Fault-tolerant weighted union-find decoding on the toric code

Shilin Huang,[*] Michael Newman,[†] and Kenneth R. Brown[‡]
Departments of Electrical and Computer Engineering, Chemistry, and Physics, Duke University, Durham, North Carolina 27708, USA

(Received 20 April 2020; accepted 26 June 2020; published 16 July 2020)

Quantum error correction requires decoders that are both accurate and efficient. To this end, union-find decoding has emerged as a promising candidate for error correction on the surface code. In this work, we benchmark a weighted variant of the union-find decoder on the toric code under circuit-level depolarizing noise. This variant preserves the almost-linear time complexity of the original while significantly increasing the performance in the fault-tolerance setting. In this noise model, weighting the union-find decoder increases the threshold from 0.38% to 0.62%, compared to an increase from 0.65% to 0.72% when weighting a matching decoder. Further assuming quantum nondemolition measurements, weighted union-find decoding achieves a threshold of 0.76% compared to the 0.90% threshold when matching. We additionally provide comparisons of timing as well as low error rate behavior.

DOI: 10.1103/PhysRevA.102.012419

I. INTRODUCTION

In order to realize scalable quantum computing, quantum information must be protected in quantum error correcting codes. Information about the errors occurring are rapidly extracted through measurements, and this information is processed through a decoder in order to determine which errors have occurred. These decoders must be accurate in providing good estimates for the error, but they should also be highly efficient in order to keep up with the quantum computation as it progresses.

One of the leading candidates for quantum error correction is the surface code [1–3], owing to its two-dimensional (2D) nearest-neighbor implementation [4], robust memory [5], optimized logical gates [6–8], and wealth of decoding schemes [9–36]. Among these schemes, decoding based on minimum-weight perfect matching (MWPM) is particularly promising due to its high performance, adaptability to circuit-level errors, and relative $O(n^3)$ efficiency on general graphs [37]. In particular, there has been significant effort aimed at accelerating and parallelizing MWPM [11,12,19].

However, performing decoding at the clock speed of a quantum computer remains a daunting task. A new type of decoder based on the union-find (UF) primitive has been proposed as an alternative to MWPM [35]. This decoder relies on generating an erasure consistent with the syndrome information, and then applying a highly efficient erasure decoder [38]. Moreover, the UF decoder remains competitive with the high performance of MWPM in a phenomenological error model [35,39,40].

In this work, we benchmark the UF decoder in the fault-tolerance setting under standard circuit-level depolarizing noise. We show that by adapting the decoder to weighted graphs, the performance increases substantially. This variant was first proposed in Ref. [36], however, it can be modified to preserve the almost-linear run time of the original UF decoder.

Weighting the decoder graph is a natural step that yields significant gains in the context of MWPM [29]. In particular, for a properly weighted graph, MWPM decides on the most likely error given a particular syndrome [1,4,29,41,42]. While UF decoding does not have a simple interpretation on weighted graphs, it is reasonable to expect that preferencing cluster growth in the direction of the most likely nearby error would be beneficial. What is remarkable is the degree to which it helps, with a significantly greater relative gain than weighted matching over unweighted matching.

II. WEIGHTED UNION-FIND

We follow the prescription of the original UF decoder described in Ref. [35], but with weighted edges on the decoder graph. The complexity of the original algorithm is dominated by the union-find primitive, which has complexity $O(n\alpha(n))$ [43], where α is the inverse Ackermann's function and n is the number of syndrome bits. For all practical sizes, this is essentially linear in n with a small constant. For fault-tolerant decoding in a distance d toric code, $n = 2d^2$ when averaged over $\propto d$ rounds of syndrome extraction. This approach straightforwardly generalizes to the open boundaries of the surface code, but we benchmark using periodic boundaries to minimize finite-size effects.

