Algorithmic Fault Tolerance for Fast Quantum Computing

Hengyun Zhou,[1,2,*] Chen Zhao,[1,†] Madelyn Cain,[2] Dolev Bluvstein,[2] Casey Duckering,[1] Hong-Ye Hu,[2] Sheng-Tao Wang,[1] Aleksander Kubica,[3,4,5] and Mikhail D. Lukin[2,‡]

[1] *QuEra Computing Inc., 1284 Soldiers Field Road, Boston, MA, 02135, US*
[2] *Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA*
[3] *AWS Center for Quantum Computing, Pasadena, California 91125, USA*
[4] *California Institute of Technology, Pasadena, California 91125, USA*
[5] *Department of Applied Physics, Yale University, New Haven, Connecticut 06511, USA USA*

Fast, reliable logical operations are essential for the realization of useful quantum computers [1–3], as they are required to implement practical quantum algorithms at large scale. By redundantly encoding logical qubits into many physical qubits and using syndrome measurements to detect and subsequently correct errors, one can achieve very low logical error rates. However, for most practical quantum error correcting (QEC) codes such as the surface code, it is generally believed that due to syndrome extraction errors, multiple extraction rounds—on the order of the code distance d—are required for fault-tolerant computation [4–14]. Here, we show that contrary to this common belief, fault-tolerant logical operations can be performed with constant time overhead for a broad class of QEC codes, including the surface code with magic state inputs and feed-forward operations, to achieve "algorithmic fault tolerance". Through the combination of transversal operations [7] and novel strategies for correlated decoding [15], despite only having access to partial syndrome information, we prove that the deviation from the ideal measurement result distribution can be made exponentially small in the code distance. We supplement this proof with circuit-level simulations in a range of relevant settings, demonstrating the fault tolerance and competitive performance of our approach. Our work sheds new light on the theory of quantum fault tolerance, potentially reducing the space-time cost of practical fault-tolerant quantum computation by orders of magnitude.

Quantum computers have the potential to solve certain computational problems much faster than their classical counterparts [1, 16]. Since most known applications require quantum computers with extremely low error rates, quantum error correction (QEC) and strategies for fault-tolerant quantum computing (FTQC) are necessary. These methods encode logical quantum information into a QEC code involving many physical qubits, such that the lowest weight logical error has weight equal to the code distance d and is therefore unlikely.

Performing large-scale computation, however, comes with significant overhead [2, 16]. By performing syndrome extraction (SE), one can reveal error information and use a classical decoder to correct physical errors in software and interpret logical measurement results. However, in the presence of noisy syndrome measurements [4–7, 10], one typically requires a number of SE rounds that scales linearly in d, i.e., $\Theta(d)$ [17] (see Fig. 1(a)). This is the case, for example, for the celebrated surface code [8–10], one of the leading candidates for practical FTQC due to its simple 2D layout and competitive error thresholds. In typical compilations based on lattice surgery or braiding [11–14, 18], each logical operation requires $\Theta(d)$ SE rounds, thus incurring a space-time volume per logical operation of $\Theta(d^3)$. This reduces the logical clock speed by a factor proportional to the code distance, typically on the order of 10 –100 [14, 16]. The same considerations also apply when performing logical operations with many quantum low-density parity-check (QLDPC) codes [19, 20]. While there have been various efforts at addressing this challenge [5, 21], these alternative approaches introduce higher hardware complexity [20, 22–24] or necessitate certain properties of the underlying codes, such as the single shot QEC property [25–29], often incurring a trade-off between space and time when executing logical operations [2, 16, 30].

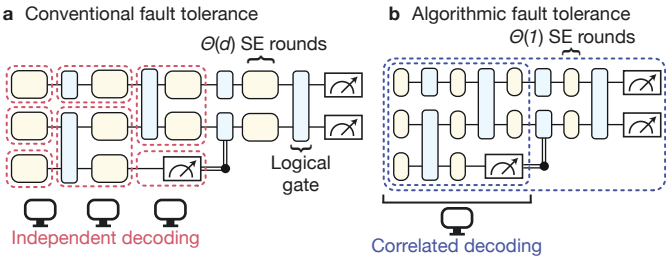


FIG. 1. **Algorithmic fault tolerance.** (a) Conventional FT analysis separately examines each gadget (red boxes) in the circuit and ensures they are individually FT [4, 7, 31]. This requires $\Theta(d)$ syndrome extraction (SE) rounds to achieve FT. (b) Algorithmic FT directly uses all accessible syndrome information up to a logical measurement (blue box), and guarantees FT of the measurement result, even if the gadgets are not individually FT and if future syndrome information is not yet accessible (partial decoding). We realize algorithmic FT through transversal operations, and only require a single SE round per logical operation, thus allowing constant time implementations of logical operations.

We introduce and develop a novel approach to FTQC that we refer to as "algorithmic fault tolerance", and show that it can lead to a substantial reduction in space-time cost. We focus on transversal implementations of Clifford circuits [7, 32] with magic state inputs and feed-

forward [33], thereby allowing universal quantum computation. Such transversal gate capabilities have already been demonstrated in multiple hardware platforms, such as neutral atoms and trapped ions [34–36]. We show that contrary to the common belief, for any Calderbank-Shor-Steane (CSS) QLDPC code [6, 37], these operations can be performed fault-tolerantly with only constant time overhead per operation, provided that decoding can be implemented efficiently. The key idea is to consider the fault tolerance of the algorithm as a whole (Fig. 1(b)) [38–40]. We achieve this by performing correlated decoding [15, 30, 36] despite only having access to partial syndrome information, and ensuring consistency in the presence of magic states and feed-forward via additional operations in software. We verify such algorithmic fault tolerance through a combination of proofs and circuit-level numerical simulations of our protocol, including a simulation of state distillation factories [13, 33], finding very little change to physical error thresholds. Specializing to the surface code, our results reduce the per-operation time cost from $\Theta(d)$ to $\Theta(1)$, including for Clifford operations used in magic state distillation. Note that unlike methods that trade space for time, our techniques represent a direct reduction in space-time volume, which is usually the ultimate quantity of interest.

ALGORITHMIC FAULT TOLERANCE VIA TRANSVERSAL OPERATIONS

We focus on transversal Clifford circuits with magic state inputs, where Clifford operations are implemented with a depth-one quantum circuit (Methods). This is interleaved with SE rounds using ancilla qubits, which reveal error information on the data qubits and enable error correction. In addition to transversal gates [7], we refer to preparation of data qubits in $|0\rangle$ followed by one SE round as transversal state preparation, and Z basis measurement of all data qubits as transversal measurement. To achieve universality, we allow teleporting in low-noise magic states with feed-forward operations based on past measurement results, and use the same Clifford operations above to prepare high quality magic states via magic state distillation [33]. We make use of CSS QLDPC codes, where each data or ancilla qubit interacts with a constant number of other qubits, and each stabilizer generator consists of all X or all Z operators [6, 37]. Within this setting, our key result can be formulated as the following theorem:

Theorem 1 (informal): Exponential error suppression for *constant time* transversal Clifford operations with any CSS QLDPC code. *For a transversal Clifford circuit with low-noise magic state inputs and feed-forward operations, that can be implemented with a given CSS QLDPC code family $\mathcal{Q}_d$ of growing code distance d, there exists a threshold p_{th}, such that if the physical error rate $p < p_{\text{th}}$ under the basic model of fault tolerance [5], then our protocol can perform constant time logical operations, with only a single SE round per operation, while suppressing the total logical error rate as $P_L = \exp(-\Theta(d_n))$.*

The formal theorem statement and the corresponding proof can be found in Supplementary Materials [41]. Our analysis assumes the basic model of fault tolerance [5]. In particular, we consider the local stochastic noise model, where we apply depolarizing errors on each data qubit every SE round and measurement errors on each SE result, with a probability that decays exponentially in the weight of the error event. This can be readily generalized to circuit-level noise by noting the bounded error propagation for constant depth SE circuits in QLDPC codes. We also assume the most likely error (MLE) decoder and fast classical computation (Methods). Finally, we assume that all code patches are identical, and the number of qubit locations within a code patch that any given qubit can be coupled to via transversal gates is bounded by some constant t, in order to control error propagation.

A key observation is that by considering the algorithm as a whole and leveraging the deterministic propagation of errors through transversal Clifford circuits, one can use the surrounding syndrome history to correct for noisy measurements (Fig. 1(b)). This correlated decoding technique has been shown to enable $\Theta(1)$ SE rounds for Clifford circuits without feed-forward [15]. However, a key component of many schemes for achieving universality is magic state teleportation, which crucially relies on the ability to realize feed-forward operations.

