Approach for the construction of non-Calderbank-Steane-Shor low-density-generator-matrix–based quantum codes

Patricio Fuentes,[1,*] Josu Etxezarreta Martinez,[1,†] Pedro M. Crespo,[1,‡] and Javier Garcia-Frias[2,§]
[1]*Tecnun–University of Navarra, 20018 San Sebastian, Spain*
[2]*University of Delaware, Newark, Delaware 19716, USA*

(Received 12 February 2020; revised 2 July 2020; accepted 6 July 2020; published 22 July 2020; corrected 21 October 2020)

Quantum low-density generator matrix (QLDGM) codes based on Calderbank-Steane-Shor (CSS) constructions have shown unprecedented error correction capabilities, displaying much improved performance in comparison to other sparse-graph codes. However, the nature of CSS designs and the manner in which they must be decoded limit the performance that is attainable with codes that are based on this construction. This motivates the search for quantum code design strategies capable of avoiding the drawbacks associated with CSS codes. In this article, we introduce non-CSS quantum code constructions based on classical LDGM codes. The proposed codes are derived from CSS QLDGM designs by performing specific row operations on their quantum parity check matrices to modify the associated decoding graphs. The application of this method results in performance improvements in comparison to CSS QLDGM codes, while also allowing for greater flexibility in the design process. The proposed non-CSS QLDGM scheme outperforms the best quantum low-density parity check codes that have appeared in the literature.

DOI: 10.1103/PhysRevA.102.012423

I. INTRODUCTION

In the realm of classical communications, turbo codes [1] and low-density parity check (LDPC) codes [2–4] are known to exhibit capacity-approaching performance at a reasonable decoding computational complexity. Turbo codes offer great flexibility in terms of their block length and rate. The first quantum codes based on turbo codes appeared in [5,6], and have since been modified and improved [7–11]. Aside from their block length and rate flexibility being on par with that of turbo codes, the sparse nature of LDPC codes guarantees that their quantum equivalents will require small numbers of quantum interactions per qubit during the error correction procedure [12], avoiding additional quantum gate errors and facilitating fault-tolerant decoding. These traits make quantum LDPC (QLDPC) codes especially well suited for quantum error correction.

Out of the existing types of LDPC codes, low-density generator matrix (LDGM) codes [13] provide a seamless manner for code design in the quantum domain. LDGM codes are a specific subset of LDPC codes whose generator matrices are also sparse, and thus their encoding complexity is similar to that of turbo codes, and much smaller than for standard LDPC codes. Given that LDGM codes form a special subclass of the LDPC code family, they can be decoded in the same manner and with the same complexity as any other LDPC code. LDGM codes have been extensively studied [3,13–15] and used in classical communications [16,17]. In [3], regular LDGM codes, which are a specific type of LDGM code whose parity check matrices have the same number of nonzero entries per row and the same number of nonzero entries per column, were shown to be asymptotically bad, displaying error floors that do not decrease with the block length. In [14,15], a concatenated LDGM scheme was shown to achieve performance similar to irregular LDPC codes at a very low encoding and decoding complexity.

Quantum LDPC codes are built by casting classical LDPC codes in the framework of stabilizer codes [18], which enables the design of quantum codes from any arbitrary classical binary and quaternary codes. In [19], the authors document the design of QLDPC codes based on their classical counterparts, detailing numerous construction and decoding techniques along with their flaws and merits. Among the discussed methods, the construction of QLDPC codes based on LDGM codes is shown to yield performance and code construction improvements, albeit at an increase in decoding complexity. This method was originally proposed in [20,21], where Calderbank-Steane-Shor (CSS) quantum codes based on regular LDGM classical codes were shown to surpass the best quantum coding schemes of the time, and performance was significantly improved in [22,23] by utilizing a parallel concatenation of two regular LDGM codes.

Quantum LDGM (QLDGM)–based quantum code implementations, as well as most QLDPC designs, are based on CSS constructions. CSS codes, simultaneously proposed by Calderbank, Shor, and Steane in [24,25], are a particular subset of the family of stabilizer codes. They provide a straightforward method to design quantum codes via existing classical codes. In general, decoding of quantum codes based on CSS designs is performed separately for bit- and

*pfuentesu@tecnun.es
†jetxezarreta@tecnun.es
‡pcrespo@tecnun.es
§jgf@udel.edu

phase-flip errors, which negatively impacts their performance. In fact, CSS constructions decoded separately are limited by an unsurpassable bound, referred to as the *CSS lower bound* [26]. Joint decoding of bit- and phase-flip errors using modified CSS decoders capable of exploiting the correlation between the aforementioned errors has been considered in the literature [27–33], and performance improvements associated with these modified CSS decoders have been shown in these articles. Nonetheless, the improvements provided by these modified CSS decoders come at the expense of an increased decoding complexity, which, along with the performance limitations of conventional CSS decoding, inspires the search for non-CSS constructions, as they should theoretically be able to outperform CSS codes and avoid complex decoding strategies. Non-CSS LDPC-based codes were proposed in [34]. However, despite showing promise, they fail to outperform existing CSS QLDPC codes for comparable block lengths.

In this paper, we propose a non-CSS scheme based on LDGM codes and compare its performance to existing CSS QLDGM codes. We explain how existing CSS QLDGM codes can be modified to create non-CSS quantum codes, and provide insight regarding how this construction can be optimized. We also show how the performance of the resulting non-CSS codes is similar to that of the CSS schemes they are derived from, despite the fact that their quantum rate is higher. When their rate is the same, the non-CSS scheme outperforms the original CSS design over the depolarizing channel. Finally, we compare the non-CSS structures proposed here with other existing QLDPC codes in the literature, illustrating that our method surpasses such error correction schemes.

The remainder of this paper is structured as follows. We commence with an overview of some important preliminary topics in Sec. II. We proceed by presenting the non-CSS LDGM-based quantum codes in Sec. III. In Sec. IV, we compare the performance of the proposed scheme to existing CSS quantum codes. Section V concludes the paper.

II. PRELIMINARIES

In this section, a brief review of the concepts, definitions, and notation used in this paper is provided.

A. Basic concepts

1. Quantum information

The simplest quantum mechanical system and the basic unit in quantum information is known as the qubit. In the state vector formulation, it is denoted by $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \in \mathcal{H}_2$, where $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$, and $\mathcal{H}_2$ refers to the complex Hilbert space of dimension 2. Another formulation of quantum mechanics can be given in terms of the so-called density matrices ρ, which is useful in order to describe systems whose state is not completely known in state vector terms.

2. The Pauli group and the effective Pauli group

The *Pauli group* is a mathematical group of significant interest for stabilizer codes. Let Π be the set of Pauli operators $\{I, X, Y, Z\}$, and $\Pi^{\otimes N} = \{I, X, Y, Z\}^{\otimes N}$ denote the set of N-fold tensor products of single-qubit Pauli operators. Then, $\Pi^{\otimes N}$ together with the possible overall factors $\pm 1, \pm i$ forms an Abelian group known as the *N-fold Pauli group* $\mathcal{G}_N$, defined as $\mathcal{G}_N = \{\beta_1 I, \beta_2 X, \beta_3 Y, \beta_4 Z\}^{\otimes N}$, where $\beta_k = \{\pm 1, \pm i\}$.

Now let $[A] = \{\beta A | \beta \in \mathcal{C}, |\beta| = 1\}$ be the equivalence class of matrices equal to A up to a phase factor. Then the set $[\bar{\mathcal{G}}_N] = [\Pi^{\otimes N}] = \{[I], [X], [Y], [Z]\}^{\otimes N}$ forms an Abelian group under the multiplication operation defined by $[A][B] = [AB]$. This group is called the *effective N-fold Pauli group*.

3. Quantum channels

The effects quantum decoherence has on quantum information are usually described by means of quantum channels, $\mathcal{N}$. A widely applied quantum channel model used to represent the decoherence effects suffered by quantum information described by a density matrix ρ is the generic Pauli channel $\mathcal{N}_P$. The effect of the Pauli channel $\mathcal{N}_P$ on an arbitrary quantum state is described by

$$\mathcal{N}_P(\rho) = (1 - p_x - p_y - p_z)\rho + p_x X\rho X + p_y Y\rho Y + p_z Z\rho Z.$$

A qubit then experiences a bit flip (X operator) with probability p_x, a phase flip (Z operator) with probability p_z, or a combination of both (a Y operator) with probability p_y.

In most of the work conducted on quantum error correction, the independent depolarizing channel model is considered [5,7,34,35]. This model is a specific instance of the Pauli channel in which the individual depolarizing probabilities are all equal, i.e., $p_x = p_z = p_y$, and the channel is characterized by the depolarizing probability p. When quantum states of N qubits are considered, the errors that take place belong to the N-fold Pauli group $\mathcal{G}_N$.[1] Because we are considering the independent instance of the Pauli channel, these errors will act independently on each qubit causing an X, Z, or Y error with probability $p/3$ and leaving it unchanged with probability $(1 - p)$.

A simpler model for a quantum channel, known as the *iid X/Z channel*, is introduced in [12], where Z and X errors are modeled as independent events identically distributed according to the flip probability f_m. This quantum channel model is analogous to two independent binary symmetric channels (BSCs) with marginal bit-flip probability $f_m = 2p/3$, where the separate BSCs can be seen as Z and X error channels, respectively. Given that Y errors occur when both a phase and a bit flip happen to the same qubit, the simplified notion of the iid X/Z channel ignores any correlation that exists between X and Z errors in the depolarizing channel.

4. The hashing bound

The quantum capacity of a quantum channel is defined as the highest possible asymptotically achievable rate at which quantum information can be asymptotically transmitted in an error-free manner. The hashing bound defines a lower bound on the achievable quantum capacity of the depolarizing channel and is computed as $C_{\text{hash}}(p) = 1 - H_2(p) - p\log_2 3$, where $H_2(p)$ is the binary entropy function and p represents the depolarizing probability [6,36]. This means that for a given

[1]Given that the global phase has no observable consequence, the instances of considered errors will be the elements of $\bar{\mathcal{G}}_N$.

value of p, the hashing bound represents a lower bound on the highest possible coding rate at which asymptotically error-free quantum communication is possible. Alternatively, for a specific quantum rate R_Q, where we have $R_Q = C_{\text{hash}}(p^*)$, p^* represents a bound on the channel's depolarizing probability [37]. Analogously to the classical domain, we can refer to p^* as the *noise limit*.

Ideally, quantum codes that are properly designed should ensure error-free communications close to the noise limit p^*. We can efficiently characterize the quality of quantum codes built for the depolarizing channel in terms of their quantum coding rate R_Q and how far away they are from the hashing bound. The distance from the hashing bound can be computed based on the expression

$$\delta = 10\log_{10}\left(\frac{p^*}{p}\right), \tag{1}$$

where we use δ to represent the distance to the hashing bound in decibels (dB), p^* is the noise limit of the depolarizing channel for a specific quantum coding rate R_Q, and p is the highest depolarizing probability at which the code in question can operate in an error-free manner.

