Classical product code constructions for quantum Calderbank-Shor-Steane codes

Dimiter Ostrev, Davide Orsucci, Francisco Lázaro, and Balazs Matuz

Institute of Communications and Navigation, German Aerospace Center (DLR), 82234 Weßling, Germany

Several notions of code products are known in quantum error correction, such as hyper-graph products, homological products, lifted products, balanced products, to name a few. In this paper we introduce a new product code construction which is a natural generalisation of classical product codes to quantum codes: starting from a set of component Calderbank-Shor-Steane (CSS) codes, a larger CSS code is obtained where both X parity checks and Z parity checks are associated to classical product codes. We deduce several properties of product CSS codes from the properties of the component codes, including bounds to the code distance, and show that built-in redundancies in the parity checks result in so-called meta-checks which can be exploited to correct syndrome read-out errors. We then specialise to the case of single-parity-check (SPC) product codes which in the classical domain are a common choice for constructing product codes. Logical error rate simulations of a SPC 3-fold product CSS code having parameters $[[512, 174, 8]]$ are shown under both a maximum likelihood decoder for the erasure channel and belief propagation decoding for depolarising noise. We compare the results with other codes of comparable length and dimension, including a code from the family of asymptotically good Tanner codes. We observe that our reference product CSS code outperforms all the other examined codes.

1 Introduction

Quantum information processing offers promising new solutions to problems in computation, communication, and cryptography [45]. The presence of noise in all quantum operations, however, results in the degradation of the quantum information present in a quantum system and is one of the major obstacles to the development and widespread application of quantum technologies. Calderbank-Shor-Steane (CSS) codes [11, 57] and, more generally, stabilizer codes [23] can be employed to encode quantum information in a way that is resilient to errors affecting the physical qubits. Quantum error correction can then be applied to preserve the encoded quantum information in the presence of noise, with the ultimate goals of realising fully fault-tolerant quantum computation and robust encoders and decoders for quantum state transmission over noisy channels [30].

A quantum error correction approach that has gained prominence in the last few years is the use of quantum low-density parity-check (LDPC) codes [10], i.e., stabilizer codes having syndrome measurements of low weight. Low-weight stabilizers are considered favorable for fault-tolerance, since the quantum gates and measurements that implement them are also subject to noise. In the quantum information community LDPC is reserved to codes that furthermore feature a high encoding rate (i.e., a high ratio of logical qubits to physical qubits). In fact, many long-established stabilizer code families, such as the well-known toric code, only encode a constant number of logical qubits and thus their encoding rate is inversely proportional to the number of employed physical qubits. Recent works have showcased that, for realistic physical error rates and practically useful blocks sizes, LDPC codes achieve error correction performances similar to that of the toric code but with encoding rates around ten times higher [8]. This demonstrates that the search for the best-performing stabilizer codes is far from concluded and that hitherto unexplored code families may still yield improvements for certain applications.

In this work we introduce a novel natural method to construct LDPC-like CSS codes based on *classical product codes*. The construction is very structured and yet flexible enough to produce codes exhibiting favourable properties. Our endeavour started from the observation that classical product codes were historically one of the first efficient coding constructions [19]. In general, classical code construction methods cannot be directly applied in the quantum domain: the commutativity of all stabilizer generators must be enforced, which imposes additional constraints on the code structure. Prior to this work, the only CSS code based on classical product codes was proposed by Hivadi in [32]; however, Hivadi's construction was restricted to the two-fold product of single parity check (SPC) codes, yielding codes with minimum distance only equal to 4.

1.1 Comparison with other product code constructions

In the classical domain, product operations produce powerful codes from two (or more) weaker component codes. For two component codes, the code words can be represented as a rectangular array whose columns are code words of one code and rows are code words of the other, see Figure 1. A further construction, *tensor product codes*, was originally introduced in [60] with applications, e.g., to channels with burst errors. The code's parity-check matrix (PCM) is obtained by taking the tensor product of the PCMs of two codes. Tensor product codes were further generalized in [34] and extended to the construction of quantum error correcting codes in [20].

The first notion of product codes introduced in the quantum information literature has been that of *hyper-graph product* [58], which gave the first family of quantum LDPC codes having constant rate and minimum distance scaling as the square root of the block length. Later, it was realised that the hyper-graph product can be seen as a special case of the *homological product* of chain complexes [6], which can be leveraged, e.g., to construct quantum error correcting codes exhibiting single-shot error correction properties [52]. Asymptotically superior codes have been built using so-called *lifted products* [47] and *balanced products* [9] constructions. These techniques have culminated in the seminal result of constructing the first known families of asymptotically good quantum LDPC codes, i.e. having constant stabilizer weight, constant rate, and linearly growing minimum distance [48]. See also [10] for more details on quantum product code constructions.

These quantum product techniques are built to automatically satisfy the stabilizer commutativity condition and therefore are in spirit and in implementation very different from classical product codes. As far as the authors know, classical product codes remained unused in quantum error correction until Hivadi [32] showed that if H is the parity check matrix of the 2-fold product of the $[4s, 4s-1]$ single parity check code, then there exists a permutation π such that $H\pi H^T = 0$. This gives a $[[16s^2, 16s^2 - 16s + 2, 4]]$ quantum CSS code for any $s \in \mathbb{N}$, with the X and Z parity check matrices given by H and $H\pi^{\mathsf{T}}$ respectively.

1.2 Classical product construction for CSS codes

This work introduces a new product code construction which greatly generalizes the one given by Hivadi in [32]. We show that the construction does not apply uniquely to SPC codes, but actually to any family of codes which can be seen as tensor products of other codes. Equipped with this interpretation, for the first time we extend the construction to the D-fold product of codes, which allows us to obtain codes of arbitrarily large distances.

These codes do not technically satisfy the LDPC condition as the stabilizer weight grows unbounded under D-fold products for $D \to \infty$, but in practice the growth is rather slow; on the positive side, the encoding rate converges to 1, as opposed to merely reaching a constant value. Crucially, we show that in the intermediate and practically relevant regime with block sizes involving a few hundred qubits, these codes exhibit superior performance compared to some other codes stemming from true LDPC code families. Furthermore, our codes come with so-called meta-checks naturally embedded in their structure. They are beneficial as they can be used to correct errors that occur in the syndrome measurement readouts, which are always present in practice. This is a step towards single-shot decoding and simplifies the route towards full fault-tolerance. We also show that it is possible to construct codes having different X and Z PCMs, which can yield codes having asymmetric protection against X and Z errors and which might be appealing for certain error models.

Within our general construction, we identify in particular a $[[512, 174, 8]]$ code that is an especially promising candidate for quantum error correction in the near term. This code combines many desirable properties: high rate; high ability to correct errors; low weight of the syndrome measurements; parallelizable and hardware friendly syndrome measurement circuit; redundancy in the syndrome measurements that guarantees the correction of one syndrome read-out error (even under adversarial noise). We then numerically evaluate the practical performance of this code using the maximum likelihood decoder on the quantum erasure channel and Belief Propagation (BP) without any post-processing on the depolarising channel. This code achieves good performance in both cases, outperforming codes chosen from other well-known families. As far as the authors know, prior work on quantum error correction offers examples that have some of these desirable properties, but no prior example has all of them at the same time.

Finally, we believe that the codes from our product code families may be particularly good candidates to be experimentally implemented, in the near future, in quantum computers based on arrays of Rydberg atoms [4, 27, 2, 33, 62]. In fact, there exists a good synergy with this quantum computing platform, as it can realise the transport of an array of atoms and, therefore, can implement a fast fully-parallelised implementation of the syndrome measurements defining our CSS product codes [63, 5].

Paper organisation

Section 2 introduces required notation and preliminary information, followed by a description of the new code constructions in 3. A comparison with other codes from the literature is presented in Section 4, while the resilience to syndrome errors is discussed in Section 5, before concluding the paper in Section 6.

2 Preliminaries and notation

2.1 Classical codes

Let $\mathbb{F}_2$ denote the field with two elements and $\mathbb{F}_2^n$ denote the vector space over $\mathbb{F}_2$ of column vectors with n components, i.e., the set of bit strings of length n. Extending the notation further, let $\mathbb{F}_2^{m\times n}$ denote the space of matrices having m rows and n columns with elements from $\mathbb{F}_2$.

Let $\mathcal{C} \subseteq \mathbb{F}_2^n$ be a linear vector space which we interpret as the set of all valid code words. We say that $\mathcal{C}$ is a (binary linear) *code*. If the $\mathbb{F}_2$-linear dimension of $\mathcal{C}$ is k (there are 2^k total code words) we say that $\mathcal{C}$ encodes k bits in n bits, where n is the *length* of the code, k is the *dimension* of the code and $R = k/n \leq 1$ is the *rate* of the code. The *distance* d of a code $\mathcal{C}$ is the minimum Hamming distance between any pair of distinct code words and, since $\mathcal{C}$ is linear, we also have $d = \min_{w\in\mathcal{C}\setminus\{0\}} |w|$, where $|\cdot|$ denotes the Hamming weight. We summarise the parameters of a classical code $\mathcal{C}$ with an ordered triple $[n, k, d]$ giving the length, dimension, and minimum distance of the code. Sometimes we omit the code minimum distance and write only $[n, k]$.

A linear code can be defined as the kernel of a PCM $H \in \mathbb{F}_2^{m\times n}$, i.e., $\mathcal{C} = \{w \in \mathbb{F}_2^n : Hw = 0\}$. For a bit string $y \in \mathbb{F}_2^n$ we call $e = Hy$ the *syndrome* associated to y. A code can be also specified by a *generator matrix* $G \in \mathbb{F}_2^{n\times k}$ so that the code space is given by the $\mathbb{F}_2$-linear combination of the columns of G, that is $\mathcal{C} = Im(G)$, and the equation $HG = 0$ holds. A PCM H may have *linearly dependent rows*, thus $r = \text{rank}(H) \leq m$ and the code dimension is given as $k = n - r \geq n - m$. Product code constructions typically result in a certain number $\overline{m} = m - r$ of linearly dependent (i.e., redundant) checks, which we call *meta-checks*. Product code constructions can result in the introduction of new meta-checks, besides the ones that may be already present in the components codes.

Given H we denote w_c (w_r) the maximum Hamming weight of any given column (row) of H. We say that a family of PCMs is *sparse* when w_c and w_r are upper bounded by a constant or by a slowly growing function of the code size.

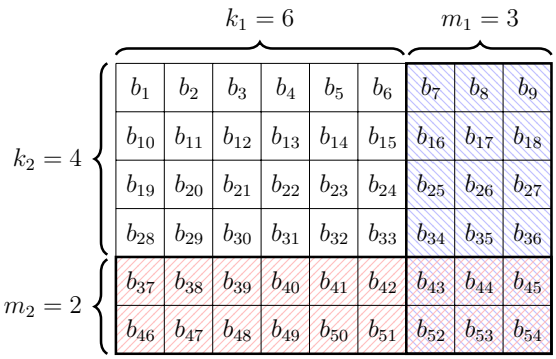

Figure 1: Array representation of the 54 bits of a code given by the product of a $[9, 6]$ and of a $[6, 4]$ systematic code. The final rate is $R = (6/9)(4/6) = 4/9$. The bits on white cells correspond to message bits, bits on hatched cells corresponds to parity-check bits, bits on doubly-hatched cells to checks-of-checks or *meta-checks*.

2.2 Classical product codes ($\mathcal{C}_\times$)

Consider two classical codes $\mathcal{C}_1 \subseteq \mathbb{F}_2^{n_1}$ and $\mathcal{C}_2 \subseteq \mathbb{F}_2^{n_2}$ having associated PCMs $H_1 \in \mathbb{F}_2^{m_1 \times n_1}$ and $H_2 \in \mathbb{F}_2^{m_2 \times n_2}$. The corresponding 2-fold product code $\mathcal{C}_\times$ is defined [19] as the code associated to the following PCM $H_\times$,

$$H_\times := \begin{pmatrix} H_1 \otimes I \\ I \otimes H_2 \end{pmatrix} \quad \in \mathbb{F}_2^{(n_1 m_2 + m_1 n_2) \times n_1 n_2}. \tag{1}$$

Above, I denotes identity matrices of appropriate dimensions, i.e., of dimension n_2 in the top block and n_1 in the bottom block, and $\otimes$ the tensor or Kronecker product. The resulting product code space $\mathcal{C}_\times$ is given by the tensor product of the component codes, i.e.,

$$\mathcal{C}_\times = \mathcal{C}_1 \otimes \mathcal{C}_2 \subseteq \mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2}. \tag{2}$$

The code dimension is $k_\times = k_1 k_2$, the maximum weight by row and columns can be directly deduced from (1), while the code minimum distance is given by $d_\times = d_1 d_2$ (see Appendix A for a proof of the minimum distance). We refer to [50, Chapter 5.7] for further details on product codes.

Remark. A 2-fold product code is visualized in Figure 1. The $n_\times = n_1 n_2$ bits are placed in a 2-dimensional array having n_1 columns and n_2 rows. Errors in the product code word can be detected by measuring the syndromes. For each row one can apply the PCM H_1 in order to extract the associated syndrome, which is equivalent to applying a PCM given by $H_1 \otimes I$ to all the n bits. Similarly, for each column, one can apply the PCM H_2 to extract the associated syndrome. For *systematic* codes (or more precisely systematic encoders), the code words $w \in \mathbb{F}_2^n$ are given by the concatenation of the message $x \in \mathbb{F}_2^k$ and a set of parity-check bits $c \in \mathbb{F}_2^{n-k}$, i.e., $w = (x, c)$. In the corresponding product code, errors are detected by checking whether the last $m_1 = n_1 - k_1$ bits of a row are valid linear combinations of the message bits. The same holds for the last $m_2 = n_2 - k_2$ bits in each column. There are $m_1 m_2$ bits that can be interpreted as checks-of-checks or *meta-checks* (see Figure 1). Meta-checks can be useful in quantum error correcting codes as discussed in Section 5.

The properties of 2-fold product codes are summarised in the table below, where the subscripts 1 and 2 refer to properties of $\mathcal{C}_1$ and of $\mathcal{C}_2$, respectively.