The UF decoder proceeds in two steps: syndrome validation, which is used to identify a candidate erasure given the syndromes, and peeling, which is used to decode the candidate erasures. The addition of edge weights changes only the growth step for each cluster during syndrome validation. In the original algorithm, we would iterate over all boundary vertices of the smallest boundary cluster and grow the incident boundary edges by one-half. In the weighted algorithm, we first iterate over the boundary edges to identify the smallest

[*]shilin.huang@duke.edu
[†]mgnewman@google.com
[‡]kenneth.r.brown@duke.edu

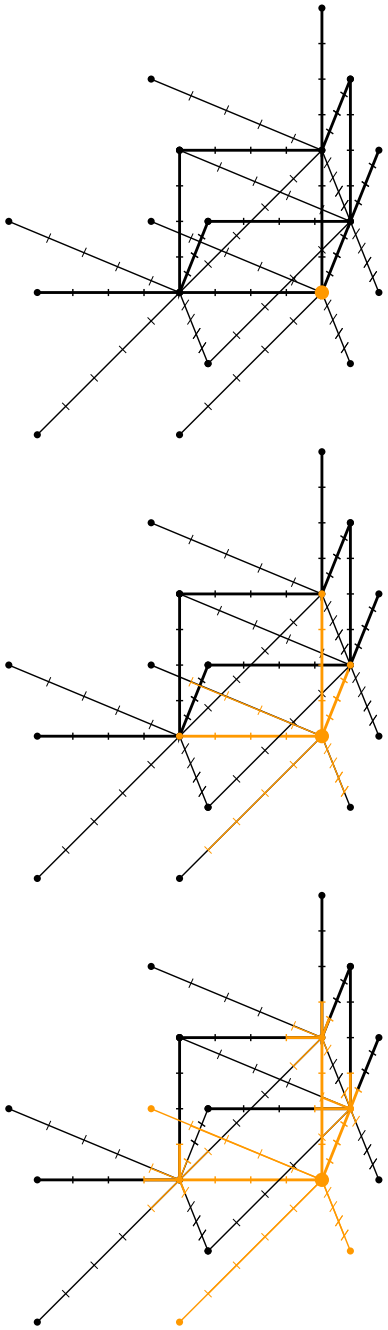


FIG. 1. Two growth steps for weighted UF on a toric code decoder graph with $p = 0.8\%$ and weights truncated to the nearest integer (for performance estimates, we truncate to the nearest tenth). Some edges are omitted for clarity. In this case, we have two types of edges: weight four edges (thick) in the cardinal directions, and weight five diagonal edges (thin). The orange (light) highlight indicates the growing cluster. In the top figure, a single excitation occurs in the corner of the decoder graph. In the middle, the cluster radius grows by four and it merges with clusters to the north, west, and up directions. At the bottom, the cluster radius grows by one and it merges with clusters diagonal to the original excitation.

boundary edge weight $w_{\min}$, and then again iterate over the boundary edges to grow the radius of the cluster by $w_{\min}$. Specifically, each edge weight is updated to $w \mapsto w - w_{\min}$.

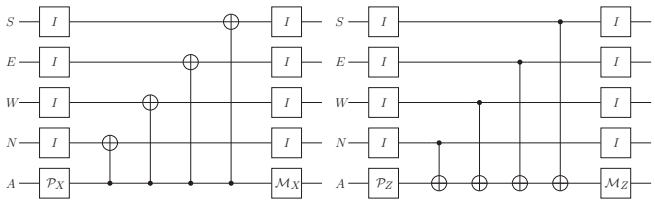


FIG. 2. Six-step syndrome extraction on the toric code for X (left) and Z (right) stabilizers. Each ancilla qubit interacts with the data qubit to its north, west, east, and south, in that order.

Figure 1 illustrates this growth step on a weighted graph [44]. We additionally find a minimum-weight spanning tree during peeling, which remains $O(n\alpha(n))$ time when presorting the edges by weight. However, this only discernibly improves the unweighted UF implementation [32].

