Robust sparse IQP sampling in constant depth

Louis Paletta[1], Anthony Leverrier[2], Alain Sarlette[1,3], Mazyar Mirrahimi[1], and Christophe Vuillot[4]

[1]Laboratoire de Physique de l'Ecole normale supérieure, ENS-PSL, CNRS, Inria, Centre Automatique et Systèmes (CAS), Mines Paris, Université PSL, Sorbonne Université, Université Paris Cité, Paris, France

[2]Inria Paris, France

[3]Department of Electronics and Information Systems, Ghent University, Belgium

[4]Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Between NISQ (noisy intermediate scale quantum) approaches without any proof of robust quantum advantage and fully fault-tolerant quantum computation, we propose a scheme to achieve a provable superpolynomial quantum advantage (under some widely accepted complexity conjectures) that is robust to noise with minimal error correction requirements. We choose a class of sampling problems with commuting gates known as sparse IQP (Instantaneous Quantum Polynomial-time) circuits and we ensure its fault-tolerant implementation by introducing the tetrahelix code. This new code is obtained by merging several tetrahedral codes (3D color codes) and has the following properties: each sparse IQP gate admits a transversal implementation, and the depth of the logical circuit can be traded for its width. Combining those, we obtain a depth-1 implementation of any sparse IQP circuit up to the preparation of encoded states. This comes at the cost of a space overhead which is only polylogarithmic in the width of the original circuit. We furthermore show that the state preparation can also be performed in constant depth with a single step of feed-forward from classical computation. Our construction thus exhibits a robust superpolynomial quantum advantage for a sampling problem implemented on a constant depth circuit with a single round of measurement and feed-forward.

1 Introduction

Recent progress on quantum hardware suggests that quantum processors will soon be able to outperform classical devices for some specific tasks. In the absence of fault-tolerant quantum computers, sampling problems [1, 2] appear to be a promising avenue to demonstrate such a quantum advantage since they can be solved with reasonably small circuits. In sampling problems, given some family $\mathfrak{C}$ of quantum circuits on N quantum registers, the goal is to sample from the output distribution p_C for any circuit $C \in \mathfrak{C}$. Well-known examples of circuit families include linear optical circuits in the case of BosonSampling [3], random quantum circuits [4] and Instantaneous Quantum Polynomial-time (IQP) circuits [5]. The original idea behind these proposals was that quantum processors can in principle sample from the corresponding distributions, while it is widely believed that classical computers cannot complete the same task efficiently. The caveat, however, is that current quantum processors are not equipped with fault-tolerance, and will instead output noisy samples, thus only solving a noisy version of the initial sampling problem. Unfortunately, the evidence for the classical hardness of this problem is thinner, and recent works have cast some serious doubts on the possibility of demonstrating a quantum advantage with this approach [6, 7, 8, 9, 10].

A potential strategy to address the issue of noise is to focus on problems for which it is possible to add some level of fault-tolerance, in an intermediate manner between Noisy Intermediate-Scale Quantum (NISQ) processors available in the near term [11] and universal fault-tolerant quantum computation. We list potential approaches to such robust quantum advantage in Table 1. The fact that IQP circuits are a non-universal class of circuits make them a good candidate in this respect since they are easier to make fault-tolerant. In particular, they can bypass the limitations of the Eastin-Knill theorem which states that a universal gate set cannot be implemented with transversal gates [12]. In this work, we show how to perform a fault-tolerant version of (sparse) IQP sampling with a constant-depth quantum circuit and with a space overhead that is only polylogarithmic in the width

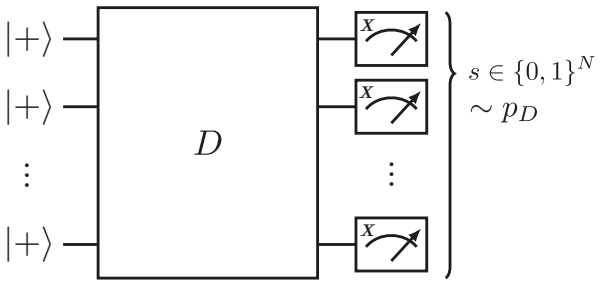

Figure 1: IQP circuits on N qubits are defined by an unitary D diagonal in the computational basis with state preparation and measurements performed in the Hadamard basis. For sparse IQP circuits, D is a logarithmic-depth circuit consisting of T and CS-gates.

of the original circuit. We note that [13] addressed a similar question for a different sampling problem, but the constant depth was obtained at the price of a polynomial overhead in terms of qubits because of differences in the initial computational problem and of the magic state distillation protocol necessary to its fault-tolerant implementation. In addition, it neglects some polynomial-time classical computation necessary for the error correction but during which errors can accumulate, while we bring down the complexity of error correction to polylogarithmic-time, making it less of an issue for future implementations. We note that similar computation times, for correcting a surface code of logarithmic size for instance, are often neglected in the literature.

1.1 Sparse IQP

An IQP circuit on N qubits takes a very simple form (see Figure 1): one applies an N-qubit gate D, diagonal in the computational basis, to an initial state $|+\rangle^{\otimes N}$ and measures the resulting state in the $\{|+\rangle, |-\rangle\}$ basis [16, 17, 5]. Here, we will focus on the sparse variant of IQP circuits introduced in [18]. In this variant, the circuit D is generated randomly from logarithmic-depth circuits with gate set $\{T, CS\}$:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}, \quad CS = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}. \quad (1)$$

More precisely, such a circuit on N qubits is generated in the following way:

- a single-qubit gate T^k is applied to every qubit, with $k \in \{0, \dots, 7\}$ chosen uniformly and independently for every qubit,
- for every pair of qubits, a gate CS^k with $k \in \{0, \dots, 3\}$ chosen uniformly at random, is applied with probability $\gamma \log N/N$, for some fixed parameter $\gamma > 0$.

Let us denote by $\mathfrak{D}_N$ the family of IQP circuits generated from the gate set $\{T, CS\}$. We associate to each circuit of $\mathfrak{D}_N$ its probability of being generated by the previous random process to define a distribution over $\mathfrak{D}_N$. We call an IQP circuit picked from this distribution sparse and in the following whenever we discuss about a fraction of sparse IQP circuits we mean a fraction of circuit in the sense of the probability distribution defined above. We note that all the considered gates commute, and can therefore be applied in any order. Given that each qubit will typically be involved in a logarithmic number of 2-qubit gates, we see that sparse IQP circuits can be implemented by circuits of average depth $\Theta(\log N)$ [18]. For each sparse IQP circuit D, we denote by p_D the probability distribution on $\{0,1\}^N$ corresponding to the output distribution of the circuit. In particular, it holds that

$$p_D(0^N) = \sum_{z \in \{0,1\}^N} e^{i\pi/8\left(\sum_{i<j} w_{i,j} z_i z_j + \sum_{k=1}^N v_k z_k\right)}, \quad (2)$$

for some integer weights $w_{i,j}, v_k$. This quantity corresponds to an Ising model partition function, which is proven to be hard to compute in the worst case [19, 20] and conjectured to be hard to compute on average. We formally recall the conjecture from [18]:

Conjecture 1 (Average Case Hardness of Ising model [18])**.** *Consider the partition function of the general Ising model,*

$$Z(\omega) = \sum_{z \in \{\pm 1\}^N} \omega^{\sum_{i<j} w_{i,j} z_i z_j + \sum_{k=1}^N v_k z_k}, \quad (3)$$

where the exponential sum is over the complete graph on N vertices, $w_{i,j} \in \mathbb{R}$ and $v_k \in \mathbb{R}$ are weights for edge ij and vertex k, and $\omega \in \mathbb{C}$.

If the weights are chosen uniformly at random from the set $\{0, \dots, 7\}$, then it is #P-hard to approximate $\left|Z(e^{i\pi/8})\right|^2$ up to multiplicative error $1/4 + o(1)$ for a $1/24$ fraction of instances, over the random choice of weights.

problem	space overhead	depth	advantage	assumptions
factoring [14]	polylog	poly, adapt.	superpoly	factoring hard
graph state [13]	poly	$\mathcal{O}(1)$, adapt.	superpoly	PH $= \infty$ & ACH
sparse IQP [this work]	polylog	$\mathcal{O}(1)$, adapt.	superpoly	PH $= \infty$ & ACH
magic square [15]	polylog	$\mathcal{O}(1)$	quasi-log	unconditional

Table 1: Potential candidates for the demonstration of robust quantum advantage. The advantage is relative between the quantum depth and its minimal classical counterpart. Factoring displays a superpolynomial advantage provided that factoring is hard classically, but requires the full machinery of fault-tolerance. Graph state sampling and sparse IQP sampling also give a large advantage, under stronger assumptions (that the Polynomial Hierarchy does not collapse, and with an Average Case Hardness conjecture) and can be implemented with an adaptive circuit of constant depth. Finally the magic square problem leads to an unconditional advantage with a non-adaptative circuit of constant depth, but only offers a logarithmic advantage compared to classical computing.