2.3 Classical binary tensor product codes ($\mathcal{C}_\otimes$)

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be codes having associated PCMs $H_1 \in \mathbb{F}_2^{m_1 \times n_1}$ and $H_2 \in \mathbb{F}_2^{m_2 \times n_2}$. The associated tensor product code $\mathcal{C}_\otimes$ is defined [60] as the code originating from the PCM

$$H_\otimes := H_1 \otimes H_2 \quad \in \mathbb{F}_2^{m_1 \times n_1} \otimes \mathbb{F}_2^{m_2 \times n_2} \cong \mathbb{F}_2^{m_1 m_2 \times n_1 n_2}. \tag{3}$$

Note that a tensor product code is *not* the tensor product of the component codes: we would obtain this result instead by taking the tensor product of the generator matrices of the codes. Using the

Properties of $\mathcal{C}_\times$	
length	$n_\times = n_1 n_2$
checks	$m_\times = n_1 m_2 + m_1 n_2$
meta-checks	$\overline{m}_\times = (m_1 - r_1)n_2 + n_1(m_2 - r_2) + r_1 r_2$
dimension	$k_\times = k_1 k_2$
distance	$d_\times = d_1 d_2$
row weight	$w_{\times,r} = \max(w_{1,r}, w_{2,r})$
column weight	$w_{\times,c} = w_{1,c} + w_{2,c}$

fact that a code $\mathcal{C}$ and its dual $\mathcal{C}^\perp := \{w \in \mathbb{F}_2^n \mid \langle v, w\rangle = 0,\ \forall v \in \mathcal{C}\}$ (where $\langle\cdot,\cdot\rangle$ denotes the $\mathbb{F}_2$ inner product) are obtained from each other by interchanging the roles of the parity check and of the generator matrix, we obtain

$$\mathcal{C}_\otimes := (\mathcal{C}_1^\perp \otimes \mathcal{C}_2^\perp)^\perp \quad \subseteq \ \mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2} \cong \mathbb{F}_2^{n_1 n_2}. \tag{4}$$

From the defining properties of the tensor product it follows that $r_\otimes = \operatorname{rank}(H_\otimes)$ is equal to $r_1 r_2 = \operatorname{rank}(H_1)\operatorname{rank}(H_2)$, hence the dimension of the code is $k_\otimes = n_1 n_2 - r_1 r_2 = k_1 n_2 + n_1 k_2 - k_1 k_2$. Moreover, one can immediately verify that the row and column weights of $H_\otimes$ are simply given by the product of the individual row and column weights of the component codes. Tensor product codes do not have meta-checks, unless some are already present in the component codes. Finally, the distance of the code is $d_\otimes = \min(d_1, d_2)$ which is discussed in Appendix A for completeness. See [61] for a short introduction to the topic and the extension to non-binary component codes.

In summary, tensor product constructions allow to create codes having higher rates than the component codes, at the cost of increasing the PCM weights and no improvement in the code distance. The properties of the tensor product code are summarised in the table below.

Properties of $\mathcal{C}_\otimes$	
length	$n_\otimes = n_1 n_2$
checks	$m_\otimes = m_1 m_2$
meta-checks	$\overline{m}_\otimes = m_1 m_2 - r_1 r_2$
dimension	$k_\otimes = n_1 n_2 - r_1 r_2$
distance	$d_\otimes = \min(d_1, d_2)$
row weight	$w_{\otimes,r} = w_{1,r} w_{2,r}$
column weight	$w_{\otimes,c} = w_{1,c} w_{2,c}$

2.4 Quantum CSS codes

Let $\mathbb{C}(\mathbb{F}_2^n) \cong \mathbb{C}^{2^n}$ be the complex Hilbert space associated to the space of n qubits, equipped with an orthonormal basis. Each of the 2^n basis vectors is indexed by an element of $\mathbb{F}_2^n$. The Pauli operators are $\mathcal{P} = \{I, X, Y, Z\}$ (with $Y = iXZ$) and the n-qubit Pauli group $\mathcal{P}_n$ acting on $\mathbb{C}(\mathbb{F}_2^n)$ is obtained by taking tensor products of n Pauli operators, together with a global phase $\alpha \in \{\pm 1, \pm i\}$. That is, an element of $\mathcal{P}_n$ has the form $P = \alpha P_1 \otimes \cdots \otimes P_n$, where $P_j \in \{I, X, Y, Z\}$ for all j. stabilizer codes are defined by a commutative subgroup $\mathcal{S} \subseteq \mathcal{P}_n$ such that $-I \notin \mathcal{S}$. The quantum code space is a $\mathbb{C}$-linear subspace $\mathcal{C} \subseteq \mathbb{C}(\mathbb{F}_2^n)$ defined as the set of quantum states that are stabilised by all the elements of $\mathcal{S}$, that is $\mathcal{C} := \{|\psi\rangle \in \mathbb{C}(\mathbb{F}_2^n) \mid s|\psi\rangle = |\psi\rangle,\ \forall s \in \mathcal{S}\}$. The stabilizer group has 2^r elements, can be generated by r independent elements, and defines a quantum code $\mathcal{C}$ having $\mathbb{C}$-dimension $2^{n-r} = 2^k$. In analogy with the classical case, we say that k is the code dimension, n the code length, and $R = k/n$ the code rate.

A CSS code is a stabilizer code where the stabilizer generators can be chosen to be either tensor products of I and X operators only, or tensor products of I and Z operators only. One can thus represent a CSS code with two binary PCMs: $H^z \in \mathbb{F}_2^{m^z \times n}$, which is associated to parity checks in the Z (computational) basis, and $H^x \in \mathbb{F}_2^{m^x \times n}$, which is associated to parity checks in the X (Hadamard) basis. For instance, the stabilizer elements $X \otimes X \otimes I$ corresponds to the row vector $(1, 1, 0)$ in H^x and $Z \otimes I \otimes Z$ to the row vector $(1, 0, 1)$ in H^z. The dimension of the resulting binary

code is given by $k = n - r^x - r^z = k^x + k^z - n$, where $r^x = \mathrm{rank}(H^x) \leq m^x$, $r^z = \mathrm{rank}(H^z) \leq m^z$, $k^x = \dim(\ker(H^x))$ and $k^z = \dim(\ker(H^z))$. Commutativity of the generators is equivalent to requiring

$$H^x (H^z)^\mathsf{T} = 0 \mod 2. \tag{5}$$

Neglecting the global phase, an error pattern $P = P_1 \otimes \cdots \otimes P_n \in \mathcal{P}_n$ is in one to one correspondence with an ordered pair of vectors (v^z, v^x), with $v^z, v^x \in \mathbb{F}_2^n$: we have $[v^z]_j = 1$ if $P_j \in \{Y, Z\}$ and $[v^z]_j = 0$ otherwise; similarly, we have $[v^x]_j = 1$ if $P_j \in \{X, Y\}$ and $[v^x]_j = 0$ otherwise. Note that X checks (specified via H^x) are sensitive to Z errors and that Z checks (specified via H^z) are sensitive to X errors.

For quantum CSS codes, one can distinguish different kinds of code minimum distance which we refer to as the *pure distance* δ and the *code distance* d. The pure distances for X and Z errors are given, respectively, by

$$\delta^x := \min\left\{|v^x| \;\middle|\; H^z v^x = 0, v^x \neq 0\right\} \tag{6}$$

$$\delta^z := \min\left\{|v^z| \;\middle|\; H^x v^z = 0, v^z \neq 0\right\} \tag{7}$$

and we simply call $\delta := \min(\delta^x, \delta^z)$ the pure distance. By contrast, the code distances for X and Z errors are obtained by considering only errors patterns that do not leave the code-subspace invariant and thus do result in a logical error:

$$d^x := \min\left\{|v^x| \;\middle|\; H^z v^x = 0, v^x \notin \mathrm{span}\big((H^x)^\mathsf{T}\big)\right\} \tag{8}$$

$$d^z := \min\left\{|v^z| \;\middle|\; H^x v^z = 0, v^z \notin \mathrm{span}\big((H^z)^\mathsf{T}\big)\right\} \tag{9}$$

We call $d := \min(d^x, d^z)$ the code distance. Note that we always have $d \geq \delta$ and d can be significantly larger than δ in certain cases[1]. We summarise the parameters of a quantum code $\mathcal{C}$ with the notation $[[n, k, d]]$ or, omitting the code distance, $[[n, k]]$.

3 Main constructions

We first present a method for obtaining asymmetric and symmetric 2-fold products of CSS codes. Then, we generalise the symmetric construction to D-fold products. Classical product and tensor product constructions are employed in such a way that the commutativity condition for CSS codes is automatically fulfilled.

3.1 Asymmetric 2-fold product construction of CSS codes ($\mathcal{C}_\ltimes$)

Definition 1. *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be CSS codes having, for $\ell \in \{1, 2\}$, PCMs associated to X and Z checks given by $H^x_\ell \in \mathbb{F}_2^{m^x_\ell \times n_\ell}$ and by $H^z_\ell \in \mathbb{F}_2^{m^z_\ell \times n_\ell}$. Because of the commutativity condition, these PCMs satisfy:*

$$H^x_\ell (H^z_\ell)^\mathsf{T} = 0 \qquad \ell \in \{1, 2\} \tag{10}$$

We define the asymmetric 2-fold product quantum CSS code $\mathcal{C}_\ltimes$ as the code associated to the following X and Z parity check matrices:

$$H^x_\ltimes := \begin{pmatrix} H^x_1 \otimes I \\ I \otimes H^x_2 \end{pmatrix} \tag{11}$$

$$H^z_\ltimes := \begin{pmatrix} H^z_1 \otimes H^z_2 \end{pmatrix}. \tag{12}$$

We have given the definition of $\mathcal{C}_\ltimes$ for two arbitrary CSS codes for sake of generality. We note that asymmetric product codes as here defined typically provide more protection against Z errors

[1] For instance, the surface code has constant pure distance, $\delta = 4$, while the code distance scales as the square root of the length $d = O(\sqrt{n})$.

than X errors. This may be beneficial for error channels where Z errors occur more frequently than X errors [1, 43, 29, 13]. Alternatively, the construction methods for the X and Z PCMs may be reversed. This CSS code features meta-checks in the PCM based on the product code construction but no meta-checks in PCM based on the tensor-product code construction. Finally, the commutativity condition $H^x_{\ltimes}(H^z_{\ltimes})^{\mathsf{T}} = 0$ can be verified using Eq. (10) and the multi-linearity of tensor products (which implies $0 \otimes M = 0$ for any matrix M).

Expressions for the pure distances can be easily derived from the distances of the underlying codes and the properties of the classical product and tensor product construction. The true code distances [see the definitions given in Eq. (8) and (9)] cannot be easily computed, but can only be lower bounded using $d^x \geq \delta^x$ and $d^z \geq \delta^z$. In contrast to the classical case, the distance of $\mathcal{C}_{\ltimes}$ can be smaller than the product of the distances of the component codes; for instance, $d^z_{\ltimes} < d^z_1 d^z_2$, as we show in Appendix A.

In the table below, we report the properties of $\mathcal{C}_{\ltimes}$ both for the general case (left column) and for the special case in which the two component codes are equal (right column). These properties can be straightforwardly derived from the classical product and tensor product constructions. The dimension of the CSS code $\mathcal{C}_{\ltimes}$ is most easily expressed in terms of the quantities $k^x_\ell = \dim(\ker(H^x_\ell)), r^z_\ell = \mathrm{rank}(H^z_\ell)$.

Properties of $\mathcal{C}_{\ltimes}$		
length	$n_{\ltimes} = n_1 n_2$	$n_{\ltimes} = n^2$
X checks Z checks	$m^x_{\ltimes} = m^x_1 n_2 + n_1 m^x_2$ $m^z_{\ltimes} = m^z_1 m^z_2$	$m^x_{\ltimes} = 2nm^x$ $m^z_{\ltimes} = (m^z)^2$
X meta-checks	$\overline{m}^x_{\ltimes} = (m^x_1 - r^x_1)n_2 + n_1(m^x_2 - r^x_2) + r^x_1 r^x_2$	$\overline{m}^x_{\ltimes} = 2n(m^x - r^x) + (r^x)^2$
dimension	$k_{\ltimes} = k^x_1 k^x_2 - r^z_1 r^z_2$	$k_{\ltimes} = (k^x)^2 - (r^z)^2$
Z pure distance X pure distance	$\delta^z_{\ltimes} = \delta^z_1 \delta^z_2$ $\delta^x_{\ltimes} = \min(\delta^x_1, \delta^x_2)$	$\delta^z_{\ltimes} = (\delta^z)^2$ $\delta^x_{\ltimes} = \delta^x$
X row weight Z row weight	$w^x_{\ltimes,r} = \max(w^x_{1,r}, w^x_{2,r})$ $w^z_{\ltimes,r} = w^z_{1,r} w^z_{2,r}$	$w^x_{\ltimes,r} = w^x_r$ $w^z_{\ltimes,r} = (w^z_r)^2$
X column weight Z column weight	$w^x_{\ltimes,r} = w^x_{1,c} + w^x_{2,c}$ $w^z_{\ltimes,c} = w^z_{1,c} w^z_{2,c}$	$w^x_{\ltimes,r} = 2w^x_c$ $w^z_{\ltimes,c} = (w^z_c)^2$

3.2 Symmetric 2-fold product construction of CSS codes ($\mathcal{C}_{\boxtimes}$)

Definition 2. *Let $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ be a set of four CSS codes such that, for each $\ell \in \{1,2,3,4\}$, the PCMs associated to X and Z checks are $H^x_\ell \in \mathbb{F}_2^{m^x_\ell \times n_\ell}$ and $H^z_\ell \in \mathbb{F}_2^{m^z_\ell \times n_\ell}$. Because of the commutativity condition, these PCMs must satisfy:*

$$H^x_\ell (H^z_\ell)^{\mathsf{T}} = 0 \qquad \ell \in \{1,2,3,4\} \tag{13}$$

We define the symmetric 2-fold product quantum CSS code $\mathcal{C}_{\boxtimes}$ as the code associated to the following X and Z parity check matrices:

$$H^x_{\boxtimes} := \begin{pmatrix} H^x_1 \otimes H^x_2 \otimes I \otimes I \\ I \otimes I \otimes H^x_3 \otimes H^x_4 \end{pmatrix} \tag{14}$$

$$H^z_{\boxtimes} := \begin{pmatrix} H^z_1 \otimes I \otimes H^z_3 \otimes I \\ I \otimes H^z_2 \otimes I \otimes H^z_4 \end{pmatrix}. \tag{15}$$

Note that both the X parity checks and Z checks are given by classical product codes. Specifically, these are the products of component codes associated to $H^x_1 \otimes H^x_2$ and $H^x_3 \otimes H^x_4$ (for $H^x_{\boxtimes}$) and $H^z_1 \otimes H^z_3$ and $H^z_2 \otimes H^z_4$ (for $H^z_{\boxtimes}$). This implies, in particular, that $H^x_{\boxtimes}$ and $H^z_{\boxtimes}$ are not full-rank. One can immediately verify the commutativity condition $H^x_{\boxtimes}(H^z_{\boxtimes})^{\mathsf{T}} = 0$ using Eq. (13).