Unfortunately, the inclusion of weighted edges has the potential to increase the runtime of the decoder. In the unweighted UF algorithm, each edge can participate in a growth step at most twice. Consequently, for a bounded degree decoder graph, the total complexity of growing the clusters is $O(n)$. More generally, given edges with real weights $\{w_i\}$ that have a common measure m, we can be assured that each edge with weight w participates in a growth step at most w/m times. However, as $\{w_i\}$ will almost surely have no common measure, we are left with a worst-case upper bound of $O(n^2)$: during each growth step, we iterate through a list of boundary edges of size $O(n)$, and in each iteration we remove at least one edge. Fortunately, this can be remedied by truncating the w_i to some finite precision ε, ensuring a common measure while incurring a negligible loss in accuracy. The corresponding weighted UF decoder then has time complexity $O(n\alpha(n) + n/\varepsilon)$, and in the parameter regimes we tested, runs nearly as quickly as the original.

III. NUMERICAL SIMULATIONS

In this work, we use a standard depolarizing error model parametrized by a single error parameter p (used, e.g., in Refs. [45,46]). Our circuits consist of four fundamental noisy gate operations.

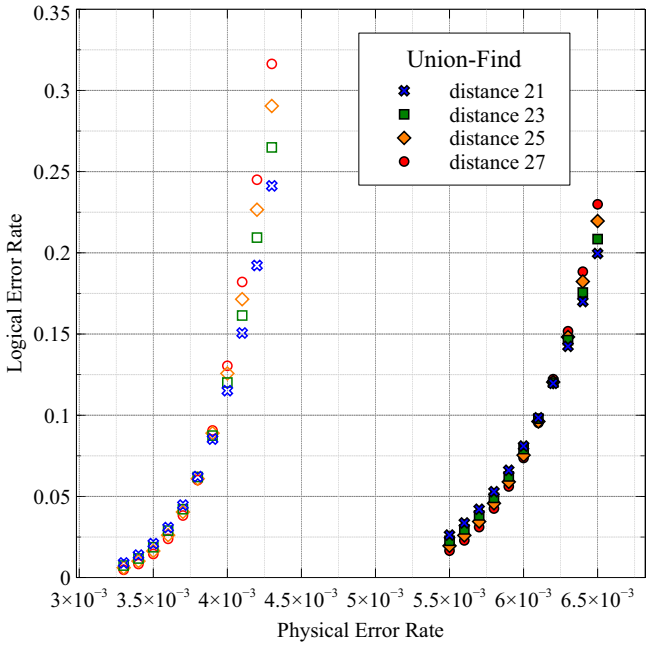
(i) With probability p, each idling step (identity gate) is followed by a Pauli error drawn uniformly at random from the set $\{X, Y, Z\}$.

(ii) With probability p, each two-qubit CNOT gate is followed by a Pauli error drawn uniformly at random from the set $\{I, X, Y, Z\}^{\otimes 2} \backslash (I \otimes I)$.

(iii) With probability $2p/3$, intended preparation of $|0\rangle$ or $|+\rangle$ wrongly prepares $|1\rangle$ or $|-\rangle$, respectively.

(iv) With probability $2p/3$, a measurement outcome in either the Z or X basis is flipped.

Syndrome extraction for the toric code proceeds in six steps: one preparation step, four two-qubit gates, and a measurement step as shown in Fig. 2. The decoder graph is formed by connecting all space-time sites that can be jointly excited by a single circuit fault. Each of these edges is then weighted by $\ln((1-p)/p)$, where p is the sum of the probabilities of those single faults occurring [1,29].