As illustrated by the example shown in Fig. 2(a), such feed-forward operations require on-the-fly interpretation of logical measurements, followed by a subsequent conditional gate, when only a subset of the logical qubits have been measured. As we do not yet have future syndrome information on the unmeasured logical qubits, one may be concerned that this can lead to an incorrect assignment of logical measurement results. Indeed, prior work analyzing circuits with magic states assumed that at least d SE rounds separated state initialization and measurements or out-going qubits [30, 42, 43]. As shown in Fig. 2(b) for the $\Theta(1)$ SE round case, with new syndrome information, one may end up concluding a different measurement result, which leads to an incorrect feed-forward operation.

Surprisingly, we find that these inconsistencies can be accounted for in classical processing, with a reinterpretation of subsequent measurement results (Fig. 2(c), Pauli frame updates). The inconsistent measurement result corresponds to an $\overline{X}$ operator applied right before the $\overline{Z}$ measurement. Tracing back, we can find an $\overline{X}$ operator on the $|\overline{+}\rangle$ initial state (Fig. 2(c)) which does not change the logical state but propagates through to apply $\overline{X}$ on the logical measurement, together with some other logi-

cal Pauli updates on the remaining logical qubits. These are stabilizers of the logical state, which leave the state invariant. Indeed, the fact that this measurement result can be affected by non-fault-tolerant state preparation implies that the measurement anti-commutes with the corresponding Pauli stabilizer, necessarily leading to a 50/50 random outcome that is not changed by a logical flip. Products of individual measurements can have nontrivial correlations only if they commute with all the Pauli stabilizers. Because they commute, however, they are also guaranteed to be insensitive to the state initialization errors.

Therefore, in the second step of our decoding procedure, we apply such Pauli operators on initial input states until the measurement results are consistent with the previous commitments (Fig. 2(c)). Beyond this specific circuit, the required pattern that leads to a consistent assignment can always be computed efficiently by solving a linear system of equations (Methods). In practice, subsets of measurements in which all measurement products are 50/50 random can be classically assigned in advance, with the future measurements determined through the above procedure to ensure consistency. This also implies that decoding of certain measurements can be delayed until joint products need to be determined, and some assignments can be performed deterministically in specific cases such as state distillation (Methods).

Our protocol that leads to Theorem 1 thus consists of two main steps: correlated decoding based on partial syndrome information, and application of logical stabilizers to guarantee consistency between multiple decoding rounds (Fig. 2).

We now sketch the intuition behind our proof of Theorem 1. There are two types of logical errors that may occur with our protocol. The first, a heralded inconsistency error, occurs when we are not able to find a set of operators to apply that yield the same outcome as previously committed measurement results. The second, a regular error, occurs when an erroneous logical operator is applied that results in a different measurement distribution.

Because imperfect readout during transversal measurements are equivalent to data qubit errors followed by perfect measurements, transversal measurements produce reliable syndrome information. Intuitively, this prevents individual errors from leading to high-weight corrections on the logical qubits we measure, the main reason for needing d SE rounds in typical FT state initialization protocols. At the same time, the use of correlated decoding, together with the structured error propagation through transversal Clifford gates, allow us to propagate this syndrome information and correct relevant errors happening throughout the circuit. With these observations, we prove that for either type of logical error to occur, the total Pauli weight s of physical error and subsequent correction in a connected cluster must satisfy $s = \Theta(d)$, which

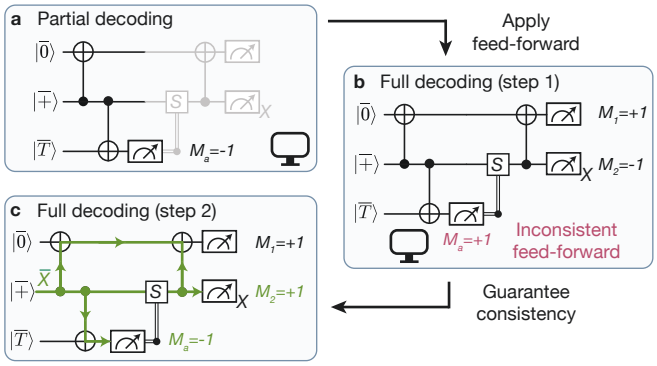


FIG. 2. **Illustration of decoding strategy.** (a) Logical quantum circuit with measurement and feed-forward. All logical operations are transversal and interleaved with a single SE round, instead of d SE rounds. We must decode and commit mid-circuit to a measurement result for the bottom qubit, despite lacking complete syndrome information on the top two qubits (partial decoding). (b) With the measurement result of the bottom qubit, a feed-forward operation is applied, the remaining circuit is executed, and decoding is performed again on the whole circuit. The second decoding round may assign a different result to the bottom qubit, causing an inconsistency in feed-forward operations. (c) To guarantee consistency, we apply an $\overline{X}$ operator on the $|\overline{+}\rangle$ initial state of the middle qubit, which acts trivially on $|\overline{+}\rangle$, but changes the interpreted logical measurement result M_a to be consistent with before. This also leads to a re-interpretation of the logical measurement result M_2.

has probability $p^{s/2}$ under the MLE decoder. Finally, we count the number of such connected clusters of size s, which scales as $(v\mathfrak{e})^s$, where $\mathfrak{e}$ is the natural base and v is a constant upper-bounding the error connectivity for a QLDPC code. The combined probability of an error thus scales as

$$P_{\text{err}} \propto p^{s/2}(2v\mathfrak{e})^s = (2v\mathfrak{e}\sqrt{p})^{\Theta(d)} \to 0 \qquad (1)$$

when the physical error rate is sufficiently low $p \ll 1/(v\mathfrak{e})^2$ (the factor of 2 comes from a combinatorial sum), thereby establishing the existence of a threshold and exponential error suppression.

Specializing to the surface code and utilizing the full transversal Clifford gate set accessible to the surface code (Methods), an immediate corollary of our main theorem is a threshold result for performing constant time logical operations with an *arbitrary* transversal Clifford circuit. This result supports universal quantum computing when we allow magic state inputs prepared with sufficiently low noise.

Preparing high quality magic state inputs, in turn, can be performed simply with the same Clifford operations and easy-to-prepare non-fault-tolerant magic states [44–46], a procedure known as magic state distillation [33] (see ED Fig. 3). We expect that the same algorithmic FT approach described above achieves a $\Theta(d)$ speed-up in distillation time as well. The distillation factory and

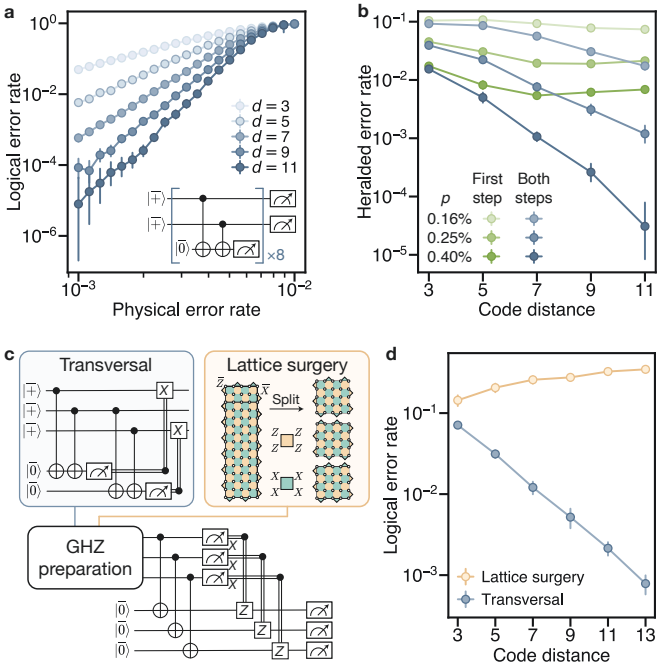


FIG. 3. **Numerical verification of fault tolerance.** (a) Simulation of circuit with repeated $\overline{ZZ}$ measurement (inset), where we commit mid-circuit to each measurement result of the logical ancilla using only the syndrome information up to that point. The total logical error rate as a function of circuit-level physical error rate p, for varying code distance d, shows clear threshold behavior. (b) Heralded error rate with and without the second step of our decoding strategy, as a function of code distance and for different physical error rates, for the same circuit as (a). Only with both steps do we observe exponential suppression of the logical error rate. (c) Comparison of two different methods for logical state preparation between three rotated surface codes and subsequent teleportation, for fixed circuit noise $p = 0.3\%$. We use transversal gates (left) and lattice surgery (right), in both cases with only a single SE round. (d) With transversal gates, the error rate decreases exponentially with the code distance. With a single round of lattice surgery, the error rate instead increases linearly with code distance, as a single stabilizer measurement error affects the logical $\overline{ZZ}$ measurement result.