The properties of the quantum CSS code $\mathcal{C}_{\boxtimes}$ are given in the table below, both for the general case and for the special case where the component codes are equal; the dimension of the code is computed via the equation $k_{\boxtimes} = n_{\boxtimes} - \mathrm{rank}(H^x_{\boxtimes}) - \mathrm{rank}(H^z_{\boxtimes})$ and expressed in terms of the quantities $r^x_\ell = \mathrm{rank}(H^x_\ell), r^z_\ell = \mathrm{rank}(H^z_\ell)$.

Properties of $\mathcal{C}_\boxtimes$		
length	$n_\boxtimes = n_1 n_2 n_3 n_4$	$n_\boxtimes = n^4$
X checks	$m^x_\boxtimes = m^x_1 m^x_2 n_3 n_4 + n_1 n_2 m^x_3 m^x_4$	$m^x_\boxtimes = 2n^2 (m^x)^2$
Z checks	$m^z_\boxtimes = m^z_1 n_2 m^z_3 n_4 + n_1 m^z_2 n_3 m^z_4$	$m^z_\boxtimes = 2n^2 (m^z)^2$
X meta-checks	$\overline{m}^x_\boxtimes = (m^x_1 m^x_2 - r^x_1 r^x_2) n_3 n_4 + n_1 n_2 (m^x_3 m^x_4 - r^x_3 r^x_4) + r^x_1 r^x_2 r^x_3 r^x_4$	$\overline{m}^x_\boxtimes = 2n^2((m^x)^2 - (r^x)^2) + (r^x)^4$
Z meta-checks	$\overline{m}^z_\boxtimes = (m^z_1 m^z_3 - r^z_1 r^z_3) n_2 n_4 + n_1 n_3 (m^z_2 m^z_4 - r^z_2 r^z_4) + r^z_1 r^z_2 r^z_3 r^z_4$	$\overline{m}^z_\boxtimes = 2n^2((m^z)^2 - (r^z)^2) + (r^z)^4$
dimension	$k_\boxtimes = \prod_\ell n_\ell + \prod_\ell r^x_\ell + \prod_\ell r^z_\ell - r^x_1 r^x_2 n_3 n_4 - n_1 n_2 r^x_3 r^x_4 - r^z_1 n_2 r^z_3 n_4 - n_1 r^z_2 n_3 r^z_4$	$k_\boxtimes = n^4 + (r^x)^4 + (r^z)^4 - 2n^2 (r^x)^2 - 2n^2 (r^z)^2$
Z pure distance	$\delta^z_\boxtimes = \min(\delta^z_1, \delta^z_2)\ \min(\delta^z_3, \delta^z_4)$	$\delta^z_\boxtimes = (\delta^z)^2$
X pure distance	$\delta^x_\boxtimes = \min(\delta^x_1, \delta^x_3)\ \min(\delta^x_2, \delta^x_4)$	$\delta^x_\boxtimes = (\delta^x)^2$
X row weight	$w^x_{\boxtimes,r} = \max(w^x_{1,r} w^x_{2,r}, w^x_{3,r} w^x_{4,r})$	$w^x_{\boxtimes,r} = (w^x_r)^2$
Z row weight	$w^z_{\boxtimes,r} = \max(w^z_{1,r} w^z_{3,r}, w^z_{2,r} w^z_{4,r})$	$w^z_{\boxtimes,r} = (w^z_r)^2$
X column weight	$w^x_{\boxtimes,r} = w^x_{1,c} w^x_{2,c} + w^x_{3,c} w^x_{4,c}$	$w^x_{\boxtimes,r} = 2(w^x_c)^2$
Z column weight	$w^z_{\boxtimes,c} = w^z_{1,c} w^z_{3,c} + w^z_{2,c} w^z_{4,c}$	$w^z_{\boxtimes,c} = 2(w^z_c)^2$

Remark. We note that there is some arbitrariness in our choice of how to present the PCMs of the product code. For instance, we may choose as Z PCM

$$H'^z_\boxtimes := \begin{pmatrix} H^z_1 \otimes P_2 \otimes P_3 \otimes H^z_4 \\ P_1 \otimes H^z_2 \otimes H^z_3 \otimes P_4 \end{pmatrix} \tag{16}$$

where P_1, P_2, P_3, P_4 are permutation matrices of appropriate size, while keeping the X PCM unchanged. However, this definition is equivalent to the one given in Eq. (14), up to a re-labelling of the qubits and of the stabilizer generators.

3.3 Symmetric **D**-fold product construction of CSS codes ($\mathcal{C}_{(\mathbf{D})}$)

The construction of symmetric product CSS codes can be easily generalised to higher dimensions, by taking the product of D^2 CSS codes. For instance, by using the PCMs of a set of 9 CSS codes one can construct the PCMs of a 3-fold product code as follows:

$$H^x_{(3)} := \begin{pmatrix} H^x_1 \otimes H^x_2 \otimes H^x_3 \otimes I \otimes I \otimes I \otimes I \otimes I \otimes I \\ I \otimes I \otimes I \otimes H^x_4 \otimes H^x_5 \otimes H^x_6 \otimes I \otimes I \otimes I \\ I \otimes I \otimes I \otimes I \otimes I \otimes I \otimes H^x_7 \otimes H^x_8 \otimes H^x_9 \end{pmatrix} \tag{17}$$

$$H^z_{(3)} := \begin{pmatrix} H^z_1 \otimes I \otimes I \otimes H^z_4 \otimes I \otimes I \otimes H^z_7 \otimes I \otimes I \\ I \otimes H^z_2 \otimes I \otimes I \otimes H^z_5 \otimes I \otimes I \otimes H^z_8 \otimes I \\ I \otimes I \otimes H^z_3 \otimes I \otimes I \otimes H^z_6 \otimes I \otimes I \otimes H^z_9 \end{pmatrix}. \tag{18}$$

It is easy to check that it is a CSS code and its properties can be derived similarly as done for the case of a 2-fold product. The general definition of the symmetric D-fold product construction of CSS codes is as follows.

Definition 3. *Let $\mathcal{C}_1, \ldots, \mathcal{C}_{D^2}$ be a set of D^2 CSS codes and, for each $\ell \in \{1, \ldots, D^2\}$, let $H^x_\ell \in \mathbb{F}_2^{m^x_\ell \times n_\ell}$ and $H^z_\ell \in \mathbb{F}_2^{m^z_\ell \times n_\ell}$ be the PCMs associated to X and Z checks, respectively. We define the symmetric D-fold product CSS code $\mathcal{C}_{(D)}$ as the code associated to the following X and Z parity check matrices:*

$$H^x_{(D)} := \left[\text{Stack}\right]_{j=0}^{D-1} \bigotimes_{\ell=1}^{D^2} \mathcal{H}^x_{\ell,j} \qquad \mathcal{H}^x_{\ell,j} = \begin{cases} H^x_\ell & \text{if } jD+1 \le \ell \le (j+1)D \\ I_{n_\ell} & \text{otherwise} \end{cases} \tag{19}$$

$$H^z_{(D)} := \left[\text{Stack}\right]_{j=0}^{D-1} \bigotimes_{\ell=1}^{D^2} \mathcal{H}^z_{\ell,j} \qquad \mathcal{H}^z_{\ell,j} = \begin{cases} H^z_\ell & \text{if } (\ell-1) \equiv j \mod D \\ I_{n_\ell} & \text{otherwise} \end{cases} \tag{20}$$

where Stack *denotes the operation of creating an $m \times n$ matrix ($m = \sum_j m_j$) by stacking $m_j \times n$ matrices one above the other.*

Other equivalent definitions of the code $\mathcal{C}_{(D)}$ could be given by choosing different orderings of the matrices H_ℓ^x and H_ℓ^z in the tensor products. The properties of this code can be obtained with similar procedures as in the previous examples. For instance, we can compute the number of meta-checks $\overline{m}_{(D)}^\alpha$, for $\alpha \in \{x, z\}$, by

$$\overline{m}_{(D)}^\alpha = m_{(D)}^\alpha - \left(n_{(D)} - k_{(D)}^\alpha\right) \qquad \alpha \in \{x, z\}\,, \tag{21}$$

where $k_{(D)}^x$ and $k_{(D)}^z$ are the classical code dimensions associated to the X and Z PCMs. These can be computed as a function of n_ℓ and r_ℓ^α using the definitions of $H_{(D)}^\alpha$ in (19) and (20), resulting in

$$k_{(D)}^x = \prod_{j=0}^{D-1} \left(\prod_{\ell=jD+1}^{(j+1)D} n_\ell - \prod_{\ell=jD+1}^{(j+1)D} r_\ell^x \right) \tag{22}$$

$$k_{(D)}^z = \prod_{j=0}^{D-1} \left(\prod_{\substack{(\ell-1)\equiv j \\ \mathrm{mod}\ D}} n_\ell - \prod_{\substack{(\ell-1)\equiv j \\ \mathrm{mod}\ D}} r_\ell^z \right). \tag{23}$$

We summarise the properties of this family of codes for the special case where all the component codes are equal (the expressions for the general case are lengthy and can be derived straightforwardly).

Properties of $\mathcal{C}_{(D)}$	
length	$n_{(D)} = n^{D^2}$
X checks	$m_{(D)}^x = D\, n^{D(D-1)}\, (m^x)^D$
Z checks	$m_{(D)}^z = D\, n^{D(D-1)}\, (m^z)^D$
X meta-checks	$\overline{m}_{(D)}^x = D\, n^{D(D-1)}\, (m^x)^D + (n^D - (r^x)^D)^D - n^{D^2}$
Z meta-checks	$\overline{m}_{(D)}^z = D\, n^{D(D-1)}\, (m^z)^D + (n^D - (r^z)^D)^D - n^{D^2}$
dimension	$k_{(D)} = (n^D - (r^x)^D)^D + (n^D - (r^z)^D)^D - n^{D^2}$
Z pure distance	$\delta_{(D)}^z = (\delta^z)^D$
X pure distance	$\delta_{(D)}^x = (\delta^x)^D$
X row weight	$w_{(D),r}^x = (w_r^x)^D$
Z row weight	$w_{(D),r}^z = (w_r^z)^D$
X column weight	$w_{(D),c}^x = D(w_c^x)^D$
Z column weight	$w_{(D),c}^z = D(w_c^z)^D$

3.4 Single-parity-check **D**-fold product CSS codes ($\mathcal{C}_{\mathbf{SPC(D)}}$)

We now specialise the symmetric D-fold product construction to the case where the component codes are all given by single-parity-check (SPC) codes. The choice of SPC codes as component CSS codes is motivated by the fact that the length of D-fold product codes grows very fast in D (as n^{D^2}), thus reasonably sized product codes can be obtained only for very small sizes of the component codes. A favourable choice is then $H_\ell^x = H_\ell^z = \begin{pmatrix} 1 & 1 \end{pmatrix}\ \forall \ell$, corresponding to the 2-qubit code having stabilizer generators $X \otimes X$ and $Z \otimes Z$. Note that this CSS code has zero encoding rate, since the only quantum state in its code space is the Bell state $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|0,0\rangle + |1,1\rangle)$. Nonetheless, the D-fold product CSS codes obtained from it have a positive encoding rate for all $D \geq 2$. We thus arrive at the following definition.

Definition 4. *Consider the following set of PCMs for the component codes $\mathcal{C}_1, \ldots, \mathcal{C}_{D^2}$:*

$$H_\ell^x = H_\ell^z := \begin{cases} (\underbrace{1 \ \cdots \ 1}_{2s}) & \text{for } \ell = (i-1)D + i,\ i = 1, \ldots, D \\ (1 \quad 1) & \text{otherwise} \end{cases} \tag{24}$$

and employ them to construct a D-fold product CSS code as described in Definition 3. We call the resulting code a single-parity-check D-fold product CSS code and denote it either $\mathcal{C}_{\mathrm{SPC}(D,s)}$ or more simply $\mathcal{C}_{\mathrm{SPC}(D)}$ in the case $s = 1$.

We remark here that in the special case $D = 2$, $\mathcal{C}_{\mathrm{SPC}(2,s)}$ is equivalent up to qubit permutations to the $[[16s^2, 16s^2 - 16s + 2, 4]]$ quantum CSS code introduced in [32, Theorem 2].

Note that the codes associated to X and Z checks are the same up to permutations of the columns of H^x and H^z, and are isomorphic to the D-fold product of the $[s2^D, s2^D - 1]$ SPC code. The free parameter $s \in \mathbb{N}$ may be used to increase the rate. The $\mathrm{SPC}(D, s)$ code family features a pure distance that grows asymptotically with D and, moreover, the distance is equal to the pure distance, $d = \delta = 2^D$, as we prove in Appendix A.

The properties of this code family are summarised in the table below.

Properties of $\mathcal{C}_{\mathrm{SPC}(D,s)}$	
length	$n = (s2^D)^D$
X and Z checks	$m^x = m^z = D(s2^D)^{D-1}$
X and Z meta-checks	$\overline{m}^x = \overline{m}^z = D(s2^D)^{D-1} + (s2^D - 1)^D - (s2^D)^D$
dimension	$k = 2(s2^D - 1)^D - (s2^D)^D$
X and Z distance	$\delta^x = \delta^z = d^x = d^z = 2^D$
X and Z row weight	$w_r^x = w_r^z = s2^D$
X and Z column weight	$w_c^x = w_c^z = D$

4 Comparison with previous works

In this section, we compare the performance of our codes against the performance of other code families.

We start by selecting the SPC(3) code, i.e., the CSS code from the family described in Section 3.4 with parameters $D = 3, s = 1$. More explicitly, this is the CSS code associated to the following 3-fold product codes:

$$H^x_{\mathrm{SPC}(3)} = \begin{pmatrix} h\otimes h\otimes h\otimes I\otimes I\otimes I\otimes I\otimes I\otimes I \\ I\otimes I\otimes I\otimes h\otimes h\otimes h\otimes I\otimes I\otimes I \\ I\otimes I\otimes I\otimes I\otimes I\otimes I\otimes h\otimes h\otimes h \end{pmatrix} \tag{25}$$

$$H^z_{\mathrm{SPC}(3)} = \begin{pmatrix} h\otimes I\otimes I\otimes h\otimes I\otimes I\otimes h\otimes I\otimes I \\ I\otimes h\otimes I\otimes I\otimes h\otimes I\otimes I\otimes h\otimes I \\ I\otimes I\otimes h\otimes I\otimes I\otimes h\otimes I\otimes I\otimes h \end{pmatrix} \tag{26}$$

where $h = \begin{pmatrix} 1 & 1 \end{pmatrix}$ and $I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. This results in a $[[512, 174, 8]]$ CSS code where all row weights are 8 and all column weights are 3; each matrix has 23 linearly dependent rows (i.e., meta-checks).

