensional product code, and then switching down to a collection of 2-dimensional codeblocks. Yet such 2-dimensional codes typically do not support constant-overhead logical state preparation, at least when preparing a single codeblock. It is therefore perhaps surprising that we are able to circumvent this barrier by preparing $\Theta(\sqrt{n})$ codeblocks at once.

Our code switching procedure can be viewed as a generalization of Bombin's code switching, or "dimensional jump," on color/toric codes ([Bom16], see also [Bom15a]). Whereas a toric code is a product of classical repetition codes and therefore has poor rate, we obtain constant space-time overhead gadgets by taking products of constant-rate classical LDPC codes.

However, the fundamental idea is similar: to switch down from an r-dimensional codeblock to a collection of $(r-1)$-dimensional codeblocks, we simply measure out a subset of the physical qubits, then we run error correction on this measurement outcome, based on which we apply an appropriate Pauli correction. To switch back up to the r-dimensional code, we run a logical teleportation circuit, which requires logical bell pairs between r- and $(r-1)$-dimensional codes. To construct such logical bell pairs, we first prepare two r-dimensional codeblocks, one with logical $|0\rangle$ states and one with logical $|+\rangle$ states. We then apply our downwards switching to one of the codeblocks, and apply transversal $CNOT$ gates from the $|+\rangle$ block to the $|0\rangle$ block. Note that this transversal $CNOT$ is applied between an $(r-1)$-dimensional code and an r-dimensional code (see Lemma 9.2).

1.2 Constant-Overhead Targeted Gates via Code Switching

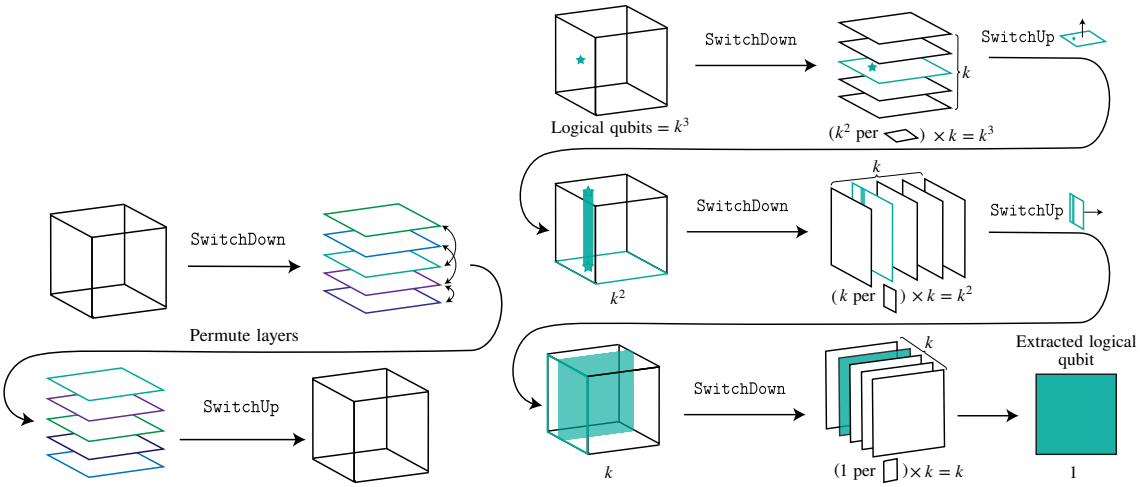
We now outline how we use code switching to construct the gadgets in Theorem 1.1. For illustrative purposes here we consider the $r = 3$ case; the generalization to arbitrary constant $r \geq 3$ is straightforward.

Permuting logical qubits. Figure 1a illustrates how we obtain item 2b of Theorem 1.1 by switching down, permuting the resulting 2-dimensional codeblocks, and then switching back up.

We then obtain item 2a by applying item 2b in all r directions. Specifically, letting $k = \ell_1 \ell_2 \cdots \ell_r$ for relatively prime $\ell_1, \ldots, \ell_r \in \mathbb{N}$, then $\mathbb{Z}_k \cong \mathbb{Z}_{\ell_1} \times \cdots \times \mathbb{Z}_{\ell_r}$. Thus cyclic shifts of k logical qubits can be realized by arranging the qubits in an (r-dimensional) $\ell_1 \times \cdots \times \ell_r$ hypercube, and then performing cyclic shifts in each of the r directions.

Parallel logical Hadamard. For item 3 in Theorem 1.1, recall that applying transversal Hadamard gates across all physical qubits in a CSS code always induces logical Hadamard gates on all logical qubits, but also switches to the "dual" code, meaning that the X and Z stabilizers are swapped. We then use our code switching gadget (iteratively in each of the r directions) to switch back to the original code.

Targeting a single logical qubit in constant space and time. Figure 1b illustrates how we can begin with k^3 logical qubits encoded into a 3-dimensional $[[\Theta(k^3), k^3, \Omega(k)]]$ codeblock, and



(a) Gadget used in item 2 of Theorem 1.1 to permute "slabs" of logical qubits.

(b) Gadget used in item 4 and item 5 of Theorem 1.1 to target a logical qubit.

Figure 1: Applications of code switching to gadgets in Theorem 1.1.

then perform code switching in each of the 3 directions interspersed with swap gates to extract any single desired logical qubit. Specifically, we begin with a 3-dimensional $k \times k \times k$ cube of logical qubits. By switching down and back up in the first direction, we are able to extract a $k \times k$ plane containing the desired logical qubit, which we encode into an ancilla 3-dimensional codeblock. Repeating this procedure in the other two directions, we ultimately extract a codeblock with a single logical qubit. We can then perform transversal $CNOT$ or Hadamard gates on this extracted qubit, before reverting the extraction procedure to insert the qubit back into its original location.

As each code switching step takes constant time, the entire procedure takes constant time. Although we extract one logical qubit into its own poly(k)-sized codeblock, the scheme retains constant rate (i.e. constant space overhead) because we do not parallelize the gadget. That is, we always have k^3 logical qubits collectively encoded into $\Theta(k^3)$ physical qubits. Thus we obtain item 4 and item 5 of Theorem 1.1.

1.3 Fault-Tolerance Analysis and Decoder

An additional contribution of our paper lies in our rigorous proofs of fault-tolerance. Recall that a noise distribution is *ϵ-locally stochastic* if the probability that a set S of physical qubits[4] lies within the support of the corruption is $\leq \epsilon^{|S|}$. As mentioned above, we prove that our gadgets exhibit fault-tolerance properties that imply a threshold under ϵ-locally stochastic noise for sufficiently small constant $\epsilon > 0$ (see Lemma 3.29 and Lemma 3.31). That is, under such physical noise, our logical error rate decays as $e^{-\text{poly}(N)}$, where N is the number of physical qubits. We prove such a threshold by adapting ideas from [KP13, Got14] to argue that errors cannot be highly concentrated inside appropriately-sized connected subgraphs of a constant-degree graph associated to our code.

We prove this threshold behavior under a new small-set flip decoder for high-dimensional product codes from lossless expanders. To construct this decoder, which uses constant quantum time

[4]We will not distinguish between physical qubit and measurement errors, as a measurement error is indistinguishable from qubit errors before and after the measurement.

and linear classical time, we generalize the decoder for 2-dimensional codes of [LTZ15, FGL20], while also applying ideas from [DHLV23, DLV24, KP25]. An informal version of our decoding result is stated below.

Proposition 1.3 (Informal statement of Proposition 4.6 and Proposition 8.1)**.** *For arbitrary constant $r \geq 2$, let Q be a $[[\Theta(n^r), \Theta(n^r), \Omega(n)]]$ qLDPC code associated to an r-dimensional tensor product of n-vertex lossless expanders. Then there exists a decoder for Q that uses constant quantum time along with $O(n^r)$ classical time, which provides:*

- *correction of up to δn adversarial errors for a sufficiently small constant $\delta > 0$, and*
- *a threshold (i.e. logical error rate $\leq e^{-\operatorname{poly}(n)}$) under ϵ-locally stochastic noise for a sufficiently small constant $\epsilon > 0$.*

1.4 Adding Magic to Achieve Universality

While the results listed above provide constant-overhead gadgets for logical Clifford circuits, we can extend our techniques to obtain a scheme for universal fault-tolerant computation. For instance, we may add in a gadget for magic state distillation. Such gadgets with low (but super-constant) space-time overhead have previously been constructed, e.g. in [NP25]. We could alternatively try to apply our code switching gadgets to qLDPC codes supporting transversal non-Clifford gates [BMD07a, BMD07b, Bom15b, ZSP+23, Lin24, GL25, BDET24, Zhu25].

We expand upon the latter approach in Appendix D by proposing a method for parallelizable logical non-Clifford gates using the codes of [GL25, Zhu25]. These codes support a transversal (i.e. constant-depth) physical circuit inducing polynomially many logical CCZ gates in a codeblock. However, we leave the problem of proving a threshold for single-shot code switching on such codes for future work; we discuss the challenges involved in Appendix D. The independent and concurrent work of [THL+25] provides more results in this direction, for a class of codes based on those of [Zhu25].

2 Technical Overview of Construction

In this section, we provide an overview of the techinques underlying the fault-tolerant gadgets we use to prove Theorem 1.1, and we outline how the remainder of the paper is organized.

2.1 Code Construction

We first describe the codes we use to prove Theorem 1.1. For this purpose, we will need the notion of *(co)chain complexes* and their products.

Definition 2.1 (Informal statement of Definition 3.11 and Definition 3.13)**.** *A r-dimensional* cochain complex $\mathcal{C}^*$ over $\mathbb{F} + 2$ *is a sequence*

$$\mathcal{C}^* = (C^0 \xrightarrow{\delta_0} C^1 \xrightarrow{\delta_1} \cdots \xrightarrow{\delta_{r-1}} C^r)$$

of $\mathbb{F}_2$-vector spaces $\mathcal{C}^i$ and coboundary maps δ_i *satisfying $\delta_i \delta_{i-1} = 0$.*

For cochain complexes $\mathcal{A}^*, \mathcal{B}^*$ of respective dimensions $r^{\mathcal{A}}, r^{\mathcal{B}}$, the tensor product $\mathcal{C}^* = \mathcal{A}^* \otimes \mathcal{B}^*$ *is the chain complex of dimension $r^{\mathcal{C}} = r^{\mathcal{A}} + r^{\mathcal{B}}$ given by $\mathcal{C}^i = \bigoplus_{j \in \mathbb{Z}} \mathcal{A}^j \otimes \mathcal{B}^{i-j}$ and $\delta^{\mathcal{C}} = \delta^{\mathcal{A}} \otimes I + I \otimes \delta^{\mathcal{B}}$.*

Recall that a length-n quantum CSS code consists of a pair $Q = (Q_X, Q_Z)$ of classical linear codes $Q_X, Q_Z \subseteq \mathbb{F}_2^n$ satisfying the orthogonality condition $Q_X^\perp := \{y \in \mathbb{F}_2^n : x \cdot y = 0 \ \forall x \in Q_X\} \subseteq Q_Z$. Therefore for an r-dimensional cochain complex $\mathcal{C}^*$, for every level $1 \leq i \leq r-1$ we have a naturally associated CSS code given by $Q_X = \ker(\delta_{i-1}^\top)$ and $Q_Z = \ker(\delta_i)$. Here the CSS orthogonality condition is equivalent to the cochain complex condition $\delta_i \delta_{i-1} = 0$.

We prove Theorem 1.1 using products of 1-dimensional cochain complexes that arise from a certain type of graph called a *lossless expander*. Random bounded-degree graphs are lossless expanders with high probability, and explicit constructions are also known (see Section 3.3). In particular, for fixed $r \in \mathbb{N}$ and for every $h \in [r]$, we let

$$\mathcal{C}^{(h)^*} = \left(\mathbb{F}_2^{V_L^{(h)}} \xrightarrow{\delta^{(h)}} \mathbb{F}_2^{V_R^{(h)}} \right)$$

be a 1-dimensional cochain complex associated to a bounded-degree bipartite lossless expander $G^{(h)} = (V^{(h)} = V_L^{(h)} \sqcup V_R^{(h)}, E^{(h)})$. The coboundary map $\delta^{(h)}$ is given by the bipartite adjacency matrix of $G^{(h)}$. We will choose such graphs for which all $|V^{(h)}|$ are the same. The quantum codes we use are those associated to some level $1 \leq i \leq r-1$ of the product cochain complex

$$\mathcal{C}^* = \mathcal{C}^{(1)^*} \otimes \cdots \otimes \mathcal{C}^{(r)^*}.$$

Letting $n = |V^{(h)}|$ grow for fixed $r = O(1)$, then these codes are qLDPC with parameters $[[\Theta(n^r), \Theta(n^r), \Omega(n)]]$. The specific details of our instantiation are given in Section 10.1.

The $r = 2$-dimensional case of these codes were introduced by [TZ14], where they were called "hypergraph product codes." A key insight in our paper is that we can efficiently switch between product codes of different dimensions r.

2.2 Error Correction, State Preparation, and Code Switching from Small-Set Flip Decoder

The fault-tolerant procedures in Theorem 1.1 are primarily based on three new gadgets that we construct for the product codes described in Section 2.1. These gadgets, which perform error correction, state preparation, and code switching, respectively, all crucially rely on a new decoder that we develop for the codes in Section 2.1. Specifically, in Section 4, we construct a generalization of the *small-set flip decoder*, which was originally introduced for $r = 2$-dimensional tensor product codes [LTZ15, FGL20], to arbitrary $r \geq 2$.

Below, we define such small-set flip decoders, and state our result constructing them for product codes. Here we restrict attention to cochain complexes $\mathcal{C}^*$ with a fixed basis C^i for each $\mathcal{C}^i$, so that $\mathcal{C}^i = \mathbb{F}_2^{C^i}$. We endow $C^0 \sqcup \cdots \sqcup C^r$ with a partial order by letting $c^i \prec c^{i+1}$ for $c^i \in C^i$, $c^{i+1} \in C^{i+1}$ if $c^{i+1} \in \mathrm{supp}(\delta_i(\mathbb{1}_{c^i}))$, and then letting $c^i \prec c^j$ if $c^i \prec c^{i+1} \prec \cdots \prec c^{j-1} \prec c^j$ for some $c^{i+1} \in C^{i+1}, \ldots, c^{j-1} \in C^{j-1}$. We extend this notation to $c^i \in C^i$ and $c^j \in \mathcal{C}^j = \mathbb{F}_2^{C^h}$ by writing $c^i \preceq c^j$ if every $c' \in \mathrm{supp}(c^j)$ satisfies $c^i \preceq c'$. The cochain complexes we consider below have constant locality, meaning that for every $c^i \in C^i$, there are only a constant number of basis elements c' satisfying $c^i \preceq c'$.

Definition 2.2 (Abridged statement of Definition 4.1)**.** *For $m \in \mathbb{N}$, we say a cochain complex $\mathcal{C}^*$ has a m-*small-set error-flip decoder at level i *if for every nonzero $e \in \mathcal{C}^i$ of weight $|e| \leq m$, there exists a basis element $c^0 \in C^0$ for which at least one of the following holds:*

1. *There exists a cochain $c^{i-1} \in \mathcal{C}^{i-1}$ with $c^0 \preceq c^{i-1}$ such that $|e + \delta(c^{i-1})| < |e|$, or*

2. *There exists a cochain* $c^i \in \mathcal{C}^i$ *with* $c^0 \preceq c^i$ *such that* $|\delta(e + c^i)| < |\delta(e)|$.

Proposition 2.3 (Informal statement of Proposition 4.6)**.** *The cochain complex* $\mathcal{C}^* = {\mathcal{C}^{(1)}}^* \otimes \cdots \otimes {\mathcal{C}^{(r)}}^*$ *in Section 2.1 has an* m*-small-set flip decoder at every level* $0 \leq i \leq r-1$ *for* $m = \Theta(|V^{(h)}|)$.[5]

To generalize the 2-dimensional argument of [LTZ15, FGL20] to the higher-dimensional case in Proposition 2.3, we adapt techniques used by [DLV24, KP25] to construct quantum locally testable codes. Specifically, [DLV24, KP25] (see also [NP25]) proved properties similar to small-set flip decodability for high-dimensional product codes that impose certain local codes exhibiting a *robustness* property on a global *spectral expander graph*. As our codes in Section 2.1 are constructed from lossless expanders instead of spectral expanders, we apply expansion in a different way, but we ultimately still leverage the robustness of local codes, which in our case are simply repetition codes (see Lemma 4.8).

We apply this small-set flip decoder for two purposes, namely to perform error correction, and to show a sort of "small-set soundness" for high-dimensional product codes. For the error correction application, we simply take e taken to be a low-weight error on a code state, so that Definition 2.2 gives a correction c^i that reduces the syndrome weight. Our error correction gadget in Section 8 simply repeatedly applies such corrections while they exist. We generalize the percolation-based arguments in [FGL20] to the high-dimensional case to show that this error correction succeeds even in the presence of random data and syndrome errors. We similarly apply such error correction within our state preparation gadget in Section 7, our code switching gadget in Section 5, and our measurement gadget in Section 9.3.

Meanwhile, for the "small-set soundness" application, we apply Definition 2.2 to a low-weight *syndrome* $e = \delta(f)$ for some $f \in C^{i-1}$. Intuitively, because $\mathcal{C}^*$ has constant locality, then by repeatedly applying Definition 2.2 to such an e, we can conclude that every sufficiently low-weight syndrome e arises from some error f of proportionally low weight $O(|e|)$. Note that this property is stronger than the "small-set coboundary expansion" property (see e.g. [DHLV23, HL22]), which only requires that low-weight f have sufficiently high-weight syndromes $e = \delta_{i-1}(f)$ so that $|f| \leq O(|e|)$.

We crucially apply this small-set soundness property in our state preparation gadget in Section 7. This gadget prepares logical code states by preparing physical $|0^n\rangle$ or $|+^n\rangle$, and then measuring all the code stabilizers. We then run error correction on these measurement outcomes to compute an estimate $\tilde{s}$ of the syndrome, and then compute some error $\tilde{f}$ with $\delta(\tilde{f}) = \tilde{s}$, and apply a correction based on $\tilde{f}$. We want to show that $\tilde{f}$ is close to the true error f on the code state, given that $\tilde{s} = \delta(\tilde{f})$ is close to the true syndrome $s = \delta(f)$. Small-set soundness precisely ensures this property; we also again apply a percolation argument to deal with large numbers of random errors.

In our code switching gadget outlined in Section 1.1 above, we similarly apply small-set soundness to argue that the error-corrected measurement outcomes lead to an accurate Pauli correction.

2.3 Fault-Tolerance Analysis

A significant technical contribution of our work lies in our proof of threshold under a constant rate of locally stochastic noise. Although our codes have sublinear distance, we are still able to handle such random linear-weight errors using percolation arguments adapted from [KP13, Got14]. Specifically, if our logical qubits are encoded in a CSS code associated to a cochain complex $\mathcal{C}^*$, we track the propagation of errors through a *connectivity graph* $G^{\mathcal{C}}$ associated to $\mathcal{C}^*$. Formally, the vertices of this graph are basis elements in $C^0 \sqcup \cdots \sqcup C^r$, and an edge connects every c, c' for which

[5] Recall here that all $h \in [r]$ have the same $|V^{(h)}|$.

either there is some $c^0 \in C^0$ with $c^0 \preceq c, c'$, or some $c^r \in C^r$ with $c, c' \preceq c^r$ (see Definition 3.32). We then argue that under faults that are not too dense within any large connected subgraph of $G^{\mathcal{C}}$, our gadgets preserve such a low density of errors within connected subgraphs (see Definition 3.30). By standard percolation arguments, locally stochastic noise will indeed have low density within such large connected subgraphs (see Lemma 3.31). We furthermore apply a standard decomposition to reduce general noise to Pauli noise (see Lemma 3.29).

Our fault-tolerance proofs crucially rely on the fact that the small-set flip decoder described in Section 2.2 by definition performs updates within constant-sized neighborhoods of the connectivity graph $G^{\mathcal{C}}$. Therefore both when performing error correction and applying small-set soundness to compute Pauli corrections (see Section 2.2), we control the propagation of errors within connected subgraphs of $G^{\mathcal{C}}$. An interesting consequence of this argument is that our proof may not immediately extend to more global decoders, such as a maximum-liklihood decoder that computes the most likely error.

2.4 Roadmap

The remainder of this paper is organized as follows. Section 3 provides necessary preliminary notions and prior results. In Section 4, we present our small-set flip decoder. We present our gadgets for downwards and upwards code switching in Section 5 and Section 6, respectively. We postpone the proof for the upwards code switching gadget to Appendix C, as it is a direct application of other gadgets in the paper. We present our gadget for state preparation in Section 7, and our gadget for error correction in Section 8; the analyses are somewhat similar to that of the downwards code switching, so we postpone them to Appendix A and Appendix B, respectively. Section 9 presents some more basic gadgets, namely for transversal $CNOT$ and Hadamard gates, as well as logical measurements. While these gadgets in Section 9 were previously known, we state and analyze them in our specific fault-tolerance setup for completeness. In Section 10, we combine all of our gadgets listed above to prove Theorem 1.1.

3 Preliminaries

This section presents preliminary notions and relevant known results.

3.1 Notation

This section describes the basic notation that we use throughout. For $n \in \mathbb{N}$, we let $[n] = \{1, 2, \ldots, n\}$. For a set S, we let $2^S = \{S' : S' \subseteq S\}$ denote the power set of S. We denote by $\mathbb{F}_2 = \{0, 1\}$ the finite field on two elements. For $x \in \mathbb{F}_2^n$, we let $|x| = |\{i \in [n] : x_i \neq 0\}|$ denote the Hamming weight of x. For $x, y \in \mathbb{F}_2^n$, we let $x \cdot y = \sum_{i=1}^n x_i y_i$ denote the standard bilinear form. For $i \in [n]$, we let $\mathbb{1}_i \in \mathbb{F}_2^n$ denote the indicator vector for component i, so that $(\mathbb{1}_i)_j = 1$ iff $i = j$. In a slight abuse of notation (that will be made clear from context), for an event E, we also sometimes write $\mathbb{1}_E$ to be the indicator function for E occuring, so that for instance $\mathbb{1}_{i=j}$ equals 1 if $i = j$ and 0 otherwise.

An n-qubit pure quantum state is specified by a vector $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$. We use the standard notation $\mathbb{C}^2 = \text{span}\{|0\rangle, |1\rangle\}$ and $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. For a set $S \subseteq \mathbb{F}_2^n$, we let $|S\rangle = (1/\sqrt{|S|}) \sum_{x \in S} |x\rangle$ denote the uniform superposition of elements in S. A n-qubit density operator is a self-adjoint positive semi-definite operator $\rho \in \mathbb{C}^{2^n \times 2^n}$ of trace 1. We refer to linear maps $\mathcal{O} : \mathbb{C}^{2^n \times 2^n} \to \mathbb{C}^{2^n \times 2^n}$ as *superoperators*. A superoperator that is completely positive and

trace-preserving (CPTP) is called a quantum channel. We let $\mathcal{I}_n : \mathbb{C}^{2^n \times 2^n} \to \mathbb{C}^{2^n \times 2^n}$ denote the identity channel $\mathcal{I}_n(\rho) = \rho$.

Letting $I, X, Y, Z \in \mathbb{C}^{2\times 2}$ denote the standard single-qubit Pauli matrices, then an n-qubit Pauli matrix is a tensor product of n single-qubit Pauli matrices. For $P \in \{I, X, Y, Z\}$ and $x \in \mathbb{F}_2^n$, we write $P^x = \bigotimes_{i=1}^n P^{x_i}$. An n-qubit Pauli superoperator is a map of the form $\rho \mapsto A\rho B$ for n-qubit Paulis A, B.

We use ordinary big-O notation $O(\cdot)$, $\Theta(\cdot)$, $\Omega(\cdot)$. Subscripts will be used to denote variables that are held fixed, so that for example $f(r, n) = O_r(g(n))$ if for every fixed r, there exists some $c(r) > 0$ for which $f(r, n) \leq c(r) \cdot g(n)$.

3.2 Classical and Quantum Codes

This section presents standard definitions of classical and quantum codes.

Definition 3.1. *A* classical (binary linear) code of length n *is a linear subspace $C \subseteq \mathbb{F}_2^n$. The code's* dimension *is $k = \dim(C)$, and the distance is $d = \min_{c \in C \setminus \{0\}} |c|$. We summarize these parameters by saying that C is an $[n, k, d]$ code.*

The dual *of C is the code*

$$C^\perp = \{x \in \mathbb{F}_2^n : x \cdot c = 0 \ \forall c \in C\}.$$

A parity-check matrix *for C is a matrix $H \in \mathbb{F}_2^{m\times n}$ such that $C = \ker(H)$. We say that C with associated parity-check matrix H is* LDPC of locality w *if every row and column of H has $\leq w$ nonzero entries. We say a family of classical codes is* LDPC *if they are all LDPC of constant locality $w = O(1)$.*

Definition 3.2. *A* quantum (binary CSS) code of length n *is a pair of classical codes $Q = (Q_X, Q_Z)$. The code's* dimension *is $k = \dim(Q_Z) - \dim(Q_X^\perp)$, and the distance is*

$$d = \min_{c \in (Q_X \setminus Q_Z^\perp) \cup (Q_Z \setminus Q_X^\perp)} |c|.$$

We summarize these parameters by saying that Q is an $[[n, k, d]]$ code.

We say that Q is LDPC of locality w *if Q_X, Q_Z are classical LDPC codes of locality w. We similarly say that a family of quantum codes is* LDPC *if they are all LDPC of constant locality $w = O(1)$.*

3.3 Classical Codes from Lossless Expanders

In this section, we describe constructions of classical codes from constant-degree lossless expander graphs. The quantum codes we study arise from tensor products of cochain complexes associated to these classical codes, and will inherit many desirable properties from these expander graphs, such as constant encoding rate and decodability from the expansion.

First, we define lossless expander graphs.

Definition 3.3. *Let $G = (V = V_L \sqcup V_R, E \subseteq V_L \times V_R)$ be a bipartite graph of maximum left and right degrees Δ_L and Δ_R respectively. For a set of vertices $S \subseteq V$, we let $N_G(S) \subseteq V$ denote the neighborhood of S, that is, the set of vertices sharing an edge with some element of S. We say G is a* (μ, ϵ)-lossless expander *if it holds for every $S \subseteq V_L$ of size $|S| \leq \mu|V_L|$ that $|N_G(S)| \geq (1-\epsilon)\Delta_L|S|$, and for every $S \subseteq V_R$ of size $|S| \leq \mu|V_R|$ that $|N_G(S)| \geq (1-\epsilon)\Delta_R|S|$.*

The following basic lemma shows that lossless expansion implies that every small set S contains some vertex with many *unique neighbors*, meaning many neighbors that avoid the neighborhoods of all other vertices in S.

Lemma 3.4. *Let $G = (V = V_L \sqcup V_R, E)$ be a (μ, ϵ)-lossless expander of maximum left degree Δ_L. Then for every $S \subseteq V_L$ of size $|S| \leq \mu|V_L|$, there exists some $v \in S$ such that*

$$|N_G(v) \setminus N_G(S \setminus \{v\})| \geq (1 - 2\epsilon)\Delta_L. \tag{1}$$

(An analogous statement also holds for subsets $S \subseteq V_R$ of size $|S| \leq \mu|V_R|$.)

Proof. Let $E' \subseteq E$ be a set of $|E'| = |N_G(S)| \geq (1-\epsilon)\Delta_L|S|$ edges such that for each $v \in N_G(S)$, E' contains a single edge from a vertex in S to v. As there are $\leq \Delta_L|S|$ outgoing edges from S, it follows that at most $\Delta_L|S| - |E'| \leq \epsilon\Delta_L|S|$ of these edges do not lie in E', so at most $\epsilon\Delta_L|S|$ vertices in $N_G(S)$ have ≥ 2 edges coming from S. Therefore there are at least $|N_G(S)|-\epsilon\Delta_L|S| \geq (1-2\epsilon)\Delta_L|S|$ vertices in $N_G(S)$ that are incident to exactly 1 vertex in S, that is,

$$\sum_{v \in S} |N_G(v) \setminus N_G(S \setminus \{v\})| \geq (1 - 2\epsilon)\Delta_L|S|.$$

Thus by the pigeonhole principle, there is some $v \in S$ satisfying

$$|N_G(v) \setminus N_G(S \setminus \{v\})| \geq (1 - 2\epsilon)\Delta_L,$$

as desired. □

The following result shows that there exist explicit constant-degree expander graphs.

Proposition 3.5 ([HLM+25])**.** *For every fixed $\epsilon > 0$ and $\beta_2 > \beta_1 > 0$, there exists $\mu > 0$ and $\Delta_L, \Delta_R \in \mathbb{N}$ with $\beta_1 < \Delta_R/\Delta_L < \beta_2$ and $\Delta_L, \Delta_R \geq 1/\epsilon$ such that there is an infinite explicit sequence $(G^i = (V^i = V_L^i, V_R^i, E^i))_{i \in \mathbb{N}}$ of (Δ_L, Δ_R)-biregular (μ, ϵ)-lossless expanders, such that each $|V^i| < |V^{i+1}| \leq 2|V^i|$.*[6]

Remark 3.6. *The explicitness condition in Proposition 3.5 simply means that each graph G^i can be constructed by a $\mathrm{poly}(|V^i|)$-time algorithm. If this condition is dropped, then it is well-known that random biregular graphs also give lossless expanders that satisfy the other conditions in Proposition 3.5 (e.g. [HLW06, Theorem 4.16]).*

We construct classical LDPC codes by defining parity-check matrices from bipartite adjacency matrices of lossless expanders, as defined below.

Definition 3.7. *For a bipartite graph $G = (V = V_L \sqcup V_R, E)$, the associated* parity-check matrix *(also called the* bipartite adjacency matrix*) $H_G \in \mathbb{F}_2^{V_L \times V_R}$, is given by $(H_G)_{u,v} = \mathbb{1}_{(u,v) \in E}$.*

Our code switching gadgets will require identifying certain code components that contain the encoded logical information. The following lemma provides equivalent conditions for a set of bits in a classical code to contain such logical information.

Lemma 3.8. *For a code $C \subseteq \mathbb{F}_2^n$ and a set $S \subseteq [n]$, the following are equivalent:*

[6] [HLM+25] only stated their result with $|V^{i+1}| \leq O(|V^i|)$. However, the constant in the big-O can without loss of generality be reduced to 2 by considering disjoint unions of copies of the graphs, at the cost of a constant-factor reduction in μ.

1. *Every $x \in \mathbb{F}_2^S$ can be extended to a (not necessarily unique) $\tilde{x} \in \mathbb{F}_2^n$, so that $\tilde{x}|_S = x$.*

2. *Every $y \in C^\perp \setminus \{0\}$ satisfies $\mathrm{supp}(y) \not\subseteq S$.*

3. *The cosets $\mathbb{1}_v + C^\perp \in \mathbb{F}_2^n/C^\perp$ for $v \in S$ are all linearly independent.*

Proof. Condition 1 simply says that $C|_S = \mathbb{F}_2^S$, which is in turn equivalent to the requirement that C satisfies no nontrivial linear constraints supported inside S; this statement in turn is precisely Condition 2.

Meanwhile, Condition 3 says that every nonzero linear combination of elements $\mathbb{1}_v$ for $v \in S$ (i.e. every nonzero vector supported inside S) lies outside of $C^\perp$, which is precisely Condition 2. □

Definition 3.9. *A set $S \subseteq [n]$ is an* extendable set *for a code $C \subseteq \mathbb{F}_2^n$ if S satisfies the equivalent conditions in Lemma 3.8. A maximal extendable set, that is an extendable set S which becomes unextendable upon adding any addition element in $[n] \setminus S$, is called an* information set*.*

Note that by Condition 3 in Lemma 3.8, an information set for C must have size $\dim(\mathbb{F}_2^n/C^\perp) = \dim(C)$. That is, the bits in an information set contain the entire encoded message (up to some basis change); all bits outside of the information set are redundant bits used for error correction. An extendable set contains a subset of the bits in the message.

Below, we present a corollary of Proposition 3.5, in which we show that there exists lossless expanders that remain lossless expanders after removing the vertices associated to a linear-sized extendable set. This additional property will be crucial for our code switching gadget in Section 5 that switches from r-dimensional to $(r-1)$-dimensional product codes.

Specifically, to perform this downwards code switching, we measure out qubits that (in one of the r directions) lie outside of a fixed extendable set. As our measurement outcomes may be noisy, we run them through our decoder described in Section 4. However, this decoder requires lossless expansion, but we have only performed measurements on positions that lie outside of our fixed extendable set. Hence we need lossless expansion to be preserved under removing the extendable set.

Logical qubits outside of this extendable set are destroyed during the code switching, and hence should not be used to encode the message. Therefore we want this fixed extendable set to be linear-sized as in Corollary 3.10 below, so that we still have a constant encoding rate.

Corollary 3.10. *For every fixed $0 < \epsilon < 1$, there exist $\mu, R > 0$ and $\Delta_L, \Delta_R \in \mathbb{N}$ with $1/4 < \Delta_R/\Delta_L < 1/2$ for which there is an infinite explicit sequence $(G^i = (V^i = V_L^i \sqcup V_R^i, E^i))_{i\in\mathbb{N}}$ of (Δ_L, Δ_R)-biregular graphs satisfying:*

1. *$|V^i| < |V^{i+1}| \leq 2|V^i|$.*

2. *G^i is a (μ, ϵ)-lossless expander.*

3. *There exists a set $S_i \subseteq V_R^i$ of size $|S_i| \geq R|V_R^i|$ that is extendable for the code $\ker(H_{G^i})$, such that G^i remains a (μ, ϵ)-lossless expander if all vertices in S are removed (along with any edges with a vertex in S).*

Proof. For a given value of $\epsilon > 0$, we let $(G^i)_{i\in\mathbb{N}}$ be a family of (Δ_L, Δ_R)-biregular (μ, ϵ')-lossless expanders from Proposition 3.5 with expansion parameter $\epsilon' = \epsilon/2$, and with $\beta_1 = 1/4$ and $\beta_2 = 1/2$.

For a given $i \in \mathbb{N}$, we define the extendable set S_i as follows. Let $\ell = \Delta_L \Delta_R + 1$, and choose a partition $V_R = U_1 \sqcup \cdots \sqcup U_\ell$ such that no length-2 path in G^i (i.e. no edge in $(G^i)^2$) connects a pair of distinct vertices in the same set U_j. Because $(G^i)^2$ has degree $\Delta_L \Delta_R$, there exists such a partition of size $\ell = \Delta_L \Delta_R + 1$.

Let $T \subseteq V_R$ denote an information set for $\ker(H_{G^i})$, so that $|T| \geq \dim(\ker(H_{G^i})) \geq |V_R^i| - |V_L^i| \geq |V_R^i|/2$. By the pigeonhole principle, there exists some j with $|U_j \cap T| \geq |T|/r \geq |V_R^i|/2(\Delta_L \Delta_R + 1)$. We then let $S_i = U_j \cap T$, so that $|S_i| \geq R|V_R^i|$ for $R = 1/2(\Delta_L \Delta_R + 1)$.

By definition $S_i \subseteq T$ is extendable for $\ker(H_{G^i})$. It remains to be shown that the graph $G^i \setminus S_i$ obtained by removing all vertices in S_i from G^i is a (μ, ϵ)-lossless expander. For right-to-left expansion, every set $W \subseteq V_R \setminus S_i$ of size $|W| \leq \mu |V_R^i \setminus S_i| \leq \mu |V_R^i|$ by definition satisfies

$$|N_{G^i \setminus S_i}(W)| = |N_{G^i}(W)| \geq (1 - \epsilon')\Delta_R |W| \geq (1 - \epsilon)\Delta_R |W|.$$

For left-to-right expansion, consider a set $W \subseteq V_L$ of size $|W| \leq \mu |V_L^i|$. By construction, every $v \in W$ is incident to at most one element of $S_i \subseteq U_j$, so $|N_{G^i \setminus S_i}(W)| \geq |N_{G^i}(W)| - |W|$. Therefore

$$|N_{G^i \setminus S_i}(W)| \geq (1 - \epsilon')\Delta_L |W| - |W| = (1 - \epsilon' - 1/\Delta_L)\Delta_L |W| \geq (1 - 2\epsilon')\Delta_L |W| = (1 - \epsilon)\Delta_L |W|,$$

where the second inequality above holds by the fact that $\Delta_L \geq 1/\epsilon'$ from Proposition 3.5. The two inequalities above imply that $G^i \setminus S_i$ is a (μ, ϵ)-lossless expander, as desired. □

3.4 Chain Complexes

This section describes the notion of (co)chain complexes, which provide a useful language for constructing quantum CSS codes.

Definition 3.11. *An* r-dimensional chain complex $\mathcal{C}_*$ over $\mathbb{F}_2$ *is a graded* $\mathbb{F}_2$*-vector space* $\mathcal{C} = \bigoplus_{i=0}^r \mathcal{C}_i$ *together with a* boundary map $\partial : \mathcal{C} \to \mathcal{C}$ *satisfying* $\partial^2 = 0$ *and* $\partial(\mathcal{C}_i) \subseteq \mathcal{C}_{i-1}$*. We write* $\partial_i = \partial|_{\mathcal{C}_i}$*, and we summarize this data by writing*

$$\mathcal{C}_* = (\mathcal{C}_r \xrightarrow{\partial_r} \mathcal{C}_{r-1} \xrightarrow{\partial_{r-1}} \cdots \xrightarrow{\partial_1} \mathcal{C}_0).$$

We call $\mathcal{C}_i$ *the space of* i-chains *of* $\mathcal{C}_*$*, and we define the*

$$\begin{aligned} i\text{-cycles } Z_i(\mathcal{C}) &= \{z \in \mathcal{C}_i : \partial(z) = 0\} \\ i\text{-boundaries } B_i(\mathcal{C}) &= \{\partial(c) : c \in \mathcal{C}_{i+1}\} \\ i\text{-homology } H_i(\mathcal{C}) &= Z_i(\mathcal{C})/B_i(\mathcal{C}). \end{aligned}$$

In a slight abuse of notation, for $i \in \mathbb{Z} \setminus \{0, \ldots, r\}$ *we write* $C_i = \emptyset$ *and* $\mathcal{C}_i = \{0\}$*.*

The cochain complex

$$\mathcal{C}^* = (C^0 \xrightarrow{\delta_0} C^1 \xrightarrow{\delta_1} \cdots \xrightarrow{\delta_{r-1}} C^r)$$

associated to $\mathcal{C}_*$ *is the chain complex with the same vector space* $\mathcal{C} = \bigoplus_{i=0}^r \mathcal{C}^i$ *with each* $\mathcal{C}^i = \mathcal{C}_i$*, but whose boundary map is the* coboundary map $\delta : \mathcal{C} \to \mathcal{C}$ *given by* $\delta = \partial^\top$*. We then write* $\delta_i = \delta|_{\mathcal{C}^i} = \partial_{i+1}^\top$*. We call* $\mathcal{C}^i$ *the space of* i-cochains*, and we similarly define the*

$$\begin{aligned} i\text{-cocycles } Z^i(\mathcal{C}) &= \{z \in \mathcal{C}^i : \delta(z) = 0\} \\ i\text{-coboundaries } B^i(\mathcal{C}) &= \{\delta(c) : c \in \mathcal{C}_{i-1}\} \\ i\text{-cohomology } H^i(\mathcal{C}) &= Z^i(\mathcal{C})/B^i(\mathcal{C}). \end{aligned}$$

We will assume our chain complexes are based*, meaning that $\mathcal{C}_i = \mathbb{F}_2^{C_i}$ for a specified set C_i. We then define Hamming weights of elements of $\mathcal{C}_i = \mathcal{C}^i$ with respect to this basis. For $c_{i-1} \in C_{i-1}$ and $c_i \in C_i$, we write $c_{i-1} \lhd c_i$ if $c_{i-1} \in \mathrm{supp}(\partial(\mathbb{1}_{c_i}))$. For $c_j \in C_j$ and $c_i \in C_i$, with $i < j$, we write $c_i \prec c_j$ if there exists a sequence $c_{i+1}, \dots, c_{j-1}$ such that $c_i \lhd c_{i+1} \lhd \cdots \lhd c_{j-1} \lhd c_j$. This partial order provides the set $C = \bigsqcup_i C_i$ with the structure of a graded poset. For $i < j$, we extend the partial order notation to apply for a basis element $c_i \in C_i$ and a chain $c_j \in \mathcal{C}_j = \mathbb{F}_2^{C_j}$, so that $c_i \prec c_j$ if every $c' \in \mathrm{supp}(c_j)$ satisfies $c_i \prec c'$.*

The i-systolic distance $d_i(\mathcal{C})$ *and* i-cosystolic distance $d^i(\mathcal{C})$ *are defined as*

$$d_i(\mathcal{C}) = \min_{z \in Z_i(\mathcal{C}) \setminus B_i(\mathcal{C})} |z|$$
$$d^i(\mathcal{C}) = \min_{z \in Z^i(\mathcal{C}) \setminus B^i(\mathcal{C})} |z|.$$

We say that $\mathcal{C}_*$ has locality w *if for every $c \in C$ there are $\leq w$ basis elements $c' \in C$ with $c' \preceq c$ or $c' \succeq c$.*

Definition 3.12. *For r-dimensional chain complexes $\mathcal{A}_*, \mathcal{B}_*$, a* chain map $\phi : \mathcal{A}_* \to \mathcal{B}_*$ *is a sequence of maps $(\phi_i : \mathcal{A}_i \to \mathcal{B}_i)_{i \in [r]}$ satisfying $\partial_i^{\mathcal{B}} \circ \phi_i = \phi_{i-1} \circ \partial_i^{\mathcal{A}}$. Every chain map naturally induces a map on homology $\tilde{\phi} : H_*(\mathcal{A}) \to H_*(\mathcal{B})$.*

Definition 3.13. *For chain complexes $\mathcal{A}_*, \mathcal{B}_*$ of respective dimensions $r_{\mathcal{A}}, r_{\mathcal{B}}$, the* tensor product $\mathcal{C}_* = \mathcal{A}_* \otimes \mathcal{B}_*$ *is the chain complex of dimension $r_{\mathcal{C}} = r_{\mathcal{A}} + r_{\mathcal{B}}$ for which the i-chain basis is $C_i = \bigsqcup_{j \in \mathbb{Z}} A_j \times B_{i-j}$, so that $\mathcal{C}_i = \bigoplus_{j \in \mathbb{Z}} \mathcal{A}_j \otimes \mathcal{B}_{i-j}$, and the boundary map is given by $\partial^{\mathcal{C}} = \partial^{\mathcal{A}} \otimes I + I \otimes \partial^{\mathcal{B}}$. The tensor product of cochain complexes is defined analogously, so that $\mathcal{A}^* \otimes \mathcal{B}^* = (\mathcal{A}_* \otimes \mathcal{B}_*)^*$.*

The following formula describing how homology behaves under tensor products is well known.

