Linked-Cluster Technique for Finding the Distance of a Quantum LDPC Code

Alexey A. Kovalev
Department of Physics & Astronomy
University of California
Riverside, CA 92521, USA
Email: alexey.kovalev@ucr.edu

Ilya Dumer
Department of Electrical Engineering
University of California
Riverside, CA 92521, USA
Email: dumer@ee.ucr.edu

Leonid P. Pryadko
Department of Physics & Astronomy
University of California
Riverside, CA 92521, USA
Email: leonid@ucr.edu

***Abstract*—We present a linked-cluster technique for calculating the distance of a quantum LDPC code. It offers an advantage over existing deterministic techniques for codes with small relative distances (which includes all known families of quantum LDPC codes), and over the probabilistic technique for codes with sufficiently high rates.**

I. Introduction

A practical implementation of a quantum computer will rely on quantum error correction (QEC) [1]–[3] due to the fragility of quantum states. There is a strong belief that surface (toric) codes [4], [5] can offer the fastest route to scalable quantum computation due to the error threshold around 1% and the locality of required gates [6]–[9]. Unfortunately, in the nearest future, the surface codes (in fact, any two-dimensional codes with local stabilizer generators [10]) can only lead to proof of the principle realizations as they encode a limited number of qubits (k), making any implementation of a useable quantum computer large (e.g., 2.2×10^8 physical qubits are required for a useful realization of Shor's algorithm [11]).

Lifting the restriction of locality but preserving the condition that the stabilizer generators should only involve a limited number of qubits, one gets the quantum LDPC codes, or, more precisely, quantum sparse-graph codes [12], [13]. Unlike the surface codes, these more general quantum LDPC codes can have a finite rate. On the other hand, while there are no known upper bounds on the parameters of such codes, in practice, all families of quantum LDPC codes where the upper limit on the distance is known, have the distance scaling as a square root of the block length [14]–[17]. Nevertheless, such codes (in fact, any family of quantum or classical LDPC codes with limited weights of the columns and rows of the parity check matrix, and distance scaling as a power or a logarithm of the block length n) have a finite error probability threshold, both in the standard setting where syndrome is measured exactly, and with the syndrome measurement errors [18].

Given that non-local two-qubit gates are relatively inexpensive with floating gates [19], superconducting and trapped-ion qubits, as well as more exotic schemes with teleportation [20]–[25], a quantum computer relying on quantum LDPC codes is quite feasible. An example of a universal set of gates based on dynamical decoupling pulses for an arbitrary number of qubits with Ising couplings forming a bipartite graph (e.g., the Tanner graph corresponding to a quantum LDPC code) has been recently suggested by one of us [26].

Compared to general quantum codes, with a quantum LDPC code, each quantum measurement involves fewer qubits, measurements can be done in parallel, and also the classical processing could potentially be enormously simplified (note, however, that belief-propagation and related decoding algorithms that work so well for classical LDPC codes [27], [28] may falter in the quantum case [29]). Compared to surface codes, more general LDPC codes have higher rates, which translates in a large reduction of the total number of qubits necessary to build a useful quantum computer. Note that while our analytical threshold estimate in Ref. [18] is quite low, there are examples of quantum LDPC codes demonstrated to beat the bounded distance decoding limit [30]. Overall, it is quite plausible that the operation of quantum computers of the future will rely on (non-local) quantum LDPC codes.

The very general proof [18] of the existence of a finite error probability threshold for quantum and classical LDPC codes with asymptotically zero relative distance is based on a simple observation that errors for such codes are likely to form small clusters affecting disjoint sets of stabilizer generators (parity check matrix rows). While the total weight of an error could be huge, the error can be surely detected if the size of each cluster is smaller than the code distance. Thus, in the case of the error detection, the threshold problem is related to the cluster size distribution for site percolation on a graph related to the Tanner graph of the code.

In this work, we apply the idea of error clustering with LDPC codes to design a numerical algorithm for finding a distance of such a code. The basic principle is formulated in Theorem 1: to find the distance of a code, one only needs to check error configurations corresponding to connected error clusters. For any error weight $w \ll n$, the number of such clusters is exponentially smaller than that of generic errors of the same weight. We consider the complexity of several well-known classical algorithms for finding code distance in application to quantum error correcting code. We conclude that the clustering algorithm beats deterministic techniques at sufficiently small relative distances (asymptotically at large n, all known families of quantum LDPC codes have zero relative distance), and the probabilistic technique for high-rate codes

with small relative distances.

II. Background.

A. Error-correcting codes

A q-ary linear code $\mathcal{C}$ with parameters $[n,k,d]_q$ is a k-dimensional subspace of the vector space $\mathbb{F}_q^n$ of all q-ary strings of length n. Code distance d is the minimal Hamming weight (number of non-zero elements) of a non-zero string in the code. A linear code is uniquely specified by the parity check matrix H, namely $\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n | H\mathbf{c} = 0\}$, where operations are done according to the $\mathbb{F}_q$ algebra.