B. Stabilizer codes

Stabilizer codes are a class of quantum error correction codes that can be efficiently designed based on existing classical codes. A stabilizer code $C(S)$ is defined by a set of operators S that generate an Abelian subgroup of the N-fold Pauli group $\mathcal{G}_N$ under multiplication. The code space defined by the stabilizer group is

$$C(S) = \left\{|\psi\rangle \in \mathcal{H}_2^{\otimes N} : S_i|\psi\rangle = |\psi\rangle, \forall i\right\},$$

i.e., the simultaneous $+1$ eigenspace defined by the elements of the stabilizer group S.

A generator of a stabilizer code, or more generally any Pauli operator on N qubits, can be described in terms of its symplectic representation [38]. Using this representation, each element of the N-fold Pauli group can be written as a unique binary string of length $2N$, which is built by joining two separate binary strings[2] of length N. Individually, each of the length-N binary strings will represent the presence of a Z or X operator on each of the N qubits. Considered jointly, the strings also represent I and Y operators. More explicitly, given an element of $\mathcal{G}_N$ represented by the length-$2N$ string $U = (U_z|U_x)$, where U_z and U_x are length-N strings, zero entries in the same position of both length-N strings represent a single-qubit I operator, a nonzero entry in U_z and a zero entry in the same position of U_x will represent a single-qubit Z operator, a zero entry in U_z and a nonzero entry in the same position of U_x will represent a single-qubit X operator, and nonzero entries in the same position of both strings represent a single-qubit Y operator. Applying this representation to the generators S_i of a stabilizer code enables the definition of the quantum parity check matrix (QPCM) for the code. The parity check matrix (PCM) of a stabilizer code will be in the form $H_Q = (H_z|H_x)$, where row i of matrix H_Q is the symplectic representation of stabilizer generator S_i.

1. The symplectic criterion

It was mentioned earlier that the set of generators S of a stabilizer code forms an Abelian subgroup of $\mathcal{G}_N$. Therefore, for a given set $\hat{S}$ of N-qubit Pauli operators to define a generator of a stabilizer code, the operators $\hat{S}_i$ must commute among themselves. This commutation constraint of the stabilizer generators is translated into binary representation by means of the symplectic product. Given any two N-qubit Pauli operators described by their binary representation $P_1 = (P_{z1}|P_{x1})$ and $P_2 = (P_{z2}|P_{x2})$, they will commute if $(P_{z1}P_{x2}^T + P_{z2}P_{x1}^T) \bmod 2 = 0$. If we now introduce the quantum parity check matrix notation mentioned earlier, this constraint can be reexpressed for the entire stabilizer code, as each of the rows must fulfill it. This is shown in (2), where the $\star$ operator, known as the symplectic product, represents the operation itself. We will refer to this expression as the *symplectic criterion*:

$$H_z \star H_x = \left(H_z H_x^T + H_x H_z^T\right) \bmod 2 = 0. \tag{2}$$

The symplectic criterion is of significant importance because it determines which existing classical codes can be used to design stabilizer codes. For instance, given two parity check matrices of any two classical codes, H_1 and H_2, the parity check matrix obtained as $H_Q = (H_1|H_2)$ will only define a stabilizer code if it satisfies (2). Essentially, for two classical codes to be applicable in the construction of a stabilizer code, their parity check matrices must fulfill $(H_1H_2^T + H_2H_1^T) \bmod 2 = 0$.

2. Quantum syndrome detection

Since the channels considered in this article (Sec. II A 3) model decoherence by means of errors that belong to the N-fold Pauli group, such quantum errors will either commute or anticommute with each of the stabilizer generators S_i of a given stabilizer code $C(S)$ [38]. An error operator E can be described using the symplectic representation as the length-$2N$ binary string e. If we write e as $(e_z|e_x)$, when multiplying e in terms of the symplectic product (mod 2) by a row of the parity check matrix of a stabilizer code, 0 will be obtained if E and the generator associated with that row commute, whereas 1 will be obtained if they anticommute.

As is shown in (3), multiplying this symplectic representation of the error operators by the quantum parity check matrix of a stabilizer code will yield the quantum syndrome s. We will later use this syndrome in the decoding process to estimate the error pattern e:

$$s = H_Q \star e = (H_z e_x + H_x e_z) \bmod 2, \tag{3}$$

where $e = (e_z|e_x)$ is the symplectic representation of the error pattern, $H_Q = (H_z|H_x)$ is the quantum parity check matrix of a stabilizer code, and s represents the quantum syndrome. Physically, the calculation of the error syndrome is performed by measuring the commutation status of the error operator

[2]Note that by doing so, the global phase is lost, and so the map is between the effective Pauli group and the binary field. Global phase has no observable consequences, so neglecting it makes good physical sense.

with each generator of the stabilizer group (0 if they commute and 1 if they do not).

C. CSS codes

Two binary classical LDPC codes can only be used to construct a stabilizer code if they satisfy the symplectic criterion (2). The first design strategy one could devise to construct stabilizer codes would be the random selection of pairs of classical LDPC codes. However, finding two LDPC codes of reasonable block size that satisfy (2) is highly unlikely. Calderbank-Shor-Steane codes [24,25], provide a more efficient design strategy than random selection of classical codes. The quantum parity check matrix of these codes is written as

$$H_Q = (H_z|H_x) = \begin{pmatrix} H'_z & 0 \\ 0 & H'_x \end{pmatrix}, \tag{4}$$

where $H_z = \binom{H'_z}{0}$ and $H_x = \binom{0}{H'_x}$.

In this construction, H'_z and H'_x are the parity check matrices of two classical LDPC codes C_1 and C_2, respectively, where each matrix is used to correct either bit flips (H'_z) or phase flips (H'_x). The classical codes are chosen so that $C_2^{\perp} \subseteq C_1$, where $C_2^{\perp}$ is the dual of the classical LDPC code C_2. This design constraint, generally referred to as the *CSS condition*, reduces (2) to $(H'_z H'^T_x) \bmod 2 = 0$.

D. Systematic classical LDGM codes

Systematic LDGM codes are useful, in both classical and quantum environments, because of the particular structure of their generator and parity check matrices. Let C be a systematic LDGM code. Then, its generator matrix $\tilde{G}$ and its parity check matrix $\tilde{H}$ can be written as

$$\tilde{G} = (\mathbb{I}\ P), \quad \tilde{H} = (P^T\ \mathbb{I}), \tag{5}$$

where $\mathbb{I}$ denotes the identity matrix, and $P = [p_{lm}]$ is a sparse matrix.

Because LDGM codes belong to the family of linear block codes, these matrices will satisfy $(\tilde{G}\tilde{H}^T) \bmod 2 = 0$ and $(\tilde{H}\tilde{G}^T) \bmod 2 = 0$. Those systematic LDGM codes in which the rows and columns of the PCM have degrees[3] X and Y, respectively, will be denoted as (X, Y) regular LDGM codes. Regular LDGM codes are known to be asymptotically bad [3], displaying error floors that do not decrease with the block length. However, in [39], codes built via the parallel concatenation of two regular LDGM codes[4] were shown to yield significant reduction in these error floors. The parallel concatenation of two regular LDGM codes with generator matrices $G_1 = [\mathbb{I}\ P_1]$ and $G_2 = [\mathbb{I}\ P_2]$, where P_1 and P_2 have degree distributions (y_1, y_1) and (y_2, z_2), is the irregular LDGM code with generator matrix $G = [\mathbb{I}\ P_1 P_2]$. Generally, this concatenation is accomplished by using a high-rate code G_2 that is able to reduce the error floor of G_1, while also causing negligible degradation of the original convergence threshold.

Classical LDPC decoding is performed by solving the equation $s = H_c e$, where s represents the received syndrome, H_c is the PCM of the code, and e is the error pattern we wish to recover. Given that LDGM codes are a specific subset of LDPC codes, they are decoded in exactly the same manner as generic LDPC codes. However, LDGM decoding can also be interpreted as a method to solve equation $c = Pu$, where c is the vector of parity bits generated at the encoder, P is the constituent sparse matrix of the LDGM generator matrix, and u is the information message we want to obtain. Expressions such as $s = H_c e$ and $c = Pu$ can be described using bipartite graphs such as Tanner graphs [40,41] or the more general factor graphs [42]. Based on these depictions, the procedure of solving these equations can be understood as the operation of a decoding algorithm over the corresponding graph. Tanner graphs were designed as a tool to represent families of codes that are generalizations of classical LDPC codes. Given the PCM of an $[n, k]$ classical linear block code, $H = [h_{i,j}]_{(n-k)\times n}$, the corresponding Tanner graph can be built using two types of interconnected nodes: each of the n columns of the PCM is associated with a variable node and every row out of the $(n - k)$ rows of the PCM is associated with a parity check node. An edge will connect parity check node i to variable node j if and only if $h_{i,j} \neq 0$.

Tanner graphs are a specific instance of a well-known generalization of bipartite graphs called factor graphs, which, given their generality, have been applied to many fields outside of communications and error correction. Factor graphs were developed to represent algorithms that deal with complicated "global" functions of many variables and that derive their computational efficiency by exploiting the way in which the global function factors into a product of simpler "local" functions, each of which depends on a subset of the variables. Factor graphs are bipartite graphs that portray these algorithms by expressing which variables are arguments of which local functions [42]. This representation also allows the description of a generic message-passing algorithm, known as the sum-product algorithm (SPA), which operates over a factor graph and attempts to compute various marginal functions associated with the global function. Algorithms used to decode classical LDPC codes (solving $s = H_c e$), such as applying belief propagation (BP) [43] over the Tanner graph defined by H_c, were shown to be specific instances of the SPA operating over the corresponding factor graph in [44]. Therefore, decoding of classical LDGM codes can be understood as the application of the SPA over the factor graph defined by equation $c = Pu$.

E. Quantum LDGM CSS codes

The first intuition to derive the QPCM of a QLDGM CSS code would be to select any classical LDGM code with parity check and generator matrices $\tilde{H}$ and $\tilde{G}$, and set $H'_z = \tilde{H}$ and $H'_x = \tilde{G}$ in (4), since the property $(\tilde{G}\tilde{H}^T) \bmod 2 = 0$ and $(\tilde{H}\tilde{G}^T) \bmod 2 = 0$ would ensure the fulfillment of the symplectic criterion. However, this results in a QPCM H_Q of

[3]The degree of the columns is the number of nonzero entries per column of the PCM. The degree of the rows is given by the number of nonzero entries per row of the PCM. An LDGM code is said to be regular when all the rows of its PCM have the same number of nonzero entries, X, and so do its columns, Y.

[4]The parallel concatenation of regular LDGM codes is equivalent to an LDGM code with an irregular degree distribution.

size $N \times 2N$, which cannot be used for encoding purposes. This is easy to see based on the following discussion.

