Linear programming decoder for hypergraph product quantum codes

Omar Fawzi, Lucien Grouès, Anthony Leverrier

Linear programming decoder for hypergraph product quantum codes

Omar Fawzi
Univ Lyon, ENS Lyon, UCBL, CNRS, Inria
LIP, F-69342, Lyon Cedex 07, France

Lucien Grouès
Inria, Paris

Anthony Leverrier
Inria, Paris

***Abstract*—We introduce a decoder for quantum CSS codes that is based on linear programming. Our definition is a priori slightly different from the one proposed by Li and Vontobel as we have a syndrome oriented approach instead of an error oriented one, but we show that the success condition is equivalent. Although we prove that this decoder fails for quantum codes that do not have good soundness property (i.e., having large errors with syndrome of small weight) such as the toric code, we obtain good results from simulations. We run our decoder for hypergraph products of two random LDPC codes, showing that it performs better than belief propagation, even combined with the small-set-flip decoder that can provably correct a constant fraction of random errors.**

I. Introduction

Going beyond NISQ (noisy intermediate-scale quantum) technologies and reaching the full power of quantum computing will require to develop quantum error correction to fight the effects of decoherence and allow for large-scale computation despite the presence of noise. While the first generation of quantum codes will likely rely on surface codes, it is important to recall that such codes are inherently costly in terms of overhead, and that much better performance will only be possible with more efficient quantum codes, such as quantum low-density parity-check (LDPC) codes. Among those, hypergraph product codes (HPCs) have recently attracted some attention by offering essentially the same level of protection as the surface code (i.e., minimum distance growing like the square-root of the number of qubits) while at the same time requiring only very low overhead, offering a constant encoding rate (ratio of protected qubits and physical qubits) instead of an asymptotically null rate for surface codes [14].

Good code parameters are not sufficient for quantum fault-tolerance: it is also crucial that errors can be corrected efficiently between steps of the computation. While the sparsity of the parity-check matrix yields efficient decoding algorithms based on message-passing in the classical case, decoding quantum LDPC codes appears to be much subtler. This is due to the degeneracy of quantum LDPC codes: there exist many errors of constant weight that do not affect the logical information, but induce short cycles in the factor graph of the code. These, in turn, make direct generalizations of classical decoding algorithms typically behave very poorly on a quantum LDPC code. For these reasons, it is necessary to carefully adapt classical decoding algorithms to the quantum regime: examples for the HPC include small-set-flip [8] as a generalization of the classical bit-flip algorithm [12], as well a quantum version of belief propagation [7] or OSD [10].

In this paper, we consider linear programming (LP) decoding for HPC. An LP decoder for quantum stabilizer codes was already defined by Li and Vontobel [9]. Our definition is slightly different and does not require finding an error having the right syndrome, but we show that the two versions of quantum LP decoder are actually equivalent. We then show that such LP decoders cannot successfully decode errors that are significantly larger than their syndrome. Such errors exist for Calderbank-Shor-Stean (CSS) codes such as the toric code, but more generally for codes that do not have the soundness property as defined in [4], [8]. Nevertheless, simulation results suggest that for random HPC codes (which satisfy the soundness property with high probability [8]) the LP decoder can perform better than the small-set-flip decoding algorithm which can provably correct a constant fraction of errors [5], even when pre-processing the error using belief propagation [7].

The paper is structured as follows. In section II we introduce the formalism that will be used in the remaining of the paper. In section III we introduce the new formulation of the LP decoder in the quantum case (QLPD) which is syndrome based, show that it is equivalent to the formulation of Li and Vontobel (LV-QLPD) which is more error based approach and establish various properties as well as obstructions for the QLPD. In section IV we give the result of our simulations on random HPC. The missing proofs will be included in the full version of the paper.