The sparse IQP problem is as follows: pick a random $D \in \mathfrak{D}_N$ according to the random process described before, and output an N-bit string s according to a distribution q_D such that

$$\|p_D - q_D\|_{\mathrm{TV}} \leq \delta, \tag{4}$$

where the total variation distance between two distributions p and q is defined as

$$\|p - q\|_{\mathrm{TV}} := \frac{1}{2} \sum_{s \in \{0,1\}^N} |p(s) - q(s)|.$$

Assuming Conjecture 1, and the non collapse of the Polynomial Hierarchy, a generalisation of the $P \neq NP$ conjecture widely considered to be true, Bremner *et al* proved that there is no efficient classical algorithm for the sparse IQP problem. More precisely,

Theorem 1 (Classical hardness of sparse IQP sampling [18])**.** *Assuming Conjecture 1, there exists $\delta > 0$ independent of N such that a constant fraction of sparse IQP circuits cannot be simulated by a polynomial-time classical algorithm up to precision δ in total variation distance unless the polynomial hierarchy collapses to its third level.*

This theorem states that on average over the choice of D from the probability distribution over $\mathfrak{D}_N$ defined previously, it is hard to sample classically from a distribution close to p_D. While a fault-tolerant quantum computer can sample efficiently from such a distribution, we do not expect that this is the case for near-term quantum processors. In fact, the initial proposal [18] partially addressed this issue by considering a simple noise model where the quantum circuit is assumed to be ideal, except for some independent and identically distributed noise added to the classical value of the final outcomes. Unfortunately, this model is too naive and a more realistic noise model should assume that *every gate* suffers from some constant level of noise. In that case, because the number of gates is of order $N \log N$ in the circuit, it is immediate that noise will accumulate through the circuit and that the level of noise per qubit cannot be assumed to be constant, independent of N. Here we choose to consider a more general error model – the *local stochastic noise model* [21] – that includes well-known error models such as the independent depolarizing noise channel but also allows for local correlated errors. In this model described in Section 4, errors are applied at each gate operation and the probability that faulty locations contain a specific set $\mathcal{A}$ is upper bounded by $p^{|\mathcal{A}|}$. In this work we propose a physical implementation of sparse IQP circuits that is robust to this kind of noise, without requiring the full machinery of fault-tolerant quantum computation.

In order to avoid multiple rounds of costly error correction, our main strategy is to make the encoded circuit of constant depth rather than logarithmic. This is challenging since the target logical circuit has logarithmic depth, and we want in addition to make it fault-tolerant. To this end, we design a family of quantum error-correcting codes on which sparse IQP circuits can be implemented in depth 1, meaning that they are fully parallelized. This is possible thanks to the commuting nature of sparse IQP gates [22]. In addition, we prove that the initial state can be encoded in constant quantum depth by performing stabilizer measurements. The only part of the process which is not implemented in constant depth is the final step of the state preparation: it consists of a single interaction with a classical computer that must compute a correction to apply, which depends on the stabilizer measurement results. In our scheme, this classi-

cal computation requires a polylogarithmic time because one needs to compute a correction for quantum patches of logarithmic size. We remark that similar time complexities are often neglected in the literature of quantum fault-tolerance [21, 23, 24], and it may in fact not be a very problematic issue in practice.

Given a circuit $D \in \mathfrak{D}_N$ and precision $\delta > 0$, we construct the circuit $C_D(\delta)$ that samples from a distribution that is δ-close to p_D in total variation distance after classical post-processing. While the final classical post-processing is not performed in constant time, this is not an issue since all the qubits have already been measured. We discuss this point in Section 4. The circuit $C_D(\delta)$ is illustrated in Figure 2 and we detail its construction in Section 2. For circuits D of depth $\Theta(\log N)$, the space overhead is polylogarithmic in the precision δ and in number N of logical qubits. Given that the average depth of sparse IQP circuits is $\Theta(\log N)$, a simple Markov inequality further implies that the fraction of circuits admiting a depth larger than $\alpha \log N$ decreases as $\mathcal{O}(1/\alpha)$. Thus an arbitrarily large fraction of such circuits benefits from the above overhead scaling. A practical difficulty that we do not address here is that our scheme requires long-range interactions. We state our main result:

Theorem 2 (Constant depth quantum advantage)**.** *There exists a universal $\varepsilon_{\text{th}} > 0$ such that, for all $N \in \mathbb{N}$, $D \in \mathfrak{D}_N$ and $\delta > 0$, running a noisy version of the quantum circuit $C_D(\delta)$ by inserting local stochastic noise of strength $\varepsilon < \varepsilon_{\text{th}}$ after each step, yields samples from p_D up to precision δ (in total variation distance) after classical post-processing.*

Combining this with Theorem 1, our scheme demonstrates a super-polynomial quantum advantage for the task of sparse IQP sampling, assuming Conjecture 1 and that the Polynomial Hierarchy does not collapse.

To summarize our contribution, we reduce the fault-tolerance space overhead required to demonstrate a superpolynomial quantum advantage with a constant depth quantum circuit, from a large degree polynomial in [13] to a polylogarithmic overhead. A similar reduction is achieved for the classical computation complexity during the quantum computation. This comes at the cost of losing the local connectivity of the scheme.

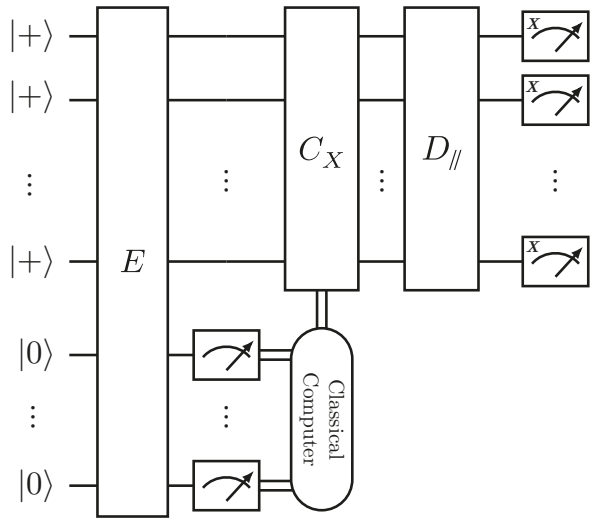

Figure 2: Our fault-tolerant implementation of a logical sparse IQP circuit D. The first layers E (stabilizer measurements) and C_X (adaptative error correction) prepare a logical state that is fed to a parallel version $D_{/\!/}$ of the sparse IQP circuit D, followed by single-qubit measurements. The overall circuit has constant (quantum) depth and acts on $N \times \text{polylog}(N)$ qubits. A single interaction with a classical computer is necessary to compute the correction C_X for the initial preparation. A final classical post-processing (not depicted) then computes a sample from the target distribution p_D.

1.2 Main concepts and ideas

1.2.1 The Tetrahelix code for fault-tolerant parallel computation

We recall that we aim to address two issues in order to get a final circuit of constant depth: we need to reduce the depth of the logical circuit for sparse IQP from logarithmic to constant, and we need to find a fault-tolerant version that remains of constant depth. We achieve this by combining two ideas.

First we rely on *3D color codes* [25, 26] which admit transversal diagonal gates. More specifically, we will focus on the *tetrahedral code* subfamily that admits a transversal T-gate. Moreover, because these codes are CSS codes [27, 28], they also admit a transversal CNOT gate. Combining both, we see that tetrahedral codes also have transversal CS-gates, as shown on Figure 3. The second idea is that it is possible to fully parallelize an IQP circuit by using a GHZ encoding of each of the input qubits, in order to trade depth for width of the circuit. This means encoding a $|+\rangle$ as $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes k} + |1\rangle^{\otimes k})$ for some k corresponding to the number of gates supposed to be applied to the qubit, so that k is logarithmic in N. Then all k gates can be performed simultaneously by act-

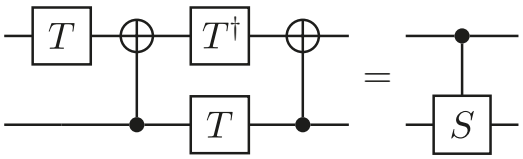

Figure 3: Implementation of the controlled-phase from controlled-not, T and $T^\dagger$ gates, all of those have transversal implementation on a tetrahedral color code. Switching T and $T^\dagger$ gives $CS^\dagger$

ing on a different qubit within the GHZ state. This is described in Figure 4. This new circuit has two shortcomings. First, despite being of constant depth, the logical phase-flip rate increases linearly with the size of the state since the measurement results of the k qubits within a GHZ state need to be aggregated. Second the preparation of bare GHZ states cannot be done fault-tolerantly in constant depth.