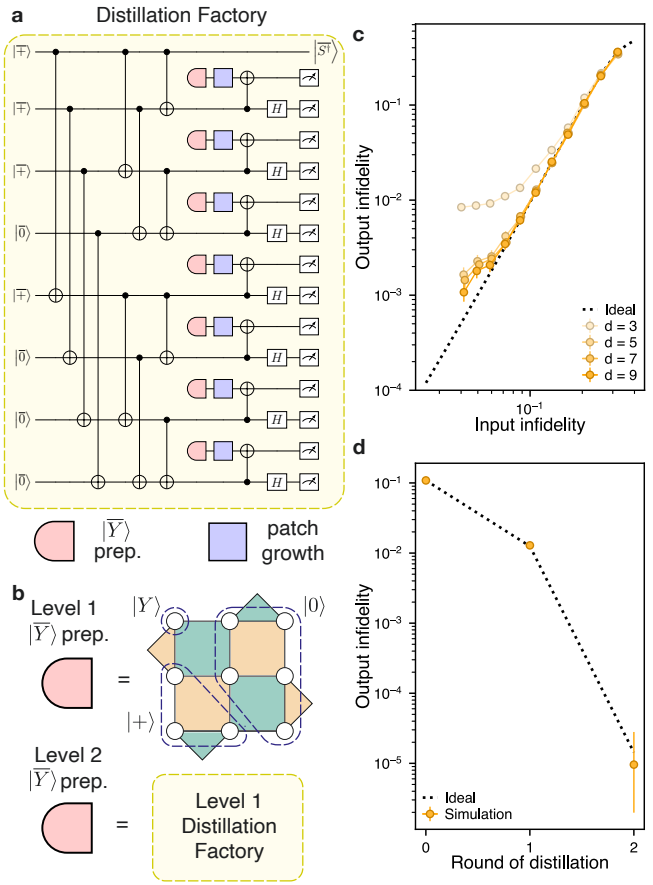


FIG. 4. $|\overline{Y}\rangle$ **state distillation factory.** (a) Illustration of a $|\overline{Y}\rangle$ state distillation factory based on the [[7,1,3]] Steane code, consisting of state initialization, layers of transversal CNOTs, followed by a teleported $\overline{S}$ gate. Each operation involves only a single SE round. Two of the CNOTs in the first layer act trivially and can be omitted. (b) The $|\overline{Y}\rangle$ resource state is prepared via state injection at the first level, and via the first-level factory for the second level. (c) 1-level factory output state infidelity as a function of input state infidelity, for fixed circuit noise $p = 0.1\%$ and varying levels of artificially injected Z errors. The ideal curve is calculated assuming the gate operations in the factory are perfect. (d) Performance for one and two rounds of distillation, showing good agreement with the expected scaling.

main computation can then be combined by applying our decoding approach to the joint system. In Methods and Supplementary Information, we further describe an extension of our results to the case of single-shot code patch growth, relevant to practical distillation factories [47, 48]. Taken together, these results provide a theoretical foundation for our factor of $\Theta(d)$ improvement in logical clock speed compared to standard FT approaches for universal quantum computation.

COMPETITIVE NUMERICAL PERFORMANCE

We now turn to circuit-level simulations of our protocol to numerically evaluate its performance [39], and contrast it with existing methods. We consider various test cases of our approach that also serve as key subroutines in large-scale algorithms.

We first consider a simple circuit with intermediate logical measurements (inset of Fig. 3(a)). In this example, two logical qubits are transversally initialized in $|\overline{+}\rangle$, and an ancilla logical qubit is used to measure the $\overline{ZZ}$ correlation a total of eight times, before the two logical qubits are transversally measured in the $\overline{Z}$ basis. While individual logical measurement results are random, a correct realization of this circuit should yield the same result for $\overline{ZZ}$ each time, which in turn should be consistent with the final logical measurement results. We employ our al-

gorithmic FT protocol to decode the circuit up to each logical measurement using only the syndrome information accessible at that point. We use the rotated surface code, a circuit-level depolarizing noise model [15, 49], a MLE decoder based on integer programming [15, 50], and employ the two-step process described above (see Supplementary Information).

Figure 3(a-b) show the results of numerical simulations. We find that the total logical error rate, defined as the probability that a logical error of either type mentioned above happened anywhere in the circuit, shows characteristic threshold behavior, with an estimated threshold $\gtrsim 0.85\%$. As an SE round involves four layers of CNOT gates, while the transversal CNOT only involves a single layer, the effective error rate is dominated by SE operations, hence it may be expected that the threshold is close to the circuit-noise memory threshold. The number of SE rounds can be further optimized: for example, in Ref. [15], performing one SE round every four gate layers minimized the space-time cost per CNOT, suggesting that the practical improvement may be $\gtrsim 2d$ in some regimes [51]. In Fig. 3(b), we further compare the scaling of heralded failure rates in the presence and absence of the second step of our decoding procedure, as a function of code distance d. We find that this additional step is crucial to achieve exponential suppression with the code distance.

We now contrast our approach with lattice surgery in a similar setting [11, 12, 18, 52]. We consider the logical circuit in Fig. 3(c), where a GHZ state preparation circuit is followed by teleportation of the GHZ state to another set of logical qubits, and then measurement in the $\overline{Z}$ basis [41]. Using transversal gates with only a single SE round during $|\overline{+}\rangle$ and $|\overline{0}\rangle$ state preparation, and decoding each logical measurement with only accessible information at that stage, we find that the logical error rate decreases exponentially with the code distance, consistent with our FT analysis. In contrast, state preparation based on a single round of lattice surgery [52], which involves performing syndrome extraction with a larger code patch and then splitting it into three individual logical qubits, does not yield improved logical error rate as the code distance increases, as a single error can lead to incorrect inference of the $\overline{ZZ}$ correlation of the GHZ state (Supplementary Information). Unlike transversal measurements, logical information here is contained in noisy stabilizer products, which require repetition to reliably infer.

Next, we simulate a state distillation factory. In order to perform a classical simulation of a full factory, we focus on distillation of the $|\overline{Y}\rangle = \overline{S}|\overline{+}\rangle$ state (Fig. 4(a)), which allows the easy implementation of $\overline{S}$ gates in the surface code. Since this circuit has a similar structure to the practically relevant $|\overline{T}\rangle$ magic state distillation factories (Methods, ED Fig. 3), we expect them to have similar performance. We fix the error rate of the circuit to $p = 0.1\%$, and vary the input infidelity P_{in} in Fig. 4(c). Examining the output $|\overline{Y}\rangle$ of a one-level factory, we find that as the code distance is increased, the output logical error rate P_{out} approaches the fidelity expected for ideal Clifford logical gates in the factory $P_{\text{out}} = 7P_{\text{in}}^3 + O(P_{\text{in}}^4)$ (see Methods for the full expression), across the explored fidelity regime.

Finally, we simulate the logical error rate for a two-level $|\overline{Y}\rangle$ state distillation factory, involving a total of 113 logical qubits, where the output $|\overline{Y}\rangle$ states of a $d_1 = 5$ factory is fed into a second factory with $d_2 = 9$, with the distance chosen such that the logical error is dominated by the input state infidelity. As shown in Fig. 4(d), the logical error rates at each level of the distillation procedure are consistent with that expected based on the ideal factory formula (Methods), confirming that our approach is FT. Since the state injection procedure is agnostic to the particular state that is injected, we expect that our results will readily generalize to the setting of $|\overline{T}\rangle$ magic state factories.

DISCUSSION AND OUTLOOK

Transversal operations and correlated decoding were recently found to be highly effective in experiments with reconfigurable neutral atom arrays [36]. The principles of algorithmic fault-tolerance described here are the core underlying mechanisms of these observations, such as correlated decoding of a logical Bell state [36], and our results here indicate that the same techniques allow for $\Theta(d)$ time reduction for universal computation. While recent work has provided strong evidence that this reduction might be possible for circuits consisting purely of Clifford gates and Pauli basis inputs [15], up to now it has generally been believed that this conclusion does not hold when performing universal quantum computation [30, 42, 43], which crucially relies on the use of magic states and feed-forward operations. The present work not only demonstrates that this $\Theta(d)$ time cost reduction is broadly applicable to universal quantum computing, but also provides a theoretical foundation for it through mathematical fault tolerance proofs.

Although our analysis focused on the use of an MLE decoder, our numerical simulations suggest that algorithms with polynomial runtime can still achieve a competitive threshold [41], and the development of improved, parallel correlated decoders is an important area of future research (Methods). Taking into account the decoding time overhead, we may eventually need to insert more SE rounds to simplify decoding or wait for decoding completion [53], as is also needed for FT protocols that rely on single-shot quantum error correction [25]. In that case, we still expect a significant practical saving over existing schemes. In light of recent experimental advances [36], a full compilation and evaluation of the space-time

savings in parallel reconfigurable architectures such as neutral atom arrays is an important next step. Finally, it will be interesting to investigate how these results can be combined with recent progress toward constant-space-overhead quantum computation [5, 20, 23, 27, 29, 54–56] or generalized to transversal non-Clifford gates [30, 57–62], in order to further reduce the space-time volume of large-scale quantum computation.