A quantum $[[n,k,d]]$ (qubit) stabilizer code $\mathcal{Q}$ is a 2^k-dimensional subspace of the n-qubit Hilbert space $\mathbb{H}_2^{\otimes n}$, a common $+1$ eigenspace of all operators in an Abelian *stabilizer group* $\mathscr{S} \subset \mathscr{P}_n$, $-\mathbb{1} \notin \mathscr{S}$, where the n-qubit Pauli group $\mathscr{P}_n$ is generated by tensor products of the X and Z single-qubit Pauli operators. The stabilizer is typically specified in terms of its generators, $\mathscr{S} = \langle S_1, \ldots, S_{n-k} \rangle$; measuring the generators S_i produces the *syndrome* vector. The weight of a Pauli operator is the number of qubits it affects. The distance d of a quantum code is the minimum weight of an operator U which commutes with all operators from the stabilizer $\mathscr{S}$, but is not a part of the stabilizer, $U \notin \mathscr{S}$. A code of distance d can detect any error of weight up to $d-1$, and correct up to $\lfloor d/2 \rfloor$.

A Pauli operator $U \equiv i^m X^{\mathbf{v}} Z^{\mathbf{u}}$, where $\mathbf{v}, \mathbf{u} \in \{0,1\}^{\otimes n}$ and $X^{\mathbf{v}} = X_1^{v_1} X_2^{v_2} \ldots X_n^{v_n}$, $Z^{\mathbf{u}} = Z_1^{u_1} Z_2^{u_2} \ldots Z_n^{u_n}$, can be mapped, up to a phase, to a quaternary vector, $\mathbf{e} \equiv \mathbf{u} + \omega \mathbf{v}$, where $\omega^2 \equiv \overline{\omega} \equiv \omega + 1$. A product of two quantum operators corresponds to a sum $(\mathrm{mod}\, 2)$ of the corresponding vectors. Two Pauli operators commute if and only if the *trace inner product* $\mathbf{e}_1 * \mathbf{e}_2 \equiv \mathbf{e}_1 \cdot \overline{\mathbf{e}}_2 + \overline{\mathbf{e}}_1 \cdot \mathbf{e}_2$ of the corresponding vectors is zero, where $\overline{\mathbf{e}} \equiv \mathbf{u} + \overline{\omega}\mathbf{v}$.

With this map, generators of a stabilizer group are mapped to rows of a parity check matrix H of an *additive* (forming a group with respect to addition but not necessarily over the full set of $\mathbb{F}_4$ operations) code over $\mathbb{F}_4$, with the condition that the trace inner product of any two rows vanishes [31]. The vectors generated by rows of H correspond to stabilizer generators which act trivially on the code; these vectors form the *degeneracy group* and are omitted from the distance calculation. For a more narrow set of CSS codes the parity check matrix is a direct sum $H = G_x \oplus \omega G_z$, and the commutativity condition simplifies to $G_x G_z^T = 0$.

An LDPC code, quantum or classical, is a code with a sparce parity check matrix. For a *regular* (j,l) LDPC code, every column and every row of H have weights j and l respectively, while for a (j,l)-limited LDPC code these weigths are limited from above by j and l.

The huge advantage of classical LDPC codes is that they can be decoded in linear time using belief propagation (BP) and related iterative methods [27], [28]. Unfortunately, this is not necessarily the case for quantum LDPC codes: Tanner graphs for quantum codes have many short loops of length 4, which cause a dramatic deterioration of the convergence of the BP algorithm [29]. This problem can be circumvented with specially designed quantum codes [17], [30], but a general solution is not known. One alternative which has polynomial complexity is n, approaching linear for very small error rates, is the cluster-based decoding suggested in Ref. [18].

III. Generic numerical techniques for distance calculation

The problem of numerically calculating the distance of a linear code (finding the minimum-weight codeword in the code) is related to the decoding problem: find the most likely (minimum-weight in the case of the q-ary symmetric channel) error which gives the same syndrome as the received codeword. The number of required steps N usually scales exponentially with the blocklength n, $N \propto q^{Fn}$; we characterize the complexity by the exponent F. For example, for a linear q-ary code with k information qubits, there are q^k distinct codewods, going over each of the codewords has the complexity exponent $F = R$, where $R = k/n$ is the code rate. When used for decoding, one can instead store the list of all q^{n-k} syndromes and coset leaders, which corresponds to the complexity $F = 1 - R$.