Consider an $[N, k]$ stabilizer code that has $(N-k)$ stabilizers and is described by means of a QPCM of size $(N-k) \times 2N$. Such a quantum code encodes k qubits into N qubits, which implies that the code has a quantum rate $R_Q = \frac{k}{N}$. Thus, the quantum rate of a code with QPCM H_Q of size $N \times 2N$ is $R_Q = 0$. Therefore, to build a valid quantum code the number of rows in H_Q must be reduced, while ensuring that the CSS condition is fulfilled. In [20], the authors successfully reduce the number of rows of the generator and parity check matrices of a classical LDGM code via linear row operations while showing that the CSS condition is kept. This is reflected in the following theorem.

Theorem 1. Given the generator and parity check matrices of a systematic LDGM code (5), define $H_{m_1 \times N} = [M_1]_{m_1 \times n_1}[\tilde{H}]_{n_1 \times N}$ and $G_{m_2 \times N} = [M_2]_{m_2 \times n_2}[\tilde{G}]_{n_2 \times N}$, where $n_1 + n_2 = N$ and M_1 and M_2 are low-density full-rank binary matrices whose numbers of rows satisfy $m_1 < n_1$ and $m_2 < n_2$, respectively. Then, the quantum PCM shown in (6), obtained by setting $H'_z = H$ and $H'_x = G$ in (4), is the quantum PCM of an LDGM-based CSS code with rate $R_Q = \frac{N-m_1-m_2}{N}$:

$$H_Q = (H_z|H_x) = \begin{pmatrix} H & 0 \\ 0 & G \end{pmatrix} = \begin{pmatrix} M_1\tilde{H} & 0 \\ 0 & M_2\tilde{G} \end{pmatrix}. \quad (6)$$

Quantum CSS LDGM codes are decoded by applying the SPA over the factor graph defined by the QPCM in (6). The derivation of this factor graph is performed as in [20], by splitting the symplectic representation of the error pattern into two parts, $e = (e_z|e_x)$, and relating it to the syndrome via a two-step process.[5] In the following, we illustrate this derivation for e_x, the part of the symplectic representation of the error sequence related to the X operators. The procedure for e_z is identical but using G instead of H in (7):

$$s = He_x = M_1\tilde{H}e_x = M_1[P^T \; \mathbb{I}]e_x. \quad (7)$$

If we now split e_x into $e_x = (e_{x_1} \; e_{x_2})^T$, we can write

$$\begin{aligned} d_x &= [P^T \; \mathbb{I}]e_x = [P^T \; \mathbb{I}]_{n_1 \times N} \begin{pmatrix} e_{x_1} \\ e_{x_2} \end{pmatrix}_{N \times 1} \\ &= P^T_{n_1 \times n_2}\left[e_{x_1}\right]_{n_2 \times 1} + \left[e_{x_2}\right]_{n_1 \times 1}. \end{aligned} \quad (8)$$

We then relate d_x to the syndrome as

$$s_{m_1 \times 1} = M_{1_{m_1 \times n_1}} d_{n_1 \times 1}. \quad (9)$$

The factor graph shown in Fig. 1 is obtained based on expressions (8) and (9), as well as their equivalents when using e_z and G in (7).

Upon closer examination of the QPCM in (6), it is easy to see that decoding for the H and G matrices can be done separately. This is also visible in Fig. 1, where the leftmost subgraph is associated with the decoding of matrix H (e_x or X containing operators) and the rightmost subgraph is associated with the decoding of matrix G (e_z or Z containing operators). Separate decoding of these matrices is made possible by the nature of CSS constructions, which results in syndrome nodes

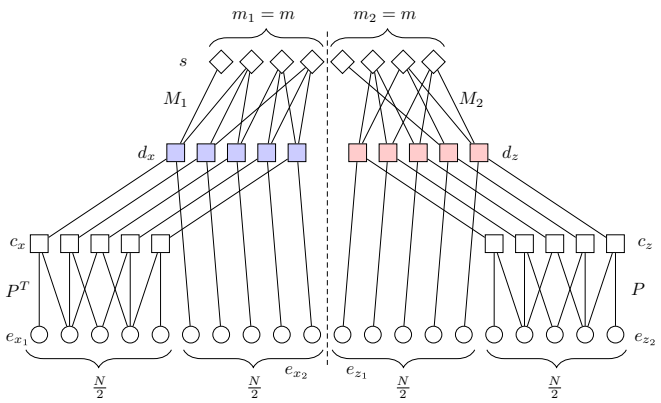

FIG. 1. Generalized factor graph for a QLDGM CSS scheme. The dotted line is included to emphasize the separation of the two constituent subgraphs. The leftmost subgraph decodes the X errors, while the one on the right decodes the Z errors. We have assumed that $m_1 = m_2 = m$ and $n_1 = n_2 = \frac{N}{2}$.

containing information only of either X or Z operators. This is reflected on the factor graph by the fact that a specific s node connects to either a d_x or a d_z node.

The matrix multiplications used to perform the linear row operations on $\tilde{H}$ and $\tilde{G}$ generate a middle layer, represented by the c and d nodes, in both decoding subgraphs of Fig. 1. This new layer hampers the decoding algorithm, especially during the initial decoding iterations, since *a priori* information regarding the aforementioned middle layer nodes is not available. This can be seen in Fig. 2, where a generic quantum communication system is shown. The LDGM decoder block of this figure, which runs the SPA over the graph shown in Fig. 1, has the syndrome s and the *a priori* probability of the error pattern P_{ch} as its inputs. However, it receives no information pertaining to the c and d nodes.

In [20], the authors circumvent this lack of information by using the so-called *doping* technique of [45]. This method introduces degree-1 syndrome nodes into the decoding graph. These degree-1 nodes, which we will refer to as s_A nodes, send exact[6] information to the d nodes they are connected to. The transmission of correct syndrome information from the s_A nodes to the d nodes represents a passing down of accurate knowledge to lower layers of the factor graph that should make up for the lack of information regarding c and d nodes during initial decoding iterations. This should have a positive impact on decoding performance and ultimately push the entire process in the right direction. The degree-1 syndrome vertices are embodied within the M_1 and M_2 matrices as rows with a single nonzero entry, which corresponds to the edge that connects a given s_A node to a d node. The other rows of matrices M_1 and M_2, which correspond to the rest of the s nodes, have as many nonzero entries as required to

[5]The syndrome is obtained as shown in (3).

[6]These messages are exact because, as required by the SPA update rule and the fact that s_A nodes are degree-1 nodes, the messages they send are strictly dependent on the syndrome information available at each s_A node.

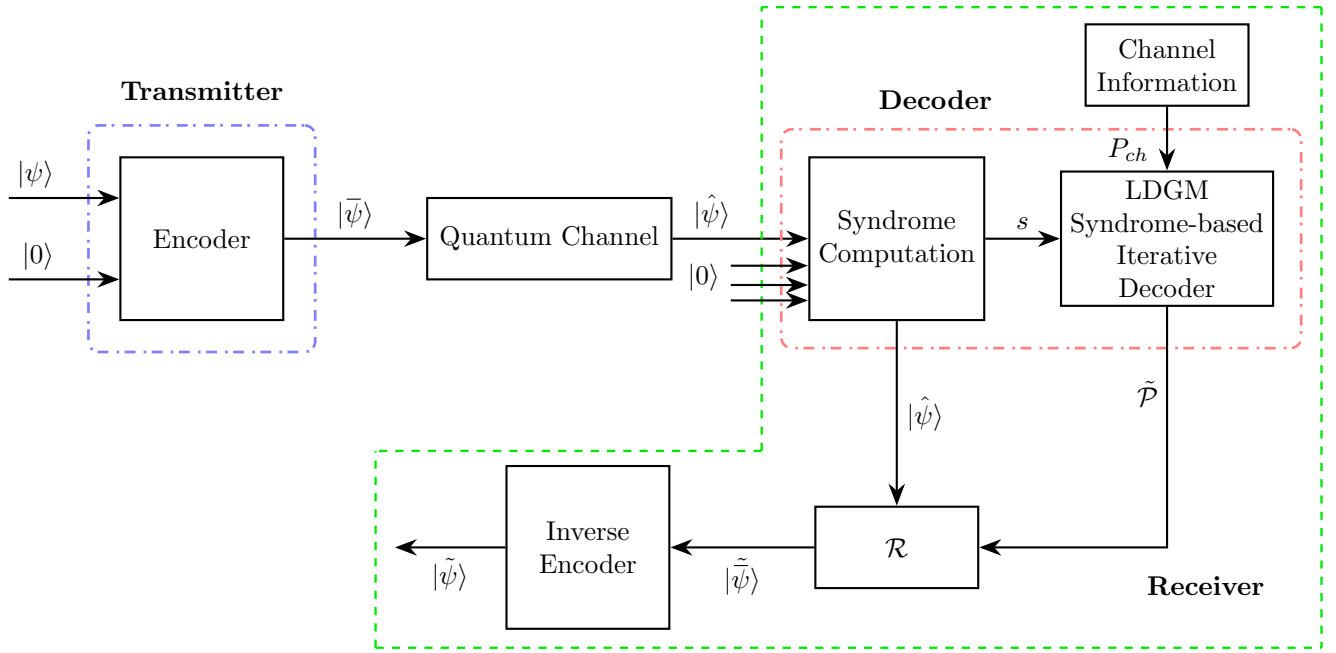

FIG. 2. Schematic of a quantum communication system using QLDGM codes. At the transmitter, a QLDGM code with QPCM H_Q maps the quantum state (information qubits) $|\psi\rangle \in \mathcal{H}_2^{\otimes k}$ onto the code word (physical qubits) $|\bar{\psi}\rangle \in \mathcal{H}_2^{\otimes N}$ by using $(N-k)$ ancilla qubits. At the receiver, the noisy code word $|\hat{\psi}\rangle = \mathcal{P}\,|\bar{\psi}\rangle$ is received, where $\mathcal{P} \in \mathcal{G}_N$ is the error inflicted by the quantum channel. The quantum state $|\hat{\psi}\rangle$ is processed by the syndrome computation block at the decoder to compute its error syndrome s. The ancilla qubits at the input of the syndrome computation block are necessary to physically implement the syndrome calculation. Together with the *a priori* channel information P_{ch}, s is provided to the syndrome-based LDGM decoder, which yields an estimate of the original error pattern $\tilde{\mathcal{P}}$. The recovery operator $\mathcal{R}$ uses $\tilde{\mathcal{P}}$ to correct the noisy code word. Finally, the inverse encoder yields an estimate of the information qubits $|\tilde{\psi}\rangle$ from the corrected code word $|\tilde{\bar{\psi}}\rangle$.

guarantee the regularity[7] of the d nodes and the necessary number of s_A nodes. This results in matrices M_1 and M_2 having a special degree distribution which is described by means of the notation $(y; 1, x)$ and the parameter t, where y represents the degree of the d nodes, t is the number of syndrome nodes that are forced to have degree 1 (they become s_A nodes), and x represents the degree of the remaining syndrome nodes, referred to as s_B nodes.