To solve these issues, we define a new stabilizer code – the *tetrahelix code* – combining the two ideas (3D color code for transversal gates and GHZ states for parallel implementation). We will detail the construction in Section 2, and only briefly explain its main properties here. The encoding is parameterized by two integers, k and L, accounting respectively for the parallelization capacity and the distance of the code. A k-tetrahelix code of distance L is defined by merging (in lattice surgery terms [29, 30, 31]) k tetrahedral codes of distance L along a 1-dimensional chain. Remarkably, the resulting $[\![\Theta(kL^3), 1, \Theta(L)]\!]$ tetrahelix code admits a depth-1 implementation of a logical sparse IQP circuit of depth k. This corresponds to a linear trade-off between the depth of the initial logical circuit and the number of physical qubits:

Lemma 1. *Any sparse IQP circuit of depth k on N qubits can be implemented in depth 1 on N logical qubits encoded in k-tetrahelix codes.*

The constant depth of the circuit, together with the arbitrarily large distance L, ensures the fault-tolerance of the circuit up to a final classical post-processing to decode the results of the qubit measurements. We discuss in Section 4 how to achieve this by exploiting efficient decoders of color codes. The complexity of this step remains negligible compared to the super-polynomial quantum advantage of the overall circuit. The remaining challenge concerns the initial preparation of the encoded states of the tetrahelix code. One needs to ensure that such a preparation can also be done in constant depth and in a fault-tolerant manner.

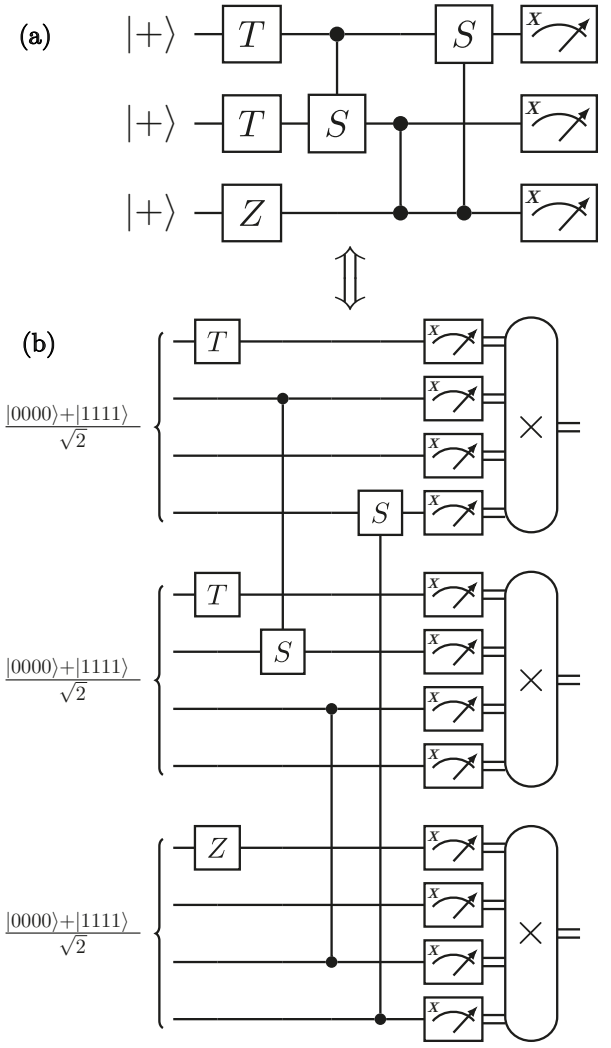

Figure 4: Each step $i \in \{1, ..., k\}$ of a circuit of depth k is simultaneously applied on the i^{th} physical qubit of all the GHZ states. The logical circuit (a) of depth 4 can be compiled in depth 1 up to classical decoding and state preparation by starting from GHZ state of size 4 and implementing circuit (b). The $\times$ blocks correspond to the classical decoding circuits.

1.2.2 Single-shot state preparation

A logical state of a quantum stabilizer code can always be prepared starting from a simple product state by measuring stabilizers and applying the appropriate correction to set the state in the code space. Such a scheme is however sensitive to measurement errors and fault-tolerance is usually achieved by repeating measurements. We circumvent this shortcoming by establishing the *single-shot* preparation of logical $|+\rangle$ states for the tetrahelix code.

In general, error correction based on erroneous measurements can induce large-weight physical errors whose accumulation could later translate into logical errors. In order to ensure fault-tolerance, one can prove that building on the particular structure of syndromes, the induced residual errors can be kept local with high probability. Such errors are then dealt with by the final classical decoding step. This property corresponds to single-shot decoding introduced by Bombín in [32]. Throughout this paper, $|\overline{x}\rangle$ denotes the logical encoded state $|x\rangle$ for $x \in \{0, 1, +, -\}$.

Lemma 2. *The tetrahelix code admits a single-shot preparation of $|\overline{+}\rangle$ / $|\overline{-}\rangle$ logical states, up to X stabilizers of the tetrahedral code.*

Note that, as argued in subsection 3.2, the X stabilizers need not be applied since they commute with sparse IQP encoded gates and hence can be propagated to the end of the circuit where they leave the final measurement unchanged.

The proof of the single-shot property of the k-tetrahelix code is detailed in Section 3 and relies on (i) the single-shot preparation of Hadamard basis states for 3D gauge color codes [32, 33], and (ii) the fact that the measurement errors occurring during code merging are detectable with the global stabilizer measurement outcomes.

We furthermore argue that the associated decoding can be performed on a classical computer in polylogarithmic-time with respect to N in Section 3. We consider it to be instantaneous to derive Theorem 2.

1.3 Sketch of proof of Theorem 2

The rest of the paper is devoted to establish Theorem 2. In Section 2, we first briefly review tetrahedral codes, from the 3D color code family. Next, we define the tetrahelix code family obtained by merging tetrahedral codes. We prove that the k-tetrahelix code reduces the depth k of a sparse IQP circuit to depth 1 (Lemma 1). In Section 3, we prove that the merging failure probability between two tetrahedral codes of distance L is exponentially suppressed in L and hence that k-tetrahelix encoded states in the Hadamard basis can be faithfully prepared in constant quantum depth (Lemma 2). In Section 4, we prove the fault-tolerance of the scheme. More precisely, we prove the existence of a non-zero error threshold independent of k, below which we arbitrarily suppress logical errors by increasing L for any encoded sparse IQP circuit.

2 Tetrahelix code

2.1 Overview of tetrahedral codes

Color codes are a family of topological codes introduced by Bombín and Martin-Delgado [34, 25, 26]. Their main feature is that they admit a transversal implementation of single-qubit phase gates, including the T-gate when the codes are 3-dimensional. In the following we focus on the subfamily of *tetrahedral codes* that encode a single qubit. Tetrahedral codes are defined on 3-dimensional color complexes, that we will call 3-colexes as in [35], of a tetrahedral shape as described in Figure 5 with the vertices corresponding to the data qubits. 3-Colexes are 3D lattices with the properties that (i) each cell is assigned one of four colors such that no two adjacent cells are of the same color; (ii) three colors appear on each external facet of the complex, and such a facet is associated with the missing color (in Figure 5 these facets correspond to the four triangular external boundaries of the tetrahedron); (iii) each vertex is incident to a cell or facet of all possible colors.

In the following we denote by $L \in \mathbb{N}$ the number of vertices on the edges of the lattice, and will correspond to the code distance, as explained below. The construction of a tetrahedral 3-colex is not unique for a given L but if one relies on tessellations of uniform density, then the resulting codes each encode a single logical qubit in $m = \Theta(L^3)$ physical qubits, and all display the properties that we will require.

Let us recall the formal definition of a tetrahedral code on m qubits, which will serve as a building block for the *tetrahelix code*. We start from a tetrahedral 3-colex with set of vertices $\mathcal{V}$, faces $\mathcal{F}$ and cells $\mathcal{C}$. Physical qubits are associates with vertices, so $m = |\mathcal{V}|$. The tetrahedral code associated to this

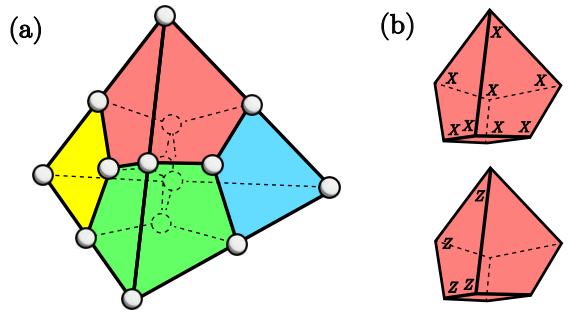

Figure 5: (a) The [15,1,3] Reed-Muller code is the smallest example of tetrahedral codes. Here qubits are on vertices and $\overline{X}$ and $\overline{Z}$ logical operators can be chosen respectively on a face and an edge of the tetrahedron. (b) X stabilizers are supported on cells (elements of $\mathcal{C}$) and Z stabilizers on faces (elements of $\mathcal{F}$).

colex is a CSS code with stabilizers given by:

$$S_X^1 = \langle X(c),\ c \in \mathcal{C} \rangle, \tag{5}$$

$$S_Z^1 = \langle Z(f),\ f \in \mathcal{F} \rangle. \tag{6}$$

Here, each cell c or face f is identified as a binary vector of length m with ones at the locations corresponding to the associated vertices, and we define $X(a) := \otimes_{i=1}^{m} X_i^{a_i}$ for $a \in \{0,1\}^m$. (and similarly for $Z(a)$). In words, the X stabilizer associated to a 3-cell c is the product of Pauli X operators on all the vertices in the boundary of c. In particular, X stabilizers are associated by the 3-cells of the colex and Z stabilizers are associated with the faces.