4.1 Previous code families participating in the comparison

Next, we search for codes from other families having similar parameters to the SPC(3) code.

Bicycle codes [42] were one of the first families of quantum LDPC codes to be proposed. A bicycle code is obtained by first randomly selecting a $n/2 \times n/2$ cyclic matrix C, which in our case has row weight 8. Next, the matrix $H_0 = \begin{pmatrix} C & C^{\mathsf{T}} \end{pmatrix}$ is constructed. Then, we proceed to remove $k/2$ rows from H_0 in order to obtain a matrix H_1 with $(n-k)/2$ rows. Finally, the PCM of the bicycle code is obtained by setting $H^x = H^z = H_1$. Different strategies can be followed in order to remove rows from H_0 to obtain H_1. In our case, we follow an approach that aims at obtaining a matrix H_1 whose column weights are as uniform as possible. In particular, we follow a greedy approach in which rows are removed one by one, and the row to be removed is always selected so

as to keep the column weights as uniform as possible. A bicycle code with parameters [[512,174,2]] is then obtained. The block length and rate are chosen to be equal to those of the SPC(3) code, but for this high encoding rate we could not find codes having minimum distance larger than 2.

Hypergraph product codes [58] were introduced as a generalization of the surface code; thus the surface code can be viewed as a hypergraph product of a certain representation of the repetition code with itself. More generally, [58, Theorem 1] gives the following construction: if H is a full rank $m \times n$ PCM of a $[n, k, d]$ classical linear code, then the quantum CSS code with parity check matrices $H^x = \begin{pmatrix} H \otimes I & I \otimes H^{\mathsf{T}} \end{pmatrix}$ and $H^z = \begin{pmatrix} I \otimes H & H^{\mathsf{T}} \otimes I \end{pmatrix}$ has parameters $[[n^2+m^2, k^2, d]]$. For the present comparison, we want a hypergraph product code with parameters as close as possible to $[[512, 174]]$; the closest that one can get within this family is for $n = 21, m = 8, k = 13$, giving a $[[505, 169]]$ quantum CSS code. It is known that the distance of a $[21, 13]$ classical linear code is at most 4 [26]. Using a computer search, we found a sparse $[21, 13, 4]$ classical code that leads to a $[[505, 169, 4]]$ hypergraph product code. The rows of the parity check matrices have weights between 9 and 13. Most of the columns have weight 3, but some have higher weights up to 10.

Quantum Tanner codes [37] were introduced as an asymptotically good family of quantum LDPC codes. Recently, [28, 38, 18, 39] have discussed efficient decoders for these codes. There are slight variations in the construction of quantum Tanner codes between the various references. For the present comparison, we adopt the approach of [38, 39] which avoids the total non-conjugacy constraint.

A quantum Tanner code is described by seven components: a finite group G, two subsets $A, B \subset G$ of size Δ each that are closed under inversion, two full rank $m \times \Delta$ parity check matrices $H_A, H_B^\perp$ and two full rank $(\Delta - m) \times m$ parity check matrices $H_A^\perp, H_B$, where $H_A, H_A^\perp$ are orthogonal, and so are $H_B, H_B^\perp$, and where $m \leq \Delta/2$ is a parameter that determines a lower bound $(1 - 2m/\Delta)^2$ on the rate.

The group G and its two subsets A, B define an incidence structure with vertices, edges and squares called a left-right Cayley complex [17]. In the approach of [38, 39], the vertices are

$$V = \big\{(g, i, j) \;\big|\; g \in G, i \in \{0, 1\}, j \in \{0, 1\}\big\} \tag{27}$$

the edges are $E = E_A \cup E_B$ with

$$E_A = \big\{((g, i, 0), (ag, i, 1)) \;\big|\; g \in G, a \in A, i \in \{0, 1\}\big\} \tag{28}$$

$$E_B = \big\{((g, 0, j), (gb, 1, j)) \;\big|\; g \in G, b \in B, j \in \{0, 1\}\big\} \tag{29}$$

and the squares are

$$Q = \big\{((g, 0, 0), (ag, 0, 1), (agb, 1, 1), (gb, 1, 0)) \;\big|\; g \in G, a \in A, b \in B\big\}. \tag{30}$$

Qubits are placed on the squares and constraints are placed on the vertices. If $i + j \mod 2 = 0$, vertex (g, i, j) imposes the x constraints with PCM $H_A \otimes H_B$ on the connected squares/qubits, and if $i + j \mod 2 = 1$, then vertex (g, i, j) imposes the z constraints with PCM $H_A^\perp \otimes H_B^\perp$ on the connected squares/qubits.

The resulting quantum CSS code has $\Delta^2|G|$ qubits and $2m(\Delta - m)|G|$ rows in each of its parity check matrices H^x, H^z. The quality of the code is determined by the spectral expansion properties of the left Cayley graph $\mathrm{Cay}_L(A, G)$ and the right Cayley graph $\mathrm{Cay}_R(G, B)$ (i.e. the second largest magnitude of an eigenvalue of the adjacency matrix should be as small as possible), as well as the distance and robustness properties of the classical linear codes with parity check matrices $H_A \otimes H_B$ and $H_A^\perp \otimes H_B^\perp$.

For the present comparison, we require a quantum Tanner code having length and dimension close to $[[512, 174]]$. Observing that $\Delta^3 \leq \Delta^2|G|$, we see that the only possible choices are those with $\Delta \leq 8$; out of these, only the combination $\Delta = 5, m = 1$ gives rate around $174/512$. Further, given $m = 1$, we want the minimum distance of $\ker(H_A \otimes H_B)$, $\ker(H_A^\perp \otimes H_B^\perp)$ to be more than 1;

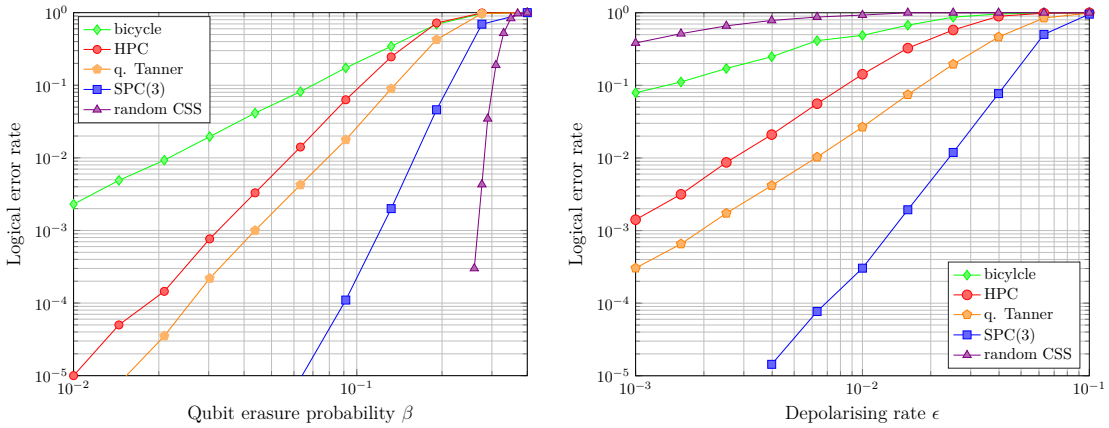

Figure 2: Monte Carlo simulation of the logical error rates of 5 different CSS codes. Left: maximum likelihood decoding under erasure channel ($\mathcal{E}_\beta$). Right: BP decoding under depolarising channel ($\mathcal{D}_\epsilon$). See main text for details.

this determines $H_A = H_B^\perp = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ and therefore a natural choice is

$$H_A^\perp = H_B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{31}$$

The last step is to choose G, A, B. Requiring a block size around 512 fixes $|G| = 20$ or 21. A computer search determines that the best spectral expansion properties are given by the choices

$$G = \langle s, t | s^4 = t^5 = 1, ts = st^2 \rangle \tag{32}$$
$$A = \{1, s, s^3, t^2 s^2, t^3 s^2\} \tag{33}$$
$$B = \{1, ts^3, t^2 s, t^2 s^2, t^4 s^2\}. \tag{34}$$

The resulting PCMs H^x, H^z have rank 156 each, so we obtain a $[[500, 188]]$ quantum Tanner code. The row weights of the parity check matrices are all 10, and the column weights are either 2 or 4. By exhaustive search of low-weight code words we have determined that the minimum distance of this code is 4.

4.2 Decoding performance with the quantum erasure channel

The first error model we consider is the quantum erasure channel $\mathcal{E}_\beta$, which acts on a single qubit state ρ as

$$\mathcal{E}_\beta(\rho) = (1-\beta)\,\rho \otimes |0\rangle\langle 0| + \beta\,\frac{I}{2} \otimes |1\rangle\langle 1| \tag{35}$$

where β is a parameter of the channel specifying the probability of erasure (see [30, Chapter 5] for an introduction to quantum channels). The rightmost qubit is interpreted as a "flag" which can be read-out deterministically (since $|0\rangle$ and $|1\rangle$ are orthogonal states) and which heralds whether an erasure has occurred.

The main reason for considering the erasure channel is that it has been observed in [16, Section III] that for any stabilizer code on the erasure channel it is possible to efficiently perform maximum likelihood decoding: it is sufficient to find any error pattern, e.g. by Gaussian elimination, supported on the erased positions that produces the same syndrome[2]. This allows to test the performance

[2]Technically, the decoder outputs an error pattern e from the most probable coset of errors, that is, maximising the sum of the probabilities of all errors that differ from e by a stabilizer of the code.

of the codes in an unbiased manner, that is, in a way that is decoupled from the performance of efficient but sub-optimal decoding algorithms, such as belief propagation. We note, however, that erasure errors do occur in certain quantum computation models or for concatenated codes where the inner decoder can signal a decoding failure.

The probability of decoding error as a function of the probability of erasure is shown in Figure 2 (left) for the bicycle, hypergraph product code (HPC), quantum Tanner code, our construction, and for a randomly chosen (not sparse) $[[512, 174]]$ CSS code drawn from the probability distribution described in [46]. In the present case, this means that a random 512×512 invertible matrix A is chosen, then the first 169 rows of A and the second 169 rows of $(A^{-1})^{\mathsf{T}}$ are chosen as H^x and H^z, respectively.

Our product code showcases the lowest logical error rate, compared to the other practically implementable codes, over the entire range of erasure probability β that we have considered. This is mainly due to the fact that our code features a larger minimum distance: the minimum distances d of the bicycle, HPC, quantum Tanner and our SPC(3) construction are 2, 4, 4 and 8, respectively. This results in an logical error rate scaling approximately as β^d. Furthermore, we have performed an exhaustive search to find the multiplicity of the minimum weight errors for the bicycle, HPC, quantum Tanner codes. The bicycle code has 28 weight-2 X errors and, being self-dual, each for each X error there exists a Y error and a Z error having the same support. The HPC has 861 weight-4 X errors, 861 weight-4 Z errors and no weight-4 Y errors. The quantum Tanner code has 250 weight-4 X errors, 250 weight-4 Z errors and no weight-4 Y errors. We remark that the lower number of weight-4 errors in the quantum Tanner code compared to the HPC results in a roughly constant offset in the logarithmic plots of their logical error rates.

4.3 Decoding performance with the depolarising channel

The second error model we consider is the depolarising channel $\mathcal{D}_\epsilon$, which acts on a single qubit state ρ as

$$\mathcal{D}_\epsilon(\rho) = \left(1-\epsilon\right)\rho + \frac{\epsilon}{3}\left(X\rho X + Y\rho Y + Z\rho Z\right) = \left(1 - \frac{4\epsilon}{3}\right)\rho + \frac{4\epsilon}{3}\frac{I}{2} \tag{36}$$

where ϵ is a parameter of the channel. The first expression clarifies that X, Y and Z Pauli errors occur with equal probability, while the second expression is based on the identity $\frac{1}{4}(\rho + X\rho X + Y\rho Y + Z\rho Z) = I/2$ and shows that the completely depolarising channel is obtained for $\epsilon = 3/4$.

On the depolarising channel, there is no known efficiently computable maximum likelihood decoder. Instead, we use a quaternary belief propagation decoder: the messages are indexed by values in $\mathbb{F}_4$ or, equivalently, by a Pauli matrix. The advantage of using this decoder, instead of using two independent binary belief propagation decoders for X and Z checks is that it keeps into account the correlation between X and Z errors in the depolarising channel. Quaternary BP decoding was first introduced in [51] (and see [3] for a review of decoding algorithms for quantum LDPC codes). For completeness we describe our implementation of the quaternary belief propagation decoder in Appendix B. Further decoding improvement may be achieved using post-processing techniques such as ordered-statistics decoding [47, 56, 59] and stabilizer inactivation [15], or by exploiting the soft syndrome information that is accessible in some quantum computing hardware [21, 49, 53].

The probability of decoding error as a function of the depolarising rate of the channel is shown in Figure 2 (right) for the bicycle code, the hypergraph product code, the quantum Tanner code, and the $[[512, 174]]$ code from our 3-fold product construction; we also include the randomly chosen dense $[[512, 174]]$ CSS code. We observe that our reference SPC(3) code surpasses all other considered codes, judging by the BP error correction performance, over the considered range of depolarising noise intensities (ϵ).

The decoding performance for the random CSS code is particularly poor. This is to be expected, since the code is non-sparse and therefore unsuitable for decoding via BP. In general, the maximum-likelihood decoding of quantum codes on the depolarising channel is a very hard problem (technically, #P-hard [35]).

The decoding performance of the four sparse codes can be heuristically explained as follows. In general, it is known that the performance of the BP decoder is very sensitive to the structure

of the Tanner graph. However, in the present case, it can be plainly seen that the two plots in Figure 2 are qualitatively the same for the sparse codes. Therefore, it is reasonable to conjecture that the dominant factor explaining the superior performance of the SPC code on the depolarizing channel is the same as on the erasure channel: its larger minimum distance.

5 Meta-checks and correction of syndrome readout errors

In this section, we show how meta-checks can be used to mitigate the effect of syndrome read-out errors. In particular, the SPC$(3, s)$ code has a distance with respect to syndrome errors that is equal to 3, allowing us to detect and locate one syndrome read-out error, which is a first step in the direction of fault-tolerance [24, 25].

5.1 Meta-check matrix

Quantum read-out measurements are prone to errors, hence the syndrome outcomes may be faulty, i.e., incorrect. Fault-tolerant error correcting codes showcase the possibility of detecting and locating errors even in the face of syndrome read-out errors. The syndrome measurements are physically implemented via quantum circuits acting on both data and ancillary qubits and measuring out the ancillary qubits alone and thus correlated errors are typically present. Circuit-level noise simulations are therefore necessary to perform an accurate evaluation of the performance of a quantum error-correcting code [22]. Nonetheless, a simplified but informative analysis can be obtained using a so-called *phenomenological error* model: data qubit errors and syndrome read-out errors are drawn independently at random, possibly with different error probabilities.