Proposition 3.14 (Künneth formula)**.** *For chain complexes $\mathcal{A}_*, \mathcal{B}_*$ with tensor product $\mathcal{C}_* = \mathcal{A}_* \otimes \mathcal{B}_*$, then there is an isomorphism*

$$H_i(\mathcal{C}) \cong \bigoplus_{j \in \mathbb{Z}} H_j(\mathcal{A}) \otimes H_{i-j}(\mathcal{B}),$$

which for $a \in Z_i(\mathcal{A})$, $b \in Z_i(\mathcal{B})$ is given by

$$a \otimes b + B_i(\mathcal{C}) \leftarrow\!\shortmid (a + B_i(\mathcal{A})) \otimes (b + B_i(\mathcal{B})).$$

For a product complex $\mathcal{C}_* = \mathcal{C}_*^{(1)} \otimes \cdots \otimes \mathcal{C}_*^{(r)}$ with basis elements $c = (c_1, \dots, c_r)$, $c' = (c_1', \dots, c_r') \in C^{(1)} \times \cdots \times C^{(r)} = C$, the graded poset structure of C by definition has $c \preceq c'$ iff it holds for every $i \in [r]$ that $c_i \preceq c_i'$.

Definition 3.15. *The* quantum CSS code associated to level i of a cochain complex $\mathcal{C}^*$ *is given by $Q = (Q_X = \ker(\partial_i),\ Q_Z = \ker(\delta_i))$.*

The quantum code Q in Definition 3.15 by definition has length $n = \dim(\mathcal{C}^i)$, dimension $k = \dim(H^i(\mathcal{C}))$, and distance $d = \min\{d^i(\mathcal{C}), d_i(\mathcal{C})\}$. Furthermore, Q has naturally associated X, Z parity-check matrices ∂_i, δ_i respectively, so if $\mathcal{C}^*$ has locality w then by definition Q is LDPC of locality w.

3.5 Fault-Tolerance Model

In this section, we describe the model of circuits, faults, and fault-tolerance that we use in this paper. Our model will largely follow that of [NP25] (and the closely related [HNP25]). The most important distinction is that whereas [NP25] consider noisy quantum circuits that are permitted to run a constant-depth noiseless classical circuit at each timestep, we allow a polynomial-depth noiseless classical circuit at each timestep. This polynomial-depth classical computation will ultimately be used for solving linear systems of equations needed in our constant-overhead state preparation and code switching gadgets.

We also introduce some notational differences compared to [NP25]. For instance, as our focus in this paper is to provide certain new gadgets with reduced overhead rather than an end-to-end fault-tolerance scheme, we allow the inputs and outputs of circuits to be quantum states, rather than simply classical bit strings.

Below, we present our definition of a quantum circuit that may also perform (noisless) classical computation between timesteps.

Definition 3.16. *We define the* Clifford gate set *to consist of the single-qubit untaries $I, X, Y, Z, H, S = \sqrt{Z}$, the 2-qubit unitary $CNOT$, the single-qubit Pauli X, Y, Z measurement channels, and the single-qubit reset channels that reset a qubit to $|0\rangle$ or $|+\rangle$. For a non-Clifford gate U such as the 3-qubit unitary $U = CCZ$, the* Clifford+U *gate set consists of the Clifford gates along with U.*

An adaptive quantum circuit $\mathcal{Q} = (\mathcal{O}, C)$ using quantum space N, classical space M, and time T *consists of a sequence $\mathcal{O} = (\mathcal{O}_1, \dots, \mathcal{O}_T)$ of quantum channels acting on N qubits, and a sequence $C = (C_1, \dots, C_T)$ of classical circuits acting on a classical register of some number $M \geq N$ of bits. Formally, we represent this classical register by an M-qubit quantum register that always remains in (a mixture of) computational basis states. If for each $t \in T$, the channel $\mathcal{O}_t = \bigotimes_{i=1}^{g_t} \mathcal{O}_{t,i}$ is a product of Clifford (resp. Clifford+U) gates $\mathcal{O}_{t,i}$ acting on disjoint sets of qubits, we say $\mathcal{Q}$ is an adaptive* Clifford (resp. Clifford+U) *circuit.*

We now define an $(N+M)$-qubit superoperator $\mathcal{Q}(\cdot)$ associated to $\mathcal{Q}$. Given an N-qubit quantum input state $\rho \in \mathbb{C}^{2^N \times 2^N}$ and an M-bit classical input state $|x\rangle\langle x| \in \mathbb{C}^{2^M \times 2^M}$ for some $x \in \mathbb{F}_2^M$, we perform the following operators on our $N + M$ qubits for each $t = 1, \dots, T$:

1. *Apply C_t to the current classical state $|x\rangle\langle x|$.*
2. *For every $i \in [g_t]$ such that either $x_i = 1$ or $\mathcal{O}_{t,i}$ is a single-qubit Pauli measurement, apply $\mathcal{O}_{t,i}$ to the (appropriate qubits of the) current quantum state ρ. If $\mathcal{O}_{t,i}$ is a measurement, update $x_i \in \mathbb{F}_2$ to equal the measurement outcome.*

The output $\mathcal{Q}(\rho, |x\rangle\langle x|) \in \mathbb{C}^{2^{N+M} \times 2^{N+M}}$ is then defined to be the resulting $N + M$ qubit state. Note that in the construction above, the entire $N + M$ qubit state is always a mixture of states $|\phi\rangle\langle\psi|$ whose partial trace down to the M-(qu)bit classical register is a computational basis state $|x\rangle\langle x|$. Furthermore, if every $\mathcal{O}_{t,i}$ is a channel, then $\mathcal{Q}(\cdot)$ is a channel.

Remark 3.17. *In all of the constructions in this paper, the number M of classical bits and the size of the classical circuits C_t will grow at most polynomially in the number of qubits N.*

Remark 3.18. *For most of the adaptive quantum circuits considered in this paper, the classical input and output are not used, and rather the classical state x is simply reset and then used as a "scratchpad" to perform some intermediate computation. In such cases, as a shorthand we often leave out the classical inputs and outputs, and for instance write $\mathcal{Q}(\rho) \in \mathbb{C}^{2^N \times 2^N}$ to denote the*

result $\mathrm{tr}_{[M]}(\mathcal{Q}(\rho \otimes |0^M\rangle\langle 0^M|))$ *from executing* $\mathcal{Q}$ *with an all-0s classical input, and then discarding (i.e. tracing out) the classical output. If the classical input (resp. output) is nonempty but contains fewer than* M *bits, we just assume the extra bits are initialized to* 0 *(resp. traced out). When constructing gadgets throughout the paper, we will explicitly state whenever we are not using this shorthand, and have meaningful classical inputs or outputs.*

Definition 3.19. *For an adaptive quantum circuit* $\mathcal{Q} = (\mathcal{O}, C)$ *using quantum space* N *and time* T*, a* fault $\mathcal{F}$ *is a sequence* $\mathcal{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_T)$ *of* N*-qubit superoperators* $\mathcal{F}_t$*. We say* $\mathcal{F}$ *is a* Pauli fault *if each* $\mathcal{F}_t$ *is a Pauli superoperator, that is,* $\mathcal{F}_t(\rho) = F_t \rho F_t'$ *for some* N*-qubit Paulis* F_t, F_t'*. We say such a Pauli fault is* diagonal *if every* $F_t = F_t'$*.*

The $\mathcal{F}$-corrupted circuit $\mathcal{Q}[\mathcal{F}]$ *is the adaptive quantum circuit using quantum space* N *and time* $2T$ *given by*

$$\mathcal{Q}[\mathcal{F}] = ((\mathcal{O}_1, \mathcal{F}_1, \mathcal{O}_2, \mathcal{F}_2, \ldots, \mathcal{O}_T, \mathcal{F}_T), (C_1, I, C_2, I, \ldots, C_t, I)),$$

where I *denotes the identity (i.e. idling) circuit.*[7]

The fault path *or* support $\mathrm{supp}(\mathcal{F}) \subseteq [N] \times [T]$ *of* $\mathcal{F}$ *is the set of all space-time locations on which* $\mathcal{F}$ *acts nontrivially, that is,* $\mathrm{supp}(\mathcal{F}) = \bigsqcup_{t \in [T]} \mathrm{supp}(\mathcal{F}_t) \times \{t\}$*.*

Remark 3.20. *One could consider a more general error model where an adversary has a private register that can perform entangled corruptions across fault locations. For simplicity, we do not pursue this direction in this paper, but refer the reader to [HNP25] for more details.*

Our primary aim is to provide gadgets that exhibit a threshold under a locally stochastic fault distribution:

Definition 3.21. *For an adaptive quantum circuit using quantum space* N *and time* T*, a probability distribution over faults* $\mathcal{F}$ *is* ϵ-locally stochastic *if for every* $S \subseteq [N] \times [T]$*,*

$$\Pr[S \subseteq \mathrm{supp}(\mathcal{F})] \leq \epsilon^{|S|}.$$

Definition 3.22. *For a family of sets* $\mathcal{E} \subseteq 2^{[n]}$*, we say that a set* $S \subseteq [n]$ *is* $\mathcal{E}$-avoiding *if no element of* $\mathcal{E}$ *is contained in* S*.*

We extend this definition to faults in the natural way, so that a fault $\mathcal{F}$ *on a circuit using quantum space* $\leq N$ *and time* $\leq T$ *is* $\mathcal{E}$-avoiding *for* $\mathcal{E} \subseteq 2^{[N] \times [T]}$ *if no element of* $\mathcal{E}$ *is contained in* $\mathrm{supp}(\mathcal{F})$*.*

For families $\mathcal{E}_1 \subseteq 2^{[n_1]}$*,* $\mathcal{E}_2 \subseteq 2^{[n_2]}$*, we let* $\mathcal{E}_1 \sqcup \mathcal{E}_2 \subseteq 2^{[n_1] \sqcup [n_2]}$ *denote the disjoint union of* $\mathcal{E}_1$ *and* $\mathcal{E}_2$ *on separate sets of underlying elements. Therefore for* $\mathcal{E} \subseteq 2^{[n]}$ *and* $t \in \mathbb{N}$*, we write* $\mathcal{E}^{\sqcup t} \subseteq 2^{[n] \times [t]}$ *to denote* $\mathcal{E} \sqcup \cdots \sqcup \mathcal{E}$ *(with* t *copies of* $\mathcal{E}$*).*

Definition 3.23. *Let* Q *be an* $[[n, k]]$ *quantum code with associated encoding isometry* $\mathrm{Enc} : (\mathbb{C}^2)^{\otimes k} \to (\mathbb{C}^2)^{\otimes n}$*, which we also view as a channel in the natural way. For* $n' \geq n$*, let* $\mathcal{E} \subseteq 2^{[n']}$ *be a family of subsets of qubits, which we call* bad sets for Q*. For a density operator* $\rho \in \mathbb{C}^{2^k \times 2^k}$*, and for* $N \geq n$*, we say an operator* $\sigma \in \mathbb{C}^{2^N \times 2^N}$ *is a* (Pauli) $\mathcal{E}$-deviation of $\mathrm{Enc}(\rho)$ *if there exists a (Pauli) superoperator* $\mathcal{F} : \mathbb{C}^{2^n \times 2^n} \to \mathbb{C}^{2^n \times 2^n}$ *whose support is* $\mathcal{E}$*-avoiding such that* $\mathrm{tr}_{[N] \setminus [n]}(\sigma) \propto \mathcal{F} \circ \mathrm{Enc}(\rho)$*.*

[7] As a slight abuse of notation, when considering the superoperator $\mathcal{Q}[\mathcal{F}](\cdot)$ associated to the $\mathcal{F}$-corruped circuit $\mathcal{Q}[\mathcal{F}]$, we assume that the M-(qu)bit classical register x in Definition 3.16 is never updated in timesteps associated to corruptions (i.e. even timesteps), even if some $\mathcal{F}_t$ performs a single-qubit Pauli measurement. This convention is useful so that we can decompose arbitrary faults into superpositions of Pauli faults without changing the behavior of the associated superoperator. Note that this convention does not meaningfully change the error model, as the adversary can still corrupt a measurement outcome by corrupting the appropriate qubit in the timestep before the measurement.

We say that the data $D = (Q, \mathrm{Enc}, \mathcal{E})$ forms a decorated (quantum) code. *We let $\emptyset$ denote the* trivial decorated code *on $n = 0$ qubits.*

For decorated codes $D = (Q, \mathrm{Enc}, \mathcal{E})$ and $D' = (Q', \mathrm{Enc}', \mathcal{E}')$, we write

$$D \sqcup D' = (Q \sqcup Q',\ \mathrm{Enc} \sqcup \mathrm{Enc}',\ \mathcal{E} \sqcup \mathcal{E}')$$

to denote the decorated code consisting of disjoint blocks of D and D'.

Definition 3.24. *For a 2-register state $\rho \in \mathbb{C}^{2^{k+M} \times 2^{k+M}}$, we say that the second (size-M) register is a* computational basis state *if ρ is invariant under measuring the second register in the computational (Pauli Z) basis. Equivalently, the second register is a computational basis state if there exists a decomposition*

$$\rho = \sum_{x \in \mathbb{F}_2^M} \rho_x \otimes |x\rangle \langle x| .$$

Definition 3.25. *For $\alpha \in \{\mathrm{in}, \mathrm{out}\}$, let $D_\alpha = (Q_\alpha, \mathrm{Enc}_\alpha, \mathcal{E}_\alpha)$ be a decorated $[[n_\alpha, k_\alpha]]$ CSS code, and let $M_\alpha \in \mathbb{Z}_{\geq 0}$. Let $\bar{\mathcal{O}}$ be a quantum channel with $k_{\mathrm{in}} + M_{\mathrm{in}}$ input qubits and $k_{\mathrm{out}} + M_{\mathrm{out}}$ output qubits, such that if the M_{in}-qubit register of the input is a computational basis state, then so is the M_{out}-qubit register of the output. Let $\mathcal{Q}$ be an adaptive quantum circuit using quantum space $N \geq n_{\mathrm{in}}, n_{\mathrm{out}}$, classical space $M \geq M_{\mathrm{in}}, M_{\mathrm{out}}$, and time T with an associated family of* bad sets *$\mathcal{E}_{\mathrm{run}} \subseteq 2^{[N] \times [T]}$.*

We say that the data $((\mathcal{Q}, \mathcal{E}_{\mathrm{run}}), D_{\mathrm{in}}, D_{\mathrm{out}})$ provides a fault-tolerant gadget for $\bar{\mathcal{O}}$ *if for every $\ell \in \mathbb{N}$, every density operator $\rho \in \mathbb{C}^{2^{k+M_{\mathrm{in}}+\ell} \times 2^{k+M_{\mathrm{in}}+\ell}}$ for which the M_{in}-qubit register is a computational basis state, every $\mathcal{E}_{\mathrm{in}}$-deviation $\sigma \in \mathbb{C}^{2^{N+M+\ell} \times 2^{N+M+\ell}}$ of $\mathrm{Enc}_{\mathrm{in}} \otimes \mathcal{I}_{M_{\mathrm{in}}+\ell}(\rho)$ for which the M-qubit register is a computational basis state, and every $\mathcal{E}_{\mathrm{run}}$-avoiding fault $\mathcal{F}$ for $\mathcal{Q}$, then $\mathcal{Q}[\mathcal{F}] \otimes \mathcal{I}_\ell(\sigma) \in \mathbb{C}^{2^{N+M+\ell} \times 2^{N+M+\ell}}$ is a $\mathcal{E}_{\mathrm{out}}$-deviation of $(\mathrm{Enc}_{\mathrm{out}} \circ \bar{\mathcal{O}}) \otimes \mathcal{I}_\ell(\rho)$.*

The gadget is said to be Pauli fault-tolerant *if the same statement holds when restricting attention to Pauli deviations and Pauli faults.*

Similarly as in Remark 3.18, unless explicitly stated otherwise, we by default assume that there is no classical input/output, that is, $M_{\mathrm{in}} = M_{\mathrm{out}} = 0$.

Remark 3.26. *In Definition 3.23, we allow N-qubit deviations of n-qubit code states for $N \geq n$, so that a gadget may use a greater number N of qubits than taken by its input or output. For notational convenience, we also allow families of bad sets to be defined on a superset of the physical qubits; bad sets containing any extraneous elements outside of the physical qubits can be ignored.*

Remark 3.27. *The ℓ-qubit ancilla system in Definition 3.25 that is never encoded can be similarly found in [NP25, Definition 2.25], where it is called a* reference system. *While this reference system does not meaningfully change the analysis of our gadgets, [NP25] use the reference system to show that the parallel execution of multiple fault-tolerant gadgets remains fault-tolerant; see Proposition 3.28 below.*

Proposition 3.28 (Proposition 2.30 of [NP25])**.** *For (Pauli) fault-tolerant gadgets $((\mathcal{Q}, \mathcal{E}_{\mathrm{run}}), D_{\mathrm{in}}, D_{\mathrm{out}})$, $((\mathcal{Q}', \mathcal{E}'_{\mathrm{run}}), D'_{\mathrm{in}}, D'_{\mathrm{out}})$ for $\bar{\mathcal{O}}$, $\bar{\mathcal{O}}'$ respectively, then the parallel composition*

$$((\mathcal{Q} \sqcup \mathcal{Q}',\ \mathcal{E}_{\mathrm{run}} \sqcup \mathcal{E}'_{\mathrm{run}}),\ D_{\mathrm{in}} \sqcup D'_{\mathrm{in}},\ D_{\mathrm{out}} \sqcup D'_{\mathrm{out}})$$

is a (Pauli) fault-tolerant gadget for $\bar{\mathcal{O}} \sqcup \bar{\mathcal{O}}' = \bar{\mathcal{O}} \otimes \bar{\mathcal{O}}'$.

A standard technique for proving that a computation is fault-tolerant is to prove that it is Pauli fault-tolerant, and then decompose a more general fault into a linear combination Pauli faults. Each term in this linear combination can then be analyzed using the Pauli fault-tolerance. For instance, the following lemma is (implicitly) shown by [NP25] (see the subsection "Fault analysis and reduction to Pauli noise" of the proof of [NP25, Theorem 3.20]).

Lemma 3.29 (Folklore; see e.g. [NP25])**.** *Let $((\mathcal{Q}, \mathcal{E}_{\mathrm{run}}), D_{\mathrm{in}}, D_{\mathrm{out}})$ be a Pauli fault-tolerant gadget for a channel $\bar{\mathcal{O}}$ with entirely classical input and output, so that $k_{\mathrm{in}} = k_{\mathrm{in}} = 0$ (see Definition 3.25). Then $((\mathcal{Q}, \mathcal{E}_{\mathrm{run}}), D_{\mathrm{in}}, D_{\mathrm{out}})$ is also a fault-tolerant gadget for $\bar{\mathcal{O}}$.*

For simplicity, we have stated Lemma 3.29 for the case of gadgets with classical input and output. As we typically think of the ultimate circuits being run on quantum computers as having classical inputs and outputs, we may compile our smaller fault-tolerant gadgets into such end-to-end algorithms, and then apply Lemma 3.29 to the gadget corresponding to the entire algorithm. Furthermore, similar results as Lemma 3.29 can also be proven for quantum inputs/outputs, though additional assumptions may be necessary, such as needing the decorated code D_{out} to be able to correct all errors that avoid its specified bad sets.

Many of the above definitions closely mirror those in [NP25, HNP25]; the reader is referred to the discussion there for more context for these definitions.

The following family of bad sets will be particularly useful.

Definition 3.30. *For a graph $G = (V, E)$ and real numbers $\eta, \gamma > 0$, define a family $\mathcal{E}(G, \eta, \gamma) \subseteq 2^V$ by*

$$\mathcal{E}(G, \gamma, \eta) = \{S \subseteq V : S \text{ lies in a connected subgraph } G' \subseteq G \text{ with } |V(G')| \geq \eta \text{ and } |S|/|V(G')| \geq \gamma\}.$$

Lemma 3.31 below, which (in a similar form) is for instance shown in [KP13, Got14], implies that if G has constant $O(1)$ maximum degree, $\eta \geq \mathrm{poly}(|V|)$, and $\gamma = \Omega(1)$, then for every sufficiently small constant $\epsilon > 0$, an ϵ-locally stochastic error will contain a bad set in $\mathcal{E}(G, \gamma, \eta)$ with exponentially small probability $\leq e^{-\mathrm{poly}(|V|)}$.

Throughout this paper, we prove that our gadgets are Pauli fault-tolerant with respect to such families $\mathcal{E}(G, \gamma, \eta)$ of bad sets. Then Lemma 3.29 and Lemma 3.31 together imply that algorithms constructed using our gadgets exhibit a threshold under locally stochastic noise.

Lemma 3.31 (Well known; see e.g. [KP13, Got14])**.** *Let $G = (V, E)$ be a graph of maximum degree Δ, and let $\eta, \gamma, \epsilon > 0$ be real numbers with ϵ is sufficient small such that[8] $\epsilon \leq \gamma/e$ and $\Delta^2 \cdot (e\epsilon/\gamma)^\gamma \leq 1/2$. Let F be a random subset $F \subseteq V$ whose distribution satisfies $\Pr[S \subseteq F] \leq \epsilon^{|S|}$ for every $S \subseteq V$. Then*

$$\Pr[F \text{ is not } \mathcal{E}(G, \eta, \gamma)\text{-avoiding}] = \Pr[\exists S \in \mathcal{E}(G, \eta, \gamma) \text{ with } S \subseteq F] \leq |V| \cdot 2 \left(\frac{e\epsilon}{\gamma}\right)^\eta.$$

We include a brief proof of Lemma 3.31 for completeness.

Proof of Lemma 3.31. For $u \in \mathbb{N}$, let $\mathcal{V}_u \subseteq 2^V$ be the collection of sets $V' \subseteq V$ of size $|V'| = u$ that induce a connected subgraph of G. Then

$$|\mathcal{V}_u| \leq |V| \cdot \Delta^{2u}. \tag{2}$$

[8] Here $e = 2.718\ldots$ denotes Euler's number.

Specifically, every $V' \in \mathcal{V}_u$ contains a u-vertex spanning tree. In turn, every u-vertex subtree of G can be traversed by starting at one of the $|V|$ vertices in G, and then taking a length-$2u$ walk in G, at each step of which there are Δ choices of a neighbor to step to. Thus there are at most $|V| \cdot \Delta^{2u}$ such subtrees, and therefore (2) holds.

For every $V' \in \mathcal{V}_u$,

$$\Pr\big[|V' \cap F| \geq \gamma u\big] \leq \binom{u}{\lceil \gamma u \rceil} \cdot \epsilon^{\lceil \gamma u \rceil} \leq \left(\frac{eu}{\lceil \gamma u \rceil}\right)^{\lceil \gamma u \rceil} \cdot \epsilon^{\lceil \gamma u \rceil} \leq \left(\frac{e\epsilon}{\gamma}\right)^{\gamma u},$$

where the final inequality above holds by the assumption that $\epsilon \leq \gamma/e$, so that $(e\epsilon/\gamma)^{\lceil \gamma u \rceil} \leq (e\epsilon/\gamma)^{\gamma u}$.

Then union-bounding over every $V' \in \mathcal{V}_u$ and over every $u \geq \eta$ gives that

$$\begin{aligned}
\Pr[\exists S \in \mathcal{E}(G, \eta, \gamma) \text{ with } S \subseteq F] &\leq \sum_{u=\lceil \eta \rceil}^{|V|} \sum_{V' \in \mathcal{V}_u} \Pr\big[|V' \cap F| \geq \gamma u\big] \\
&\leq \sum_{u=\lceil \eta \rceil}^{|V|} |V| \cdot \Delta^{2u} \cdot \left(\frac{e\epsilon}{\gamma}\right)^{\gamma u} \\
&\leq |V| \cdot 2 \left(\frac{e\epsilon}{\gamma}\right)^{\eta},
\end{aligned}$$

where the third inequality above holds by the assumption that $\Delta^2 \cdot (e\epsilon/\gamma)^{\gamma} \leq 1/2$. □

We now describe how we decorate the quantum codes underlying Theorem 1.1, which come from products $\mathcal{C}^*$ of 1-dimensional cochain complexes arising from bipartite graphs. For this purpose, we need to specify both an encoding map and a graph G with which to define bad sets via Definition 3.30. As defined below, the encoding map will arise naturally from encoding maps for the underlying classical codes. Meanwhile, we choose the graph G to describe the incidence structure of $\mathcal{C}^*$.

Definition 3.32 (Encoding map and connectivity graph from cochain complex)**.** *Let $\mathcal{C}^*$ be an r-dimensional cochain complex. We say a* CSS encoding map *for the $[[n, k]]$ CSS code Q associated to level i of $\mathcal{C}^*$ is an isomorphism* $\mathrm{Enc} : \mathbb{F}_2^k \xrightarrow{\sim} H^i(\mathcal{C})$*. Such an encoding map naturally induces an encoding isometry* $\mathrm{Enc} : (\mathbb{C}^2)^{\otimes k} \to (\mathbb{C}^2)^{\otimes n}$ *given by* $\mathrm{Enc}\,|x\rangle = |\mathrm{Enc}(x)\rangle$*, which we may also view as a quantum channel.*

For $I \subseteq [r]$, we define a level-I connectivity graph $G_I^{\mathcal{C}}$ for $\mathcal{C}^*$ *to have vertex set $\bigsqcup_{i \in I} C^i \subseteq C$, and have an edge connecting every two vertices $c, c' \in C$ for which either there exists some $c^0 \in C^0$ with $c, c' \succeq c^0$, or there exists some $c^r \in C^r$ with $c, c' \preceq c^r$.*

Given an r-dimensional cochain complex $\mathcal{C}^*$ with a level $i \in [r]$ and real numbers $\eta, \gamma > 0$, Definition 3.32 then gives an associated decorated code

$$(Q, \mathrm{Enc}, \mathcal{E}(G_i^{\mathcal{C}}, \eta, \gamma)).$$

In our fault-tolerance analysis, we will often use such decorated codes, but with $G_i^{\mathcal{C}}$ replaced by some larger graph that contains $G_i^{\mathcal{C}}$ as a subgraph.

We now provide more details on how we construct CSS encoding maps for cochain complexes given by tensor products of 1-dimensional complexes. Note that there is a bijection between 1-dimensional cochain complexes and bipartite graphs, where the coboundary map of the complex corresponds to the parity-check/bipartite adjacency matrix of the graph.

Lemma 3.33. *Let*

$$\mathcal{C}^* = \left(\mathbb{F}_2^{V_L} \xrightarrow{\delta^{\mathcal{C}} = H_G} \mathbb{F}_2^{V_R} \right)$$

be a 1-dimensional cochain complex associated to a bipartite graph $G = (V_L \sqcup V_R, E)$. Let $M^0 \subseteq V_L$ and $M^1 \subseteq V_R$ be information sets for $\ker(\delta^{\mathcal{C}})$ and $\ker(\partial^{\mathcal{C}})$ respectively. Define a cochain complex

$$\mathcal{M}^* = (\mathbb{F}_2^{M^0} \xrightarrow{\delta^{\mathcal{M}}} \mathbb{F}_2^{M^1})$$

with coboundary map $\delta^{\mathcal{M}} = 0$. Define the cochain map $\mathrm{Enc} : \mathcal{M}^* \to \mathcal{C}^*$ such that for $x \in M^0$, then $\mathrm{Enc}^0(x)$ equals the unique codeword $\tilde{x} \in \ker(\delta)$ whose restriction $\tilde{x}|_{S_L} = x$, and for $x \in M^1$, then $\mathrm{Enc}^1(x) = (x, 0^{V_R \setminus M^1})$. Then Enc induces an isomorphism on cohomology.

Proof. Note that $\mathrm{Enc}^0(x)$ is well-defined and unique by the definition of an information set. Then Enc is a well-defined chain map because for every $x \in \mathbb{F}_2^{M^0}$, by definition $\delta^{\mathcal{C}}(\mathrm{Enc}^0(x)) = 0 = \mathrm{Enc}^1(\delta^{\mathcal{M}}(x))$. Now Enc^0 induces an isomorphism on 0-cohomology by item 1 of Lemma 3.8, while Enc^1 induces an isomorphism on 1-cohomology by item 3 of Lemma 3.8. □

Definition 3.34 (Product encoding map)**.** *For $r \in \mathbb{N}$ and $h \in [r]$, let*

$$\mathcal{C}^{(h)*} = \left(\mathbb{F}_2^{V_L^{(h)}} \xrightarrow{\delta^{(h)} = H_{G^{(h)}}^{\top}} \mathbb{F}_2^{V_R^{(h)}} \right)$$

be a 1-dimensional cochain complex associated to a bipartite graph $G^{(h)} = (V_L^{(h)} \sqcup V_R^{(h)}, E^{(h)})$. For $h \in [r]$, let $M^{(h)0} \subseteq V_L^{(h)}$ and $M^{(h)1} \subseteq V_R^{(h)}$ be information sets for $\ker(\delta^{(h)})$ and $\ker(\partial^{(h)})$ respectively, and let $\mathcal{M}^{(h)}$ and $\mathrm{Enc}^{(h)}$ be the associated cochain complex and cochain map respectively defined in Lemma 3.33.

Let $\mathcal{C}^* = \mathcal{C}^{(1)} \otimes \cdots \otimes \mathcal{C}^{(r)}$, $\mathcal{M}^* = \mathcal{M}^{(1)} \otimes \cdots \otimes \mathcal{M}^{(r)}$, and $\mathrm{Enc} = \mathrm{Enc}^{(1)} \otimes \cdots \otimes \mathrm{Enc}^{(r)}$. For $i \in [r]$, by Lemma 3.33 along with the Künneth formula (Proposition 3.14), Enc induces an isomorphism $\widetilde{\mathrm{Enc}}^i : \mathcal{M}^i = H^i(\mathcal{M}) \xrightarrow{\sim} H^i(\mathcal{C})$ on cohomology. We call this isomorphism a product encoding map*.*

3.6 Basic Subroutines

Lemma 3.35 (Vizing's theorem (e.g. [MG92]))**.** *For every graph $G = (V, E)$ of maximum degree Δ, there exists a $O(|V| \cdot |E|)$-time algorithm to compute a coloring of the edges with $\Delta + 1$ colors such that two edges sharing a vertex have the same color.*

The following standard corollary applies Vizing's theorem to obtain a low-depth syndrome extraction circuit for a CSS code, as given in Algorithm 1.

Corollary 3.36 (Well known)**.** *Let $H \in \mathbb{F}_2^{m \times n}$ be a parity-check matrix of locality w, and for $i \in [m]$ let H_i denote the ith row of H. Then for $P \in \{X, Z\}$, there exists a Clifford circuit using quantum space $N = n + m$ and time $T = w + 4$ that takes as input a n-qubit state ρ, and outputs ρ following measurements of the n-qubit Pauli operators P^{H_i} for $i \in [m]$, along with the m measurement outcomes.*

Algorithm 1: Syndrome extraction circuit in Corollary 3.36. Note that by Lemma 3.35, each step of the outer loop over t in line 4 only performs gates with disjoint supports, and hence can be implemented entirely within timestep t. We call the n qubits in the input ρ *data qubits*.

Input : n-qubit state ρ, Pauli $P \in \{X, Z\}$, parity-check matrix $H \in \mathbb{F}_2^{m \times n}$
Output: Outcomes of measuring P^{H_i} on ρ for $i \in [m]$, along with post-measurement state

1 **Function** `SyndExt`$(\rho; P, H)$**:**
2 Initialize m ancilla qubits to $|0\rangle$ (if $P = Z$) or $|+\rangle$ (if $P = X$)
3 For the bipartite graph $G = ([m] \sqcup [n], E)$ given by the parity-check matrix $H = H_G$, compute a coloring $\chi : E \to \{2, \dots, w+3\}$ using Lemma 3.35.
4 **foreach** $t = 2, \dots, w+3$ **do**
5 **foreach** $(i, j) \in E : \chi(i, j) = t$ **do**
6 **if** $P = X$ **then**
7 Apply $CNOT$ to ancilla qubit i and data qubit j
8 **if** $P = Z$ **then**
9 Apply $CNOT$ to data qubit j and ancilla qubit i
10 Measure Pauli P on all m ancilla qubits

4 Small-Set Flip Decoding of Product Codes

In this section, we construct a small-set flip decoder for product codes from lossless expanders. This decoder is a key ingredient in many of our fault-tolerance gadgets.

Definition 4.1. *For $m \in \mathbb{N}$, we say a cochain complex $\mathcal{C}^*$ has a* m-small-set error-flip decoder at level i *if for every nonzero $e \in \mathcal{C}^i$ of weight $|e| \le m$, there exists a basis element $c^0 \in C^0$ for which at least one of the following holds:*

1. *There exists a cochain $c^{i-1} \in \mathcal{C}^{i-1}$ with $c^0 \preceq c^{i-1}$ such that $|e + \delta(c^{i-1})| < |e|$, or*
2. *There exists a cochain $c^i \in \mathcal{C}^i$ with $c^0 \preceq c^i$ such that $|\delta(e + c^i)| < |\delta(e)|$.*

For $0 < \nu < 1$, we say $\mathcal{C}^*$ has a (m, ν)-small-set error-flip decoder at level i *if in condition 2 above, we impose the stronger requirement that $|\delta(e + c^i)| < |\delta(e)| - (1 - \nu)|\delta(c^i)|$.*

We say $\mathcal{C}^*$ has a (m, γ)-small-set syndrome-flip decoder at level i *if for every nonzero $e \in \mathcal{C}^i$ of weight $|e| \le m$ and every $f \in \mathcal{C}^{i+1}$ of weight $|f| \le \gamma|\delta(e)|$, there exists a basis element $c^0 \in C^0$ and a cochain $c^i \in \mathcal{C}^i$ with $c^0 \preceq c^i$ such that $|\delta(e + c^i) + f| < |\delta(e) + f|$.*

We say that $\mathcal{C}^*$ has a (m, γ)-small-set flip decoder at level i *if $\mathcal{C}^*$ has both a m-small-set error-flip decoder and a (m, γ)-small-set syndrome-flip decoder at level i.*

In Definition 4.1, an error-flip decoder is given access to a corrupted codeword and hence can compute a perfect syndrome. However, a syndrome-flip decoder is only given access to a syndrome, which may be corrupted in $|f| \le \gamma|\delta(e)|$ locations. In Algorithm 2, we translate these abstract notions of decoders into concrete algorithms.

Remark 4.2. *A (m, γ)-small-set syndrome-flip decoder must have $\gamma < 1$, as otherwise setting $f = \delta(e)$ for arbitrary $e \neq 0$ would yield $\delta(e) + f = 0$, whose weight cannot be reduced.*

Algorithm 2: (Classical) decoding algorithms associated to a small-set flip decoder from Definition 4.1.

Input : $(i+1)$-cochain $s \in \mathcal{C}^{i+1}$ of a cochain complex $\mathcal{C}^*$
Output: $a^i \in \mathcal{C}^i$ with $\delta(a^i)$ close to s

Function `SSFlipSyn`$(s; i, \mathcal{C}^*)$**:**
  Initialize $a^i \leftarrow 0 \in \mathcal{C}^i$
  while $\exists c^0 \in \mathcal{C}^0, c^i \in \mathcal{C}^i$ *with* $c^0 \preceq c^i$ *and* $|s + \delta(a^i + c^i)| < |s + \delta(a^i)|$ **do**
    $a^i \leftarrow a^i + c^i$
  return a^i

Input : i-cochain $e \in \mathcal{C}^i$ of a cochain complex $\mathcal{C}^*$
Output: $a^{i-1} \in \mathcal{C}^{i-1}, a^i \in \mathcal{C}^i$ with $e = a^i + \delta(a^{i-1})$

Function `SSFlipErr`$(e; i, \mathcal{C}^*)$**:**
  Initialize $a^{i-1} \leftarrow 0 \in \mathcal{C}^{i-1}$, $a^i \leftarrow 0 \in \mathcal{C}^i$
  while $e \neq a^i + \delta(a^{i-1})$ **do**
    if $\exists c^0 \in \mathcal{C}^0, c^{i-1} \in \mathcal{C}^{i-1}$ *with* $c^0 \preceq c^{i-1}$ *and* $|e + a^i + \delta(a^{i-1} + c^{i-1})| < |e + a^i + \delta(a^{i-1})|$ **then**
      $a^{i-1} \leftarrow a^{i-1} + c^{i-1}$
    else if $\exists c^0 \in \mathcal{C}^0, c^i \in \mathcal{C}^i$ *with* $c^0 \preceq c^i$ *and* $|\delta(e + a^i + c^i)| < |\delta(e + a^i)|$ **then**
      $a^i \leftarrow a^i + c^i$
    else
      return *FAIL*
  return a^{i-1}, a^i

Lemma 4.4 and Lemma 4.5 below apply the notion that `SSFlipSyn()` and `SSFlipErr()` in Algorithm 2 are "local algorithms" (see e.g. [FGL18]), meaning that they act independently on different connected components of the subgraph of the connectivity graph defined in Definition 3.32 induced by the flipped cochain basis elements. The statements of these lemmas require the following definition.

Definition 4.3. *Fix an r-dimensional cochain complex $\mathcal{C}^*$ and some level $0 \leq i \leq r-1$.*

For a data error $e \in \mathcal{C}^i$ and a syndrome error $f \in \mathcal{C}^{i+1}$, we say the footprint at time t *of* $\texttt{SSFlipSyn}(\delta(e)+f; i, \mathcal{C}^*)$ *is the subset of $C^i \sqcup C^{i+1}$ given by the union of the values of the sets* $\mathrm{supp}(e+a^i) \subseteq C^i$ *and* $\mathrm{supp}(\delta(e+a^i)+f) \subseteq C^{i+1}$ *across the first t iterations of the while loop.*

Similarly, for $e \in \mathcal{C}^i$, we say the footprint at time t *of* $\texttt{SSFlipErr}(e; i, \mathcal{C}^*)$ *is the subset of $C^{i-1} \sqcup C^i \sqcup C^{i+1}$ given by the union of the values of the sets $a^{i-1} \subseteq C^{i-1}$,* $\mathrm{supp}(e+a^i+\delta(a^{i-1})) \subseteq C^i$ *and* $\mathrm{supp}(\delta(e+a^i)) \subseteq C^{i+1}$ *across the first t iterations of the while loop.*

In both cases, by the footprint *we mean the footprint at time $t = \infty$ after the respective while loop has terminated.*

In Lemma 4.4 below, we slightly abuse the standard notation of restriction of vectors. Specifically, for $x \in \mathbb{F}_2^n$ and $I \subseteq [n]$, we let $x|_I \in \mathbb{F}_2^n$ denote the vector obtained from x by replacing the values of every component in $[n] \setminus I$ with 0; this notation can be seen as a shorthand for writing $(x|_I, 0^{[n]\setminus I})$.

Lemma 4.4. *Fix an r-dimensional cochain complex $\mathcal{C}^*$. For $0 \leq i \leq r-1$, $e \in \mathcal{C}^i$, $f \in \mathcal{C}^{i+1}$, let $S_{\mathrm{syn}} \subseteq C^i \sqcup C^{i+1}$ and $a^i \in \mathcal{C}^i$ denote the footprint and output, respectively, of $\texttt{SSFlipSyn}(\delta(e)+f; i, \mathcal{C}^*)$. For every connected component V of the subgraph of $G^{\mathcal{C}}_{i,i+1}$ induced by S_{syn}, the output of $\texttt{SSFlipSyn}(\delta(e|_{V\cap C^i}) + (f|_{V\cap C^{i+1}}); i, \mathcal{C}^*)$ must equal $a^i|_{V\cap C^i}$.*

Similarly, for $0 \leq i \leq r-1$, $e \in \mathcal{C}^i$, let $S_{\mathrm{err}} \subseteq C^{i-1} \sqcup C^i \sqcup C^{i+1}$ and $a^{i-1} \in \mathcal{C}^{i-1}$, $a^i \in \mathcal{C}^i$ denote the footprint and output, respectively, of $\texttt{SSFlipErr}(e; i, \mathcal{C}^*)$. For every connected component V of the subgraph of $G^{\mathcal{C}}_{i-1,i,i+1}$ induced by S_{err}, the output of $\texttt{SSFlipErr}(e|_{V\cap C^i}; i, \mathcal{C}^*)$ must equal $a^{i-1}|_{V\cap C^{i-1}}$, $a^i|_{\cap C^i}$.