II. Formalism

A. CSS codes and decoding problem

A quantum CSS code is a particular instance of quantum code characterized by two parity-check matrices H_X and H_Z with the property that $H_X H_Z^T = 0$. An error pattern is defined as a pair (e_X, e_Z) with e_X and e_Z, two binary vectors corresponding respectively to X-type and Z-type errors. The quantum decoding problem is as follows: given the pair of syndromes $(s_X, s_Z) = (H_X e_X^T, H_Z e_Z^T)$, the decoder succeeds if it returns $(e_X + f_X, e_Z + f_Z)$ with f_X in the row span of H_Z and f_Z in the row span of H_X. In other words, the decoder need not recover the original error exactly, but only up to elements from the row spans of H_Z and H_X. The quantum code is called LDPC when the parity-check matrices H_X and

H_Z are both sparse. In this case, there are many low-weight errors of the form (f_X, f_Z) that do not affect the encoded information: this is the degeneracy phenomenon.

A convenient (but suboptimal in general) way to solve the decoding problem is to try to recover e_X and e_Z independently, and this is the approach we will take here. The symmetry of the problem allows us to focus on X-type errors for instance, and treat Z-type errors similarly. In the following, we call *generators* the rows of H_Z and say that two errors e, e' are equivalent ($e \sim e'$) if they differ by a sum of generators. Given some error e, the decoder receives the syndrome $s = He^T$ (where we wrote H for H_X) and succeeds if it returns an error $e' \sim e$. We will denote by $\mathcal{C}$ the classical code associated with the parity check matrix H. It will also be convenient to consider the Tanner graph $\mathcal{T} = (V \cup C, E)$ of $\mathcal{C}$ where the nodes V correspond to the bits and the nodes C correspond to the checks and $(v_i, c_j) \in E$ iff $H_{j,i} = 1$.

B. Toric codes and hypergraph product codes

We focus on only two kinds of CSS quantum codes: the toric code and hypergraph product codes. The toric code [2] is a LDPC quantum code defined on a torus, i.e. a square lattice where the opposite borders are identified. The qubits sit on the edges of the lattice, and X-type (resp. Z-type) generators sit on the vertices (resp. on the faces) and act nontrivially on their 4 neighbouring qubits. HPCs are a generalisation of the toric code obtained by taking two arbitrary classical LDPC codes and then performing the hypergraph product operation [3], [14]. We refer readers to [14] for a precise definition of this product operation. The toric code is recovered as the product of two repetition codes.

III. LP DECODER FOR QUANTUM CODES

A. Quantum syndrome based LP decoder

The linear programming decoder for classical codes first proposed by [6] is based on a linear programming relaxation of the problem of maximum likelihood decoding. Given a noisy word y, the objective is to find x in the code that maximizes the likelihood. The linear program depends on y and either returns the correct x in the code, in which case the decoding was successful, or it returns a fractional solution of the linear program, in which case we say the decoding failed. For quantum codes, we do not have access to the noisy word, but only to the syndrome $s = Hy^T$ and the objective is to find the most likely error corresponding to this syndrome. As such, to be applicable to the quantum case, the LP decoder needs to be adapted.

Li and Vontobel suggested to adapt it by considering an arbitrary word with the desired syndrome [9], we will refer to their decoder as LV-QLPD. Here, we will follow a different approach relying on defining the *syndrome polytope*, we will refer to this decoder as QLPD. However we will then show that QLPD is in fact equivalent to LV-QLPD, while allowing us to avoid solving a linear system over $\mathbb{F}_2$.

Definition 1 (Syndrome polytope)**.** *Let H be an $m \times n$ parity-check matrix and $s \in \{0,1\}^m$. For each row $j \in [m]$, we define the polytope*

$$\mathcal{P}_j^s := \mathrm{Conv}(\{x \in \{0,1\}^n | (Hx)_j = s_j\}).$$

The syndrome polytope $\mathcal{P}^s(\mathcal{C})$ *of the code $\mathcal{C} = \ker H$ and syndrome s is given by:*

$$\mathcal{P}^s(\mathcal{C}) := \bigcap_{j \in [m]} \mathcal{P}_j^s.$$

Note that a syndrome polytope depends on the specific parity-check matrix (or Tanner graph) chosen to describe the code. When the syndrome is zero, $\mathcal{P}^0(\mathcal{C})$ is called the fundamental polytope [6].