The fundamental property of tetrahedral color codes is that for each code of the family there exists a partition of vertices $\mathcal{V} = \mathcal{V}^+ \cup \mathcal{V}^-$ such that applying the gate T on $\mathcal{V}^+$ and $T^\dagger$ on $\mathcal{V}^-$ implements an encoded logical T-gate [33]:

$$T(\mathcal{V}^+)T^\dagger(\mathcal{V}^-)\,|\overline{x}\rangle = \overline{T}\,|\overline{x}\rangle\,. \tag{7}$$

Similarly to encoded states $|x\rangle$ denoted by $|\overline{x}\rangle$, we denote by $\overline{U}$ the encoded logical unitary U. Together with the existence of transversal controlled-not gates, this implies the transversal implementation of the CS-gate (see Figure 3):

$$CS(\mathcal{V}^+)CS^\dagger(\mathcal{V}^-)\,|\overline{x}\rangle\,|\overline{y}\rangle = \overline{CS}\,|\overline{x}\rangle\,|\overline{y}\rangle\,, \tag{8}$$

where $CS(\mathcal{V}^+)$ denotes the transversal application of CS between the analogous sets $\mathcal{V}^+$ of two code blocks. $\overline{X}$ and $\overline{Z}$ logical operators are respectively surface-like and string-like and the X distance and Z distance scale as $\Theta(L^2)$ and $\Theta(L)$, respectively.

2.2 Construction of the tetrahelix code

The transversality of the sparse IQP gate set paves the way towards the fault-tolerant implementation of such circuits. This would however require repeated error correction cycles at each circuit step, that is a logarithmic number of times. Concatenating a tetrahedral code with a repetition code gives a family of codes that present the desired parallelization property. Unfortunately, it does not meet the criteria of constant depth preparation for the initial encoded states. We now define a new code, the tetrahelix code, that displays both properties: (i) depth-1 implementation of a sparse IQP circuit, (ii) constant-depth encoded state preparation in the Hadamard basis.

As briefly mentioned in subsection 1.2.1, a tetrahelix code is obtained by merging tetrahedral codes in lattice surgery terms [29, 30, 36]. Here we detail the construction starting by merging two such codes of distance L, with respective sets of vertices $\mathcal{V}_1$ and $\mathcal{V}_2$ as described in Figure 6(a). We consider two codes which are exact mirror images of one another and we denote by $\varphi : \mathcal{V}_1 \to \mathcal{V}_2$ the bijection between the two sets of vertices. An external triangular facet, $B_1 \subset \mathcal{V}_1$, together with its mirror image, $B_2 = \varphi(B_1)$, are chosen and every vertex is paired with the corresponding one on the other code. This pairing defines a set of pairs $\mathcal{P}_{1,2} := \{(v, \varphi(v)) | \forall v \in B_1\}$.

The merge operation consists in fusing the X stabilizers on the boundaries and adding new Z stabilizers of weight 2 associated to the paired qubits in $\mathcal{P}_{1,2}$. More precisely, the Z stabilizers are defined as $\langle Z(f), f \in \mathcal{F}^2 \rangle$ with

$$\mathcal{F}^2 := \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{P}_{1,2}. \tag{9}$$

Similarly, we define $\mathcal{C}^2$ the union of merged stabilizers and unmerged ones:

$$\mathcal{C}^2 := \mathcal{C}_1^* \cup \mathcal{C}_2^* \cup \mathcal{M}_{1,2} \tag{10}$$

with

$$\mathcal{C}_i^* := \{C \in \mathcal{C}_i | C \cap B_i = \varnothing\}, \tag{11}$$

and

$$\mathcal{M}_{1,2} := \{C_1 \cup C_2 | C_1 \in \mathcal{C}_1, C_2 \in \mathcal{C}_2, \\ \varphi(C_1 \cap B_1) = C_2 \cap B_2 \neq \varnothing\}. \tag{12}$$

X stabilizers are then defined as $\langle X(c), c \in \mathcal{C}^2 \rangle$ corresponding to non-adjacent cells of each code and fused adjacent ones (paired according to their color as in Figure 6(a)). This construction ensures that X stabilizers commute with every Z stabilizer including

the newly defined Z stabilizers with support on $\mathcal{P}_{1,2}$. We denote by $\overline{X}_i$ and $\overline{Z}_i$ the logical operators of the initial tetrahedral codes from which we define the logical operators of the new code. The 2-tetrahelix code encodes a single logical qubit for which the logical operators can be taken of the form $\overline{Z} = \overline{Z}_1$ (or $\overline{Z}_2$) and $\overline{X} = \overline{X}_1\overline{X}_2$ with $\overline{X}_2$ chosen so that its support intersects on the same subset of $\mathcal{P}_{1,2}$ than $\overline{X}_1$ (to commute with the associated Z stabilizers).

Merging additional tetrahedra does not fundamentally change the analysis. Tetrahedra can be aligned in the shape of Figure 6(c) to form a chain of length $k \in \mathbb{N}$ so that each extremal vertex is shared between at most four tetrahedra. This ensures that at most four X stabilizers are fused together. This linear packing of regular tetrahedra is known as a Boerdijk-Coxeter helix or tetrahelix [37, 38] which motivates the name of the code.

Denoting by $\mathcal{V}_i$ the set of vertices of the i^{th} tetrahedral color code, we get a partition of the set of all vertices $\mathcal{V} = \cup_{i=1}^{k}\mathcal{V}_i$. With $\mathcal{P}_{i,i+1}$ denoting new Z stabilizers between adjacent tetrahedra i and $i+1$, $\mathcal{F}^k$ and $\mathcal{C}^k$ are defined analogously as $\mathcal{F}^2$, and $\mathcal{C}^2$ to ensure stabilizer commutation:

$$\mathcal{F}^k := \bigcup_{i=1}^{k-1}(\mathcal{F}_i \cup \mathcal{P}_{i,i+1}) \cup \mathcal{F}_k, \tag{13}$$

$$\mathcal{C}^k := \bigcup_{i=1}^{k-1}(\mathcal{C}_i^* \cup \mathcal{M}_{i,i+1}) \cup \mathcal{C}_k^*, \tag{14}$$

where here $\mathcal{C}_i^*$ and $\mathcal{M}_{i,i+1}$ are defined recursively so that stabilizers can be merged across several tetrahedra (up to three on edges and up to four on summits). We define the k-tetrahelix code that encodes a single logical qubit in $\Theta(kL^3)$ physical qubits from its set of stabilizers:

$$S_X^k := \langle X(c),\ c \in \mathcal{C}^k\rangle, \tag{15}$$

$$S_Z^k := \langle X(f),\ f \in \mathcal{F}^k\rangle. \tag{16}$$

The logical $\overline{Z}$ operator can be chosen as any of the logical $\overline{Z}_i$ operators of the composing tetrahedral codes, while the $\overline{X}$ logical operator is a product of $\overline{X}_i$ operators recursively chosen so that $\overline{X}_i$ and $\overline{X}_{i+1}$ intersect with the same subset of $\mathcal{P}_{i,i+1}$.

2.3 Code distance

The X and Z distances of a code correspond to the minimal weights of X and Z logical operators. We denote by d_X^k and d_Z^k the X and Z distances of the k-tetrahelix code and prove that:

$$d_Z^k = \Theta(L) \quad \text{and} \quad d_X^k = \Theta(kL^2). \tag{17}$$

We prove the result for the 2-tetrahelix code by relating logical operators of the tetrahelix code to those of the initial tetrahedral codes, the result generalizes to arbitrary k-tetrahelix code by recursion. In this subsection we denote by $d_X^1 = \Theta(L^2)$ and $d_Z^1 = \Theta(L)$ the X and Z distances of a tetrahedral code of edge length L and number of qubits $m = \Theta(L^3)$.

Let us consider a logical operator $\overline{Z}$ of the 2-tetrahelix code. We index the vertices of the two composing tetrahedra in a symmetric manner with respect to the paired facets. The logical operator $\overline{Z}$ is of the form of the tensor product of Pauli Z operators on each tetrahedron $\overline{Z} = Z(\mu) \otimes Z(\nu)$, with $\mu, \nu \in \{0,1\}^m$. We will show that up to multiplication by Z stabilizers we can transfer $Z(\mu) \otimes Z(\nu)$ to $Z(\mu+\nu) \otimes \mathbb{1}$. This means that we transfer the physical Pauli Z operators from the second tetrahedron to the symmetric ones in the first one. We can then conclude by noticing that $Z(\mu+\nu)$ is a logical operator of the first tetrahedron and hence of weight larger or equal to d_Z^1. We thus have

$$|\overline{Z}| = |Z(\mu)\otimes Z(\nu)| \geq |\overline{Z}(\mu+\nu)\otimes\mathbb{1}| \geq d_Z^1, \tag{18}$$

which concludes the argument.