We now explain briefly how the presence of redundant checks helps to correct errors that occur during the syndrome read-out measurements. We start considering a generic CSS code and then we specialise to the SPC$(3, s)$ code. Let $H \in \mathbb{F}_2^{m\times n}$ be a matrix with rank $r < m$ (so that $m - r$ checks are redundant), H can be either the X or the Z PCM of the CSS code. Consider the subspace $\mathcal{S} = \mathrm{span}(H) \subset \mathbb{F}_2^m$ of dimension r. If the device performing the syndrome measurement was perfect, the measured syndrome would be an element of $\mathcal{S}$. However, when the syndrome measurement is faulty, the measured syndrome is an element $s' \in \mathbb{F}_2^m$ that is not necessarily in $\mathcal{S}$; we may write $s' = He + e_s$, where e_s represents a syndrome readout error. In that case, we regard $\mathcal{S}$ as a linear error-correcting code and try to correct the faulty s' to the subspace $\mathcal{S} = \mathrm{span}(H)$. The faulty syndrome decoding works independently from any data-qubit error that may be present since the meta-checks, by construction, are insensitive to them.

The matrix H plays the role of a PCM for the quantum CSS code and of a generator matrix for $\mathcal{S} = \mathrm{span}(H)$. We now want to find another matrix M, such that $\ker(M) = \mathrm{span}(H)$; we call M a *meta-check* matrix. A systematic way for obtaining M is the following: use Gaussian elimination to find an invertible $M'' \in \mathbb{F}_2^{m\times m}$ such that

$$\underbrace{\left(\begin{array}{c} M' \\ \hline M \end{array}\right)}_{=M''} \left(\begin{array}{c} \\ H \\ \\ \end{array}\right) = \left(\begin{array}{c} E \\ \hline 0 \end{array}\right) \tag{37}$$

where $M' \in \mathbb{F}_2^{r\times m}$ and $M \in \mathbb{F}_2^{(m-r)\times m}$ are full rank matrices, and where $E \in \mathbb{F}_2^{r\times n}$ is the reduced row echelon form of H. Then we have $\ker(M) = \mathrm{span}(H)$ as needed. Therefore $Ms' = M(He + e_s) = Me_s$ and M is insensitive to the data-qubit error e, as claimed.

The minimum distance d_M of the meta-check matrix M can be upper bounded as $d_M \leq w_c^{\min}$, where $w_c^{\min}$ is the minimum column weight of H. To see this, let H^i a minimum-weight column of H; by assumption $H^i \in \mathrm{span}(H) = \ker(M)$, meaning that H^i is a code word of $\mathcal{S}$ having weight $w_c^{\min}$, thus proving the claim. The proof that SPC$(3, s)$ codes have meta-check distance $d_M = w_c^{\min} = 3$ can be found in Appendix A, where furthermore we provide an explicit construction for a meta-check matrix M.

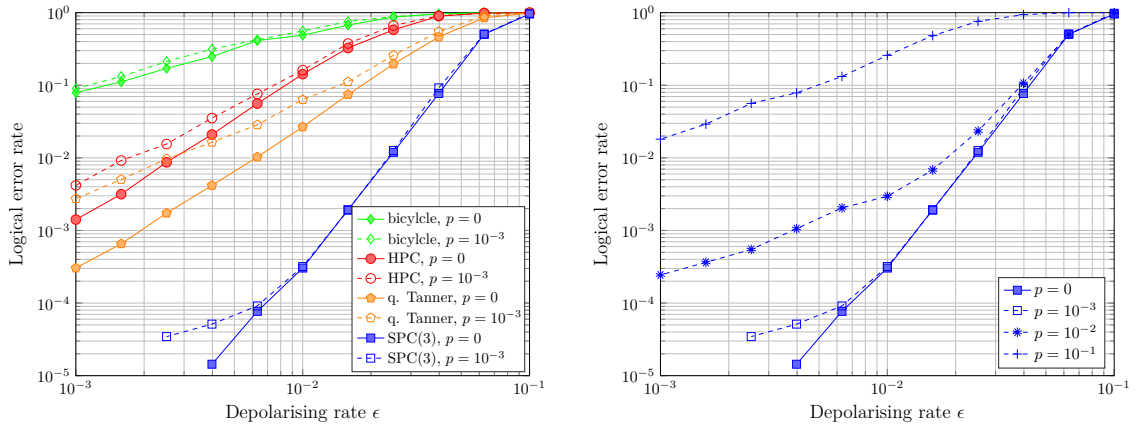

Figure 3: Monte Carlo simulations of the logical error rates under BP decoding under a depolarizing channel with syndrome measurement errors. Left: all codes are subject to either no syndrome readout error (solid lines) or to a syndrome readout flip with probability $p = 10^{-3}$ (dashed lines). Right: performance of the SPC(3) code for different values of the syndrome readout error probability p.

5.2 Extended parity-check matrix

Note that syndrome read-out errors e_s do not directly affect the data qubits, so there is no physical correction to be applied. Nonetheless, a meta-syndrome $\sigma \in \mathbb{F}_2^{m-r}$ containing information about the location of the faulty syndrome can be obtained as $\sigma = Ms$ and then forwarded to the decoder that is tasked with decoding the quantum CSS code associated to H, thus yielding a two-step decoding. However, as noted in [31, Section VII] the decoding performance typically improves if one uses a single stage BP decoder directly. To this end we define *extended PCM, extended error* and *extended syndrome* $\widetilde{H}, \widetilde{e}, \widetilde{s}$ via

$$\underbrace{\left(\begin{array}{c|c} H & I_m \\ \hline 0 & M \end{array}\right)}_{=\widetilde{H}} \underbrace{\left(\begin{array}{c} e \\ \hline e_s \end{array}\right)}_{=\widetilde{e}} = \left(\begin{array}{c} He + e_s \\ \hline Me_s \end{array}\right) = \underbrace{\left(\begin{array}{c} s' \\ \hline \sigma \end{array}\right)}_{=\widetilde{s}} \tag{38}$$

so that $\widetilde{s} = \widetilde{H}\widetilde{e}$ holds. The extended PCM $\widetilde{H}$ can be interpreted as adding one hidden variable for each syndrome which may flip the result of the measurements and adding a meta-check code M to correct the effect of these hidden variables. The single-stage BP decoder takes as input the extended syndrome $\widetilde{s} \in \mathbb{F}_2^{m+(m-r)}$ and aims at reconstructing the $\widetilde{e} \in \mathbb{F}_2^{n+m}$ employing $\widetilde{H}$ to specify the bipartite graph for the massage passing algorithm. It is also possible to extend the quaternary BP that simultaneously decodes the X and Z components of the CSS code to include the meta-checks, see Appendix B for further details.

A standard way for obtaining resilience to readout errors is to repeat all the measurements k times. This can be interpreted as protecting the readout data via a repetition code; a code with PCM H would be modified to $H_{(k)} = 1_k \otimes H$, where $1_k = (1, \cdots, 1)^\mathsf{T} \in \mathbb{F}_2^k$, while the associated meta-check matrix is $M = H_{k\text{-rep}} \otimes I$, where $H_{k\text{-rep}}$ is the PCM of the k-repetition code. Doing so results in a large overhead in the number of measurements, scaling linearly in the number of repetitions. Product codes can increase the meta-check distance much more efficiently. For instance, the SPC(3) code requires a minimum of 338 measurements to be stabilized, while the associated PCM defined in Eqs. (25) and (26) achieve a meta-check distance equal to 3 while requiring only 384 measurements, an increase of about 13.6% compared to the minimum.

5.3 Decoding performance including syndrome readout errors

We have benchmarked the resilience to syndrome readout errors of the SPC(3) code under extended quaternary BP decoding. The employed phenomenological error model is given by the depolarising channel of Eq. (36) acting on the data qubits, while each syndrome is subject to readout errors that may flip the outcome with probability p (i.e., it is a classical binary symmetric channel). The

BP messages are passed along the edges of a bipartite graph implicitly defined by $\widetilde{H}$ as in Eq. (38), where the meta-check matrix M is the one provided in Appendix A. The results are presented in Figure 3.

As a preliminary step, we have verified that by fixing $p = 0$ and running the quaternary BP decoding over the graphs defined by $\widetilde{H}$ and H, respectively, yield results that are essentially indistiguishable (blue solid line in Figure 2, right, and in Figure 3, left). This shows that BP is neither impaired nor enhanced by the extension of the decoding graph for the SPC(3) code.

We have then assessed the decoding performance of the codes described in Section 4 in presence of readout errors, with the exclusion of the random CSS code, which cannot be decoded via BP. The simulation results are presented as dashed lines in Figure 3, left, for $p = 10^{-3}$. This readout fidelity is comparable with what can be obtained, e.g., in state of the art quantum computers based on Rydberg atom arrays [5]. As expected, the codes that do not feature meta-checks, are significantly affected by readout errors. In contrast, the SPC(3) logical error rate is essentially unaffected for depolarisation rates $\epsilon > 6 \cdot 10^{-4}$, below which it converges to an error floor of around $3 \cdot 10^{-5}$.

When increasing p to higher values, $p = 10^{-2}$ or $p = 10^{-1}$, the performance of SPC(3) with quaternary BP decoding significantly degrades, see Figure 3, right. This means that repeating the syndrome measurements a few times may be required in order to increase the meta-check distance and thus counter the effect of readout errors.

6 Conclusions and outlook

In this work, we have introduced new methods for constructing families of CSS codes associated to classical product codes. Quantum CSS codes are challenging to construct (compared to classical ones) since a commutativity condition between X and Z checks must be satisfied, as given in Eq. (5). Our first construction for CSS product codes is asymmetrical, in the sense that X parity checks are constructed from a classical product code and Z parity checks from a classical tensor product code, with the goal of automatically satisfying Eq. (5). A second construction is more symmetrical, having both X and Z checks associated to classical product codes, while the component codes are in the form of classical tensor product codes; this second construction showcases an increased (pure) distance against both kinds of Pauli errors compared to the component CSS codes it is constructed from. The third construction is a multi-dimensional generalisation of the second one, allowing to obtain D-fold products of component CSS codes.

An advantage of our classically-inspired product code construction is that extensive parallelisation can be achieved in measuring the parity checks. Using the array representation of a 2-fold product code (see Figure 1) it is clear that the parity checks of the component codes can be applied in parallel over all the rows of the array and, in a second step, measured in parallel over all the columns of the array. The same is true for CSS product codes and, if the qubits are physically arranged in a 2-dimensional array, as in the quantum computing architectures based on Rydberg atom [4, 27, 2, 33, 62, 63, 5], this can result in simplified hardware implementations. A caveat is that the columns of the PCM associated to Z are permuted compared to the PCM associated to Z checks: this means that the Z syndrome measurements act on qubits that are no longer aligned along single rows or columns of the 2-dimensional array.

Product codes yield good performances when using SPC codes as the component codes (albeit other choices are possible [14, 36, 44]). The heuristic motivation is that SPC codes have high encoding rate (only one parity bit of overhead) but small distance ($d = 2$); the product construction allows to increase the distance at the expense of decreasing the rate and yields codes having good properties for intermediate code sizes. These good properties are retained also by the corresponding quantum CSS codes. For instance, we obtain a 3-fold product SPC code that can encode 174 logical qubits in 512 physical qubits ($R = 174/512 \simeq 0.34$) and having distance $d = 8$. This code has good performances both for erasure channels and for depolarising noise under BP decoding; for instance, the second best code in our comparison is a $[[500, 188]]$ quantum Tanner code, which exhibits logical error rates that are around two orders of magnitude higher for realistic values of the depolarising noise.

Given the promising results of these product codes, we aim to further investigate their proper-

ties. Open questions regard whether there are other good choices for the component codes, besides SPC codes. This could be especially true for the asymmetric product construction, where the fact that only two component CSS codes are needed (instead of the four required in the symmetric construction) gives leeway in the choice of larger component CSS codes. Another interesting question is the possibility of achieving better practical results via code concatenation, e.g., using a surface code for low-level error correction and a product code as a high-level code to further boost the fidelity of the encoded logical qubits [13, 40, 54, 12].

Regarding the code minimum distance d, it would be interesting to find better lower bounds, since bounds based on the pure distance δ are in general not tight (but we have shown that for the special case of SPC product codes we have $d = \delta$). The fault-tolerance properties of these codes could be further investigated, simulating the performance of decoders under realistic circuit-level noise [22]. Finally, methods for implementing fault-tolerant Clifford gates [55] and magic-state distillation [7, 41] within these code families should be ascertained for realising universal quantum computation with CSS product codes.

Acknowledgements

We acknowledge Dr. Francesco Guatieri for optimising the code for the exhaustive search of the minimum weight errors. This project was funded within the QuantERA II Programme that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 101017733, through the project EQUIP.

A Omitted proofs of code distance properties

A.1 Classical product code minimum distance: $d_\times = d_1 d_2$

The codes we consider are $\mathbb{F}_2$-linear, hence the code minimum distance is equal to the minimum Hamming weight of any non-zero code-word. Consider then minimum distance code-words $x_1 \in \mathcal{C}_1$ and $x_2 \in \mathcal{C}_2$, and denote their Hamming weights d_1 and d_2, respectively. The non-zero code-word $x = x_1 \otimes x_2 \in \mathcal{C}_1 \times \mathcal{C}_2$ has weight $d_1 d_2$, thus the product code has distance $d_\times \leq d_1 d_2$. Conversely, all non-zero code-words must have at least d_2 rows that are not identically zero and each of these rows must have at least d_1 ones in it. Therefore, $d_\times \geq d_1 d_2$.