We now define the QLPD. As mentioned, the decoder will correct for X-type and Z-type errors independently, and we focus only on X-type errors, for which the associated parity-check matrix is denoted by H.

Definition 2 (QLPD)**.** *Given a parity-check matrix H and a syndrome s, the decoder returns the correction $\hat{e}$:*

$$\hat{e} = \arg\min_{x \in \mathcal{P}^s} (|x|), \tag{1}$$

where $|x| := \sum_{i=1}^n x_i$.

Note that $\hat{e}$ could be a vector in $[0,1]^n$ that is not necessarily integral. In this case, we say that the decoder fails. We show that this problem is indeed a linear program, which can therefore be solved efficiently.

Given some check $c \in C$, we denote by $\mathcal{N}(c) \subset V$ its neighbourhood in the Tanner graph $\mathcal{T} = (V \cup C, E)$.

Definition 3 (Even/odd cardinality sub-neighbourhood)**.** *Let $\mathcal{T}$ be a Tanner graph. For each check node $c_j \in C$, we define E_i^0 (resp. E_i^1), the set of even (resp. odd) cardinality sub-neighbourhoods of c_j, as:*

$$E_j^0 := \{S \subseteq \mathcal{N}(c_i) \mid |S| \equiv 0 \mod 2\},$$

$$E_j^1 := \{S \subseteq \mathcal{N}(c_i) \mid |S| \equiv 1 \mod 2\}.$$

We arrive at the explicit description of the linear program based on a description of the syndrome polytope using linear inequalities:

Lemma 1 (Explicit definition of the QLPD)**.** *Given the syndrome s the syndrome based quantum LP decoder returns the output of the following LP:*

$$\begin{aligned} \textit{Minimise} \quad & \sum_{i=1}^n x_i \\ \textit{s.t.} \quad & 0 \le x_i \le 1, \quad \forall i \in [n]; \\ & \sum_{v_i \in S} x_i + \sum_{v_i \in \mathcal{N}(c_j) \setminus S} (1 - x_i) \le |\mathcal{N}(c_j)| - 1, \\ & \qquad \forall j \in [m],\ \forall S \in E_j^{1-s_j}. \end{aligned}$$

Proof. It suffices to show that for any $j \in [m]$ $x \in \mathcal{P}_j^s$ iff

$$\sum_{v_i \in S} x_i + \sum_{v_i \in \mathcal{N}(c_j) \setminus S} (1 - x_i) \leq |\mathcal{N}(c_j)| - 1$$

holds for any subset $S \in E_j^{s_j}$, which can easily be done by inspection. □

We can now show that our decoder is equivalent to the one introduced by Li and Vontobel in terms of success probability.

Theorem 1. *The QLPD succeeds iff the LV-QLPD succeeds.*

Proof. To show this we use the fact that we can go from the syndrome polytope to the fundamental polytope by applying to it several reflections, thus making an explicit link between them that we can further exploit. □

The QLPD and the LV-QLPD being equivalent in terms of success probability, it is interesting to compare them computationally. While the QLPD skips the linear system resolution needed at the beginning of LV-QLPD, the polytope on which we optimise our function depends on the input syndrome and so changes with each decoding. Thus the QLPD should be faster than the LV-QLPD for one decoding, yet it might be slower when doing several decodings in a row depending on the LP solver. Indeed both the solving of the linear system and the solving of the LP can be at least partly pre-computed.

Ref. [13] showed that linear programs of the form above can be solved more efficiently by considering relaxations where the constraints indexed by j and S are added progressively. We have implemented this technique for the simulations presented in Section IV. Further optimisations such as [1] are available for this special type of LP, and run times similar to belief-propagation (BP) decoder, but we leave this exploration for future work.