Indeed, since an arbitrary logical $\overline{Z}_1$ operator of the first tetrahedron is also a logical operator of the tetrahelix code, there exists a Z stabilizer R_Z such that $\overline{Z} = \overline{Z}_1 \times R_Z$. Such a stabilizer is necessarily of the form:

$$R_Z = R_Z^1 \times R_Z^2 \times R_Z^{1,2}, \tag{19}$$

where R_Z^1 is a Z stabilizer of tetrahedron 1, R_Z^2 of tetrahedron 2, and $R_Z^{1,2}$ is a product of Z-stabilizers defined at the boundary (paired qubits in $\mathcal{P}_{1,2}$). It is clear that, multiplying $\overline{Z}$ by R_Z^2, maps the support from the bulk of the second tetrahedron to the paired facet of this tetrahedron. Next, we completely transfer this support to the facet of first tetrahedron by multiplying by $R_Z^{1,2}$. At this point, the support of the logical operator is entirely contained in the first tetrahedron. Now we apply the symmetric version of R_Z^2 defined on the first tetrahedron. This maps the original logical operator to $Z(\mu+\nu)\otimes\mathbb{1}$.

The case of the X distance is straightforward as the product of $\overline{X}_1$ and $\overline{X}_2$ logical operators whose supports intersect with the same pairs of $\mathcal{P}_{1,2}$ yields

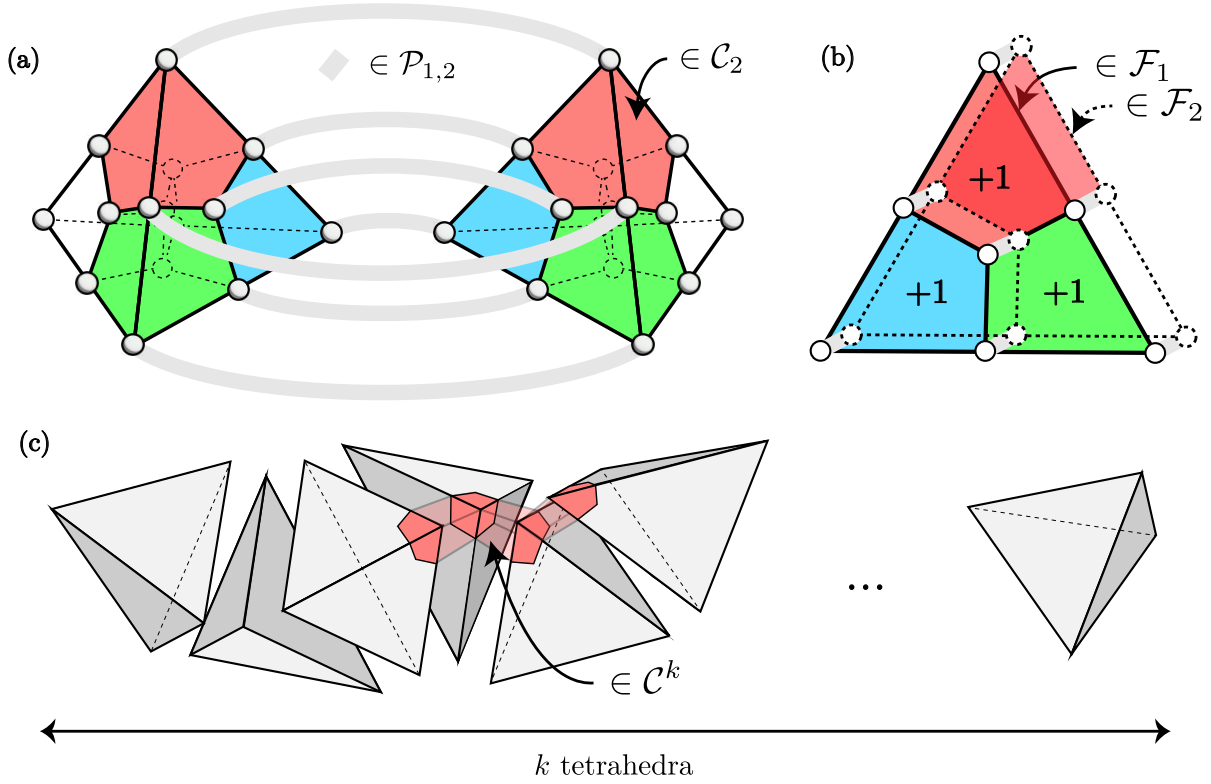

Figure 6: (a) Adjacent tetrahedra (here for $L = 3$) are merged by measuring pairs of qubits from $\mathcal{P}_{1,2}$ that become Z stabilizers of the new code. The colors of the second tetrahedron are chosen by convention so that merged X stabilizers are of the same color. (b) Every new Z stabilizer generator must be set to 1 to project the state in the code space. The new pair stabilizers value are not independent since they are all related to Z stabilizers of the two tetrahedral codes on the face on which the merge is performed. In particular the product of four pairs overlapping the same Z stabilizer (of support in $\mathcal{F}_1$) is equal to 1. (c) Merging additional tetrahedra with each other enables to form a chain of tetrahedra of length k. The optimal chain has the shape of an helix and corresponds to minimizing the number of merged X stabilizers (of support in $\mathcal{C}^k$) that is equal to four in this packing.

a logical operator of the 2-tetrahelix code, and this product form is stable upon multiplication by X stabilizers. This stability is a direct consequence of the fact that the restriction of a tetrahelix X stabilizer to a single tetrahedron is a stabilizer of the tetrahedral code. Merging an additional tetrahedral code hence increases the X distance by d_X^1:

$$|\overline{X}| = |\overline{X}_1 \overline{X}_2| \geq 2d_X^1. \tag{20}$$

The same discussion between a $(k-1)$-tetrahelix code and a tetrahedral code generalises the proof by recursion to k-tetrahelix code for arbitrary k. Recalling that $d_Z^1 = \Theta(L)$ and $d_X^1 = \Theta(L^2)$, we obtain the bounds of (17).

2.4 Parallel computation

We turn to the properties of the code concerning parallel computation. We establish Lemma 1 by showing that the encoded T-gate can be implemented in depth 1 on a single tetrahedron of the chain. A tetrahedral code on m physical qubits is a CSS code and logical states can therefore be written in the form

$$|\overline{x}_1\rangle = \frac{1}{\sqrt{|\mathcal{S}^1|}} \sum_{s_1 \in \mathcal{S}^1} |s_1 + x_1 L_1\rangle\,, \tag{21}$$

with the addition taken modulo 2 and $x_1 \in \{0,1\}$ and $\mathcal{S}^1 \subset \{0,1\}^m$ such that for $s_1 \in \mathcal{S}^1$ we have $X(s_1) \in S_X^1$. Similarly, $L_1 \in \{0,1\}^m$ represents an arbitrary $\overline{X}_1$ logical operator. The transversal implementation of the T-gate $T(\mathcal{V}^+)T^\dagger(\mathcal{V}^-) = \overline{T}$ on the tetrahedral code implies that each codeword gains the same phase from the application of $T(\mathcal{V}^+)T^\dagger(\mathcal{V}^-)$:

$$T(\mathcal{V}^+)T^\dagger(\mathcal{V}^-)\,|s_1 + x_1 L_1\rangle = e^{\frac{i\pi}{4}|x_1|}\,|s_1 + x_1 L_1\rangle\,. \tag{22}$$

The logical computational states of the k-tetrahelix code are given by

$$|\overline{x}\rangle = \frac{1}{\sqrt{|\mathcal{S}^k|}} \sum_{s \in \mathcal{S}^k} |s + xL\rangle\,, \tag{23}$$

for $x \in \{0,1\}$. Here $\mathcal{S}^k \subset \{0,1\}^{k\times m}$ is such that for $s \in \mathcal{S}^k$, we have $X(s) \in S_X^k$ and L is the vector associated to an arbitrary logical $\overline{X}$ operator. For each X stabilizer, $s \in \mathcal{S}^k$ is a concatenation of k vectors $s_i \in \mathcal{S}^1, i \in \{1,\dots,k\}$: $s = [s_1, ..., s_k]$. Similarly, for the logical operator, the binary vector L is a concatenation of vectors L_i each representing a logical $\overline{X}_i$ operator of the i^{th} tetrahedral code. Focusing on tetrahedron i_0, and up to qubit re-ordering, we can thus write

$$|\overline{x}\rangle = \frac{1}{\sqrt{|\mathcal{S}^k|}} \sum_{s_{i_0}\in\mathcal{S}^1} |s_{i_0} + xL_{i_0}\rangle \otimes |\psi(s_{i_0}, x)\rangle \,. \tag{24}$$

The terms $|\psi(s_{i_0}, x)\rangle$ depend on s_{i_0} because of correlations between codewords restricted to different tetrahedra induced by overlapping X stabilizers, but this does not impact our argument. Taking $\mathcal{V}_{i_0}^+$ and $\mathcal{V}_{i_0}^-$ as in (7), we have

$$T(\mathcal{V}_{i_0}^+)T^\dagger(\mathcal{V}_{i_0}^-)\,|s_{i_0} + xL_{i_0}\rangle = e^{\frac{i\pi}{4}|x|}\,|s_{i_0} + xL_{i_0}\rangle \,. \tag{25}$$

Combining (24) and (25) directly implies:

$$T(\mathcal{V}_{i_0}^+)T^\dagger(\mathcal{V}_{i_0}^-)\,|\overline{x}\rangle = \overline{T}\,|\overline{x}\rangle \,. \tag{26}$$

To extend the arguments to the CS-gate, we would need to use the gadget of Figure 3. Notice that while the T and $T^\dagger$-gates are applied on a single tetrahedron, the CNOT gates would need to be applied on all physical qubits (CSS property). However, as these CNOT gates come in pairs, they cancel each other outside the tetrahedron where the T-gates are applied. Thus, the CS-gate can also be applied between two arbitrary tetrahedral blocks of two tetrahelix codes:

$$CS(\mathcal{V}_{i_0}^+)CS^\dagger(\mathcal{V}_{i_0}^-)\,|\overline{x}\rangle\,|\overline{y}\rangle = \overline{CS}\,|\overline{x}\rangle\,|\overline{y}\rangle \,. \tag{27}$$

This implies that the k-tetrahelix code can implement in a single step an encoded T or a CS-gate on each tetrahedral block of the code. Therefore, we obtain a depth-1 parallel implementation of a depth-k sparse IQP circuit up to state preparation. This finishes the proof of Lemma 1. In the next section we prove that encoded states in the Hadamard basis can be prepared fault-tolerantly in constant quantum depth.

