# Belief Propagation Decoding of Quantum LDPC Codes with Guided Decimation

Hanwen Yao*†, Waleed Abu Laban‡, Christian Häger‡, Alexandre Graell i Amat‡, and Henry D. Pfister*†§
*Duke Quantum Center, Duke University, Durham, NC, USA
†Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA
‡Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden
§Department of Mathematics, Duke University, Durham, NC, USA

2024 IEEE International Symposium on Information Theory (ISIT) | 979-8-3503-8284-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISIT57864.2024.10619083

***Abstract*—Quantum low-density parity-check (QLDPC) codes have emerged as a promising technique for quantum error correction. A variety of decoders have been proposed for QLDPC codes and many utilize belief propagation (BP) decoding in some fashion. However, the use of BP decoding for degenerate QLDPC codes is known to have issues with convergence. These issues are typically attributed to short cycles in the Tanner graph and error patterns with the same syndrome due to code degeneracy.**

**In this work, we propose a decoder for QLDPC codes based on BP guided decimation (BPGD), which has been previously studied for constraint satisfaction and lossy compression problems. This decimation process is applicable to both binary and quaternary BP and it involves sequentially freezing the value of the most reliable qubits to encourage BP convergence. We find that BPGD significantly reduces the BP failure rate due to non-convergence, achieving performance on par with BP with ordered statistics decoding and BP with stabilizer inactivation, without the need to solve systems of linear equations. To explore how and why BPGD improves performance, we discuss several interpretations of BPGD and their connection to BP syndrome decoding.**

## I. Introduction

In a quantum computing system, error correction is an essential building block to protect fragile quantum information against noise. The general framework of quantum stabilizer codes has been studied extensively [1]–[3]. Using this framework, various quantum error-correcting codes have been constructed over the past two decades including toric codes [4], [5], surface codes [6]–[10], and various quantum low-density parity-check (QLDPC) codes [11]–[17]. Among them, QLDPC codes provide a promising direction because they support multiple logical qubits and their low-weight stabilizers allow reliable syndrome measurement in practice.

For classical communication, low-density parity-check (LDPC) codes are decoded with the belief propagation (BP) algorithm [18], which has low complexity and can provide good performance for code rates close to the channel capacity [19]–[22]. However, in the quantum scenario, the performance of BP decoding based on syndrome measurement is hindered by cycles in the Tanner graph and code degeneracy [23]–[25]. Since the initial application of BP to decode QLDPC codes by Poulin and Chung [23], significant efforts have been made to enhance its performance. Various methods have been proposed to modify the BP decoding process itself, including random perturbation [23], enhanced feedback [26], grouping check nodes as super nodes [24], parity-check matrix augmentation [27], neural BP [28]–[30], generalized BP [31], and adaptive BP with memory [32]. Alternatively, post-processing methods have also been explored to improve performance. In [16], when BP fails to converge, they use ordered statistics decoding (OSD) to construct a syndrome-matching error pattern based on the soft information provided by BP. This is called the BP-OSD algorithm. Another post-processing approach introduced in [33] involves iteratively running BP with stabilizer inactivation (BP-SI).

In this work, we improve the BP decoding performance for QLDPC codes by combining it with guided decimation. The term "decimation" refers to the process of sequentially fixing variables to hard decisions during iterative decoding [34], [35]. In the proposed belief propagation guided decimation (BPGD) algorithm, we sequentially freeze the most reliable qubit based on its soft information after a certain number of BP iterations. Despite its simplicity, we show that BPGD achieves performance on par with both order-0 BP-OSD and BP-SI. Notably, BPGD exhibits lower complexity than BP-OSD and comparable complexity to BP-SI, without the need to solve any systems of linear equations. To better understand how guided decimation improves BP performance for QLDPC codes, we also discuss some interpretations of BPGD and their connection to the BP syndrome decoding problem.

## II. Preliminaries

### A. Stabilizer Formalism

An $[[n, k]]$ quantum stabilizer code is an error correction code designed to protect $k$ logical qubits with $n$ physical qubits against noise. For a single qubit, its pure quantum state is represented as a unit vector in the two-dimensional Hilbert space $\mathbb{C}_2$. The Pauli operators for a single qubit system are defined as the $2 \times 2$ complex Hermitian matrices

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \sigma_y = \imath \sigma_x \sigma_z,$$

where $\imath = \sqrt{-1}$. For an $n$-qubit system, we are working in the $n$-fold Kronecker product of the two-dimensional Hilbert space $\mathbb{C}_2^{\otimes n}$. Given two length-$n$ binary vectors $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ in $\mathbb{F}_2^n$, we define the $n$-fold Pauli operator $D(a, b)$ as

$$D(a, b) = \sigma_x^{a_1} \sigma_z^{b_1} \otimes \sigma_x^{a_2} \sigma_z^{b_2} \otimes \ldots \otimes \sigma_x^{a_n} \sigma_z^{b_n}. \quad (1)$$