We denote by $\mathcal{V}$ the vertices of the fundamental polytope $\mathcal{P} = \mathcal{P}^0$. By construction, the set $\mathcal{V}$ contains the codewords of $\mathcal{C}$. The points of $\mathcal{V}$ which do not correspond to codewords admit at least one nonintegral coordinate (see [6]), and are called *pseudocodewords*.

Our first two characterizations of the syndrome polytope (Def. 1 and the one used in Lemma 1) are not very convenient to check whether a given fractional word belongs to the polytope, and we therefore introduce a third characterization.

Lemma 2 (Valid configurations)**.** *Let $\mathcal{T} = (V \cup C, E)$ be a Tanner graph and $s \in \{0,1\}^m$ be a syndrome. Then $x \in \mathcal{P}^s$ iff there exists a configuration $\{w_{j,S}\}_{j \in [m], S \subset E_j^{s_j}}$ such that:*

$$0 \leq w_{j,S} \leq 1, \qquad \forall j \in [m], \forall S \subset E_j^{s_j} \tag{2}$$

$$\sum_{S \subset E_j^{s_j}} w_{j,S} = 1, \qquad \forall j \in [m], \tag{3}$$

$$\sum_{S \subset E_j^{s_j}} w_{j,S} \mathbb{1}_S(v_i) = x_i, \qquad \forall j \in [m], \forall v_i \in \mathcal{N}(c_j) \tag{4}$$

This definition provides a systematic way to check whether a given point in $[0,1]^n$ belongs to the syndrome polytope. To do so we just have to give for each check c_j its part of the valid configuration, which is an explanation of its neighbourhood as a convex sum of integral neighbourhoods which are coherent with s_j. For example if $s_j = 1$ a coherent neighbourhood is be such that there are an odd number of bits set to 1.

We give examples for the toric code in Figures 1 and 2.

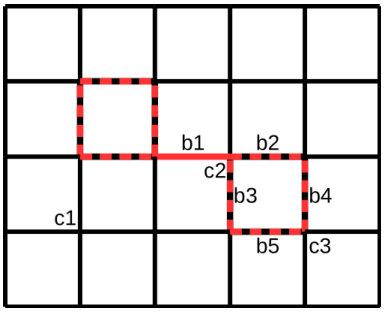

Fig. 1: The set V corresponds to the edges of the torus, and the checks C are the vertices. The vector x is represented in red, with the dotted edges taking value 0.5 and the full edges a value of 1. A valid configuration for x for syndrome zero can be constructed as follows. There are three kinds of checks for which we have to give a valid configuration. Those like c1 which see only bits equal to 0, we set $w_{c1,\varnothing} = 1$. Those like c2 which see one bit set to 0, one set to 1, and two set to 0.5, we set $w_{c2,\{v1,v2\}} = w_{c2,\{v1,v3\}} = 0.5$. Finally checks like c3 which see two bits set to 0 and 2 set to 0.5, we set $w_{c3,\{v4,v5\}} = w_{c3,\varnothing} = 0.5$.

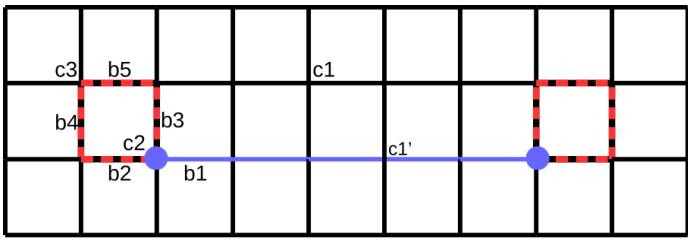

Fig. 2: Example of a valid configuration in the case where the syndrome is of weight 2. The unsatisfied checks are in blue, as well as the shortest error explaining such a syndrome (we zoomed in on a larger toric code so the borders of this lattice are not identified). However by setting the bits in red and black to 0.5 we get a lighter pseudo codeword. There is only one kind of checks we did not deal with in the previous example, it is when the check is unsatisfied. For this we set $w_{c2,\{v2\}} = w_{c2,\{v3\}} = 0.5$.