3 Constant-depth preparation of encoded states

The encoded states of the k-tetrahelix code in the Hadamard basis can be prepared by merging k associated encoded states of the composing tetrahedral blocks. In this section, we show that both the steps of preparing encoded tetrahedral states and their merging can be done in constant quantum depth.

3.1 Single-shot decoding of tetrahedral code

Since the state $|+\rangle^{\otimes m}$ is stabilized by all X stabilizers and by the $\overline{X}$ logical operator of a CSS quantum code, the projection over the logical $|\overline{+}\rangle$ can ideally be done by measuring the Z stabilizers and a single step of Pauli X corrections. Measurement errors however usually prevent such reliable encoded state preparation in constant depth. Indeed, measurement errors induce residual data errors after Pauli corrections, which usually calls for many repeated measurements before such a correction is applied. Repeating measurements gives an extra dimension to the error syndrome where ancilla and data errors can be separated by the decoding algorithm which infers an error pattern close to the most likely one and hence an appropriate correction.

An alternative approach is to build on the structure of the error syndrome of some codes to ensure that a single round of local measurements is sufficient to ensure the locality of the residual errors with high probability. This strategy was first proposed by Bombín in [32] for 3D gauge color codes and is known as *single-shot decoding* which is a property of a quantum error-correcting code in conjunction with its decoder. In the case of 3D color codes, Z stabilizers on faces correspond to Z gauge operators of 3D gauge color codes. This ensures the single-shot decoding property up to a classical computation of polynomial complexity in the code size. Furthermore, the topological nature of the code implies that the measurements can be parallelized to a constant quantum depth. In conclusion, we have a constant depth preparation of encoded states in the Hadamard basis for the tetrahedral code up to local residual errors.

3.2 Single-shot merging of tetrahedral states

Merging two tetrahedral codes of distance L into a 2-tetrahelix code is described in Figure 6(a-b). Pairs of qubits from $\mathcal{P}_{1,2}$ are measured over the faces on which tetrahedral codes are merged and a correction is applied depending on the measurement outcomes. Since facets of a tetrahedral code have the structure of a triangular code (2D color code) of size L, in the absence of errors, measurements yield a binary codeword w of the corresponding classical code. This codeword can be written as the sum of an X stabi-

lizer and an X logical operator of the 2D code with the same formalism used in Section 2.4 for the 3D code.

The appropriate correction then can be seen to be a 3D color code codeword whose restriction to the triangular code vertices gives w. This can be obtained by determining first the decomposition of w into facets of the triangular code (X stabilizer generators) and the logical operator X over the entire triangle (so that it is a logical operator of both the 2D and the 3D codes) before mapping the facets to cells to get a 3D code codeword

$$\begin{aligned} w = s_{2D} + xL_{2D} \\ \rightarrow s_{3D} + xL_{3D} = \mathrm{corr}(w) \end{aligned} \tag{28}$$

for $x \in \{0, 1\}$. Importantly, an X stabilizer of the tetrahedral code commutes with encoded T and CS-gates on the tetrahelix code since it does not change the structure of the codewords described in subsection 2.4. Since the circuit ends with X measurements this means that it is sufficient for our purpose to compute x and only apply the logical part of the correction. In other words, we only need to prepare tetrahelix encoded states up to tetrahedral codes X stabilizers.

If the two tetrahedral codes are not perfectly in their code space, or in the case of measurement errors, the measurement results deviate from w:

$$w_e = w + e_r + e_m. \tag{29}$$

Here, e_r accounts for the residual errors of the tetrahedral states preparation, and e_m stands for measurement errors. Because the preparation of encoded states in the Hadamard basis for the tetrahedral code is single-shot, the resulting errors follow a local stochastic noise model. This is also the case for measurement errors and hence decoding the triangular code yields the correct value of $\overline{Z}_1\overline{Z}_2$ with probability exponentially close to 1 in L. $\overline{Z}_1\overline{Z}_2$ is then set to +1 by applying or not $\overline{X}_1$.

A k-tetrahelix code encoded state can then be prepared in a similar manner simply by repeating the merging operation with additional tetrahedral codes, while always applying the logical correction on the tetrahedron for example on the left of the merge. This scheme can be seen as similar to preparing a GHZ state of size k from parity measurements and logical correction, with the difference that here, measurement errors are exponentially suppressed, thus giving Lemma 2.

Efficient decoding algorithms exist for 2D color codes [39, 40] and 3D color codes [32, 41] and are single-shot for the 3D case. These algorithms have a complexity polynomial in the code size. The different tetrahedral encoded states can be prepared in parallel. Parallel merge measurements followed by iterative computation of the associated correction then give a preparation of tetrahelix encoded states with polynomial in L and proportional to k classical computation. We prove in Section 4 that we need a polylogarithmic number of qubits per code block which hence gives a polylogarithmic-time classical computation.

4 Application to sparse IQP circuits

In this section, we apply the results of the two previous sections to demonstrate the main result of this paper stated in Theorem 2. We start by presenting the error model. Next, we show that the encoding of the circuit of Figure 2 is fault-tolerant by proving the existence of an error threshold. Finally, we provide an estimation of the space overhead of the scheme.

4.1 Error model

The coupling of the quantum system with the environment generates noise that can later induce errors in the computation. We use the *local stochastic quantum noise model* from [21] where the set of faulty locations is a random variable of a discrete space-time and local correlations are allowed. No assumption is made on a particular type of error operator. This makes the model general enough to cover a wide class of applications. In particular this captures commonly studied noise channels such as depolarizing and dephasing noise, or amplitude damping.

A noise model of parameter ε that satisfies the following two properties is said to be locally stochastic: (i) the faults are confined to a random set of space-time locations $\mathcal{A} \subset V$ with probability $p(\mathcal{A})$ and (ii) the probability that a set of faulty locations contains a specific set of $\mathcal{A}$ locations is upper bounded by $\varepsilon^{|\mathcal{A}|}$.

$$\sum_{\mathcal{A}' \supseteq \mathcal{A}} p(\mathcal{A}') \leq \varepsilon^{|\mathcal{A}|}. \tag{30}$$

Final measurements are performed in the Hadamard basis and hence at the end of the circuit only Z-type errors induce errors on the classical output. Z errors can either be environmentally induced or generated during the propagation of X-type errors in the

circuit. X errors can also arise due to the coupling with the environment but also from incorrect preparation of encoded states (recall that the preparation only includes X correction). Local stochastic errors propagate as such through the constant depth circuit but residual errors after encoded states preparation are not necessarily local. We showed in Section 3 that their non-local representatives admit exponentially low probabilities which implies that the correction of local stochastic errors by the final decoding is sufficient to exponentially suppress the logical error rate.

More formally, residual errors after preparation of tetrahedral encoded states are characterised in [32] such that (i) correctable physical errors follow a local stochastic noise model $\mathcal{N}^{\varepsilon}_{\text{T,loc}}$, (ii) non-correctable physical errors (that is to say errors whose correction attempt induces a logical error) are exponentially suppressed, we denote by $\mathcal{N}^{\tilde{\varepsilon}_1}_{\text{T,nc}}$ the corresponding error channel. Using a similar notation we call $\mathcal{N}^{\tilde{\varepsilon}_2}_{\text{M,nc}}$ the channel associated to logical errors due to unsuccessful merging, with probability exponentially suppressed in the code distance.

The encoded states preparation error channel then writes with $\tilde{\varepsilon}_1$ and $\tilde{\varepsilon}_2$ exponentially suppressed in ε:

$$\mathcal{N}^{\varepsilon}_{\text{prep}} = \mathcal{N}^{\varepsilon}_{\text{T,loc}} \circ \mathcal{N}^{\tilde{\varepsilon}_1}_{\text{T,nc}} \circ \mathcal{N}^{\tilde{\varepsilon}_2}_{\text{M,nc}}. \tag{31}$$

The two non-correctable terms contribute to the final logical error rate but are exponentially rare. In the following subsections we prove that low enough local stochastic noise is corrected by the tetrahelix code. Post-processing of final single qubit measurements in the form of the tetrahelix code decoding then yields the value of the logical measurement. In the following we analyze error configurations and describe an efficient decoder from 2D and 3D color code decoders.