AUTHOR CONTRIBUTIONS

H.Z. formulated the decoding strategy and developed an initial proof sketch through discussions with C.Z., M.C., D.B., C.D., S.-T.W., A.K., and M.D.L.. C.Z., M.C., H.Z., and H.H. performed numerical simulations. H.Z., A.K., C.Z., M.C., and C.D. proved the fault tolerance of the scheme. All authors contributed to writing the manuscript.

ACKNOWLEDGEMENTS

We acknowledge helpful discussions with G. Baranes, P. Bonilla, E. Campbell, S. Evered, S. Geim, L. Jiang, M. Kalinowski, A. Krishna, S. Li, D. Litinski, T. Manovitz, N. Maskara, Y. Wu, and Q. Xu. We would particularly like to thank C. Pattison for early discussions and suggesting the simulation of the $|\overline{Y}\rangle$ state distillation factory, and J. Haah for stimulating discussions and deep insights. We acknowledge financial support from IARPA and the Army Research Office, under the Entangled Logical Qubits program (Cooperative Agreement Number W911NF-23-2-0219), the DARPA ONISQ program (W911NF2010021), the DARPA IMPAQT program (HR0011-23-3-0012), the DARPA MeasQuIT program (HR0011-24-9-0359), the Center for Ultracold Atoms (a NSF Physics Frontiers Center, PHY-1734011), the National Science Foundation (grant number PHY-2012023 and grant number CCF-2313084), the Army Research Office MURI (grant number W911NF-20-1-0082), DOE/LBNL (grant number DE-AC02-05CH11231), the Wellcome Leap Quantum for Bio program. M.C. acknowledges support from Department of Energy Computational Science Graduate Fellowship under award number DE-SC0020347. D.B. acknowledges support from the NSF Graduate Research Fellowship Program (grant DGE1745303) and The Fannie and John Hertz Foundation. This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Data Availability: The data that supports the findings of this study are available from the corresponding authors upon reasonable request.

Code Availability: All code used for data analysis are available from the corresponding author upon reasonable request.

Competing interests: M.D.L. is a co-founder and shareholder and H.Z., C.Z., C.D., S.-T.W are employees of QuEra Computing.

Correspondence and requests for materials should be addressed to H.Z. and M.D.L.

* These authors contributed equally; hyzhou@quera.com
† These authors contributed equally
‡ lukin@physics.harvard.edu

[1] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, Quantum algorithms: A survey of applications and end-to-end complexities, arXiv preprint arXiv:2310.03011 (2023).

[2] M. E. Beverland, P. Murali, M. Troyer, K. M. Svore, T. Hoeffler, V. Kliuchnikov, G. H. Low, M. Soeken, A. Sundaram, and A. Vaschillo, Assessing requirements to scale to practical quantum advantage, arXiv preprint arXiv:2211.07629 10.48550/arxiv.2211.07629 (2022).

[3] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage, PRX Quantum **2**, 010103 (2021).

[4] D. Gottesman, An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation, arXiv preprint arXiv:0904.2557 , 13 (2009).

[5] D. Gottesman, Fault-Tolerant Quantum Computation with Constant Overhead, Quantum Information and Computation **14**, 1338 (2013).

[6] A. M. Steane, Error Correcting Codes in Quantum Theory, Physical Review Letters **77**, 793 (1996).

[7] P. W. Shor, Fault-tolerant quantum computation, arXiv preprint arXiv:quant-ph/9605011 (1996).

[8] A. Y. Kitaev, Fault-tolerant quantum computation by anyons, Annals of Physics **303**, 2 (2003).

[9] S. B. Bravyi, A. Y. Kitaev, and L. D. Landau, Quantum codes on a lattice with boundary, arXiv preprint arXiv:quant-ph/9811052 (1998).

[10] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, Journal of Mathematical Physics **43**, 4452 (2002).

[11] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, New Journal of Physics **14**, 123011 (2012).

[12] D. Litinski, A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery, Quantum **3**, 128 (2019).

[13] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Physical Review A **86**, 032324 (2012).

[14] D. Litinski and N. Nickerson, Active volume: An architecture for efficient fault-tolerant quantum comput-

ers with limited non-local connections, arXiv preprint arXiv:2211.15465 (2022).
[15] M. Cain, C. Zhao, H. Zhou, N. Meister, J. Pablo, B. Ataides, A. Jaffe, D. Bluvstein, and M. D. Lukin, Correlated decoding of logical algorithms with transversal gates, arXiv preprint arXiv:2403.03272 (2024).
[16] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, Quantum **5**, 1 (2019).
[17] The notation $g(x) = \Theta(f(x))$ indicates that two functions $f(x)$ and $g(x)$ have the same asymptotic scaling with x, or more precisely, that there exists some constants c_1 and c_2 such that $c_1 f(x) \leq g(x) \leq c_2 f(x)$ for sufficiently large x.
[18] A. G. Fowler and C. Gidney, Low overhead quantum computation using lattice surgery, arXiv preprint arXiv:1808.06709 10.48550/arxiv.1808.06709 (2018).
[19] L. Z. Cohen, I. H. Kim, S. D. Bartlett, and B. J. Brown, Low-overhead fault-tolerant quantum computing using long-range connectivity, Science Advances **8**, 10.1126/sciadv.abn1717 (2022).
[20] Q. Xu, J. P. Bonilla Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasić, M. D. Lukin, L. Jiang, and H. Zhou, Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays, Nature Physics 10.1038/s41567-024-02479-z (2024).
[21] H. Yamasaki and M. Koashi, Time-Efficient Constant-Space-Overhead Fault-Tolerant Quantum Computation, arXiv preprint arXiv:2207.08826 10.48550/arxiv.2207.08826 (2022).
[22] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, Nature **627**, 778 (2024).
[23] M. A. Tremblay, N. Delfosse, and M. E. Beverland, Constant-Overhead Quantum Error Correction with Thin Planar Connectivity, Physical Review Letters **129**, 050504 (2022).
[24] O. Higgott and N. P. Breuckmann, Constructions and performance of hyperbolic and semi-hyperbolic Floquet codes, arXiv preprint arXiv:2308.03750 (2023).
[25] H. Bombín, Single-shot fault-tolerant quantum error correction, Physical Review X **5**, 031043 (2015).
[26] E. T. Campbell, A theory of single-shot error correction for adversarial noise, Quantum Science and Technology **4**, 025006 (2019).
[27] O. Fawzi, A. Grospellier, and A. Leverrier, Constant overhead quantum fault-tolerance with quantum expander codes, Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS **2018-Octob**, 743 (2018).
[28] A. Kubica and M. Vasmer, Single-shot quantum error correction with the three-dimensional subsystem toric code, Nature Communications 2022 13:1 **13**, 1 (2022).
[29] S. Gu, E. Tang, L. Caha, S. H. Choe, Z. He, and A. Kubica, Single-shot decoding of good quantum LDPC codes, arXiv preprint arXiv:2306.12470 (2023).
[30] M. E. Beverland, A. Kubica, and K. M. Svore, Cost of Universality: A Comparative Study of the Overhead of State Distillation and Code Switching with Color Codes, PRX Quantum **2**, 020341 (2021).
[31] D. Aharonov and M. Ben-Or, Fault-Tolerant Quantum Computation With Constant Error Rate, SIAM Journal on Computing **38**, 1207 (1999).
[32] C. Wang, J. Harrington, and J. Preskill, Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory, Annals of Physics **303**, 31 (2003).
[33] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Physical Review A **71**, 022316 (2005).
[34] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, R. Blatt, P. Schindler, M. Müller, and T. Monz, Demonstration of fault-tolerant universal quantum gate operations, Nature **605**, 675 (2022).
[35] C. Ryan-Anderson, N. C. Brown, M. S. Allman, B. Arkin, G. Asa-Attuah, C. Baldwin, J. Berg, J. G. Bohnet, S. Braxton, N. Burdick, J. P. Campora, A. Chernoguzov, J. Esposito, B. Evans, D. Francois, J. P. Gaebler, T. M. Gatterman, J. Gerber, K. Gilmore, D. Gresh, A. Hall, A. Hankin, J. Hostetter, D. Lucchetti, K. Mayer, J. Myers, B. Neyenhuis, J. Santiago, J. Sedlacek, T. Skripka, A. Slattery, R. P. Stutz, J. Tait, R. Tobey, G. Vittorini, J. Walker, and D. Hayes, Implementing Fault-tolerant Entangling Gates on the Five-qubit Code and the Color Code, arXiv preprint arXiv:2208.01863 10.48550/arxiv.2208.01863 (2022).
[36] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataides, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, Logical quantum processor based on reconfigurable atom arrays, Nature **626**, 58 (2024).
[37] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, Physical Review A **54**, 1098 (1996).
[38] N. Delfosse and A. Paetznick, Spacetime codes of Clifford circuits, arXiv preprint arXiv:2304.05943 (2023).
[39] C. Gidney, Stim: a fast stabilizer circuit simulator, Quantum **5**, 10.22331/q-2021-07-06-497 (2021).
[40] D. Gottesman, Opportunities and Challenges in Fault-Tolerant Quantum Computation, arXiv preprint arXiv:2210.15844 10.48550/arxiv.2210.15844 (2022).
[41] See Supplemental Material for more details.
[42] Z. Cai, A. Siegel, and S. Benjamin, Looped Pipelines Enabling Effective 3D Qubit Lattices in a Strictly 2D Device, PRX Quantum **4**, 020345 (2023).
[43] C. Duckering, J. M. Baker, D. I. Schuster, and F. T. Chong, Virtualized Logical Qubits: A 2.5D Architecture for Error-Corrected Quantum Computing, Proceedings of the Annual International Symposium on Microarchitecture, MICRO **2020-Octob**, 173 (2020).
[44] Y. Li, A magic state's fidelity can be superior to the operations that created it, New Journal of Physics **17**, 023037 (2015).
[45] L. Lao and B. Criger, Magic state injection on the rotated surface code, ACM International Conference Proceeding Series , 113 (2022).
[46] C. Gidney, Cleaner magic states with hook injection, arXiv preprint arXiv:2302.12292 (2023).
[47] D. Litinski, Magic State Distillation: Not as Costly as You Think, Quantum **3**, 205 (2019).
[48] C. Gidney and A. G. Fowler, Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation, Quantum **3**, 135 (2019).