It follows that the Pauli operators $\imath^k D(a,b)$ with $a,b \in \mathbb{F}_2^n$ and an overall phase $\imath^k$ with $k \in \{0,1,2,3\}$ form the $n$-qubit Pauli group, denoted as $\mathcal{P}_n$, with the multiplication rule

$$D(a,b)D(a',b') = (-1)^{a'b^T + b'a^T} D(a',b')D(a,b). \quad (2)$$

The *symplectic inner product* between length-$2n$ binary vectors $(a,b)$ and $(a',b')$ is defined by

$$\langle (a,b),(a',b')\rangle_s \triangleq (a',b') \underbrace{\begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}}_{\text{denoted by } \Lambda} (a,b)^T \bmod 2. \quad (3)$$

A quantum stabilizer code $\mathcal{C}$ with $n$ physical qubits is defined via a commutative subgroup $\mathcal{S} \subseteq \mathcal{P}_n$ referred to as the *stabilizer group* with $-I_2^{\otimes n} \notin \mathcal{S}$. The Pauli operators in $\mathcal{S}$ are called the *stabilizers*. The code space consists of all states in $\mathbb{C}_2^{\otimes n}$ stabilized by $\mathcal{S}$ as

$$\mathcal{C} = \{|\psi\rangle \in \mathbb{C}_2^{\otimes n} : M|\psi\rangle = |\psi\rangle,\ \forall M \in \mathcal{S}\}. \quad (4)$$

The code space $\mathcal{C}$ has dimension $k$ if $\mathcal{S}$ has $n-k$ independent generators. The *weight* of a Pauli operator in $\mathcal{P}_n$ is defined to be the number of elements in its $n$-fold Kronecker product that are not equal to $I$. The *distance* of $\mathcal{C}$ is defined as the minimum weight of all Pauli operators in $N(\mathcal{S})\backslash\mathcal{S}$, where $N(\mathcal{S})$ denotes the normalizer group of $\mathcal{S}$ in $\mathcal{P}_n$. If code $\mathcal{C}$ has distance $d$, we call $\mathcal{C}$ an $[[n,k,d]]$ quantum stabilizer code. In particular, $\mathcal{C}$ is called *degenerate* if its distance $d$ is larger than the minimum weight of its stabilizers.

In the symplectic representation, the stabilizer group $\mathcal{S}$ is constructed from the rows of the stabilizer matrix $H = [H_x, H_z]$, where $H_x, H_z \in \mathbb{F}_2^{m\times n}$ are binary matrices with $m$ rows and $n$ columns. In particular, each row $(h_x, h_z)$ of $H$ defines the stabilizer $D(h_x,h_z)$, and the set of stabilizers defined by all rows generates $\mathcal{S}$. In this way, the constraint requiring $\mathcal{S}$ to be commutative can be expressed as $H\Lambda H^T = H_x H_z^T + H_z H_x^T = 0$. An important class of stabilizer codes, known as Calderbank–Shor–Steane (CSS) codes [36], [37], where each stabilizer has the form $D(a,0)$ or $D(0,b)$ have

$$H_x = \begin{bmatrix} 0 \\ G_2 \end{bmatrix}, \quad H_z = \begin{bmatrix} H_1 \\ 0 \end{bmatrix}, \quad (5)$$

where $H_1 \in \mathbb{F}_2^{(n-k_1)\times n}$ is the parity-check matrix of a classical $[n,k_1]$ code $\mathcal{C}_1$ and $G_2 \in \mathbb{F}_2^{k_2\times n}$ is the generator matrix of a classical $[n,k_2]$ code $\mathcal{C}_2$ with $\mathcal{C}_2 \subseteq \mathcal{C}_1$. In this work, we will focus our discussion on the CSS codes.

### B. Syndrome Decoding of Stabilizer Codes

For an $[[n,k]]$ stabilizer code $\mathcal{C}$, we consider the error model where the encoded state $|\psi\rangle$ is corrupted by an $n$-qubit Pauli error $E = D(x,z) \in \mathcal{P}_n$ as $|\psi\rangle \to E|\psi\rangle$. The goal of the decoder is to detect and correct this error by conducting measurements on all the stabilizers in $H$. The stabilizer measurement result can be expressed as a length-$m$ binary syndrome vector $s = (x,z)\Lambda H^T$. Here, we assume that all syndrome measurements are perfect.

After obtaining the syndrome $s$, the decoder aims to find an estimated $\widehat{E}$ yielding this syndrome, so that the inverse operator $\widehat{E}^\dagger$ can be applied to map $E|\psi\rangle \to \widehat{E}^\dagger E|\psi\rangle$ for error correction. This decoding process has the following four possible outcomes:

1) **Failure**: The decoder fails to provide an estimated $\widehat{E}$ that yields the syndrome $s$.
2) **Successful (Exact Match)**: The estimated error is equal to the channel error, i.e., $\widehat{E} = E$.
3) **Successful (Degenerate Error)**: The difference between $\widehat{E}$ and $E$ is a stabilizer, i.e., $\widehat{E}E \in \mathcal{S}$.
4) **Failure (Logical Error)**: The difference between $\widehat{E}$ and $E$ is a logical operator, i.e., $\widehat{E}E \in N(\mathcal{S})\backslash\mathcal{S}$.