Proof. By definition, every update c^i or c^{i-1} in any iteration of the while loop of either algorithm $\texttt{SSFlipSyn}(\delta(e)+f; i, \mathcal{C}^*)$ or $\texttt{SSFlipErr}(e; i, \mathcal{C}^*)$ must be supported within the neighborhood in $G^{\mathcal{C}}_{i,i+1}$ or $G^{\mathcal{C}}_{i-1,i,i+1}$ respectively of the footprint from the previous step. Thus different connected components of the subgraph induced by the footprint do not interact at all during the algorithm's execution, so the result follows. □

Lemma 4.5. *Fix an r-dimensional cochain complex $\mathcal{C}^*$ of locality w. Then for every $0 \leq i \leq r-1$, $e \in \mathcal{C}^i$, $f \in \mathcal{C}^{i+1}$, in Algorithm 2, at least $\gamma_{\mathrm{syn}} = 1/4w^3$-fraction of the vertices in every connected component in the subgraph of $G^{\mathcal{C}}_{i,i+1}$ induced by the footprint at every time t of $\texttt{SSFlipSyn}(\delta(e)+f; i, \mathcal{C}^*)$ must lie inside* $\mathrm{supp}(e) \sqcup \mathrm{supp}(f)$.

Similarly, for every $0 \leq i \leq r-1$, $e \in \mathcal{C}^i$, at least $\gamma_{\mathrm{err}} = 1/8w^4$-fraction of the vertices in every connected component in the subgraph of $G^{\mathcal{C}}_{i-1,i,i+1}$ induced by the footprint at every time t of $\texttt{SSFlipErr}(e; i, \mathcal{C}^*)$ must lie inside $\mathrm{supp}(e)$.

Proof. We first prove the claim regarding $\texttt{SSFlipSyn}(\delta(e)+f; i, \mathcal{C}^*)$. In this algorithm, by definition every update c^i (that gets added in to a^i) at a given timestep has $\mathrm{supp}(c^i)$ within the neighborhood of the footprint at the prior timestep. Therefore if S denotes the footprint at time t, then within every connected component V of the subgraph of $G^{\mathcal{C}}_{i,i+1}$ induced by S, the sequence of updates c^i

is the same as if all entries of e, f outside of V were replaced with 0s. It also follows that each $\mathrm{supp}(c^i)$ either lies entirely inside or outside of V.

Assume for a contradiction that $< \gamma_{\mathrm{syn}}$-fraction of the vertices in V lie in $\mathrm{supp}(e) \sqcup \mathrm{supp}(f)$. Then because the value $|\mathrm{supp}(\delta(e + a^i) + f) \cap V|$ must decrease in each iteration of the while loop prior to time t for which $\mathrm{supp}(c^i) \subseteq V$, starting from the value $|\mathrm{supp}(\delta(e) + f) \cap V| < w \cdot \gamma_{\mathrm{syn}}|V|$ at time 0, and each $|c^i| \leq w$ and $|\delta(c^i)| \leq w^2$, we must have that

$$\begin{aligned}
|V| &\leq |\mathrm{supp}(e \sqcup \delta(e) \sqcup f) \cap V| + |(\delta(e) + f) \cap V| \cdot (w + w^2) \\
&< \gamma_{\mathrm{syn}}|V|((w+1) + w \cdot (w + w^2)) \\
&\leq 4\gamma_{\mathrm{syn}} w^3 |V| \\
&= |V|.
\end{aligned}$$

Specifically, the size of the intersection of the footprint and V is at most $|\mathrm{supp}(e \sqcup \delta(e) \sqcup f) \cap V|$ at time 0, and increases by at most $|c^i| + |\delta(c^i)| \leq w + w^2$ in at most $|\mathrm{supp}(\delta(e) + f) \cap V|$ iterations of the while loop prior to time t. But the above inequality $|V| < |V|$ is a contradiction, so the assumption that $< \gamma_{\mathrm{syn}}$-fraction of the vertices in V lie in $\mathrm{supp}(e) \sqcup \mathrm{supp}(f)$ was false, as desired.

We now similarly prove the claim regarding `SSFlipErr`$(e; i, \mathcal{C}^*)$. In this algorithm, by definition every update c^{i-1} or c^i at a given timestep has support within the neighborhood of the footprint at the prior timestep. Therefore if S denotes the footprint at time t, then within every connected component V of the subgraph of $G^{\mathcal{C}}_{i,i+1,i+2}$ induced by S, the sequence of updates c^{i-1} and c^i is the same as if all entries of e outside of V were replaced with 0s. It also follows that each $\mathrm{supp}(c^{i-1})$ and $\mathrm{supp}(c^i)$ either lies intirely inside or outside of V.

Assume for a contradiction that $< \gamma_{\mathrm{err}}$-fraction of the vertices in V lie in $\mathrm{supp}(e)$. Then either $|\mathrm{supp}(e + a^i + \delta(a^{i-1})) \cap V|$ or $|\mathrm{supp}(\delta(e + a^i)) \cap V|$ must decrease in each iteration of the while loop prior to time t for which $\mathrm{supp}(c^{i-1}) \subseteq V$ or $\mathrm{supp}(c^i) \subseteq V$. By definition $|\mathrm{supp}(\delta(e + a^i)) \cap V|$ can only decrease, starting from the value $|\mathrm{supp}(\delta(e)) \cap V| \leq w|\mathrm{supp}(e) \cap V|$. Every time this value decreases, the value $|\mathrm{supp}(e + a^i + \delta(a^{i-1})) \cap V|$ increases by at most $|c^i| \leq w$, and hence there are at most $|\mathrm{supp}(e) \cap V| + w \cdot w|\mathrm{supp}(e) \cap V| = (1 + w^2)|\mathrm{supp}(e) \cap V|$ iterations in which the algorithm adds in an update c^{i-1} to decrease $|\mathrm{supp}(e + a^i + \delta(a^{i-1})) \cap V|$. Therefore we obtain a contradiction

$$\begin{aligned}
|V| &\leq |\mathrm{supp}(e \sqcup \delta(e)) \cap V| + w|\mathrm{supp}(e) \cap V| \cdot (w + w^2) + (1 + w^2)|\mathrm{supp}(e) \cap V| \cdot (w + w^2) \\
&\leq |\mathrm{supp}(e) \cap V|((1 + w) + w(w + w^2) + (1 + w^2)(w + w^2)) \\
&< \gamma_{\mathrm{err}}|V| \cdot 8w^4 \\
&\leq |V|,
\end{aligned}$$

so the assumption that $< \gamma_{\mathrm{err}}$-fraction of the vertices in V lie in $\mathrm{supp}(e)$ was false, as desired. □

We now prove the main result of this section, which is a small-set flip decoder for the tensor product of 1-dimensional cochain complexes associated to lossless expanders.

Proposition 4.6. *For every $r \in \mathbb{N}$ and $0 < \beta \leq 1$, there exists $\epsilon = \epsilon(r, \beta) > 0$ such that the following holds. For some $\mu > 0$ and $\Delta_{\max} \in \mathbb{N}$, for each $h \in [r]$ let $G^{(h)} = (V^{(h)} = V_L^{(h)} \sqcup V_R^{(h)}, E^{(h)})$ be a (μ, ϵ)-lossless expander of maximum left-degree $\Delta_L^{(h)} \leq \Delta_{\max}$ and minimum left-degree $\geq \beta\Delta_{\max}$. Let*

$$\mathcal{C}^{(h)*} = \left(\mathbb{F}_2^{V_L^{(h)}} \xrightarrow{\delta^{(h)} = H_{G^{(h)}}^\top} \mathbb{F}_2^{V_R^{(h)}} \right)$$

be the associated 1-dimensional cochain complex. Then the tensor product $\mathcal{A}^* := \mathcal{C}^{(1)} \otimes \cdots \otimes \mathcal{C}^{(r)}$ *has an* (m, γ)*-small-set flip decoder at every level* $0 \le i \le r-1$ *for* $m = \min_{h \in [r]} \mu |V_L^{(h)}| / (r\Delta_{\max}^{r+1} + 1)$ *and* $\gamma = 1/10$.

To prove Proposition 4.6, we will use the notion of *robustness* studied in [KP23, DLV24, KP25]. While this notion of robustness applies to arbitrary collections of classical codes, we will only need to consider it for collections of repetition codes (of possibly different lengths), as described below.

Definition 4.7. *For* $r \in \mathbb{N}$ *and* $n_1, \ldots, n_r \in \mathbb{N}$*, consider the 1-dimensional cochain complexes*

$$\mathcal{R}^{(h)} = \left(\mathbb{F}_2 \xrightarrow{\delta^{(h)}} \mathbb{F}_2^{n_h} \right)$$

in which each coboundary map $\delta^{(h)}$ *is simply given by the* $n_h \times 1$ *matrix of all* 1*s. We say that the collection of* r *repetion codes* $(\mathrm{im}(\delta^{(1)}), \ldots, \mathrm{im}(\delta^{(r)}))$ *(of lengths* $n_1, \ldots, n_r$ *respectively) is* η-robust *if the* r*-dimensional product cochain complex*

$$\mathcal{Q}^* = \mathcal{R}^{(1)} \otimes \cdots \otimes \mathcal{R}^{(h)}$$

satisfies the following: for every $0 \le i \le r-1$ *and every* $c \in \mathcal{Q}^i$*, there exists* $b \in B^i(\mathcal{Q})$ *such that*

$$\eta \cdot |b + c| \le |\delta(c)|.$$

Note that if all r repetition codes have the same length $n = n_1 = \cdots = n_r$, then our robustness parameter η equals n times the robustness parameter defined in prior works [KP23, DLV24, KP25]. This slightly different convention is convenient for us because we will consider codes of different lengths $n_1 \neq \cdots \neq n_r$.

Lemma 4.8 (Follows from [DLV24, KP25])**.** *For every* $r \in \mathbb{N}$ *and* $\beta > 0$*, there exists* $\kappa = \kappa(r, \beta) > 0$ *such that for every* $n \in \mathbb{N}$*, every* r*-tuple of repetition codes whose lengths all lie in* $[\beta n, n]$ *is* κn*-robust.*

Proof sketch. We briefly explain how the lemma follows from results in [DLV24, KP25]. [DLV24] show that a collection of classical codes is robust if it satisfies a notion called *product-expansion*. [KP25] in turn showed that a collection of codes is product-expanding if all of the codes satisfy a more standard property called *local testability*. But it is well known that classical repetition codes are locally testable, for instance with parity-check matrix given by the vertex-edge incidence matrix of a constant-degree spectral-expander graph $G = (V, E)$. Specifically, we place code bits on vertices in V, and parity-checks on edges in E enforce that the bits assigned to the incident vertices are equal. Then for a repetition codeword with an error supported on vertices in a set $S \subseteq V$, the probability that a random parity-check fails equals the fraction of edges with one vertex in S and one vertex outside of S, which by the expander mixing lemma (see e.g. [Vad12]) is proportional to $\min\{|S|, |V| - |S|\}/|V|$. Therefore a random check fails with probability proportional to the error weight, so the repetition code is locally testable, and thus [KP25] implies the desired product-expansion result. □

Remark 4.9. *While some of the results in [DLV24, KP25] are stated for collections of codes that all have the same length, these results and their proofs extend naturally to our setting where all code lengths lie in an interval* $[\beta n, n]$ *for some constant* $\beta > 0$*, at the cost of just worse constants in the expressions bounding product-expansion and robustness.*

For conciseness above we deduced that repetition codes are product-expanding by applying their local testability along with the results of [KP25]. Alternatively, it can be shown more directly that collections of repetition codes are product-expanding.

In Lemma 4.10 below, we construct the error-flip decoder for Proposition 4.6. We will subsequently apply this error-flip decoder to construct a syndrome-flip decoder.

Lemma 4.10. *For every $r \in \mathbb{N}$, $0 < \beta \leq 1$, and $0 < \nu < 1$, there exists $\epsilon = \epsilon(r, \beta, \nu) > 0$ such that the following holds. Define $\mu, \Delta_{\max}, G^{(h)}, \Delta_L^{(h)}, \mathcal{C}^{(h)^*}, \mathcal{A}^*$ as in Proposition 4.6. Then $\mathcal{A}^*$ has an (m, ν)-small-set error-flip decoder at every level $0 \leq i \leq r-1$ for $m = \min_{h \in [r]} \mu |V_L^{(h)}|$.*

Proof. Define $\kappa = \kappa(r, \beta/2) > 0$ to be the value from Lemma 4.8. Set

$$\epsilon = \min \left\{ \frac{\beta\nu}{8}, \frac{\kappa\nu}{16r} \right\}. \tag{3}$$

Fix some $0 \leq i \leq r-1$ and some nonzero $e \in \mathcal{A}^i$ of weight $|e| \leq m$. Our goal is to show that there exists some basis element $a^0 \in A^0$ satisfying one of the two conditions in Definition 4.1 (with $\mathcal{C}^* = \mathcal{A}^*$).

For this purpose, we first construct a sequence of vertices $(v_h \in V_L^{(h)})_{h \in [r]}$ and subsets $(U_h \subseteq N_{G^{(h)}}(v_h))_{h \in [r]}$ inductively as follows, where each $|U_h| \geq (1-\epsilon)\Delta_L^{(h)}$. We will then appeal to the fact that the restriction of $\mathcal{A}^*$ to basis elements corresponding to vertices in $\{v_h\} \cup U_h$ for $h \in [r]$ has the structure of the product of repetition codes, as considered in Definition 4.7 and Lemma 4.8.

For $t \in [r]$, assume we have already determined $v_1, \ldots, v_{t-1}$ and $U_1, \ldots, U_{t-1}$. Recall that $A = \bigsqcup_{j \in [r]} A^r = V^{(1)} \times \cdots \times V^{(r)}$, and for $h \in [r]$ let $\Pi^{(h)} : A \to V^{(h)}$ denote projection onto the hth component. Define $B_t \subseteq A$ by

$$B_t = \{a \in A : a_h \in (\{v_h\} \cup U_h) \; \forall \; h \in [t-1]\},$$

and define $F_t \subseteq V^{(t)}$ by

$$F_t = \Pi^{(t)}(\mathrm{supp}(e) \cap B_t).$$

That is, F_t contains vertices in $G^{(t)}$ that equal the tth component of some $a \in \mathrm{supp}(e)$ whose first $t-1$ components lie in the respective sets $\{v_h\} \cup U_h$ for $h \in [t-1]$.

We will enforce the inductive hypothesis that $F_t \neq \emptyset$, and that for every $b = (b_1, \ldots, b_r) \in B_t$ and $h \in [t-1]$, every $v \in V_L^{(h)}$ with $v \triangleleft b_h$ and $(b_1, \ldots, b_{h-1}, v, b_{h+1}, \ldots, b_r) \in \mathrm{supp}(e)$ must equal $v = v_h$.

For the base case, when $t = 1$ by definition $F_t = \Pi^{(1)}(\mathrm{supp}(e))$ is nonempty by the assumption that $e \neq 0$. Now for $t \geq 1$, assume the inductive hypothesis holds. If $F_t \cap V_L^{(t)} = \emptyset$, then we must have $F_t \cap V_R^{(t)} \neq \emptyset$, so we set v_t to be any vertex in $V_L^{(t)}$ that is incident to any vertex in F_t, and we set $U_t = N_{G^{(t)}}(v_t)$. Otherwise, if $F_t \cap V_L^{(t)} \neq \emptyset$, then because by assumption $|F_t| \leq |e| \leq m \leq \mu |V_L^{(t)}|$, we may apply Lemma 3.4 with $S = F_t \cap V_L^{(t)}$ to choose some $v_t \in S$ such that the set

$$U_t := N_{G^{(t)}}(v_t) \setminus N_{G^{(t)}}(S \setminus \{v_t\})$$

satisfies

$$|U_t| \geq (1 - 2\epsilon)\Delta_L^{(t)}. \tag{4}$$

In both cases above, we chose v_t, U_t so that $(\{v_t\} \cup U_t) \cap F_t \neq \emptyset$. Therefore by the definition of F_t, there exists some $b \in \mathrm{supp}(e) \cap B_t$ with $b_t \in (\{v_t\} \cup U_t) \cap F_t$ and therefore $B_{t+1} = \{a \in B_t : a_t \in (\{v_t\} \cup U_t)\}$ also contains b, and thus is nonempty. To complete the proof of the inductive hypothesis, we must show that for every $b \in B_{t+1}$, every $v \in V_L^{(t)}$ with $v \triangleleft b_t$ and $(b_1, \ldots, b_{t-1}, v, b_{t+1}, \ldots, b_r) \in$

$\mathrm{supp}(e)$ must equal $v = v_t$. But if $v \neq v_t$, then because $v \in F_t \cap V_L^{(t)} = S$ by definition, we have that $v \in S \setminus \{v_t\}$, so $b_t \in N_{G^{(t)}}(v) \subseteq N_{G^{(t)}}(S \setminus \{v_t\})$, and therefore $b_t \notin U_t$, contradicting the assumption that $b \in B_{t+1}$. Therefore the inductive hypothesis must indeed hold, as desired.

Having completed the definition of $v_1, \ldots, v_r$ and $U_1, \ldots, U_r$, we let $B = B_r \subseteq A$, and we define an r-dimensional cochain complex $\mathcal{B}^*$ with j-cochain basis $B^j = B \cap A^j$, and with j-coboundary map $\delta_j^{\mathcal{B}}(b) = \delta_j^{\mathcal{A}}(b)|_{B^{j+1}}$ (where we view $b \in \mathbb{F}_2^{B^j} \subseteq \mathbb{F}_2^{A^j}$ via the inclusion $B^j \hookrightarrow A^j$). As a point of notation, for $a \in \mathcal{A}^j$, we let $a_B = a|_{B^j}$. By definition $\mathcal{B}^*$ is the tensor product of r 1-dimensional complexes obtained from the restriction of the graphs $G^{(h)}$ to the subgraphs induced by vertices in $\{v_h\} \cup U_h$ for $h \in [r]$. As $U_h \subseteq N_{G^{(h)}}(v_h)$, these subgraphs are all star graphs, so $\mathcal{B}^*$ is precisely the cochain complex associated to r repetition codes of lengths $|U_1|, \ldots, |U_r|$ as described in Definition 4.7.

Also note that the final ($t = r$) iteration of the inductive hypothesis above implies the following:

1. $\mathrm{supp}(e) \cap B \neq \emptyset$, that is, $e_B \neq 0$, and
2. For every $a \in \mathrm{supp}(e)$ and $b \in B$ satisfying $a \triangleleft b$, then $a \in B$, that is,

$$a = (b_1, \ldots, b_{h-1}, v_h, b_{h+1}, \ldots, b_r)$$

for some $h \in [r]$.

Now define $a^0 = (v_1, \ldots, v_r) \in A^0$. Our goal is to show that a^0 satisfies one of the two conditions in Definition 4.1 (with $\mathcal{C}^* = \mathcal{A}^*$). We consider two cases separately:

1. $|\delta^{\mathcal{B}}(e_B)| > 4r\epsilon\Delta_{\max}|e_B|/\nu$: In this case, we show that condition 2 holds in Definition 4.7. Intuitively, because $\delta^{\mathcal{B}}(e_B)$ is large enough, we will show that flipping all bits of e in $\mathrm{supp}(e_B)$ reduces the weight of the image under $\delta^{\mathcal{A}}$, as we will zero out all bits in B^{i+1}, while at most flipping a small number of 0s to 1s outside of B^{i+1}.

   Specifically, define $a^i \in \mathcal{A}^i$ by $a^i = e_B \in \mathbb{F}_2^{B^i} \subseteq \mathbb{F}_2^{A^i}$. Note that if $i = 0$, we have defined a^0 twice, but both definitions agree in setting a^0 to be the indicator of the basis element $(v_1, \ldots, v_r) \in A^0$ (as by construction $e_B \neq 0$ and $B^0 = \{(v_1, \ldots, v_r)\}$).

   We now must show that

$$|\delta^{\mathcal{A}}(e + a^i)| < |\delta^{\mathcal{A}}(e)| - (1 - \nu)|\delta^{\mathcal{A}}(a^i)|.$$

   For this purpose, we have that

$$\delta^{\mathcal{A}}(e)|_{B^{i+1}} = \delta^{\mathcal{A}}(a^i)|_{B^{i+1}} = \delta^{\mathcal{B}}(a^i) = \delta^{\mathcal{B}}(e_B). \tag{5}$$

   Specifically, the latter two equalities above hold by definition, so we only need to show that $\delta^{\mathcal{A}}(e)|_{B^{i+1}} = \delta^{\mathcal{B}}(e_B)$. If instead some $b \in B^{i+1}$ had $\delta^{\mathcal{A}}(e)_b \neq \delta^{\mathcal{B}}(e_B)_b$, then there must be some $a \in \mathrm{supp}(e) \setminus B^i$ with $a \triangleleft b$. But as shown above from the inductive definition of B, the conditions $a \in \mathrm{supp}(e)$, $b \in B^{i+1}$, and $a \triangleleft b$ imply that $a \in B^i$, so $a \notin \mathrm{supp}(e) \setminus B^i$. Therefore (5) holds. Furthermore, we also must have that

$$\left|\delta^{\mathcal{A}}(a^i)|_{A^{i+1} \setminus B^{i+1}}\right| = \left|\delta^{\mathcal{A}}(e_B)|_{A^{i+1} \setminus B^{i+1}}\right| \leq 2r\epsilon\Delta_{\max}|e_B| < \frac{\nu}{2}|\delta^{\mathcal{B}}(a^i)| \leq \frac{\nu}{2}|\delta^{\mathcal{A}}(a^i)|, \tag{6}$$

   as for every $a \in \mathrm{supp}(e_B) = \mathrm{supp}(e) \cap B$, then every $b \in A^{i+1} \setminus B^{i+1}$ satisfying $a \triangleleft b$ must be of the form $b = (a_1, \ldots, a_{h-1}, u, a_{h+1}, \ldots, a_r)$ for some $h \in [r]$ with $a_h = v_h$ and some

$u \in N_{G^{(h)}}(v_h) \setminus U_h$; for each of the $\le r$ choices of h, there are $\le 2\epsilon\Delta_L^{(h)} \le 2\epsilon\Delta_{\max}$ such u by (4). Thus there are at most $|e_B| \cdot r \cdot 2\epsilon\Delta_{\max}$ elements $b \in A^{i+1} \setminus B^{i+1}$ that can satisfy $a \triangleleft b$ for some $a \in \mathrm{supp}(e_B)$, which is a necessary condition for $b \in \mathrm{supp}(\delta^{\mathcal{A}}(e_B)|_{A^{i+1}\setminus B^{i+1}})$, and therefore the first inequality in (6) holds. The second inequality in (6) follows by the assumption that $|\delta^{\mathcal{B}}(e_B)| > 4r\epsilon\Delta_{\max}|e_B|/\nu$, and the third inequality in (6) follows from (5). Combining (5) and (6), we obtain the desired bound

$$\begin{aligned}
|\delta^{\mathcal{A}}(e+a^i)| &= \left|\delta^{\mathcal{A}}(e+a^i)|_{B^{i+1}}\right| + \left|\delta^{\mathcal{A}}(e+a^i)|_{A^{i+1}\setminus B^{i+1}}\right| \\
&\le \left|\delta^{\mathcal{A}}(e)|_{A^{i+1}\setminus B^{i+1}}\right| + \left|\delta^{\mathcal{A}}(a^i)|_{A^{i+1}\setminus B^{i+1}}\right| \\
&= |\delta^{\mathcal{A}}(e)| - \left|\delta^{\mathcal{A}}(e)|_{B^{i+1}}\right| + \left|\delta^{\mathcal{A}}(a^i)|_{A^{i+1}\setminus B^{i+1}}\right| \\
&= |\delta^{\mathcal{A}}(e)| - \left|\delta^{\mathcal{A}}(a^i)|_{B^{i+1}}\right| + \left|\delta^{\mathcal{A}}(a^i)|_{A^{i+1}\setminus B^{i+1}}\right| \\
&= |\delta^{\mathcal{A}}(e)| - \left|\delta^{\mathcal{A}}(a^i)\right| + 2\left|\delta^{\mathcal{A}}(a^i)|_{A^{i+1}\setminus B^{i+1}}\right| \\
&< |\delta^{\mathcal{A}}(e)| - (1-\nu)|\delta^{\mathcal{A}}(a^i)|.
\end{aligned}$$

Specifically, the first inequality above holds by (5), the first inequality holds by the triangle inequality, the second equality holds by definition, the third equality holds by (5), the fourth equality holds by definition, and the final inequality holds by (6).

2. $|\delta^{\mathcal{B}}(e_B)| \le 4r\epsilon\Delta_{\max}|e_B|/\nu$: In this case, we show that condition 1 holds in Definition 4.7. Intuitively, because $|\delta^{\mathcal{B}}(e_B)|$ is small enough, we apply robustness of $\mathcal{B}^*$ (Lemma 4.8) to show that e_B is close to a coboundary $\delta^{\mathcal{B}}(a^{i-1})$. We then argue that we can reduce the weight of e by adding in the coboundary $\delta^{\mathcal{A}}(a^{i-1})$.

   First, if $i = 0$, then as $B^0 = \{(v_1, \dots, v_r)\}$, we have $e_B = \mathbb{1}_{(v_1,\dots,v_r)}$ and

$$|\delta^{\mathcal{B}}(e_B)| = \sum_{h=1}^{r} |N_{G^{(h)}}(v_h)| \ge r\beta\Delta_{\max}.$$

   The above inequality contradicts the assumption that $|\delta^{\mathcal{B}}(e_B)| \le 4r\epsilon\Delta_{\max}|e_B|/\nu = 4r\epsilon\Delta_{\max}/\nu$ because $\epsilon < \beta\nu/4$ by (3). Therefore we may assume that $i \ge 1$.

   Recall that $\mathcal{B}$ is by definition the product of r 1-dimensional cochain complexes associated to repetition codes of lengths at most $\Delta_{\max}$ and at least $|U_h| \ge (1-2\epsilon)\Delta_L^{(h)} \ge (1-2\epsilon)\beta\Delta_{\max} \ge (\beta/2)\Delta_{\max}$ (as $\epsilon \le 1/4$ by (3)). Therefore Lemma 4.8 implies that this tuple of repetition codes is $\kappa\Delta_{\max}$-robust in the sense of Definition 4.7 for $\kappa = \kappa(r, \beta/2)$ as defined above.

   Applying the definition of robustness to e_B, there exists some i-coboundary $e'_B \in B^i(\mathcal{B})$ such that

$$\kappa\Delta_{\max}|e_B + e'_B| \le |\delta^{\mathcal{B}}(e_B)| \le 4r\epsilon\Delta_{\max}|e_B|/\nu,$$

   that is,

$$|e_B + e'_B| \le \frac{4r\epsilon}{\kappa\nu}|e_B|.$$

   Now applying robustness to any element of $\mathcal{B}^{i-1}$ whose coboundary equals e'_B, we obtain some $(i-1)$-cochain $a^{i-1} \in \mathcal{B}^{i-1} \subseteq \mathcal{A}^{i-1}$ with $\delta^{\mathcal{B}}(a^{i-1}) = e'_B$ and

$$|a^{i-1}| \le \frac{1}{\kappa\Delta_{\max}}|e'_B| \le \frac{1}{\kappa\Delta_{\max}}\left(\frac{4r\epsilon}{\kappa\nu} + 1\right)|e_B|$$

   By definition $|e'_B| \ge (1 - 4r\epsilon/\kappa\nu)|e_B| > 0$ because $e_B \ne 0$ and $\epsilon \le \kappa\nu/16r$, so $a^{i-1} \ne 0$. As a point of notation, if $i = 1$ then it appears we have overloaded the variable $a^0 = a^{i-1}$, but

indeed in this case a^{i-1} must equal a nonzero element of $\mathcal{B}^0$, and the unique such element is $a^0 = \mathbb{1}_{(v_1,\dots,v_r)}$.

Our goal is to show that

$$\left|e + \delta^{\mathcal{A}}(a^{i-1})\right| < |e|.$$

For this purpose, we have that

$$\left|e + \delta^{\mathcal{A}}(a^{i-1})|_{B^i}\right| = |e_B + e'_B| \leq \frac{4r\epsilon}{\kappa\nu}|e_B|$$

and

$$\left|(e + \delta^{\mathcal{A}}(a^{i-1}))|_{A^i \setminus B^i} - e|_{A^i \setminus B^i}\right| = \left|\delta^{\mathcal{A}}(a^{i-1})_{A^i \setminus B^i}\right| \leq 2r\epsilon\Delta_{\max}|a^{i-1}|,$$

where the final inequality above holds by the definition of B, by similar reasoning used to show the first inequality in (6) above. Combining the above inequalities then gives the desired bound

$$\begin{aligned}
\left|e + \delta^{\mathcal{A}}(a^{i-1})\right| &= \left|e + \delta^{\mathcal{A}}(a^{i-1})|_{B^i}\right| + \left|e + \delta^{\mathcal{A}}(a^{i-1})|_{A^i \setminus B^i}\right| \\
&\leq \frac{4r\epsilon}{\kappa\nu}|e_B| + \left|e|_{A^i \setminus B^i}\right| + 2r\epsilon\Delta_{\max}|a^{i-1}| \\
&= \frac{4r\epsilon}{\kappa\nu}|e_B| + |e| - |e_B| + 2r\epsilon\Delta_{\max}|a^{i-1}| \\
&\leq |e| - \left(1 - \frac{4r\epsilon}{\kappa\nu} - \frac{2r\epsilon}{\kappa}\left(\frac{4r\epsilon}{\kappa\nu} + 1\right)\right)|e_B| \\
&\leq |e| - \left(1 - \frac{4r\epsilon}{\kappa\nu}\left(\frac{4r\epsilon}{\kappa\nu} + 2\right)\right)|e_B| \\
&< |e|,
\end{aligned}$$

where the final inequality above holds because $\epsilon \leq \kappa\nu/16r$ by definition from (3).

□

We now apply the error-flip decoder from Lemma 4.10 to construct a syndrome-flip decoder for Proposition 4.6.

Lemma 4.11. *For every $r \in \mathbb{N}$, $0 < \beta \leq 1$, let $\nu = 1/10$ and define $\epsilon = \epsilon(r, \beta, \nu)$ as in Lemma 4.10. Then define $\mu, \Delta_{\max}, G^{(h)}, \Delta_L^{(h)}, \mathcal{C}^*$ as in Proposition 4.6. Then $\mathcal{A}^*$ has an (m, γ)-small-set syndrome-flip decoder at every level $0 \leq i \leq r-1$ for $m = \min_{h\in[r]} \mu|V_L^{(h)}|/(r\Delta_{\max}^{r+1} + 1)$ and $\gamma = \nu$.*

Proof. Fix some $0 \leq i \leq r-1$, some nonzero $e \in \mathcal{A}^i$ of weight $|e| \leq m$, and some $f \in \mathcal{A}^{i+1}$ of weight $|f| \leq \gamma|\delta(e)|$. Our goal is to show that there exists some basis element $\tilde{a}^0 \in A^0$ and some cochain $\tilde{a}^i \in \mathcal{A}^i$ with $\tilde{a}^0 \preceq \tilde{a}^i$ and $|\delta(e + \tilde{a}^i) + f| < |\delta(e) + f|$. Note that in this proof, we let $\delta = \delta^{\mathcal{A}}$ denote the coboundary map for $\mathcal{A}^*$.

For this purpose, we first consider running `SSFlipErr`$(e; i, \mathcal{A}^*)$ from Algorithm 2. At each execution of line 11, there are potentially multiple valid choices of c^0, c^i; assume that we choose whichever pair yields the greatest value of $(|\delta(e)| - |\delta(e+c^i)|)/|\delta(c^i)|$. Then let $c_1^{i-1}, \dots, c_{t_{i-1}}^{i-1} \in \mathcal{A}^{i-1}$ and $c_1^i, \dots, c_{t_i}^i \in \mathcal{A}^i$ be the sequence of all values for c^{i-1} and c^i chosen in line 9 and line 11 respectively, across all runs of the while loop.

We begin with the following claim, which follows by the assumption that $|e| \leq m$.

Claim 4.12. *When we run* `SSFlipErr`$(e; i, \mathcal{A}^*)$ *as described above, the output is some valid* $a^{i-1} \in \mathcal{A}^{i-1}, a^i \in \mathcal{A}^i$ *with* $e = a^i + \delta(a^{i-1})$*, and the while loop must execute for* $t_{i-1} + t_i$ *iterations with* $t_{i-1} \le (r\Delta_{\max}^{r+1} + 1)|e|$ *and* $t_i \le |\delta(e)| \le r\Delta_{\max}|e|$*. Furthermore, for every* $j \in [t_i]$ *we have*

$$\left|\delta\left(e + \sum_{\ell=1}^{j-1} c_\ell^i\right)\right| - \left|\delta\left(e + \sum_{\ell=1}^{j} c_\ell^i\right)\right| > (1-\nu)|\delta(c_j^i)|. \tag{7}$$

Proof. Recall that the algorithm initializes $a^{i-1} \leftarrow 0$ and $a^i \leftarrow 0$. Every time the if statement in line 11 is satisfied, we decrease $|\delta(e + a^i)|$ by at least 1, so this if statement can be satisfied at most

$$t_i \le |\delta(e)| \le r\Delta_{\max}|e|$$

times. Each such update from this if statement can increase $|e + a^i + \delta(a^{i-1})|$ by at most $|c^i| \le \Delta_{\max}^r$. Meanwhile, every time the if statement in line 9 is satisfied, we decrease $|e + a^i + \delta(a^{i-1})|$ by at least 1. Therefore the if statement in line 9 can be satisfied at most

$$t_{i-1} \le \Delta_{\max}^r \cdot t_i + |e| \le (r\Delta_{\max}^{r+1} + 1)|e|$$

times, and the RHS above is also an upper bound on the maximum value of $|e + a^i + \delta(a^{i-1})|$ at any point in the algorithm's execution. Therefore because $|e| \le m = \min_{h\in[r]} \mu|V_L^{(h)}|/(r\Delta_{\max}^{r+1} + 1)$, we always have $|e + a^i + \delta(a^{i-1})| \le \min_{h\in[r]} \mu|V_L^{(h)}|$. Lemma 4.10 then implies that one of the two if statements is satisfied in every iteration of the while loop, and that when the if statement in line 11 is satisfied, we have $|\delta(e + a^i)| - |\delta(e + a^i + c^i)| < (1-\nu)|\delta(c^i)|$, and hence (7) holds. It also then follows that the algorithm successfully returns some final values a^{i-1}, a^i satisfying $e = a^i + \delta(a^{i-1})$ (and does not return FAIL). □

Therefore `SSFlipErr`$(e; i, \mathcal{A}^*)$ returns $a^i = \sum_{j=1}^{t_i} c_j^i$ with

$$\delta(e) = \delta(a^i) = \sum_{j=1}^{t_i} \delta(c_j^i) \tag{8}$$

and

$$|\delta(e)| = \sum_{j=1}^{t_i} \left(\left|\delta\left(e + \sum_{\ell=1}^{j-1} c_\ell^i\right)\right| - \left|\delta\left(e + \sum_{\ell=1}^{j} c_\ell^i\right)\right|\right) > (1-\nu)\sum_{j=1}^{t_i} |\delta(c_j^i)|. \tag{9}$$

For each $j \in [t_i]$, let $U_j = \mathrm{supp}(\delta(e)) \cap \mathrm{supp}(\delta(c_j^i)) \setminus \bigcup_{j'\neq j} \mathrm{supp}(\delta(c_{j'}^i))$ denote the components that lie in the support of $\delta(e)$ and $\delta(c_j^i)$ but not in any other $\delta(c_{j'}^i)$. By (8), we have

$$\mathrm{supp}(\delta(e)) \subseteq \bigcup_{j\in[t_i]} \mathrm{supp}(\delta(c_j^i)).$$

Therefore across all the sets $\mathrm{supp}(\delta(c_j^i))$ for $j \in [t_i]$, there are $|\delta(e)| > (1-\nu)\sum_{j\in[t_i]} |\delta(c_j^i)|$ distinct elements that lie in $\mathrm{supp}(e)$, plus less than $\nu \sum_{j\in[t_i]} |\delta(c_j^i)|$ additional elements that either do not lie in $\mathrm{supp}(e)$, or may equal some of these $|\delta(e)|$ distinct elements. Thus every element of $\mathrm{supp}(e)$ lies in some set $\mathrm{supp}(c_j^i)$, and $< \nu \sum_{j\in[t_i]} |\delta(c_j^i)|$ of these elements lie in more than one such set, so by (9),

$$\sum_{j\in[t_i]} |U_j| > (1-2\nu) \sum_{j\in[t_i]} |\delta(c_j^i)|.$$

Thus as by assumption $|f| \leq \gamma |\delta(e)| \leq \gamma \sum_{j \in [t_i]} |\delta(c_j^i)|$ with $\gamma = \nu$, we have

$$\sum_{j \in [t_i]} |U_j \setminus \mathrm{supp}(f)| \geq \sum_{j \in [t_i]} |U_j| - |f| > (1 - 3\nu) \sum_{j \in [t_i]} |\delta(c_j^i)|.$$

Therefore there must exist some $\tilde{j} \in [t_i]$ with

$$|U_{\tilde{j}} \setminus \mathrm{supp}(f)| > (1 - 3\nu) |\delta(c_{\tilde{j}}^i)|. \tag{10}$$

Letting $\tilde{a}^i = c_{\tilde{j}}^i$, then by definition there exists some basis element $\tilde{a}^0 \in A^0$ with $\tilde{a}^0 \preceq \tilde{a}^i$; specifically, $\tilde{a}^0$ equals the value of c^0 during the execution of line 11 in Algorithm 2 in which $c^i = c_{\tilde{j}}^i$. Furthermore,

$$\begin{aligned}
|\delta(e + \tilde{a}^i) + f| &= \left|\delta(e + \tilde{a}^i) + f|_{U_{\tilde{j}} \setminus \mathrm{supp}(f)}\right| + \left|\delta(e + \tilde{a}^i) + f|_{A^i \setminus (U_{\tilde{j}} \setminus \mathrm{supp}(f))}\right| \\
&= \left|\delta(e + \tilde{a}^i) + f|_{A^i \setminus (U_{\tilde{j}} \setminus \mathrm{supp}(f))}\right| \\
&\leq |\delta(e) + f| - \left|\delta(e) + f|_{U_{\tilde{j}} \setminus \mathrm{supp}(f)}\right| + \left|\delta(\tilde{a}^i)|_{A^i \setminus (U_{\tilde{j}} \setminus \mathrm{supp}(f))}\right| \\
&\leq |\delta(e) + f| - |U_{\tilde{j}} \setminus \mathrm{supp}(f)| + (|\delta(\tilde{a}^i)| - |U_{\tilde{j}} \setminus \mathrm{supp}(f)|) \\
&< |\delta(e) + f| - (1 - 6\nu) |\delta(\tilde{a}^i)|.
\end{aligned}$$

where the second equality above holds because by construction $\delta(e)_b = \delta(\tilde{a}^i)_b = 1$ and $f_b = 0$ for every $b \in U_{\tilde{j}} \setminus \mathrm{supp}(f)$, the first inequality holds by the triangle inequality, the second inequality holds because by definition $U_{\tilde{j}} \subseteq \mathrm{supp}(\delta(e)) \cap \mathrm{supp}(\tilde{a}^i)$ and $U_{\tilde{j}} \setminus \mathrm{supp}(f) \subseteq \mathrm{supp}(\tilde{a}^i)$, and the third inequality holds by (10). Thus as by definition $\nu = 1/10$, we conclude that $|\delta(e + \tilde{a}^i) + f| < |\delta(e) + f|$, as desired. □

We have now completed the proof of Proposition 4.6:

Proof of Proposition 4.6. The result follows immediately from Lemma 4.10 and Lemma 4.11. □

5 Downwards Code Switching Gadget via Direct Measurement

This section presents our gadget for switching from an r-dimensional code to an $(r-1)$-dimensional code by directly measuring out some of the code's data qubits, and then applying an appropriate Pauli correction. We introduce notation and state our result in Section 5.1. The gadget is given in Algorithm 3. We describe the noiseless execution of this gadget in Section 5.2, and then we prove fault-tolerance for the noisy execution in Section 5.3.

A complementary gadget for switching up from an $(r-1)$-dimensional code to an r-dimensional code is given in Section 6. As described in Section 1.1, this upwards code switching performs logical teleportation using logical bell pairs between $(r-1)$- and r-dimensional codeblocks. We require the downwards code switching gadget, as well as the $CNOT$ and measurement gadgets in Section 9 below, in order to construct these bell pairs and perform the teleportation.

5.1 Result Statement

In this section, we state our result providing a gadget for switching to a lower-dimensional code. We will first need the following notation.

For $r \in \mathbb{N}$, let $\mathcal{A}$ be an $(r-1)$-dimensional cochain complex. Let[9] $\mathcal{M}^{\mathcal{A},*}$ be the $(r-1)$-dimensional cochain complex with $\mathcal{M}^{\mathcal{A},i} = \mathbb{F}_2^{\dim(H^i(\mathcal{A}))}$ and $\delta^{\mathcal{M}^{\mathcal{A}}} = 0$, and let $\mathrm{Enc}^{\mathcal{A}} : \mathcal{M}^{\mathcal{A},*} \to \mathcal{A}^*$ be a cochain map inducing an isomorphism on cohomology[10].