Given the particular structure of the M_1 and M_2 matrices, and the number of different types of nodes that are present in the factor graph shown in Fig. 1, the sum-product decoding of these quantum LDGM CSS codes becomes relatively nuanced. In [23], a technique known as discretized density evolution (DDE) [46] is applied to optimize quantum LDGM CSS codes, which also provides a complete description of how the decoding process unfolds over the graph shown in Fig. 1.

III. DESIGN OF LDGM-BASED NON-CSS CODES

As mentioned in the previous section, LDGM-based CSS codes can be decoded over two separate (sub)graphs like the ones shown in Fig. 1. One graph will be used to decode the X errors and the other to decode the Z errors. This is made possible by the specific nature of the quantum PCMs of CSS codes (4), and is visible on a CSS decoding graph by the fact that any given s node can only be connected to either d_x or d_z nodes: a subset of s nodes is used to decode the X operators and another subset is used to decode the Z operators.

The main appeal of non-CSS codes is their ability to exploit redundancy more efficiently than CSS schemes. In our proposed non-CSS construction, we achieve this by allowing edges from a given s node to go to both d_x and d_z nodes. The first method that comes to mind to implement this idea is to randomly distribute the edges of the upper layer of the graph in a manner that ensures that s nodes are connected to both d_x and d_z nodes. However, attempting to decode the X and Z parts over a decoding graph with s nodes whose edges have been haphazardly assigned to both d_x and d_z nodes will cause numerous decoding problems. For instance, not defining a specific distribution for these edges inadvertently causes a reduction in the number of s_A nodes, and not limiting their total number causes a reduction in the values of the log-likelihood messages exchanged in the decoding process, which severely degrades the decoding performance. Therefore, it is important to optimally design the upper layer of the decoding graph when constructing a non-CSS QLDGM-based code. Devising a proper way of distributing the connections among s, d_x, and d_z nodes in the decoding graph is paramount to construct good non-CSS LDGM-based quantum codes.

A. Proposed procedure for the construction of non-CSS QLDGM codes

We begin the non-CSS design process by using a CSS quantum code based on classical LDGM codes [21–23] as

[7]Regularity in this context implies that all the d nodes have the same degree, i.e., that they are all connected to the same number of s nodes.

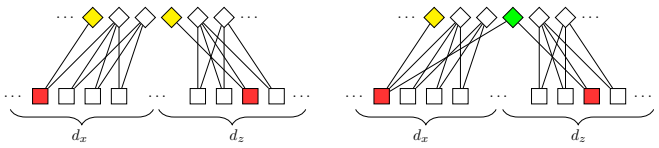

FIG. 3. Generation of an s_C node. The upper nodes represent the syndrome nodes while the bottom nodes represent the d nodes (d_x and d_z denote the d nodes associated with each of the separate CSS decoding subgraphs). The s_A nodes are represented in yellow, the d_A nodes are shown in red, and the s_C node is pictured in green.

the starting point. For the sake of simplicity and comparison continuity, we maintain the requirements enforced in [21–23]: the matrices used to perform linear row operations are equal to each other $M_1 = M_2 = M$, and the degree distribution of P^T and P are the same.

The CSS QLDGM code used as a starting point will be associated with two separate decoding subgraphs, one for H and the other for G. The upper layers of these subgraphs (the number and degree distribution of the d, s_A, and s_B nodes) will be defined by two identical matrices M of size $m \times \frac{N}{2}$ described by $(y; 1, x)$ which have t rows with a single nonzero entry.

We can build our non-CSS scheme using two different strategies. Given that both of them involve very similar procedures, we begin by explaining the simplest construction method. Then, we will present the second proposed design technique.

1. Method 1: Syndrome node combination

Non-CSS codes based on the first strategy are constructed as follows:

(1) First, generate a new matrix, M_d, as

$$M_d = \begin{pmatrix} M_{m\times\frac{N}{2}} & 0_{m\times\frac{N}{2}} \\ 0_{m\times\frac{N}{2}} & M_{m\times\frac{N}{2}} \end{pmatrix}_{2m\times N}. \tag{10}$$

(2) Select q nodes out of the $2t$ s_A nodes of matrix M_d,[8] which we will refer to as s_C nodes, and add an edge from these nodes to the d nodes on the side of the decoding graph they are not connected to. We apply a criterion to ensure that these new connections are not made randomly: *the edges added to the q selected s_A nodes can only be made to a d node (d_x or d_z) that is a d_A node*. We define d_A nodes as any d nodes that are connected to an s_A node. Of the q s_C nodes, half of them proceed from s_A nodes in the CSS subgraph used to decode the X operators, while the other half come from s_A nodes in the CSS subgraph used to decode the Z operators. Figure 3 illustrates how an s_C node is generated.

The reasoning behind adding edges that traverse the X and Z sides of the graph only to s_A nodes is based on the following considerations: First, transforming an s_A node into an s_C node implies that the new node no longer provides perfect syndrome information, given that it is now connected to two d nodes. However, the fact that an s_C node only has two edges implies that its syndrome information, although not transferred exactly, still has high impact when computing messages for associated nodes. At the same time, the edge that traverses from the s_C node to the other side of the factor graph (either d_x or d_z) reaches a d_A node. Considering that messages from d_A nodes are more likely to be correct (they are connected to an s_A node), coupled with the fact that s_C node syndrome information still plays an important role in the messages that the node computes, it is reasonable to assume that s_C nodes relay accurate information and that they behave in a similar manner to s_A nodes. Therefore, s_C nodes provide a way in which reliable messages can be exchanged between both sides of the factor graph, which should have a positive impact on decoding and improve the code performance.

It is important to note that if we were to add a cross-graph edge to an s_B node, because of its high degree, the messages received over this new edge would play a limited role in the computations made by the node. By association, the cross-graph messages transmitted over the new edge would also have a very limited effect on the computation of messages exchanged on the other side of the graph and little performance improvement, if any, would be obtained.

Notice that at this stage we have transformed M_d into a new matrix M'_d, modifying the upper layer of the original CSS decoding graph of Fig. 1 in the following manner:

(i) There are q s_C nodes that are connected to both sides of the graph.

(ii) Some d nodes are connected to both s_A and s_C nodes.

These modifications force the s and d nodes of the non-CSS decoding graph to have a somewhat irregular edge distribution. Indeed, the "regularity" of the d nodes has been broken in order to connect the separate CSS decoding subgraphs, resulting in $\frac{q}{2}$ d_x nodes and $\frac{q}{2}$ d_z nodes having an additional edge. Furthermore, q s nodes now have two edges, one of them directed toward a d_x node and the other toward a d_z node.

It is intuitive that the performance of this novel non-CSS structure should at least be as good, if not better, than that of the CSS scheme utilized as a starting point, provided that the parameter q is chosen properly. If we select $q \ll m$, the decrease in the number of s nodes providing perfect information will be small and should have negligible impact in the decoding process.[9] On the contrary, the degree-2 s_C nodes allow the exchange of information between both sides of the graph as the iterative decoding process progresses, potentially improving the decoding performance. Therefore, we expect this scheme to present its best performance for a specific range of small values of q, with deterioration occurring when q is increased beyond this range.

2. Method 2: Syndrome node combination + removal of s_A nodes

The second design technique removes q syndrome nodes from the decoding graph generated by the first method,

[8]Note that M_d, as defined in (10), is a simple algebraic representation of the upper layer of the graph in Fig. 1.

[9]A total of q s_A nodes get converted into s_C nodes, which do not provide perfect information.

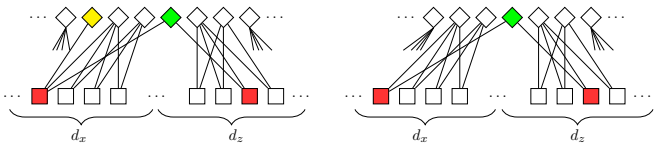

FIG. 4. Removal of an s_A (yellow) node that was previously used to generate an s_C (green) node.

specifically the q s_A nodes connected to a d_A node that is linked to an s_C node. The process is shown in Fig. 4. Notice that with this construction the regularity of the d nodes is maintained and the data rate of the code is increased.[10] Moreover, as long as the number of removed s_A nodes is not too large, the impact on decoding should be minimal: although s_C nodes do not provide "perfect" information, much of the reliability of the messages they transmit to the corresponding d (previously d_A) nodes will be kept, as they only have two edges. This should mitigate any impact on performance. Because method 2 requires the removal of some s_A nodes, compared with the CSS code used as a starting point, a larger number of s_A nodes may be necessary in the non-CSS structure to ensure acceptable performance as the value of q grows. This hints toward an increased importance of applying *doping* (i.e., increasing the value of t) to the constituent matrices M.

Once the specific s_A nodes are removed, we obtain a new matrix M'_d of size $(2m-q)\times N$, which has the following characteristics:

(i) It has $2(t-q)$ rows with a single nonzero entry. In terms of the decoding graph, this means that there are $2(t-q)$ s_A nodes.

(ii) There are q rows with two nonzero entries. The first entry must be placed in any of the first $\frac{N}{2}$ columns of the matrix while the second one must be placed in any of the last $\frac{N}{2}$ columns. This ensures that s_C nodes connect both sides of the decoding graph.

(iii) The other $2(m-t)$ rows have x nonzero entries. In the decoding graph, these rows correspond to the s_B nodes, which remain the same as in the CSS structure used as a starting point.

3. Non-CSS QPCM

The quantum PCM of the proposed non-CSS code obtained using either of the aforementioned methods is calculated as

$$\begin{aligned} H_{Q_{\text{nonCSS}}} &= M'_d H_{\text{CSS}} = M'_d(H_z|H_x) \\ &= M'_d \begin{pmatrix} \tilde{H} & 0 \\ 0 & \tilde{G} \end{pmatrix} = (H''_z|H''_x), \end{aligned} \tag{11}$$

where $H_{\text{CSS}_{N\times 2N}}$ is defined as in (4), $\tilde{H}$ and $\tilde{G}$ are the parity check and generator matrices of a classical LDGM code, and M'_d is obtained using any of the construction methods.

As shown below, the construction in (11) satisfies the symplectic criterion given in (2). We assume that M'_d is obtained based on the second construction method presented above. With the goal of simplifying the proof, we write $[M'_d]_{m_r\times N}$ as the concatenation of two sub-matrices, i.e., $[M'_d]_{m_r\times N} = [M'_{\alpha_{m_r\times\frac{N}{2}}} M'_{\gamma_{m_r\times\frac{N}{2}}}]$. Substituting this expression into (11), we obtain

$$\begin{aligned} H_{Q_{\text{nonCSS}}} &= (H''_z|H''_x) \\ &= [M'_d]_{m_r\times N} H_{\text{CSS}_{N\times 2N}} \\ &= \left[M'_{\alpha_{m_r\times\frac{N}{2}}} M'_{\gamma_{m_r\times\frac{N}{2}}}\right] \begin{pmatrix} \tilde{H}_{\frac{N}{2}\times N} & 0_{\frac{N}{2}\times N} \\ 0_{\frac{N}{2}\times N} & \tilde{G}_{\frac{N}{2}\times N} \end{pmatrix} \\ &= \left(\left[M'_{\alpha_{m_r\times\frac{N}{2}}} \tilde{H}_{\frac{N}{2}\times N}\right]_{m_r\times N} \mid \left[M'_{\gamma_{m_r\times\frac{N}{2}}} \tilde{G}_{\frac{N}{2}\times N}\right]_{m_r\times N}\right), \end{aligned}$$

where $m_r = 2m-q$ is the total number of rows of M'_d.