Using the names from [38], two decoding strategies for stabilizer codes are *Quantum Maximum Likelihood Decoding* (QMLD), where the decoder aims at finding the most probable error $\widehat{E}$ given the syndrome, and *Degenerate Quantum Maximum Likelihood Decoding* (DQMLD), the optimal strategy, where the decoder aims at finding the most probable coset of the stabilizer group $\mathcal{S}$ given the syndrome.

In Sections III and IV that follow, we will focus on the independent Pauli-$\sigma_x$ error channel with probability $p_x$, where the Pauli error has the form $E = D(X,0)$, with $X \in \mathbb{F}_2^n$ being a random vector with

$$\Pr\big(X = (x_1, x_2, \ldots, x_n)\big) = \prod_{i=1}^{n} p_x^{x_i}(1-p_x)^{1-x_i}. \quad (6)$$

For CSS codes in this case, given $X$, the decoder only needs to consider part of the syndrome $S_x = XH_1^T$, where $H_1$ is the submatrix of $H_z$ as in (5). The QMLD decoding problem then becomes computing $\widehat{X}$ where

$$\widehat{X} = \underset{x\in\mathbb{F}_2^n}{\arg\max}\ \Pr(X = x \,|\, S_x = s_x) = \underset{x\in\mathbb{F}_2^n\,:\,xH_1^T = s_x}{\arg\min}\ \mathrm{wt}(x). \quad (7)$$

This is equivalent to the maximum-likelihood decoding problem for the binary symmetric channel (BSC) in classical coding theory, which is known to be NP-complete [39]. Thus, we turn to the low-complexity belief propagation (BP) algorithm.

## III. Belief Propagation Decoding

### A. Belief Propagation Syndrome Decoding

BP is an iterative message-passing algorithm first introduced for the LDPC codes by Gallager [40], and first applied to the quantum decoding problem by Poulin and Chung in [23]. For the syndrome decoding problem of stabilizer codes over independent Pauli-$\sigma_x$ errors, BP runs on a Tanner graph $G = (V,C,E)$ representing $H_1$ in (5). In this Tanner graph, the variable nodes in $V = \{v_1,\ldots,v_n\}$ represent the elements of $X = (X_1,\ldots,X_n)$ and the check nodes in $C = \{c_1,\ldots,c_m\}$ represent the $\sigma_z$-stabilizers in $H_1$. A variable node $v_i$ is connected to a check node $c_j$ if $H_1(i,j) = 1$. In addition, each variable node $v_i$ is connected to a degree 1 check node that represents the source of the channel log-likelihood ratio (LLR) $\mu_{v_i}$, and each check node $c_j$ is connected to a degree 1 variable node that represents the source of a syndrome bit $s_{x,j}$ in $s_x = (s_{x,1},\ldots,s_{x,m})$. An example Tanner graph representing $H_1$ for the $[[7,1,3]]$ Steane code [37] is shown in Figure 1.

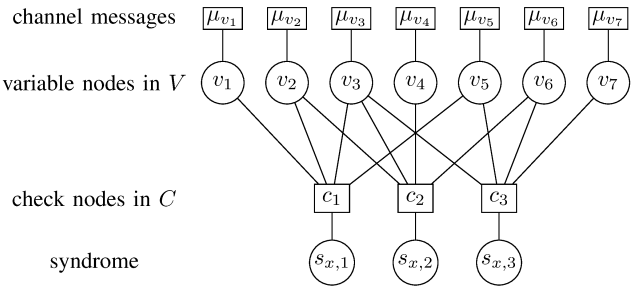


**Figure 1.** Tanner graph of $H_1$ for the $[[7, 1, 3]]$ Steane code

Set $\mu_{v_i} = \log((1-p_x)/p_x)$ as the initial estimate of the overall LLR of $X_i$, and set the message from each $v_i$ to all its connected $c_j$ as $m^{(0)}_{v_i \to c_j} = \mu_{v_i}$ at iteration $t = 0$. Then for iteration $t = 0, 1, \ldots, T$, BP updates the messages between variable nodes and check nodes as

$$m^{(t)}_{c_j \to v_i} = (-1)^{s_{x,j}} \tanh^{-1}\left( \prod_{v_k \in \partial c_j \setminus v_i} \tanh m^{(t)}_{v_k \to c_j} \right), \quad (8)$$

and

$$m^{(t+1)}_{v_i \to c_j} = \mu_{v_i} + \sum_{c_k \in \partial v_i \setminus c_j} m^{(t)}_{c_k \to v_i}, \quad (9)$$

This update rule is called the sum-product algorithm [18]. The *bias* for variable node $v_i$ after $t$ iterations can be computed as

$$m^{(t)}_{v_i} = \mu_{v_i} + \sum_{c_k \in \partial v_i} m^{(t)}_{c_k \to v_i}. \quad (10)$$

For sufficiently large $t$, the bias $m^{(t)}_{v_i}$ approximates the LLR of the marginal probabilities for $X_i$ as

$$m^{(t)}_{v_i} \approx \log \frac{\Pr(X_i = 0 \,|\, S_z = s_z)}{\Pr(X_i = 1 \,|\, S_z = s_z)}. \quad (11)$$

This approximation is exact if $G$ is a tree [18]. The sign of $m^{(t)}_{v_i}$ represents the hard value toward which $v_i$ is biased, and the absolute value of the bias denoted as $\gamma^{(t)}(v_i) = |m^{(t)}_{v_i}|$ represents the *reliability* of this variable node.