4.2 Existence of a good decoder

In the 3D color code, the logical $\overline{Z}$ operator corresponds to strings of Pauli Z connecting the four boundaries of different colors. An extremal vertex of the tetrahedron belongs to three boundaries and a string connecting this vertex to the opposite face of the remaining color hence yields an example of a $\overline{Z}$ logical operator. Logical errors arise when more than half of the respective phases of any such path are flipped. Errors on the tetrahelix code have a similar origin except that error strings can jump between tetrahedra to connect boundaries of different colors as described in Figure 7. This means that we cannot individually decode tetrahedra and that we first need to split (in lattice surgery language) the chain to retrieve tetrahedral codes.

This can be performed in software after final single-qubit X measurements by reconstructing the value of X stabilizers of the tetrahedral codes. Considering the example of the 2-tetrahelix code for simplicity, X stabilizers at the interface between the two tetrahedra were merged and hence, taken individually, do not stabilize the tetrahelix code. This means that they will initially not necessarily be in their +1 eigenspace even without errors. This can be fixed by applying Z stabilizers from $\mathcal{P}_{1,2}$ to set them to +1 while acting trivially on the code space of the tetrahelix code. This can be seen as preparing two triangular codes (2D color codes) logical states on the two facets by applying a Pauli operator of the form

$$Z(\sigma) \otimes Z(\varphi(\sigma)), \tag{32}$$

with $\sigma \subset B_1$ a set of vertices from the triangular facet on which the merge was performed, and φ the bijection between the two tetrahedra sets of vertices defined in Section 2. Note here that any potential logical error applied to one tetrahedron would also be applied to the second one.

In reality, errors arising on the support of tetrahedral codes X stabilizers prevents all such stabilizer to be set back to +1 by applying Z stabilizers from $\mathcal{P}_{1,2}$. Since in this scheme we aim at correcting errors at the next step during individual tetrahedral codes decoding we only need here to approach the tetrahedral code spaces. This can be done by minimizing the number of tetrahedral code X stabilizers with value -1 in the chain. For the k-tetrahelix code we start by X stabilizers merged between more than two tetrahedra, that is to say on tetrahedra vertices and edges, followed by those on the bulk of the facet on which tetrahedral codes are merged.

Once each tetrahedron is back to the tetrahedral code space (up to physical errors) it suffices to individually decode each code and multiply the logical values of the $\overline{X}_i$'s to recover the desired logical information (and hence pairs of tetrahedral codes logical errors possibly introduced at the splitting step cancel each other). For a low enough error rate we thus expect the logical error rate $\bar{\varepsilon}$ after such decoding to be proportional to the number of tetrahedra in the chain and to the logical error rate of a single tetrahedral code:

$$\bar{\varepsilon} = \mathcal{O}\Big(kL^3\Big)(\varepsilon/\varepsilon_{\text{th}})^{\mathcal{O}(L)}. \tag{33}$$

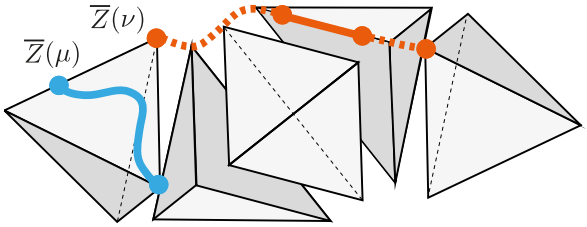

Figure 7: Representation of $\overline{Z}$ logical error configurations. Error strings are no longer restricted to a single tetrahedron (blue) but can also connect neighbouring tetrahedra (red). In the tetrahelix stacking, error strings can jump up to two tetrahedra at once.

Here we have only used 2D and 3D color codes decoders and therefore the existence of efficient 2D and 3D color code decoders [39, 42, 40, 41] implies the existence of an efficient decoder for the tetrahelix code. The formal definition and analysis of such a decoder under the general noise model considered here is beyond the scope of this paper and in the following we will prove Theorem 2 by relying on existing results on quantum LDPC codes. To do so we show that the code admits a non-zero threshold independent of k so that for low enough noise the logical error can be made arbitrarily low by increasing L.

4.3 Existence of a threshold for minimum weight decoder and local stochastic noise

The tetrahelix code is a quantum LDPC code since its generators have a bounded weight and each qubit is involved in a bounded number of generators. It is known that a family of $[[\tilde{n}, \tilde{k}, \tilde{d}]]$ quantum LDPC codes, with $\tilde{n}$ and $\tilde{d}$ scaling to infinity, and experiencing local stochastic noise of parameter ε, admits a non-zero error threshold ε_{th} [21]. More precisely, below this threshold the logical error rate is exponentially suppressed as

$$\overline{\varepsilon} = \mathcal{O}\left(\tilde{n}(\varepsilon/\varepsilon_{\text{th}})^{\tilde{d}/2}\right), \tag{34}$$

using the *minimum weight decoder*.

In the case of tetrahelix code, k is in general an independent parameter from L the distance of the code. Applying directly the results of [21] would lead to a threshold dependent on the value of k. We take care of this issue by imposing $k = \mathcal{O}(L)$. Thus, the associated family of k-tetrahelix codes admits a number of physical qubits $\tilde{n} = \mathcal{O}(L^4)$, and such a $[[\tilde{n} = \Theta(kL^3), \tilde{k} = 1, \tilde{d} = \Theta(L)]]$ code admits a non-zero threshold ε_{th} with L scaling to infinity. Note that, while the minimum weight decoder is not efficient in general, we expect the efficient decoder of the previous subsection to present similar error suppression property.

In the following subsection, we show that imposing $k = \mathcal{O}(L)$ is compatible with the desired parallelization and fault-tolerance properties.

4.4 Proof of Theorem 2

When implementing sparse IQP circuits on N qubits on k-tetrahelix codes, k, L and N are related through three relations. First, as discussed in the previous subsection, to ensure the existence of threshold independent of k, we need to have

$$k = \mathcal{O}(L). \tag{35}$$

Second, an arbitrarily large fraction of sparse IQP circuits on N qubits are of depth $\Theta(\log N)$ and can hence be implemented on k-tetrahelix codes with:

$$k = \Theta(\log N), \tag{36}$$

Third, another relation between L and N results from the required code size to reach the target precision δ of the sparse IQP problem. A logical sparse IQP circuit D is implemented with a k-tetrahelix code by the circuit C_D. After the final decoding, one obtains samples from a distribution $p_{D,\overline{\varepsilon}}$. For a constant logical error rate $\overline{\varepsilon}$ per logical qubit, the union bound gives an upper bound to the distance between the noisy and the ideal probability distributions with respect to N:

$$\|p_D - p_{D,\overline{\varepsilon}}\|_{\text{TV}} \leq \mathcal{O}(N\overline{\varepsilon}). \tag{37}$$

Keeping the noisy probability distribution δ-close to the ideal distribution thus imposes a logical error rate $\overline{\varepsilon}$ at most $\mathcal{O}(\delta/N)$. Logical errors can arise both from local stochastic errors and remaining non-local residual errors induced by merging errors or tetrahedral encoded states preparation errors, all of which are exponentially suppressed in L below some threshold:

$$\overline{\varepsilon} = \left(\frac{\varepsilon}{\varepsilon_{\text{th}}}\right)^{\Theta(L)}, \tag{38}$$

where the polynomial dependency on L in equation (34) is absorbed by the exponential. From (37) and (38), we derive the third equation relating L and N,

$$L = \Omega\left(\frac{\log(N/\delta)}{\log(\varepsilon_{\text{th}}/\varepsilon)}\right). \tag{39}$$

For a given N, it is enough to take

$$L = \Theta\left(\frac{\log(N/\delta)}{\log(\varepsilon_{\text{th}}/\varepsilon)}\right) \quad \text{and} \quad k = \Theta(\log N), \tag{40}$$

which automatically also satisfy (36). The total number of qubits n of each code block for a sparse IQP circuit of width N then reduces to:

$$n = \Theta(kL^3) = \Theta\left(\frac{\mathrm{polylog}(N/\delta)}{\mathrm{polylog}(\varepsilon_{\mathrm{th}}/\varepsilon)}\right). \qquad (41)$$

This completes the proof of Theorem 2. We note that for the (arbitrarily small) fraction of sparse IQP circuits of super-logarithmic depth the overhead is at most polynomial since the depth of sparse IQP circuits is at most linear.

5 Discussion

We have proposed a fault-tolerant implementation of sparse IQP circuits, paving the way for demonstrations of super-polynomial quantum advantage in near or mid-term experiments. It consists in a constant-depth quantum circuit and involves a single step of feed-forward from classical computation. To do this we have introduced the tetrahelix code admitting single-shot preparation of logical $|+\rangle$ states and transversal implementation of IQP circuits. The qubit overhead and classical computation time of our scheme are only polylogarithmic in the width of the original sparse IQP circuit. The requirements of our protocol are almost met by current NISQ experiments. We hope it can bring within reach demonstration of super-polynomial advantage of quantum over classical computation.