[49] O. Higgott and N. P. Breuckmann, Improved Single-Shot Decoding of Higher-Dimensional Hypergraph-Product Codes, PRX Quantum **4**, 020332 (2023).

[50] Gurobi Optimization, LLC, Gurobi Optimizer Reference Manual (2023).

[51] Four transversal gates and one SE round require 8 CNOT gates in total, compared to $16d + 4$ with d SE rounds.

[52] I. H. Kim, Y. H. Liu, S. Pallister, W. Pol, S. Roberts, and E. Lee, Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules, Physical Review Research **4**, 023019 (2022).

[53] B. M. Terhal, Quantum error correction for quantum memories, Reviews of Modern Physics **87**, 307 (2015).

[54] N. P. Breuckmann and J. N. Eberhardt, Quantum Low-Density Parity-Check Codes, PRX Quantum **2**, 10.1103/prxquantum.2.040101 (2021).

[55] J. P. Tillich and G. Zemor, Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength, IEEE Transactions on Information Theory **60**, 1193 (2014).

[56] P. Panteleev and G. Kalachev, Asymptotically good Quantum and locally testable classical LDPC codes, Proceedings of the Annual ACM Symposium on Theory of Computing , 375 (2022).

[57] H. Bombín, Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes, New Journal of Physics **17**, 083002 (2015).

[58] A. Kubica, B. Yoshida, and F. Pastawski, Unfolding the color code, New Journal of Physics **17**, 083026 (2015).

[59] M. Vasmer and D. E. Browne, Three-dimensional surface codes: Transversal gates and fault-tolerant architectures, Physical Review A **100**, 012312 (2019).

[60] B. J. Brown, A fault-tolerant non-Clifford gate for the surface code in two dimensions, Science Advances **6**, 4929 (2020).

[61] H. Bombin, Transversal gates and error propagation in 3D topological codes, arXiv preprint arXiv:1810.09575 (2018).

[62] G. Zhu, S. Sikander, E. Portnoy, A. W. Cross, and B. J. Brown, Non-Clifford and parallelizable fault-tolerant logical gates on constant and almost-constant rate homological quantum LDPC codes via higher symmetries, arXiv preprint arXiv:2310.16982 (2023).

[63] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010) p. 676.

[64] C. A. Pattison, A. Krishna, and J. Preskill, Hierarchical memories: Simulating quantum LDPC codes with local gates, arXiv preprint arXiv:2303.04798 10.48550/arxiv.2303.04798 (2023).

[65] S. Bravyi, D. Gosset, R. König, and M. Tomamichel, Quantum advantage with noisy shallow circuits, Nature Physics 2020 16:10 **16**, 1040 (2020).

[66] B. Eastin and E. Knill, Restrictions on Transversal Encoded Quantum Gate Sets, Physical Review Letters **102**, 110502 (2009).

[67] T. Jochym-O'Connor, A. Kubica, and T. J. Yoder, Disjointness of Stabilizer Codes and Limitations on Fault-Tolerant Logical Gates, Physical Review X **8**, 021047 (2018).

[68] J. E. Moussa, Transversal Clifford gates on folded surface codes, Physical Review A **94**, 10.1103/physreva.94.042316 (2016).

[69] N. P. Breuckmann and S. Burton, Fold-Transversal Clifford Gates for Quantum Codes, arXiv preprint arXiv:2202.06647 (2022).

[70] A. O. Quintavalle, P. Webster, and M. Vasmer, Partitioning qubits in hypergraph product codes to implement logical gates, arXiv preprint arXiv:2204.10812 (2022).

[71] H. Bombín, C. Dawson, Y.-H. Liu, N. Nickerson, F. Pastawski, and S. Roberts, Modular decoding: parallelizable real-time decoding for quantum computers, arXiv preprint arXiv:2303.04846 10.48550/arxiv.2303.04846 (2023).

[72] O. Higgott, T. C. Bohdanowicz, A. Kubica, S. T. Flammia, and E. T. Campbell, Improved Decoding of Circuit Noise and Fragile Boundaries of Tailored Surface Codes, Physical Review X **13**, 031007 (2023).

[73] A. J. Landahl, J. T. Anderson, and P. R. Rice, Fault-tolerant quantum computing with color codes, arXiv preprint arXiv:1108.5738 (2011).

[74] S. Bravyi and A. Cross, Doubled Color Codes, arXiv preprint arXiv:1509.03239 10.48550/arxiv.1509.03239 (2015).

[75] D. Bacon, S. T. Flammia, A. W. Harrow, and J. Shi, Sparse Quantum Codes from Quantum Circuits, Proceedings of the forty-seventh annual ACM symposium on Theory of Computing , 327 (2014).

[76] P. Aliferis, D. Gottesman, and J. Preskill, Accuracy threshold for postselected quantum computation, Quantum Information and Computation **8**, 181 (2007).

[77] A. A. Kovalev and L. P. Pryadko, Fault tolerance of quantum low-density parity check codes with sublinear distance scaling, Physical Review A - Atomic, Molecular, and Optical Physics **87**, 020304 (2013).

[78] S. Bravyi, G. Smith, and J. A. Smolin, Trading classical and quantum computational resources, Physical Review X **6**, 021043 (2016).

[79] M. Yoganathan, R. Jozsa, and S. Strelchuk, Quantum advantage of unitary Clifford circuits with magic state inputs, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **475**, 10.1098/rspa.2018.0427 (2018).

[80] C. Gidney, Halving the cost of quantum addition, Quantum **2**, 74 (2018).

[81] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, A new quantum ripple-carry addition circuit, arXiv preprint arXiv:quant-ph/0410184 (2004).

[82] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity, Physical Review X **8**, 10.1103/physrevx.8.041015 (2018).

[83] A. G. Fowler, Time-optimal quantum computation, arXiv preprint arXiv:1210.4626 10.48550/arxiv.1210.4626 (2012).

[84] E. Knill, Quantum Computing with Very Noisy Devices, Nature **434**, 39 (2004).

[85] C. Gidney and A. G. Fowler, Flexible layout of surface code computations using AutoCCZ states, arXiv preprint arXiv:1905.08916 10.48550/arxiv.1905.08916 (2019).

[86] S. Bravyi and J. Haah, Magic-state distillation with low overhead, Physical Review A - Atomic, Molecular, and Optical Physics **86**, 052329 (2012).