B. Fractional distance, success certificate

We first adapt the maximum likelihood certificate of the classical LP decoder which assesses that any integral correction output is the most likely integral correction possible.

Lemma 3 (Minimum weight correction certificate)**.** *If the QLPD outputs a correction $\hat{e}$ having integral coordinates, then*

$\hat{e}$ is the smallest correction with the right syndrome:

$$\hat{e} = \underset{x\in\{0,1\}^n, Hx^T=s}{\arg\min} (|x|).$$

Proof. Follows from the definition of the QLPD and the fact that all the words with syndrome s are vertices of $\mathcal{P}^s$. □

Remark 1. *Contrary to the LP decoder in the classical case, in the quantum case a minimum weight property does not always translate into a maximum likelihood property.*

Li and Vontobel showed in [9] that the classical success certificate of the LP decoder also applies in the quantum setting. This is however only interesting if the minimum distance of the code corresponding to H_X (or H_Z) is large enough (e.g., growing with n). This is not the case for quantum LDPC codes, where the minimum distances of H_X and H_Z are constant. For this reason, we now introduce a quantum version of the fractional distance which will also provide a sufficient condition to guarantee the success of the QLPD.

Definition 4 (Quantum minimum fractional distance)**.** *Given a code with fundamental polytope $\mathcal{P}$, the quantum minimum fractional distance $\mathrm{d}_{\mathrm{qfrac}}$ is defined as*

$$\mathrm{d}_{\mathrm{qfrac}} = \min_{x\in\mathcal{V}, x\nsim 0} (|x|).$$

Remark 2. *This is an adaptation of the classical minimum fractional distance. Since this new definition takes degeneracy into account, it is unclear whether it can be computed efficiently by solving linear programs similar to the ones given in [6].*

We now adapt to the quantum setting the classical success certificate proven in [6].

Theorem 2 (Success certificate)**.** *If the error is equivalent to an error of weight lower than $\lceil \frac{\mathrm{d}_{\mathrm{qfrac}}}{2} \rceil - 1$ then the QLPD will properly correct it.*

C. Limitations of the QLPD codes lacking the soundness property

While we expect the *quantum* minimum fractional distance to be larger than the classical fractional distance. Unfortunately it cannot be much larger for the toric code.

Theorem 3 (Bounded quantum minimum fractional distance)**.** *The quantum minimum fractional distance of the toric code is at most 5.*

Proof. The proof simply relies on the fact that the example shown in Figure 1 is not only a point of the fundamental polytope, but is in fact one of its vertices. Thus we can upper bound $\mathrm{d}_{\mathrm{qfrac}}$ by its weight which is 5. □

It is not easy to generalise this result to other codes, yet while a low value of d_{qfrac} does not imply that the decoder will fail, we show that large errors with syndrome of small weight will not be corrected by the QLPD, and this holds for any CSS code.

Theorem 4 (Failure for errors with syndrome of small weight)**.** *Given a CSS code with Z-type generators of weight at most w. Consider an X-type error with syndrome s and let e be an equivalent error with minimal weight. If $|e| > \frac{1}{2}w|s|$, then this error cannot be corrected by the QLPD.*

Proof. The idea is that any unsatisfied check can be explained in the syndrome polytope by one neighbouring generator with each of its bits set to 0.5. We could have a glimpse of this idea in figure 2 where even though the smallest word having the right syndrome was the blue string of weight 5, the fractional word drawn in red was also part of the syndrome polytope. Hence whenever the syndrome used for the syndrome polytope is of weight $|s|$, we know that there exists a fractional word of weight at most $\frac{1}{2}w|s|$ in the syndrome polytope (we show this in the full proof). Because of this and the fact that the decoder outputs the point of the syndrome polytope of smallest weight, we are sure that the decoder will never output a correction of weight greater that $\frac{1}{2}w|s|$. In particular if one of the minimal errors equivalent to the actual error has a weight greater than $\frac{1}{2}w|s|$ we are sure that the QLPD will not decode properly. See full proof in the supplementary material. □

This result implies in particular that the QLPD can only perform well for codes for quantum codes having the property that a correctable error cannot have a syndrome that has a weight significantly smaller than the error weight. This property of a code is sometimes called soundness [4], [8].