A. Sliding window technique

This decoding technique has been proposed in Ref. [32], and generalized in Ref. [33]. A related technique has also been independently invented in Refs. [34], [35]. For a q-ary code with relative distance $\delta \equiv d/n$, the complexity exponent is $F_A = RH_q(\delta)$, where $H_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$ is the q-ary entropy function; for a code with the rate $R \equiv k/n$ on the Gilbert-Varshamov bound, $R = 1 - H_q(\delta)$, this gives the complexity exponent $F_A^{(GV)} = R(1-R)$, reaching the maximum of $1/4$ at $R = 1/2$.

The idea is to use only $k + o(n)$ consecutive positions to recover any codeword of a q-ary linear $[n,k]$ code. For example, any k consecutive positions suffice in a cyclic code. Similarly, it is easy to verify that in most (random) $k \times n$ generator matrices G any $s = k + 2\left\lfloor \log_q n \right\rfloor$ consecutive columns form a submatrix G_s of a maximum rank k. Thus, s (error free) consecutive bits suffice to recover a codeword in most random $[n,k]$ codes.

To find a codeword c of a minimum weight w, we choose a sliding window $I(i,s)$ that begins in a position $i = 0, \ldots, n-1$ and has length s. Our goal is to find the window that has the average Hamming weight, $v \equiv \lfloor ws/n \rfloor$. (Note that a sliding window can change its weight only by one when it moves from any position i to $i+1$; thus at least one of the n windows will have weight v.) For each i and for each $w = 1, 2, \ldots$, we encode all possible

$$L = (q-1)^v \binom{s}{v} \tag{1}$$

vectors of length s and weight v. We stop the procedure once we find an encoded codeword of weight w. The overall procedure has complexity of the order $Ln^2 \asymp q^{F_A n}$, where $F_A = RH_q(\delta)$.

Unfortunately, the performance suffers when the technique is applied to a quantum code. Indeed, the additive quaternary

code corresponding to an $[[n,k]]$ stabilizer code operates in a space with 4^n symbols with only $2^r = 4^{r/2}$ distinct syndromes, where $r \equiv n-k$ is the redundancy of the quantum code; the effective rate is thus[1] $R' = (n-r/2)/n = (1+R)/2$. The same effective rate is obtained if we take a CSS code with $\operatorname{rank} G_x = \operatorname{rank} G_z = (n-k)/2$, as there are $k' = n-(n-k)/2 = (n+k)/2$ information bits for both codes. In addition to an increased number of the information symbols, each obtained vector of small weight has to be tested on linear dependence with the rows of the parity check matrix H. In addition to the considered mapping, one can also map an additive $[[n,k,d]]$ code to a binary code with block length $3n$ and weight of each codeword doubled; such mapping typically gives a larger complexity and will not be considered here [31], [36].

For a generic stabilizer code with relative distance δ, the binary complexity exponent of the sliding-window technique is $F = 2R'H_4(\delta)$. Similarly, for a CSS code, the sliding-window technique gives the complexity exponent $F_{Aq} = 2R'H_2(\delta)$. Both results produce the same complexity exponent

$$F_{Aq}^{(GV)} = (1-R^2)/2$$

on the quantum GV bound, namely $R = 1-2H_4(\delta)$ for generic quantum codes [37], and $R = 1-2H_2(\delta)$ for CSS codes [38]. The dependence $F_{Aq}^{(GV)}(R)$ is shown in Fig. 1 with a solid red line. Note that for codes with small relative distance δ, the complexity exponent is logarithmic in δ, e.g., $F_{Aq} \sim \delta(1+R)\log_2(e/\delta)$ in the case of a CSS code.

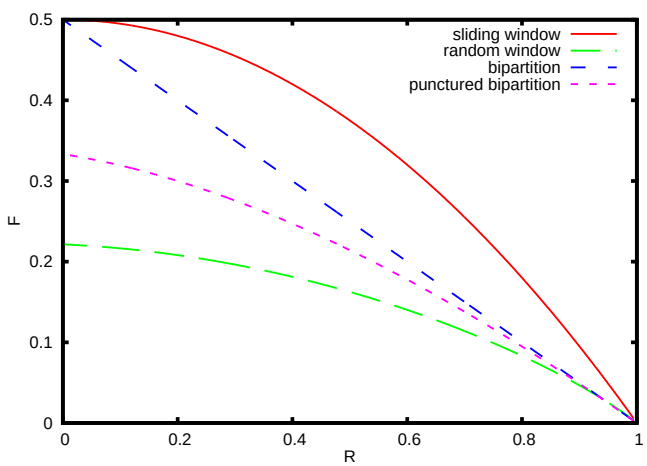

Fig. 1. Comparison of the binary complexity exponents for the four classical decoding techniques applied to quantum codes at the quantum GV bound, see Sec. III. Note that for high-rate codes, $R \to 1$, the curves for the sliding window and the random window techniques have logarithmically-divergent slopes, while the slopes for the two other techniques remain finite. In this limit of $R \to 1$ the punctured bipartition technique gives the best performance.