Since for an LDGM code $\tilde{G}\tilde{H}^T = \tilde{H}\tilde{G}^T = 0$, we obtain

$$\begin{aligned} H''_z \star H''_x &= \left(H''_z H''^T_x + H''_x H''^T_z\right) \bmod 2 \\ &= \left(\left[M'_{\alpha_{m_r\times\frac{N}{2}}} \tilde{H}_{\frac{N}{2}\times N}\right]\left[M'_{\gamma_{m_r\times\frac{N}{2}}} \tilde{G}_{\frac{N}{2}\times N}\right]^T \right. \\ &\quad \left. + \left[M'_{\gamma_{m_r\times\frac{N}{2}}} \tilde{G}_{\frac{N}{2}\times N}\right]\left[M'_{\alpha_{m_r\times\frac{N}{2}}} \tilde{H}_{\frac{N}{2}\times N}\right]^T\right) \bmod 2 \\ &= \left(M'_{\alpha_{m_r\times\frac{N}{2}}} \underbrace{\tilde{H}_{\frac{N}{2}\times N}\tilde{G}^T_{N\times\frac{N}{2}}}_{0_{\frac{N}{2}\times\frac{N}{2}}} M'^T_{\gamma_{\frac{N}{2}\times m_r}} \right. \\ &\quad \left. + M'_{\gamma_{m_r\times\frac{N}{2}}} \underbrace{\tilde{G}_{\frac{N}{2}\times N}\tilde{H}^T_{N\times\frac{N}{2}}}_{0_{\frac{N}{2}\times\frac{N}{2}}} M'^T_{\alpha_{\frac{N}{2}\times m_r}}\right) \bmod 2 \\ &= 0_{m_r\times m_r}, \end{aligned}$$

proving that $H_{Q_{\text{non-CSS}}}$ satisfies the symplectic criterion.

4. Mixture of both methods

Another possibility to design non-CSS codes is to remove only a fraction of the s_A nodes that are used to generate s_C nodes in the first construction method. This procedure is identical to the second technique, with the sole exception that instead of removing the entire subset of q s_A syndrome nodes involved in the generation of the s_C nodes, only $l < q$ s_A nodes are removed. In the following section, we focus on codes obtained using the first two methods. Optimizing the performance of codes derived using this third approach may be of interest in our future work.

B. Decoding non-CSS QLDGM codes

Independently of the design method, decoding of the non-CSS quantum codes is performed utilizing the sum-product algorithm over the factor graph defined by the product $M'_d \times H_{\text{CSS}}$. Message passing will differ from that performed at a CSS QLDGM decoder, in the sense that both sides of the graph interact during the decoding process (messages are exchanged between nodes on the left and right subsections of the graph). This occurs because of the modified upper layer in the decoding graph of the non-CSS code, as shown in Fig. 5 for a non-CSS code derived using the first design method.

[10] By eliminating syndrome nodes we are decoding the same number of information qubits using less syndrome information.

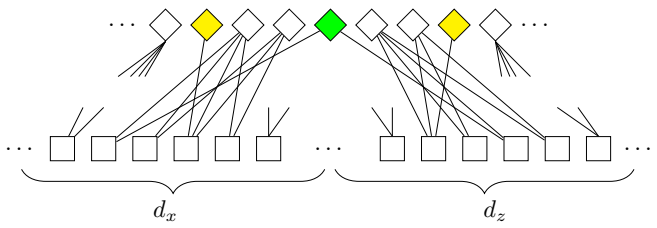

FIG. 5. Upper layer of the decoding graph associated with a non-CSS code obtained using method 1. The upper nodes represent the syndrome nodes, while the bottom nodes represent the d nodes (d_x and d_z denote the d nodes associated with each of the original separated CSS decoding subgraphs). The s_A nodes are represented in yellow and the s_C nodes are pictured in green.

C. Rate considerations

Both design methods allow a high degree of flexibility in terms of selecting the rate of the non-CSS quantum code. In fact, different non-CSS codes of the same rate can be obtained depending on the selected design method. Consider an arbitrary non-CSS code of quantum rate R_Q obtained based on the first design method taking as a starting point a CSS code of the same rate. A different non-CSS code of rate R_Q can be obtained by using the second design method, taking as a starting point a CSS code of lower rate. In fact, the second construction technique allows the construction of multiple matrices M'_d of equal size (which will lead to codes of equal quantum rate) by starting from different matrices M (and thus from CSS codes of different rates) and varying the number of s_A nodes that are removed, q. For instance, we could design a matrix M'_{d_1} of size $(2m_1 - q_1) \times N$ using two matrices M_1 of size $m_1 \times \frac{N}{2}$, and construct a second matrix M'_{d_2} of size $(2m_2 - q_2) \times N$ using two matrices M_2 of size $m_2 \times \frac{N}{2}$. If $2m_1 - q_1 = 2m_2 - q_2$, then both codes will have the same rate. Therefore, when the second method is utilized, code optimization depends on the choice of matrix M_d, as well as parameters m and q.

Notice that for a fixed $m \times N$ matrix M, by using the second design method and varying the value of q, we obtain different matrices M'_d of size $(2m - q) \times 2N$. This will result in non-CSS codes that encode $N - (2m - q)$ qubits into N qubits, and thus their rate will be

$$R_{Q-\text{non-CSS}} = \frac{N - (2m - q)}{N}, \tag{12}$$

which, since $q > 0$, is always higher than the rate of the CSS QLDGM code used as a starting point, given by

$$R_{Q-\text{CSS}} = \frac{N - 2m}{N}. \tag{13}$$

Notice that if the non-CSS code obtained from method 2 maintains the same performance as the original CSS code, this will be achieved with a higher rate. We introduce the parameter R_I, defined as

$$R_I = R_{Q-\text{non-CSS}} - R_{Q-\text{CSS}} = \frac{q}{N - 2m}, \tag{14}$$

to quantify the rate increase provided by the non-CSS scheme derived using the second construction method when compared to the original CSS scheme. This rate increase is determined by the value of q, which for the second design method represents the number of s_C nodes in the non-CSS decoding graph, as well as the number of nodes s_A removed from the decoding graph of the CSS code used as a starting point. The value of q will influence the performance of the resulting non-CSS code: intuitively, large increases in its value should lead to worsened performance, as the doping effect is reduced, but $q > 0$ allows for information exchange between the left and the right sides of the decoding graph, which should have a positive effect on performance. The impact of q on the proposed schemes is studied in the following section.

IV. SIMULATION RESULTS

In this section we compare the performance of the proposed non-CSS codes to that of the CSS codes of [22,23] when they are used over the X/Z channel and the depolarizing channel. The CSS codes in [22,23] have rate $R_Q \approx \frac{1}{4}$ and block length $N = 19014$, encoding $k = 4752$ qubits into N qubits. Matrix P, of size 9507×9507, has the same degree distribution as its transpose P^T, and corresponds to a rate $\frac{1}{2}$ classical LDGM code. Hence, both $\tilde{G}$ and $\tilde{H}$ have size 9507×19014. Matrix M, which is used to transform $\tilde{G}$ and $\tilde{H}$ into G and H, is full rank, low density, and has size 7131×9507. Results are depicted in terms of either the qubit error rate (QBER) or the word error rate (WER). QBER represents the fraction of qubits that experience an error, while WER is the fraction of transmitted blocks that have at least one qubit error. We use the QBER metric for some of our simulations because it can be estimated with high confidence faster than the WER, which is helpful in shortening the required simulation time of some of our codes.

First, the codes are simulated over the iid X/Z channel model of [12], where Z and X errors are modeled as independent events identically distributed according to the flip probability f_m. We begin by using as a starting point the family of CSS codes of the first proposed structure in [22,23], which are individual regular LDGM codes. The simplicity of the channel model and of the code structure allows us to assess, in a rapid and efficient manner, the values of q that optimize the performance of the non-CSS construction. Using these values of q, we will repeat the simulations when the CSS codes used as the starting point consist of the parallel concatenation of regular LDGM codes, as described in [22,23]. Both design methods are utilized to obtain the resulting non-CSS codes. Finally, we repeat the same process for the depolarizing channel.

A. The iid X/Z channel: Non-CSS codes based on individual regular LDGM codes

For these simulations, the CSS code utilized as a starting point is an individual regular LDGM code. Matrix P is generated pseudorandomly and corresponds to a regular (X, X) LDGM code. M is characterized by the parameter values[11]

[11]The fractional number 8.72 represents the fact that 72% of the s_B nodes will have degree 9 while 28% of them will have degree 8.

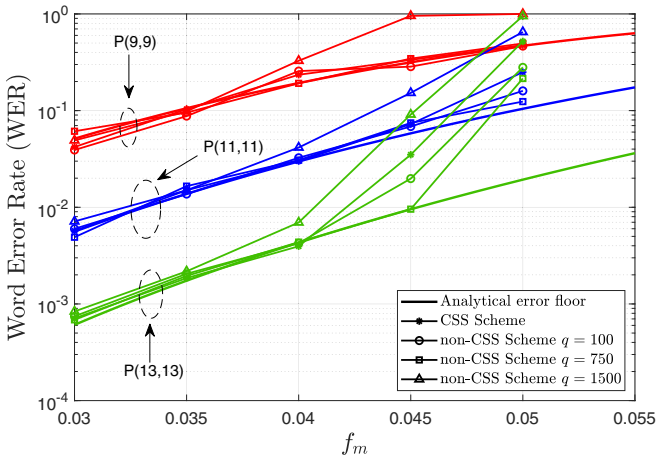

FIG. 6. Simulated WER for non-CSS QLDGM codes based on individual regular LDGM codes obtained using the first design method when they are applied over the flipping channel. f_m is the probability of error (iid) in the binary representation of the X and Z containing operators.

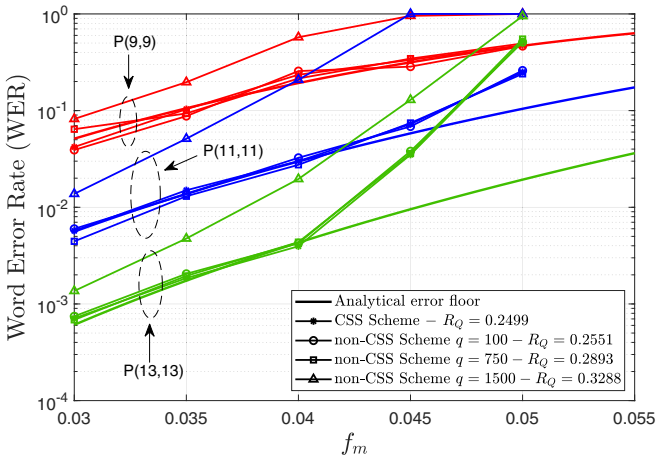

FIG. 7. Simulated WER for non-CSS QLDGM codes based on individual regular LDGM codes obtained using the second design method when they are applied over the flipping channel. f_m is the probability of error (iid) in the binary representation of the X and Z containing operators.