Let $\mathcal{B}^*$ be a 1-dimensional cochain complex. Fix information sets $M^{\mathcal{B},0} \subseteq B^0$ and $M^{\mathcal{B},1} \subseteq B^1$ for $\ker(\delta^{\mathcal{B}})$ and $\ker(\partial^{\mathcal{B}})$ respectively, let $\mathcal{M}^{\mathcal{B},*} = (\mathbb{F}_2^{M^{\mathcal{B},0}} \xrightarrow{0} \mathbb{F}_2^{M^{\mathcal{B},1}})$, and let $\mathrm{Enc}^{\mathcal{B}} : \mathcal{M}^{\mathcal{B},*} \to \mathcal{B}^*$ be the cochain map defined in Lemma 3.33. Also fix a subset $L^{\mathcal{B},1} \subseteq M^{\mathcal{B},1}$, and let $\bar{L}^{\mathcal{B},1} = B^1 \setminus L^{\mathcal{B},1}$. We let $\mathcal{B}^*_{\bar{L}} = (\mathbb{F}_2^{B^0} \xrightarrow{\delta^{\mathcal{B}}_{\bar{L}}} \mathbb{F}_2^{\bar{L}^{\mathcal{B},1}})$ denote the 1-dimensional cochain complex with coboundary map given by $\delta^{\mathcal{B}}_{\bar{L}}(b) = \delta^{\mathcal{B}}(b)|_{\bar{L}^{\mathcal{B},1}}$, and we similarly let $\mathcal{M}^{\mathcal{B},*}_{\bar{L}} = (\mathbb{F}_2^{M^{\mathcal{B},0}} \xrightarrow{0} \mathbb{F}_2^{M^{\mathcal{B},1} \setminus L^{\mathcal{B},1}})$.

Because $L^{\mathcal{B},1}$ is extendable for $\ker(\partial^{\mathcal{B}})$, no element of $B^1(\mathcal{B}) = \mathrm{im}(\delta^{\mathcal{B}})$ is supported inside $L^{\mathcal{B},1}$, and therefore $Z^0(\mathcal{B}_{\bar{L}}) = \ker(\delta^{\mathcal{B}}) = Z^0(\mathcal{B})$. Furthermore, the cosets $\mathbb{1}_x + B^1(\mathcal{B}_{\bar{L}})$ for $x \in M^{\mathcal{B},1} \setminus L^{\mathcal{B},1}$ must be linearly independent, as any nontrivial linear dependence among them would yield a nonzero coboundary in $B^1(\mathcal{B})$ supported inside $M^{\mathcal{B},1}$. These cosets must also span all of $\mathcal{B}^1_{\bar{L}}/B^1(\mathcal{B}_{\bar{L}})$ because the cosets $\mathbb{1}_x + B^1(\mathcal{B})$ for $x \in M^{\mathcal{B},1}$ span all of $\mathcal{B}^1/B^1(\mathcal{B})$ by Lemma 3.8. It follows by Lemma 3.8 that $M^{\mathcal{B},0}$ and $M^{\mathcal{B},1} \setminus L^{\mathcal{B},1}$ are information sets for $Z^0(\mathcal{B}_{\bar{L}})$ and $Z_1(\mathcal{B}_{\bar{L}})$ respectively. We then let $\mathrm{Enc}^{\mathcal{B}}_{\bar{L}} : \mathcal{M}^{\mathcal{B},*}_{\bar{L}} \to \mathcal{B}^*_{\bar{L}}$ be the encoding map for $\mathcal{B}^*_{\bar{L}}$ given by Lemma 3.33. It follows by definition that for every $i \in \{0,1\}$ and $x \in \mathcal{M}^{\mathcal{B},i}$,

$$\mathrm{Enc}^{\mathcal{B}}(x)|_{B^i_{\bar{L}}} = \mathrm{Enc}^{\mathcal{B}}_{\bar{L}}(x|_{M^{\mathcal{B},i}_{\bar{L}}}). \tag{11}$$

Let $\mathcal{C}^* = \mathcal{A}^* \otimes \mathcal{B}^*$, let $\mathcal{M}^{\mathcal{C},*} = \mathcal{M}^{\mathcal{A},*} \otimes \mathcal{M}^{\mathcal{B},*}$, and let $\mathrm{Enc}^{\mathcal{C}} = \mathrm{Enc}^{\mathcal{A}} \otimes \mathrm{Enc}^{\mathcal{B}} : \mathcal{M}^{\mathcal{C},*} \to \mathcal{C}^*$, which by the Künneth formula induces an isomorphism on cohomology and hence provides a CSS encoding map for codes associated to $\mathcal{C}^*$.

Let $\mathcal{C}^*_{\bar{L}} = \mathcal{A}^* \otimes \mathcal{B}^*_{\bar{L}}$. For some $2 \le i \le r-1$, assume that $\mathcal{C}^*_{\bar{L}}$ has a $(m,0)$-small-set flip decoder at level i. We similarly define $\mathcal{M}^{\mathcal{C},*}_{\bar{L}} = \mathcal{M}^{\mathcal{A},*} \otimes \mathcal{M}^{\mathcal{B},*}_{\bar{L}}$ and $\mathrm{Enc}^{\mathcal{C}}_{\bar{L}} = \mathrm{Enc}^{\mathcal{A}} \otimes \mathrm{Enc}^{\mathcal{B}}_{\bar{L}}$. Then by (11), for every $x \in \mathcal{M}^{\mathcal{C},i}$,

$$\mathrm{Enc}^{\mathcal{C}}(x)|_{C^i_{\bar{L}}} = \mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x|_{M^{\mathcal{C},i}_{\bar{L}}}). \tag{12}$$

Let Q_{in} be the $[[n_{\mathrm{in}}, k_{\mathrm{in}}]]$ CSS code associated to level i of $\mathcal{C}^*$, and let Q_{out} be the $[[n_{\mathrm{out}}, k_{\mathrm{out}}]]$ CSS code consisting of $|L^{\mathcal{B},1}|$ copies of the CSS code associated to level $i-1$ of $\mathcal{A}^*$. Given some graph $G_{\mathrm{run}} = (V_{\mathrm{run}}, E_{\mathrm{run}})$ that contains $G^{\mathcal{C}}_{i-1,i,i+1} \subseteq G_{\mathrm{run}}$ as a subgraph and some

$$\eta_{\mathrm{run}} \le \min\{m,\ d_i(\mathcal{C}_{\bar{L}})\}$$
$$\gamma_{\mathrm{run}} \le 1/150w^7,$$

let

$$\mathcal{E}_{\mathrm{run}} = \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{run}},\ \gamma_{\mathrm{run}})$$
$$\mathcal{E}_{\mathrm{out}} = \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{run}},\ 300w^7\gamma_{\mathrm{run}}),$$

[9] Here we write $M^{\mathcal{A},*}$ as a shorthand for $(M^{\mathcal{A}})^*$ (and similarly for $M^{\mathcal{B}}$, etc.).

[10] Such a cochain map can be constructed by mapping each basis element in $M^{\mathcal{A},i}$ to some basis element of $H^i(\mathcal{A})$.

and define the decorated codes[11]

$$D_{\text{in}} = (Q_{\text{in}},\ \text{Enc}^{\mathcal{C}},\ \mathcal{E}_{\text{run}})$$
$$D_{\text{out}} = (Q_{\text{out}},\ (\text{Enc}^{\mathcal{A}})^{\sqcup L^{\mathcal{B},1}},\ \mathcal{E}_{\text{out}})$$

In these decorated codes, the $n_{\text{in}} = |C^i|$ physical qubits of Q_{in} are naturally associated with the set $C^i \subseteq C^{i-1} \sqcup C^i \sqcup C^{i+1} = V(G^{\mathcal{C}}_{i-1,i,i+1}) \subseteq V_{\text{run}}$. The $n_{\text{out}} = |A^{i-1}| \cdot |L^{\mathcal{B},1}|$ physical qubits of Q_{out} are then naturally associated to the subset $A^{i-1} \times L^{\mathcal{B},1} \subseteq A^{i-1} \times B^1 \subseteq C^i \subseteq V_{\text{run}}$.

The k_{in} logical qubits of Q_{in} are naturally associated to the set $M^{\mathcal{C},i} = (M^{\mathcal{A},i} \times M^{\mathcal{B},0}) \sqcup (M^{\mathcal{A},i-1} \times M^{\mathcal{B},1})$. The k_{out} logical qubits of Q_{out} are naturally associated to the set $M^{\mathcal{A},i-1} \times L^{\mathcal{B},1} \subseteq M^{\mathcal{A},i-1} \times M^{\mathcal{B},1} \subseteq M^{\mathcal{C},i}$. Under these associations, we let $\bar{\mathcal{O}} : \mathbb{C}^{2^{k_{\text{in}}} \times 2^{k_{\text{in}}}} \to \mathbb{C}^{2^{k_{\text{out}}} \times 2^{k_{\text{out}}}}$ be the map that simply discards (i.e. traces out) all $k_{\text{in}} - k_{\text{out}}$ logical qubits in $M^{\mathcal{C},i} \setminus (M^{\mathcal{A},i-1} \times L^{\mathcal{B},1})$. That is, $\bar{\mathcal{O}}(\rho) = \text{tr}_{M^{\mathcal{C},i} \setminus (M^{\mathcal{A},i-1} \times L^{\mathcal{B},1})}(\rho)$

Proposition 5.1. *Define all variables as above in Section 5.1. Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, \mathcal{E}_{\text{run}}^{\sqcup T}), D_{\text{in}}, D_{\text{out}})$ for $\bar{\mathcal{O}}$, where $\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = n_{\text{in}}$ and time $T = 2$.*

The gadget $\mathcal{Q}$ in Proposition 5.1 is given in Algorithm 3.

Remark 5.2. *While we have stated Proposition 5.1 and presented Algorithm 3 to perform Z-basis measurements, an analogous result holds where we dualize to a chain complex, and exchange the roles of the Pauli X and Z bases everywhere in the construction and analysis.*

Algorithm 3: Gadget in Proposition 5.1 for code switching from $\mathcal{C}^* = \mathcal{A}^* \otimes \mathcal{B}^*$ down to multiple copies of $\mathcal{A}^*$. All variables are defined as in Section 5.1. The physical qubits of the circuit are labeled by the set $C^i = (A^i \times B^0) \sqcup (A^{i-1} \times B^1)$. Below, we state the ideal input and output under a noisless execution. Note that $\delta^{\mathcal{C}}(c) \in \mathcal{C}^i$ in line 5 is well-defined because $c \in \mathcal{C}^{i-1}_{\bar{L}} \subseteq \mathcal{C}^{i-1}$ via the inclusion $C^{i-1}_{\bar{L}} \subseteq C^{i-1}$.

Input : $\sigma = \text{Enc}^{\mathcal{C},i}(\rho)$
Output: $(\text{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(\text{tr}_{M^{\mathcal{C},i} \setminus (M^{\mathcal{A},i-1} \times L^{\mathcal{B},1})}(\rho))$

1 **Function** `SwitchDown`$(\sigma;\ i,\ \mathcal{A}^*,\ \mathcal{B}^*)$**:**
2 Measure Pauli Z on all qubits in $C^i_{\bar{L}} = (A^i \times B^0) \sqcup (A^{i-1} \times \bar{L}^{\mathcal{B},1})$, and let $z \in \mathcal{C}^i_{\bar{L}}$ be the resulting outcome
3 Run `SSFlipSyn`$(\delta^{\mathcal{C}_{\bar{L}}}(z); i, \mathcal{C}^*_{\bar{L}})$ from Algorithm 2, and let $a^i \in \mathcal{C}^i_{\bar{L}}$ be the output
4 Run Gaussian elimination to find some $c_{\bar{L}} \in \mathcal{C}^{i-1}_{\bar{L}}$ with $z + a^i + \delta^{\mathcal{C}_{\bar{L}}}(c_{\bar{L}}) \in \text{im}(\text{Enc}^{\mathcal{C},i-1}_{\bar{L}})$
5 Apply $X^{\delta^{\mathcal{C}}(c_{\bar{L}})|_{A^{i-1} \times L^{\mathcal{B},1}}}$ to σ
6 **return** $\text{tr}_{C^i \setminus (A^{i-1} \times L^{\mathcal{B},1})}(\sigma)$

5.2 Noiseless Execution

In this section, as a warm-up for Proposition 5.1, we show that a noiseless execution of Algorithm 3 performs the desired code switching.

[11] Here we write $(\text{Enc}^{\mathcal{A}})^{\sqcup L^{\mathcal{B},1}}$ (or simply $(\text{Enc}^{\mathcal{A}})^{L^{\mathcal{B},1}}$) to denote $\text{Enc}^{\mathcal{A}}$ applied to $|L^{\mathcal{B},1}|$ disjoint copies of Q_{out}, labeled by elements of $L^{\mathcal{B},1}$.

Lemma 5.3. *Define all variables as in Section 5.1. For every $\ell \in \mathbb{N}$ and every $\rho \in \mathbb{C}^{(\mathcal{M}^{\mathcal{C}} \otimes \mathbb{F}_2^\ell) \times (\mathcal{M}^{\mathcal{C}} \otimes \mathbb{F}_2^\ell)}$, the output of* `SwitchDown`$(\mathrm{Enc}^{\mathcal{C},i} \otimes \mathcal{I}_\ell(\rho); i, \mathcal{A}^*, \mathcal{B}^*)$ *in Algorithm 3 is proportional to* $(\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}} \otimes \mathcal{I}_\ell(\mathrm{tr}_{M^{\mathcal{C},i} \setminus (M^{\mathcal{A},i-1} \times L^{\mathcal{B},1})}(\rho))$.

Proof. To begin, we expand ρ as

$$\rho = \sum_{x,x' \in \mathcal{M}^{\mathcal{C},i}} \rho_{x,x'} |x\rangle \langle x'| \otimes \nu_{x,x'} \tag{13}$$

for coefficients $\rho_{x,x'} \in \mathbb{C}$ and reference-system matrices $\nu_{x,x'} \in \mathbb{C}^{2^\ell \times 2^\ell}$. Then[12]

$$\mathrm{Enc}^{\mathcal{C}}(\rho) = \mathrm{Enc}^{\mathcal{C}} \rho \mathrm{Enc}^{\mathcal{C}\dagger} = \sum_{x,x' \in \mathcal{M}^{\mathcal{C},i},\ b,b' \in B^i(\mathcal{C})} \frac{\rho_{x,x'}}{|B^i(\mathcal{C})|} \left|\mathrm{Enc}^{\mathcal{C}}(x) + b\right\rangle \left\langle \mathrm{Enc}^{\mathcal{C}}(x) + b'\right|.$$

The Pauli Z measurements in line 2 then collapse the above state to

$$\sum_{z \in \mathcal{C}^i_{\bar{L}}} |z\rangle\langle z| \otimes \sum_{x,x',b,b'} \rho_{x,x'} \left|\mathrm{Enc}^{\mathcal{C}}(x) + b|_{A^{i-1} \times L^{\mathcal{B},1}}\right\rangle \left\langle \mathrm{Enc}^{\mathcal{C}}(x') + b'|_{A^{i-1} \times L^{\mathcal{B},1}}\right| \otimes \nu_{x,x'}, \tag{14}$$

where the second sum above is over all $x, x' \in \mathcal{M}^{\mathcal{C},i}$ and $b, b' \in B^i(\mathcal{C})$ satisfying

$$\mathrm{Enc}^{\mathcal{C}}(x) + b|_{C^i_{\bar{L}}} = z = \mathrm{Enc}^{\mathcal{C}}(x') + b'|_{C^i_{\bar{L}}}. \tag{15}$$

Note that here we only track the state up to global scalars/phases (such as the dropped factor of $1/|B^i(\mathcal{C})|$ above). Now by (12),

$$\mathrm{Enc}^{\mathcal{C}}(x) + b|_{C^i_{\bar{L}}} = \mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x|_{M^{\mathcal{C},i}_{\bar{L}}}) + (b|_{C^i_{\bar{L}}}).$$

The following claim shows that the decomposition of z on the RHS above is unique.

Claim 5.4. *For every $z \in Z^i(\mathcal{C}_{\bar{L}})$, there is a unique choice of $x_{\bar{L}} \in \mathcal{M}^{\mathcal{C},i}_{\bar{L}}$ and $b_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})$ for which*

$$z = \mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}}) + b_{\bar{L}}.$$

Furthermore, for $z \in \mathcal{C}^i_{\bar{L}} \setminus Z^i(\mathcal{C}_{\bar{L}})$, there is no such $x_{\bar{L}}, b_{\bar{L}}$.

Proof. Because $\mathrm{Enc}^{\mathcal{C}}_{\bar{L}} : \mathcal{M}^{\mathcal{C},*}_{\bar{L}} \to \mathcal{C}^*_{\bar{L}}$ is a chain map inducing an isomorphism on cohomology, and $\mathcal{M}^{\mathcal{C},*}_{\bar{L}}$ has coboundary map $\delta^{\mathcal{C}}_{\bar{L}} = 0$, it follows that $\mathrm{Enc}^{\mathcal{C}}_{\bar{L}}$ maps every nonzero input to a nontrivial cohomology class in $H^*(\mathcal{C}_{\bar{L}})$, and every such cohomology class has a unique representative with such a (unique) preimage under $\mathrm{Enc}^{\mathcal{C}}_{\bar{L}}$. Thus the claim holds. □

Claim 5.4 implies that every term with a nonzero contribution to the sum in (14) has $z \in Z^i(\mathcal{C}_{\bar{L}})$, so line 3 has a trivial decoding problem and computes $a^i = 0$.

Furthermore, Claim 5.4 implies that for (15) to hold, we must have $z = \mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}}) + b_{\bar{L}} \in Z^i(\mathcal{C}_{\bar{L}})$ for $x_{\bar{L}} := x|_{M^{\mathcal{C},i}_{\bar{L}}} = x'|_{M^{\mathcal{C},i}_{\bar{L}}}$ and $b_{\bar{L}} := b|_{C^i_{\bar{L}}} = b'|_{C^i_{\bar{L}}}$. Fix an arbitrary $c_{\bar{L}}(b_{\bar{L}}) \in \mathcal{C}^{i-1}_{\bar{L}}$ with $\delta^{\mathcal{C}}_{\bar{L}}(c_{\bar{L}}(b_{\bar{L}})) = b_{\bar{L}}$. Then for (15) to hold for a given z, we must have $x = (x_{\bar{L}}, x_L)$ for some $x_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$,

[12] Note that here (and throughout this section) we abuse notation by letting $\mathrm{Enc}^{\mathcal{C}} = \mathrm{Enc}^{\mathcal{C},i}$ denote the encoding chain map, its induced map on cohomology, the associated encoding isometry, and the associated encoding channel. We also sometimes write $\mathrm{Enc}^{\mathcal{C}}$ as a shorthand for $\mathrm{Enc}^{\mathcal{C}} \otimes \mathcal{I}_\ell$. The meaning will always be made clear from the argument.

and $b = \delta^{\mathcal{C}}(c_{\bar{L}}(b_{\bar{L}})) + b_L$ for some $b_L \in B^{i-1}(\mathcal{A}) \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$. Note that here we can apply $\delta^{\mathcal{C}}$ to $c_{\bar{L}}(b_{\bar{L}})$ using the natural inclusion $\mathcal{C}_{\bar{L}}^{i-1} \subseteq \mathcal{C}_{\bar{L}}^{i-1} \oplus (\mathcal{A}^{i-2} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}) = \mathcal{C}^{i-1}$; the term b_L is the image under $\delta^{\mathcal{C}}$ of some $c_L \in \mathcal{A}^{i-2} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$. By the same reasoning, (15) also implies that $x' = (x_{\bar{L}}, x'_L)$ for some $x'_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$ and $b = \delta^{\mathcal{C}}(c_{\bar{L}}(b_{\bar{L}})) + b'_L$ for some $b'_L \in B^{i-1}(\mathcal{A}) \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$.

Thus (14) can be expressed as

$$\begin{aligned}
&\sum_{x_{\bar{L}} \in \mathcal{M}_{\bar{L}}^{\mathcal{C},i},\, b_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})} \left|\mathrm{Enc}_{\bar{L}}^{\mathcal{C}}(x_{\bar{L}}) + b_{\bar{L}}\right\rangle \left\langle \mathrm{Enc}_{\bar{L}}^{\mathcal{C}}(x_{\bar{L}}) + b_{\bar{L}}\right| \\
&\otimes \sum_{x_L, x'_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}},\, b_L, b'_L \in B^{i-1}(\mathcal{A}) \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}} \rho_{x,x'} \left|(\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x_L) + b_L + (\delta^{\mathcal{C}}(c_{\bar{L}}(b_{\bar{L}}))|_{A^{i-1} \times L^{\mathcal{B},1}})\right\rangle \\
&\qquad\qquad \left\langle (\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x'_L) + b'_L + (\delta^{\mathcal{C}}(c_{\bar{L}}(b_{\bar{L}}))|_{A^{i-1} \times L^{\mathcal{B},1}})\right| \\
&\otimes \nu_{x,x'}.
\end{aligned} \tag{16}$$

The $c_{\bar{L}}$ computed in line 4 of Algorithm 3 must have the same image under $\delta^{\mathcal{C}}$ as $c_{\bar{L}}(b_{\bar{L}})$. Specifically, by definition we have $z + \delta^{\mathcal{C}_{\bar{L}}}(c_{\bar{L}}),\ z + \delta^{\mathcal{C}_{\bar{L}}}(c_{\bar{L}}) \in \mathrm{im}(\mathrm{Enc}_{\bar{L}}^{\mathcal{C},i-1})$. Because $\mathrm{Enc}_{\bar{L}}^{\mathcal{C},i-1}$ induces an isomorphism on cohomology, every nonzero element of $\mathrm{im}(\mathrm{Enc}_{\bar{L}}^{\mathcal{C},i-1})$ lies in a nontrivial cohomology class, and hence outside of $\mathrm{im}(\delta^{\mathcal{C}_{\bar{L}}})$. Thus $\delta^{\mathcal{C}_{\bar{L}}}(c_{\bar{L}}) = \delta^{\mathcal{C}_{\bar{L}}}(c_{\bar{L}}(b_{\bar{L}}))$, so if $\delta^{\mathcal{C}}(c_{\bar{L}}) \neq \delta^{\mathcal{C}}(c_{\bar{L}}(b_{\bar{L}}))$, then $\delta^{\mathcal{C}}(c_{\bar{L}} + c_{\bar{L}}(b_{\bar{L}})) \neq 0$ must be supported entirely outside of $C_{\bar{L}}^i$. But such an element cannot exist because $L^{\mathcal{B},1}$ is extendable for $\ker(\partial^{\mathcal{C}})$ so that $\mathrm{im}(\delta^{\mathcal{B}})$ has no nonzero elements supported entirely inside $L^{\mathcal{B},1}$, and therefore $\delta^{\mathcal{C}}(\mathcal{C}_{\bar{L}}^{i-1})$ has no nonzero elements supported entirely inside $A^{i-1} \times L^{\mathcal{B},1}$. Thus the correction in line 5 maps the above state to

$$\begin{aligned}
&\sum_{x_{\bar{L}} \in \mathcal{M}_{\bar{L}}^{\mathcal{C},i},\, b_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})} \left|\mathrm{Enc}_{\bar{L}}^{\mathcal{C}}(x_{\bar{L}}) + b_{\bar{L}}\right\rangle \left\langle \mathrm{Enc}_{\bar{L}}^{\mathcal{C}}(x_{\bar{L}}) + b_{\bar{L}}\right| \\
&\otimes \sum_{x_L, x'_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}},\, b_L, b'_L \in B^{i-1}(\mathcal{A}) \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}} \rho_{x,x'} \left|(\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x_L) + b_L\right\rangle \left\langle (\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x'_L) + b'_L\right| \\
&\otimes \nu_{x,x'}.
\end{aligned}$$

Tracing out over qubits $M_{\bar{L}}^{\mathcal{C},i}$ (i.e. the first register above) then gives

$$\begin{aligned}
&\sum_{x_{\bar{L}} \in \mathcal{M}_{\bar{L}}^{\mathcal{C},i}} (\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}} \left(\sum_{x_L, x'_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}} \rho_{x,x'} \left|x_L\right\rangle \left\langle x_L\right| \right) \otimes \nu_{x,x'} \\
&\quad = (\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(\mathrm{tr}_{M^{\mathcal{C},i} \setminus (M^{\mathcal{A},i-1} \times L^{\mathcal{B},1})}(\rho)),
\end{aligned}$$

as desired. □

5.3 Noisy Execution

In this section, we prove Proposition 5.1. Here we define all variables as in Section 5.1. We will perform a similar analysis as in Section 5.2, but now allowing for Pauli errors.

Proof of Proposition 5.1. The desired circuit $\mathcal{Q}$ is given in Algorithm 3. By definition line 2 and line 5 each use 1 timestep, while line 3 and line 4 just perform classical computations and therefore use 0 timesteps (in the sense of adaptive quantum circuits as defined in Definition 3.16). Thus the entire circuit uses $T = 2$ timesteps. Furthermore, the circuit by definition uses $N = |C^i| = n_{\text{in}}$ physical qubits.

Our goal is to show that for every $\ell \in \mathbb{N}$ and every $\rho \in \mathbb{C}^{(\mathcal{M}^{\mathcal{C}} \otimes \mathbb{F}_2^\ell) \times (\mathcal{M}^{\mathcal{C}} \otimes \mathbb{F}_2^\ell)}$, every Pauli $\mathcal{E}_{\text{run}}$-deviation[13] $\sigma = E_0 \operatorname{Enc}^{\mathcal{C},i}(\rho) E_0'$ of $\operatorname{Enc}^{\mathcal{C},i}(\rho)$ (for Paulis E_0, E_0' such that $\operatorname{supp}(E_0) \cup \operatorname{supp}(E_0') \subseteq C^i$ is $\mathcal{E}_{\text{run}}$-avoiding) and every $\mathcal{E}_{\text{run}}^{\sqcup T}$-avoiding Pauli fault $\mathcal{F}$, the resulting output $\mathcal{Q}[\mathcal{F}](\sigma)$ is a Pauli $\mathcal{E}_{\text{out}}$-deviation of $(\operatorname{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(\operatorname{tr}_{M^{\mathcal{C},i} \setminus (M^{\mathcal{A},i-1} \times L^{\mathcal{B},1})}(\rho))$.

We will perform a similar analysis as in the proof of Lemma 5.3, but while tracking all errors that arise at any point during the execution of $\mathcal{Q}$ using the graph G_{run}. Let

$$S_{\mathcal{F}} = (\operatorname{supp}(E_0) \cup \operatorname{supp}(E_0')) \cup \operatorname{supp}(\mathcal{F}_1) \cup \operatorname{supp}(\mathcal{F}_2) \subseteq C^i$$

denote the set of all physical qubits that lie in the support of the input's Pauli error E_0, E_0' or of the fault $\mathcal{F}$ at any timestep. Note that because $\mathcal{Q}$ has only single-qubit gates, these errors cannot propagate through gates to different qubits.

Claim 5.5. *The set $S_{\mathcal{F}}$ is $\mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ 3\gamma_{\text{run}})$-avoiding.*

Proof. The claim holds because by definition $S_{\mathcal{F}}$ is the union of three $\mathcal{E}_{\text{run}} = \mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ \gamma_{\text{run}})$-avoiding sets. □

To begin, we expand ρ as in (13). Writing $E_0 = Z^{e_Z} X^{e_X}$ and $E_0' = X^{e_X'} Z^{e_Z'}$, then the input state just prior to the measurements in line 2 is

$$E_0 \operatorname{Enc}^{\mathcal{C}}(\rho) E_0' \propto \sum_{x,x' \in \mathcal{M}^{\mathcal{C},i},\ b,b' \in B^i(\mathcal{C})} \rho_{x,x'} Z^{e_Z} X^{e_X} \left|\operatorname{Enc}^{\mathcal{C}}(x) + b\right\rangle \left\langle \operatorname{Enc}^{\mathcal{C}}(x) + b'\right| X^{e_X'} Z^{e_Z'} \otimes \nu_{x,x'}.$$

For $P \in \{X, Z\}$, write $e_P = (e_{P,\bar{L}}, e_{P,L})$ for $e_{P,\bar{L}} = e_P|_{C^i_{\bar{L}}}$ and $e_{P,L} = e_P|_{A^{i-1} \times L^{\mathcal{B},1}}$ (and analogously for the primed versions $e_P = (e'_{P,\bar{L}}, e'_{P,L})$). The measurements in line 2 project the state above to

$$\begin{aligned} &\sum_{z \in \mathcal{C}^i_{\bar{L}}} Z^{e_{Z,\bar{L}}} |z\rangle \langle z| Z^{e'_{Z,\bar{L}}} \\ &\quad \otimes \sum_{x,x',b,b'} \rho_{x,x'} Z^{e_{Z,L}} X^{e_{X,L}} \left|\operatorname{Enc}^{\mathcal{C}}(x) + b|_{A^{i-1} \times L^{\mathcal{B},1}}\right\rangle \left\langle \operatorname{Enc}^{\mathcal{C}}(x') + b'|_{A^{i-1} \times L^{\mathcal{B},1}}\right| X^{e'_{X,L}} Z^{e'_{Z,L}} \otimes \nu_{x,x'}, \end{aligned} \tag{17}$$

where the second sum above is over all $x, x' \in \mathcal{M}^{\mathcal{C},i}$ and $b, b' \in B^i(\mathcal{C})$ satisfying

$$\operatorname{Enc}^{\mathcal{C}}(x) + b|_{C^i_{\bar{L}}} + e_{X,\bar{L}} = z = \operatorname{Enc}^{\mathcal{C}}(x') + b'|_{C^i_{\bar{L}}} + e'_{X,\bar{L}}. \tag{18}$$

Let $S_{\text{syn}} \subseteq C^i \sqcup C^{i+1}$ be the footprint (see Definition 4.3) of the call to $\texttt{SSFlipSyn}(\delta^{\mathcal{C}_{\bar{L}}}(z); i, \mathcal{C}^*_{\bar{L}})$ in line 3 of Algorithm 3.

Claim 5.6. *The set $S_{\mathcal{F}} \cup S_{\text{syn}}$ is $\mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ 15w^3\gamma_{\text{run}})$-avoiding.*

[13] See Footnote 12.

Proof. Assume for a contradiction that there exists some $V \subseteq V_{\text{run}}$ of size $|V| \geq \eta_{\text{run}}$ that induces a connected subgraph of G_{run} with $|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}})|/|V| \geq 15w^3\gamma_{\text{run}}$. We may repeatedly add points in $(S_{\mathcal{F}} \cup S_{\text{syn}}) \setminus V$ that lie in the neighborhood of V, until no more such points exist. The resulting set V can only have larger size $|V| \geq \eta_{\text{run}}$ and $(S_{\mathcal{F}} \cup S_{\text{syn}})$-density $|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}})|/|V| \geq 15w^3\gamma_{\text{run}}$, and contains all or none of every connected component of the subgraph of G_{run} induced by $S_{\mathcal{F}}$.

By (18), the call $\texttt{SSFlipSyn}(\delta^{\mathcal{C}_{\bar{L}}}(z); i, \mathcal{C}^*_{\bar{L}})$ in line 3 of Algorithm 3 has $z = \text{Enc}^{\mathcal{C}}(x) + b|_{C^i_{\bar{L}}} + e_{X,\bar{L}}$, and by definition $\delta^{\mathcal{C}_{\bar{L}}}(\text{Enc}^{\mathcal{C}}(x) + b|_{C^i_{\bar{L}}}) = 0$, so $\delta^{\mathcal{C}_{\bar{L}}}(z) = \delta^{\mathcal{C}_{\bar{L}}}(e_{X,\bar{L}})$. Then by Lemma 4.5, we have

$$\begin{aligned} |V \cap S_{\text{syn}}| &\leq |V \cap \text{supp}(e_{X,\bar{L}})|/\gamma_{\text{syn}} \\ &\leq |V \cap S_{\mathcal{F}}| \cdot 4w^3 \\ &< 3\gamma_{\text{run}}|V| \cdot 4w^3, \end{aligned}$$

where the third inequality above holds by Claim 5.5. We then again apply Claim 5.5 to conclude that

$$|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}})| \leq |V \cap S_{\mathcal{F}}| + |V \cap S_{\text{syn}}| < (4w^3 + 1) \cdot 3\gamma_{\text{run}}|V|,$$

which contradicts the assumption that $|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}})| \geq 15w^3\gamma_{\text{run}}$, as desired. □

Recall that by assumption $\eta_{\text{run}} \leq m$. Also by definition, $\gamma_{\text{run}} \leq 1/15w^3$, so it follows by Claim 5.6 that the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup S_{\text{syn}}$ has no connected components containing $\geq \eta_{\text{run}}$ vertices. Therefore within every connected component V of this induced subgraph, by the assumption that $\mathcal{C}^*_{\bar{L}}$ has a $(m, 0)$-small-set syndrome-flip decoder at level i, the while loop in the restricted execution $\texttt{SSFlipSyn}(\delta^{\mathcal{C}_{\bar{L}}}(z|_{V \cap C^i_{\bar{L}}}); i, \mathcal{C}^*_{\bar{L}})$ will not terminate until the computed output $a^i|_{V \cap C^i_{\bar{L}}}$ satisfies $\delta^{\mathcal{C}_{\bar{L}}}(z + a^i|_{V \cap C^i_{\bar{L}}}) = 0$. By definition $\text{supp}(e_{X,\bar{L}}) \subseteq S_{\mathcal{F}}$, and by (18), it follows that $\delta^{\mathcal{C}_{\bar{L}}}(e_{X,\bar{L}} + a^i|_{V \cap C^i_{\bar{L}}}) = \delta^{\mathcal{C}_{\bar{L}}}(z + a^i|_{V \cap C^i_{\bar{L}}}) = 0$. We analogously have $\text{supp}(e'_{X,\bar{L}}) \subseteq S_{\mathcal{F}}$ and $\delta^{\mathcal{C}_{\bar{L}}}(e'_{X,\bar{L}} + a^i|_{V \cap C^i_{\bar{L}}}) = 0$.

We now define S_{err} and $\tilde{a}^{i-1}, \tilde{a}^i$ (resp. S'_{err} and $(\tilde{a}^{i-1})', (\tilde{a}^i)'$) to be the footprint and output of $\texttt{SSFlipErr}(e_{X,\bar{L}} + a^i; i, \mathcal{C}^*_{\bar{L}})$ (resp. $\texttt{SSFlipErr}(e'_{X,\bar{L}} + a^i; i, \mathcal{C}^*_{\bar{L}})$), respectively. We prove the following claim for the unprimed variables $e_{X,\bar{L}}, \ldots$, but it analogously applies to the primed variables $e'_{X,\bar{L}}, \ldots$.

Claim 5.7. *The set $S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ is $\mathcal{E}(G_{\text{run}}, \eta_{\text{run}}, 150w^7\gamma_{\text{run}})$-avoiding.*

Proof. Assume for a contradiction that there exists some $V \subseteq V_{\text{run}}$ of size $|V| \geq \eta_{\text{run}}$ that induces a connected subgraph of G_{run} with $|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \geq 150w^7\gamma_{\text{run}}$. Similarly as in Claim 5.6, we may repeatedly add points in $(S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}) \setminus V$ that lie in the neighborhood of V, in order to assume that V contains all or none of every connected component of the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$.

By Lemma 4.5, we have

$$\begin{aligned} |V \cap S_{\text{err}}| &\leq |V \cap \text{supp}(e_{X,\bar{L}} + a^i)|/\gamma_{\text{err}} \\ &\leq |V \cap (S_{\mathcal{F}} \cup S_{\text{syn}})| \cdot 8w^4 \\ &< 15w^3\gamma_{\text{run}}|V| \cdot 8w^4, \end{aligned}$$

where the third inequality above holds by Claim 5.6. We then again apply Claim 5.6 to conclude that

$$|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \leq |V \cap (S_{\mathcal{F}} \cup S_{\text{syn}})| + |V \cap S_{\text{err}}| < (8w^4 + 1)15w^3\gamma_{\text{run}}|V|,$$

which contradicts the assumption that $|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \geq 150w^7\gamma_{\text{run}}$, as desired. □

Recall that we use $\tilde{a}^{i-1}, \tilde{a}^i$ to denote the variables that are updated throughout the execution of and then returned by $\texttt{SSFlipErr}(e_{X,\bar{L}} + a^i; i, \mathcal{C}^*_{\bar{L}})$. As shown above, we have $\delta^{\mathcal{C}_{\bar{L}}}(e_{X,\bar{L}} + a^i)$, and hence the entire execution of $\texttt{SSFlipErr}(e_{X,\bar{L}} + a^i; i, \mathcal{C}^*_{\bar{L}})$ will have $\tilde{a}^i = 0$; that is, every iteration of the while loop (in line 8 of Algorithm 2) that does not fail must add some c^{i-1} in to $\tilde{a}^{i-1}$. By definition, every such c^{i-1} must have $\text{supp}(c^{i-1})$ contained within the neighborhood in $G^{\mathcal{C}}_{i-1,i,i+1} \subseteq G_{\text{run}}$ of the footprint from the previous step. As a consequence, if we restrict $e_{X,\bar{L}} + a^i$ to a connected component V if $S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ (and zero out all values outside of V), the output of $\texttt{SSFlipErr}(e_{X,\bar{L}} + a^i|_{V \cap C^i_{\bar{L}}}; i, \mathcal{C}^*_{\bar{L}})$ must equal the restriction $\tilde{a}^{i-1}|_{V \cap C^{i-1}_{\bar{L}}}, \tilde{a}^i|_{V \cap C^i_{\bar{L}}}$ of the output $\tilde{a}^{i-1}, \tilde{a}^i$ of $\texttt{SSFlipErr}(e_{X,\bar{L}} + a^i; i, \mathcal{C}^*_{\bar{L}})$ (see Lemma 4.4).

Recall that by assumption $\eta_{\text{run}} \leq m$. Also by definition, $\gamma_{\text{run}} \leq 1/150w^7$, so it follows by Claim 5.7 that the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ has no connected components containing $\geq \eta_{\text{run}}$ vertices. Therefore within every connected component V of this induced subgraph, by the assumption that $\mathcal{C}^*_{\bar{L}}$ has a m-small-set error-flip decoder at level i, the while loop in the restricted execution $\texttt{SSFlipErr}(e_{X,\bar{L}} + a^i|_{V \cap C^i_{\bar{L}}}; i, \mathcal{C}^*_{\bar{L}})$ will not terminate until the computed output $\tilde{a}^{i-1}|_{V \cap C^{i-1}_{\bar{L}}}$ satisfies $\delta^{\mathcal{C}_{\bar{L}}}(\tilde{a}^{i-1}|_{V \cap C^{i-1}_{\bar{L}}}) = e_{X,\bar{L}} + a^i|_{V \cap C^i_{\bar{L}}}$; recall from above that we will have $\tilde{a}^i|_{V \cap C^i_{\bar{L}}} = 0$. It also follows that the output will always be such a valid $\tilde{a}^{i-1}|_{V \cap C^{i-1}_{\bar{L}}}$, and never FAIL. Therefore we conclude that

$$\delta^{\mathcal{C}}(\tilde{a}^{i-1}) = e_{X,\bar{L}} + a^i, \tag{19}$$

and by definition $\text{supp}(\tilde{a}^{i-1}) \subseteq S_{\text{err}}$.

As in Lemma 5.3, for every $b_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})$, fix an arbitrary $c_{\bar{L}}(b_{\bar{L}}) \in \mathcal{C}^{i-1}_{\bar{L}}$ satisfying $\delta^{\mathcal{C}_{\bar{L}}}(c_{\bar{L}}(b_{\bar{L}})) = b_{\bar{L}}$. Then it follows by (18) that for every measurement result z, then $c_{\bar{L}}(b|_{C^i_{\bar{L}}}) + \tilde{a}^{i-1}$ is a valid choice for $c_{\bar{L}}$ in line 4 of Algorithm 3. Thus because no nonzero element of $\text{im}(\delta^{\mathcal{C}}_{\bar{L}})$ lies in $\text{im}(\text{Enc}^{\mathcal{C},i-1}_{\bar{L}})$ (as every nonzero element of the latter lies in a nontrivial cohomology class; see the proof of Lemma 5.3), there is a unique valid value of $\delta^{\mathcal{C}_{\bar{L}}}(c_{\bar{L}})$, so we must have

$$c_{\bar{L}} = c_{\bar{L}}(b|_{C^i_{\bar{L}}}) + \tilde{a}^{i-1} + z^{i-1}$$

for some cocycle $z^{i-1} \in Z^{i-1}(\mathcal{C}_{\bar{L}})$, and hence

$$\delta^{\mathcal{C}}(c_{\bar{L}}) = \delta^{\mathcal{C}}(c_{\bar{L}}(b|_{C^i_{\bar{L}}}) + \tilde{a}^{i-1}) \tag{20}$$

because $\delta^{\mathcal{C}}(z^{i-1}) = 0$ as every nonzero element of $\delta^{\mathcal{C}}(\mathcal{C}^{i-1}_{\bar{L}})$ has nontrivial support inside $C^i_{\bar{L}}$ (see the proof of Lemma 5.3).