B. Random window technique [39]–[42]

Given a q-ary linear (n,k) code, we randomly choose $s = k+\tau$ positions with positive number $\tau = o(k)$. For any codeword, we wish to find an s-set of small weight t. Given an error pattern (or a codeword) of weight w, we only need to estimate the number of random trials $T_t(n,s,w)$ needed to find such set with a high probability. This number is well known (up to a factor of order n) [43] and it is upper-bounded by

$$T_0(n,s,w) \asymp \binom{n}{w}\Big/\binom{n-s}{w}. \tag{2}$$

To determine the distance of a code, we choose $w = 1, 2, \ldots$. Then we perform $nT_0(n,s,w)$ trials of choosing s random positions.

Note that a randomly chosen $k \times s$ submatrix G_s of a random generator matrix G has full rank k with a high probability $1-q^{-\tau}$ (also, most matrices G have *all possible* submatrices G_k of rank $k-n^{1/2}$ or more). If the current s-set includes any information k-subset, we only consider s vectors $(0...010...0)$ of weight $t=1$. We then re-encode them into the codewords of length n. Otherwise, we discard an s-set and proceed further. We stop the algorithm, once we obtain a codeword of weight w. The overall complexity has the order of $n^4 T_0(n,s,w)$, and it is independent of q, in contrast to the sliding-window technique. This corresponds to the binary complexity exponent $F_B = H_2(\delta) - (1-R)H_2(\delta/(1-R))$. For binary random linear codes meeting the GV bound, the complexity exponent is easily verified to be

$$F_B^{(GV)} = (1-R)\left(1 - H_2\left(\delta/(1-R)\right)\right),$$

where $H_2(\cdot)$ is the binary entropy and $\delta \equiv H_2^{-1}(1-R)$ is the relative GV distance w/n. In particular, $F \approx 0.11$ for the rate $R = 1/2$. For small $w \le (n-k)^{1/2}$, we can also use a simpler estimate

$$T_0(n,s,w) \asymp \left(\frac{n}{n-s}\right)^w \asymp (1-R)^{-w}$$

that has the exponent linear in code distance w.

Just as the sliding window technique, this technique relies on recoding (re-encoding). Thus, the quantum complexity can be obtained by substituting the effective rate $R' = (1+R)/2$. In particular, for a generic stabilizer code meeting the quantum GV bound, we have the binary complexity exponent as shown in Fig. 1 with the green dashed line; it reaches the maximum of $F_{\max} \approx 0.22$ at $R = 0$, i.e., for small-rate codes.

C. Bipartition technique [44]

The idea is to use a sliding ("left") window of length $s_l = \lfloor n/2 \rfloor$ starting in any position i. For any vector of weight w, at least one position i will produce a window of weight $v_l = \lfloor w/2 \rfloor$. The remaining (right) window of length $s_r = \lceil n/2 \rceil$ will have the weight $v_r = \lceil w/2 \rceil$. We calculate the syndromes of all vectors e_l and e_r of weights v_l and v_r on the left and right windows, respectively, and try to find a pair of vectors $\{e_l, e_r\}$ that produce identical syndromes, and therefore form a codeword. Clearly, each set $\{e_l\}$ and $\{e_r\}$ have exponential size of order $L = (q-1)^{w/2}\binom{n/2}{w/2}$. Finding two elements e_l, e_r with equal syndromes can be performed, e.g., by sorting the elements of the combined set, or, to save on memory,

[1] This construction is analogous to pseudogenerators introduced in Ref. [36].

sorting the elements of the left set and using binary search for each of the syndromes from the right set. This has a similar complexity of order $L \log_2 L$. Thus, finding a code vector of weight $w = \delta n$ requires complexity of order

$$q^{F_C n},\ F_C = H_q(\delta)/2.$$

For random binary codes which meet the GV bound, we have exponent $F_C^{(GV)} = (1-R)/2$. For code rate $R = 1/2$, this gives $F_C^{(GV)} = F_A^{(GV)} = 1/4$. For higher code rates, the bipartition technique gives exponent $F_C^{(GV)} < F_A^{(GV)}$. It can also be verified that $F_C^{(GV)} < F_B^{(GV)}$ for code rates R approaching 1. In addition, bipartition technique is guaranteed to work with any linear code, as opposed to two previous techniques provably valid for random codes.