[87] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, J. Atalaya, R. Babbush, D. Bacon, J. C. Bardin, J. Basso, A. Bengtsson, S. Boixo, G. Bortoli, A. Bourassa, J. Bovaird, L. Brill, M. Broughton, B. B. Buckley, D. A. Buell, T. Burger, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, B. Chiaro, J. Cogan, R. Collins, P. Conner, W. Courtney, A. L. Crook, B. Curtin, D. M. Debroy, A. D. T. Barba, S. Demura, A. Dunsworth, D. Eppens, C. Erickson, L. Faoro, E. Farhi, R. Fatemi, L. F. Burgos, E. Forati, A. G. Fowler, B. Foxen, W. Giang, C. Gidney, D. Gilboa, M. Giustina, A. G. Dau, J. A. Gross, S. Habegger, M. C. Hamilton, M. P. Harrigan, S. D. Harrington, O. Higgott, J. Hilton, M. Hoffmann, S. Hong, T. Huang, A. Huff, W. J. Huggins, L. B. Ioffe, S. V. Isakov, J. Iveland, E. Jeffrey, Z. Jiang, C. Jones, P. Juhas, D. Kafri, K. Kechedzhi, J. Kelly, T. Khattar, M. Khezri, M. Kieferová, S. Kim, A. Kitaev, P. V. Klimov, A. R. Klots, A. N. Korotkov, F. Kostritsa, J. M. Kreikebaum, D. Landhuis, P. Laptev, K.-M. Lau, L. Laws, J. Lee, K. Lee, B. J. Lester, A. Lill, W. Liu, A. Locharla, E. Lucero, F. D. Malone, J. Marshall, O. Martin, J. R. McClean, T. Mccourt, M. McEwen, A. Megrant, B. M. Costa, X. Mi, K. C. Miao, M. Mohseni, S. Montazeri, A. Morvan, E. Mount, W. Mruczkiewicz, O. Naaman, M. Neeley, C. Neill, A. Nersisyan, H. Neven, M. Newman, J. H. Ng, A. Nguyen, M. Nguyen, M. Y. Niu, T. E. O'Brien, A. Opremcak, J. Platt, A. Petukhov, R. Potter, L. P. Pryadko, C. Quintana, P. Roushan, N. C. Rubin, N. Saei, D. Sank, K. Sankaragomathi, K. J. Satzinger, H. F. Schurkus, C. Schuster, M. J. Shearn, A. Shorter, V. Shvarts, J. Skruzny, V. Smelyanskiy, W. C. Smith, G. Sterling, D. Strain, M. Szalay, A. Torres, G. Vidal, B. Villalonga, C. V. Heidweiller, T. White, C. Xing, Z. J. Yao, P. Yeh, J. Yoo, G. Young, A. Zalcman, Y. Zhang, and N. Zhu, Suppressing quantum errors by scaling a surface code logical qubit, arXiv preprint arXiv:2207.06431 10.48550/arxiv.2207.06431 (2022).

[88] J. R. Wootton and D. Loss, High threshold error correction for the surface code, Physical Review Letters **109**, 10.1103/PhysRevLett.109.160503 (2012).

[89] A. G. Fowler, Optimal complexity correction of correlated errors in the surface code, arXiv preprint arXiv:1310.0863 (2013).

[90] N. Delfosse, V. Londe, and M. E. Beverland, Toward a Union-Find Decoder for Quantum LDPC Codes, IEEE Transactions on Information Theory **68**, 3187 (2022).

[91] P. Panteleev and G. Kalachev, Degenerate Quantum LDPC Codes With Good Finite Length Performance, Quantum **5**, 585 (2019).

[92] Y. Wu, L. Zhong, and S. Puri, Hypergraph Minimum-Weight Parity Factor Decoder for QEC, in *Bulletin of the American Physical Society* (American Physical Society, 2024).

[93] H. Bombín, Gauge Color Codes: Optimal Transversal Gates and Gauge Fixing in Topological Stabilizer Codes, New Journal of Physics **17**, 10.48550/arxiv.1311.0879 (2013).

[94] N. Liyanage, Y. Wu, A. Deters, and L. Zhong, Scalable Quantum Error Correction for Surface Codes using FPGA, Proceedings - 31st IEEE International Symposium on Field-Programmable Custom Computing Machines, FCCM 2023 , 217 (2023).

[95] T. Richardson and R. Urbanke, *Modern coding theory* (Cambridge University Press, 2008).

[96] A. G. Fowler, Minimum weight perfect matching of fault-tolerant topological quantum error correction in average O(1) parallel time, Quantum Information and Computation **15**, 145 (2015).

[97] Y. Wu and L. Zhong, Fusion Blossom: Fast MWPM Decoders for QEC, in *Proceedings - 2023 IEEE International Conference on Quantum Computing and Engineering, QCE 2023*, Vol. 1 (2023) pp. 928–938.

[98] A. Grospellier, *Constant time decoding of quantum expander codes and application to fault-tolerant quantum computation*, Ph.D. thesis, Sorbonne Université (2019).

[99] X. Tan, F. Zhang, R. Chao, Y. Shi, and J. Chen, Scalable surface code decoders with parallelization in time, arXiv preprint arXiv:2209.09219 10.48550/arxiv.2209.09219 (2022).

[100] L. Skoric, D. E. Browne, K. M. Barnes, N. I. Gillespie, and E. T. Campbell, Parallel window decoding enables scalable fault tolerant quantum computation, arXiv preprint arXiv:2209.08552 10.48550/arxiv.2209.08552 (2022).

[101] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, V. Vuletić, and M. D. Lukin, A quantum processor based on coherent transport of entangled atom arrays, Nature 2022 604:7906 **604**, 451 (2022).

[102] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, Demonstration of the trapped-ion quantum CCD computer architecture, Nature 2021 592:7853 **592**, 209 (2021).

[103] S. Bartolucci, P. Birchall, D. Bonneau, H. Cable, M. Gimeno-Segovia, K. Kieling, N. Nickerson, T. Rudolph, and C. Sparrow, Switch networks for photonic fusion-based quantum computing, arXiv preprint arXiv:2109.13760 (2021).

METHODS

Background Concepts

In this section, we review some common concepts and definitions used to establish the fault tolerance of our scheme. We will focus on a high-level description here, and defer the formal definitions to the supplementary information. Experienced QEC researchers may wish to skip ahead to the **key concepts** section, where we discuss a number of less commonly used concepts that are key to our results.

We start by reviewing the ideal circuits we aim to perform, based on Clifford operations and magic state teleportation. We then describe how to turn this into an error-corrected circuit. First, we define the local stochastic noise model that our proof assumes, which covers a wide range of realistic scenarios. We then describe the quantum LDPC codes that we use to perform quantum error correction and how to perform transversal logical operations on them. A noisy transversal realization of the ideal circuit is thus obtained by replacing each ideal operation by the corresponding transversal gate, followed by a single SE round. The error-corrected realization also determines how errors trigger syndromes, which is captured in the detector error model (decoding hypergraph). Using the detector error model and observed syndromes, we can infer a recovery operator which attempts to correct the actual errors.

Together, these concepts establish the basic procedures that are typically used for quantum error correction and conventional FT analysis. However, in order to establish fault tolerance for our algorithmic FT protocol, we need to introduce the additional notion of frame variables, which capture the randomness of initial stabilizer projections during state preparation, and we discuss how to interpret logical measurement results in the presence of such degrees of freedom in the next section.

Ideal circuit $\mathcal{C}$. We consider ideal circuits $\mathcal{C}$ in a model of quantum computation consisting of Clifford operations and magic state inputs. $\mathcal{C}$ includes state preparation and measurement in the computational basis for any qubit, single-qubit I, Z, H, S gates, $CNOT$ gates between any pair of qubits. This allows the implementation of any Clifford unitary. $\mathcal{C}$ can also include conditional operations of the above types, conditioned on previous measurement results. Finally, $\mathcal{C}$ can also include non-Clifford magic state inputs of the form $|T\rangle = T|+\rangle$ inputs, where the T gate is a $\pi/4$ rotation around the Z axis. This set of operations is known to be universal for quantum computation [63]. We require that all qubits are measured by the end of the circuit.

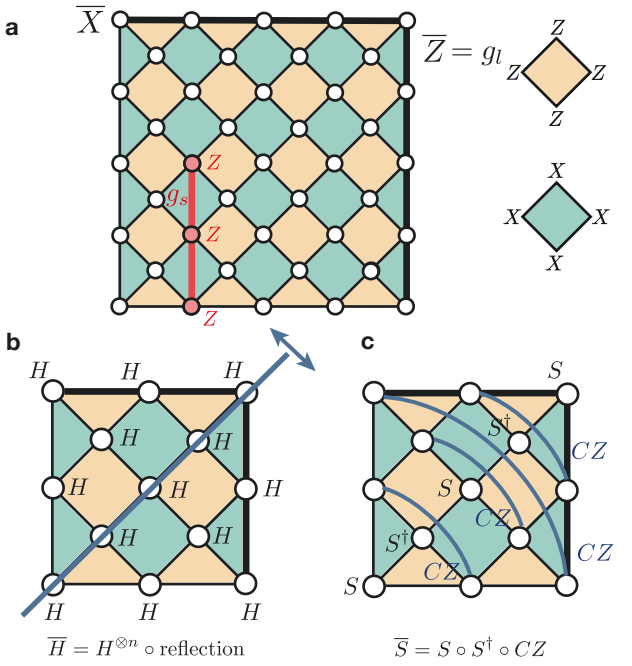
Measurement distribution $f_{\mathcal{C}}$ of ideal circuit $\mathcal{C}$. Ultimately, we are only interested in the classical results that our quantum computation returns. Denote the total number of logical measurements performed throughout $\mathcal{C}$ as M. The output of each execution of $\mathcal{C}$ is a bit string $\vec{b}_{\mathcal{C}} \in \mathbb{Z}_2^M$, sampled from a probability distribution $f_{\mathcal{C}}$. This probability distribution fully characterizes the output of the quantum computation.