BP for syndrome decoding is commonly equipped with the early termination rule when the hard values of the variable nodes $\widehat{x}^{(t)} = (\widehat{x}^{(t)}_1, \widehat{x}^{(t)}_2, \ldots, \widehat{x}^{(t)}_n)$ match the syndrome by satisfying $\widehat{x}^{(t)} H_1^T = s_z$. This syndrome match event is called *convergence*. If none of the estimated $\widehat{x}^{(t)}$ vectors match the syndrome when $t$ reaches a preset maximum iteration number $T$, then the BP decoder reports failure due to *non-convergence*.

### B. Prior Works that Mitigate the Non-Convergence Issue

BP for the syndrome decoding problem of QLDPC codes is known to have issues with convergence [23], [25]. Consider the $[[882, 24, 18 \leqslant d \leqslant 24]]$ generalized bicycle B1 code proposed in [16]. We simulated its BP decoding performance with the sum-product algorithm over independent Pauli-$\sigma_x$ errors with probability $p_x$ and the performance can be found in the blue curve in Figure 2. In our simulation, all the block error cases that we observe are due to non-convergence.

To mitigate the non-convergence issue, Pantaleev and Kalachev [16] proposed the use of ordered statistics decoding as a post-processor when BP fails to converge. In Figure 2, the red curve shows the BP-OSD performance with order 0. In this simulation, BP-OSD employs the normalized min-sum algorithm for message passing with the normalization factor $\alpha = 0.625$, matching the decoder in [16]. The BP-OSD decoder shows a significant performance improvement over BP and this gain has been observed across various families of quantum stabilizer codes [16], [41], with the cost of a higher computational complexity of $O(n^3)$ [16].

Another approach to mitigate the non-convergence issue of BP, called BP with stabilizer inactivation, is proposed by Crest, Mhalla and Savin in [33]. In Figure 2, the performance of BP-SI with $\lambda = 10$ on the B1 code is shown in the yellow curve, where $\lambda$ denotes the maximum number of inactivated stabilizers attempted. The data points for the yellow curve are taken and translated directly from [33, Figure 2], where the decoder uses the serial message-passing scheduling for the normalized min-sum algorithm with normalization factor $\alpha = 0.9$. In [33], the worst-case complexity and the average complexity of BP-SI are claimed to be $O(\lambda_{\max} n \log n)$ and $O(\lambda_{\text{avg}} n \log n)$, respectively, assuming BP has complexity $O(n \log n)$. Here, $\lambda_{\max}$ represents the maximum number of inactivated stabilizers and $\lambda_{\text{avg}}$ represents the average number of inactivated stabilizers. Notably, at the end of post-processing for both BP-OSD and BP-SI, one needs to solve a system of linear equations.

## IV. Belief Propagation with Guided Decimation

In this work, we improve BP on QLDPC codes by combining it with guided decimation. Message-passing algorithms with "decimation" were first introduced for the $K$-SAT constraint satisfaction problem based on insights from statistical physics [34]. In such problems, there are typically many valid solutions and the goal is to find just one of them. In [34], decimation was first combined with a related message-passing algorithm called survey propagation. Later, the idea of decimation was extended to define the BP guided decimation (BPGD) algorithm [35] and related approaches were applied to the lossy compression problem [42], [43].

### A. Intuition Behind BPGD for Quantum Decoding

In [35], the BPGD algorithm is understood to approximate the process of sampling a vector from the distribution implied by the Tanner graph. From Figure 1, we see that the Tanner graph for BP syndrome decoding constrains the error vector to match the observed syndrome and assumes each error bit is drawn independently with probability $p_x$. Applying this picture to the quantum decoding problem, we can think of BPGD decoding as sampling from the error patterns that match the syndrome with weights proportional to the distribution in (6). While this type of decoding is not equivalent to either QMLD or DQMLD, we will argue that the sampling approach is closer in spirit to DQMLD than QMLD. The reason is that the sampling probability for a correction operator is proportional to the sum of the probabilities of all the error patterns associated with that correction operator. Thus,