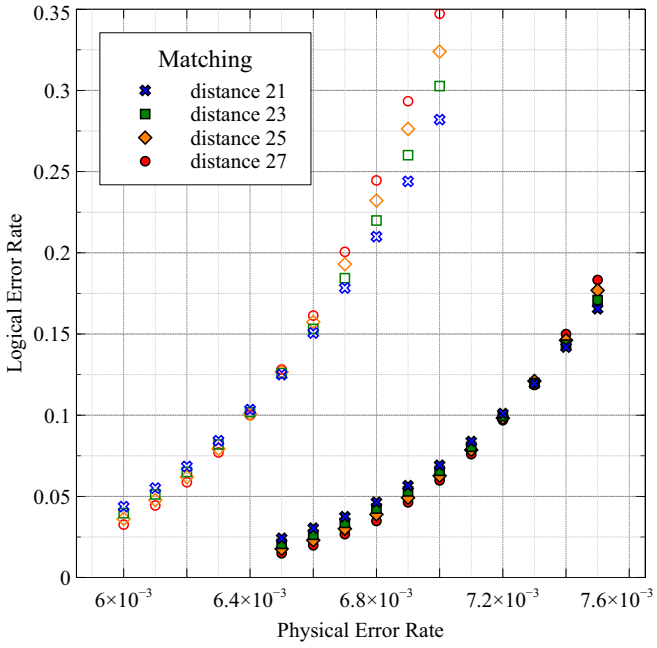
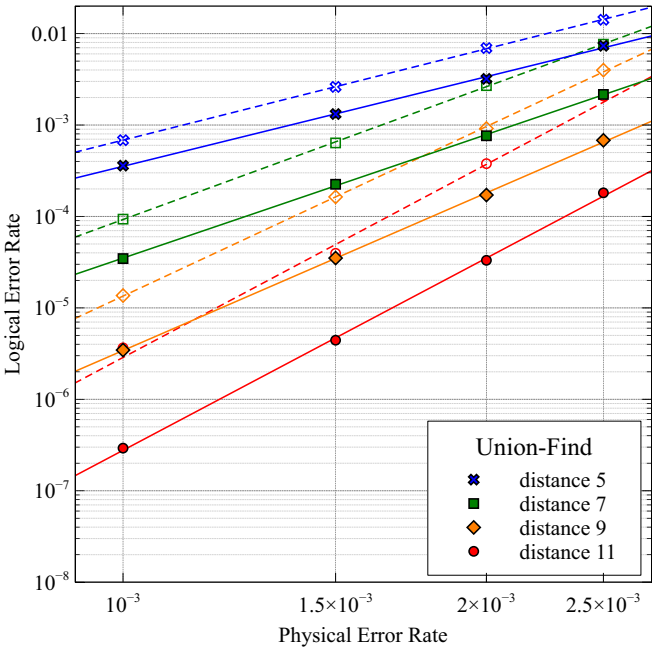


FIG. 3. Threshold behavior for unweighted (empty) and weighted (filled) decoders using both UF and MWPM. Weighting the UF decoder increases the threshold from ≈0.38% to ≈0.62%, compared with an increase from ≈0.65% to ≈0.72% for MWPM. Each point was obtained from 10^6 trials. Error bars lie within points, and are omitted throughout.



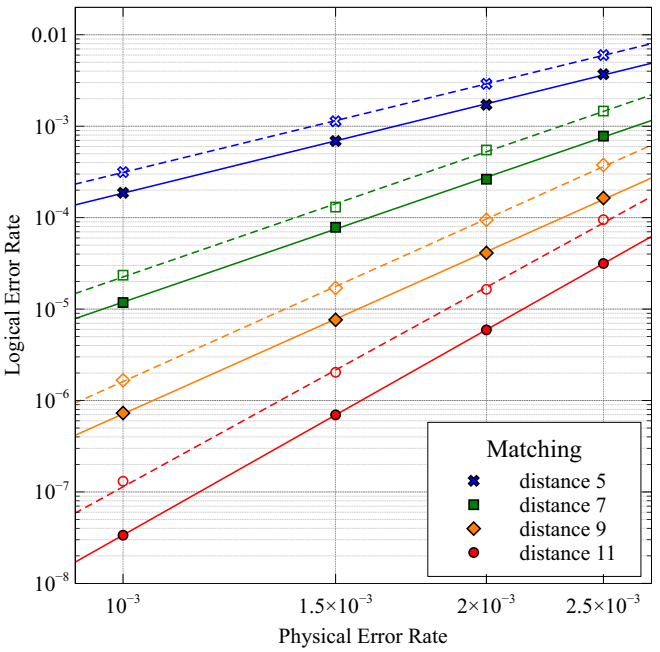


FIG. 4. Low error rate behavior for unweighted (empty, dotted) and weighted (filled, solid) decoders. Each point was obtained from at least 10^6 trials and 10^3 failures.