Recall that the entire analysis above also applies to the primed variables $e'_{X,\bar{L}}, \ldots$, so we also have

$$\delta^{\mathcal{C}_{\bar{L}}}((\tilde{a}^{i-1})') = e'_{X,\bar{L}} + a^i, \tag{21}$$

with $\text{supp}(\tilde{a}^{i-1}) \subseteq S'_{\text{err}}$ and

$$\delta^{\mathcal{C}}(c_{\bar{L}}) = \delta^{\mathcal{C}}(c_{\bar{L}}(b'|_{C^i_{\bar{L}}}) + (\tilde{a}^{i-1})'). \tag{22}$$

By (18) and Claim 5.4, for every $x_{\bar{L}} \in \mathcal{M}^{\mathcal{C},i}_{\bar{L}}$, $b_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})$, there exists a unique choice of $x'_{\bar{L}} \in \mathcal{M}^{\mathcal{C},i}_{\bar{L}}$, $b'_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})$ such that

$$\text{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}}) + b_{\bar{L}} + e_{X,\bar{L}} = \text{Enc}^{\mathcal{C}}_{\bar{L}}(x'_{\bar{L}}) + b'_{\bar{L}} + e'_{X,\bar{L}}.$$

By (19) and (21), we have $\delta^{\mathcal{C}_{\bar{L}}}(\tilde{a}^{i-1}) = e_{X,\bar{L}} + e'_{X,\bar{L}}$, so $\mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}} + x'_{\bar{L}}) \in B^i(\mathcal{C}_{\bar{L}})$, and thus we must have $x_{\bar{L}} = x'_{\bar{L}}$ because $\mathrm{Enc}^{\mathcal{C}}_{\bar{L}}$ maps every nonzero input to a nontrivial cohomology class. Thus

$$b_{\bar{L}} + e_{X,\bar{L}} = b'_{\bar{L}} + e'_{X,\bar{L}}. \tag{23}$$

Therefore by analogous reasoning as used to derive (16) in the proof of Lemma 5.3, the state (17) can be expressed as

$$\begin{aligned}
&\sum_{x_{\bar{L}} \in \mathcal{M}^{\mathcal{C},i}_{\bar{L}},\, b_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})} Z^{e_{Z,\bar{L}}} \left|\mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}}) + b_{\bar{L}} + e_{X,\bar{L}}\right\rangle \left\langle \mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}}) + b_{\bar{L}} + e_{X,\bar{L}}\right| Z^{e_{Z',\bar{L}}} \\
&\quad \otimes \sum_{x_L, x'_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}},\, b_L, b'_L \in B^{i-1}(\mathcal{A}) \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}} \\
&\qquad \rho_{x,x'} Z^{e_{Z,L}} X^{e_{X,L}} \left|(\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x_L) + b_L + (\delta^{\mathcal{C}}(c_{\bar{L}}(b_{\bar{L}}))|_{A^{i-1} \times L^{\mathcal{B},1}})\right\rangle \\
&\qquad\qquad \left\langle (\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x'_L) + b'_L + (\delta^{\mathcal{C}}(c_{\bar{L}}(b'_{\bar{L}}))|_{A^{i-1} \times L^{\mathcal{B},1}})\right| X^{e'_{X,L}} Z^{e'_{Z,L}} \\
&\quad \otimes \nu_{x,x'},
\end{aligned} \tag{24}$$

where above $b'_{\bar{L}}$ is defined by (23), that is $b'_{\bar{L}} = b_{\bar{L}} + e_{X,\bar{L}} + e'_{X,\bar{L}}$.

Then running line 6 of Algorithm 3 on the state in (24) simply applies the correction $X^{\delta^{\mathcal{C}}(c_{\bar{L}})|_{A^{i-1} \times L^{\mathcal{B},1}}}$ and incurs the fault $\mathcal{F}_2$ at timestep $t = 2$, so it follows by (20) and (22), the final state is

$$\begin{aligned}
&\sum_{x_{\bar{L}} \in \mathcal{M}^{\mathcal{C},i}_{\bar{L}},\, b_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})} Z^{f_{Z,\bar{L}}} X^{f_{X,\bar{L}}} \left|\mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}}) + b_{\bar{L}}\right\rangle \left\langle \mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}}) + b_{\bar{L}}\right| X^{e_{X,\bar{L}} + e'_{X,\bar{L}} + f'_{X,\bar{L}}} Z^{f_{Z,\bar{L}}} \\
&\quad \otimes \sum_{x_L, x'_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}},\, b_L, b'_L \in B^{i-1}(\mathcal{A}) \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}} \\
&\qquad \rho_{x,x'} Z^{f_{Z,L}} X^{f_{X,L}} \left|(\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x_L) + b_L + (\delta^{\mathcal{C}}(\tilde{a}^{i-1})|_{A^{i-1} \times L^{\mathcal{B},1}})\right\rangle \\
&\qquad\qquad \left\langle (\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x'_L) + b'_L + (\delta^{\mathcal{C}}(\tilde{a}^{i-1})'|_{A^{i-1} \times L^{\mathcal{B},1}})\right| X^{f'_{X,L}} Z^{f'_{Z,L}} \\
&\quad \otimes \nu_{x,x'},
\end{aligned} \tag{25}$$

where all f_α and f'_α are obtained by adding the errors from $\mathcal{F}_2$ in to e_α and e'_α, respectively. Therefore by definition $f_X = (f_{X,\bar{L}}, f_{X,L})$ and $f_Z = (f_{Z,\bar{L}}, f_{Z,L})$ are supported inside $S_{\mathcal{F}}$.

Algorithm 3 returns the trace over the first register (i.e. qubits in $C^i_{\bar{L}}$) of the state in (25). If $e_{X,\bar{L}} + e'_{X,\bar{L}} + f_{X,\bar{L}} + f'_{X,\bar{L}} \neq 0$, then this state is 0, so assume that $e_{X,\bar{L}} + e'_{X,\bar{L}} + f_{X,\bar{L}} + f'_{X,\bar{L}} = 0$.

If $f_{Z,\bar{L}} + f'_{Z,\bar{L}} \notin Z_i(\mathcal{C}_{\bar{L}}) = B^i(\mathcal{C}_{\bar{L}})^\perp$, then for every $x_{\bar{L}} \in \mathcal{M}^{\mathcal{C},i}_{\bar{L}}$, half the terms in the sum over $b_{\bar{L}} \in B^i(\mathcal{C}_{\bar{L}})$ receive a $+1$ phase from the Paulis $Z^{f_{Z,\bar{L}}}, Z^{f'_{Z,\bar{L}}}$ and half receive a -1 phase, so the overall state vanishes. Thus assume that $f_{Z,\bar{L}} + f'_{Z,\bar{L}} \in Z_i(\mathcal{C}_{\bar{L}})$. Recall that $\mathrm{supp}(f_{Z,\bar{L}} + f'_{Z,\bar{L}}) \subseteq S_{\mathcal{F}}$. Because $\gamma_{\mathrm{run}} \leq 1/3$, Claim 5.5 implies that the subgraph of G_{run} induced by $S_{\mathcal{F}}$ has no connected components with $\geq \eta_{\mathrm{run}}$ vertices. Hence every connected component of the subgraph of $G_i^{\mathcal{C}_{\bar{L}}} \subseteq G_{\mathrm{run}}$ induced by $\mathrm{supp}(f_{Z,\bar{L}} + f'_{Z,\bar{L}})$ has $< \eta_{\mathrm{run}}$ vertices, and by assumption $\eta_{\mathrm{run}} \leq d_i(\mathcal{C}_{\bar{L}})$. The restriction of $f_{Z,\bar{L}} + f'_{Z,\bar{L}}$ to every such connected component must lie in $Z_i(\mathcal{C}_{\bar{L}})$, as if any two such connected components shared a parity-check in $C_{\bar{L},i-1}$, they would necessarily be connected

in $G_i^{\mathcal{C}_{\bar{L}}}$, a contradiction. But then by the definition of the i-systolic distance $d_i(\mathcal{C}_{\bar{L}})$, it follows that the restriction of $f_{Z,\bar{L}} + f'_{Z,\bar{L}}$ is a boundary in $B_i(\mathcal{C}_{\bar{L}})$, and thence $f_{Z,\bar{L}} + f'_{Z,\bar{L}} \in B_i(\mathcal{C}_{\bar{L}})$. But then because every $\mathrm{Enc}^{\mathcal{C}}_{\bar{L}}(x_{\bar{L}}) + b_{\bar{L}} \in Z^i(\mathcal{C}_{\bar{L}}) = B_i(\mathcal{C}_{\bar{L}})^{\perp}$, every term in the sum over $x_{\bar{L}}, b_{\bar{L}}$ receives the same phase $(-1)^{(f_{Z,\bar{L}}+f_{Z,\bar{L}})\cdot(e_{X,\bar{L}}+e'_{X,\bar{L}}+f_{X,\bar{L}}+f'_{X,\bar{L}})}$. Hence the final output returned by Algorithm 3 is proportional to

$$\sum_{x_{\bar{L}} \in \mathcal{M}^{\mathcal{C},i}_{\bar{L}}} \sum_{x_L, x'_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}},\, b_L, b'_L \in B^{i-1}(\mathcal{A}) \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}} \rho_{x,x'} Z^{f_{Z,L}} X^{f_{X,L} + (\delta^{\mathcal{C}}(\tilde{a}^{i-1})|_{A^{i-1} \times L^{\mathcal{B},1}})} \left| (\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x_L) + b_L \right\rangle \left\langle (\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}}(x'_L) + b'_L \right| X^{f'_{X,L} + (\delta^{\mathcal{C}}(\tilde{a}^{i-1})'|_{A^{i-1} \times L^{\mathcal{B},1}})} Z^{f'_{Z,L}} \otimes \nu_{x,x'},$$

which differs from the state

$$\sum_{x_{\bar{L}} \in \mathcal{M}^{\mathcal{C},i}_{\bar{L}}} ((\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}} \otimes \mathcal{I}_{\mathcal{L}}) \left(\sum_{x_L, x'_L \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}} \rho_{x,x'} |x_L\rangle \langle x'_L| \otimes \nu_{x,x'} \right) = ((\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}} \otimes \mathcal{I}_{\mathcal{L}})(\mathrm{tr}_{M^{\mathcal{C},i} \setminus (M^{\mathcal{A},i-1} \times L^{\mathcal{B},1})}(\rho))$$

by a Pauli error supported inside

$$S_{\mathcal{F}} \cup S_{\mathrm{run}} \cup S_{\mathrm{err}} \cup S'_{\mathrm{err}} = (S_{\mathcal{F}} \cup S_{\mathrm{run}} \cup S_{\mathrm{err}}) \cup S_{\mathcal{F}} \cup S_{\mathrm{run}} \cup S_{\mathrm{err}'}.$$

The set above is $\mathcal{C}_{\mathrm{out}} = \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{run}},\ 300w^7\gamma_{\mathrm{run}})$-avoiding by Claim 5.7, so we have shown that the output of Algorithm 3 is a Pauli $\mathcal{E}_{\mathrm{out}}$-deviation of $((\mathrm{Enc}^{\mathcal{A},i-1})^{L^{\mathcal{B},1}} \otimes \mathcal{I}_{\mathcal{L}})(\mathrm{tr}_{M^{\mathcal{C},i} \setminus (M^{\mathcal{A},i-1} \times L^{\mathcal{B},1})}(\rho))$, as desired. □

6 Upwards Code Switching Gadget via Teleportation

This section presents our gadget for switching from an $(r-1)$-dimensional code to an r-dimensional code. Specifically, we prepare logical bell pairs between the two codes, which we use to teleport a state from the lower-dimensional code to the higher-dimensional code.

We will use our downwards code switching gadget from Proposition 5.1 to prepare these logical bell pairs. In Remark C.1 in Appendix C, we explain why we need to use this gadget when $r = 3$, and how we can avoid using it when $r \geq 4$.

We describe notation and state our upwards-switching result below. We prove the result in Appendix C, as it is a fairly direct application of other gadgets we develop in this paper, some of which we only present in the sections below. Specifically, the logical bell pair preparation will require the state preparation in Proposition 7.1, the downwards code switching in Proposition 5.1, and the transvesal $CNOT$ in Lemma 9.2; the teleportation will then require the transersal $CNOT$ in Lemma 9.1 and the logical Pauli measurements in Lemma 9.5. The application of these gadgets is perhaps unsurprising, given that a teleportation circuit can be performed using qubit state preparation, $CNOT$ gates, Pauli measurements, and unitary Pauli gates (see Figure 2).

Fix $r \in \mathbb{N}$, and define $\mathcal{A}^*$, $\mathcal{M}^{\mathcal{A},*}$, $\mathrm{Enc}^{\mathcal{A}}$, $\mathcal{B}^*$, $\mathcal{M}^{\mathcal{B},*}$, $L^{\mathcal{B},1}$, $\mathrm{Enc}^{\mathcal{B}}$, $\mathcal{B}^*_{\bar{L}}$, $\mathcal{M}^{\mathcal{B},*}$, $\mathrm{Enc}^{\mathcal{B}}_{\bar{L}}$, $\mathcal{C}^*$, $\mathcal{M}^{\mathcal{C},*}$, $\mathrm{Enc}^{\mathcal{C}}$ as in Section 5.1.

Fix some $2 \le i \le r-1$. Assume that the chain complex[14] $\mathcal{C}_* = \mathcal{A}_* \otimes \mathcal{B}_*$ has a $(m,0)$-small-set flip decoder at level $i-1$. Also assume that the cochain complex[15] $\mathcal{D}^* := \mathcal{A}^* \otimes \mathcal{B}_*$ has a $(m,0)$-small-set flip decoder at level i, and that the chain complex $\mathcal{D}_{\bar{L},*} := \mathcal{A}_* \otimes \mathcal{B}^*_{\bar{L}}$ has a $(m,0)$-small-set flip decoder at level $i-1$.

We also let $\mathrm{Enc}^{\mathcal{D}}$, $\mathrm{Enc}^{\mathcal{D}}_{\bar{L}}$ be encoding maps for $\mathcal{D}$, $\mathcal{D}_{\bar{L}}$ respectively that are defined analogously to $\mathrm{Enc}^{\mathcal{C}}$, $\mathrm{Enc}^{\mathcal{C}}_{\bar{L}}$ in Section 5.1 (with $\mathcal{D}$ replacing $\mathcal{C}$).

Let $G_{\mathrm{run}} = (V_{\mathrm{run}}, E_{\mathrm{run}})$ be some graph that contains[16] $G^{\mathcal{C}}_{[r]} \cong G^{\mathcal{D}}_{[r]} \subseteq G_{\mathrm{run}}$ as a subgraph. There exists some sufficiently small $\gamma_{\mathrm{in}}(w) > 0$ and some sufficiently large $\zeta_{\mathrm{out}}(w) > 0$ such that for every

$$\begin{aligned} \eta_{\mathrm{in}} &\le \min\{m,\ d_{i-1}(\mathcal{A}),\ d^{i-1}(\mathcal{A}),\ d^{i}(\mathcal{D}_{\bar{L}})\} \\ \gamma_{\mathrm{in}} &\le \gamma_{\mathrm{in}}(w) \\ \gamma_{\mathrm{out}} &\ge \zeta_{\mathrm{out}}(w) \cdot \gamma_{\mathrm{in}}, \end{aligned}$$

then Proposition 6.1 below holds.

Let Q_{in} be the $[[n_{\mathrm{in}}, k_{\mathrm{in}}]]$ CSS code consisting of $|L^{\mathcal{B},1}|$ copies of the CSS code associated to level $i-1$ of $\mathcal{A}^*$, and let Q_{out} be the $[[n_{\mathrm{out}}, k_{\mathrm{out}}]]$ CSS code associated to level i of $\mathcal{C}^*$. Let

$$\begin{aligned} \mathcal{E}_{\mathrm{in}} &= \mathcal{E}(G^{\mathcal{A}}_{[r-1]},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{in}}) \\ \mathcal{E}_{\mathrm{run}} &= \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{in}}) \\ \mathcal{E}_{\mathrm{out}} &= \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{out}}), \end{aligned}$$

and define the decorated codes

$$\begin{aligned} D_{\mathrm{in}} &= (Q_{\mathrm{in}},\ \mathrm{Enc}_{\mathrm{in}} = (\mathrm{Enc}^{\mathcal{A}})^{\sqcup L^{\mathcal{B},1}},\ \mathcal{E}_{\mathrm{in}}^{\sqcup L^{\mathcal{B},1}}) \\ D_{\mathrm{out}} &= (Q_{\mathrm{out}},\ \mathrm{Enc}_{\mathrm{out}} = \mathrm{Enc}^{\mathcal{C}},\ \mathcal{E}_{\mathrm{out}}). \end{aligned}$$

In these decorated codes, the n_{in} physical qubits of Q_{in} are naturally associated with the set $A^{i-1} \times L^{\mathcal{B},1}$, while the n_{out} physical qubits of Q_{out} are naturally associated with the set C^i; both sets are naturally subsets of $C^i \subseteq V_{\mathrm{run}}$.

Similarly as described in Section 5.1 (but with the input and output codes reversed), the logical qubits $M^{\mathcal{A},i-1} \times L^{\mathcal{B},1}$ are naturally a subset of the logical qubits $M^{\mathcal{C},i}$ of Q_{out}. We therefore let $\bar{\mathcal{O}} : \mathbb{C}^{2^{k_{\mathrm{in}}} \times 2^{k_{\mathrm{in}}}} \to \mathbb{C}^{2^{k_{\mathrm{out}}} \times 2^{k_{\mathrm{out}}}}$ be the channel that acts as the identity on all the input qubits, and pads the input with an additional $k_{\mathrm{out}} - k_{\mathrm{in}}$ qubits in the $|0\rangle$ state. That is, $\bar{\mathcal{O}}(\rho) = \rho \otimes (|0\rangle\langle 0|)^{\otimes k_{\mathrm{out}} - k_{\mathrm{in}}}$.

Proposition 6.1. *Define all variables as above in Section 6. Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, (\mathcal{E}_{\mathrm{run}}^{\sqcup u})^{\sqcup T}), D_{\mathrm{in}}, D_{\mathrm{out}})$ for $\bar{\mathcal{O}}$, where $\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = O(n_{\mathrm{out}} w)$ and time $T = O(nw)$, and $u = O(1)$.*

Remark 6.2. *As described above (and more formally in Appendix C), the gadget $\mathcal{Q}$ is entirely composed of (a constant number of) gadgets previously presented in the paper. Therefore the physical qubits are naturally partitioned into a constant number of blocks, each of which is naturally a subset of $C \cong D \subseteq V_{\mathrm{run}}$; the constant u is the number of such blocks used.*

[14] Recall the label inversion for chain complexes vs cochain complexes; if $\mathcal{C}_*$ here were viewed as a cochain complex, it would have a $(m,0)$-small-set flip decoder at level $r-(i-1)$.

[15] A further clarification following Footnote 14: here $\mathcal{D}^* = \mathcal{A}^* \otimes \mathcal{B}_*$ is not equal to $\mathcal{C}^* = \mathcal{A}^* \otimes \mathcal{B}^*$, as we have dualized $\mathcal{B}$.

[16] Note that by definition there is an isomorphism between the basis sets $C \cong D$ given by an appropriate permutation, which induces an isomorphism between the graphs $G^{\mathcal{C}}_{[r]} \cong G^{\mathcal{D}}_{[r]}$.

7 State Preparation Gadget

In this section, we present our gadget for preparing logical $|0\rangle$ and $|+\rangle$ states of quantum codes given by ≥ 3-dimensional tensor products of classical lossless-expander codes. Our gadget crucially relies on the small-set flip decoder in Proposition 4.6.

Algorithm 4: State preparation gadget in Proposition 7.1.

Input : None
Output: $\mathrm{Enc}(\left|+^k\right\rangle\left\langle+^k\right|)$

1 **Function** `StatePrep`$(i, \mathcal{C}^*)$**:**
2 Initialize an $n = |C^i|$-qubit code register $\rho \leftarrow \left|+^{C^i}\right\rangle\left\langle+^{C^i}\right|$, and an additional $|C^{i+1}|$-qubit ancilla register $\alpha \leftarrow \left|0^{C^{i+1}}\right\rangle\left\langle 0^{C^{i+1}}\right|$
3 Run `SyndExt`$(\rho; Z, \delta_i)$ from Algorithm 1, and let $s \in \mathcal{C}^{i+1}$ be the resulting syndrome
4 Run `SSFlipSyn`$(\delta(s); i+1, \mathcal{C}^*)$ from Algorithm 2, and let $a^{i+1} \in \mathcal{C}^{i+1}$ be the output
5 Run Gaussian elimination to find some $x \in \mathcal{C}^i$ with $\delta(x) = s + a^{i+1}$
6 Apply the Pauli X^x to ρ
7 **return** ρ

Proposition 7.1. *For $r \in \mathbb{N}$ and $1 \leq i \leq r-2$, let $\mathcal{C}^*$ be an r-dimensional cochain complex of locality w with a $(m, 0)$-small-set flip decoder at level $i+1$ (see Definition 4.1). Let Q be the $[[n, k]]$ CSS code associated to level i of $\mathcal{C}^*$, with an associated CSS encoding map* Enc *(see Definition 3.32). Given some graph $G_{\mathrm{run}} = (V_{\mathrm{run}}, E_{\mathrm{run}})$ that contains $G^{\mathcal{C}}_{i,i+1,i+2} \subseteq G_{\mathrm{run}}$ as a subgraph and some*

$$\eta_{\mathrm{run}} \leq m$$
$$\gamma_{\mathrm{run}} \leq \frac{1}{50w^7(w+10)2^{w+10}},$$

let

$$\mathcal{E}_{\mathrm{run}} = \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{run}},\ \gamma_{\mathrm{run}})$$
$$\mathcal{E}_{\mathrm{out}} = \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{run}},\ 100w^7(w+10)2^{w+10}\gamma_{\mathrm{run}}),$$

and define the decorated code (see Definition 3.23) $D_{\mathrm{out}} = (Q,\ \mathrm{Enc},\ \mathcal{E}_{\mathrm{out}})$. Let $\bar{\mathcal{O}} : \mathbb{C} \to \mathbb{C}^{2^k \times 2^k}$ be the channel given by $\bar{\mathcal{O}}(1) = \left|+^k\right\rangle\left\langle+^k\right|$.

Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, \mathcal{E}^{\sqcup T}_{\mathrm{run}}), \emptyset, D_{\mathrm{out}})$ for $\bar{\mathcal{O}}$, where $\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = O(nw)$ and time $T = O(w)$.

The gadget $\mathcal{Q}$ in Proposition 7.1. is given in Algorithm 4. It is well-known that the noisless execution prepares the desired logical $\left|+^k\right\rangle$ state; we include a brief proof below for completeness.

Lemma 7.2 (Well known)**.** *In the absence of errors,* `StatePrep`$(i, \mathcal{C}^*)$ *in Algorithm 4 returns* $\mathrm{Enc}(\left|+^k\right\rangle\left\langle+^k\right|)$.

Proof. By Corollary 3.36, following the syndrome-extraction measurements in line 3, the code register must equal $X^b\,\mathrm{Enc}(\left|+^k\right\rangle\left\langle+^k\right|)X^b$ for some $b \in \mathcal{C}^i$ such that the measured syndrome is $s = \delta(b)$. Specifically, the overall state is given by the mixture

$$\sum_{b \in \mathcal{C}^i} X^b\,\mathrm{Enc}\left(\left|+^k\right\rangle\left\langle+^k\right|\right)X^b \otimes |\delta(b)\rangle\langle\delta(b)|\,.$$

This statement holds because the X and Z stabilizers of a CSS code always commute, and the initial state $|+^n\rangle$ lies in the $+1$ eigenstate of all X stabilizers and logical operators, so the post-measurement state also lies in the $+1$ eigenstate of all X stabilizers and logical operators, and hence must equal $\mathrm{Enc}(|+^k\rangle\langle+^k|)$ with some (superposition of) X errors applied. The Z measurements we performed must collapse this superposition down to a single Pauli error X^b, and by definition the syndrome is $s = \delta(b)$.

Because $s = \delta(b)$, we have $\delta(s) = 0$, so line 4 has a trivial decoding problem that gives $a^{i+1} = 0$. Then line 5 computes some $x \in \mathcal{C}^i$ with $\delta(x) = s$, and line 6 applies X^x to compute the final output state $\rho = X^{b+x}\,\mathrm{Enc}(|+^k\rangle\langle+^k|)X^{b+x}$. By definition $\delta(b+x) = s + s = 0$, so X^{b+x} is a logical X-operator and hence preserves $\mathrm{Enc}\,|+^k\rangle$. Equivalently, $\mathrm{Enc}\,|+^k\rangle$ is by definition a uniform superposition over all i-cocycles in $Z^i(\mathcal{C})$, and this uniform superposition is preserved under adding in any given i-cocycle (such as $b+x$). Thus the output is $\rho = \mathrm{Enc}(|+^k\rangle\langle+^k|)$, as desired. □

Remark 7.3. *While Proposition 7.1 is stated for the preparation of logical $|+\rangle$ states, the same result holds for preparation of logical $|0\rangle$ states, by simply exchanging the roles of the Pauli X and Z bases everywhere in the construction and analysis.*

To prove Proposition 7.1, we analyze the noisy execution of Algorithm 4. The proof is largely similar to the proof of Proposition 5.1 given in Section 5.3 above, so we postpone the proof to Appendix A.

8 Error Correction Gadget

This section presents our gadget for performing error correction on codes with small-set syndrome-flip decoders.

Algorithm 5: Error correction gadget in Proposition 8.1; all variables are defined as in the proposition. In particular, the output avoids a larger family $\mathcal{E}_{\mathrm{out}} \supseteq \mathcal{E}_{\mathrm{in}}$ of bad sets than the input, and hence is less corrupted. We call the $n = |C^i|$-qubit input the *code register*, and the $|C^{i+1}|$ and $|C^{i-1}|$ qubit ancilla systems used by the syndrome extraction subroutines in line 2 and line 5 respectively the Z and X *syndrome registers*.

Input : $\sigma \in \mathbb{C}^{\mathcal{C}^i \times \mathcal{C}^i}$ that is a $\mathcal{E}_{\mathrm{in}}$-deviation of $\mathrm{Enc}(\rho)$ for $\rho \in \mathbb{C}^{2^k \times 2^k}$
Output: $\sigma' \in \mathbb{C}^{\mathcal{C}^i \times \mathcal{C}^i}$ that is a $\mathcal{E}_{\mathrm{out}}$-deviation of $\mathrm{Enc}(\rho)$

1 **Function** `ErrCorr`$(\sigma; i, \mathcal{C}^*)$**:**
2 Run `SyndExt`$(\sigma; Z, \delta_i^{\mathcal{C}})$ from Algorithm 1, and let $s_Z \in \mathcal{C}^{i+1}$ be the resulting syndrome
3 Run `SSFlipSyn`$(s_Z; i, \mathcal{C}^*)$ from Algorithm 2, and let $a_Z^i \in \mathcal{C}^i$ be the output
4 Apply the Pauli $X^{a_Z^i}$ to σ
5 Run `SyndExt`$(\sigma; X, \partial_i^{\mathcal{C}})$ from Algorithm 1, and let $s_X \in \mathcal{C}_{i-1}$ be the resulting syndrome
6 Run `SSFlipSyn`$(s_X; i, \mathcal{C}_*)$ from Algorithm 2, and let $a_{X,i} \in \mathcal{C}_i$ be the output
7 Apply the Pauli $X^{a_{X,i}}$ to σ
8 **return** σ

Proposition 8.1. *For $r \in \mathbb{N}$ and $1 \le i \le r-1$, let $\mathcal{C}^*$ be an r-dimensional cochain complex of locality w with a $(m, \gamma_{\mathrm{dec}})$-small-set flip decoder at level i. Let Q be the $[[n,k]]$ CSS code associated to level i of $\mathcal{C}^*$, with an associated CSS encoding map* Enc *(see Definition 3.32). Given some graph*

$G_{\text{run}} = (V_{\text{run}}, E_{\text{run}})$ *that contains* $G^{\mathcal{C}}_{i-1,i,i+1} \subseteq G_{\text{run}}$ *as a subgraph and some*

$$\begin{aligned} \eta_{\text{in}} &\leq m \\ \gamma_{\text{in}} &\leq \frac{1}{50w^7T2^{T+1}} \\ \eta_{\text{run}} &\leq \eta_{\text{in}} \\ \gamma_{\text{run}} &\leq \gamma_{\text{in}}, \end{aligned} \tag{26}$$

let

$$\begin{aligned} \gamma_{\text{out}} &= \frac{(w+1)T2^{T+6}\gamma_{\text{run}}}{\gamma_{\text{dec}}} \\ \mathcal{E}_{\text{in}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ \gamma_{\text{in}}) \\ \mathcal{E}_{\text{run}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ \gamma_{\text{run}}) \\ \mathcal{E}_{\text{out}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ \gamma_{\text{out}}). \end{aligned}$$

Define the decorated codes (see Definition 3.23) $D_{\text{in}} = (Q,\ \text{Enc},\ \mathcal{E}_{\text{in}})$ *and* $D_{\text{out}} = (Q,\ \text{Enc},\ \mathcal{E}_{\text{out}})$. *Let* $\bar{\mathcal{O}} = \mathcal{I}_k : \mathbb{C}^{2^k \times 2^k} \to \mathbb{C}^{2^k \times 2^k}$ *be the identity channel.*

Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, \mathcal{E}^{\sqcup T}_{\text{run}}), D_{\text{in}}, D_{\text{out}})$ *for* $\bar{\mathcal{O}}$, *where* $\mathcal{Q}$ *is an adaptive quantum circuit using quantum space* $N = O(nw)$ *and time* $T = O(w)$.

The proof of Proposition 8.1 has a similar structure as the proof of Proposition 5.1 in Section 5.3, as well as the proof of Proposition 7.1 in Appendix A.

An important difference is that here in Proposition 8.1, we must obtain a decorated output code that can have the specified family of bad sets $\mathcal{E}_{\text{out}} = \mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ \gamma_{\text{out}})$ satisfying $\gamma_{\text{out}} < \gamma_{\text{in}}$. That is, here we must actually reduce the size of the error, rather than simply controlling its growth as in other gadgets. We accomplish this error reduction by arguing that if the output error were too large, then the small-set flip decoder in Algorithm 5 could not have terminated, which gives a contradiction. Related techniques are for instance used in [FGL18]. For readability, we have postponed the full proof to Appendix B.

9 Basic Gadgets

In this section, we present basic gadgets that are (in some similar form) well-known or folklore in the literature. However, we provide formal result statements and brief proofs in our fault-tolerance framework so that we can integrate these gadgets with our other gadgets.

9.1 *CNOT* Gadgets

We begin with a transversal *CNOT* gadget between two identical codeblocks.

Lemma 9.1. *For* $r \in \mathbb{N}$ *and* $1 \leq i \leq r-1$, *let* $\mathcal{C}^*$ *be an* r*-dimensional cochain complex. Let* Q *be the* $[[n,k]]$ *CSS code associated to level* i *of* $\mathcal{C}^*$, *with an associated CSS encoding map* Enc *(see Definition 3.32).*

Given some graph $G_{\text{run}} = (V_{\text{run}}, E_{\text{run}})$ with $C^i \subseteq V_{\text{run}}$ and some $\eta_{\text{in}}, \gamma_{\text{in}}, \gamma_{\text{run}} > 0$, let

$$\begin{aligned}\gamma_{\text{out}} &= 2\gamma_{\text{in}} + \gamma_{\text{run}}\\ \mathcal{E}_{\text{in}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ \gamma_{\text{in}})\\ \mathcal{E}_{\text{run}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ \gamma_{\text{run}})\\ \mathcal{E}_{\text{in}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ \gamma_{\text{out}}).\end{aligned}$$

Define the decorated codes (see Definition 3.23) $D_{\text{in}} = (Q,\ \text{Enc},\ \mathcal{E}_{\text{in}})^{\sqcup 2}$ and $D_{\text{out}} = (Q,\ \text{Enc},\ \mathcal{E}_{\text{out}})^{\sqcup 2}$. Let $\bar{\mathcal{O}} : \mathbb{C}^{2^{[k]\sqcup[k]}\times 2^{[k]\sqcup[k]}} \to \mathbb{C}^{2^{[k]\sqcup[k]}\times 2^{[k]\sqcup[k]}}$ be the channel acting on two blocks of k qubits that applies transversal $CNOT$ between the blocks, meaning that $CNOT_{(1,x),(2,x)}$ is applied for every $x \in [k]$, where our $2k$ logical qubits are labeled by the set $[2] \times [k] = [k] \sqcup [k]$. Equivalently, $\bar{\mathcal{O}}(\rho) = CNOT^{\otimes k}\, \rho\, CNOT^{\otimes k}$.

Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup 2})^{\sqcup T}), D_{\text{in}}, D_{\text{out}})$ for $\bar{\mathcal{O}}$, where $\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = 2n$ and time $T = 1$.

Proof. The desired adaptive quantum circuit $\mathcal{Q}$ is simply the depth-1 circuit that applies transversal physical $CNOT$, meaning that physical $CNOT_{(1,c),(2,c)}$ is applied for every $c \in C^i$, where our $2n$ physical qubits are labeled by the set $[2] \times C^i = C^i \sqcup C^i$.

In the absence of noise, this physical circuit implements the desired logical $CNOT$ gates, as for every $x, x' \in \mathbb{F}_2^k$, then

$$\begin{aligned}CNOT^{\otimes n} \left|\text{Enc}(x)\right\rangle \left|\text{Enc}(x')\right\rangle &= \sum_{y\in\text{Enc}(x), y'\in\text{Enc}(x')} \left|y\right\rangle \left|y+y'\right\rangle\\ &= \sum_{y\in\text{Enc}(x), y''\in\text{Enc}(x+x')} \left|y\right\rangle \left|y''\right\rangle\\ &= \left|\text{Enc}(x)\right\rangle \left|\text{Enc}(x+x')\right\rangle,\end{aligned}$$

where the second equality above holds because $\text{Enc} : \mathbb{F}_2^k \xrightarrow{\sim} H^i(\mathcal{C}) = Z^i(\mathcal{C})/B^i(\mathcal{C})$ is a linear map.

Now consider running the circuit $\mathcal{Q}[\mathcal{F}]$ with noise from a $\mathcal{E}_{\text{run}}^{\sqcup 2}$-avoiding fault $\mathcal{F}$ and a $\mathcal{E}_{\text{in}}^{\sqcup 2}$-avoiding Pauli error on the input. Within every connected component of $G_{\text{run}} \sqcup G_{\text{run}}$ with $\geq \eta_{\text{in}}$ vertices, at most γ_{in}-fraction of the qubits receive input errors from the first codeblock, at most γ_{in}-fraction of the qubits receive input errors from the second codeblock, and at most γ_{run}-fraction of the qubits receive errors from the fault $\mathcal{F}$. Note that the connected component must be contained within one of the two copies of G_{run}, yet the physical $CNOT$ gates can propagate errors between qubits at the same positions in the two blocks, and hence we can have γ_{in}-fraction corruptions from each block. Thus the total fraction of qubits in the connected component with Pauli errors is $\leq 2\gamma_{\text{in}} + \gamma_{\text{run}}$, as desired. □

We now present a transversal $CNOT$ between two codes of different dimensions. Below, recall from Definition 3.23 that we use $\sqcup$ to denote the combination of disjoint code blocks.

Lemma 9.2. *For $r \in \mathbb{N}$ define $\mathcal{A}^*$, $\mathcal{M}^{\mathcal{A},*}$, $\text{Enc}^{\mathcal{A}}$, $\mathcal{B}^*$, $\mathcal{M}^{\mathcal{B},*}$, $L^{\mathcal{B},1}$, $\text{Enc}^{\mathcal{B}}$ $\mathcal{C}^* = \mathcal{A}^* \otimes \mathcal{B}^*$, $\mathcal{M}^{\mathcal{C},*} = \mathcal{M}^{\mathcal{A},*} \otimes \mathcal{M}^{\mathcal{B},*}$, $\text{Enc}^{\mathcal{C}} = \text{Enc}^{\mathcal{A}} \otimes \text{Enc}^{\mathcal{B}}$ as in Section 5.1 (though in this lemma we make no assumptions regarding small-set flip decodability). Let Q_{con} be the $[[n_{\text{con}}, k_{\text{con}}]]$ CSS code consisting of $|L^{\mathcal{B},1}|$ copies (labeled by the set $L^{\mathcal{B},1}$) of the CSS code associated to level $i-1$ of $\mathcal{A}^*$, and let Q_{tar} be the $[[n_{\text{tar}}, k_{\text{tar}}]]$ CSS code associated to level i of $\mathcal{C}^*$. The physical qubits of Q_{tar} are naturally labeled by the set C^i, while the physical qubits of Q_{con} are labeled by the set $A^{i-1} \times L^{\mathcal{B},1} \subseteq A^{i-1} \times B^1 \subseteq C^i$.*

Given some graph $G_{\text{run}} = (V_{\text{run}}, E_{\text{run}})$ *with* $C^i \subseteq V_{\text{run}}$ *and some* $\eta_{\text{in}}, \gamma_{\text{in}}, \gamma_{\text{run}} > 0$*, let*

$$\begin{aligned}\gamma_{\text{out}} &= 2\gamma_{\text{in}} + \gamma_{\text{run}}\\ \mathcal{E}_{\text{in}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ \gamma_{\text{in}})\\ \mathcal{E}_{\text{run}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ \gamma_{\text{run}})\\ \mathcal{E}_{\text{out}} &= \mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ \gamma_{\text{out}}).\end{aligned}$$

Let $Q = Q_{\text{con}} \sqcup Q_{\text{tar}}$ *with encoding map* $\text{Enc} = (\text{Enc}^{\mathcal{A}})^{L^{\mathcal{B},1}} \sqcup \text{Enc}^{\mathcal{C}}$*. Define the decorated codes (see Definition 3.23)* $D_{\text{in}} = (Q,\ \text{Enc},\ \mathcal{E}_{\text{in}}^{\sqcup 2})$ *and* $D_{\text{out}} = (Q,\ \text{Enc},\ \mathcal{E}_{\text{out}}^{\sqcup 2})$*, where we associate each set of physical qubits* $A^{i-1} \times L^{\mathcal{B},1} \subseteq V_{\text{run}}$ *and* $C^i \subseteq V_{\text{run}}$ *with a separate copy of* G_{run}*.*

Let $\bar{\mathcal{O}} : \mathbb{C}^{2^{[k_{\text{con}}]\sqcup[k_{\text{tar}}]} \times 2^{[k_{\text{con}}]\sqcup[k_{\text{tar}}]}} \to \mathbb{C}^{2^{[k_{\text{con}}]\sqcup[k_{\text{tar}}]} \times 2^{[k_{\text{con}}]\sqcup[k_{\text{tar}}]}}$ *be the channel acting on* $k_{\text{con}} + k_{\text{tar}}$ *logical qubits labeled by* $(M^{\mathcal{A},i-1} \times L^{\mathcal{B},1}) \sqcup M^{\mathcal{C},i}$ *that applies applies* $CNOT_{x,x}$ *for every* $x \in M^{\mathcal{A},i-1} \times L^{\mathcal{B},1} \subseteq M^{\mathcal{C},i}$*, with control in the first (size-*k_{con}*) block and target in the second (size-*k_{tar}*) block. Equivalently,* $\bar{\mathcal{O}}(\rho) = CNOT^{\otimes k_{\text{con}}}\ \rho\ CNOT^{\otimes k_{\text{con}}}$*.*

Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup 2})^{\sqcup T}), D_{\text{in}}, D_{\text{out}})$ *for* $\bar{\mathcal{O}}$*, where* $\mathcal{Q}$ *is an adaptive quantum circuit using quantum space* $N = n_{\text{con}} + n_{\text{tar}}$ *and time* $T = 1$*.*

Proof. The desired adaptive quantum circuit $\mathcal{Q}$ is simply the depth-1 circuit that applies transversal physical $CNOT$ between the entire first (control) codeblock and the subset of the second (target) codeblock whose qubit labels are shared with the first codeblock. That is we apply physical $CNOT_{x,x}$ for every $x \in A^{i-1} \times L^{\mathcal{B},1} \subseteq C^i$, where the control is in the first (size-n_{con}) block and the target is in the second (size-n_{tar}) block.

In the absence of noise, this physical circuit implements the desired logical $CNOT$ gates, as for every $x \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$, $x' \in \mathcal{M}^{\mathcal{B},i}$, then

$$\begin{aligned}CNOT^{\otimes k_{\text{con}}} \left|(\text{Enc}^{\mathcal{A}})^{L^{\mathcal{B},1}}(x)\right\rangle \left|\text{Enc}^{\mathcal{C}}(x')\right\rangle &= \sum_{y \in (\text{Enc}^{\mathcal{A}})^{L^{\mathcal{B},1}}(x), y' \in \text{Enc}^{\mathcal{C}}(x')} |y\rangle \left|y + y'\right\rangle\\ &= \sum_{y \in (\text{Enc}^{\mathcal{A}})^{L^{\mathcal{B},1}}(x), y'' \in \text{Enc}^{\mathcal{C}}(x+x')} |y\rangle \left|y''\right\rangle\\ &= \left|(\text{Enc}^{\mathcal{A}})^{L^{\mathcal{B},1}}(x)\right\rangle \left|\text{Enc}^{\mathcal{C}}(x+x')\right\rangle.\end{aligned}$$

Here we view $x \in \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$ as also belonging to $\mathcal{M}^{\mathcal{B},i} \supseteq \mathcal{M}^{\mathcal{A},i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$ (by simply padding x with 0s in the additional components), and similarly we view $y' \in \mathcal{A}^{i-1} \otimes \mathbb{F}_2^{L^{\mathcal{B},1}}$ as also belonging to $\mathcal{C}^i$. The second equality above then holds because by definition $(\text{Enc}^{\mathcal{A}})^{L^{\mathcal{B},1}}(x) \subseteq \text{Enc}^{\mathcal{C}}(x)$.

The analysis of $\mathcal{Q}$ in the presence of noise, which proves the fault-tolerance of our gadget, is essentially identical to that in the proof of Lemma 9.1, so we omit it to avoid redundancy. □

9.2 Hadamard Gadget

We now present a transversal Hadamard gadget on a codeblock. Note that this gadget has different (specifically, dual) input and output codes. We will later combine this gadget with our code switching gadgets to obtain a constant-overhead gadget for logical Hadamard that has the same input and out codes (see Proposition 10.4).