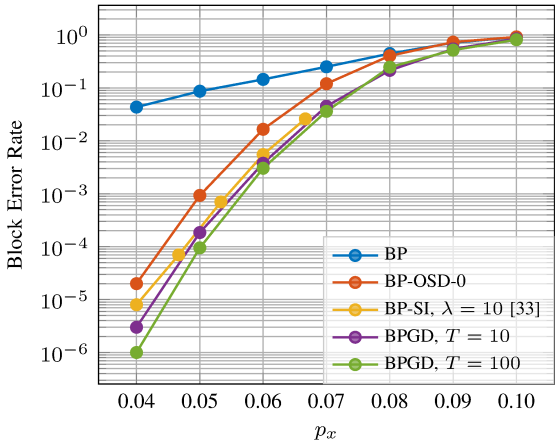


**Figure 2.** Performance of B1 code [16] over Pauli-$\sigma_x$ errors.

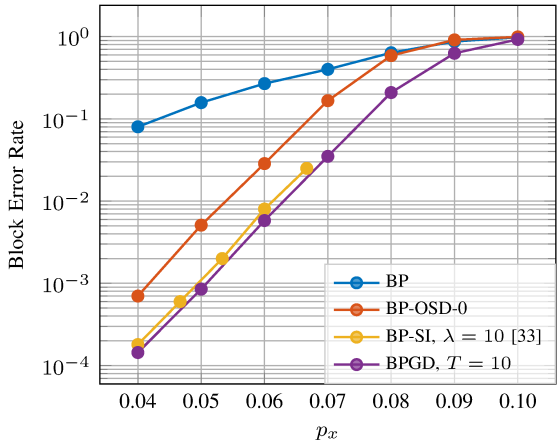


**Figure 3.** Performance of C2 code [16] over Pauli-$\sigma_x$ errors.

the error is more likely to be corrected. This connection is discussed further in [44].

### *B. The BPGD Algorithm*

Now, the BPGD algorithm is described in detail. First, the channel LLRs for all the variable nodes in the Tanner graph are initialized as $\mu_{v_i} = \log((1-p_x)/p_x)$ like BP. Then, decoding proceeds in rounds. In the $r$-th round, the BP algorithm is run for $T$ iterations following (8) and (9). If it converges to an error pattern matching the syndrome, then the decoder terminates and returns the hard values of the variable nodes as the estimated error. Otherwise, out of all the variable nodes that are not yet decimated, the variable node $v_i$ with the largest reliability $\gamma(v_i)$ is decimated by updating its channel message $\mu_{v_i}$ based on its bias $m_{v_i}^{(rT)}$ to

$$\mu_{v_i} = \begin{cases} \text{llr}_{\max}, & \text{if } m_{v_i}^{(rT)} > 0 \\ -\text{llr}_{\max}, & \text{if } m_{v_i}^{(rT)} \leqslant 0, \end{cases} \tag{12}$$

where $\text{llr}_{\max}$ is a fixed large value (e.g., $\text{llr}_{\max} = 25$ throughout this paper). If we choose $\text{llr}_{\max} = \infty$, then decimation effectively assigns a hard value for the variable node $v_i$ based on its bias. While such a choice makes sense in theory, it can also introduce numerical problems in practice.

After decimating the most reliable variable node, the process starts round $r+1$ and continues either until convergence (i.e., the estimated error pattern matches the syndrome) or all the variable nodes have been decimated. After all variable nodes are decimated, if the hard values of the variable nodes do not match the syndrome, then this is called a *non-convergence failure* of the BPGD algorithm. The pseudo-code for the BPGD algorithm is provided in the extended version of this paper [44, Algorithm 1].

### *C. Numerical Results*

In Figure 2, we show the simulation result of the BPGD decoder for the $[[882, 24, 18 \leqslant d \leqslant 24]]$ B1 code [16] over independent Pauli-$\sigma_x$ errors with probability $p_x$. BPGD with $T = 100$, shown by the green curve, yields the best performance. Using $T = 100$ ensures that in each round, BP is run for a sufficient number of iterations to provide accurate approximate marginal probabilities for the variable nodes. Reducing the number of BP iterations per round from $T = 100$ to $T = 10$, shown by the purple curve, does not significantly degrade the BPGD performance. It is worth noting that, in our simulations, the majority of the errors (over 90%) from the BPGD runs for both $T = 100$ and $T = 10$ are attributed to non-convergence. Therefore, similar to the BP algorithm, we rarely encounter logical errors upon convergence from running BPGD. For comparison, Figure 2 also includes the decoding performance of BP, BP-OSD with order 0, and BP-SI with $\lambda = 10$, whose settings are explained in Section III-B.

In Figure 3, we also show the performance of BPGD with $T = 10$ for the $[[1922, 50, 16]]$ hypergraph product C2 code [16] over independent Pauli-$\sigma_x$ errors with probability $p_x$. For comparison, we include the performances for BP, BP-OSD with order 0, and BP-SI with $\lambda = 10$, whose respective settings remain consistent with those in Figure 2: 1) the BP decoder runs the basic sum-product algorithm; 2) the BP-OSD decoder with order 0 runs the serial normalized min-sum algorithm with normalization factor $\alpha = 0.625$; 3) the BP-SI decoder with $\lambda = 10$ runs the serial normalized min-sum algorithm with normalization factor $\alpha = 0.9$, whose data points are taken and translated directly from [33, Figure 2]. We can see that for the two QLDPC codes we considered in Figure 2 and Figure 3, BPGD shows better performance compared with order-0 BP-OSD and BP-SI with $\lambda = 10$.