We analyze five different decoding strategies. We consider MWPM and UF on both weighted and unweighted decoder graphs, as well as UF on a decoder graph with weights truncated to the nearest tenth. MWPM accuracy has been characterized in a number of works [4,11,12,29,47]; however, as performance depends closely on the microscopic details of the gate and error model, we include it for the sake of direct comparison. Note that the unweighted decoder graph is not equivalent to a phenomenological decoder graph due to the inclusion of diagonal single circuit-fault edges.

For MWPM, there have been a number of runtime optimizations [11,12,19]. Here, we use a simple localized strategy inspired by Ref. [19] that forms a box around each excitation with dimensions determined by the nearest excitations in the six cardinal directions. Then, we only check for matchings in which each excitation is matched with another inside its corresponding box. This simple heuristic speeds up sequential matching, and has performance consistent with previous benchmarks [48]. We use Blossom V to perform the

TABLE I. A summary of the accuracy performance of each decoder. We approximate the logical performance scaling with d at fixed error rate p as $\propto \Lambda_p^{(d+1)/2}$. Here, Λ_p is estimated by averaging over the three intervals from $d = 5$ to $d = 11$.

	p_{thr}	$\Lambda_{.10\%}$	$\Lambda_{.15\%}$	$\Lambda_{.20\%}$	$\Lambda_{.25\%}$
Unweighted UF	0.38%	0.184	0.247	0.380	0.528
Truncated UF	0.61%	0.096	0.156	0.231	0.292
Weighted UF	0.62%	0.094	0.151	0.219	0.292
Unweighted MWPM	0.65%	0.075	0.122	0.178	0.251
Weighted MWPM	0.72%	0.057	0.101	0.151	0.204

matching itself, although additional customization to matching can considerably improve performance [3,49,50].

We analyze these decoders in three areas: threshold behavior, low error rate behavior, and (serial) efficiency. The accuracy of the truncated decoder is omitted, as it is indistinguishable from the weighted decoder in the tested regime. Each trial was decided by performing d rounds of faulty syndrome extraction followed by a terminal perfect round of syndrome extraction. If any nontrivial logical operator was applied to the two encoded qubits, the trial was declared a failure; otherwise, it was declared a success.

Figure 3 shows the relative gain in the threshold behavior of weighted versus unweighted decoding. Weighting the decoder graph significantly improves UF decoding with respect to a less dramatic increase in MWPM. The threshold for the truncated UF decoder (not shown) approximately matches that of the weighted UF decoder, with a value of ≈0.61%. Note that truncation can cause small discontinuities in the logical error rate where the weights jump in value.