$M(3; 1, 8.72)$ and $t = 4361$. The degrees[12] of P (and P^T) are varied between (9,9) and (13,13). In the figures that follow, f_m is the probability of error (iid) in the binary representation of the X and Z containing operators and the analytical error floors of the LDGM codes have been obtained as shown in [47].

1. Non-CSS codes derived using method 1

As explained before, non-CSS codes based on this method are obtained by transforming q s_A nodes of the CSS decoding graph into degree-2 s_C nodes. All the non-CSS codes obtained in this manner have the same quantum rate ($R_Q = \frac{1}{4}$). Matrix M_d is built from M as shown in (10). The rest of the underlying components of the non-CSS configuration are identical to those of the CSS designs.[13] We test three different configurations, $q = [100, 750, 1500]$, for different degrees of the P matrices. The results are shown in Fig. 6. The simulation results displayed in this figure show how for $q = 100$ and $q = 750$, regardless of the degree of the pseudorandom matrix P, the performance of the proposed non-CSS codes is slightly better than that of the CSS schemes used as a starting point. This performance improvement is most notable for the case of $P(13, 13)$, where, as f_m increases, the non-CSS schemes operate closer than the original CSS code to the analytical error floor. However, for $q = 1500$, the performance of the non-CSS code is worse than that of the CSS scheme. The best results are observed when the non-CSS code is designed using $q = 750$.

2. Non-CSS codes derived using method 2

As explained before, non-CSS codes based on this method are obtained by removing q syndrome nodes from the decoding graph generated by the first method, specifically the q s_A nodes connected to a d_A node that is linked to an s_C node. Note that removing syndrome nodes will result in codes with different quantum rates. As previously, we test the values $q = [100, 750, 1500]$ for different degrees of the P matrices. The results are shown in Fig. 7.

The results displayed in Fig. 7 are once again consistent regardless of the degree of P. They show how for the two smaller q values, 100 and 750, the non-CSS codes yield the same performance as the CSS schemes of [22]. For $q = 1500$, the performance of the non-CSS codes is significantly worse. For instance, at a value of $f_m = 0.04$, the word error rate for the $q = 1500$ schemes is around an order of magnitude higher than for the $q \leqslant 750$ non-CSS schemes. This corroborates our intuition that there is an optimum range of values for q, and that utilizing values outside of that range degrades the code performance. Based on the results in Fig. 7, the optimal value of q should be a small percentage of the total number of syndrome nodes m, $q < 0.1 \times m$ ($\frac{1500}{14262} \approx 0.1$).

Table I presents the WER performance for the codes considered in Fig. 7 when matrix P has degrees (13,13) and $f_m = 0.03$. The CSS scheme has rate $R_{Q-\mathrm{CSS}} = \frac{19014-2\times7131}{19014} = 0.2499$. Results for the $q = 1500$ non-CSS scheme have not been included, since the performance of original CSS scheme

TABLE I. Comparison between the codes shown in Fig. 7. The WER data included in the table correspond to the codes whose P matrix has degrees (13,13).

Code Type	R_Q	q	WER at $f_m = 0.03$
CSS	0.2499		7.12×10^{-3}
non-CSS	0.2551	100	7.26×10^{-3}
non-CSS	0.2893	750	7.13×10^{-3}

[12]Given that the degree distribution of P^T and P is the same, we refer to them indistinctly throughout this paper.

[13]The only difference between the CSS and non-CSS codes lies in matrix M'_d. The rate-$\frac{1}{2}$ classical LDGM code and, therefore, matrices P, $\tilde{G}$, and $\tilde{H}$ are identical.

is better. As with the first method, the best result here is obtained for the $q = 750$ non-CSS scheme, since it achieves a performance similar to that of the CSS code with a 15% higher rate.

Overall, the results obtained for both design methods illustrate that our proposed non-CSS codes, regardless of the design method, outperform QLDGM CSS codes when individual regular LDGM codes are utilized. The first design method provides a way to obtain non-CSS codes that outperform CSS codes of the same rate. The second method enables us to construct non-CSS codes with error correcting capabilities comparable to those of lower-rate CSS schemes. Therefore, to design non-CSS codes of a fixed rate, we could apply the first design method to a CSS code of the same rate, or we could start with a CSS code of lower rate and apply the second method. In this case, we should carefully choose the rate of the original CSS code to obtain the desired rate in the designed non-CSS code.

B. The iid X/Z channel: Non-CSS codes based on the parallel concatenation of LDGM codes

As mentioned in Sec. II D, regular LDGM codes used in classical channels present error floors. Fortunately, these error floors can be substantially lowered if we use the parallel concatenation of two regular LDGM codes. In [20,21] CSS quantum codes based on the use of single regular LDGM codes were shown to also lead to error floors. Inspired by the good performance displayed by parallel LDGM codes in the classical domain, a scheme based on the parallel concatenation of LDGM codes was designed and applied to the iid X/Z channel in [22,23]. Similarly to the classical scenario, CSS codes built based on the parallel structure display lower error floors and better performance overall.

We now repeat the study carried out in the previous subsection, but using the parallel concatenation of regular LDGM codes as the starting point to derive our proposed non-CSS codes. In [23], the parallel LDGM structure $P[(y_1, y_1); (y_2, z_2)]$[14] was optimized for various configurations of matrices M. For our analysis we use the structure with the lowest degrees in [23]: $P[(8, 8); (3, 60)]$. Although performance is better for larger degrees (y_2, z_2), codes with smaller degrees require less processing and simulation time. To ease simulation requirements even further, results in this subsection are presented in terms of the QBER. The best configuration for $P[(y_1, y_1); (y_2, z_2)]$ in [23] will be used later for the simulations over the depolarizing channel.

As we did for individual regular LDGM codes in the previous subsection, we analyze the performance of the non-CSS codes obtained using the two design methods proposed in Sec. III. The CSS code used as a starting point is the same for both methods, and utilizes the same M as in the scenario for individual regular LDGM codes: $M_{7131\times 9507}(3; 1, 8.72)$ and $t = 4361$. We consider the values of $q = \{100, 500, 1000, 1500\}$. As before, f_m is the probability of error (iid) in the binary representation of the X and Z containing operators and the analytical error floors of the LDGM codes have been obtained as shown in [47].

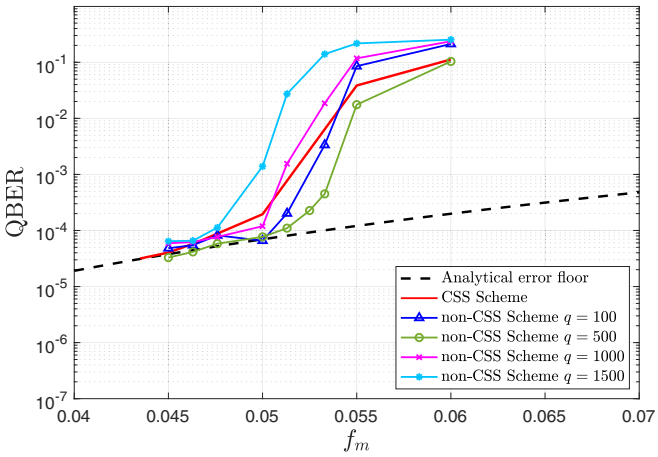

FIG. 8. Simulated QBER for a $R_Q = \frac{1}{4}$ CSS code and $R_Q = \frac{1}{4}$ non-CSS codes derived using the first design methodology. The underlying classical LDGM code is the same for all the codes and has degree distribution $P[(8, 8); (3, 60)]$. f_m is the probability of error (iid) in the binary representation of the X and Z containing operators.

Figure 8 presents the performance of the $R = \frac{1}{4}$ original CSS code and of the $R = \frac{1}{4}$ non-CSS codes obtained by applying the first design method. As shown in the figure, the non-CSS codes derived by selecting $q = 100$ and $q = 500$ outperform the original CSS structure. For $q > 500$, performance of the non-CSS codes gradually deteriorates, with the result for $q = 1500$ being substantially worse than that of the original CSS code.

Figure 9 shows the results for the non-CSS codes derived using the second proposed design method. The curves shown

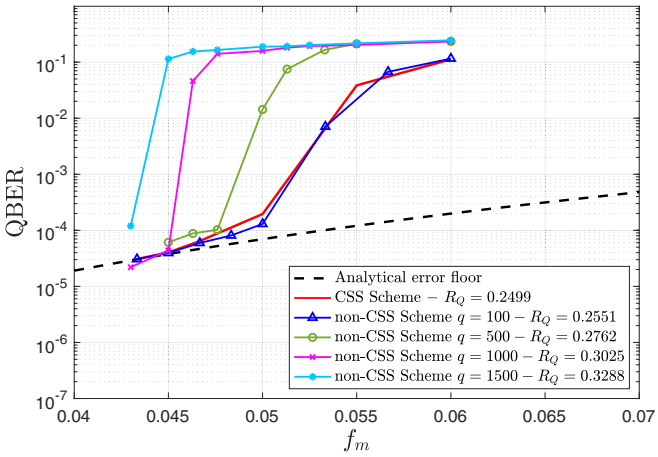

FIG. 9. Simulated QBER for a $R_Q = \frac{1}{4}$ CSS code and non-CSS codes of different rate derived using the second design methodology. The underlying classical LDGM code is the same for all the codes and has degree distribution $P[(8, 8); (3, 60)]$. f_m is the probability of error (iid) in the binary representation of the X and Z containing operators.

[14]The notation $P[(y_1, y_1); (y_2, z_2)]$ indicates the degree distributions of the constituent regular LDGM codes utilized in the parallel concatenation. The parameters of the second code were typically chosen as $z_2 = 20y_2$.

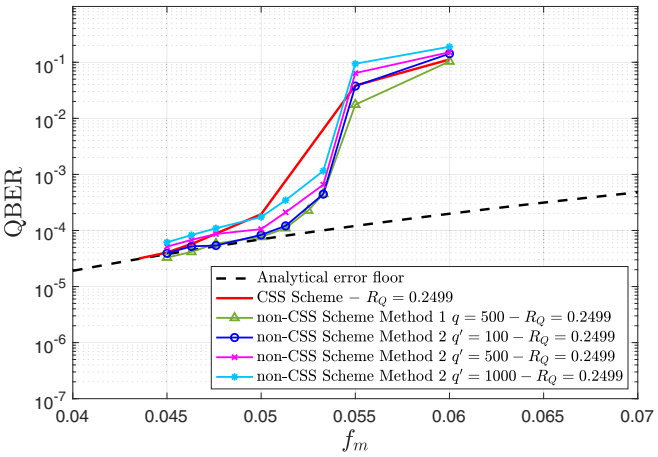

FIG. 10. Simulated QBER for a $R_Q = \frac{1}{4}$ CSS code and $R_Q = \frac{1}{4}$ non-CSS codes constructed using the two proposed design methods. q denotes how many s_A nodes have become s_C nodes in the upper layer of the decoding graph when method 1 is applied. q' denotes how many s_A nodes have become s_C nodes and how many s_A nodes have been removed from the upper layer of the decoding graph when method 2 is used. The underlying classical LDGM code is the same for all the codes and has degree distribution $P[(8, 8); (3, 60)]$. f_m is the probability of error (iid) in the binary representation of the X and Z containing operators.