### *D. Complexity Analysis for BPGD*

BPGD requires at most $n$ decimation rounds, each involving $T$ iterations of message-passing updates of complexity $O(n)$ and a search for the most reliable qubit for decimation with complexity $O(n)$. Therefore, the worst-case complexity of BPGD is $O(Tn^2)$. This worst-case complexity scales roughly the same as BP with SI [33], and it is better compared with BP-OSD with order 0, which has complexity $O(n^3)$ [16]. Moreover, unlike BP-OSD and BP-SI, BPGD does not require solving systems of linear equations, which makes it potentially more friendly for hardware implementation.

| $p_x$ | 0.05 | 0.06 | 0.07 | 0.08 |
|---|---|---|---|---|
| simulation runs | 1000000 | 100000 | 100000 | 10000 |
| $r_{\text{avg}}$ | 2.91 | 9.82 | 60.46 | 231.7 |

**Table 1.** Average number of decimated variables nodes of BPGD with $T = 10$ when decoding the B1 code [16] over Pauli-$\sigma_x$ errors.

For the BPGD algorithm, if we assume BP in each round has complexity $O(Tn)$, then its average-case complexity is $O(r_{\text{avg}}Tn)$, where $r_{\text{avg}}$ denotes the average number of decimated variable nodes. The value of $r_{\text{avg}}$ is highly dependent on the Pauli-$\sigma_x$ error rate $p_x$, meaning BPGD has different average complexity in different error rate regimes. In Table 1, we show the average number of decimated variable nodes for BPGD with $T = 10$ when decoding the B1 code. From Table 1 we can see that, in the low error rate regime such as $p_x = 0.05$, $r_{\text{avg}}$ becomes very small, making the average complexity of the BPGD approach $O(Tn)$, the complexity of the BP algorithm. We note that a similar observation has also been made for BP-SI concerning the average number of inactivated stabilizers in the low error rate regime [33].

## V. Quaternary BP with Guided Decimation

In this section, for the syndrome decoding problem of stabilizer codes over the depolarizing channel, we present a natural extension of the binary BPGD algorithm that gives the quaternary version. In the depolarizing channel with physical error probability $p$, each encoded qubit is independently affected by a Pauli $\sigma_x$, $\sigma_y$, or $\sigma_z$ error, each occurring with a probability of $p/3$. Here, we represent the Pauli error as a random quaternary vector $Q = (Q_1, \ldots, Q_n)$ with its realization $q = (q_1, \ldots, q_n) \in \{0, 1, 2, 3\}^n$, with $0, 1, 2, 3$ representing the absence of error, or the presence of a Pauli $\sigma_x$, $\sigma_y$ or $\sigma_z$ error, respectively.

For the depolarizing channel, we consider the quaternary BP (Q-BP) algorithm that jointly decodes the $\sigma_x$ and $\sigma_z$ errors from depolarizing noise. For detailed descriptions of Q-BP, we refer the readers to [45] and to [24]. An efficient log domain implementation of Q-BP has also been discussed in [46].

For the syndrome decoding problem on a CSS code, Q-BP runs on a Tanner graph $G = (V, C, E)$ similar to the binary BP, except that here $C$ contains check nodes representing both the $\sigma_x$-stabilizers in $G_2$ and the $\sigma_z$-stabilizers in $H_1$, where $H_1$ and $G_2$ are submatrices of $H_x$ and $H_z$, respectively, as shown in (5). In addition, channel LLR becomes the channel message initialized as $\mu_{v_i} = (1-p, p/3, p/3, p/3)$, and each variable node $v_i \in V$ contains four normalized probabilities $(p_{v_i,0}, p_{v_i,1}, p_{v_i,2}, p_{v_i,3})$ which, after a sufficient number of message-passing iterations, approximate the marginal probabilities for $Q_i$ conditioned on the syndrome $s$ as

$$p_{v_i,j} \approx \Pr(Q_i = j \mid S = s), \qquad \text{for } j \in \{0, 1, 2, 3\}. \tag{13}$$

Define the reliability for the variable node $v_i$ as $\gamma(v_i) = \max\{p_{v_i,0}, p_{v_i,1}, p_{v_i,2}, p_{v_i,3}\}$. Similar to its binary counterpart, the Q-BP algorithm can also be improved by guided decimation, denoted as the quaternary belief propagation guided decimation (Q-BPGD) algorithm. After $T$ iterations of Q-BP in each round, out of all the variable nodes we have not yet decimated, we pick the variable node $v_i$ with the largest reliability $\gamma(v_i)$ and decimate it by changing its channel message as

$$\mu_{v_i} = \begin{cases} (1-\epsilon, \epsilon, \epsilon, \epsilon), & \text{if } \gamma_i^* = 0 \\ (\epsilon, 1-\epsilon, \epsilon, \epsilon), & \text{if } \gamma_i^* = 1 \\ (\epsilon, \epsilon, 1-\epsilon, \epsilon), & \text{if } \gamma_i^* = 2 \\ (\epsilon, \epsilon, \epsilon, 1-\epsilon), & \text{if } \gamma_i^* = 3 \end{cases}, \quad \gamma_i^* = \underset{j \in \{0,1,2,3\}}{\arg\max}\ p_{v_i,j}. \tag{14}$$