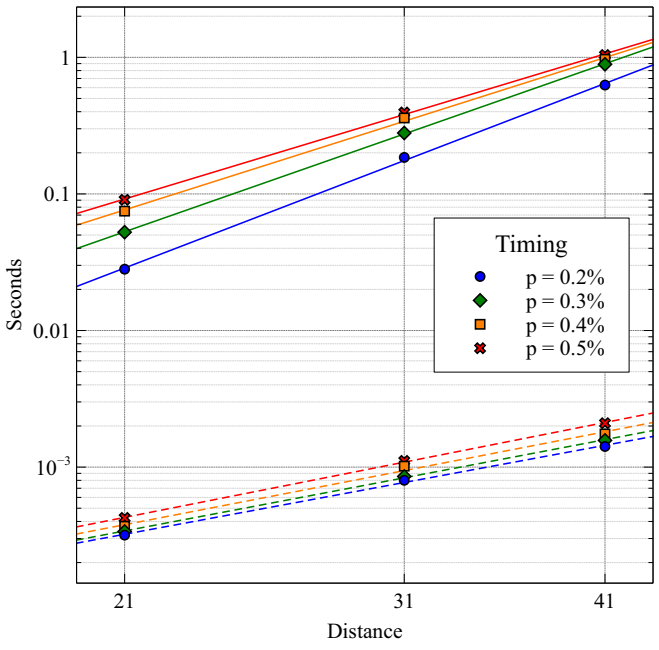


FIG. 5. Timing for weighted MWPM and UF decoders. Times are reported per extraction cycle, obtained from timing the off-line decoders on a $\propto d \times d \times d$ decoding instance and dividing over the d cycles. Weighted UF (dotted) scales as $\propto d^{2.2}$, which is nearly linear in $n \propto d^2$. In comparison, our variant of localized matching (solid) empirically runs in time $\propto d^{4.5}$, although much faster customized implementations are known [12]. Each point is the average of at least 10^3 trials on a single 2.9 GHz Intel Core i9 CPU.

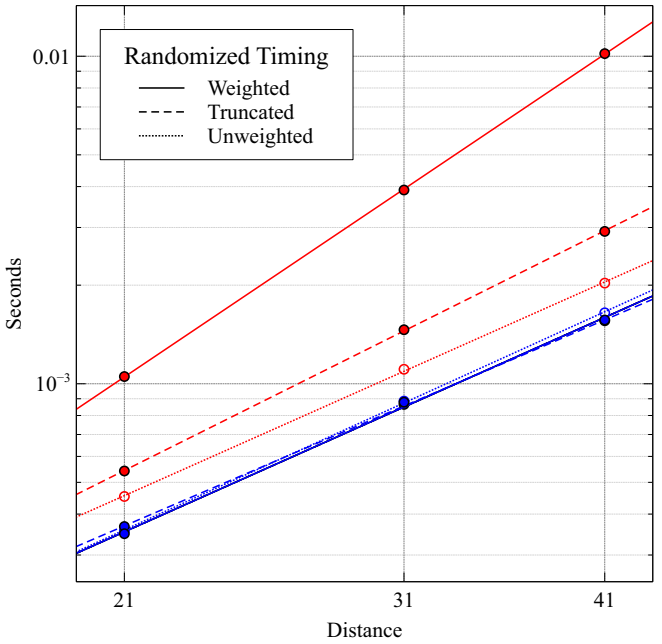


FIG. 6. Timing on a decoder graph with edge weights $\ln((1-w)/w)$, for w drawn uniformly at random from the range 0.1%–0.5%. At high physical error rates $p = 0.5\%$ (red, top three lines), weighted UF suffers a significant slowdown, but truncating the weights recoups most of the unweighted efficiency. At sufficiently low error rates $p = 0.2\%$ (blue, bottom three lines), the weighted UF slowdown does not occur.

We reiterate that these threshold values depend heavily on the specifics of the noise model and operations. For example, if one assumes quantum nondemolition measurements, the threshold can increase to as high as ≈0.90% in the case of MWPM [11]. In such a model, the weighted UF threshold increases to ≈0.76%. However, this model favorably assumes that measurements both report the wrong outcome and project into the wrong eigenstate upon failure, whereas stand-alone declaration errors can be more damaging. Thus, it is important to consider the details of the noise model when comparing different absolute threshold estimates.

The low error rate behavior in Fig. 4 mirrors that of the threshold behavior (see also Table I). We observe that weighting UF significantly increases its fault-tolerance performance, remaining competitive with matching despite its comparative simplicity and efficiency.