Lemma 9.3. *For* $r \in \mathbb{N}$ *and* $1 \le i \le r-1$*, let* $\mathcal{C}^*$ *be an* r*-dimensional cochain complex. Let* Q_{in} *be the* $[[n,k]]$ *CSS code associated to level* i *of the cochain complex* $\mathcal{C}^*$*, with an associated CSS*

encoding map $\mathrm{Enc}_{\mathrm{in}}$ *(see Definition 3.32). Let* Q_{out} *be the* $[[n,k]]$ *CSS code associated to level* i *of the chain complex* $\mathcal{C}_*$*, with associated CSS encoding map* $\mathrm{Enc}_{\mathrm{out}}$*. Assume that for every* $x, x' \in \mathbb{F}_2^k$ *and every* $y \in \mathrm{Enc}_{\mathrm{in}}(x)$*,* $y' \in \mathrm{Enc}_{\mathrm{out}}(x')$*, we have*

$$x \cdot x' = y \cdot y' \tag{27}$$

Given some graph $G_{\mathrm{run}} = (V_{\mathrm{run}}, E_{\mathrm{run}})$ *with* $C^i \subseteq V_{\mathrm{run}}$ *and some* $\eta_{\mathrm{in}}, \gamma_{\mathrm{in}}, \gamma_{\mathrm{run}} > 0$*, let*

$$\begin{aligned}
\gamma_{\mathrm{out}} &= \gamma_{\mathrm{in}} + \gamma_{\mathrm{run}} \\
\mathcal{E}_{\mathrm{in}} &= \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{in}}) \\
\mathcal{E}_{\mathrm{run}} &= \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{run}}) \\
\mathcal{E}_{\mathrm{in}} &= \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{out}}).
\end{aligned}$$

Define the decorated codes (see Definition 3.23) $D_{\mathrm{in}} = (Q_{\mathrm{in}},\ \mathrm{Enc}_{\mathrm{in}},\ \mathcal{E}_{\mathrm{in}})$ *and* $D_{\mathrm{out}} = (Q_{\mathrm{out}},\ \mathrm{Enc}_{\mathrm{out}},\ \mathcal{E}_{\mathrm{out}})$*. Let* $\bar{\mathcal{O}} : \mathbb{C}^{2^k \times 2^k} \to \mathbb{C}^{2^k \times 2^k}$ *be the channel acting on* k *qubits that applies a Hadamard gate to every qubit. Equivalently,* $\bar{\mathcal{O}}(\rho) = H^{\otimes k}\, \rho\, H^{\otimes k}$*.*

Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, \mathcal{E}_{\mathrm{run}}^{\sqcup T}), D_{\mathrm{in}}, D_{\mathrm{out}})$ *for* $\bar{\mathcal{O}}$*, where* $\mathcal{Q}$ *is an adaptive quantum circuit using quantum space* $N = n$ *and time* $T = 1$*.*

Proof. The desired adaptive quantum circuit $\mathcal{Q}$ is simply the depth-1 circuit that applies transversal physical Hadamard, that is, applies $H^{\otimes n}$.

In the absence of noise, this physical circuit implements the desired logical H gates, as for every $x \in \mathbb{F}_2^k$, then

$$\begin{aligned}
H^{\otimes n} \mathrm{Enc}_{\mathrm{in}} |x\rangle &= \sum_{y \in \mathrm{Enc}_{\mathrm{in}}(x)} H^{\otimes n} |y\rangle \\
&\propto \sum_{y \in \mathrm{Enc}_{\mathrm{in}}(x)} \sum_{y' \in \mathbb{F}_2^n} (-1)^{y \cdot y'} |y'\rangle \\
&= \sum_{y \in \mathrm{Enc}_{\mathrm{in}}(x),\ y' \in Z_i(\mathcal{C})} (-1)^{y \cdot y'} |y'\rangle ,
\end{aligned}$$

while

$$\begin{aligned}
\mathrm{Enc}_{\mathrm{out}}\, H^{\otimes k} |x\rangle &\propto \mathrm{Enc}_{\mathrm{out}} \sum_{x' \in \mathbb{F}_2^k} (-1)^{x \cdot x'} |x'\rangle \\
&\propto \sum_{x' \in \mathbb{F}_2^k} (-1)^{x \cdot x'} \sum_{y' \in \mathrm{Enc}_{\mathrm{out}}(x')} |y'\rangle \\
&\propto \sum_{y \in \mathrm{Enc}_{\mathrm{in}}(x)} \sum_{y' \in Z_i(\mathcal{C})} (-1)^{y \cdot y'} |y'\rangle ,
\end{aligned}$$

where the final $\propto$ above holds by (27). The two equations above have the same RHS, so we must have $H^{\otimes n} \mathrm{Enc}_{\mathrm{in}} = \mathrm{Enc}_{\mathrm{out}}\, H^{\otimes k}$. Thus $\mathcal{Q} = H^{\otimes n}$ implements the desired logical $H^{\otimes k}$ in the absence of noise.

Now consider running the circuit $\mathcal{Q}[\mathcal{F}]$ with noise from a $\mathcal{E}_{\mathrm{run}}$-avoiding fault $\mathcal{F}$ and a $\mathcal{E}_{\mathrm{in}}$-avoiding Pauli error on the input. Within every connected component of G_{run} with $\geq \eta_{\mathrm{in}}$ vertices, at most γ_{in}-fraction of the qubits receive input errors, and at most γ_{run}-fraction of the qubits receive fault errors. Thus the total fraction of qubits in the connected component with Pauli errors is $\leq \gamma_{\mathrm{in}} + \gamma_{\mathrm{run}}$, as desired. □

Remark 9.4. *Note that the condition (27) in Lemma 9.3 is by definition satisfied for* $\mathrm{Enc}_{\mathrm{in}} = \mathrm{Enc}^i$, $\mathrm{Enc}_{\mathrm{out}} = \mathrm{Enc}_i$ *for* Enc^i, Enc_i *defined as in Lemma 3.33, and therefore also for the product maps* Enc^i, Enc_i *defined as in Definition 3.34.*

9.3 Logical Pauli X and Z Measurement Gadgets

In this section, we show that measuring out an entire code block in the Pauli Z basis induces logical Z measurements on all of the encoded logical qubits. The same analysis implies an analogous result for Pauli X.

Algorithm 6: Logical Pauli Z measurement gadget in Lemma 9.5; all variables are defined as in the lemma. Here we write $\delta = \delta^{\mathcal{C}}$.

Input : $\sigma \in \mathbb{C}^{\mathcal{C}^i \times \mathcal{C}^i}$ that is a $\mathcal{E}_{\mathrm{in}}$-deviation of $\mathrm{Enc}(\rho)$ for $\rho \in \mathbb{C}^{2^k \times 2^k}$
Output: Classical string $x \in \mathbb{F}_2^k$ giving outcome of Pauli Z measurements on ρ

1 **Function** `Measure`$(\sigma; i, \mathcal{C}^*)$**:**
2 Measure Pauli Z on all n qubits, and let $z \in \mathcal{C}^i$ be the measurement outcome
3 Run `SSFlipSyn`$(\delta(z); i, \mathcal{C}^*)$ from Algorithm 2, and let $a^i \in \mathcal{C}^i$ be the output
4 **return** $\mathrm{Enc}^{-1}(z + a^i)$

Lemma 9.5. *For* $r \in \mathbb{N}$ *and* $1 \le i \le r-1$, *let* $\mathcal{C}^*$ *be an* r*-dimensional cochain complex of locality* w *with a* $(m, 0)$*-small-set flip decoder (see Definition 4.1) at level* i. *Let* Q *be the* $[[n, k]]$ *CSS code associated to level* i *of* $\mathcal{C}^*$, *with an associated CSS encoding map* Enc *(see Definition 3.32). Given some graph* $G_{\mathrm{in}} = (V_{\mathrm{in}}, E_{\mathrm{in}})$ *that contains* $G^{\mathcal{C}}_{i-1,i,i+1} \subseteq G_{\mathrm{in}}$ *as a subgraph and some*

$$\eta_{\mathrm{in}} \le \min\{m,\ d_i(\mathcal{C})\}$$
$$\gamma_{\mathrm{in}} \le \frac{1}{50w^7},$$

let

$$\mathcal{E}_{\mathrm{in}} = \mathcal{E}(G_{\mathrm{in}},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{in}}),$$

and define the decorated code (see Definition 3.23) $D_{\mathrm{in}} = (Q,\ \mathrm{Enc},\ \mathcal{E}_{\mathrm{in}})$.

Let $\bar{\mathcal{O}} : \mathbb{C}^{2^k \times 2^k} \to \mathbb{C}^{2^k \times 2^k}$ *be the channel that takes as input* $k_{\mathrm{in}} = k$ *logical qubits (and* $M_{\mathrm{in}} = 0$ *classical bits) and outputs* $M_{\mathrm{out}} = k$ *classical bits (and* $k_{\mathrm{out}} = 0$ *qubits), which is given by measuring the* k *input qubits in the Pauli* Z *basis, and outputting the measurement outcomes (see Definition 3.25).*

Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, \emptyset), D_{\mathrm{in}}, \emptyset)$ *for* $\bar{\mathcal{O}}$, *where* $\mathcal{Q}$ *is an adaptive quantum circuit using quantum space* $N = n$ *and time* $T = 2$.

Proof. The proof will be similar to that of Proposition 5.1, but even simpler as here there is no product involved. We will therefore omit some details to avoid redundancy.

The desired adaptive quantum circuit $\mathcal{Q}$ is given in Algorithm 6. By definition line 2 and line 3 each use 1 timestep, so the entire circuit uses $T = 2$ timesteps. Furthermore, the circuit by definition uses $N = |C^i| = n$ physical qubits.

Our goal is to show that for every $\ell \in \mathbb{N}$, every $\rho \in \mathbb{C}^{2^{k+\ell} \times 2^{k+\ell}}$, and every Pauli $\mathcal{E}_{\text{in}}$-deviation[17] $\sigma = E \operatorname{Enc}(\rho) E'$ (for Paulis E, E' such that $\operatorname{supp}(E) \cup \operatorname{supp}(E') \subseteq C^i$ is $\mathcal{E}_{\text{in}}$-avoiding), the resulting output $\mathcal{Q}(\sigma) \in \mathbb{F}_2^k$ is the result of measuring the first (k-qubit) register of ρ. Note that by Definition 3.19, a fault $\mathcal{F}$ will only act on the post-measurement qubits in Algorithm 6, which are discarded (i.e. traced out) anyways. Hence the output of Algorithm 6 remains the same under arbitrary faults.

Write $E = Z^{e_Z} X^{e_X}$, $E' = X^{e'_X} Z^{e'_Z}$, and expand

$$\rho = \sum_{x,x' \in \mathbb{F}_2^k} \rho_{x,x'} \left|x\right\rangle \bar{x'} \otimes \nu_{x,x'}$$

for some $\nu_{x,x'} \in \mathbb{C}^{2^\ell \times 2^\ell}$. Then the input state just prior to the measurements in line 2 of Algorithm 6 is

$$E \operatorname{Enc}(\rho) E' = \sum_{x,x',y,y'} \rho_{x,x'} Z^{e_Z} \left|y + e_X\right\rangle \left\langle y' + e'_X\right| Z^{e'_Z} \otimes \nu_{x,x'}$$

where the sum above is over all $x, x' \in \mathbb{F}_2^k$, $y \in \operatorname{Enc}(x)$, and $y' \in \operatorname{Enc}(x')$. Performing the measurements in line 2 then collapses the state to

$$\sum_{z \in C^i} \sum_{x,x',y,y'} \rho_{x,x'} Z^{e_Z} \left|z\right\rangle \left\langle z\right| Z^{e'_Z} \otimes \nu_{x,x'}, \tag{28}$$

where the inner sum above is over all $x, x' \in \mathbb{F}_2^k$, $y \in \operatorname{Enc}(x)$, $y' \in \operatorname{Enc}(x')$ for which

$$y + e_X = z = y' + e'_X. \tag{29}$$

Let $S_{\text{in}} = \operatorname{supp}(E) \cup \operatorname{supp}(E')$, so that by definition S_{in} is $\mathcal{E}_{\text{in}}$-avoiding.

Let $S_{\text{syn}} \subseteq C^i \sqcup C^{i+1}$ be the footprint (see Definition 4.3) of the call to $\texttt{SSFlipSyn}(\delta(z); i, \mathcal{C}^*)$ in line 3 of Algorithm 6.

Claim 9.6. *The set $S_{\text{in}} \cup S_{\text{syn}}$ is $\mathcal{E}(G_{\text{in}},\ \eta_{\text{in}},\ 5w^3\gamma_{\text{in}})$-avoiding.*

The proof of Claim 9.6 is nearly identical to that of Claim 5.6 (just with the subscript "in" replacing the subscript "run," and the coefficient $5w^3$ replacing $15w^3$ as by definition here S_{in} is $\mathcal{E}_{\text{in}}$-avoiding), so we omit it to avoid redundancy.

Furthermore, similarly as described following Claim 5.6 in the proof of Proposition 5.1, because $\gamma_{\text{in}} \leq 1/5w^3$, the subgraph of G_{in} induced by $S_{\text{in}} \cup S_{\text{syn}}$ has no connected components containing $\geq \eta_{\text{in}}$ vertices. Therefore because $\eta_{\text{in}} \leq m$, within every such connected component, the restricted execution $\texttt{SSFlipSyn}(\delta(z|_{V \cap C^i}); i, \mathcal{C}^*)$ outputs $a^i|_{V \cap C^i}$, so that $\delta(z + a^i|_{V \cap C^i}) = \delta(e_X + a^i|_{V \cap C^i}) = 0$. We analogously have $\delta(e'_X + a^i|_{V \cap C^i}) = 0$.

We now define S_{err} and $\tilde{a}^{i-1}, \tilde{a}^i$ (resp. S'_{err} and $(\tilde{a}^{i-1})', (\tilde{a}^i)'$) to be the footprint and output of $\texttt{SSFlipErr}(e_X + a^i; i, \mathcal{C}^*)$ (resp. $\texttt{SSFlipErr}(e'_X + a^i; i, \mathcal{C}^*)$), respectively. We prove the following claim for the unprimed variables $e_X, \ldots$, but it analogously applies to the primed variables $e'_X, \ldots$.

Claim 9.7. *The set $S_{\text{in}} \cup S_{\text{syn}} \cup S_{\text{err}}$ is $\mathcal{E}(G_{\text{in}},\ \eta_{\text{in}},\ 50w^7\gamma_{\text{in}})$-avoiding.*

[17] See Footnote 12.

The proof of Claim 9.7 is nearly identical to that of Claim 5.7, so we omit it to avoid redundancy.

Furthermore, similarly as described following Claim 5.7 in the proof of Proposition 5.1, we must have $\tilde{a}^i = 0$. Also, because $\gamma_{\text{in}} \leq 1/50w^7$, we conclude that the subgraph of G_{in} induced by $S_{\text{in}} \cup S_{\text{syn}} \cup S_{\text{err}}$ has no connected components containing $\geq \eta_{\text{in}}$ vertices. Then because $\eta_{\text{in}} \leq m$, by the assumption that $\mathcal{C}^*$ has an m-small-set error-flip decoder at level i, we conclude that within every connected component V we have $\delta(\tilde{a}^{i-1}|_{V \cap C^{i-1}}) = e_X + a^i|_{V \cap C^i}$, and thus

$$\delta(\tilde{a}^{i-1}) = e_X + a^i. \tag{30}$$

As the analysis above also applies to the primed variables, we also have

$$\delta((\tilde{a}^{i-1})') = e'_X + a^i.$$

The above equations combined with (29) imply that

$$y + y' = e_X + e'_X = \delta(\tilde{a}^{i-1} + (\tilde{a}^{i-1})') \in B^i(\mathcal{C}).$$

Thus we must have $x = x'$, so if the post-measurement state in (28) is nonzero, then we must have $e_X + e'_X \in B^i(\mathcal{C})$, and the state can be equivalently written as

$$\sum_{x \in \mathbb{F}_2^k} \sum_{y \in \text{Enc}(x)} \rho_{x,x'} Z^{e_Z} |y + e_X\rangle \langle y + e_X| Z^{e'_Z} \otimes \nu_{x,x'}. \tag{31}$$

Now similarly as in the proof of Proposition 5.1, if $e_Z + e'_Z \notin Z_i(\mathcal{C}) = B^i(\mathcal{C})^\perp$, then the phases induced by the Paulis $Z^{e_Z}, Z^{e'_Z}$ in the inner sum in (31) cancel, and the expression becomes 0. Thus assume $e_Z + e'_Z \in Z_i(\mathcal{C})$. Now because $\text{supp}(e_Z + e'_Z) \subseteq S_{\text{in}}$ is $\mathcal{E}_{\text{in}}$-avoiding, the subgraph of $G_i^{\mathcal{C}} \subseteq G_{\text{in}}$ induced by this set contains no connected components with $\geq \eta_{\text{in}}$ vertices. Therefore by the assumption that $\eta_{\text{in}} \leq d_i(\mathcal{C})$, we cannot have $e_Z + e'_Z \in Z_i(\mathcal{C})$ belonging to a nontrivial homology class, that is, we msut have $e_Z + e'_Z \in B_i(\mathcal{C})$. Thus all terms in the sum in (31) receive the same phase $(-1)^{(e_Z + e'_Z) \cdot e_X}$ from the Paulis $Z^{e_Z}, Z^{e'_Z}$, so we can remove these Paulis while simply inducing a global phase on the overall expression. Tracing out the code register of the resulting state yields

$$\sum_{x \in \mathbb{F}_2^k} \rho_{x,x} \nu_{x,x} = \text{tr}_{[k]}(\rho),$$

where for a given $x \in \mathbb{F}_2^k$, by (30), Algorithm 6 will return

$$\text{Enc}^{-1}(z + a^i) = \text{Enc}^{-1}(y + e_X + a^i) = \text{Enc}^{-1}(y + B^i(\mathcal{C})) = x,$$

as desired. □

10 Applications

In this section, we combine our gadgets from the sections above to create new gadgets for constant-overhead fault-tolerant logical operations.

In Section 10.1 below, we describe the lossless-expander-based codes with which we instantiate our gadgets. The subsequent sections apply these codes to construct our desired fault-tolerant gadgets in Theorem 1.1. Recall that Proposition 7.1 implies item 1 of Theorem 1.1. Section 10.2 and Section 10.3 below prove item 2 and item 3 respectively in Theorem 1.1. Section 10.4 then proves item 4 and item 5 in Theorem 1.1. Section 10.5 highlights a surprising consequence of item 1 in Theorem 1.1 (that follows from Proposition 7.1 and Proposition 5.1), namely, that we can prepare 2-dimensional product code states in bulk with constant space-time overhead.

Remark 10.1. *As Theorem 1.1 is stated with fault-tolerance defined in the sense of a threshold under locally stochastic noise, it follows from the results listed above combined with Proposition 8.1, Lemma 3.29, and Lemma 3.31.*

10.1 Code Instantiation

In this section, we describe the codes arising from tensor products of 1-dimensional cochain complexes from lossless expanders that we will use to instantiate our gadgets throughout the remainder of Section 10.

We first fix some $r \in \mathbb{N}$ and $\epsilon > 0$, and we define $\epsilon = \epsilon(r, 1/4)$ to be the value defined in Proposition 4.6. For this value of ϵ, we then define $\mu, R, \Delta_L, \Delta_R$ as in Corollary 3.10, and we fix some (Δ_L, Δ_R)-biregular (μ, ϵ)-lossless expander $G = (V_L \sqcup V_R, E)$ with specified set[18] $L^{G,1} \subseteq V_R$ from the family in Corollary 3.10.

Following the notation in Lemma 3.33, let $\mathcal{C}^{G,*}$ denote the 1-dimensional cochain complex associated to G. We fix some information sets $M^{G,0} \subseteq V_L = C^{G,0}$, $M^{G,1} \subseteq V_R = C^{G,1}$ for $Z^0(\mathcal{C}^G)$, $Z_1(\mathcal{C}^G)$ respectively such that $L^{G,1} \subseteq M^{G,1}$. We then define $\mathcal{M}^{G,*}$ and $\mathrm{Enc}^G : \mathcal{M}^{G,*} \to \mathcal{C}^{G,*}$ as in Lemma 3.33.

In this section, we consider codes associated to cochain complexes of the form $\mathcal{C}^* = \mathcal{C}^{(1)^*} \otimes \cdots \otimes \mathcal{C}^{(r)^*}$ with encoding map $\mathrm{Enc}^{\mathcal{C}} = \mathrm{Enc}^{(1)} \otimes \cdots \otimes \mathrm{Enc}^{(r)}$ as defined in Definition 3.34, where each cochain complex $\mathcal{C}^{(h)^*}$ equals either $\mathcal{C}^{G,*}$ or its dual complex $\mathcal{C}^G_*$. By Proposition 4.6, this cochain complex $\mathcal{C}^*$ (resp. the chain complex $\mathcal{C}_*$) has a $(m, \gamma_{\mathrm{dec}})$-small-set flip decoder at every level $0 \le i \le r-1$ (resp. $1 \le i \le r$) for $m = \mu|V_L|/(r\Delta_L^{r+1} + 1)$ and $\gamma_{\mathrm{dec}} = 1/10$.

By Corollary 3.10 and Proposition 4.6, we similarly have a $(m, \gamma_{\mathrm{dec}})$-small-set flip decoder for the cochain complex $\mathcal{C}^{(1)^*} \otimes \cdots \otimes \mathcal{C}^{(r-1)^*} \otimes \mathcal{B}^*_{\bar{L}}$ (and its associated chain complex), where $\mathcal{B}^*_{\bar{L}}$ is defined as in Section 5.1 for $\mathcal{B} = \mathcal{C}^{(r)^*}$ and $L^{\mathcal{B},1} = L^{G,1}$. Note that this statement holds when replacing any of the r factors $\mathcal{C}^{(h)^*}$ with $\mathcal{B}^*_{\bar{L}}$.

Because the classical codes $Z^i(\mathcal{C}^G), Z_i(\mathcal{C}^G)$ have distance $\ge \mu|V_L|$ by Lemma 3.4, the quantum code at every level $1 \le i \le r-1$ of $\mathcal{C}^*$ must have distance $\min\{d^i(\mathcal{C}), d_i(\mathcal{C})\} \ge \mu|V_L|$ (see e.g. [GL25, Lemma 3.36], which in turn is based on arguments in [TZ14, BH14]).

Here we think of $|V_L| = \Theta(|V_R|)$ as growing while r and the resulting $r\mu, R, \Delta_L, \Delta_R$ remain constant. Hence the code associated to some level $1 \le i \le r-1$ of $\mathcal{C}^*$ has length

$$n = \Theta_r(|V_L|^r)$$

and distance

$$d \ge \mu|V_L| = \Omega_r(n^{1/r}).$$

We will always assume that i of the values $h \in [r]$ have $\mathcal{C}^{(h)^*} = \mathcal{C}^{G,*}$ and $r-i$ of these values have $\mathcal{C}^{(h)^*} = \mathcal{C}^G_*$, so that the code at level i has dimension

$$k \ge (R|V_R|)^r = \Theta_r(n)$$

by Corollary 3.10 and Proposition 3.14. Also by definition, the locality w of $\mathcal{C}^*$ does not depend on $|V_L| = \Theta(|V_R|)$, so that

$$w \le O_r(1).$$

[18] The set $L^{G,1}$ here is called S_i in Corollary 3.10.

10.2 Logical Qubit Permutations

In this section, we apply our code switching gadgets to perform highly flexible and parallel logical permutations with constant space-time overhead. We begin with a gadget that arbitrarily permutes the "slabs," or axis-parallel codimension-1 hyperplanes, within a set (of arbitrary size $\kappa \in \mathbb{N}$) of r-dimensional hypercubes $(L^{G,1})^r$ of logical qubits.

Proposition 10.2. *For arbitrary fixed $r \geq 3$, define all variables as in Section 10.1, and let Q be the $[[n,k]]$ code associated to some level $1 \leq i \leq r-1$ of $\mathcal{C}^* = \mathcal{C}^{(1)^*} \otimes \cdots \otimes \mathcal{C}^{(r)^*}$ with encoding map $\mathrm{Enc}^{\mathcal{C}}$ as in Section 10.1.*

There exist sufficiently small $\gamma_{\mathrm{in}}(r) > 0$ and $\zeta_{\mathrm{run}}(r) > 0$ such that for every

$$\begin{aligned} \eta_{\mathrm{in}} &\leq m \\ \gamma_{\mathrm{in}} &\leq \gamma_{\mathrm{in}}(r) \\ \gamma_{\mathrm{run}} &\leq \zeta_{\mathrm{run}}(r) \cdot \gamma_{\mathrm{in}}, \end{aligned}$$

the following holds.

Let $G_{\mathrm{run}} = (V_{\mathrm{run}}, E_{\mathrm{run}}) := G^{\mathcal{C}}_{[r]}$ and

$$\begin{aligned} \mathcal{E}_{\mathrm{in}} = \mathcal{E}_{\mathrm{out}} &= \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{in}}) \\ \mathcal{E}_{\mathrm{run}} &= \mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{in}},\ \gamma_{\mathrm{run}}), \end{aligned}$$

and define the decorated code

$$D = (Q,\ \mathrm{Enc}^{\mathcal{C}},\ \mathcal{E}_{\mathrm{in}}).$$

Here the $n = |C^i|$ physical qubits of Q are naturally associated with the set $C^i \subseteq V_{\mathrm{run}}$.

For every $\kappa \in \mathbb{N}$, every $h \in [r]$, and every permutation[19] *$\pi : (L^{G,1})^{\sqcup\kappa} \to (L^{G,1})^{\sqcup\kappa}$, there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, ((\mathcal{E}_{\mathrm{run}}^{\sqcup\kappa})^{\sqcup u})^{\sqcup T}), D^{\sqcup\kappa}, D^{\sqcup\kappa})$ for $\bar{\mathcal{O}}$, where:*

- *$\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = O_r(\kappa \cdot n)$ and time $T = O_r(1)$, and $u = O(1)$.*
- *$\bar{\mathcal{O}} : \mathbb{C}^{(\mathcal{M}^{\mathcal{C},i})^\kappa \times (\mathcal{M}^{\mathcal{C},i})^\kappa} \to \mathbb{C}^{(\mathcal{M}^{\mathcal{C},i})^\kappa \times (\mathcal{M}^{\mathcal{C},i})^\kappa}$ is a channel that acts on $\kappa \cdot k$ qubits labeled by the set $(M^{\mathcal{C},i})^{\sqcup\kappa}$, that permutes the qubits in $((L^{G,1})^r)^{\sqcup\kappa} \subseteq (M^{\mathcal{C},i})^{\sqcup\kappa}$ as follows: under the isomorphism $((L^{G,1})^r)^{\sqcup\kappa} \cong (L^{G,1})^{h-1} \times (L^{G,1})^{\sqcup\kappa} \times (L^{G,1})^{r-h}$, then $\bar{\mathcal{O}}$ permutes the qubits according to $I^{\otimes h-1} \otimes \pi \otimes I^{\otimes r-h}$ (so that the qubit at position $(b_1, \ldots, b_h, \ldots, b_r) \in (L^{G,1})^{h-1} \times (L^{G,1})^{\sqcup\kappa} \times (L^{G,1})^{r-h}$ is moved to $(b_1, \ldots, \pi(b_h), \ldots, b_r)$); the qubits in $(M^{\mathcal{C},i})^{\sqcup\kappa} \setminus ((L^{G,1})^r)^{\sqcup\kappa}$ may be moved or reset arbitrarily.*

Analogously as described in Remark 6.2, the gadget $\mathcal{Q}$ in Proposition 10.2 invokes a constant number of gadgets previously presented in the paper. As a result, the physical qubits are naturally partitioned into some constant number $u = O(1)$ of blocks, each of which is naturally a subset of $C^{\sqcup\kappa} = V_{\mathrm{run}}^{\sqcup\kappa}$.

Proof of Proposition 10.2. We first present the desired circuit $\mathcal{Q}$ assuming that $\mathcal{C}^{(h)^*} = \mathcal{C}^{G,*}$ and $i \geq 2$; we then show how to relax these assumptions to also allow for $\mathcal{C}^{(h)^*} = \mathcal{C}^G_*$ or $i = 1$.

By the assumptions above, we may apply Proposition 5.1 with $\mathcal{A}^* = \bigotimes_{h' \in [r] \setminus \{h\}} \mathcal{C}^{(h')^*}$ and $\mathcal{B}^* = \mathcal{C}^{(h)^*}$ to map each of the κ input Q-codeblocks down to a set of $|L^{G,1}|$ codeblocks of the code Q' associated to level $i-1$ of $\mathcal{A}^*$ (so we obtain a total of $\kappa \cdot |L^{G,1}|$ codeblocks of Q').

[19] Recall that by definition $(L^{G,1})^{\sqcup\kappa} = [\kappa] \times L^{G,1}$.

To apply the logical permutation $I^{\otimes h-1} \otimes \pi \otimes I^{\otimes r-h}$ to this state, we first prepare an ancilla block of qubits, also labeled by the set $(A^{i-1} \times L^{G,1})^{\sqcup \kappa}$, and initialized arbitrarily. We swap our $\kappa \cdot |L^{G,1}|$ codeblocks of Q' into this ancilla block, and then swap the qubits back into the original block. However, when performing this second round of swaps, each Q'-block (of which there are $\kappa \cdot |L^{G,1}|$) with label $b \in (L^{G,1})^{\sqcup \kappa}$ is moved into the location with label $\pi(b) \in (L^{G,1})^{\sqcup \kappa}$. Each physical swap here is implemented with three $CNOT$ gates. We subsequently apply Proposition 6.1 to each of the κ sets of $|L^{G,1}|$ codeblocks of Q' to return to κ codeblocks of Q. Finally, we apply the error correction gadget in Proposition 8.1 to each resulting Q-codeblock.

If instead ${\mathcal{C}^{(h)}}^* = \mathcal{C}^G_*$ and $i \leq r-2$, an analogous circuit as described above works, with simply the roles of the X and Z bases swapped. The only remaining cases are when ${\mathcal{C}^{(h)}}^* = \mathcal{C}^{G,*}$ and $i = 1$, or when ${\mathcal{C}^{(h)}}^* = \mathcal{C}^G_*$ and $i = r-1$; we consider the former, as the latter is analogous.

Specifically, in this case, we cannot switch down with $\mathcal{A}^*$ and $\mathcal{B}^*$ as described above, as then the qubits would be in the code associated to level 0 of $\mathcal{A}^*$, which as poor distance and small-set flip decodability in the X-basis. Instead, for each input Q-codeblock, we first apply Proposition 5.1 followed by Proposition 6.1 to change one of the other factors $h' \neq h \in [r]$ from ${\mathcal{C}^{(h')}}^* = \mathcal{C}^G_*$ to $\mathcal{C}^{G,*}$, thereby bringing our logical qubits into the code associated to level 2 of our r-dimensional complex. We can then apply Proposition 5.1, the swap gates (see above), and Proposition 6.1 as described above to induce the desired logical permutation, where these applications switch the hth factor. Finally, we switch back to the original code Q with another more application of Proposition 5.1 followed by Proposition 6.1, where these final applications switch back the h'th factor to its original status. We then finish by applying the error correction gadget in Proposition 8.1 to each resulting Q-codeblock.

We will now show that the results referenced above imply that this resulting circuit $\mathcal{Q}$ yields a Pauli fault-tolerant gadget $((\mathcal{Q}, ((\mathcal{E}_{\text{run}}^{\sqcup \kappa})^{\sqcup u})^{\sqcup T}), D^{\sqcup \kappa}, D^{\sqcup \kappa})$ for $\bar{\mathcal{O}}$ using quantum space $N = O_r(\kappa \cdot n)$ and time $T = O_r(1)$, as desired.

The two layers of swap gates described above each by definition use space $\leq 2\kappa n$ and time 3, and cannot propagate errors between different physical qubits in $(A^{i-1} \times L^{G,1})^{\sqcup \kappa}$. Thus if the Pauli error on each A^{i-1}-block (of which there are $\kappa \cdot |L^{G,1}|$) within these qubits is $\mathcal{E}(G^{\mathcal{A}}_{[r-1]},\ \eta_{\text{in}},\ \gamma)$-avoiding just prior to performing the swaps, then the error is $\mathcal{E}(G^{\mathcal{A}}_{[r-1]},\ \eta_{\text{in}},\ \gamma + 12\gamma_{\text{run}})$-avoiding following the swaps. The factor of 12 here arises because the swaps are comprised of 6 layers of $CNOT$ gates, and in each of these 6 timesteps there are two $(A^{i-1} \times L^{G,1})^{\sqcup \kappa}$-blocks of physical qubits that could experience an error from the fault.

Combining the above reasoning with Proposition 5.1, Proposition 6.1, and Proposition 8.1 then yields the desired result. First, it follows immediately that $\mathcal{Q}$ uses quantum space $N = O_r(\kappa \cdot n)$ and time $T = O_r(1)$. Furthermore, by the fault-tolerance guarantees in these propositions and of the swap gates described above, every gadget invoked within $\mathcal{Q}$ prior to the error correction gadget will take as input a set of codeblocks, each of which is a Pauli $\mathcal{E}(G,\ \eta_{\text{in}},\ \gamma)$-deviation of the desired logical state at that timestep for some $\gamma = O_r(\gamma_{\text{in}})$. Here $G = G^{\mathcal{C}}_{[r]}$ or $G = G^{\mathcal{A}}_{[r-1]}$ for a C^i- or A^{i-1}-block of physical qubits, respectively. By the results listed above, each such gadget must then output a Pauli $\mathcal{E}(G,\ \eta_{\text{in}},\ \gamma')$-deviation of the desired logical state at the timestep following that gadget, for some $\gamma' = O_r(\gamma)$.

Thus the error correction gadget receives as input a set of codeblocks, each of which is a Pauli $\mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ O_r(\gamma_{\text{in}}))$-deviation of the desired output state. As long as $\gamma_{\text{in}}(r) > 0$ and $\zeta_{\text{run}}(r) > 0$ are sufficiently small constants depending only on r, then applying the assumption that $\gamma_{\text{run}} \leq \zeta_{\text{run}}(r) \cdot \gamma_{\text{in}}$ with Proposition 8.1 implies that the error correction gadget outputs a $\mathcal{E}_{\text{in}} = \mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ \gamma_{\text{in}})$-deviation of the desired output codeblock. □

For intuition, it is helpful to consider the $\kappa = 1$ case in Proposition 10.2. In this case, the logical qubits of Q are labeled by a single r-dimensional hypercube $(L^{G,1})^r$, and Proposition 10.2 provides a constant-overhead gadget for arbirarily permuting the slabs (i.e. axis-parallel codimension-1 hyperplanes) in any given direction $h \in [r]$.

The corollary below provides an example application of Proposition 10.2. Specifically, we show that by applying a cyclic permutation in each direction $h \in [r]$, we can induce a global cyclic permutation on a $\Theta(k)$-sized set of logical qubits.

Corollary 10.3. *Define $r, Q, n, k, i, \mathcal{C}^*, G_{\text{run}}, \mathcal{E}_{\text{in}} = \mathcal{E}_{\text{out}}, \mathcal{E}_{\text{run}}, D$ as in Proposition 10.2. Choose some relatively prime positive integers $\ell_1, \dots, \ell_r \le |L^{G,1}|$. Assume that we fix some ordering of the set $L^{G,1} \cong [|L^{G,1}|]$. For every $h \in [r]$, given some $s_h \in [\ell_h]$, let $\pi_h^{s_h} : L^{G,1} \to L^{G,1}$ be the permutation that cyclically rotates the first ℓ_h elements forwards by s_h positions, that is,*

$$\pi_h^{s_h}(j) = \begin{cases} j + s_h \pmod{\ell_h}, & j \in [\ell_h] \\ j, & j \in [|L^{G,1}|] \setminus [\ell_h]. \end{cases}$$

Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup u})^{\sqcup T}), D, D)$ for $\bar{\mathcal{O}}$, where:

- *$\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = O_r(n)$ and time $T = O_r(1)$, and $u = O(1)$.*
- *$\bar{\mathcal{O}} : \mathbb{C}^{\mathcal{M}^{\mathcal{C},i} \times \mathcal{M}^{\mathcal{C},i}} \to \mathbb{C}^{\mathcal{M}^{\mathcal{C},i} \times \mathcal{M}^{\mathcal{C},i}}$ is a channel that acts on k qubits labeled by the set $M^{\mathcal{C},i}$, that permutes the qubits in $(L^{G,1})^r \subseteq M^{\mathcal{C},i}$ according to $\pi_1^{s_1} \otimes \cdots \otimes \pi_r^{s_r}$ (so that the qubit at position $(b_1, \dots, b_r) \in (L^{G,1})^r$ is moved to $(\pi_1^{s_1}(b_1), \dots, \pi_r^{s_r}(b_r))$); the qubits in $M^{\mathcal{C},i} \setminus (L^{G,1})^r$ may be moved or reset arbitrarily.*

Proof. The desired circuit $\mathcal{Q}$ simply applies Proposition 10.2 r times, with the permutation $\pi_h^{s_h}$ for every $h \in [r]$. The analysis follows directly from Proposition 10.2. □

The action of the logical permutation $\pi_1^{s_1} \otimes \cdots \otimes \pi_r^{s_r}$ on the logical qubits in $[\ell_1] \times \cdots \times [\ell_r] \subseteq (L^{G,1})^r$ is by definition equivalent to the action of addition by $(s_1, \dots, s_r)$ on the abelian group $\mathbb{Z}_{\ell_1} \times \cdots \times \mathbb{Z}_{\ell_r}$. Because we chose $\ell_1, \dots, \ell_r$ to be relatively prime, this group is isomorphic to a the cyclic group $\mathbb{Z}_{\ell_1 \cdots \ell_r}$. Thus as we can choose relatively prime $\ell_1, \dots, \ell_r = \Theta(|L^{G,1}|) = \Theta(|V_L|)$, we have obtained a constant-overhead fault-tolerant gadget for performing arbitrary cyclic shifts on a linear number of logical qubits (in a constant-rate code Q).

An alternative cyclic permutation gadget was given by [XZZ+24, Section A.3]; however, this gadget relied on codes with a strong cyclic symmetry, for which it is unclear to what extend good asymptotic families can be found. In contrast, our gadget in Corollary 10.3 makes no symmetry assumptions on the code.

10.3 Parallel Logical Hadamard

The proposition below obtains a gadget to perform Hadamard gates in parallel on $\Theta(k)$ logical qubits, with the same input and output codes. For this purpose, we combine Lemma 9.3, which performs transversal Hadamard but with different input and output codes, with our code switching gadgets. Our gadget below uses constant space-time overhead, which significantly improves upon the $\Omega(\sqrt{k})$ time overhead in the Hadamard gadget of [XZZ+24, Section A.5]. In contrast to such prior works, our gadget also makes no symmetry assumptions on the underlying codes.

Proposition 10.4. *Define $r, Q, n, k, i, \mathcal{C}^*, G_{\text{run}}, \mathcal{E}_{\text{in}} = \mathcal{E}_{\text{out}}, \mathcal{E}_{\text{run}}, D$ as in Proposition 10.2. Then there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup u})^{\sqcup T}), D, D)$ for $\bar{\mathcal{O}}$, where:*

- *$\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = O_r(n)$ and time $T = O_r(1)$, and $u = O(1)$.*
- *$\bar{\mathcal{O}} : \mathbb{C}^{\mathcal{M}^{\mathcal{C},i} \times \mathcal{M}^{\mathcal{C},i}} \to \mathbb{C}^{\mathcal{M}^{\mathcal{C},i} \times \mathcal{M}^{\mathcal{C},i}}$ is a channel that acts on k qubits labeled by the set $M^{\mathcal{C},i}$, that performs a logical Hadamard gate on every qubit in $(L^{G,1})^r \subseteq M^{\mathcal{C},i}$; the qubits in $M^{\mathcal{C},i} \backslash (L^{G,1})^r$ may be moved or reset arbitrarily.*

Proof. The desired gadget $\mathcal{Q}$ first applies transversal Hadamard to the input codeblock using Lemma 9.3. By Lemma 9.3 and Remark 9.4, the input logical state with $H^{\otimes k}$ applied is now encoded in the code associated to level i of the *chain* complex $\mathcal{C}_* = \mathcal{C}_*^{(1)} \otimes \cdots \otimes \mathcal{C}_*^{(r)}$ with encoding map $\text{Enc}_*^{\mathcal{C}}$. By definition, this complex $\mathcal{C}_*$ is given by our original cochain complex $\mathcal{C}^* = \mathcal{C}^{(1)^*} \otimes \cdots \otimes \mathcal{C}^{(1)^*}$ with every factor of $\mathcal{C}^{G,*}$ replaced by $\mathcal{C}_*^G$ and vice versa. To return to the original code Q, we therefore loop through the directions $h \in [r]$ one at a time, and for each h we apply Proposition 5.1 to switch down to $|L^{G,1}|$ copies of a code associated to $\bigotimes_{h' \in [r] \backslash \{h\}} \mathcal{C}_0^{(h')^*}$ (where $\mathcal{C}_0^{(h')^*}$ denotes the current value of the factor in direction h'), and then we apply Proposition 6.1 to switch up to a code associated to $\bigotimes_{h' \in [r]} \mathcal{C}_1^{(h')^*}$, where $\mathcal{C}_1^{(h')^*} = \mathcal{C}^{(h')^*}$ if $h' = h$ and $\mathcal{C}_1^{(h')^*} = \mathcal{C}_0^{(h')^*}$ if $h' \neq h$. After all of these code switchings, we apply the error correction gadget in Proposition 8.1.

Similarly as in the proof of Proposition 10.2, the above procedure may fail if at some point (say at step $h \in [r]$) we end up with in the code at the bottom or top level of the cochain complex. However, this issue may be resolved exactly as in Proposition 10.2: we may temporarily switch one of the other factors $h' \neq h$ from $\mathcal{C}_0^{(h')^*}$ to ${\mathcal{C}_0^{(h')}}_*$, perform the desired switching on the hth factor, and then switch the h'th factor back to $\mathcal{C}_0^{(h')^*}$.