As $\epsilon$ approaches zero, this decimation rule essentially fixes $Q_i$ to one of the values in $\{0, 1, 2, 3\}$ according to the bias of its corresponding variable node $v_i$. In our implementation, we set $\epsilon = 1 \times 10^{-10}$ for numerical stability. We repeat this process until either convergence is reached at some round $r$, or until all the variable nodes have been decimated. The pseudocode for the Q-BPGD algorithm is provided in the extended version of this paper [44, Algorithm 3].

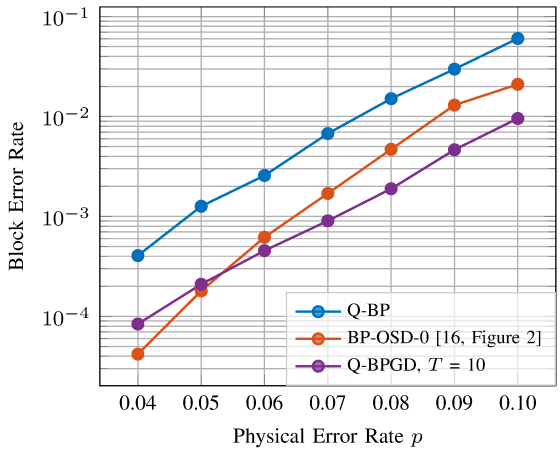


**Figure 4.** Performance of B2 code [16] over depolarizing noise.

In Figure 4, we present the Q-BPGD performance with $T = 10$ for the $[[882, 48, 16]]$ generalized hypergraph product B2 code proposed in [16]. The comparison includes both the Q-BP decoder with the sum-product algorithm and the order-0 BP-OSD decoder from [16, Figure 2]. For the B2 code, Q-BPGD outperforms the BP-OSD decoder in the high-error regime but is surpassed by BP-OSD in the low-error-rate regime.

## VI. Conclusion

In this paper, we introduce and evaluate the use of BPGD for the decoding of QLDPC codes in order to encourage convergence. BPGD shows strong performance compared with BP-OSD and BP-SI under independent Pauli-$\sigma_x$ errors. To better understand how BPGD boosts convergence, we provide an alternative view of the BP syndrome decoding setup for stabilizer codes as a sampling problem. Furthermore, we extend our guided decimation from binary BP to quaternary BP, demonstrating performance competitive compared to BP-OSD in the high-error regime under depolarizing noise.

## Acknowledgements

This material is based on work supported by the NSF under Grants 2106213 and 2120757.

## REFERENCES


[1] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. Sloane, "Quantum error correction and orthogonal geometry," *Physical Review Letters*, vol. 78, no. 3, p. 405, 1997.

[2] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. Sloane, "Quantum error correction via codes over GF(4)," *IEEE Transactions on Information Theory*, vol. 44, no. 4, pp. 1369–1387, 1998.

[3] D. Gottesman, *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.

[4] A. Y. Kitaev, "Quantum computations: algorithms and error correction," *Russian Mathematical Surveys*, vol. 52, no. 6, p. 1191, 1997.

[5] ——, "Fault-tolerant quantum computation by anyons," *Annals of physics*, vol. 303, no. 1, pp. 2–30, 2003.

[6] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4452–4505, 2002.

[7] M. H. Freedman, D. A. Meyer, and F. Luo, "$Z_2$-systolic freedom and quantum codes," *Mathematics of quantum computation, Chapman & Hall/CRC*, pp. 287–320, 2002.

[8] S. Bravyi, D. Poulin, and B. Terhal, "Tradeoffs for reliable quantum information storage in 2D systems," *Physical review letters*, vol. 104, no. 5, p. 050503, 2010.

[9] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A*, vol. 86, no. 3, p. 032324, 2012.

[10] D. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, "Surface code quantum computing by lattice surgery," *New Journal of Physics*, vol. 14, no. 12, p. 123011, 2012.

[11] D. J. MacKay, G. Mitchison, and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2315–2330, 2004.

[12] J.-P. Tillich and G. Zémor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1193–1202, 2013.

[13] A. A. Kovalev and L. P. Pryadko, "Improved quantum hypergraph-product LDPC codes," in *2012 IEEE International Symposium on Information Theory Proceedings*. IEEE, 2012, pp. 348–352.

[14] ——, "Quantum kronecker sum-product low-density parity-check codes with finite rate," *Physical Review A*, vol. 88, no. 1, p. 012311, 2013.

[15] J. Haah, "Local stabilizer codes in three dimensions without string logical operators," *Physical Review A*, vol. 83, no. 4, p. 042330, 2011.