Unsurprisingly, sequential UF runs significantly faster than sequential MWPM even when using localizing heuristics. Figures 5 and 6 show timings in the case of translation-invariant and non-translation-invariant edge weights. The slowdown to weighted UF is exacerbated by larger clusters at higher error rates when in the presence of more edge weights. Practically, one would expect to use a variety of weights tuned according to benchmarks on individual gates, and so this slowdown (from $\propto d^{2.2}$ to $\propto d^{3.4}$) is significant.

Fortunately, simply truncating the edge weights approximately preserves the scaling and performance of weighted UF. Note also that if the error rate is sufficiently low across the entire lattice, then the slowdown does not occur. This is likely due to a smaller number of boundary edge weights in any one cluster.

Of course, one should take these off-line sequential timings with a grain of salt. MWPM has enjoyed several refinements that have been empirically shown to reduce the runtime to average linear time at sufficiently low error rates, and in principle to parallelized average $O(1)$ time [11,12,19]. In addition, recent work has demonstrated microarchitectures and accelerations that allow for UF decoding in the μs regime per extraction cycle [51,52], and extending to a weighted graph could likely be accommodated. While absolutely comparing runtimes is difficult, we expect that the speed of the UF decoder should ultimately outstrip matching due its local flavor and simplicity.

IV. CONCLUSIONS

In this paper, we benchmarked a weighted variant of the UF decoder in the full fault-tolerance setting, and demonstrated that it performs comparably to matching while preserving the almost-linear runtime of the original. Although there can be some slowdown, this can be remedied by truncating the edge weights without a significant loss in accuracy.

Compared to the difficult task of building reliable quantum components, one would ideally use decoders that optimize performance, shifting the burden from a quantum problem to a classical one. However, depending on the size and details of the decoding problem, and given the simplicity, efficiency, and relatively high performance of weighted UF, it might prove a promising avenue towards practical decoding of the surface code.

ACKNOWLEDGMENTS

The authors thank Andrew Cross and Martin Suchara for providing their code for matching, with permission from IBM. They additionally thank Christopher Chamberland and Nicolas Delfosse for helpful comments. The computational resources for simulations were provided by the Duke Computing Cluster. This research was supported in part by the ODNI/IARPA LogiQ program through ARO Grant No. (W911NF-16-1-0082), ARO MURI (Grants No. W911NF-16-1-0349 and No. W911NF-18-1-0218), and EPiQC - an NSF Expedition in Computing (1832377).