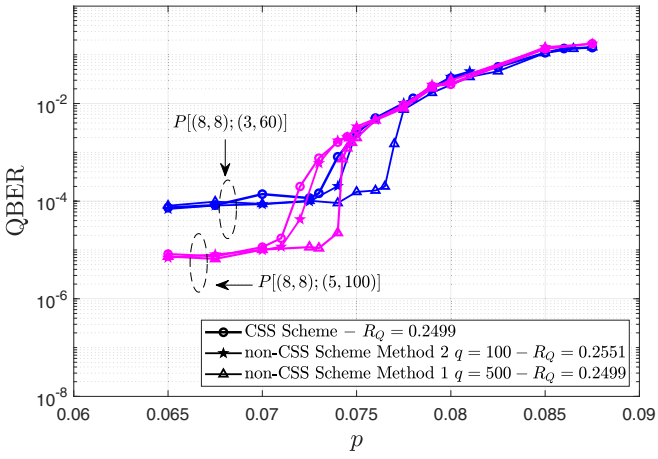

FIG. 11. Simulated QBER for three quantum codes: An $R_Q = \frac{1}{4}$ CSS QLDGM code, an $R_Q = \frac{1}{4}$ non-CSS QLDGM code obtained using the first proposed method and $q = 500$, and an $R_Q = 0.255$ non-CSS QLDGM code derived via the second design technique with $q = 100$. All three codes have been simulated for two different degree distributions of the underlying parallel concatenated LDGM scheme: $P[(8, 8); (3, 60)]$ and $P[(8, 8); (5, 100)]$. p is the depolarizing probability.

in this figure portray how the performance of the non-CSS codes is drastically degraded as we increase the value of q. This effect is much more noticeable than for codes based on a single regular LDGM code. In fact, the only value for which the non-CSS configuration based on parallel concatenation matches the performance of the original CSS scheme is $q = 100$, whereas in Fig. 7 we could see that schemes based on individual regular LDGM codes matched the performance of the original CSS code at least up to $q = 750$. In essence, although the s_A node elimination step explained in Sec. III A still yields a small benefit, this is much lower than for the case of single regular LDGM codes.

Earlier in this paper we mentioned that the second non-CSS code design methodology could be used not only to obtain higher-rate non-CSS codes with performance similar to that of lower-rate CSS codes, but also to generate different non-CSS codes of a fixed rate by varying the CSS codes used as starting points and selecting the appropriate value of q. Analyzing the performance of $R_Q = \frac{1}{4}$ non-CSS codes obtained in this manner, and comparing the results to those obtained using the first design method, will allow us to determine which design technique yields codes with better performance.

This comparison is shown in Fig. 10, where various $R_Q = \frac{1}{4}$ non-CSS codes obtained using the second design method are compared to the best $R_Q = \frac{1}{4}$ code generated by the first design method ($q = 500$ in Fig. 8) and to the original $R_Q = \frac{1}{4}$ CSS structure. The non-CSS codes generated by method 2 are derived by building matrices M'_d using matrices M of different size, which are designed according to the analysis conducted in [23]. We introduce parameter q' to refer to the values of q used to derive these codes and to distinguish them from the code built using the first method.

The $q' = 100$ code uses $M = M_{7181\times9507}(3; 1, 8.56)$ with $t = 4361$ (using this M for a CSS QLDGM scheme would yield a code of rate $R_Q = 0.244$). The $q' = 500$ code uses $M = M_{7381\times9507}(3; 1, 8.36)$ with $t = 4561$ (using this M for a CSS QLDGM scheme would yield a code of rate $R_Q = 0.2236$). The $q' = 1000$ code uses $M = M_{7631\times9507}(3; 1, 8.27)$ with $t = 4761$ (using this M for a CSS QLDGM scheme would yield a code of rate $R_Q = 0.1973$).

Figure 10 shows that the non-CSS codes designed using the second method outperform the CSS scheme for all values of q'. However, the $q = 500$ non-CSS code designed using the first method is still better than any of the aforementioned codes, although the performance of the $q' = 100$ non-CSS code is only slightly worse.

In this subsection, we have discussed results for the iid X/Z channel, where the X and Z operators are modeled independently. Our analysis has allowed us to determine the design methodologies and the values of q that lead to the best performance. Making use of this knowledge, we will now consider a more realistic quantum channel model, the depolarizing channel.

C. Depolarizing channel

We now focus on the depolarizing channel introduced in Sec. II A 3. We consider the best non-CSS codes obtained before for the iid X/Z channel: the $q = 500$, $R_Q = \frac{1}{4}$ non-CSS code obtained using the first design method and the $q = 100$, $R_Q = 0.255$ non-CSS code obtained using the second method. Figure 11 shows the QBER of the aforementioned codes for two different degree distributions of the parallel concatenated LDGM scheme. The curves associated with the original CSS codes are also included.

Similarly to the results for the iid X/Z channel, Fig. 11 portrays how the $R_Q = 0.255$ non-CSS codes display perfor-

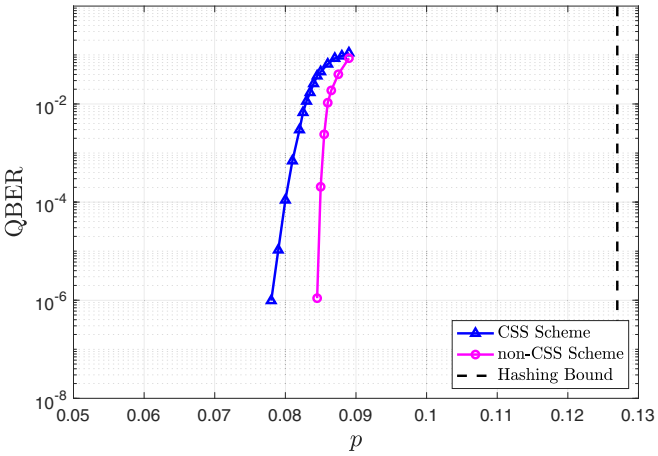

FIG. 12. Simulated QBER for the best $R = \frac{1}{4}$ CSS QLDGM in [23] and the best non-CSS QLDGM designed in this paper. The hashing bound is also shown. The codes are based on the parallel concatenation of regular LDGM codes with degree distribution $P[(8, 8); (8, 160)]$. p is the depolarizing probability.

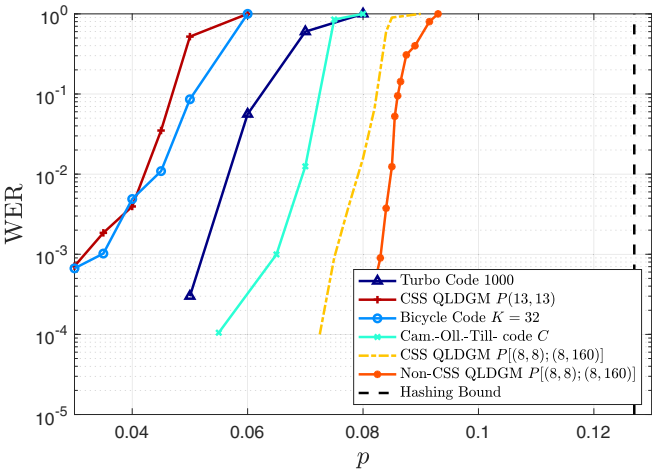

FIG. 13. Word error rate for various quantum codes when they are applied over the depolarizing channel. The hashing bound is also included. p is the depolarizing probability.

mance very close to that of the $R_Q = \frac{1}{4}$ CSS schemes. This phenomenon can be explained for both the depolarizing and the iid X/Z channels by the nature of the non-CSS construction, which adds a very small number of edges and removes very few syndrome nodes from the original CSS factor graph. As in the case of the X/Z channel model, the $q = 500, R_Q = \frac{1}{4}$ non-CSS codes designed utilizing method 1 also outperform the CSS code.

An important observation, which is reflected in Fig. 11, is that the results are consistent regardless of the degrees (y_2, z_2) that are chosen. This is significant, since increasing y_2 and z_2 enables us to lower the error floor of the QLDGM code. As shown in [22,23] for CSS codes, we can see from Fig. 11 that selecting P_2 with larger degrees lowers the error floor at the expense of a worse decoding threshold.

1. Distance to the theoretical limit

The most effective way to characterize the performance improvement of our proposed non-CSS codes is to measure their gap with respect to the hashing bound. For this comparison, we will employ the design parameters that yield the best possible code. Such a scheme is obtained by using the first construction method with $t = 5000, q = 500$, $M(3; 1, 11.04)$, and a parallel concatenated LDGM code of degree distribution $P[(8, 8); (8, 160)]$.

Figure 12 depicts the performance of this non-CSS scheme as well as that of the CSS code used as a starting point for the design, which is the best CSS code proposed in [23]. The hashing bound for $R_Q = \frac{1}{4}$ is also shown. We compute the distance to the hashing bound δ as defined in (1), knowing that the noise limit for $R_Q = \frac{1}{4}$ is $p^* \approx 0.127$, and taking $p_{\text{CSS}} = 0.0825$ and $p_{\text{non-CSS}} = 0.0865$ as the depolarizing probabilities at which the CSS and non-CSS codes enter the waterfall region, respectively. This yields $\delta_{\text{CSS}} = 1.873$ dB and $\delta_{\text{non-CSS}} = 1.668$ dB. In other words, the non-CSS scheme is about 0.2 dB closer to the hashing bound. Thus, in terms of overall performance over the depolarizing quantum channel, the non-CSS codes proposed in this article outperform existing CSS techniques.

2. Comparison with existing QLDPC schemes

We close out this section by studying how the proposed non-CSS codes measure up against other QLDPC schemes in the literature. For this purpose, we conduct two different comparisons. We begin by comparing the codes in Fig. 12 to other quantum codes of rate $\frac{1}{4}$. Then, we study how nonconventional CSS decoders capable of exploiting the correlation between X and Z operators match up to our non-CSS construction. The first comparison is shown in Fig. 13, which includes the performance for the following codes:

(i) The CSS QLDGM code based on a single regular LDGM code from [21]. The degree distribution of the underlying classical LDGM code is $P(13, 13)$. The block length of the code is 19 014.

(ii) The $K = 32$ bicycle code of block length 19 014 introduced by MacKay *et al.* in [12].

(iii) The quantum serial turbo code of [5], with block length 4000.

(iv) The non-CSS concatenated code (code C) from [34], with block length 138 240.