Analogously as in the proof of Proposition 10.2, the results listed above imply that $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup u})^{\sqcup T}), D, D)$ is a Pauli fault-tolerant gadget for $\bar{\mathcal{O}}$, as desired. □

10.4 Targeting of Individual Logical Qubits

In this section, for the code Q associated to some level $1 \leq i \leq r-1$ of $\mathcal{C}^* = \mathcal{C}^{(1)^*} \otimes \cdots \mathcal{C}^{(r)^*}$, we present a gadget for swapping an arbitrary pair of logical qubits across a pair of Q-codeblocks, while leaving the remaining logical qubits unchanged. This gadget requires just constant space-time overhead, and is based on repeated applications of our logical permutation gadget in Proposition 10.2.

By using this gadget to extract a desired logical qubit (or pair of logical qubits) into an ancilla codeblock, and then performing transversal (e.g. $CNOT$ or Hadamard) gates on the ancilla block, we can thereby perform constant-overhead targeted logical (e.g. $CNOT$ or Hadamard) gates.

Proposition 10.5. *Define $r, Q, n, k, i, \mathcal{C}^*, G_{\text{run}}, \mathcal{E}_{\text{in}} = \mathcal{E}_{\text{out}}, \mathcal{E}_{\text{run}}, D$ as in Proposition 10.2. Then for every $b, b' \in ((L^{G,1})^r)^{\sqcup 2}$, there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup u})^{\sqcup T}), D, D)$ for $\bar{\mathcal{O}}$, where:*

- *$\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = O_r(n)$ and time $T = O_r(1)$, and $u = O_r(1)$.*
- *$\bar{\mathcal{O}} : \mathbb{C}^{(\mathcal{M}^{\mathcal{C},i})^2 \times (\mathcal{M}^{\mathcal{C},i})^2} \to \mathbb{C}^{(\mathcal{M}^{\mathcal{C},i})^2 \times (\mathcal{M}^{\mathcal{C},i})^2}$ is a channel that acts on $2k$ qubits labeled by the set $(M^{\mathcal{C},i})^{\sqcup 2}$ (i.e. two blocks of $M^{\mathcal{C},i}$), that swaps the two qubits $b, b' \in \times((L^{G,1})^r)^{\sqcup 2}$, and leaves*

all other qubits in $((L^{G,1})^r)^{\sqcup 2}$ *unchanged; the qubits in* $(M^{\mathcal{C},i})^{\sqcup 2} \setminus ((L^{G,1})^r)^{\sqcup 2}$ *may be moved or reset arbitrarily.*

Proof. We first describe a subroutine (i.e. subgadget) that can swap the logical qubit b out into some fixed position $\tilde{b} \in (L^{G,1})^r$ of a fresh ancilla codeblock; we can then run the same subroutine to extract b' into another fresh ancilla codeblock, and then run the subroutine in reverse to insert each of these two logical qubits back into the other one's original location.

The desired subroutine was illustrated in Figure 1b in Section 1. It simply prepares r fresh ancilla Q-codeblocks using Proposition 7.1, and then performs $2r-1$ applications of Proposition 10.2 to swap slabs between these different codeblocks. Call the Q-codeblock containing $b = (b_1, \dots, b_h) \in (L^{G,1})^r$ codeblock 0, and label the ancilla Q-codeblocks with $1, \dots, r$. The first r swaps loop through $h = 1, \dots, r$, and apply Proposition 10.2 to swap the direction-h slab at position b_h in codeblock $h-1$ with the direction-h slab at position $\tilde{b}_h$ in codeblock h. The final $r-1$ swaps then loop back through $h = r-1, \dots, 1$, and apply Proposition 10.2 to again swap the direction-h slab at position b_h in codeblock $h-1$ with the direction-h slab at position $\tilde{b}_h$ in codeblock h.

By definition, following the swaps above, position $b \in (L^{G,1})^r$ of codeblock 0 will contain the logical qubit that was originally at position $\tilde{b} \in (L^{G,1})^r$ of codeblock r; all logical qubits in positions $(L^{G,1})^r \setminus \{b\}$ in codeblock 0 remain unchanged. Meanwhile, position $\tilde{b}$ of codeblock r will contain the logical qubit that was originally at position b of codeblock 0.

Thus we have a procedure to extract a single logical qubit from a codeblock. We can apply this procedure to both logical qubits b, b' (each with their own r ancilla codeblocks), and then swap the resulting codeblocks containing the extracted qubits. Subsequently, running the same extraction procedure but with codeblock r set to contain the extracted logical qubit (and fresh ancilla code states only for codeblocks $1, \dots, r-1$) has the effect of inserting the extracted qubits back into the original codeblocks. Hence we have given a procedure to insert logical qubit b into the position previously held by b', and vice versa. At the end of this entire procedure, we run the error correction gadget in Proposition 8.1.

The algorithm described above yields the desired gadget $\mathcal{Q}$. This gadget by definition runs $O_r(1)$ applications of the gadgets in Proposition 7.1, Proposition 10.2, and Proposition 8.1. Each such gadget uses quantum space $O_r(n)$ and time $O_r(1)$, so our entire quantum space and time usage is $O_r(n)$ and $O_r(1)$, respectively. These propositions also directly imply that $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup u})^{\sqcup T}), D, D)$ is a Pauli fault-tolerant gadget for $\bar{\mathcal{O}}$, by analogous reasoning as used in the proof of Proposition 10.2. □

As an immediate corollary to Proposition 10.2, we obtain gadgets that perform a targeted $CNOT$ or Hadamard gate using constant space-time overhead.

Corollary 10.6. *Define* $r, Q, n, k, i, \mathcal{C}^*, G_{\text{run}}, \mathcal{E}_{\text{in}} = \mathcal{E}_{\text{out}}, \mathcal{E}_{\text{run}}, D$ *as in Proposition 10.2. Then the following hold:*

1. *(Targeted inter-block* $CNOT$*) Given arbitrary* $b, b' \in ((L^{G,1})^r)^{\sqcup 2}$*, there exists a Pauli fault-tolerant gadget* $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup u})^{\sqcup T}), D^{\sqcup 2}, D^{\sqcup 2})$ *for* $\bar{\mathcal{O}}$*, where* $\bar{\mathcal{O}} : \mathbb{C}^{(\mathcal{M}^{\mathcal{C},i})^2 \times (\mathcal{M}^{\mathcal{C},i})^2} \to \mathbb{C}^{(\mathcal{M}^{\mathcal{C},i})^2 \times (\mathcal{M}^{\mathcal{C},i})^2}$ *is a channel that acts on* $2k$ *qubits labeled by the set* $(M^{\mathcal{C},i})^{\sqcup 2}$ *(i.e. two blocks of* $M^{\mathcal{C},i}$*), that performs a logical* $CNOT$ *gate on the two qubits* $b, b' \in ((L^{G,1})^r)^{\sqcup 2}$*, and leaves all other qubits inside* $((L^{G,1})^r)^{\sqcup 2}$ *unchanged; the qubits in* $(M^{\mathcal{C},i})^{\sqcup 2} \setminus ((L^{G,1})^r)^{\sqcup 2}$ *may be moved or reset arbitrarily.*

2. *(Targeted intra-block* $CNOT$*) Given arbitrary* $b, b' \in (L^{G,1})^r$*, there exists a Pauli fault-tolerant gadget* $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup u})^{\sqcup T}), D, D)$ *for* $\bar{\mathcal{O}}$*, where* $\bar{\mathcal{O}} : \mathbb{C}^{\mathcal{M}^{\mathcal{C},i} \times \mathcal{M}^{\mathcal{C},i}} \to \mathbb{C}^{\mathcal{M}^{\mathcal{C},i} \times \mathcal{M}^{\mathcal{C},i}}$ *is a channel that*

acts on k qubits labeled by the set $M^{\mathcal{C},i}$, that performs a logical CNOT gate on the two qubits $b, b' \in (L^{G,1})^r$, and leaves all other qubits inside $(L^{G,1})^r$ unchanged; the qubits in $(M^{\mathcal{C},i})^{\sqcup 2} \setminus ((L^{G,1})^r)^{\sqcup 2}$ may be moved or reset arbitrarily.

3. *(Targeted Hadamard) Given arbitrary $b \in (L^{G,1})^r$, there exists a Pauli fault-tolerant gadget $((\mathcal{Q}, (\mathcal{E}_{\text{run}}^{\sqcup u})^{\sqcup T}), D, D)$ for $\bar{\mathcal{O}}$, where $\bar{\mathcal{O}} : \mathbb{C}^{\mathcal{M}^{\mathcal{C},i} \times \mathcal{M}^{\mathcal{C},i}} \to \mathbb{C}^{\mathcal{M}^{\mathcal{C},i} \times \mathcal{M}^{\mathcal{C},i}}$ is a channel that acts on k qubits labeled by the set $M^{\mathcal{C},i}$, that performs a logical Hadamard gate on the qubit $b \in (L^{G,1})^r$, and leaves all other qubits inside $(L^{G,1})^r$ unchanged; the qubits in $(M^{\mathcal{C},i})^{\sqcup 2} \setminus ((L^{G,1})^r)^{\sqcup 2}$ may be moved or reset arbitrarily.*

In each case above, $\mathcal{Q}$ is an adaptive quantum circuit using quantum space $N = O_r(n)$ and time $T = O_r(1)$, and $u = O(1)$.

Proof. We consider the three statements separately:

1. The desired gadget $\mathcal{Q}$ performs the following:
   - Prepare two ancilla Q-codeblocks using Proposition 7.1.
   - Swap the two logical qubits b, b' from the input codeblocks into some fixed position $\tilde{b} \in (L^{G,1})^r$ of these two respective ancilla codeblocks using Proposition 10.5.
   - Apply transversal $CNOT$ between the two ancilla codeblocks using Lemma 9.1.
   - Swap the two logical qubits back into their original positions in the input codeblocks using Proposition 10.5.
   - Run the error correction gadget in Proposition 8.1.

   The fault-tolerance analysis follows directly from the results referenced above.
2. The proof is analogous to that for the targeted inter-block $CNOT$ in item 1 above, except that in the intra-block case, all four applications of Proposition 10.5 are applied to the unique input codeblock (along with the appropriate ancilla codeblock).
3. The desired gadget $\mathcal{Q}$ performs the following:
   - Prepare an ancilla Q-codeblock using Proposition 7.1.
   - Swap the logical qubit b from the input codeblock into some position $b \in (L^{G,1})^r$ of this ancilla codeblock using Proposition 10.5.
   - Apply logical $H^{\otimes (L^{G,1})^r}$ to this ancilla codeblock using Proposition 10.4.
   - Swap the logical qubit b back into its original position in the input codeblock using Proposition 10.5.
   - Run the error correction gadget in Proposition 8.1.

   The fault-tolerance analysis follows directly from the results referenced above.

□

Our gadget for targeted logical swap in Proposition 10.5 is related to the selective inter-block teleportation gadget in [XZZ+24, Section A.2]. Specifically, [XZZ+24] show how to swap the logical qubits at a given position across two codeblocks using a constant number of "logical cycles" (which corresponds to a constant-depth circuit in our setting where we have constant-depth state

preparation). They also extend this result to swap more such pairs of qubits, though with a higher time overhead.

Using a single logical swap, we are able to obtain the constant-overhead targeted $CNOT$ gate in Corollary 10.6 because Proposition 10.5 allows us to swap two logical qubits at *arbitrary* (possibly different) positions within the codeblock; in contrast, the techniques in [XZZ+24, Section A.2] most naturally apply to swapping logical qubits at the same position in different codeblocks, and hence seem to require additional properties such as cyclic code symmetries to obtain targeted $CNOT$ gates.

For demonstrative purposes, we have stated Proposition 10.5 and Corollary 10.6 for the case of performing a single logical swap, $CNOT$, or Hadamard gate. However, our gadgets described in Section 10.2 naturally perform highly parallel and flexible logical permutations, which can be used to efficiently perform many such gates in parallel. Such techniques will are particularly well suited for performing the same logical gate on many logical qubits within a given slab.

10.5 Constant-Overhead State Preparation of 2-Dimensional Codes in Bulk

In this section, we simply remark that combining our state preparation and downwards code switching gadgets yield a gadget for preparing a stack of 2-dimensional product codes with constant-space-time overhead.

Specifically, set $r = 3$, and consider applying Proposition 7.1 to construct either a $\left|+^k\right\rangle$ logical state (if $i = 1$) or a $\left|0^k\right\rangle$ logical state (if $i = 2$) in the $[[n, k]]$ code Q associated to level i of $\mathcal{C}^* = \mathcal{C}^{(1)^*} \otimes \mathcal{C}^{(2)^*} \otimes \mathcal{C}^{(3)^*}$ (as defined in Section 10.1). Then applying Proposition 5.1 to this state results in a stack of $|L^{G,1}| = \Theta(\sqrt{n'})$ copies of the $[[n', k']]$ code Q' associated to level 1 of $\mathcal{C}^{G,*} \otimes \mathcal{C}^G_*$, in logical state either $\left|+^{k'}\right\rangle$ or $\left|0^{k'}\right\rangle$. The propositions listed above imply the fault-tolerance of this procedure. Indeed, we implicitly use this procedure in our proof of Proposition 6.1 in Appendix C.

The code Q' is a 2-dimensional product code (often called a hypergraph product code [TZ14]). It is uncommon to have state preparation gadgets for such 2-dimensional codes with constant space-time overhead. Indeed, we are unaware of any such gadgets for a single such code state. It is therefore perhaps interesting that we are able to circumvent this challenge by simultaneously preparing $\Theta(k')$ states of Q' with constant overhead.

11 Acknowledgments

We thank Ting-Chun Lin for helpful discussions.

References

[BB24] Nikolas P. Breuckmann and Simon Burton. Fold-Transversal Clifford Gates for Quantum Codes. *Quantum*, 8:1372, June 2024. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[BDET24] Nikolas P. Breuckmann, Margarita Davydova, Jens N. Eberhardt, and Nathanan Tantivasadakarn. Cups and Gates I: Cohomology invariants and logical quantum operations, October 2024. arXiv:2410.16250.

[BE21] Nikolas P. Breuckmann and Jens N. Eberhardt. Balanced Product Quantum Codes. *IEEE Transactions on Information Theory*, 67(10):6653–6674, October 2021. Conference Name: IEEE Transactions on Information Theory.

[BH12] Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Physical Review A*, 86(5):052329, November 2012. Publisher: American Physical Society.

[BH14] Sergey Bravyi and Matthew B. Hastings. Homological product codes. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, STOC '14, pages 273–282, New York, NY, USA, May 2014. Association for Computing Machinery.

[BL25] Thiago Bergamaschi and Yunchao Liu. On fault tolerant single-shot logical state preparation and robust long-range entanglement. *LIPIcs, Volume 325, ITCS 2025*, 325:16:1–16:9, 2025. arXiv:2411.04405 [quant-ph].

[BMD07a] H. Bombin and M. A. Martin-Delgado. Exact topological quantum order in $D=3$ and beyond: Branyons and brane-net condensates. *Physical Review B*, 75(7):075103, February 2007. Publisher: American Physical Society.

[BMD07b] H. Bombin and M. A. Martin-Delgado. Topological Computation without Braiding. *Physical Review Letters*, 98(16):160502, April 2007. Publisher: American Physical Society.

[Bom15a] H. Bombin. Single-shot fault-tolerant quantum error correction. *Physical Review X*, 5(3):031043, September 2015. arXiv:1404.5504 [quant-ph].

[Bom15b] Héctor Bombín. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New Journal of Physics*, 17(8):083002, August 2015. Publisher: IOP Publishing.

[Bom16] H. Bombin. Dimensional Jump in Quantum Error Correction, May 2016. arXiv:1412.5079 [quant-ph].

[CHRY24] Andrew Cross, Zhiyang He, Patrick Rall, and Theodore Yoder. Improved QLDPC Surgery: Logical Measurements and Bridging Codes, November 2024. arXiv:2407.18393 [quant-ph].

[CKBB22] Lawrence Z. Cohen, Isaac H. Kim, Stephen D. Bartlett, and Benjamin J. Brown. Low-overhead fault-tolerant quantum computing using long-range connectivity. *Science Advances*, 8(20):eabn1717, May 2022. Publisher: American Association for the Advancement of Science.

[DHLV23] Irit Dinur, Min-Hsiu Hsieh, Ting-Chun Lin, and Thomas Vidick. Good Quantum LDPC Codes with Linear Time Decoders. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 905–918, New York, NY, USA, June 2023. Association for Computing Machinery.

[DLV24] Irit Dinur, Ting-Chun Lin, and Thomas Vidick. Expansion of higher-dimensional cubical complexes with application to quantum locally testable codes, April 2024. arXiv:2402.07476 [quant-ph].

[FGL18] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Efficient decoding of random errors for quantum expander codes. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 521–534, New York, NY, USA, June 2018. Association for Computing Machinery.

[FGL20] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Constant overhead quantum fault tolerance with quantum expander codes. *Communications of the ACM*, 64(1):106–114, December 2020.

[FH21] Michael Freedman and Matthew Hastings. Building manifolds from quantum codes. *Geometric and Functional Analysis*, 31(4):855–894, August 2021.

[GL25] Louis Golowich and Ting-Chun Lin. Quantum LDPC Codes with Transversal Non-Clifford Gates via Products of Algebraic Codes. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 689–696, New York, NY, USA, June 2025. Association for Computing Machinery.

[Got14] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead. *Quantum Information & Computation*, 14(15-16):1338–1372, November 2014.

[Has23] M. B. Hastings. On Quantum Weight Reduction, July 2023. arXiv:2102.10030 [quant-ph].

[HCWY25] Zhiyang He, Alexander Cowtan, Dominic J. Williamson, and Theodore J. Yoder. Extractors: QLDPC Architectures for Efficient Pauli-Based Computation, March 2025. arXiv:2503.10390 [quant-ph].

[HL22] Max Hopkins and Ting-Chun Lin. Explicit Lower Bounds Against Omega(n)-Rounds of Sum-of-Squares. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 662–673. IEEE Computer Society, October 2022.

[HLM+25] Jun-Ting Hsieh, Alexander Lubotzky, Sidhanth Mohanty, Assaf Reiner, and Rachel Yun Zhang. Explicit Lossless Vertex Expanders, April 2025. arXiv:2504.15087 [math].

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[HNP25] Zhiyang He, Quynh T. Nguyen, and Christopher A. Pattison. Composable Quantum Fault-Tolerance, August 2025. arXiv:2508.08246 [quant-ph].

[HVWZ25a] Zhiyang He, Vinod Vaikuntanathan, Adam Wills, and Rachel Yun Zhang. Asymptotically Good Quantum Codes with Addressable and Transversal Non-Clifford Gates, July 2025. arXiv:2507.05392 [quant-ph].

[HVWZ25b] Zhiyang He, Vinod Vaikuntanathan, Adam Wills, and Rachel Yun Zhang. Quantum Codes with Addressable and Transversal Non-Clifford Gates, February 2025. arXiv:2502.01864 [quant-ph].

[KP13] Alexey A. Kovalev and Leonid P. Pryadko. Fault tolerance of quantum low-density parity check codes with sublinear distance scaling. *Physical Review A*, 87(2):020304, February 2013. Publisher: American Physical Society.

[KP23] Gleb Kalachev and Pavel Panteleev. Two-sided Robustly Testable Codes, August 2023. arXiv:2206.09973 [cs, math].

[KP25] Gleb Kalachev and Pavel Panteleev. Maximally Extendable Product Codes are Good Coboundary Expanders, January 2025. arXiv:2501.01411 [cs].

[KYP15] Aleksander Kubica, Beni Yoshida, and Fernando Pastawski. Unfolding the color code. *New Journal of Physics*, 17(8):083026, August 2015. Publisher: IOP Publishing.

[LH22] Ting-Chun Lin and Min-Hsiu Hsieh. Good quantum LDPC codes with linear time decoder from lossless expanders. *arXiv:2203.03581 [quant-ph]*, March 2022. arXiv: 2203.03581.

[Lin24] Ting-Chun Lin. Transversal non-Clifford gates for quantum LDPC codes on sheaves, October 2024. arXiv:2410.14631 [quant-ph].

[LTZ15] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. Quantum Expander Codes. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 810–824, October 2015. arXiv:1504.00822 [quant-ph].

[LZ22] Anthony Leverrier and Gilles Zémor. Quantum Tanner codes. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 872–883. IEEE Computer Society, October 2022.

[MG92] J. Misra and David Gries. A constructive proof of Vizing's theorem. *Information Processing Letters*, 41(3):131–133, March 1992.

[NP25] Quynh T. Nguyen and Christopher A. Pattison. Quantum Fault Tolerance with Constant-Space and Logarithmic-Time Overheads. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 730–737, New York, NY, USA, June 2025. Association for Computing Machinery.

[Pat24] Christopher A. Pattison. Personal Communication, 2024.

[PK22] Pavel Panteleev and Gleb Kalachev. Asymptotically good Quantum and locally testable classical LDPC codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 375–388, New York, NY, USA, June 2022. Association for Computing Machinery.

[PKP23] Christopher A. Pattison, Anirudh Krishna, and John Preskill. Hierarchical memories: Simulating quantum LDPC codes with local gates, March 2023. arXiv:2303.04798 [quant-ph].

[QWV23] Armanda O. Quintavalle, Paul Webster, and Michael Vasmer. Partitioning qubits in hypergraph product codes to implement logical gates. *Quantum*, 7:1153, October 2023. Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.

[SJOY25] Esha Swaroop, Tomas Jochym-O'Connor, and Theodore J. Yoder. Universal adapters between quantum LDPC codes, March 2025. arXiv:2410.03628 [quant-ph].

[SS96] M. Sipser and D.A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, November 1996.

[THL+25] Shi Jie Samuel Tan, Yifan Hong, Ting-Chun Lin, Michael J. Gullans, and Min-Hsiu Hsieh. Single-shot universality in quantum ldpc codes via code switching. 2025.

[TZ14] Jean-Pierre Tillich and Gilles Zemor. Quantum LDPC codes with positive rate and minimum distance proportional to n^{1/2}. *IEEE Transactions on Information Theory*, 60(2):1193–1202, February 2014. arXiv: 0903.0566.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, December 2012.

[WY24] Dominic J. Williamson and Theodore J. Yoder. Low-overhead fault-tolerant quantum computation by gauging logical operators, October 2024. arXiv:2410.02213 [quant-ph].

[XZZ+24] Qian Xu, Hengyun Zhou, Guo Zheng, Dolev Bluvstein, J. Pablo Bonilla Ataides, Mikhail D. Lukin, and Liang Jiang. Fast and Parallelizable Logical Computation with Homological Product Codes, July 2024. arXiv:2407.18490 [quant-ph].

[Zhu25] Guanyu Zhu. A topological theory for qLDPC: non-Clifford gates and magic state fountain on homological product codes with constant rate and beyond the $N^{1/3}$ distance barrier, January 2025. arXiv:2501.19375 [quant-ph].

[ZSP+23] Guanyu Zhu, Shehryar Sikander, Elia Portnoy, Andrew W. Cross, and Benjamin J. Brown. Non-Clifford and parallelizable fault-tolerant logical gates on constant and almost-constant rate homological quantum LDPC codes via higher symmetries, October 2023. arXiv:2310.16982 [cond-mat, physics:hep-th, physics:quant-ph].

A Fault-Tolerance Proof for State Preparation

In this section, we provide the proof of Proposition 7.1. The general structure if this proof is similar to that of Proposition 5.1 given in Section 5.3.

Proof of Proposition 7.1. The desired circuit $\mathcal{Q}$ is given in Algorithm 4. By definition line 2 uses 1 timestep, line 3 uses $w+4$ timesteps (see Corollary 3.36), line 4 and line 5 together use 0 timesteps (as they simply perform a classical computation), and line 6 uses 1 timestep. Thus the entire circuit uses $T = w+6 = O(w)$ timesteps. Furthermore, line 3 uses $|C^i|+|C^{i+1}| \leq n+nw = n(1+w) = O(nw)$ physical qubits, and every other line uses a subset of these qubits, so the entire space usage is $N = O(nw)$.

Also note that the vertex set $V_{\text{run}} \supseteq V(G^{\mathcal{C}}_{i,i+1,i+2}) = C^i \sqcup C^{i+1} \sqcup C^{i+2}$ contains the set $C^i \sqcup C^{i+1}$ of physical qubits, so $\mathcal{E}_{\text{run}}$ is a well-defined family of bad sets for $\mathcal{Q}$.

Our goal is to show that for every $\mathcal{E}_{\text{run}}$-avoiding Pauli fault $\mathcal{F}$, the resulting output $\mathcal{Q}[\mathcal{F}](1)$ is a Pauli $\mathcal{E}_{\text{out}}$-deviation of $\text{Enc}(|+^k\rangle\langle+^k|)$. The reference system in Definition 3.25 has no effect on the proof, so we ignore it here for simplicity (see Remark 3.27).

For this purpose, we track all errors that arise at any point during the execution of the circuit $\mathcal{Q}$ using the graph G_{run}. By definition Algorithm 4 consists entirely of Clifford gates, and we have restricted attention to a Pauli fault $\mathcal{F}$. Therefore all of the unitary (i.e. non-measurement) Clifford gates simply propagate Pauli errors to (possibly) different Pauli errors.

Let $S_{\mathcal{F}} \subseteq C^i \sqcup C^{i+1}$ denote the set of all physical qubits that lie in (any timestep of) the forward-looking lightcone in $\mathcal{Q}$ (given by Algorithm 4) of any qubit in $\bigcup_{t\in[T]} \text{supp}(\mathcal{F}_t)$, where we take the lightcones starting from the start of the circuit (i.e. timestep 1). Note that this foward-looking

lightcone only counts quantum gates in line 2, line 3, and line 6, and does not count classical gates from line 4 or line 5. Let $S_{\text{syn}} \subseteq C^{i+1} \sqcup C^{i+2}$ denote the footprint (see Definition 4.3) of the call to `SSFlipSyn`$(\delta(s); i+1, \mathcal{C}^*)$ in line 4.

Furthermore, because errors remain Pauli when propagating through our Clifford circuit as described above, just prior to performing the Pauli-Z measurements in the syndrome extraction subroutine in line 3, our corrupted state equals

$$E_0 \left(\sum_{y \in \mathcal{C}^i} |y\rangle \otimes |\delta(y)\rangle \right) \left(\sum_{y \in \mathcal{C}^i} \langle y| \otimes \langle \delta(y)| \right) E_0'$$

which is the true state from the execution described in Lemma 7.2 with some Pauli error E_0, E_0' applied. This error E_0, E_0' is precisely determined by propagating the Pauli errors from $\mathcal{F}$ through the Clifford gates in the circuit, and $\text{supp}(E_0), \text{supp}(E_0') \subseteq S_{\mathcal{F}}$. Write $E_0 = Z^{(e_Z, f_Z)} X^{(e_X, f_X)}$ and $E_0' = X^{(e_X', f_X')} Z^{(e_Z', f_Z')}$ where $e_P, e_P' \in \mathcal{C}^i$ and $f_P, f_P' \in \mathcal{C}^{i+1}$ for $P \in \{X, Z\}$. Then we can rewrite the state above as

$$\sum_{s, s' \in \mathcal{C}^{i+1}} \sum_{y, y'} Z^{e_Z} X^{e_X} |y\rangle \left\langle y' \right| X^{e_X'} Z^{e_Z'} \otimes Z^{f_Z} |s\rangle \left\langle s' \right| Z^{f_Z'},$$

where the second sum is over all $y, y' \in \mathcal{C}^i$ for which $\delta(y) + f_X = s$ and $\delta(y') + f_X' = s'$. Then after performing the Pauli-Z measurements on the ancilla register in the call to `SyndExt`$(\rho; Z, \delta_i)$, the state collapses to only those terms in the sum with $s = s'$, which gives

$$\sum_{s \in \mathcal{C}^{i+1}} \sum_{y, y' \in \delta^{-1}(s)} Z^{e_Z} X^{e_X} |y + g(f_X)\rangle \left\langle y' + g'(f_X') \right| X^{e_X'} Z^{e_Z'} \otimes Z^{f_Z} |s\rangle \langle s| Z^{f_Z'}, \tag{32}$$

where $g = g(f_X)$, $g' = g'(f_X') \in \mathcal{C}^i$ are any fixed elements satisfying $\delta(g) = f_X$, $\delta(g') = f_X'$. Algorithm 4 then uses the syndrome s to compute an appropriate $x \in \mathcal{C}^i$, and returns the state above with the correction X^x applied (with possibly some additional Pauli errors supported inside $S_{\mathcal{F}}$ from the fault during this final Pauli correction).

To complete the proof of the proposition, we show Lemma A.1 below, which analyzes the output of Algorithm 4 resulting from the term associated to each possible syndrome in (32). Specifically, because $T \leq w + 6$ as shown below, Lemma A.1 implies that Algorithm 4 outputs a Pauli $\mathcal{E}_{\text{out}}$-deviation of $\text{Enc}(\left|+^k\right\rangle \left\langle +^k\right|)$, as desired.

Lemma A.1. *There exist Pauli errors E, E' whose supports are $\mathcal{E}(G_{\text{run}}, \eta_{\text{run}}, 100w^7 T 2^T \gamma_{\text{run}})$-avoiding such that when lines 4–7 of Algorithm 4 are applied to each term*

$$\sum_{y, y' \in \delta^{-1}(s)} Z^{e_Z} X^{e_X} |y + g\rangle \left\langle y' + g' \right| X^{e_X'} Z^{e_Z'} \otimes |s\rangle \langle s| \tag{33}$$

in the sum in (32),[20] *the output is of the form $E \,\text{Enc}(\left|+^k\right\rangle \left\langle +^k\right|) E'$.*

Proof. Fix some $\bar{y} \in \mathcal{C}^i$ with $\delta(\bar{y}) = s$, and let $b = \bar{y} + g$, $f = f_X$, $b' = \bar{y} + g'$, $f' = f_X'$. Then by definition, the measurement syndrome is

$$s = \delta(b) + f = \delta(b') + f'$$

[20] Note that $Z^{f_Z}, Z^{f_Z'}$ applied to $|s\rangle \langle s|$ simply induce a global phase on each such term.

where $\mathrm{supp}(f), \mathrm{supp}(f') \subseteq S_{\mathcal{F}} \cap C^{i+1}$. Letting $a^{i+1} \in \mathcal{C}^{i+1}$ be the variable defined in line 4 of Algorithm 4, we then define S_{err} and $\tilde{a}^i, \tilde{a}^{i+1}$ (resp. S'_{err} and $(\tilde{a}^i)', (\tilde{a}^{i+1})'$) to be the footprint and output of `SSFlipErr(`$f + a^{i+1}; i+1, \mathcal{C}^*$`)` (resp. `SSFlipErr(`$f' + a^{i+1}; i+1, \mathcal{C}^*$`)`), respectively. Note that by definition $f + a^{i+1} = s + a^{i+1} + \delta(b)$ (resp. $f' + a^{i+1} = s + a^{i+1} + \delta(b')$). We state and prove the following claims for the unprimed variables $b, f, \ldots$, but they analogously apply to the primed variables $b', f', \ldots$.

Claim A.2. *The set $S_{\mathcal{F}}$ is $\mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{run}},\ T2^T\gamma_{\mathrm{run}})$-avoiding.*

Proof. Assume for a contradiction that there exists some $V \subseteq V_{\mathrm{run}}$ of size $|V| \geq \eta_{\mathrm{run}}$ that induces a connected subgraph of G_{run} with $|V \cap S_{\mathcal{F}}|/|V| \geq T2^T\gamma_{\mathrm{run}}$. We can repeatedly add to V any point in $S_{\mathcal{F}} \setminus V$ that lies in the neighborhood of V until there are no more such points. The resulting set V can only have larger size $|V| \geq \eta_{\mathrm{run}}$ and $S_{\mathcal{F}}$-density $|V \cap S_{\mathcal{F}}|/|V| \geq 2^T\gamma_{\mathrm{run}}$, and contains all or none of every connected component of the subgraph of G_{run} induced by $S_{\mathcal{F}}$.

By definition, the circuit $\mathcal{Q}$ has T timesteps and consists of 1 and 2-qubit gates, where every 2-qubit gate acts on a pair of qubits connected by an edge in G_{run}. Therefore every point in $\bigcup_{t\in[T]} \mathrm{supp}(\mathcal{F}_t)$ has a forward-lightcone hitting $< 2^T$ qubits in $C^i \sqcup C^{i+1}$, which induce a conected subgraph of G_{run}. Thus

$$|V \cap S_{\mathcal{F}}| < \left| V \cap \bigcup_{t\in[T]} \mathrm{supp}(\mathcal{F}_t) \right| \cdot 2^T \leq T2^T\gamma_{\mathrm{run}}|V|,$$

which contradicts the assumption that $|V \cap S_{\mathcal{F}}|/|V| \geq T2^T\gamma_{\mathrm{run}}$, as desired. The second inequality above holds by the assumption that $\mathcal{F}$ is $\mathcal{E}_{\mathrm{run}}^{\sqcup T}$-avoiding, so that each $\mathcal{F}_t$ is $\mathcal{E}_{\mathrm{run}}$-avoiding. □

Claim A.3. *The set $S_{\mathcal{F}} \cup S_{\mathrm{syn}}$ is $\mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{run}},\ 5w^3T2^T\gamma_{\mathrm{run}})$-avoiding.*

Proof. Assume for a contradiction that there exists some $V \subseteq V_{\mathrm{run}}$ of size $|V| \geq \eta_{\mathrm{run}}$ that induces a connected subgraph of G_{run} with $|V \cap (S_{\mathcal{F}} \cup S_{\mathrm{syn}})| \geq 5w^3T2^T\gamma_{\mathrm{run}}$. As in the proof of Claim A.2, we may repeatedly add points in $(S_{\mathcal{F}} \cup S_{\mathrm{syn}}) \setminus V$ that lie in the neighborhood of V, in order to assume that V contains all or none of every connected component of the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup S_{\mathrm{syn}}$.

Recall as described above that the call `SSFlipSyn(`$\delta(s); i+1, \mathcal{C}^*$`)` in line 4 of Algorithm 4 has $s = \delta(b) + f$ with $\mathrm{supp}(f) \subseteq S_{\mathcal{F}} \cap C^{i+1}$. Then by Lemma 4.5, we have

$$\begin{aligned} |V \cap S_{\mathrm{syn}}| &\leq |V \cap \mathrm{supp}(f)|/\gamma_{\mathrm{syn}} \\ &\leq |V \cap S_{\mathcal{F}}| \cdot 4w^3 \\ &< T2^T\gamma_{\mathrm{run}}|V| \cdot 4w^3, \end{aligned}$$

where the third inequality above holds by Claim A.2. We then again apply Claim A.2 to conclude that

$$|V \cap (S_{\mathcal{F}} \cup S_{\mathrm{syn}})| \leq |V \cap S_{\mathcal{F}}| + |V \cap S_{\mathrm{syn}}| < (4w^3 + 1)T2^T\gamma_{\mathrm{run}}|V|,$$

which contradicts the assumption that $|V \cap (S_{\mathcal{F}} \cup S_{\mathrm{syn}})| \geq 5w^3T2^T\gamma_{\mathrm{run}}$, as desired. □

By definition, each update c^{i+1} that gets added into a^{i+1} during the a given step in the execution of `SSFlipSyn(`$\delta(s); i+1, \mathcal{C}^*$`)` has $\mathrm{supp}(c^{i+1})$ contained within the neighborhood in $G^{\mathcal{C}}_{i,i+1,i+2} \subseteq G_{\mathrm{run}}$ of the footprint from the previous step. As a consequence, if we restrict s to a given connected component V of the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup S_{\mathrm{syn}}$ (and zero out all values outside of

V), the output of $\texttt{SSFlipSyn}(\delta(s|_{V\cap C^{i+1}}); i+1, \mathcal{C}^*)$ must equal the restriction $a^{i+1}|_{V\cap C^{i+1}}$ of the output a^{i+1} of $\texttt{SSFlipSyn}(\delta(s); i+1, \mathcal{C}^*)$ (see Lemma 4.4).

Recall that by assumption $\eta_{\text{run}} \le m$. Also by definition, $\gamma_{\text{run}} \le 1/5w^3T2^T$, so it follows by Claim A.3 that the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup S_{\text{syn}}$ has no connected components containing $\ge \eta_{\text{run}}$ vertices. Therefore within every connected component V of this induced subgraph, by the assumption that $\mathcal{C}^*$ has a $(m, 0)$-small-set syndrome-flip decoder at level $i+1$, the while loop in the restricted execution $\texttt{SSFlipSyn}(\delta(s|_{V\cap C^{i+1}}); i+1, \mathcal{C}^*)$ will not terminate until the computed output $a^{i+1}|_{V\cap C^{i+1}}$ satisfies $\delta(s + a^{i+1}|_{V\cap C^{i+1}}) = 0$. Furthermore, by definition $s = \delta(b) + f$ with $\text{supp}(f) \subseteq S_{\mathcal{F}}$ and $\text{supp}(a^{i+1}) \subseteq S_{\text{syn}}$, and hence $s + a^{i+1} + \delta(b) = f + a^{i+1}$ has $\text{supp}(f + a^{i+1}) \subseteq S_{\mathcal{F}} \cup S_{\text{syn}}$.

Claim A.4. *The set $S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ is $\mathcal{E}(G_{\text{run}}, \eta_{\text{run}}, 50w^7T2^T\gamma_{\text{run}})$-avoiding.*

Proof. The proof is similar to that of Claim A.3. Assume for a contradiction that there exists some $V \subseteq V_{\text{run}}$ of size $|V| \ge \eta_{\text{run}}$ that induces a connected subgraph of G_{run} with $|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \ge 50w^7T2^T\gamma_{\text{run}}$. As in the proof of Claim A.3, we may repeatedly add points in $(S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}) \setminus V$ that lie in the neighborhood of V, in order to assume that V contains all or none of every connected component of the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$.

By Lemma 4.5, we have

$$\begin{aligned}
|V \cap S_{\text{err}}| &\le |V \cap \text{supp}(f + a^{i+1})|/\gamma_{\text{err}} \\
&\le |V \cap (S_{\mathcal{F}} \cup S_{\text{syn}})| \cdot 8w^4 \\
&< 5w^3T2^T\gamma_{\text{run}}|V| \cdot 8w^4,
\end{aligned}$$

where the third inequality above holds by Claim A.3. We then again apply Claim A.3 to conclude that

$$|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \le |V \cap (S_{\mathcal{F}} \cup S_{\text{syn}})| + |V \cap S_{\text{err}}| < (8w^4 + 1)5w^3T2^T\gamma_{\text{run}}|V|,$$

which contradicts the assumption that $|V \cap (S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \ge 50w^7T2^T\gamma_{\text{run}}$, as desired. □

Recall that we use $\tilde{a}^i, \tilde{a}^{i+1}$ to denote the variables that are updated throughout the execution of and then returned by $\texttt{SSFlipErr}(f + a^{i+1}; i+1, \mathcal{C}^*)$ (corresponding to a^{i-1}, a^i respectively in Algorithm 2). As shown above, we have

$$\delta(f + a^{i+1}) = \delta(s + a^{i+1} + \delta(b)) = \delta(s + a^{i+1}) = 0,$$

and hence the entire execution of $\texttt{SSFlipErr}(f + a^{i+1}; i+1, \mathcal{C}^*)$ will have $\tilde{a}^{i+1} = 0$; that is, every iteration of the while loop that does not fail must add some c^i in to $\tilde{a}^i$. By definition, every such c^i must have $\text{supp}(c^i)$ contained within the neighborhood in $G^{\mathcal{C}}_{i,i+1,i+2} \subseteq G_{\text{run}}$ of the footprint from the previous step. As a consequence, if we restrict $f + a^{i+1}$ to a connected component V if $S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ (and zero out all values outside of V), the output of $\texttt{SSFlipErr}(f + a^{i+1}|_{V\cap C^{i+1}}; i+1, \mathcal{C}^*)$ must equal the restriction $\tilde{a}^i|_{V\cap C^i}, \tilde{a}^{i+1}|_{V\cap C^{i+1}}$ of the output $\tilde{a}^i, \tilde{a}^{i+1}$ of $\texttt{SSFlipErr}(f + a^{i+1}; i+1, \mathcal{C}^*)$ (see Lemma 4.4).

Recall that by assumption $\eta_{\text{run}} \le m$. Also by definition, $\gamma_{\text{run}} \le 1/50w^7T2^T$, so it follows by Claim A.4 that the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ has no connected components containing $\ge \eta_{\text{run}}$ vertices. Therefore within every connected component V of this induced subgraph, by the assumption that $\mathcal{C}^*$ has a m-small-set error-flip decoder at level $i+1$, the while loop in the restricted execution $\texttt{SSFlipErr}(f + a^{i+1}|_{V\cap C^{i+1}}; i+1, \mathcal{C}^*)$ will not terminate until the

computed output $\tilde{a}^i|_{V\cap C^i}$ satisfies $\delta(\tilde{a}^i|_{V\cap C^i}) = f + a^{i+1}|_{V\cap C^{i+1}}$; recall from above that we will have $\tilde{a}^{i+1}|_{V\cap C^{i+1}} = 0$. It also follows that the output will always be such a valid $\tilde{a}^i|_{V\cap C^i}$, and never FAIL. Therefore we conclude that

$$\delta(\tilde{a}^i) = f + a^{i+1},$$

and by definition $\mathrm{supp}(\tilde{a}^i) \subseteq S_{\mathrm{err}}$.