Depending on the physical platform the main complexity of the protocol may be coming from the required connectivity. A single tetrahelix code requires 3D connectivity. On top of this, each physical qubit has to interact with a single other qubit from another tetrahelix code during the implementation of the IQP circuit. These additional interactions are potentially long range. In the same spirit as in [30], the interaction range can be reduced using longer and branching tetrahelix codes while staying 3D. Logical CZ-gates can be realized facet to facet [43], but CS-gates will still require transversal connectivity between tetrahedra. Finding other codes with similar properties but simpler connectivity could ease the implementation even more.

The tetrahelix code that we propose in our implementation has interesting properties in itself. The ability to implement many different non-Clifford unitaries in a transversal manner could potentially be leveraged in other settings. One can take inspiration from this construction to design other codes with large sets of transversal non-Clifford unitaries to locally trade depth for width in larger scale algorithms. The key ingredient of our approach is the commuting nature of sparse IQP gates that enables their parallelization, in the spirit of [44] for MBQC, but in a fault-tolerant manner. Note that a generalization of the construction to any set of commuting gate would have powerful applications [22].

Concerning the encoding of sparse IQP circuits on tetrahelix codes, it is not clear whether or not the trade-off between depth and width is optimal but the noticeable asymmetry between the X and Z distances of the tetrahelix code seems to indicate the overhead could be reduced by balancing them out. Another direction would be to improve the error threshold of the scheme, possibly with post-selection in the spirit of [45]. This would participate bringing the scheme further within reach of current experiments [46].

During the preparation of this work, we became aware of a similar work on reducing error correction requirements for fault-tolerant quantum advantage [47].

Acknowledgements

We acknowledge support from the Plan France 2030 through the project NISQ2LSQ ANR-22-PETQ-0006, HQI ANR-22-PNCQ-0002 and from Inria EQIP challenge.

References

[1] Austin P Lund, Michael J Bremner, and Timothy C Ralph. "Quantum sampling problems, bosonsampling and quantum supremacy". npj Quantum Information **3**, 1–8 (2017).

[2] Ramis Movassagh. "The hardness of random quantum circuits". Nature Physics **19**, 1719–1724 (2023).

[3] Scott Aaronson and Alex Arkhipov. "The computational complexity of linear optics". In Proceedings of the forty-third annual ACM symposium on Theory of computing. Pages 333–342. (2011).

[4] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. "On the complexity and verification of quantum random circuit sampling". Nature Physics **15**, 159–163 (2019).

[5] Dan Shepherd and Michael J Bremner. "Temporally unstructured quantum computation". Pro-

ceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **465**, 1413–1439 (2009).

[6] Dorit Aharonov, Xun Gao, Zeph Landau, Yunchao Liu, and Umesh Vazirani. "A polynomial-time classical algorithm for noisy random circuit sampling". In Proceedings of the 55th Annual ACM Symposium on Theory of Computing. Pages 945–957. (2023).

[7] Yiqing Zhou, E Miles Stoudenmire, and Xavier Waintal. "What limits the simulation of quantum computers?". Physical Review X **10**, 041038 (2020).

[8] John C Napp, Rolando L La Placa, Alexander M Dalzell, Fernando GSL Brandao, and Aram W Harrow. "Efficient classical simulation of random shallow 2d quantum circuits". Physical Review X **12**, 021021 (2022).

[9] Xun Gao, Marcin Kalinowski, Chi-Ning Chou, Mikhail D Lukin, Boaz Barak, and Soonwon Choi. "Limitations of linear cross-entropy as a measure for quantum advantage". PRX Quantum **5**, 010334 (2024).

[10] Bill Fefferman, Soumik Ghosh, Michael Gullans, Kohdai Kuroiwa, and Kunal Sharma. "Effect of non-unital noise on random circuit sampling" (2023). arXiv:2306.16659.

[11] John Preskill. "Quantum computing in the nisq era and beyond". Quantum **2**, 79 (2018).

[12] Bryan Eastin and Emanuel Knill. "Restrictions on transversal encoded quantum gate sets". Physical Review Letters **102**, 110502 (2009).

[13] Rawad Mezher, Joe Ghalbouni, Joseph Dgheim, and Damian Markham. "Fault-tolerant quantum speedup from constant depth quantum circuits". Physical Review Research **2**, 033444 (2020).

[14] Craig Gidney and Martin Ekerå. "How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits". Quantum **5**, 433 (2021).

[15] Sergey Bravyi, David Gosset, Robert Koenig, and Marco Tomamichel. "Quantum advantage with noisy shallow circuits". Nature Physics **16**, 1040–1045 (2020).

[16] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. "Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy". Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **467**, 459–472 (2011).

[17] Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. "Average-case complexity versus approximate simulation of commuting quantum computations". Physical Review Letters **117**, 080501 (2016).

[18] Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. "Achieving quantum supremacy with sparse and noisy commuting quantum computations". Quantum **1**, 8 (2017).

[19] Leslie Ann Goldberg and Heng Guo. "The complexity of approximating complex-valued ising and tutte partition functions". computational complexity **26**, 765–833 (2017).

[20] Keisuke Fujii and Tomoyuki Morimae. "Commuting quantum circuits and complexity of ising partition functions". New Journal of Physics **19**, 033003 (2017).

[21] Daniel Gottesman. "Fault-tolerant quantum computation with constant overhead" (2014). arXiv:1310.2984.

[22] Peter Høyer and Robert Špalek. "Quantum fan-out is powerful". Theory of computing **1**, 81–103 (2005).

[23] Dorit Aharonov and Michael Ben-Or. "Fault-tolerant quantum computation with constant error". In Proceedings of the twenty-ninth annual ACM symposium on Theory of computing. Pages 176–188. (1997).

[24] Emanuel Knill, Raymond Laflamme, and Wojciech H Zurek. "Resilient quantum computation". Science **279**, 342–345 (1998).

[25] Hector Bombin and Miguel-Angel Martin-Delgado. "Topological computation without braiding". Physical Review Letters **98**, 160502 (2007).

[26] Aleksander Kubica and Michael E Beverland. "Universal transversal gates with color codes: A simplified approach". Physical Review A **91**, 032330 (2015).

[27] A Robert Calderbank and Peter W Shor. "Good quantum error-correcting codes exist". Physical Review A **54**, 1098 (1996).

[28] Andrew M Steane. "Error correcting codes in quantum theory". Physical Review Letters **77**, 793 (1996).

[29] Clare Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. "Surface code quantum computing by lattice surgery". New Journal of Physics **14**, 123011 (2012).

[30] Daniel Litinski. "A game of surface codes: Large-scale quantum computing with lattice surgery". Quantum **3**, 128 (2019).

[31] Andrew J. Landahl and Ciaran Ryan-Anderson.

"Quantum computing by color-code lattice surgery" (2014). arXiv:1407.5103.

[32] Héctor Bombín. "Single-shot fault-tolerant quantum error correction". Physical Review X **5**, 031043 (2015).

[33] Héctor Bombín. "Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes". New Journal of Physics **17**, 083002 (2015).

[34] H Bombin and MA Martin-Delgado. "Exact topological quantum order in d= 3 and beyond: Branyons and brane-net condensates". Physical Review B **75**, 075103 (2007).

[35] Hector Bombin and Miguel Angel Martin-Delgado. "Topological quantum distillation". Physical Review Letters **97**, 180501 (2006).

[36] Christophe Vuillot. "Fault-tolerant quantum computation: Theory and practice". PhD thesis. TU Delft. (2020).

[37] AH Boerdijk. "Some remarks concerning close-packing of equal spheres". Philips Research Reports **7**, 303–313 (1952).

[38] HSM Coxeter and JM Wills. "Regular complex polytopes". Jahresbericht der Deutschen Mathematiker Vereinigung **96**, 2–2 (1994).

[39] Christopher Chamberland, Aleksander Kubica, Theodore J Yoder, and Guanyu Zhu. "Triangular color codes on trivalent graphs with flag qubits". New Journal of Physics **22**, 023019 (2020).

[40] Kaavya Sahay and Benjamin J Brown. "Decoder for the triangular color code by matching on a möbius strip". PRX Quantum **3**, 010310 (2022).

[41] Aleksander Kubica and Nicolas Delfosse. "Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders". Quantum **7**, 929 (2023).

[42] Michael E Beverland, Aleksander Kubica, and Krysta M Svore. "Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes". PRX Quantum **2**, 020341 (2021).

[43] Hector Bombin. "Transversal gates and error propagation in 3d topological codes" (2018). arXiv:1810.09575.

[44] Dan Browne, Elham Kashefi, and Simon Perdrix. "Computational depth complexity of measurement-based quantum computation". In Theory of Quantum Computation, Communication, and Cryptography: 5th Conference, TQC 2010, Leeds, UK, April 13-15, 2010, Revised Selected Papers 5. Pages 35–46. Springer (2011).

[45] Keisuke Fujii. "Noise threshold of quantum supremacy" (2016). arXiv:1610.03632.

[46] Dolev Bluvstein, Simon J Evered, Alexandra A Geim, Sophie H Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al. "Logical quantum processor based on reconfigurable atom arrays". Nature **626**, 58–65 (2024).

[47] Gregoire de Gliniasty, Rawad Mezher, and Damian Markham. In preparation.