The bipartition technique is also the only technique that can be transferred to quantum codes without any performance loss. For a generic quantum code and a CSS code corresponding binary complexity exponents are $F_{Cq} = H_4(\delta)$ and $H_2(\delta)$, respectively. On the quantum GV bound, this gives binary exponent $F_{Cq}^{(GV)} = (1-R)/2$ in both cases, see Fig. 1. Note that this line is always below that for $F_{Aq}^{(GV)}$, and for high-rate codes the corresponding line is below that for the random window technique, $F_{Bq}^{(GV)}$.

D. Punctured bipartition technique [45]

Here we combine the sliding-window technique with bipartition. Consider a relatively large sliding window of length

$$s = \lceil 2nR/(1+R) \rceil . \tag{3}$$

Note that most random $[n,k]$ codes include at least one information set on any sliding s-window $I(i,s)$ with initial position $i = 0, ..., n-1$. Thus, any such window forms a punctured linear $[s,k]$ code with a smaller redundancy $s-k$. Also, any codeword of weight w has weight $v = \lfloor ws/n \rfloor$ on some sliding window. For simplicity, let s and v be even. We then use bipartition on each s-window and consider all vectors e_l and e_r of weight $v/2$ on either half of length $s/2$. The corresponding sets $\{e_l\}$ and $\{e_r\}$ have size $L_s = (q-1)^{v/2}\binom{s/2}{v/2}$. We then seek all matching pairs $\{e_l, e_r\}$ that have the same syndrome h. Each such pair $\{e_l, e_r\}$ represents some code vector of the punctured $[s,k]$ code and is re-encoded to the full length n. For each $w = 1, 2, ...$, we stop the procedure once we find a re-encoded vector of weight w. Obviously, this technique can lower the complexity to the order L_s. Note, however, that many vectors e_l and e_r of length $s/2$ can simultaneously have the same syndrome h of size $s-k$. Thus, our task is to encode *all code vectors* of weight v in a random $[s,k]$ code. It can be shown [45] that our choice of parameter s limits the number of such codewords by the same order L_s. Thus, we can find any codeword of weight $w = \delta n$ with a smaller complexity

$$q^{F_C s} = q^{F_D n},\ F_D = H_q(\delta)R/(1+R).$$

For codes meeting the GV bound, $F_D^{(GV)} = R(1-R)/(1+R)$. Note however, that this combined technique cannot be provably applied to any linear code, in contrast to a simpler bipartition technique.

Somewhat similarly to regular case in Sec. III-C, the performance of the bipartition in this technique is not affected when we consider quantum codes. However, in the expression (3) for the optimal block size, one needs to use the effective quantum rate $R' = (1+R)/2$. As a result, the complexity exponent for regular stabilizer codes becomes

$$F_{Dq} = \frac{2R'}{1+R'} H_4(\delta) = \frac{2(1+R)}{3+R} H_4(\delta); \tag{4}$$

on the GV bound this gives

$$F_{Dq}^{(GV)} = \frac{(1-R^2)}{3+R}.$$

This technique is the best for high-rate quantum codes, $R \to 1$.

IV. Linked-cluster technique

Here we present a technique which is designed specifically for very sparse quantum LDPC codes, as an alternative to the belief propagation technique.

For a (j,ℓ)-limited LDPC code, we represent all (qu)bits as nodes of a graph $\mathcal{G}_1$ of degree at most z: two nodes are connected by an edge iff there is a row in the parity check matrix which has non-zero values at both positions. An error with support in a subset $\mathcal{E} \subseteq V(\mathcal{G}_1)$ of the vertices defines the subgraph $\mathcal{G}_1(\mathcal{E})$ induced by $\mathcal{E}$. Generally, we will not make a distinction between a set of vertices and the corresponding induced subgraph. In particular, a (connected) cluster in $\mathcal{E}$ corresponds to a connected subgraph of $\mathcal{G}_1(\mathcal{E})$. Different clusters affect disjoint sets of rows of the parity check matrix. This implies the following

Theorem 1. *The support of a minimum-weight code word of a q-ary code with the parity check matrix $\mathcal{H}$ forms a linked cluster on $\mathcal{G}_1$.*

Proof. Indeed, let us assume this is not so, and a minimum-weight code word $\mathbf{c}$ is supported by two or more disconnected parts. By construction, these affect different rows of the parity check matrix and, therefore, the vectors corresponding to subsets of non-zero symbols in $\mathbf{c}$ are in the null-space of $\mathcal{H}$, contrary to the assumption that $\mathbf{c}$ has minimum weight. □

Thus, in order to determine the distance of a code by an exhaustive search, we do not have to list all error patterns; instead, one can go over all linked clusters of increasing sizes. We used the following variant of the breadth-first algorithm to construct all linked cluster of a given size w:

Start with a position $i = 0, 1, \ldots, n-w$, and add to the list all neighboring positions to the right of i. At each subsequent level of recursion, only go over the positions in the list to the right of the position added on the previous level. Once a new position is selected, add all new neighboring positions which are to the right of the original starting point i. The recursion should stop after the desired cluster size w is reached. This

way, the algorithm generates all clusters of size w, and no repeated clusters are produced. In the case of a binary code, each linked cluster of weight w directly corresponds to a potential code word of same weight. In the case of a q-ary code, one needs to check the rank of a matrix formed by the corresponding columns of the parity check matrix.