It follows that $\delta(b) = s + f = s + a^{i+1} + \delta(\tilde{a}^i)$, so $b + \tilde{a}^i$ is a valid choice for $x \in \mathcal{C}^i$ in line 5 of Algorithm 4. Recall that the entire analysis above also applies to the primed variables $b', f', (\tilde{a}^i)'$, so that we also have $\delta(b') = s + f' = s + a^{i+1} + \delta((\tilde{a}^i)')$, and $b' + (\tilde{a}^i)'$ is also a valid choice for $x \in \mathcal{C}^i$.

Therefore assuming Algorithm 4 is in the state (33) in prior to line 4, it will compute $x = b + \tilde{a}^i + z = b' + (\tilde{a}^i)' + z'$ for some i-cocycles $z, z' \in Z^i(\mathcal{C})$, and then output the first register (i.e. qubits C^i) of

$$\begin{aligned}
&\tilde{E}\left(\sum_{y,y'\in\delta^{-1}(s)} |y+g+x\rangle\langle y'+g'+x| \otimes |s\rangle\langle s|\right)\tilde{E}' \\
&= \tilde{E}\left(\sum_{y,y'\in\delta^{-1}(s)} \left|(y+\bar{y}+z)+\tilde{a}^i\right\rangle\left\langle(y'+\bar{y}+z')+(\tilde{a}^i)'\right| \otimes |s\rangle\langle s|\right)\tilde{E}' \\
&= \tilde{E}\left(\sum_{y,y'\in Z^i(\mathcal{C})} X^{\tilde{a}^i}|y\rangle\langle y'| X^{(\tilde{a}^i)'} \otimes |s\rangle\langle s|\right)\tilde{E}' \\
&= \tilde{E}\left(X^{\tilde{a}^i}\,\mathrm{Enc}\left(\left|+^k\right\rangle\left\langle+^k\right|\right)X^{(\tilde{a}^i)'} \otimes |s\rangle\langle s|\right)\tilde{E}',
\end{aligned}$$

where $\tilde{E}, \tilde{E}'$ are simply the Pauli errors arising from propagating the errors from $\mathcal{F}$ through the unitary Clifford gates. Hence these errors do not depend on the measurement outcome s, and satisfy $\mathrm{supp}(\tilde{E}), \mathrm{supp}(\tilde{E}') \subseteq S_{\mathcal{F}}$. Also, by definition $\tilde{a}^i, (\tilde{a}^i)'$ only depend on f, f' and a^{i+1}, the latter of which only depends on $\delta(s) = \delta(f) = \delta(f')$. By definition $\mathrm{supp}(\tilde{a}^i) \subseteq S_{\mathrm{err}}$ and $\mathrm{supp}((\tilde{a}^i)') \subseteq S'_{\mathrm{err}}$, so we conclude that the output above differs from $\mathrm{Enc}(\left|+^k\right\rangle\left\langle+^k\right|)$ by a Pauli error that only depends on $\mathcal{F}$, and is supported inside

$$S_{\mathcal{F}} \cup S_{\mathrm{syn}} \cup S_{\mathrm{err}} \cup S'_{\mathrm{err}} = (S_{\mathcal{F}} \cup S_{\mathrm{syn}} \cup S_{\mathrm{err}}) \cup (S_{\mathcal{F}} \cup S_{\mathrm{syn}} \cup S'_{\mathrm{err}}),$$

which by Claim A.4 is $\mathcal{E}(G_{\mathrm{run}},\ \eta_{\mathrm{run}},\ 100w^7T2^T\gamma_{\mathrm{run}})$-avoiding, as desired. □

□

B Fault-Tolerance Proof for Error Correction

In this section, we present the proof of Proposition 8.1, which is somewhat similar in structure to the proofs of Propostion 5.1 and Proposition 7.1.

Proof of Proposition 8.1. The desired gadget $\mathcal{Q}$ is given in Algorithm 5. By definition line 2 and line 5 each use $w + 4$ timesteps (see Corollary 3.36), line 3 and line 6 each use 0 timesteps (as they perform classical computations), and line 4 and line 7 each use 1 timestep, for a total of $T = 2(w + 5) = 2w + 10 = O(w)$ timesteps. Furthermore, in addition to the $n = |C^i|$ physical

qubits, line 2 uses $|C^{i+1}|$ ancilla qubits and line 5 uses $|C^{i-1}$ ancilla qubits, for a total space usage of $N = |C^{i-1}| + |C^i| + |C^{i+1}| \le nw + n + nw \le 3nw = O(nw)$ physical qubits.

Also note that the vertex set $V_{\text{run}} \supseteq V(G^{\mathcal{C}}_{i,i+1,i+2}) = C^i \sqcup C^{i+1} \sqcup C^{i+2}$ contains the set $C^i \sqcup C^{i+1}$ of physical qubits, so $\mathcal{E}_{\text{run}}$ is a well-defined family of bad sets for $\mathcal{Q}$.

Our goal is to show that for every $\ell \in \mathbb{N}$, every $\rho \in \mathbb{C}^{2^{k+\ell} \times 2^{k+\ell}}$, every $\mathcal{E}_{\text{in}}$-deviation $\sigma = E_0 \operatorname{Enc}(\rho) E_0' \in \mathbb{C}^{2^{n+\ell} \times 2^{n+\ell}}$ of[21] $\operatorname{Enc}(\rho)$ (for Paulis E_0, E_0' such that $\operatorname{supp}(E_0) \cup \operatorname{supp}(E_0') \subseteq C^i$ is $\mathcal{E}_{\text{in}}$-avoiding), and every $\mathcal{E}_{\text{run}}^{\sqcup T}$-avoiding Pauli fault $\mathcal{F}$, the resulting output $\mathcal{Q}[\mathcal{F}](\sigma)$ is a Pauli $\mathcal{E}_{\text{out}}$-deviation of $\operatorname{Enc}(\rho)$.

Let $S_{\mathcal{F}} \subseteq C^{i-1} \sqcup C^i \sqcup C^{i+1}$ denote the set of all physical qubits that lie in (any timestep of) the forward-looking lightcone in $\mathcal{Q}$ (given by Algorithm 5) of any qubit in $\bigcup_{t \in [T]} \operatorname{supp}(\mathcal{F}_t)$, where we take each lightcone starting from timestep 1 for data qubits in C^i, or the timestep the qubit was initialized for ancilla qubits in C^{i-1} and C^{i+1}. Note that this foward-looking lightcone only counts quantum gates in line 2, line 4, line 5, and line 7, and does not count classical gates from line 3 and line 6.

Similarly, let $S_{\text{in}} \subseteq C^{i-1} \sqcup C^i \sqcup C^{i+1}$ denote the set of all physical qubits that lie in (any timestep of) the forward-looking lightcone in $\mathcal{Q}$ of any point in $\operatorname{supp}(E_0) \cup \operatorname{supp}(E_0')$.

The following claim follows from the same reasoning used to show Claim A.2 in Section 7, so we omit the proof to avoid redundancy.

Claim B.1. *The set $S_{\mathcal{F}}$ is $\mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ T2^T\gamma_{\text{run}})$-avoiding, and the set S_{in} is $\mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ 2^T\gamma_{\text{in}})$-avoiding.*

Lemma B.2 below analyzes the Z-correction in lines 2–4 of Algorithm 5; the analysis of the X-correction in lines 5–7 is then exactly analogous.

Lemma B.2. *Following the execution of line 4 in Algorithm 5, the joint state of the code register C^i and the ℓ-qubit ancilla system is proportional to $E_2 \operatorname{Enc}(\rho) E_2'$ for a Pauli error $E_2 = Z^{e_{2,Z}} X^{e_{2,X}}$, $E_2' = X^{e'_{2,X}} Z^{e_{2,Z'}}$ such that*

$$\operatorname{supp}(e_{2,X}) \cup \operatorname{supp}(e'_{2,X}) \subseteq C^i \quad \text{is} \quad \mathcal{E}\left(G_{\text{run}},\ \eta_{\text{run}},\ \frac{(w+1)T2^{T+4}\gamma_{\text{run}}}{\gamma_{\text{dec}}}\right)\text{-avoiding},$$

and

$$\operatorname{supp}(e_{2,Z}) \cup \operatorname{supp}(e_{2,Z}) \subseteq (S_{\text{in}} \cup S_{\mathcal{F}}) \cap C^i.$$

Proof. By the assumption that all on the input and in the fault $\mathcal{F}$ are Pauli errors, and all gates that we apply are Clifford. Therefore throughout the execution from the beginning of Algorithm 5 through the moment just prior to the Pauli Z measurements in the call to $\texttt{SSFlipSyn}(s_Z; i, \mathcal{C}^*)$ in line 2, the state differs from the noisless execution (with no input or fault errors) by a Pauli error. In this noiseless execution, all measurement outcomes are 0. Therefore in the noisy execution, the measurement outcome $s_Z \in \mathcal{C}^{i+1}$ is also deterministic, and the post-measurement state is either 0 (if one of the syndrome qubits has an X error from one side but not the other) or else is

$$Z^{e_{1,Z}} X^{e_{1,X}} \operatorname{Enc}(\rho) X^{e'_{1,X}} Z^{e'_{1,Z}} \tag{34}$$

[21] Similarly as described in Footnote 12, we abuse notation by letting Enc denote the isomorphism from $\mathbb{F}_2^k \xrightarrow{\sim} H^i(\mathcal{C})$, the associated encoding isometry, and the associated encoding channel. We also sometimes write Enc as a shorthand for $\operatorname{Enc} \otimes \mathcal{I}_\ell$. The meaning will always be made clear from the argument.

for some $e_{1,X}, e_{1,Z}, e'_{1,X}, e'_{1,Z} \in \mathcal{C}^i$ supported inside $S_{\text{in}} \cup S_{\mathcal{F}}$, such that[22]

$$\delta(e_{1,X}) + f_X = s_Z = \delta(e'_{1,X}) + f'_X \tag{35}$$

for some $f_X, f'_X \in \mathcal{C}^{i+1}$ supported inside $S_{\mathcal{F}}$. If the state collapses to 0, then the output is 0 and the lemma holds, so assume that the post-measurement state is of the form in (34).

Let $S_{\text{syn}} \subseteq C^i \sqcup C^{i+1}$ (resp. $S'_{\text{syn}} \subseteq C^i \sqcup C^{i+1}$) denote the footprint (see Definition 4.3) of the call to $\texttt{SSFlipSyn}(s_Z = \delta(e_{1,X}) + f_X;\ i,\ \mathcal{C}^*)$ (resp. $\texttt{SSFlipSyn}(s_Z = \delta(e'_{1,X}) + f'_X;\ i,\ \mathcal{C}^*)$) in line 3 of Algorithm 5. Note that we could have $S_{\text{syn}} \neq S'_{\text{syn}}$ even though by (35) holds because by Definition 4.3, the footprint can depend on the choice decomposition of s_Z into a syndrome plus a syndrome error. The claims we prove below for the unprimed variables $e_{1,X}, f_X, \ldots$ apply analogously to the primed variables $e'_{1,X}, f'_X, \ldots$.

Claim B.3. *The set $S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}}$ is $\mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ 5w^3T2^{T+1}\gamma_{\text{in}})$-avoiding.*

Proof. The proof is similar to that of Claim A.3. Assume for a contradiction that there exists some $V \subseteq V_{\text{run}}$ of size $|V| \geq \eta_{\text{in}}$ that induces a connected subgraph of G_{run} with $|V \cap (S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}})| \geq 5w^3T2^{T+1}\gamma_{\text{in}}$. As in the proof of Claim A.3, we may repeatedly add points in $(S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}}) \setminus V$ that lie in the neighborhood of V, in order to assume that V contains all or none of every connected component of the subgraph of G_{run} induced by $S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}}$.

Recall as described above that the call $\texttt{SSFlipSyn}(s_Z; i, \mathcal{C}^*)$ in line 3 of Algorithm 5 has $s_Z = \delta(e_{1,X}) + f_X$ with $\text{supp}(e_{1,X}) \subseteq S_{\text{in}} \cup S_{\mathcal{F}} \cap C^i$ and $\text{supp}(f_X) \subseteq S_{\text{in}} \cup S_{\mathcal{F}} \cap C^{i+1}$. Then by Lemma 4.5, we have

$$\begin{aligned}
|V \cap S_{\text{syn}}| &\leq |V \cap (\text{supp}(e_{1,X}) \cup \text{supp}(f))|/\gamma_{\text{syn}} \\
&\leq |V \cap (S_{\text{in}} \cup S_{\mathcal{F}})| \cdot 4w^3 \\
&< (2^T\gamma_{\text{in}} + T2^T\gamma_{\text{run}})|V| \cdot 4w^3 \\
&\leq T2^{T+1}\gamma_{\text{in}}|V| \cdot 4w^3.
\end{aligned}$$

where the third inequality above holds by Claim B.1 and because $\eta_{\text{run}} \leq \eta_{\text{in}}$, and the fourth inequality holds because $\gamma_{\text{run}} \leq \gamma_{\text{in}}$ (see (26). We then again apply Claim B.1 to conclude that

$$|V \cap (S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}})| \leq |V \cap (S_{\text{in}} \cup S_{\mathcal{F}})| + |V \cap S_{\text{syn}}| < (4w^3 + 1)T2^{T+1}\gamma_{\text{in}}|V|,$$

which contradicts the assumption that $|V \cap (S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}})| \geq 5w^3T2^{T+1}\gamma_{\text{in}}$, as desired. □

Similarly as in the proof of Proposition 7.1 and Proposition 5.1, each update c^i computed in $\texttt{SSFlipSyn}(s_Z; i, \mathcal{C}^*)$ is supported within the neighborhood in $G^{\mathcal{C}}_{i-1,i,i+1} \subseteq G_{\text{run}}$ of the footprint from the previous step. As a consequence, if we restrict s_Z to a given connected component V of the subgraph of G_{run} induced by $S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}}$ (and zero out all values outside of V), the output of $\texttt{SSFlipSyn}(\delta(s_Z|_{V \cap C^i}); i, \mathcal{C}^*)$ must equal the restriction $a^i_Z|_{V \cap C^i}$ of the output a^i_Z of $\texttt{SSFlipSyn}(s_Z; i+1, \mathcal{C}^*)$ (see Lemma 4.4).

Recall that by assumption $\eta_{\text{in}} \leq m$. Also by definition, $\gamma_{\text{in}} \leq 1/5w^3T2^{T+1}$, so it follows by Claim B.3 that the subgraph of G_{run} induced by $S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}}$ has no connected components containing $\geq \eta_{\text{in}}$ vertices.

Furthermore, by definition $\text{supp}(e_{1,X}) \subseteq S_{\text{in}} \cup S_{\mathcal{F}}$ and $\text{supp}(a^i_Z) \subseteq S_{\text{syn}}$, so $\text{supp}(e_{1,X} + a^i_Z) \subseteq S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}}$.

[22] Throughout this proof, we write $\delta = \delta^{\mathcal{C}}$.

Let S_{err} and $\tilde{a}_Z^{i-1}, \tilde{a}_Z^i$ (resp. S'_{err} and $(\tilde{a}_Z^{i-1})', (\tilde{a}_Z^i)'$) denote the footprint and output respectively of $\texttt{SSFlipErr}(e_{1,X} + a_Z^i\ i, \mathcal{C}^*)$ (resp. $\texttt{SSFlipErr}(e'_{1,X} + a_Z^i\ i, \mathcal{C}^*)$).

Claim B.4. *The set $S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ is $\mathcal{E}(G_{\text{run}},\ \eta_{\text{in}},\ 50w^7T2^{T+1}\gamma_{\text{in}})$-avoiding.*

Proof. The proof is similar to that of Claim A.4. Assume for a contradiction that there exists some $V \subseteq V_{\text{run}}$ of size $|V| \geq \eta_{\text{in}}$ that induces a connected subgraph of G_{run} with $|V \cap (S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \geq 50w^7T2^{T+1}\gamma_{\text{in}}$. As in the proof of Claim B.3, we may repeatedly add points in $(S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}) \setminus V$ that lie in the neighborhood of V, in order to assume that V contains all or none of every connected component of the subgraph of G_{run} induced by $S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$.

By Lemma 4.5, we have

$$\begin{aligned}|V \cap S_{\text{err}}| &\leq |V \cap \text{supp}(e_{1,X} + a_Z^i)|/\gamma_{\text{err}} \\ &\leq |V \cap (S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}})| \cdot 8w^4 \\ &< 5w^3T2^{T+1}\gamma_{\text{in}}|V| \cdot 8w^4,\end{aligned}$$

where the third inequality above holds by Claim B.3. We then again apply Claim B.3 to conclude that

$$|V \cap (S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \leq |V \cap (S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}})| + |V \cap S_{\text{err}}| < (8w^4 + 1)5w^3T2^{T+1}\gamma_{\text{in}}|V|,$$

which contradicts the assumption that $|V \cap (S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}})| \geq 50w^7T2^{T+1}\gamma_{\text{in}}$, as desired. □

Similarly as in the proof of Proposition 7.1 and Proposition 5.1, each update c^{i-1} and c^i computed in $\texttt{SSFlipErr}(e_{1,X} + a_Z^i\ i, \mathcal{C}^*)$ is supported within the neighborhood in $G^{\mathcal{C}}_{i-1,i,i+1} \subseteq G_{\text{run}}$ of the footprint from the previous step. As a consequence, if we restrict $e_{1,X} + a_Z^i$ to a given connected component V of the subgraph of G_{un} induced by $S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ (and zero out all values outside of V), the output of $\texttt{SSFlipErr}(e_{1,X} + a_Z^i|_{V \cap C^i}\ i, \mathcal{C}^*)$ must equal the restriction $\tilde{a}_Z^{i-1}|_{V \cap C^{i-1}}, \tilde{a}_Z^i|_{V \cap C^i}$ of the output $\tilde{a}_Z^{i-1}, \tilde{a}_Z^i$ of $\texttt{SSFlipErr}(e_{1,X} + a_Z^i\ i, \mathcal{C}^*)$ (see Lemma 4.4).

Recall that by assumption $\eta_{\text{in}} \leq m$. Also by assumption (26), $\gamma_{\text{in}} \leq 1/50w^7T2^{T+1}$, so it follows by Claim B.4 that the subgraph of G_{run} induced by $S_{\text{in}} \cup S_{\mathcal{F}} \cup S_{\text{syn}} \cup S_{\text{err}}$ has no connected components containing $\geq \eta_{\text{in}}$ vertices. Therefore within every connected component V of this induced subgraph, by the assumption that $\mathcal{C}^*$ has a m-small-set error-flip decoder at level i, the while loop in the restricted execution $\texttt{SSFlipErr}(e_{1,X} + a_Z^i|_{V \cap C^i}\ i, \mathcal{C}^*)$ will not terminate until the computed output $\tilde{a}_Z^{i-1}|_{V \cap C^{i-1}}, \tilde{a}_Z^i|_{V \cap C^i}$ satisfies

$$\delta(\tilde{a}_Z^{i-1}|_{V \cap C^{i-1}}) + (\tilde{a}_Z^i|_{V \cap C^i}) = e_{1,X} + a_Z^i|_{V \cap C^i}.$$

It also follows that the output will never be FAIL, and that

$$\delta(\tilde{a}_Z^{i-1}) + \tilde{a}_Z^i = e_{1,X} + a_Z^i, \tag{36}$$

where by definition $\text{supp}(\tilde{a}_Z^{i-1}), \delta(\text{supp}(\tilde{a}_Z^{i-1})), \text{supp}(\tilde{a}_Z^i) \subseteq S_{\text{err}}$.

Furthermore, because every update to $\tilde{a}_Z^i|_{V \cap C^i}$ in the execution of $\texttt{SSFlipErr}(e_{1,X} + a_Z^i|_{V \cap C^i}\ i, \mathcal{C}^*)$ increases $|\tilde{a}_Z^i|_{V \cap C^i}|$ by at most $|c^i| \leq w$ while decreasing $|\delta(e_{1,X} + a_Z^i + \tilde{a}_Z^i|_{V \cap C^i})|$ by at least 1, we must have

$$|\tilde{a}_Z^i|_{V \cap C^i}| \leq w|\delta(e_{1,X} + a_Z^i|_{V \cap C^i})| = w|\delta(\tilde{a}_Z^i|_{V \cap C^i})|. \tag{37}$$

Claim B.5. *The set $\tilde{S} := \text{supp}(\tilde{a}_Z^i) \cup \text{supp}(\delta(\tilde{a}_Z^i))$ is $\mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ \tilde{\gamma})$-avoiding for*

$$\tilde{\gamma} = \frac{(w+1)T2^{T+2}\gamma_{\text{run}}}{\gamma_{\text{dec}}}. \tag{38}$$

Proof. Assume for a contradiction that there exists some $V \subseteq V_{\text{run}}$ of size $|V| \geq \eta_{\text{run}}$ that induces a connected subgraph of G_{run} with $|V \cap \tilde{S}|/|V| \geq \tilde{\gamma}$. We may repeatedly add points in $(S_{\mathcal{F}} \cup \tilde{S}) \setminus V$ that lie in the neighborhood of V, in order to assume that V contains all or none of every connected component of the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup \tilde{S}$. As adding points in $S_{\mathcal{F}}$ cannot decrease the fraction of points in V that lie in $S_{\mathcal{F}} \cap \tilde{S}$, We also then must have

$$|V \cap (S_{\mathcal{F}} \cup \tilde{S})| \geq \tilde{\gamma}|V|.$$

However, by Claim B.1

$$|V \cap S_{\mathcal{F}}| \leq T2^T \gamma_{\text{run}}|V|, \tag{39}$$

so we must have

$$|V \cap \tilde{S}| \geq (\tilde{\gamma} - T2^T \gamma_{\text{run}})|V|. \tag{40}$$

Now because $\tilde{S} \subseteq S_{\text{err}}$ and $\gamma_{\text{in}} \leq 1/50w^7T2^{T+1}$, Claim B.4 implies that every connected component $V' \subseteq V_{\text{run}}$ of the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup \tilde{S} \subseteq S_{\mathcal{F}} \cup S_{\text{err}}$ contains at most $|V'| \leq \eta_{\text{in}} \leq m$ vertices. Therefore for every such V', we must have

$$|\delta(\tilde{a}^i_Z|_{V' \cap C^i})| \leq \frac{|f_X|_{V' \cap C^{i+1}}|}{\gamma_{\text{dec}}}. \tag{41}$$

Specifically, if (41) did not hold, then by the assumption that $\mathcal{C}^*$ has a (m, γ_{dec})-small-set syndrome-flip decoder at level i (see Definition 4.1), there would exist some $c^0 \in C^0$ and $c^i \in \mathcal{C}^i$ with $c^0 \preceq c^i$ such that

$$|\delta(c^i) + (\delta(\tilde{a}^i_Z) + f_X)|_{V' \cap C^{i+1}}| < |\delta(\tilde{a}^i_Z) + f_X|_{V' \cap C^{i+1}}|.$$

In order for the above inequality to hold, we must have $\text{supp}(\delta(c^i)) \cap \text{supp}(\delta(\tilde{a}^i_Z) + f_X|_{V' \cap C^{i+1}}) \neq \emptyset$, and hence c^0 lies above some element of $\text{supp}(\delta(\tilde{a}^i_Z) + f_X|_{V' \cap C^{i+1}})$, so every element of $\text{supp}(c^i)$ and $\text{supp}(\delta(c^i))$ lie inside the neighborhood in G_{run} of $\text{supp}(\delta(\tilde{a}^i_Z) + f_X|_{V' \cap C^{i+1}}) \subseteq V'$. But by definition no other connected components of the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup \tilde{S}$ intersect this neighborhood, and $\text{supp}(\tilde{a}^i_Z), \text{supp}(\delta(\tilde{a}^i_Z)) \subseteq \tilde{S}$ and $\text{supp}(f_X) \subseteq S_{\mathcal{F}}$, so it follows that

$$|\delta(c^i) + \delta(\tilde{a}^i_Z) + f_X| < |\delta(\tilde{a}^i_Z) + f_X|.$$

By (36) we have $\delta(\tilde{a}^i_Z) = \delta(e_{1,X} + a^i_Z)$, so the above inequality is equivalent to

$$|\delta(c^i) + \delta(e_{1,X} + a^i_Z) + f_X| < |\delta(e_{1,X} + a^i_Z) + f_X|.$$

But this inequality contradicts the assumption that $\texttt{SSFlipSyn}(s_Z = \delta(e_{1,X}) + f_X; i+1, \mathcal{C}^*)$ terminated with output a^i_Z, as c^i is a valid update to add in to a^i_Z that further reduces the sydrome weight. Thus (41) holds for every connected component $V' \subseteq V_{\text{run}}$ of the subgraph of G_{run} induced by $S_{\mathcal{F}} \cup \tilde{S}$.

By construction V contains all or none of every such connected component V', and $\tilde{a}^i_Z, \delta(\tilde{a}^i_Z), f_X$ are supported inside $S_{\mathcal{F}} \cup \tilde{S}$, so it follows that

$$|\delta(\tilde{a}^i_Z|_{V \cap C^i})| \leq \frac{|f_X|_{V \cap C^{i+1}}|}{\gamma_{\text{dec}}}.$$

However, because $\text{supp}(f_X) \subseteq S_{\mathcal{F}}$, by (39) we have

$$|f_X|_{V \cap C^{i+1}}| \leq |V \cap S_{\mathcal{F}}| \leq T2^T \gamma_{\text{run}}|V|.$$

Meanwhile, because by (37)

$$|V \cap \tilde{S}| = |\tilde{a}^i_Z|_{V\cap C^i}| + |\delta(\tilde{a}^i_Z|_{V\cap C^i})| \le (w+1)|\delta(\tilde{a}^i_Z|_{V\cap C^i})|,$$

by (40) we have

$$|\delta(\tilde{a}^i_Z|_{V\cap C^i})| \ge \frac{|V\cap\tilde{S}|}{w+1} \ge \frac{\tilde{\gamma} - T2^T\gamma_{\text{run}}}{w+1}|V|.$$

Combining the above inequalities, we conclude that

$$\frac{\tilde{\gamma} - T2^T\gamma_{\text{run}}}{w+1} \le \frac{T2^T\gamma_{\text{run}}}{\gamma_{\text{dec}}},$$

which contradicts the definition of $\tilde{\gamma}$ in (38), where here we use the fact that $\gamma_{\text{dec}} \le 1$ by Remark 4.2. Thus the assumption that there exists some $V \subseteq V_{\text{run}}$ of size $|V| \ge \eta_{\text{run}}$ that induces a connected subgraph of G_{run} with $|V\cap\tilde{S}|/|V| \ge \tilde{\gamma}$ was false, as desired. □

Recall that the entire analysis above was stated for the unprimed variables $e_{1,X}, f_X, \ldots$ but similarly applies to the primed variables $e'_{1,X}, f'_X, \ldots$, so that for instance (36) becomes

$$\delta((\tilde{a}^{i-1}_Z)') + (\tilde{a}^i_Z)' = e'_{1,X} + a^i_Z. \tag{42}$$

Then by (36) and (42), applying the correction $X^{a^i_Z}$ in line 4 of Algorithm 5 to the state in (34) yields the corrected state

$$Z^{e_{1,Z}}X^{e_{1,X}+a^i_Z}\,\text{Enc}(\rho)X^{e'_{1,X}+a^i_Z}Z^{e'_{1,Z}} = Z^{e_{1,Z}}X^{\tilde{a}^i_Z}\,\text{Enc}(\rho)X^{(\tilde{a}^i_Z)'}Z^{e'_{1,Z}},$$

where we use the fact that $X^{\delta(\tilde{a}^{i-1}_Z)}, X^{\delta((\tilde{a}^{i-1}_Z)')}$ are by definition stabilizers of our code Q and hence preserve the code state $\text{Enc}(\rho)$. Recall here that $\text{supp}(e_{1,Z}), \text{supp}(e_{1,Z'}) \subseteq (S_{\text{in}} \cup S_{\mathcal{F}}) \cap C^i$, while $\text{supp}(\tilde{a}^i_Z), \text{supp}(\tilde{a}^i_Z)'$ are both $\mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ \tilde{\gamma})$-avoiding by Claim B.5.

An additional Pauli error from $\mathcal{F}$, which is supported inside $S_{\mathcal{F}}$, will also occur at the timestep of line 4. Therefore the final state after line 4 will be

$$Z^{e_{2,Z}}X^{e_{2,X}}\,\text{Enc}(\rho)X^{e'_{2,X}}Z^{e'_{2,Z}},$$

where $\text{supp}(e_{2,Z})\cup\text{supp}(e'_{2,Z}) \subseteq (S_{\text{in}}\cup S_{\mathcal{F}})\cap C^i$, and $\text{supp}(e_{2,X})\cup\text{supp}(e'_{2,X}) \subseteq C^i$ is $\mathcal{E}(G_{\text{run}},\ \eta_{\text{run}},\ \gamma)$-avoiding for

$$\gamma = 2\tilde{\gamma} + T2^T\gamma_{\text{run}} \le \frac{(w+1)T2^{T+4}\gamma_{\text{run}}}{\gamma_{\text{dec}}},$$

as desired, where we have also applied Claim B.1 □

Applying exactly analogous reasoning as used to prove Lemma B.2 regarding the Z-correction in lines 2–4 of Algorithm 5, but instead to the X-correction in lines 5–7, yields the desired result. Specifically, because the only additional X errors after line 7 arise from $\mathcal{F}$, we conclude that the final output of Algorithm 5 is proportional to $E_3\,\text{Enc}(\rho)E'_3$ for a Pauli error $E_3 = Z^{e_{3,Z}}X^{e_{3,X}}$, $E'_3 = X^{e'_{3,X}}Z^{e_{3,Z'}}$ such that $\text{supp}(e_{3,X})\cup\text{supp}(e'_{3,X}) \subseteq C^i$ and $\text{supp}(e_{3,Z})\cup\text{supp}(e'_{3,Z}) \subseteq C^i$ are both

$$\mathcal{E}\left(G_{\text{run}},\ \eta_{\text{run}},\ \frac{(w+1)T2^{T+5}\gamma_{\text{run}}}{\gamma_{\text{dec}}}\right)\text{-avoiding},$$

so the overall error support $\text{supp}(E_3)\cup\text{supp}(E'_3)$ is

$$\mathcal{E}\left(G_{\text{run}},\ \eta_{\text{run}},\ \frac{(w+1)T2^{T+6}\gamma_{\text{run}}}{\gamma_{\text{dec}}}\right)\text{-avoiding},$$

as desired. □

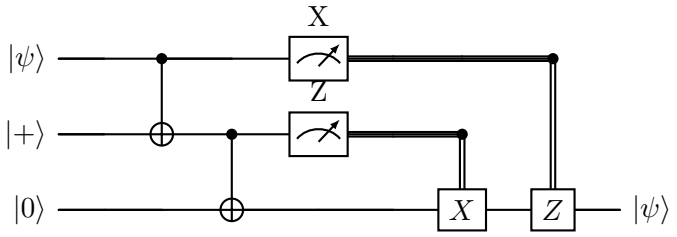


Figure 2: Standard circuit for teleporting a qubit $|\psi\rangle$.

C Fault-Tolerance Proof for Upwards Code Switching

In this section, we present the proof of Proposition 6.1. As described in Section 6, this proof simply combines gadgets presented previously in the paper.

Proof of Proposition 6.1. We construct the gadget $\mathcal{Q}$ by implementing the standard teleportation gadget in Figure 2 using the gadgets previously presented in this paper. Specifically, Figure 2 shows the circuit for teleporting a single qubit. We implement this circuit on k_{in} logical qubits simultaneously, where the first wire corresponds to the input code state in Q_{in}, the second wire corresponds to the state $\text{Enc}_{\text{in}}(|+\rangle\langle+|^{\otimes k_{\text{in}}})$ the we prepare, and the third wire corresponds to the state $\text{Enc}_{\text{out}}(|0\rangle\langle 0|^{\otimes k_{\text{out}}})$ that we also prepare. The extra $k_{\text{out}} - k_{\text{in}}$ logical $|0\rangle$ qubits in this third codeblock are precisely the logical $|0\rangle$ qubits that $\bar{\mathcal{O}}$ pads onto the input.

Specifically, we prepare $\text{Enc}_{\text{out}}(|0\rangle\langle 0|^{\otimes k_{\text{out}}})$ directly using the gadget in Proposition 7.1, though where we dualize to the chain complex $\mathcal{C}_*$ and exchange the roles of the X and Z bases (see Remark 7.3). Here we rely on the $(m, 0)$-small-set flip decodability of $\mathcal{C}_*$ at level $i-1$.

We are unable to similarly prepare $\text{Enc}_{\text{in}}(|+\rangle\langle+|^{\otimes k_{\text{in}}})$ by directly applying Proposition 7.1, as we do not assume the necessary small-set flip decoders for this lower-dimensional code (see Remark C.1). Instead, letting $k_{\mathcal{D}} = \dim(H^{i-1}(\mathcal{D}))$, we first apply Proposition 7.1 to prepare $\text{Enc}^{\mathcal{D},i-1}(|0\rangle\langle 0|^{\otimes k_{\mathcal{D}}})$. Here we use the $(m, 0)$-small-set flip decodability of $\mathcal{D}^*$ at level i. We then apply Proposition 5.1 to map this r-dimensional code state state down to the $(r-1)$-dimensional code state $\text{Enc}_{\text{in}}(|0\rangle\langle 0|^{\otimes k_{\text{in}}}) = (\text{Enc}^{\mathcal{A}})^{\sqcup L^{\mathcal{B},1}}(|0\rangle\langle 0|^{\otimes k_{\text{in}}})$. Here we use the $(m, 0)$-small-set flip decodability of $\mathcal{D}_{\bar{L},*}$ at level $i-1$. Also note that in this application of Proposition 5.1, we have dualized to a chain complex, and exchanged the roles of the X and Z bases (see Remark 5.2).

Once we have prepared these logical ancilla states, we then perform the two logical $CNOT$ gates in Figure 2 using Lemma 9.1 and Lemma 9.2 respectively. We then perform the logical Pauli measurements using Lemma 9.5, and the Pauli corrections by simply applying the appropriate logical Pauli operators (in a depth-1 Pauli circuit; the fault-tolerance analysis of this step is straightforward).

The desired result then follows directly from the results listed above proving fault-tolerance of our gadgets; Proposition 3.28 implies that the fault-tolerance holds even when each gadget in the construction only acts on a subset of the physical qubits in the overall circuit. □

Remark C.1. *When $r = 3$, then $(r-1) = 2$-dimensional codes typically will not support the single-shot state preparation gadget given in Section 7. Hence as described in the proof of Proposition 6.1 above, we prepare the logical $|+\rangle$ state in Figure 2 using the downwards code switching gadget from Section 5.*

If we consider codes of one dimension higher, and only consider $r \geq 4$ and $2 \leq i \leq r-2$, then

we are able to avoid using the downwards code switching gadget, and instead directly prepare the logical $|+\rangle$ states using Proposition 7.1. In fact, this approach also allows us to perform downwards code switching without Proposition 5.1 (using a similar construction as in Proposition 6.1 with state preparation via Proposition 7.1). This approach furthermore allows us to instantiate these gadgets with codes of rate arbitrarily close to 1, as opposed to just some small constant rate, because we would no longer need small-set flip decodability of the restricted complex $\mathcal{D}_{\bar{L},*}$ (recall that $R > 0$ is just a small constant in Corollary 3.10).

To avoid redundancy in our presentation, we primarily state results that apply to arbitrary $r \geq 3$; the extensions described above for the $r \geq 4$ case are then straightforward adaptations of our arguments.

D Proposed Method for Universal Computation via Transversal Non-Clifford Gates

In the main text above, we focused on constant-overhead gadgets for various addressable and parallel Clifford gates. To achieve universal quantum computation, it suffices to add a gadget for performing a non-Clifford gate, such as CCZ or T. Standard approaches include magic state distillation (see e.g. [BH12]), or switching into a code that natively supports transversal (i.e. constant-depth fault-tolerant) non-Clifford gates. As this latter approach seems well-suited for our techniques, in this appendix we discuss the possibility of applying our code switching to codes with transversal non-Clifford gates. Such a protocol could be viewed as an extension of the color/surface-code-based protocol of [Bom16] to qLDPC codes with better parameters. As explained below, here we simply propose a high-level approach without proving fault-tolerance. The independent and concurrent work of [THL+25] provides a more rigorous treatment for a specific family of codes based on those of [Zhu25].

We emphasize that while this section describes one proposal for universal fault-tolerant computation, our gadgets in the main text can already be applied to reduce the space-time overhead of specific components of existing schemes for universal fault-tolerant computation.

QLDPC codes with transversal non-Clifford (in particular, CCZ) gates often arise from 3-dimensional tensor products of classical LDPC codes (e.g. [BMD07b, GL25, Zhu25])[23]. If our code-switching-based gadgets for Clifford gates could be applied to these codes, then the transversal CCZ gate would provide a constant-overhead gadget for CCZ gates, yielding a complete scheme for universal fault-tolerant quantum computation. In particular, the constructions of [GL25] and [Zhu25] are able to perform many (specifically, $\Theta(n^{1-\epsilon})$ and $\Theta(n^{1/3})$ respectively; see Section D.1 below) logical CCZ gates in a codeblock using a constant-depth physical circuit. Hence a code-switching-based scheme using these codes may yield improvements over distillation-based schemes, which often only perform a single logical non-Clifford gate in a codeblock in any given gadget (see e.g. [NP25]).

However, the construction of qLDPC codes with transversal non-Clifford gates has proven to be a challenging problem, and the constructions of [GL25, Zhu25] still have far from optimal parameters. It therefore remains an interesting direction of future work to determine if the overall space-time overhead of a fault-tolerance scheme based on these codes would improve upon existing schemes such as in [NP25].

Furthermore, our proofs of a fault-tolerance threshold rely on our small-set flip decoder in

[23]The codes of [GL25] have polylogarithmic rather than constant locality (i.e. check weight), so they may be considered "nearly LDPC."

Section 4. We only construct such a decoder for quantum codes arising as products of classical codes associated to lossless expanders. Yet known product constructions of qLDPC codes with transversal non-Clifford gates (see above) require (at least some of) the classical factor codes to have structure that is incompatible with lossless expansion. Hence our fault-tolerance proofs do not immediately apply to such codes.

Below, we provide some more details on the codes of [GL25, Zhu25], which exhibit some of the best known parameters for qLDPC codes with transversal non-Clifford gates. We highlight interesting directions for future work regarding applying our code switching techniques to these codes.

D.1 Codes with Transversal Non-Clifford Gates

We now provide more details on qLPDC codes with transversal non-Clifford (specifically CCZ) gates, to provide context for the question of performing code switching on such codes. In this section, for convenience we will use the non-standard notation that a quantum code is $[[n, k, d]]_{CCZ}$ if it supports a constant-depth physical circuit that induces logical CCZ gates on k disjoint triples of logical qubits.

For many years, the state-of-the-art qLDPC codes with transversal CCZ were given by the $[[n, \Theta(1), \Theta(n^{1/3})]]_{CCZ}$ 3-dimensional color/surface code [BMD07b] (see also [BMD07a, Bom15b, KYP15]), which can be viewed as a tensor product of three chain complexes associated to classical repetition codes.

Recently, for arbitrarily small constant $\epsilon > 0$, [GL25] obtained a $[[n, \Theta(n^{1-\epsilon}), \tilde{\Theta}(n^{1/3})]]_{CCZ}$ construction that is nearly LDPC, meaning that the locality is polylogarithmic instead of constant. This almost-linear code dimension $k = \Theta(n^{1-\epsilon})$ was achieved by taking a tensor product of three chain complexes associated to classical algebraic codes, which can be viewed as Reed-Muller codes with sparsified parity-check matrices.

Subsequently, [Zhu25] gave constructions of $[[n, \Theta(n^{1/3}), \Theta(n^{1/3})]]_{CCZ}$ and $[[n, \Theta(n^{1/2}), \Theta(n^{1/2})]]_{CCZ}$ qLDPC codes. These codes are obtained by "thickening" 3-dimensional product codes, meaning that the product code is mapped to a manifold or higher-dimensional CW complex using a modified version of the mapping from [FH21]. The underlying classical codes used in the product consist of both classical repetition codes and good classical LDPC codes (based on expander graphs, e.g. [SS96]).

Both the construction of [GL25] and the $\Theta(n^{1/3})$-distance construction of [Zhu25] are based on tensor products of classical LDPC codes, and hence may be amenable to similar code switching techniques as we develop. The $\Theta(n^{1/2})$-distance construction of [Zhu25] uses the related but more involved lifted/balanced product [BE21, PK22, LZ22, DHLV23, LH22]. However, as the input codes in these constructions are not based on lossless expanders (see above), our fault-tolerance proofs do not directly apply. It is an interesting direction for future work to see if a threshold can still be proven for analogues of our code switching gadgets on these codes.

A positive resolution to this question would immediately yield a new scheme for universal fault-tolerant quantum computation. By the code parameters listed above, this scheme would be capable of performing polynomially many logical CCZ gates in a codeblock using a constant-depth physical circuit. Specifically, the codes of [GL25] can perform $n^{1-\epsilon}$ such CCZ gates in parallel for arbitrarily small constant $\epsilon > 0$, while the $\Theta(n^{1/3})$-distance codes of [Zhu25] can perform $\Theta(n^{1/3})$ such CCZ gates in parallel. Note that while the polylogarithmic locality of the codes of [GL25] may present an apparent barrier against establishing a fault-tolerance threshold, this issue may be resolved by concatenating with a small code such as a polylogarithmic-sized surface code (see e.g. [PKP23]).

We emphasize again that our gadgets for Clifford gates in the main text already improve the efficiency of specific components of existing fault-tolerance schemes. If single-shot code switching for the qLDPC codes with CCZ described above could be established, it would be an interesting direction to compare the overall scheme's space-time overhead to that of other methods for performing non-Clifford gates.