Local stochastic noise model. Our proof assumes the local stochastic noise model that is widely used in fault-tolerance analysis, see for example Ref. [5]. This noise model allows for noise correlations, but requires that the probability of any set of s errors is upper-bounded by p^s, where p is a parameter characterizing the noise strength. We will use the local stochastic noise model in Ref. [5, 20], where the noise is applied to data qubits and the output syndrome bit. A basis of the errors is denoted as $\mathcal{E}$ and its size scales with the space-time volume of the circuit. For a QLDPC code (see below) and syndrome extraction circuit with bounded depth, this can be readily generalized to show a circuit-level threshold by using the fact that error propagation is bounded in a constant depth circuit [5, 64, 65].

Quantum LDPC Code. An $[[n, k, d]]$ stabilizer quantum code $\mathcal{Q}$ is an (r, c)-LDPC (low-density parity check) code if each stabilizer generator has weight $\leq r$ and each data qubit is involved in at most c stabilizer generators. Here, n denotes the number of phyiscal data qubits, k the number of encoded logical qubits, and d the code distance. Here and below, we will use an overline to indicate logical operations and logical states, e.g. $\overline{U}$ and $|\overline{0}\rangle$. Due to the random initial stabilizer projection, we also use the separate double-bar notation $|\overline{\overline{0}}\rangle$ to denote the ideal logical code state with all stabilizers fixed to $+1$.

A widely-used family of quantum LDPC codes is the surface code, due to its 2D planar layout and high threshold. The surface code, together with its X and Z stabilizers and logical operators, are illustrated in ED Fig. 1(a).

Transversal operations. Consider a fixed partition of a code block, where each part contains at most t qubits. We call a physical implementation U of a logical operation $\overline{U}$ transversal with respect to this partition, if it exclusively couples qubits within the same part [66, 67]. We will also restrict our attention to the case where the logical operation, excluding SE rounds, has depth 1, motivated by the fact that the elementary gates in the ideal circuit $\mathcal{C}$ have depth 1. We consider the same, fixed partition for all logical qubits throughout the algorithm. This definition includes common transversal gates such as $\overline{CNOT}$ on CSS codes, for the partition where each physical qubit is an individual part. For the surface code, we can choose a partition of size at most two, which pairs together qubits connected by a reflection. Common Clifford operations are transversal with respect to this partition, see ED Fig. 1(c-d): $\overline{H}$ can be implemented via a physical H on each qubit, followed by a code patch reflection in a single step. The $\overline{S}$ gate can be implemented via CZ on pairs of qubits connected by a reflection and $S/S^\dagger$ along



Extended Data Fig. 1. (a) Illustration of the surface code. White circles indicate data qubits. Orange (green) plaquettes are Z (X) stabilizers. The logical $\overline{Z}$ ($\overline{X}$) operator runs vertically (horizontally), and we choose our convention for fixing Z (X) stabilizers to be performing a chain of X (Z) flips to the left (bottom) boundary, as illustrated by the red line. (b) Illustration of transversal $\overline{H}$ gate, consisting of transversal H gates followed by a reflection along the diagonal. Note that this differs from the usual transversal $\overline{H}$ gate, which applies a rotation in the second step. For the non-rotated surface code, both choices map X (Z) stabilizers to Z (X) stabilizers and hence are valid, but our choice leads to a smaller transversal partition size for the full circuit. (c) Illustration of transversal $\overline{S}$ gate, consisting of S and $S^\dagger$ gates along the diagonal, together with CZ gates between mirrored qubits.

the diagonal [58, 68–70]. We also refer to the following state preparation and measurement in the computational basis as transversal, where $|\overline{0}\rangle$ state preparation involves preparing all physical qubits in $|0\rangle$ and measuring all stabilizers once, while measurement involves measuring all physical qubits in the Z basis. Note that the $|\overline{0}\rangle$ state preparation procedure does not prepare the actual code state, but rather an equivalent version with random X stabilizers, where information regarding the random stabilizer initialization can be deduced later.

Transversal realization $\tilde{\mathcal{C}}$ of ideal circuit $\mathcal{C}$. If the set of operations involved in the ideal circuit (other than magic state preparation, see below) admit a transversal implementation with the QEC code $\mathcal{Q}$, then we can obtain a transversal error-corrected realization $\tilde{\mathcal{C}}$ of the ideal circuit $\mathcal{C}$. $\tilde{\mathcal{C}}$ is obtained from $\mathcal{C}$ by replacing each operation by the corresponding transversal operation and inserting only one round of syndrome extraction following each gate. Here, all transversal gate operations are Clifford gates, and non-Clifford gates are implemented via magic state teleportation. The number of syndrome extraction rounds can be further optimized in practice [15]. We denote the noiseless version of this circuit as $\tilde{\mathcal{C}}_0$, and the circuit with a given error realization e from the local stochastic noise model as $\tilde{\mathcal{C}}_e$.

The surface code provides a concrete example of a code that admits a transversal implementation of all transversal Clifford operations mentioned above. Although we use the surface code as a concrete instance that realizes all required transversal gates, the transversal algorithmic FT construction we propose works more generally. For a specific quantum circuit, it may be possible to compile it into, e.g. transversal CNOTs and fold-transversal gates for multiple copies of other QLDPC codes [69, 70], where our results also apply.

When considering magic state inputs, we assume that the magic state is initialized in the desired state with all stabilizer values fixed to $+1$, up to local stochastic noise on each physical qubit of strength p. However, we also generalize this in Theorem 3 below to the case where the magic state input is at a smaller code distance, and show FT of single step patch growth, closely mirroring the situation in practical multi-level magic state distillation factories [47, 48]. Since magic states for the surface code are typically prepared using magic state distillation, we expect that our methods allow single-shot logical operations during these procedures as well, which consist of Clifford operations and noisy magic state inputs (see the following section on State Distillation Factories). Therefore, compared to standard techniques such as lattice surgery, we expect the transversal realization $\mathcal{C}$ to have a time cost that is a factor of $\Theta(d)$ smaller.

Detector error model. To diagnose errors, we form detectors (also known as checks), which are products of stabilizer measurement outcomes that are deterministic in the absence of errors. A basis of detectors is denoted as $\mathcal{D}$. We denote the set of detectors that a given error triggers as ∂e, which can be efficiently inferred [39]. In other words, we have a linear map

$$\partial : \mathbb{Z}_2^{|\mathcal{E}|} \to \mathbb{Z}_2^{|\mathcal{D}|}. \tag{2}$$

The error model, together with the pattern of detectors a given set of errors triggers, forms a decoding hypergraph Γ, also known as a detector error model, see e.g. Ref. [15, 38, 39, 71, 72]. The vertices of this graph are detectors, hyperedges are elementary errors, and a hyperedge is connected to the detectors that the corresponding error triggers. During a given execution of the noisy circuit, there will be some pattern of errors e that occur, giving some detection event ∂e. Since the circuit is adaptive based on past measurement results, the detector error model must also be constructed adaptively to incorporate the conditional feed-forward operations. More specifically, the decoding hypergraph $\Gamma|_j$ for the

jth logical measurement in a given run is constructed after committing to the previous $j-1$ logical measurement results, and similarly for other objects.

To analyze error clusters, we also introduce the related notion of the syndrome adjacency graph Ξ [5]. In this hypergraph, vertices are elementary fault locations, and hyperedges are detectors connecting the fault locations they flip.

Inferred recovery operator κ. Given the detection events and the detector error model, we can perform decoding to identify a recovery operator $\kappa \in \mathbb{Z}_2^{|\mathcal{E}|}$ which triggers the same detector pattern $\partial\kappa = \partial e$. Our proof makes use of the most-likely-error (MLE) decoder [15, 73, 74], which returns the most probable error event κ with the same detector pattern $\partial\kappa = \partial e$. We will refer to the combination $f = e \oplus \kappa$ as the "fault configuration", where $\oplus$ denotes addition modulo 2. By linearity, the fault configuration $e \oplus \kappa$ will not trigger any detectors,

$$\partial(e \oplus \kappa) = 0. \tag{3}$$

Forward-propagated error $P(e)$. A Pauli error E occurring before a unitary U is equivalent to an error $UEU^\dagger$ occurring after the unitary. For a set of errors e, we can forward-propagate it through the circuit until it reaches measurements. We denote the final operator the errors transform into as $P(e)$, and denote its restriction onto the jth logical measurement as $P(e)|_j$. This is related to the cumulant defined in Ref. [38] and the spackle operator in Ref. [75].