The upper cup on the total number of the linked clusters of size w for a given (j,ℓ)-limited LDPC code can be obtained from the cluster distribution for a regular tree . The degrees of the graph $\mathcal{G}_1$ are limited from above by $z \equiv (\ell-1)j$. Among the degree-limited graphs, the z-regular tree has the largest number of clusters (it does not have any loops). Namely, the number of weight-w clusters containing a given vertex is [46]

$$N_w = \frac{z}{w-1}\binom{(z-1)w}{w-2} \asymp \frac{z}{(z-2)^2 w} 2^{(z-1)wH_2(1/(z-1))}, \tag{5}$$

where the asymptotic form is valid for large w. Note that the loops present in the actual graph tend to reduce the exponent; also, at $w \gtrsim n/(z-1)$ there is further reduction in N_w due to finite-size effect. Thus, we expect that for $w \lesssim n/(z-1)$, the complexity exponent for the linked-cluster method can be written as

$$F_{\rm LC} = \delta(z_{\rm eff}-1)H_2(1/(z_{\rm eff}-1)) \asymp \delta \log_2(e(z_{\rm eff}-1)), \tag{6}$$

where $z_{\rm eff} < z$. For example, a number of generalized hypergraph-product codes have been constructed in Ref. [16] from different binary cyclic codes. Codes originating from the same check polynomial correspond to graphs with the same local structure as the graph $\mathcal{G}_1$ for the original hypergraph-product codes [14]. Examples of the cluster-number scaling with weight for several such codes are given in Fig. 2.

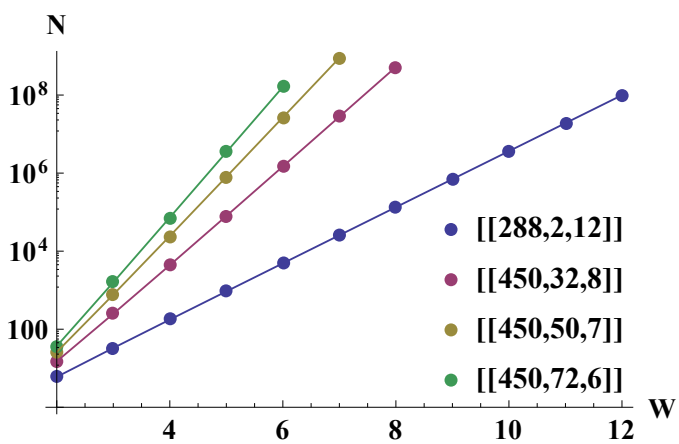

Fig. 2. Dependence of the number of clusters N on the corresponding weight w for hypergraph-product codes obtained from cyclic codes with the check polynomials of weights $w_h = 2, 3, 4$, and 5. The corresponding parity-check matrices are $(w_h, 2w_h)$-regular, and the graphs $\mathcal{G}_1$ have degrees $z = (l-1)j = (2w_h-1)w_h = 6, 15, 28$, and 45. The fits to $N = Ay^w$ give $y \equiv e(z_{\rm eff}-1) = 5.2, 18.8, 33.4$, and 47.0, respectively.

While the performance of the cluster-based technique deteriorates rapidly with large z, and for larger distances, one advantage evident from Eq. (6) is that the complexity exponent is proportional to the relative distance δ. In comparison, any other deterministic technique in Sec. III has the complexity scaling as $F \propto \delta \log(1/\delta)$ in this limit. Thus, the presented linked-cluster technique has the best asymptotic performance for all known quantum LDPC codes with limited-weight stabilizer generators, where $\delta \propto n^{-1/2}$. Compared with the random window technique (which has the smallest complexity exponent in a wide range of rates), $F_{Bq} \asymp \delta \log_2(1/(1-R))$ for small relative distances, also linear in δ, this technique is expected to win at rates such that $1-R \lesssim (e\, z_{\rm eff})^{-1}$.

V. Conclusion

We suggested a cluster-based technique for finding the distance of very sparse quantum LDPC codes. It beats the existing non-probabilistic algorithms for codes with sufficiently small relative distances (all known families of quantum LDPC codes have distance scaling as $n^{1/2}$ or lower at large n). It also beats the probabilistic random window technique for codes with sufficiently high rates.

Acknowledgment

This work was supported in part by the U.S. Army Research Office Grant No. W911NF-11-1-0027, and by the NSF Grant No. 1018935.