A.2 Classical tensor product code minimum distance: $d_\otimes = \min(d_1, d_2)$

Given any minimum weight words $x \in \mathcal{C}_1, y \in \mathcal{C}_2$ (i.e., $|x| = d_1$, $|y| = d_2$, $H_1 x = 0$, $H_2 y = 0$) and given any two unit vectors e_i and e_j (having a one in position i and in position j, respectively, and zeros elsewhere) we have $|x \otimes \mathrm{e}_i| = d_1$, $|\mathrm{e}_j \otimes y| = d_2$, and moreover

$$H_\otimes(x \otimes \mathrm{e}_i) = H_1 x \otimes H_2 \mathrm{e}_i = 0 \otimes H_2 \mathrm{e}_i = 0\,. \tag{39}$$

Analogously we have $H_\otimes(\mathrm{e}_j \otimes y) = 0$, which proves $d_\otimes \leq \min(d_1, d_2)$. Conversely, assume by contradiction that $w \in \mathcal{C}_\otimes$ is a non-zero code-word having Hamming weight $d < \min(d_1, d_2)$. We can write w as the sum of d unit vectors, $w = \sum_{\alpha=1}^{d} \mathrm{e}_{i(\alpha)} \otimes \mathrm{e}_{j(\alpha)}$. The vectors $x_\alpha := H_1 \mathrm{e}_{i(\alpha)}$ are linearly independent, since otherwise we could find a word of $\mathcal{C}_1$ with Hamming weight less or equal to $d < d_1$. Similarly, $y_\alpha := H_2 \mathrm{e}_{j(\alpha)}$ are linearly independent. By the definition of tensor product, $\{x_\alpha \otimes y_\beta\}_{\alpha,\beta}$ are linearly independent, which implies

$$0 \neq \sum_{\alpha=1}^{d} x_\alpha \otimes y_\alpha = \sum_{\alpha=1}^{d} H_1 \mathrm{e}_{i(\alpha)} \otimes H_2 \mathrm{e}_{j(\alpha)} = (H_1 \otimes H_2) w \tag{40}$$

contra the hypothesis that $w \in \mathcal{C}_\otimes$, hence we must have $d_\otimes \geq \min(d_1, d_2)$.

A.3 Examples of asymmetric CSS product codes having $d^z_{\ltimes} < d^z_1 d^z_2$

Let us construct an asymmetric product CSS codes where the two base codes are both the 9-qubit Shor code. The X and Z PCMs of the Shor code are given by, respectively,

$$H^x := \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad H^z := \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{41}$$

and note that it is a CSS code having distances $d^x = 3, d^z = 3$ and pure distances $\delta^x = 3, \delta^z = 2$; for instance, $Z^{\otimes 2} \otimes I^{\otimes 7}$ (corresponding to the vector $\mathrm{e}_1 + \mathrm{e}_2 \in \mathbb{F}_2^9$) is a weight 2 undetectable Pauli operator which acts as the identity on the logical subspace. We define the asymmetric product CSS code $\mathcal{C}^{\mathrm{Shor}}_{\ltimes}$ via the PCMs:

$$H^x_{\ltimes} = \begin{pmatrix} H^x \otimes I \\ I \otimes H^x \end{pmatrix} \qquad H^z_{\ltimes} = \begin{pmatrix} H^z \otimes H^z \end{pmatrix}. \tag{42}$$

The weight-6 Pauli-Z operator associated to the vector

$$v^z := (\mathrm{e}_1 + \mathrm{e}_2) \otimes (\mathrm{e}_1 + \mathrm{e}_4 + \mathrm{e}_7) \tag{43}$$

is as undetectable logical error for $\mathcal{C}^{\mathrm{Shor}}_{\ltimes}$. First, notice that $H^x(\mathrm{e}_1 + \mathrm{e}_2) = H^x(\mathrm{e}_1 + \mathrm{e}_4 + \mathrm{e}_7) = 0$, hence v^z is undetected by X parity checks, $H^x_{\ltimes} v^z = 0$. Second, let $u = e_1 \otimes \sum_{i=1}^{9} e_i$, and note that $H^z_{\ltimes} u = 0$ while $u^{\mathsf{T}} v^z = 1$; therefore v^z is not a stabiliser, $v^z \notin \mathrm{span}\big((H^z_{\ltimes})^{\mathsf{T}}\big)$. This shows that $d^x_{\ltimes} \leq 6 < 9 = d^x_1 d^x_2$.

The pattern can be extended as follows. Let

$$H_D = \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{(D-1)\times D} \qquad 1_D = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{F}_2^D \tag{44}$$

The generalised code $\mathrm{Shor}(D)$ has parity check matrices $H^x = H_D \otimes 1_D^{\mathsf{T}}$, $H^z = I_D \otimes H_D$ and distances $d^x = d^z = \delta^x = D$ and $\delta^z = 2$. We then consider the asymmetric product CSS code $\mathcal{C}^{\mathrm{Shor}(D)}_{\ltimes}$ using again Eq. (42). Let us consider the weight-$2D$ Pauli-Z operator associated to the vector $v^z := e_1 \otimes (e_1 + e_2) \otimes 1_D \otimes e_1$ as well as the vector $u = e_1 \otimes e_1 \otimes e_1 \otimes 1_D$. From $H^x_{\ltimes} v^z = 0$, $H^z_{\ltimes} u = 0$, $u^{\mathsf{T}} v^z = 1$, we see that v^z is an undetected logical error, therefore the distance of $\mathcal{C}^{\mathrm{Shor}(D)}_{\ltimes}$ is upper bounded as $d^z_{\ltimes} \leq 2D$.

A.4 Single-parity-check D-fold product CSS code distance: $d_{\mathrm{SPC}(D,s)} = 2^D$

We have shown in the main text that $\delta_{\mathrm{SPC}(D,s)} = 2^D$, thus $d_{\mathrm{SPC}(D,s)} \geq 2^D$. Here we show that there exist logical errors of weight 2^D, thus establishing $d_{\mathrm{SPC}(D,s)} \leq 2^D$.

We claim that the following vector corresponds to undetectable logical error of weight 2^D for both $H^x_{\mathrm{SPC}(D)}$ and $H^z_{\mathrm{SPC}(D)}$:

$$w_{(D)} = \bigotimes_{\ell=1}^{D^2} a_\ell \qquad a_\ell := \begin{cases} u := (1, 1, \underbrace{0 \cdots 0}_{2(s-1)})^{\mathsf{T}} & \text{for } \ell = (i-1)D + i,\ i = 1, \ldots D \\ \mathrm{e}_1 := (1, 0)^{\mathsf{T}} & \text{otherwise} \end{cases} \tag{45}$$

For example, $w_{(2)} = u \otimes \mathrm{e}_1 \otimes \mathrm{e}_1 \otimes u$ and $w_{(3)} = u \otimes \mathrm{e}_1 \otimes \mathrm{e}_1 \otimes \mathrm{e}_1 \otimes u \otimes \mathrm{e}_1 \otimes \mathrm{e}_1 \otimes \mathrm{e}_1 \otimes u$. The vector $w_{(D)}$ is a tensor product of D copies of u and of vectors of Hamming weight one, hence it has weight 2^D. Note that exactly one PCM in the tensor products given in Eqs. (19) and (20) is

in a position ℓ of the form $(i-1)D+i$ and therefore we obtain $H^x_{\mathrm{SPC}(D)} w_{(D)} = H^z_{\mathrm{SPC}(D)} w_{(D)} = 0$. To see that $w_{(D)}$ is not in the image of $(H^x_{\mathrm{SPC}(D)})^\mathsf{T}$ or of $(H^z_{\mathrm{SPC}(D)})^\mathsf{T}$ consider the vector

$$v_{(D)} = \bigotimes_{\ell=1}^{D^2} b_\ell \qquad b_\ell := \begin{cases} \mathrm{e}_1 := (1,0,\underbrace{0\ \cdots\ 0}_{2(s-1)})^\mathsf{T} & \text{if for } \ell = (i-1)D+i,\ i=1,\dots D \\ u' := (1,1)^\mathsf{T} & \text{otherwise} \end{cases} \tag{46}$$

Now, note that $H^x_{\mathrm{SPC}(D)} v_{(D)} = H^z_{\mathrm{SPC}(D)} v_{(D)} = 0$ while $w^\mathsf{T}_{(D)} v_{(D)} = 1$.

A.5 Meta-check distance for SPC(3, s) codes: $d_M = 3$

We now show that for a SPC$(3,s)$ code the meta-check distance is exactly $d_M = w_c = 3$ and therefore it can be used to detect and locate one faulty syndrome. Up to a permutation of the columns, the PCMs for both X and Z checks of the SPC$(3,s)$ have the form

$$H_{\mathrm{SPC}(3,s)} = \begin{pmatrix} H_s \otimes I \otimes I \\ I \otimes H_s \otimes I \\ I \otimes I \otimes H_s \end{pmatrix} \tag{47}$$

where $H_s = (1 \cdots 1)$ is the PCM associated to a $[8s, 8s-1]$ SPC code. An associated meta-check matrix can be constructed as the following block matrix:

$$M_{\mathrm{SPC}(3,s)} = \begin{pmatrix} 0 & I \otimes 1 \otimes H_s & I \otimes H_s \otimes 1 \\ 1 \otimes I \otimes H_s & 0 & H_s \otimes I \otimes 1 \\ 1 \otimes H_s \otimes I & H_s \otimes 1 \otimes I & 0 \end{pmatrix} \tag{48}$$

and one can easily verify that $M_{\mathrm{SPC}(3,s)} H_{\mathrm{SPC}(3,s)} = 0$. The corresponding meta-check code $\mathcal{S}$ has distance three if and only if all weight one errors $e \in \mathbb{F}_2^{192s^2}$ result in distinct nonzero meta-syndrome outcomes $M_{\mathrm{SPC}(3,s)} e \in \mathbb{F}_2^{24s}$. To show it, notice that for any unit vectors $\mathrm{e}_i, \mathrm{e}_j \in \mathbb{F}_2^{8s}$ we have $(I \otimes H_s \otimes 1)(\mathrm{e}_i \otimes \mathrm{e}_j \otimes 1) = \mathrm{e}_i \otimes 1 \otimes 1$ and similarly for other permutations of the tensor components; note that $\mathrm{e}_i \otimes 1 \otimes 1$ is different from zero. It is then possible to determine from the position of the non-zero entries of $M_{\mathrm{SPC}(3,s)} e$ whether the non-zero entry of the error e is located in the first, second or third block of $64s^2$ bits. Supposing, without loss of generality, that the non-zero entry of e is located in one of the first $64s^2$ bits means that $e = 1 \otimes \mathrm{e}_i \otimes \mathrm{e}_j$ for some values i, j. Then the two meta-syndrome outcomes $(1 \otimes I \otimes H_s) e = 1 \otimes \mathrm{e}_i \otimes 1$ and $(1 \otimes H_s \otimes I) e = 1 \otimes 1 \otimes \mathrm{e}_j$ contain sufficient information to determine the values i, j and thus locate the position of the syndrome error.

B Quaternary belief propagation decoding of stabiliser codes

We describe an efficient implementation of quaternary BP decoding used in this work. This is a message passing algorithm where information about the likelihood of a Pauli error $\mathcal{P} = \{I, X, Y, Z\}$ on a qubit is exchanged along the edges of a graph.

B.1 Preliminaries and notation

Stabiliser code representation Let $\mathcal{S} = \{S_1, \dots, S_m\}$ be a set of generators of a stabiliser group for a quantum linear code, where each generator is a n-qubit Pauli operator, $S_i \in \mathcal{P}_n$. The set of generators $\mathcal{S}$ can be specified via a matrix $\mathrm{H} \in \mathbb{F}_4^{m \times n}$ where each matrix element $\mathrm{H}_{i,j}$ is in $\mathbb{F}_4 = \{\mathrm{I}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$ and in one-to-one correspondence to a Pauli operator $\mathcal{P} = \{I, X, Y, Z\}$. The rows of the matrix H correspond to different stabilisers while the columns correspond to different physical qubits. The matrix H is sufficient to fully specify a stabiliser code, in contrast with the notation used in the rest of the paper where two binary matrices (H^x and H^z) are needed to define a CSS code.

Commutation and anti-commutation relations We define an inner product $\langle \cdot, \cdot \rangle : \mathbb{F}_4 \times \mathbb{F}_4 \to \mathbb{F}_2$ as

$\langle \cdot, \cdot \rangle$	I	X	Y	Z
I	0	0	0	0
X	0	0	1	1
Y	0	1	0	1
Z	0	1	1	0

i.e., the inner product $\langle \cdot, \cdot \rangle$ is equal to 0 if the corresponding Pauli matrices commute and is equal to 1 if the Pauli matrices anti-commute. The commutativity of the stabiliser generators then translates to

$$\sum_{j=1}^{n} \langle \mathrm{H}_{i,j}, \mathrm{H}_{k,j} \rangle = 0 \mod 2 \qquad \forall i, k \in \{1, \ldots, m\} \tag{49}$$

which generalises the CSS commutativity condition in Eq. (5). Similarly, given a n-qubit Pauli error $E \in \mathcal{P}_n$ having associated a vector $\mathrm{E} \in \mathbb{F}_4^n$, the corresponding syndrome outcomes $s \in \mathbb{F}_2^m$ are obtained as

$$s_i = \sum_{j=1}^{n} \langle \mathrm{H}_{i,j}, \mathrm{E}_j \rangle \mod 2\,. \tag{50}$$

Let the set $\mathcal{Z}_{i,j}$ and $(\overline{\mathcal{Z}}_{i,j})$ be

$$\mathcal{Z}_{i,j} := \left\{ \mathrm{P} \in \mathbb{F}_4 \,\middle|\, \langle \mathrm{H}_{i,j}, \mathrm{P} \rangle = 0 \right\} \qquad \overline{\mathcal{Z}}_{i,j} := \mathbb{F}_4 \setminus \mathcal{Z}_{i,j}\,. \tag{51}$$

In other words, $\mathcal{Z}_{i,j}$ $(\overline{\mathcal{Z}}_{i,j})$ corresponds to the set of Pauli operators that stabiliser S_i commutes with (anti-commutes with) at qubit j.

Quantum Tanner graph representation The code's stabiliser matrix H can be represented as a quantum Tanner graph, i.e., a labeled bipartite graph having two distinct sets of vertices, called *variable nodes* (VNs) and *check nodes* (CNs), as well as a set of $\mathbb{F}_4$-labeled edges connecting vertices of different types. VNs are associated to physical qubits and CNs to stabiliser generators. An edge connects a VN $\mathtt{v}_j$ to a CN $\mathtt{c}_i$ if and only if $\mathrm{H}_{i,j} \in \mathbb{F}_4 \setminus \{\mathrm{I}\}$. The edge label corresponds to $\mathrm{H}_{i,j}$ (for $\mathrm{H}_{i,j} \neq \mathrm{I}$). The set of neighbours of a VN $\mathtt{v}_j$ (or of a CN $\mathtt{c}_i$) is denoted by $\mathcal{N}(\mathtt{v}_j)$ $(\mathcal{N}(\mathtt{c}_i))$. See Figure 4 for a graphical representation.