Figure 13 shows that our proposed non-CSS QLDGM codes outperform existing quantum turbo codes and previously proposed quantum LDPC codes.

As explained in the introduction, CSS decoders capable of improving performance by exploiting the correlation that exists between X and Z operators over the depolarizing channel have been proposed in the literature. Some of the earliest work on this topic was conducted in [29], where a set of modified BP decoders for CSS codes that reintroduce X/Z correlations are proposed. The most notable of these decoding schemes is known as the random perturbation decoder. In [48], further work on the topic of modified BP decoders for CSS codes is conducted and two novel CSS decoders are presented: the adjusted decoder and the augmented decoder. The adjusted decoder attempts to reintroduce the correlations between X and Z operators, neglected by a standard binary

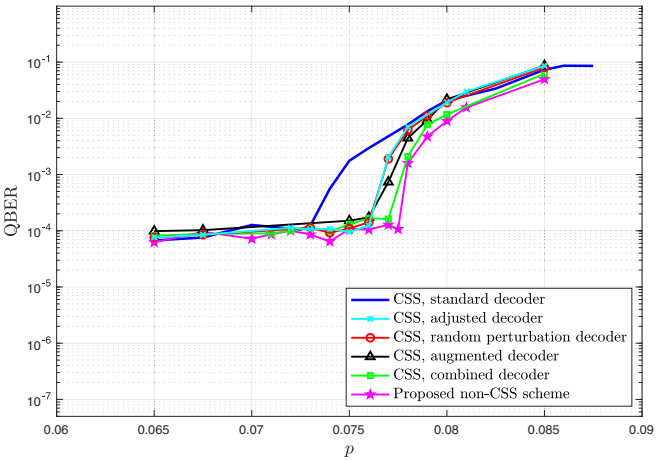

FIG. 14. Simulated QBER for different types of modified CSS BP decoders over the depolarizing channel. Results for the separate decoding of X and Z errors and for our proposed non-CSS scheme are also included. p is the depolarizing probability of the depolarizing channel.

BP decoder, by adjusting prior probabilities. The augmented decoder operates by appending a specific subset of rows H_δ from the original PCM H into an exact copy of that matrix, resulting in an augmented PCM $H_A = \binom{H}{H_\delta}$ which is associated with a larger factor graph over which the decoding algorithm is run. Also in [48], the adjusted and augmentation techniques are combined to form a new CSS decoder, which is called the combined decoder.

In Fig. 14, we compare the performance of the random perturbation decoder, as well as that of the adjusted, augmented, and combined decoders, to that of our proposed non-CSS codes when they are applied over the depolarizing channel. The underlying LDGM code and CSS configuration is the same as in most of our previous simulations: the classical code is defined by the parallel concatenation $P[(8, 8); (3, 60)]$ and the CSS construction is achieved using $M(3; 1, 8.72)$ and $t = 4361$. Performance of a typical CSS decoder (separate decoding of X and Z operators) is also included in Fig. 14. The non-CSS scheme is obtained from the CSS code by using the first design method and setting $q = 500$, which is the parameter configuration that produced the best results in our earlier simulations. Both the CSS code and the non-CSS configuration have rate $R_Q = \frac{1}{4}$.

As shown in Fig. 14, all of the modified CSS BP decoders yield performance improvements when compared to the generic CSS decoding scheme. Nonetheless, they are all outperformed by our proposed non-CSS construction, despite the close proximity between the QBER curves of the combined decoder and the non-CSS scheme. Importantly, we must note that although these modified decoders achieve performance close to that of our proposed scheme, they require specific modifications to the decoding algorithm that result in a higher decoding complexity. This increase in decoding complexity arises from the fact that the modified CSS schemes require the execution of a standard decoder prior to their application. Specifically, they rely on the failure of a standard decoder to find the correct estimate of the syndrome, to then apply modifications to the factor graph according to the channel error estimate produced by the failed standard decoder. Once these modifications have been made (either to the *a priori* probabilities or to the factor graph itself) decoding is reattempted. In some instances, decoding must be reattempted multiple times before the correct syndrome estimate is found. Moreover, the augmented decoder and the combined decoder operate over larger factor graphs, which further increases the decoding complexity.

Given the nature of our proposed construction, decoding never has to be reattempted, and so the complexity of our decoder is essentially the same as that of a standard CSS decoder. Therefore, the proposed non-CSS scheme outperforms the aforementioned nonconventional CSS decoders while displaying a lower decoding complexity.

V. CONCLUSION

We have introduced a technique to design non-CSS quantum codes based on the use of the generator and parity check matrices of LDGM codes. The proposed methods are based on modifying the upper layer of the decoding graph in CSS QLDGM constructions. The simplicity of the proposed scheme ensures that the high degree of flexibility in the choice of the quantum rate and the block length for the CSS code utilized as a starting point is translated to the non-CSS design. Compared with quantum CSS codes based on the use of LDGM codes, the proposed non-CSS scheme is 0.2 dB closer to the hashing bound in the depolarizing channel, and outperforms all other existing quantum codes of comparable complexity.

ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Economy and Competitiveness through the ADELE (Grant No. PID2019-104958RB-C44) and CARMEN (Grant No. TEC2016-75067-C4-3-R) projects. J.E. is funded by a Basque Government predoctoral research grant (Hezkuntza, Hizkuntza Politika Eta Kultura Saila, Eusko Jaurlaritza).

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, in *Proceedings of ICC '93: IEEE International Conference on Communications* (IEEE, Piscataway, NJ, 1993).
[2] R. G. Gallagher, IRE Trans. Inf. Theory **8**, 21 (1962).
[3] D. J. C. MacKay, IEEE Trans. Inf. Theory **45**, 399 (1999).
[4] S.-Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, IEEE Commun. Lett. **5**, 58 (2001).
[5] D. Poulin, J. Tillich, and H. Ollivier, IEEE Trans. Inf. Theory **55**, 2776 (2009).
[6] M. Wilde, M. H. Hsieh, and Z. Babar, IEEE Trans. Inf. Theory **60**, 1203 (2014).

[7] Z. Babar, S. X. Ng, and L. Hanzo, IEEE Trans. Veh. Technol. **64**, 866 (2015).
[8] H. V. Nguyen, Z. Babar, D. Alanis, P. Botsinis, D. Chandra, S. X. Ng, and L. Hanzo, IEEE Access **4**, 10194 (2016).
[9] D. Chandra, Z. Babar, S. X. Ng, and L. Hanzo, IEEE Access **7**, 52712 (2019).
[10] J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, Entropy **21**, 633 (2019).
[11] J. Etxezarreta Martinez, P. M. Crespo, and J. Garcia-Frias, Entropy **21**, 1133 (2019).
[12] D. J. Mackay, G. Mitchison, and P. L. McFadden, IEEE Trans. Inf. Theory **50**, 2315 (2004).
[13] T. R. Oenning and J. Moon, IEEE Trans. Magn. **37**, 737 (2001).
[14] J. Garcia-Frias and W. Zhong, IEEE Commun. Lett. **7**, 266 (2003).
[15] W. Zhong, H. Lou, and J. Garcia-Frias, in *Proceedings of the 2003 International Conference on Image Processing* (IEEE, Piscataway, NJ, 2003).
[16] I. Granada, P. M. Crespo, and J. Garcia-Frias, EURASIP J. Wireless Commun. Networking **2019**, 11 (2019).
[17] I. Granada, P. M. Crespo, and J. Garcia-Frias, Entropy **21**, 378 (2019).
[18] D. Gottesman, Phys. Rev. A **54**, 1862 (1996).
[19] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, IEEE Access **3**, 2492 (2015).
[20] H. Lou and J. Garcia-Frias, in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications* (IEEE, Piscataway, NJ, 2005).
[21] H. Lou and J. Garcia-Frias, in *4th International Symposium on Turbo Codes and Related Topics* (IEEE, Piscataway, NJ, 2006).
[22] J. Garcia-Frias and K. Liu, in *42nd Annual Conference on Information Sciences and Systems* (IEEE, Piscataway, NJ, 2008).
[23] K. Liu and J. Garcia-Frias, in *48th Annual Allerton Conference on Communication, Control, and Computing* (IEEE, Piscataway, NJ, 2010).
[24] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).
[25] A. Steane, Proc. R. Soc. London, Ser. A **452**, 2551 (1996).
[26] D. Maurice, J.-P. Tillich, and I. Andriyanova, in *IEEE International Symposium on Information Theory* (IEEE, Piscataway, NJ, 2013).
[27] C. Hadley, A. Serafini, and S. Bose, Phys. Rev. A **72**, 052333 (2005).
[28] G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. **104**, 050504 (2010).
[29] D. Poulin and Y. Chung, Quantum Information & Computation **8**, 987 (2008).
[30] S. Bravyi, M. Suchara, and A. Vargo, Phys. Rev. A **90**, 032326 (2014).
[31] P. Baireuther, T. E. O'Brien, B. Tarasinski, and C. W. J. Beenakker, Quantum **2**, 48 (2018).
[32] S. Varsamopoulos, B. Criger, and K. Bertels, Quantum Sci. Technol. **3**, 015004 (2018).
[33] B. Criger and I. Ashraf, Quantum **2**, 102 (2018).
[34] T. Camara, H. Ollivier, and J.-P. Tillich, in *IEEE International Symposium on Information Theory* (IEEE, Piscataway, NJ, 2007).
[35] D. Maurice, J.-P. Tillich, and I. Andriyanova, in *IEEE Information Theory Workshop* (IEEE, Piscataway, NJ, 2012).
[36] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Phys. Rev. A **54**, 3824 (1996).
[37] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, IEEE Trans. Inf. Theory **3**, 146 (2015).
[38] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, IEEE Access **3**, 146 (2015).
[39] W. Zhong, H. Chai, and Javier Garcia-Frias, in *Proceedings of the 2005 International Symposium on Information Theory* (IEEE, Piscataway, NJ, 2005).
[40] R. M. Tanner, IEEE Trans. Inf. Theory **27**, 533 (1981).
[41] N. Wiberg, H.-A. Loeliger, and R. Kotter, Eur. Trans. Telecommun. **6**, 513 (1995).
[42] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, IEEE Trans. Inf. Theory **47**, 498 (2001).
[43] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kauffman, San Francisco, 1988).
[44] N. Wiberg, Codes and decoding on general graphs, Ph.D. thesis, Linköping University, 1996.
[45] S. ten Brink, in *Proceedings of the 2001 IEEE International Symposium on Information Theory* (IEEE, Piscataway, NJ, 2001).
[46] K. Liu and J. Garcia-Frias, in *Proceedings of the 2010 48th Annual Allerton Conference on Communication, Control, and Computing* (IEEE, Piscataway, NJ, 2010).
[47] K. Liu and J. Garcia-Frias, in *43rd Annual Conference on Information Sciences and Systems* (IEEE, Piscataway, NJ, 2009).
[48] A. Rigby, J. C. Olivier, and P. Jarvis, Phys. Rev. A **100**, 012330 (2019).

Correction: An incorrect page number and the underlying doi for Ref. [10] have been fixed.