References

[1] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, p. R2493, 1995. [Online]. Available: http://link.aps.org/abstract/PRA/v52/pR2493

[2] E. Knill and R. Laflamme, "Theory of quantum error-correcting codes," *Phys. Rev. A*, vol. 55, pp. 900–911, 1997. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.55.900

[3] C. Bennett, D. DiVincenzo, J. Smolin, and W. Wootters, "Mixed state entanglement and quantum error correction," *Phys. Rev. A*, vol. 54, p. 3824, 1996. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.54.3824

[4] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Ann. Phys.*, vol. 303, p. 2, 2003. [Online]. Available: http://arxiv.org/abs/quant-ph/9707021

[5] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *J. Math. Phys.*, vol. 43, p. 4452, 2002. [Online]. Available: http://dx.doi.org/10.1063/1.1499754

[6] R. Raussendorf and J. Harrington, "Fault-tolerant quantum computation with high threshold in two dimensions," *Phys. Rev. Lett.*, vol. 98, p. 190504, 2007. [Online]. Available: http://link.aps.org/abstract/PRL/v98/e190504

[7] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, "Surface code quantum computing with error rates over 1%," *Phys. Rev. A*, vol. 83, p. 020302, Feb 2011. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.83.020302

[8] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Phys. Rev. A*, vol. 86, p. 032324, Sep 2012. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevA.86.032324

[9] H. Bombin, R. S. Andrist, M. Ohzeki, H. G. Katzgraber, and M. A. Martin-Delgado, "Strong resilience of topological codes to depolarization," *Phys. Rev. X*, vol. 2, p. 021004, Apr 2012. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevX.2.021004

[10] S. Bravyi, D. Poulin, and B. Terhal, "Tradeoffs for reliable quantum information storage in 2d systems," *Phys. Rev. Lett.*, vol. 104, p. 050503, Feb 2010. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.104.050503

[11] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "A primer on surface codes: Developing a machine language for a quantum computer," *ArXiv e-prints*, Aug. 2012.

[12] M. S. Postol, "A proposed quantum low density parity check code," 2001, unpublished. [Online]. Available: http://arxiv.org/abs/quant-ph/0108131

[13] D. J. C. MacKay, G. Mitchison, and P. L. McFadden, "Sparse-graph codes for quantum error correction," *IEEE Transactions on Information Theory*, vol. 59, pp. 2315–30, 2004. [Online]. Available: http://dx.doi.org/10.1109/TIT.2004.834737

[14] J.-P. Tillich and G. Zemor, "Quantum ldpc codes with positive rate and minimum distance proportional to $\sqrt{n}$," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, 28 2009-july 3 2009, pp. 799 –803.

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

[16] ——, "Quantum "hyperbicycle" low-density parity check codes with finite rate," 2012, unpublished. [Online]. Available: http://arxiv.org/abs/1212.6703

[17] I. Andriyanova, D. Maurice, and J.-P. Tillich, "New constructions of CSS codes obtained by moving to higher alphabets," 2012, unpublished.

[18] A. A. Kovalev and L. P. Pryadko, "Fault-tolerance of "bad" quantum low-density parity check codes," 2012, submitted to Phys. Rev. Lett. [Online]. Available: http://arxiv.org/abs/1208.2317

[19] L. Trifunovic, O. Dial, M. Trif, J. R. Wootton, R. Abebe, A. Yacoby, and D. Loss, "Long-distance spin-spin coupling via floating gates," *Phys. Rev. X*, vol. 2, p. 011006, Jan 2012.

[20] T. Yamamoto, Y. A. Pashkin, O. Astafiev, Y. Nakamura, and J. S. Tsai, "Demonstration of conditional gate operation using superconducting charge qubits," *Nature*, vol. 425, pp. 941–4, 2003. [Online]. Available: http://dx.doi.org/10.1038/nature02015

[21] R. McDermott, R. W. Simmonds, M. Steffen, K. B. Cooper, K. Cicak, K. D. Osborn, S. Oh, D. P. Pappas, and J. M. Martinis, "Simultaneous state measurement of coupled josephson phase qubits," *Science*, vol. 307, p. 1299, 2005. [Online]. Available: http://dx.doi.org/10.1126/science.1107572

[22] J. Benhelm, G. Kirchmair, C. F. Roos, and R. Blatt, "Towards fault-tolerant quantum computing with trapped ions," *Nature Physics*, vol. 4, pp. 463–466, 2008. [Online]. Available: http://dx.doi.org/10.1038/nphys961

[23] A. Friedenauer, H. Schmitz, J. T. Glueckert, D. Porras, and T. Schaetz, "Simulating a quantum magnet with trapped ions," *Nature Physics*, 2008.