[16] P. Panteleev and G. Kalachev, "Degenerate quantum LDPC codes with good finite length performance," *Quantum*, vol. 5, p. 585, 2021.

[17] S. Yang and R. Calderbank, "Quantum spatially-coupled codes," *arXiv preprint arXiv:2305.00137*, 2023.

[18] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[19] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[20] ——, "Improved low-density parity-check codes using irregular graphs," *IEEE Transactions on information Theory*, vol. 47, no. 2, pp. 585–598, 2001.

[21] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 599–618, 2001.

[22] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE transactions on information theory*, vol. 47, no. 2, pp. 619–637, 2001.

[23] D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *arXiv preprint arXiv:0801.1241*, 2008.

[24] Z. Babar, P. Botsinis, D. Alanis, S. X. Ng, and L. Hanzo, "Fifteen years of quantum LDPC coding and improved decoding strategies," *iEEE Access*, vol. 3, pp. 2492–2519, 2015.

[25] N. Raveendran and B. Vasić, "Trapping sets of quantum LDPC codes," *Quantum*, vol. 5, p. 562, 2021.

[26] Y.-J. Wang, B. C. Sanders, B.-M. Bai, and X.-M. Wang, "Enhanced feedback iterative decoding of sparse quantum codes," *IEEE transactions on information theory*, vol. 58, no. 2, pp. 1231–1241, 2012.

[27] A. Rigby, J. Olivier, and P. Jarvis, "Modified belief propagation decoders for quantum low-density parity-check codes," *Physical Review A*, vol. 100, no. 1, p. 012330, 2019.

[28] Y.-H. Liu and D. Poulin, "Neural belief-propagation decoders for quantum error-correcting codes," *Physical review letters*, vol. 122, no. 20, p. 200501, 2019.

[29] X. Xiao, N. Raveendran, and B. Vasić, "Neural-net decoding of quantum LDPC codes with straight-through estimators," in *in Proc. Information Theory and Applications Workshop (ITA)*, 2019.

[30] S. Miao, A. Schnerring, H. Li, and L. Schmalen, "Neural belief propagation decoding of quantum LDPC codes using overcomplete check matrices," in *2023 IEEE Information Theory Workshop (ITW)*. IEEE, 2023, pp. 215–220.

[31] J. Old and M. Rispler, "Generalized belief propagation algorithms for decoding of surface codes," *Quantum*, vol. 7, p. 1037, 2023.

[32] K.-Y. Kuo and C.-Y. Lai, "Exploiting degeneracy in belief propagation decoding of quantum codes," *npj Quantum Information*, vol. 8, no. 1, p. 111, 2022.

[33] J. Du Crest, M. Mhalla, and V. Savin, "Stabilizer inactivation for message-passing decoding of quantum LDPC codes," in *2022 IEEE Information Theory Workshop (ITW)*. IEEE, 2022, pp. 488–493.

[34] M. Mézard, G. Parisi, and R. Zecchina, "Analytic and algorithmic solution of random satisfiability problems," *Science*, vol. 297, no. 5582, pp. 812–815, 2002.

[35] A. Montanari, F. Ricci-Tersenghi, and G. Semerjian, "Solving constraint satisfaction problems through belief propagation-guided decimation," *arXiv preprint arXiv:0709.1667*, 2007.

[36] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Physical Review A*, vol. 54, no. 2, p. 1098, 1996.

[37] A. Steane, "Multiple-particle interference and quantum error correction," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 452, no. 1954, pp. 2551–2577, 1996.

[38] P. Iyer and D. Poulin, "Hardness of decoding quantum stabilizer codes," *IEEE Transactions on Information Theory*, vol. 61, no. 9, pp. 5209–5223, 2015.

[39] E. Berlekamp, R. McEliece, and H. Van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, 1978.

[40] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.

[41] J. Roffe, D. R. White, S. Burton, and E. Campbell, "Decoding across the quantum low-density parity-check code landscape," *Physical Review Research*, vol. 2, no. 4, p. 043423, 2020.

[42] T. Filler and J. Fridrich, "Binary quantization using belief propagation with decimation over factor graphs of LDGM codes," *arXiv preprint arXiv:0710.0192*, 2007.

[43] V. Aref, N. Macris, and M. Vuffray, "Approaching the rate-distortion limit with spatial coupling, belief propagation, and decimation," *IEEE Transactions on Information Theory*, vol. 61, no. 7, pp. 3954–3979, 2015.

[44] H. Yao, W. A. Laban, C. Häger, H. D. Pfister *et al.*, "Belief propagation decoding of quantum LDPC codes with guided decimation," *arXiv preprint arXiv:2312.10950*, 2023.

[45] K.-Y. Kuo and C.-Y. Lai, "Refined belief propagation decoding of sparse-graph quantum codes," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 2, pp. 487–498, 2020.

[46] C.-Y. Lai and K.-Y. Kuo, "Log-domain decoding of quantum LDPC codes over binary finite fields," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–15, 2021.