[1] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[2] A. Y. Kitaev, Ann. Phys. (NY) **303**, 2 (2003).
[3] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).
[4] R. Raussendorf and J. Harrington, Phys. Rev. Lett. **98**, 190504 (2007).
[5] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A **80**, 052312 (2009).
[6] B. J. Brown, K. Laubscher, M. S. Kesselring, and J. R. Wootton, Phys. Rev. X **7**, 021029 (2017).
[7] C. Gidney and A. G. Fowler, Quantum **3**, 135 (2019).
[8] D. Litinski, Quantum **3**, 128 (2019).
[9] G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. **104**, 050504 (2010).
[10] A. G. Fowler, A. C. Whiteside, A. L. McInnes, and A. Rabbani, Phys. Rev. X **2**, 041003 (2012).
[11] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Phys. Rev. Lett. **108**, 180501 (2012).
[12] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Phys. Rev. A **86**, 042313 (2012).
[13] M. Suchara, A. W. Cross, and J. M. Gambetta, in *Information Theory (ISIT), 2015 IEEE International Symposium on* (IEEE, Piscataway, 2015), pp. 1119–1123.
[14] J. R. Wootton and D. Loss, Phys. Rev. Lett. **109**, 160503 (2012).
[15] G. Duclos-Cianci and D. Poulin, Quantum Inf. Comput. **14**, 721 (2014).
[16] A. Hutter, J. R. Wootton, and D. Loss, Phys. Rev. A **89**, 022326 (2014).
[17] S. Bravyi, M. Suchara, and A. Vargo, Phys. Rev. A **90**, 032326 (2014).
[18] J. Wootton, Entropy **17**, 1946 (2015).
[19] A. G. Fowler, Quantum Inf. Comput. **15**, 145 (2015).
[20] F. H. E. Watson, H. Anwar, and D. E. Browne, Phys. Rev. A **92**, 032309 (2015).
[21] S. Varsamopoulos, B. Criger, and K. Bertels, Quant. Sci. Technol. **3**, 015004 (2017).
[22] M. Herold, M. J. Kastoryano, E. T. Campbell, and J. Eisert, New J. Phys. **19**, 063012 (2017).
[23] A. Kubica and J. Preskill, Phys. Rev. Lett. **123**, 020501 (2019).
[24] G. Torlai and R. G. Melko, Phys. Rev. Lett. **119**, 030501 (2017).
[25] D. K. Tuckett, A. S. Darmawan, C. T. Chubb, S. Bravyi, S. D. Bartlett, and S. T. Flammia, Phys. Rev. X **9**, 041031 (2019).
[26] D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, Phys. Rev. Lett. **124**, 130501 (2020).
[27] A. J. Landahl, J. T. Anderson, and P. R. Rice, arXiv:1108.5738.
[28] D. S. Wang, A. G. Fowler, C. D. Hill, and L. C. L. Hollenberg, Quantum Inf. Comput. **10**, 780 (2010).
[29] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Phys. Rev. A **83**, 020302(R) (2011).
[30] H. Bombin, G. Duclos-Cianci, and D. Poulin, New J. Phys. **14**, 073048 (2012).
[31] C. Chamberland and P. Ronagh, Quant. Sci. Technol. **3**, 044002 (2018).
[32] M. Li, D. Miller, M. Newman, Y. Wu, and K. R. Brown, Phys. Rev. X **9**, 021041 (2019).
[33] N. Maskara, A. Kubica, and T. Jochym-O'Connor, Phys. Rev. A **99**, 052351 (2019).
[34] N. H. Nickerson and B. J. Brown, Quantum **3**, 131 (2019).
[35] N. Delfosse and N. H. Nickerson, arXiv:1709.06218.
[36] S. Huang and K. R. Brown, Phys. Rev. A **101**, 042312 (2020).
[37] J. Edmonds, *Classic Papers in Combinatorics* (Springer, Berlin, 2009), pp. 361–379.
[38] N. Delfosse and G. Zémor, Phys. Rev. Research **2**, 033042 (2020).
[39] N. Nickerson and H. Bombín, arXiv:1810.09621.

[40] M. Newman, L. A. de Castro, and K. R. Brown, Quantum **4**, 295 (2020).
[41] R. Raussendorf, J. Harrington, and K. Goyal, New J. Phys. **9**, 199 (2007).
[42] C. Wang, J. Harrington, and J. Preskill, Ann. Phys. (NY) **303**, 31 (2003).
[43] R. E. Tarjan, J. ACM **22**, 215 (1975).
[44] In practice, unerased edges with both endpoints in the cluster are simply removed, as they cannot change the excitation parity of the cluster.
[45] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Phys. Rev. X **10**, 011022 (2020).
[46] C. Chamberland, A. Kubica, T. Yoder, and G. Zhu, New J. Phys. **22**, 023019 (2020).
[47] D. Wang, A. Fowler, A. Stephens, and L. Hollenberg, Quant. Inf. Comput. **10**, 456 (2009).
[48] Over 10^6 trials tested at various sizes and error rates, this localized variant always yielded a correct minimum-weight perfect matching. Matching code provided by IBM.
[49] A. G. Fowler, arXiv:1310.0863.
[50] V. Kolmogorov, Mathematical Program. Comput. **1**, 43 (2009).
[51] P. Das, C. A. Pattison, S. Manne, D. Carmean, K. Svore, M. Qureshi, and N. Delfosse, arXiv:2001.06598.
[52] N. Delfosse, arXiv:2001.11427.