Message passing Decoding can be cast as passing messages between VNs and CNs along the edges of the Tanner graph. In a successive manner, all CNs send messages to all the neighbouring VNs, and all VNs exchange messages with all the neighbouring CNs until a stopping criterion is reached. CNs and VNs process incoming messages according to certain rules. Messages are (for instance) quaternary probability vectors representing an estimate of the probability of a given value of E_j. Observe that the inner product in Eq. (50) yields the same binary value for two distinct values of E_j, i.e., $\langle \mathrm{H}_{i,j}, \mathrm{E}_j \rangle$ is zero for $\mathrm{E}_j \in \mathcal{Z}_{i,j}$ and one for $\mathrm{E}_j \in \overline{\mathcal{Z}}_{i,j}$. Hence, we can add up the respective probabilities and apply the CN processing on a binary probability vector, rather than on a quaternary probability vector.

Log-domain decoding For numerical stability message passing decoders are often implemented using ratios of logarithmic probabilities, referred to as log-likelihood ratios (LLR). For probability estimates p_0 and $p_1 = 1 - p_0$ of binary outcomes we define the associated LLR as

$$L = \log \frac{p_0}{p_1}. \tag{52}$$

For quaternary outcomes with probabilities $p_\mathrm{I}, p_\mathrm{X}, p_\mathrm{Y}, p_\mathrm{Z}$ we define an LLR vector as

$$\mathcal{L} = \left(0, \log \frac{p_\mathrm{X}}{p_\mathrm{I}}, \log \frac{p_\mathrm{Y}}{p_\mathrm{I}}, \log \frac{p_\mathrm{Z}}{p_\mathrm{I}}\right). \tag{53}$$

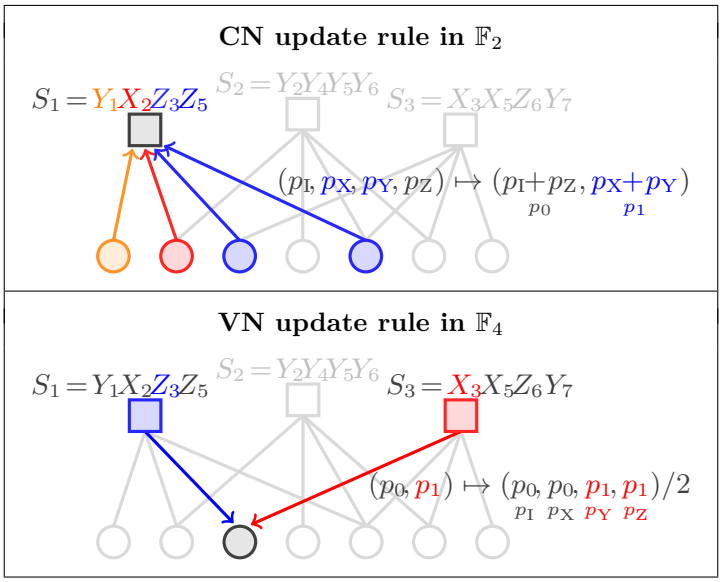

Figure 4: Example binary/quaternary inter-conversion in message passing for a quantum Tanner graph having 7 VNs (qubits) and 3 CNs (stabiliser generators). Edges transmitting messages to one CN c_j (top) and to one VN v_i (bottom) are highlighted and the edges are labelled (coloured) according to the Pauli operator measured by the stabiliser S_i on qubit j.

Conversions between binary and quaternary message passing Consider a stabiliser S_i and a qubit j. Quaternary messages from v_j to c_i are converted to binary messages by computing p_1 as the sum of the probabilities associated to operators in $\overline{\mathcal{Z}}_{i,j}$ and then $p_0 = 1 - p_1$. Binary messages from c_i to v_j are converted to quaternary messages by equally splitting the probability p_1 among the operators in $\overline{\mathcal{Z}}_{i,j}$ and equally splitting p_0 among the other two Pauli operators (i.e., those in $\mathcal{Z}_{i,j}$). See Figure 4 for an example. For conversion of quaternary to binary messages, we will make use of the soft-max operator ($\max^*$) which is defined as

$$\max^*(a,b) := \log(\exp(a) + \exp(b)) \tag{54}$$
$$= \max(a,b) + \log\left(1 + \exp\left(-|a-b|\right)\right). \tag{55}$$

The first expression makes it obvious that $\max^*$ is symmetric and associative, the second expression is the one that shall be used for a numerically stable implementation.

B.2 Update rules

Let us focus on iteration number t. An iteration consists in messages sent from CNs to VNs followed by messages sent from VNs to CNs.

Let the message received by CN c_i from the VN v_j be $L^{(t-1)}_{\mathsf{v}_j \to \mathsf{c}_i}$. The message that the CN c_i sends to VN v_j, $L^{(t)}_{\mathsf{c}_i \to \mathsf{v}_j}$, is obtained by using the standard update rule for binary message passing decoders (embedding the syndrome value s_i):

$$L^{(t)}_{\mathsf{c}_i \to \mathsf{v}_j} = (-1)^{s_i} \, 2 \tanh^{-1}\left(\prod_{j' : \mathsf{v}_{j'} \in \mathcal{N}(\mathsf{c}_i) \setminus \mathsf{v}_j} \tanh\left(\frac{1}{2} L^{(t-1)}_{\mathsf{v}_{j'} \to \mathsf{c}_i} \right) \right). \tag{56}$$

Remark. There are a manifold of approximations of the check node (CN) update rule in the literature which permit a hardware-friendly implementation.

The LLR $L^{(t)}_{\mathsf{c}_i \to \mathsf{v}_j}$ is converted into a quaternary LLR $l^{(t)}_{\mathsf{c}_i \to \mathsf{v}_j}$ for further processing at the VNs:

$$\mathcal{L}^{(t,\mathrm{P})}_{\mathsf{c}_i \to \mathsf{v}_j} = \begin{cases} 0 & \text{for } \mathrm{P} \in \mathcal{Z}_{i,j} \\ -L^{(t)}_{\mathsf{c}_i \to \mathsf{v}_j} & \text{for } \mathrm{P} \in \overline{\mathcal{Z}}_{i,j} \end{cases} \tag{57}$$

where $\mathrm{P} \in \mathbb{F}_4$ indexes the values of the message. The VN update operation is the standard one for quaternary message passing and yields the error probability estimates

$$\mathcal{L}^{(t)}_{\mathtt{v}_j \to \mathtt{c}_i} = \overline{\mathcal{L}}_j + \sum_{i' \colon \mathtt{c}_{i'} \in \mathcal{N}(\mathtt{v}_j) \setminus \mathtt{c}_i} \mathcal{L}^{(t)}_{\mathtt{c}_{i'} \to \mathtt{v}_j} \tag{58}$$

where $\overline{\mathcal{L}}_j$ is the channel message (a.k.a., *prior* LLR) and is a P-indexed vector which depends on the channel model. For the depolarising noise in Eq. (36) we have

$$\overline{\mathcal{L}}_j^{(\mathrm{P})} = \begin{cases} 0 & \text{for } \mathrm{P} = \mathrm{I} \\ \log\left(\frac{\epsilon}{3(1-\epsilon)}\right) & \text{for } \mathrm{P} \in \mathbb{F}_4 \setminus \{\mathrm{I}\} \end{cases}. \tag{59}$$

Remark. The evaluation of Eq. (58) can be simplified by avoiding to construct a quaternary LLR vector $\mathcal{L}^{(t)}_{\mathtt{c}_{i'} \to \mathtt{v}_j}$. It is sufficient to add its two non-zero elements to the respective elements of $\mathcal{L}^{(t)}_{\mathtt{v}_j \to \mathtt{c}_i}$ in (58).

Finally, we convert $\mathcal{L}^{(t)}_{\mathtt{v}_j \to \mathtt{c}_i}$ to a scalar LLR $L^{(t)}_{\mathtt{v}_j \to \mathtt{c}_i}$ that is given as input the respective CN,

$$L^{(t)}_{\mathtt{v}_j \to \mathtt{c}_i} = \max_{\mathrm{P} \in \mathcal{Z}_{i,j}}{}^{*} (\mathcal{L}^{(t,\mathrm{P})}_{\mathtt{v}_j \to \mathtt{c}_i}) - \max_{\mathrm{P} \in \overline{\mathcal{Z}}_{i,j}}{}^{*} (\mathcal{L}^{(t,\mathrm{P})}_{\mathtt{v}_j \to \mathtt{c}_i}). \tag{60}$$

B.3 Initialization and final decision

We give the message from VN $\mathtt{v}_j$ to CN $\mathtt{c}_i$ at the beginning of the very first decoding iteration. We have

$$\mathcal{L}^{(0)}_{\mathtt{v}_j \to \mathtt{c}_i} = \overline{\mathcal{L}}_j \tag{61}$$

and from (60) we obtain

$$L^{(0)}_{\mathtt{v}_j \to \mathtt{c}_i} = \log\left(\frac{3-2\epsilon}{2\epsilon}\right). \tag{62}$$

A hard decision $\hat{\mathrm{E}}_j$ on the value of E_j is made according to

$$\hat{\mathrm{E}}_j = \operatorname{argmax}\left(\overline{\mathcal{L}}_j + \sum_{i \colon \mathtt{c}_i \in \mathcal{N}(\mathtt{v}_j)} \mathcal{L}_{\mathtt{c}_i \to \mathtt{v}_j}\right). \tag{63}$$

Recall that for convenience the elements of quaternary LLR are indexed by $\mathrm{P} \in \mathbb{F}_4$. The decoder returns this hard decision after a maximum number of decoding iterations is reached or when the estimated error vector $\hat{\mathrm{E}}$ satisfies all parity-checks.

B.4 Inclusion of meta-checks

The Tanner graph can be expanded to include readout errors and meta-checks. In particular, a new VN is added for each CN, which represents the classical readout error channel. These new VN are therefore labelled as "binary" and will process the received messages differently from the "quaternary" representing the Pauli errors on the qubits. Furthermore, these new variable nodes are connected to a new set of CNs, corresponding to the meta-checks. The extended Tanner graph is associated to a labelled adjacency matrix of the form

$$\widetilde{\mathrm{H}} = \left(\begin{array}{c|c} \mathrm{H} & I \\ \hline 0 & M \end{array}\right) \tag{64}$$

where the leftmost columns (comprising H and 0) have entries in $\mathbb{F}_4$ and the rightmost columns (comprising I and M) have entries in $\mathbb{F}_2$. The BP decoder then becomes hybrid, where either quaternary or binary messages are passed, depending on the VN type ($\mathbb{F}_4$-type or $\mathbb{F}_2$-type).

The BP update rules are modified as follows. The CN to VN messages are always binary and are directly given by Eq. (56), for the extended set of VN, CN and neighbourhoods $\mathcal{N}(\mathtt{c}_i)$. The channel message for each new variable node $\mathtt{v}_j$ is given by $\overline{L}_j = \log \frac{p}{1-p}$, where p is the probability of flipping the readout value. Thus, the VN to CN are computed in two different ways, depending on the VN type:

$$L^{(t)}_{\mathtt{v}_j \to \mathtt{c}_i} = \begin{cases} \max^*_{\mathrm{P} \in \mathcal{Z}_{i,j}} (\mathcal{L}^{(t,\mathrm{P})}_{\mathtt{v}_j \to \mathtt{c}_i}) - \max^*_{\mathrm{P} \in \overline{\mathcal{Z}}_{i,j}} (\mathcal{L}^{(t,\mathrm{P})}_{\mathtt{v}_j \to \mathtt{c}_i}) & \mathbb{F}_4\text{-type VN} \\ \overline{L}_j + \sum_{i' \colon \mathtt{c}_{i'} \in \mathcal{N}(\mathtt{v}_j) \setminus \mathtt{c}_i} L^{(t)}_{\mathtt{c}_{i'} \to \mathtt{v}_j} & \mathbb{F}_2\text{-type VN} \end{cases} \tag{65}$$

with the notation given in Section B.2. The hard decision for the data-qubit errors, Eq. (63), is unmodified, while the estimate $\hat{e}_s$ of the readout errors e_s is obtained via the hard decision

$$(\hat{e}_s)_j = \operatorname{argmax} \left(\overline{L}_j + \sum_{i : \mathtt{c}_i \in \mathcal{N}(\mathtt{v}_j)} L_{\mathtt{c}_i \to \mathtt{v}_j} \right). \tag{66}$$

References

[1] Panos Aliferis and John Preskill. Fault-tolerant quantum computation against biased noise. *Phys. Rev. A*, 78:052331, Nov 2008. DOI: 10.1103/PhysRevA.78.052331. URL https://link.aps.org/doi/10.1103/PhysRevA.78.052331.

[2] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. DOI: 10.1038/s41586-019-1666-5.

[3] Zunaira Babar, Panagiotis Botsinis, Dimitrios Alanis, Soon Xin Ng, and Lajos Hanzo. Fifteen years of quantum ldpc coding and improved decoding strategies. *IEEE Access*, 3:2492–2519, 2015. ISSN 2169-3536. DOI: 10.1109/ACCESS.2015.2503267.

[4] Daniel Barredo, Sylvain de Léséleuc, Vincent Lienhard, Thierry Lahaye, and Antoine Browaeys. An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays. *Science*, 354(6315):1021–1023, 2016. DOI: 10.1126/science.aah3778. URL https://www.science.org/doi/abs/10.1126/science.aah3778.

[5] 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(7997):58–65, 2024. DOI: 10.1038/s41586-023-06927-3.

[6] Sergey Bravyi and Matthew B. Hastings. Homological product codes. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 273–282, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450327107. DOI: 10.1145/2591796.2591870. URL https://doi.org/10.1145/2591796.2591870.

[7] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005. DOI: 10.1103/PhysRevA.71.022316. URL https://link.aps.org/doi/10.1103/PhysRevA.71.022316.

[8] Sergey Bravyi, Andrew W Cross, Jay M Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J Yoder. High-threshold and low-overhead fault-tolerant quantum memory. *Nature*, 627(8005):778–782, 2024. DOI: 10.1038/s41586-024-07107-7. URL https://doi.org/10.1038/s41586-024-07107-7.

[9] Nikolas P. Breuckmann and Jens N. Eberhardt. Balanced product quantum codes. *IEEE Transactions on Information Theory*, 67(10):6653–6674, 2021. DOI: 10.1109/TIT.2021.3097347. URL https://doi.org/10.1109/TIT.2021.3097347.

[10] Nikolas P. Breuckmann and Jens Niklas Eberhardt. Quantum low-density parity-check codes. *PRX Quantum*, 2(4), oct 2021. DOI: 10.1103/prxquantum.2.040101. URL https://doi.org/10.1103/prxquantum.2.040101.

[11] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996. DOI: 10.1103/PhysRevA.54.1098. URL https://link.aps.org/doi/10.1103/PhysRevA.54.1098.