[24] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and einstein-podolsky-rosen channels," *Phys. Rev. Lett.*, vol. 70, pp. 1895–1899, Mar 1993. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.70.1895

[25] D. Gottesman and I. L. Chuang, "Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations," *Nature*, vol. 402, pp. 390–393, 1999. [Online]. Available: http://dx.doi.org/10.1038/46503

[26] A. De and L. P. Pryadko, "Universal set of scalable dynamically corrected gates for quantum error correction with always-on qubit couplings," 2013, phys. Rev. Letters, to be published. [Online]. Available: http://arxiv.org/abs/1209.2764

[27] R. Gallager, "Low-density parity-check codes," *Information Theory, IRE Transactions on*, vol. 8, no. 1, pp. 21 –28, january 1962.

[28] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.

[29] D. Poulin and Y. Chung, "On the iterative decoding of sparse quantum codes," *Quant. Info. and Comp.*, vol. 8, p. 987, 2008.

[30] K. Kasai, M. Hagiwara, H. Imai, and K. Sakaniwa, "Quantum error correction beyond the bounded distance decoding limit," *Information Theory, IEEE Transactions on*, vol. 58, no. 2, pp. 1223 –1230, feb. 2012.

[31] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, "Quantum error correction via codes over GF(4)," *IEEE Trans. Inf. Th.*, vol. 44, pp. 1369–1387, 1998. [Online]. Available: http://dx.doi.org/10.1109/18.681315

[32] G. S. Evseev, "Complexity of decoding for linear codes." *Probl. Peredachi Informacii (USSR)*, vol. 19, pp. 3–8, 1983, [Probl. Inf. Transm. (USSR), vol. 19, p. 1-6 (1983)]. [Online]. Available: http://mi.mathnet.ru/ppi1159

[33] I. Dumer, "Suboptimal decoding of linear codes: partition technique," *Information Theory, IEEE Transactions on*, vol. 42, no. 6, pp. 1971 –1986, nov 1996.

[34] K.-H. Zimmermann, "Integral hecke modules, integral generalized reed-muller codes, and linear codes," Technische Universit at Hamburg-Harburg, Tech. Rep. Tech. Rep. 3-96, 1996.

[35] M. Grassl, "Searching for linear codes with large minimum distance," in *Discovering Mathematics with Magma*, ser. Algorithms and Computation in Mathematics, W. Bosma and J. Cannon, Eds. Springer Berlin Heidelberg, 2006, vol. 19, pp. 287–313. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-37634-7_13

[36] G. White and M. Grassl, "A new minimum weight algorithm for additive codes," in *Information Theory, 2006 IEEE International Symposium on*, july 2006, pp. 1119 –1123.

[37] K. Feng and Z. Ma, "A finite gilbert-varshamov bound for pure stabilizer quantum codes," *Information Theory, IEEE Transactions on*, vol. 50, no. 12, pp. 3323 – 3325, dec. 2004.

[38] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," *Phys. Rev. A*, vol. 54, no. 2, pp. 1098–1105, Aug 1996.

[39] J. S. Leon, "A probabilistic algorithm for computing minimum weights of large error-correcting codes," *Information Theory, IEEE Transactions on*, vol. 34, no. 5, pp. 1354 –1359, sep 1988.

[40] J. Stern, "A method for finding codewords of small weight," *in* Coding Theory and Applications*, ser. Lecture Notes in Computer Science, G. Cohen and J. Wolfmann, Eds.*, vol. 388, pp. 106 –113, 1989.

[41] E. A. Kruk, "Decoding complexity bound for linear block codes," *Probl. Peredachi Inf.*, vol. 25, no. 3, pp. 103–107, 1989, (In Russian). [Online]. Available: http://mi.mathnet.ru/eng/ppi665

[42] J. T. Coffey and R. M. Goodman, "The complexity of information set decoding," *Information Theory, IEEE Transactions on*, vol. 36, no. 5, pp. 1031 –1037, sep 1990.

[43] P. Erdos and J. Spencer, *Probabilistic methods in combinatorics*. Budapest: Akademiai Kiado, 1974.

[44] I. I. Dumer, "Two decoding algorithms for linear codes," *Probl. Peredachi Inf. (USSR)*, vol. 25, pp. 24–32, 1989, [Probl. Inf. Transm., 25, 17-23 (1989)]. [Online]. Available: http://mi.mathnet.ru/ppi635

[45] I. Dumer, "Soft-decision decoding using punctured codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 1, pp. 59–71, 2001.

[46] C.-K. Hu, "Exact cluster size distributions and mean cluster sizes for the q-state bond-correlated percolation model," *Journal of Physics A: Mathematical and General*, vol. 20, no. 18, p. 6617, 1987. [Online]. Available: http://stacks.iop.org/0305-4470/20/i=18/a=059