IV. Simulations

For the numerical simulations we sample errors from the uncorrelated X-Z noise model. Moreover since the decoding problems for X- and Z- type errors are equivalent it is enough to simulate only the decoding of X-type errors to compute the decoding performances. Thus we will sample only X-type errors and solve the decoding problem $s = He'^T$.

We performed the decoding using QLPD, but we do not abort when the solution is fractional but we rather round it to the closest integer. We use the name RQLPD for this variant. We applied this decoder to both toric codes and random hypergraph product codes, looking for evidence of a threshold (i.e., below a constant physical error rate, errors are decoding with high probability) and comparing the block error rate with the one of other algorithms.

Figure 3 shows numerical simulations on toric codes with various block lengths. In the presence of a threshold, we expect that below some fixed physical error rate, the probability of a logical error decreases with the block length. This is not what is observed in Figure 3, the different curves cross each other at different values. This is consistent with the theoretical results of Section III-C, that prove that there are some constant-size errors for the toric code that cannot be corrected by the QLPD.

Figure 4 shows numerical simulations for HPC obtained by taking the product of a regular classical LDPC code by itself, for codes with check degree equal to 4 and bit degree equal to 3. The classical LDPC codes are generated randomly with the PEG algorithm [15]. Even though the length 225

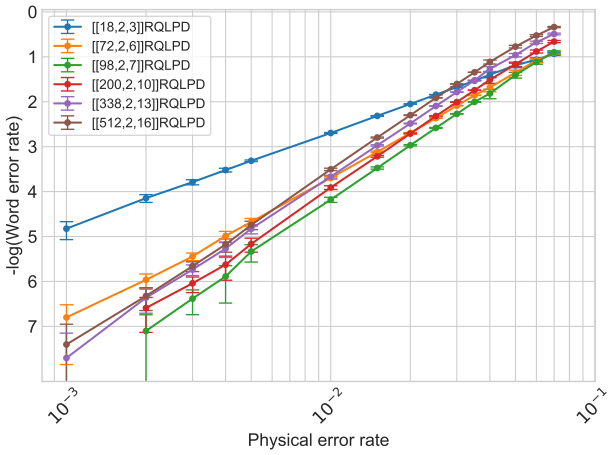

Fig. 3: Word error rate for the RQLPD on toric codes using independent X-type errors as noise model.

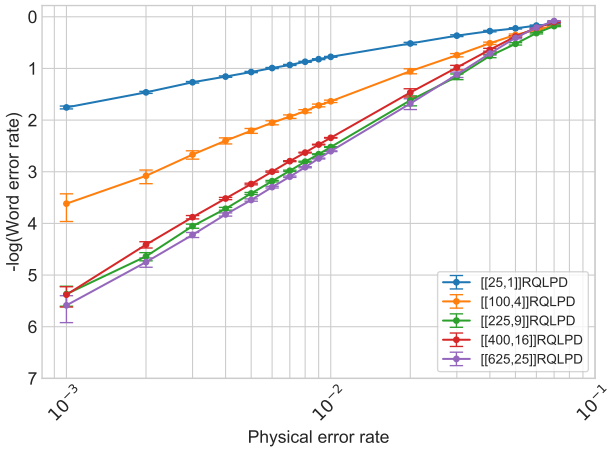

Fig. 4: Word error rate for the RQLPD on HPC starting from classical 3,4-regular random LDPC codes using independent X-type errors as noise model.

code seems to exhibit performance close to that of the code of length 625, this might simply result from the fact that the classical code used to design it was particularly good. Looking only at the other codes, the simulation results are consistent with a threshold around 6.5%. Note that this is not in contradiction with Theorem 4 as random HPC usually have a good soundness property (see e.g., [8, Corollary 9]).