[12] ChunJun Cao and Brad Lackey. Quantum lego: Building quantum error correction codes from tensor networks. *PRX Quantum*, 3:020332, May 2022. DOI: 10.1103/PRXQuantum.3.020332. URL https://link.aps.org/doi/10.1103/PRXQuantum.3.020332.

[13] Christopher Chamberland, Kyungjoo Noh, Patricio Arrangoiz-Arriola, Earl T. Campbell, Connor T. Hann, Joseph Iverson, Harald Putterman, Thomas C. Bohdanowicz, Steven T. Flammia, Andrew Keller, Gil Refael, John Preskill, Liang Jiang, Amir H. Safavi-Naeini, Oskar Painter, and Fernando G.S.L. Brandão. Building a fault-tolerant quantum computer using concatenated cat codes. *PRX Quantum*, 3:010329, Feb 2022. DOI: 10.1103/PRXQuantum.3.010329. URL https://link.aps.org/doi/10.1103/PRXQuantum.3.010329.

[14] F. Chiaraluce and R. Garello. Extended hamming product codes analytical performance evaluation for low error rate applications. *IEEE Transactions on Wireless Communications*, 3(6):2353–2361, Nov 2004. ISSN 1558-2248. DOI: 10.1109/TWC.2004.837405.

[15] Julien Du Crest, Mehdi Mhalla, and Valentin Savin. Stabilizer inactivation for message-passing decoding of quantum ldpc codes. In *2022 IEEE Information Theory Workshop (ITW)*, pages 488–493, Nov 2022. DOI: 10.1109/ITW54588.2022.9965902.

[16] Nicolas Delfosse and Gilles Zémor. Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel. *Phys. Rev. Research*, 2:033042, Jul 2020. DOI: 10.1103/PhysRevResearch.2.033042. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.033042.

[17] Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, and Shahar Mozes. Locally testable codes with constant rate, distance, and locality. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 357–374, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392648. DOI: 10.1145/3519935.3520024. URL https://doi.org/10.1145/3519935.3520024.

[18] 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, page 905–918, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. DOI: 10.1145/3564246.3585101. URL https://doi.org/10.1145/3564246.3585101.

[19] P. Elias. Error-free coding. *Transactions of the IRE Professional Group on Information Theory*, 4(4):29–37, Sep. 1954. ISSN 2168-2704. DOI: 10.1109/TIT.1954.1057464. URL https://doi.org/10.1109/TIT.1954.1057464.

[20] Jihao Fan, Yonghui Li, Min-Hsiu Hsieh, and Hanwu Chen. On quantum tensor product codes. *Quantum Info. Comput.*, 17(13-14):1105–1122, nov 2017. ISSN 1533-7146. DOI: 10.26421/QIC17.13-14-3. URL https://doi.org/10.26421/QIC17.13-14-3.

[21] Kosuke Fukui, Akihisa Tomita, Atsushi Okamoto, and Keisuke Fujii. High-threshold fault-tolerant quantum computation with analog quantum error correction. *Phys. Rev. X*, 8:021054, May 2018. DOI: 10.1103/PhysRevX.8.021054. URL https://link.aps.org/doi/10.1103/PhysRevX.8.021054.

[22] Craig Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, July 2021. ISSN 2521-327X. DOI: 10.22331/q-2021-07-06-497. URL https://doi.org/10.22331/q-2021-07-06-497.

[23] Daniel Gottesman. *Stabilizer codes and quantum error correction*. PhD thesis, California Institute of Technology, 1997. DOI: 10.7907/rzr7-dt72 URL https://resolver.caltech.edu/CaltechETD:etd-07162004-113028.

[24] Daniel Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57:127–137, Jan 1998. DOI: 10.1103/PhysRevA.57.127. URL https://link.aps.org/doi/10.1103/PhysRevA.57.127.

[25] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. In *Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics*, volume 68, pages 13–58, 2010. DOI: 10.1090/psapm/068.

[26] Markus Grassl. Bounds on the minimum distance of linear codes and quantum codes. Online available at http://www.codetables.de, 2007. Accessed on 2022-09-26.

[27] Christian Gross and Immanuel Bloch. Quantum simulations with ultracold atoms in optical

lattices. *Science*, 357(6355):995–1001, 2017. DOI: 10.1126/science.aal3837. URL https://www.science.org/doi/abs/10.1126/science.aal3837.

[28] Shouzhen Gu, Christopher A. Pattison, and Eugene Tang. An efficient decoder for a linear distance quantum ldpc code. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 919–932, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. DOI: 10.1145/3564246.3585169. URL https://doi.org/10.1145/3564246.3585169.

[29] Jérémie Guillaud and Mazyar Mirrahimi. Repetition cat qubits for fault-tolerant quantum computation. *Phys. Rev. X*, 9:041053, Dec 2019. DOI: 10.1103/PhysRevX.9.041053. URL https://link.aps.org/doi/10.1103/PhysRevX.9.041053.

[30] Masahito Hayashi. *Quantum information*. Springer, 2006. DOI: 10.1007/3-540-30266-2.

[31] Oscar Higgott and Nikolas P. Breuckmann. Improved single-shot decoding of higher-dimensional hypergraph-product codes. *PRX Quantum*, 4:020332, May 2023. DOI: 10.1103/PRXQuantum.4.020332. URL https://link.aps.org/doi/10.1103/PRXQuantum.4.020332.

[32] Morteza Hivadi. On quantum spc product codes. *Quantum Information Processing*, 17(12): 1–14, 2018. URL https://doi.org/10.1007/s11128-018-2095-3.

[33] Philip C. Holz, Silke Auchter, Gerald Stocker, Marco Valentini, Kirill Lakhmanskiy, Clemens Rössler, Paul Stampfer, Sokratis Sgouridis, Elmar Aschauer, Yves Colombe, and Rainer Blatt. 2d linear trap array for quantum information processing. *Advanced Quantum Technologies*, 3(11):2000031, 2020. DOI: https://doi.org/10.1002/qute.202000031. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.202000031.

[34] H. Imai and H. Fujiya. Generalized tensor product codes. *IEEE Transactions on Information Theory*, 27(2):181–187, March 1981. ISSN 1557-9654. DOI: 10.1109/TIT.1981.1056329.

[35] Pavithran Iyer and David Poulin. Hardness of decoding quantum stabilizer codes. *IEEE Transactions on Information Theory*, 61(9):5209–5223, Sep. 2015. ISSN 1557-9654. DOI: 10.1109/TIT.2015.2422294.

[36] Raphaël Le Bidan, Camille Leroux, Christophe Jego, Patrick Adde, and Ramesh Pyndiah. Reed-solomon turbo product codes for optical communications: From code optimization to decoder design. *EURASIP Journal on Wireless Communications and Networking*, 2008:1–14, 2008. DOI: 10.1155/2008/658042.

[37] Anthony Leverrier and Gilles Zémor. Quantum tanner codes. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 872–883, Oct 2022. DOI: 10.1109/FOCS54457.2022.00117. URL https://doi.org/10.1109/FOCS54457.2022.00117.

[38] Anthony Leverrier and Gilles Zémor. *Efficient decoding up to a constant fraction of the code length for asymptotically good quantum codes*, pages 1216–1244. 2023. DOI: 10.1137/1.9781611977554.ch45. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977554.ch45.

[39] Anthony Leverrier and Gilles Zémor. Decoding quantum tanner codes. *IEEE Transactions on Information Theory*, 69(8):5100–5115, Aug 2023. ISSN 1557-9654. DOI: 10.1109/TIT.2023.3267945.

[40] Zhaoyi Li, Isaac Kim, and Patrick Hayden. Concatenation Schemes for Topological Fault-tolerant Quantum Error Correction. *Quantum*, 7:1089, August 2023. ISSN 2521-327X. DOI: 10.22331/q-2023-08-22-1089. URL https://doi.org/10.22331/q-2023-08-22-1089.

[41] Daniel Litinski. Magic State Distillation: Not as Costly as You Think. *Quantum*, 3:205, December 2019. ISSN 2521-327X. DOI: 10.22331/q-2019-12-02-205. URL https://doi.org/10.22331/q-2019-12-02-205.

[42] D.J.C. MacKay, G. Mitchison, and P.L. McFadden. Sparse-graph codes for quantum error correction. *IEEE Transactions on Information Theory*, 50(10):2315–2330, Oct 2004. ISSN 1557-9654. DOI: 10.1109/TIT.2004.834737. URL https://doi.org/10.1109/TIT.2004.834737.

[43] John M Martinis. Superconducting phase qubits. *Quantum information processing*, 8:81–103, 2009. DOI: 10.1007/s11128-009-0105-1.

[44] H. Mukhtar, A. Al-Dweik, and A. Shami. Turbo product codes: Applications, challenges, and future directions. *IEEE Communications Surveys & Tutorials*, 18(4):3052–3069, Fourthquarter 2016. ISSN 1553-877X. DOI: 10.1109/COMST.2016.2587863.

[45] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, jun 2010. DOI: 10.1017/cbo9780511976667. URL https://doi.org/10.1017%2Fcbo9780511976667.

[46] Dimiter Ostrev. QKD parameter estimation by two-universal hashing. *Quantum*, 7:894, January 2023. ISSN 2521-327X. DOI: 10.22331/q-2023-01-13-894. URL https://doi.org/10.22331/q-2023-01-13-894.

[47] Pavel Panteleev and Gleb Kalachev. Degenerate Quantum LDPC Codes With Good Finite Length Performance. *Quantum*, 5:585, November 2021. ISSN 2521-327X. DOI: 10.22331/q-2021-11-22-585. URL https://doi.org/10.22331/q-2021-11-22-585.

[48] 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, page 375–388, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392648. DOI: 10.1145/3519935.3520017. URL https://doi.org/10.1145/3519935.3520017.

[49] Christopher A. Pattison, Michael E. Beverland, Marcus P. da Silva, and Nicolas Delfosse. Improved quantum error correction using soft information, 2021. DOI: 10.48550/arXiv.2107.13589 URL https://arxiv.org/abs/2107.13589.

[50] W. Wesley Peterson and E. J. Weldon Jr. *Error-correcting codes*. MIT press, 1972. URL https://mitpress.mit.edu/9780262527316/error-correcting-codes/.

[51] David Poulin and Yeojin Chung. On the iterative decoding of sparse quantum codes. *Quantum Info. Comput.*, 8(10):0987–1000, nov 2008. ISSN 1533-7146. DOI: 10.26421/QIC8.10-8. URL https://doi.org/10.26421/QIC8.10-8.

[52] Armanda O. Quintavalle, Michael Vasmer, Joschka Roffe, and Earl T. Campbell. Single-shot error correction of three-dimensional homological product codes. *PRX Quantum*, 2:020340, Jun 2021. DOI: 10.1103/PRXQuantum.2.020340. URL https://link.aps.org/doi/10.1103/PRXQuantum.2.020340.

[53] Nithin Raveendran, Narayanan Rengaswamy, Asit Kumar Pradhan, and Bane Vasić. Soft syndrome decoding of quantum ldpc codes for joint correction of data and syndrome errors. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 275–281, Sep. 2022. DOI: 10.1109/QCE53715.2022.00047.

[54] Nithin Raveendran, Narayanan Rengaswamy, Filip Rozpędek, Ankur Raina, Liang Jiang, and Bane Vasić. Finite Rate QLDPC-GKP Coding Scheme that Surpasses the CSS Hamming Bound. *Quantum*, 6:767, July 2022. ISSN 2521-327X. DOI: 10.22331/q-2022-07-20-767. URL https://doi.org/10.22331/q-2022-07-20-767.

[55] Narayanan Rengaswamy, Robert Calderbank, Henry D. Pfister, and Swanand Kadhe. Synthesis of logical clifford operators via symplectic geometry. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 791–795, June 2018. DOI: 10.1109/ISIT.2018.8437652.

[56] Joschka Roffe, David R. White, Simon Burton, and Earl Campbell. Decoding across the quantum low-density parity-check code landscape. *Phys. Rev. Res.*, 2:043423, Dec 2020. DOI: 10.1103/PhysRevResearch.2.043423. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.043423.

[57] A. M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77:793–797, Jul 1996. DOI: 10.1103/PhysRevLett.77.793. URL https://link.aps.org/doi/10.1103/PhysRevLett.77.793.

[58] Jean-Pierre Tillich and Gilles Zémor. Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–1202, 2014. DOI: 10.1109/TIT.2013.2292061. URL https://doi.org/10.1109/TIT.2013.2292061.

[59] Javier Valls, Francisco Garcia-Herrero, Nithin Raveendran, and Bane Vasić. Syndrome-based min-sum vs osd-0 decoders: Fpga implementation and analysis for quantum ldpc codes. *IEEE Access*, 9:138734–138743, 2021. ISSN 2169-3536. DOI: 10.1109/ACCESS.2021.3118544.

[60] J. Wolf. On codes derivable from the tensor product of check matrices. *IEEE Transactions on Information Theory*, 11(2):281–284, April 1965. ISSN 1557-9654. DOI: 10.1109/TIT.1965.1053771.

[61] Jack Keil Wolf. An introduction to tensor product codes and applications to digital storage systems. In *2006 IEEE Information Theory Workshop - ITW '06 Chengdu*, pages 6–10, Oct 2006. DOI: 10.1109/ITW2.2006.323741.

[62] Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, Ming Gong, Cheng Guo, Chu Guo, Shaojun Guo, Lianchen Han, Linyin Hong, He-Liang Huang, Yong-Heng Huo, Liping Li, Na Li, Shaowei Li, Yuan Li, Futian Liang, Chun Lin, Jin Lin, Haoran Qian, Dan Qiao, Hao Rong, Hong Su, Lihua Sun, Liangyuan Wang, Shiyu Wang, Dachao Wu, Yu Xu, Kai Yan, Weifeng Yang, Yang Yang, Yangsen Ye, Jianghan Yin, Chong Ying, Jiale Yu, Chen Zha, Cha Zhang, Haibin Zhang, Kaili Zhang, Yiming Zhang, Han Zhao, Youwei Zhao, Liang Zhou, Qingling Zhu, Chao-Yang Lu, Cheng-Zhi Peng, Xiaobo Zhu, and Jian-Wei Pan. Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.*, 127:180501, Oct 2021. DOI: 10.1103/PhysRevLett.127.180501. URL https://link.aps.org/doi/10.1103/PhysRevLett.127.180501.

[63] Qian Xu, J. Pablo Bonilla Ataides, Christopher A. Pattison, Nithin Raveendran, Dolev Bluvstein, Jonathan Wurtz, Bane Vasić, Mikhail D. Lukin, Liang Jiang, and Hengyun Zhou. Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays. *Nature Physics*, 2024. DOI: 10.1038/s41567-024-02479-z. URL https://doi.org/10.1038/s41567-024-02479-z.