Finally we compared in Figure 5 the performance of various decoding algorithms: RQLPD, belief propagation (BP) decoder, BP+SSF decoder [7] and BP+OSD0 decoder [10] and [11] for the smallest and largest HPC. While performances are close for the code of small block length, differences appear for the codes with larger block length. As expected, BP alone performs poorly. Interestingly, the RQLPD decoder provides improved performance compared to BP+SSF, and performs almost as well as BP+OSD0 which is the state of the art decoder for random HPC.

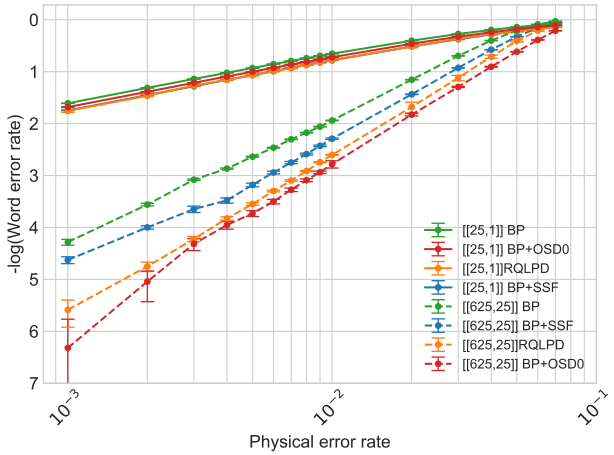

Fig. 5: Word error rate for several decoders on HPC starting from classical 3,4-regular random LDPC codes using independent X-type errors as noise model.

Acknowledgements

We thank Hamza Fawzi for initial discussions at the start of this project. OF acknowledges support from the European Research Council (ERC Grant Agreement No. 851716). LG and AL acknowledge support from the ANR through the QuantERA project QCDA.

References

[1] Siddharth Barman, Xishuo Liu, Stark C. Draper, and Benjamin Recht. Decomposition Methods for Large Scale LP Decoding, 2013.

[2] S. B. Bravyi and A. Yu. Kitaev. Quantum codes on a lattice with boundary, 1998.

[3] Sergey Bravyi and Matthew B Hastings. Homological product codes. In *STOC*, pages 273–282, 2014.

[4] Earl T Campbell. A theory of single-shot error correction for adversarial noise. *Quantum Science and Technology*, 4(2):025006, Feb 2019.

[5] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Efficient decoding of random errors for quantum expander codes. In *FOCS*, page 521–534, 2018.

[6] J. Feldman, M. Wainwright, and D. Karger. Using linear programming to decode binary linear codes. *IEEE TIT*, 51:954–972, 2005.

[7] Antoine Grospellier, Lucien Grouès, Anirudh Krishna, and Anthony Leverrier. Combining hard and soft decoders for hypergraph product codes, 2020.

[8] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. Quantum expander codes. *FOCS*, Oct 2015.

[9] J. X. Li and P. O. Vontobel. LP Decoding of Quantum Stabilizer Codes. In *ISIT*, pages 1306–1310, 2018.

[10] Pavel Panteleev and Gleb Kalachev. Degenerate Quantum LDPC Codes With Good Finite Length Performance, 2019.

[11] Joschka Roffe, David R. White, Simon Burton, and Earl T. Campbell. Decoding Across the Quantum LDPC Code Landscape, 2020.

[12] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE TIT*, 42:1710–1722, 1996.

[13] Mohammad H. Taghavi and Paul H. Siegel. Adaptive methods for linear programming decoding. *CoRR*, abs/cs/0703123, 2007.

[14] Jean-Pierre Tillich and Gilles Zémor. Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength. *IEEE TIT*, 60(2):1193–1202, Feb 2014.

[15] Xiao-Yu Hu, E. Eleftheriou, and D. . Arnold. Progressive edge-growth tanner graphs. In *IEEE GLOBECOM*, volume 2, pages 995–1001 vol.2, 2001.