Key Concepts

We now introduce a few concepts that are less commonly discussed in the literature, but are important for our analysis. We start by describing the randomness associated with transversal state initialization and stabilizer projections. To do so, we introduce frame variables g. To capture the random reference frame corresponding to random initialization of stabilizer values upon projection, we introduce frame stabilizer variables g_s. These correspond to certain Pauli Z operators that flip a subset of X stabilizers, and we call both these operators and the binary vector that describes them as frame variables, where the meaning should be clear from context. The Pauli logical initial state, e.g. $|\overline{0}\rangle$, also has a logical stabilizer $\overline{Z}$, which we describe with frame logical variables g_l. Applying frame logical variables on the initial state does not change the logical state, since we are applying a logical stabilizer, but this does change the interpretation of a given logical measurement shot. To interpret logical measurement results, we must perform a frame repair operation that returns all stabilizers to $+1$, mirroring the error recovery inference. However, there can be some degree of freedom in choosing the frame logical variable, which allows us to ensure consistency between multiple rounds of decoding. These understandings lead us to propose the decoding strategy shown in Fig. 2, and will be crucial to our FT proofs below.

Frame variables g. When performing transversal state initialization, all physical qubits are prepared in $|0\rangle$, and stabilizers are measured with an ancilla. The outcome of the X stabilizers will thus be random. Following the approach taken in Ref. [39], this randomness can be captured by additional Z operators acting at initialization. Concretely, for each data qubit i, we add Z_i to a basis of frame operators $\mathcal{G}$ if it is not equivalent to any combination of operators in $\mathcal{G}$ up to stabilizers. The state after random stabilizer projection is equivalent to starting with the ideal code state $|\overline{\overline{0}}\rangle$ and applying a set of Z operators; in other words, $|\overline{0}\rangle = g|\overline{\overline{0}}\rangle$. We refer to these operators as frame operators, as they describe the effective code space ("reference frame") with random stabilizers that we projected into, and help interpret logical measurement results. The set of Z operators that produces a given pattern of initial stabilizer values can be efficiently determined by solving a linear system of equations. We choose a basis $\mathcal{G}$ for these operators, as defined above, and denote with g both the Pauli operator corresponding to a frame variable as well as the binary vector describing it:

$$g \in \mathbb{Z}_2^{|\mathcal{G}|}, \quad |\mathcal{G}| = B(n - r_Z), \tag{4}$$

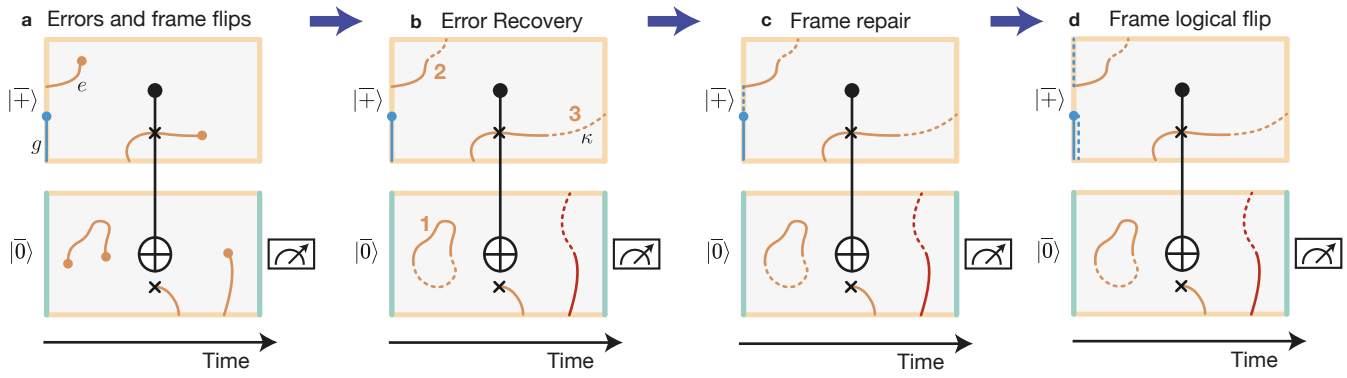
where B is the number of code blocks used, n is the number of data qubits per block and r_Z is the number of independent Z stabilizer generators per block. In the presence of noise, we can imagine first performing the random stabilizer projection perfectly, and then performing a noisy measurement of the syndromes via ancillae and recording the results. Although this does not allow the reliable inference of frame variables, we will show that the transversal measurement provides enough information to infer the relevant degrees of freedom for interpreting logical measurement results.

Frame logical variables g_l. A special subset of frame variables are frame logical variables

$$g_l \in \mathbb{Z}_2^{Bk}, \tag{5}$$

which are combinations of the Z operators that form a logical $\overline{Z}$ operator of the code block, and therefore act trivially on the code state $|\overline{0}\rangle$. Here, B is the number of code blocks and k is the number of logical qubits per block. While they do not change the initialized physical state, nor do they flip any stabilizers, different choices of the frame logical variables when decoding will lead to different interpretations of the logical measurement result, as we explain next.

Frame stabilizer variables g_s. We refer to frame variables that are not frame logical variables as frame stabilizer variables. These variables will flip the randomly



Extended Data Fig. 2. **Illustration of error recovery and frame repair procedures.** We illustrate the procedure for the surface code, where a cross-sectional view with one spatial axis and one time axis is shown. We only illustrate X errors and Z stabilizer measurement errors, which are relevant to interpreting the $\overline{Z}$ measurement. X errors can terminate on orange boundaries, but cannot terminate on cyan boundaries. The transversal $\overline{CNOT}$ copies X errors from the top to the bottom, resulting in a branching point (black cross) and an error cluster spanning both code blocks. (a) Error chains and frame flips. Chains of X-type errors (orange lines) lead to syndromes (end points) or terminate on appropriate boundaries. A line segment in the vertical direction is a data qubit X error, while a line segment in the horizontal direction is a measurement error. Note that the X-type error cannot terminate on the transversal Z measurement boundary. The random stabilizer initialization leads to a frame configuration on the logical $|\overline{+}\rangle$ initialization, as illustrated by the blue line and the flipped Z stabilizer (blue point). This is similar to the frame stabilizer operator g_s illustrated in ED Fig. 1(a). (b) We first infer an error recovery operator, which has the same boundary as the error chain. Together, the error and recovery operator form the fault configuration, which triggers no detectors. We illustrate a few examples (orange lines) that do not lead to a logical error: (1) the fault configuration forms a closed loop and is equivalent to applying a stabilizer; (2) the fault configuration terminates on an initialization boundary; (3) the fault configuration terminates on a future time boundary (unmeasured logical qubit), but the forward-propagated errors onto the measured logical qubit are equivalent to a stabilizer. A logical error can only happen when the fault configuration spans across two opposing spatial boundaries (red line), which requires an error of weight $\Theta(d)$. (c,d) The frame repair operation returns the logical qubit to the code space with all stabilizers $+1$, corresponding to cancelling any residual flipped stabilizers on the initialization boundary. Note that the error recovery process may also lead to a change that needs to be accounted for by frame repair. An example choice of frame repair is shown in (c), which applies an overall X operator on the logical measurement result. Alternatively, a different choice of frame repair shown in (d), related to the previous one by a frame logical flip, results in identity operation on the logical measurement result.

initialized stabilizer values. An example is shown in ED Fig. 1(a), in which a chain of Z errors connecting to the bottom boundary flips a single stabilizer.

Interpreting logical measurement outcomes in the presence of frame variables. We now describe how to interpret logical measurement results in the presence of randomly initialized frame variables.

First, in the presence of noise, we apply the decoding procedure and obtain an error recovery operator κ such that $\partial(\kappa \oplus e) = 0$. Note that $\kappa \oplus e$ may have some nontrivial projection onto the initialization boundary, such as string 2 that terminates on the $|\overline{+}\rangle$ boundary in ED Fig. 2(b). This projection can modify the effective frame, and must be taken into account when returning things to the code space.

Next, we perform an analogous procedure to error recovery for the frame variables. Specifically, we perform a frame repair operation

$$\lambda \in \mathbb{Z}_2^{|\mathcal{G}|} \tag{6}$$

to return to the code space with all stabilizers set to $+1$. This corresponds to an inference of what the reference frame was after the random stabilizer projection during initialization, and the repair operation should be viewed as being applied on the corresponding initialization boundary as well. In other words, we require $(e \oplus \kappa) \oplus (g \oplus \lambda)$ to act as a stabilizer or logical operator, such that the stabilizer values are the same as the ideal code state $|\overline{\overline{0}}\rangle$. We will refer to the combination $h = g \oplus \lambda$ as the "frame configuration". Following this step, all frame stabilizer variables g_s have been determined, but we still have freedom to choose our frame logical variables g_l.

Finally, we evaluate the product of Pauli operators to determine the logical measurement result. Denote the raw logical observable inferred from the bit strings as

$$\overline{L}(z) = \bigoplus_{z_i \in \overline{L}} z_i, \tag{7}$$

and the corrected logical observable after applying the error recovery operation κ and frame repair operation λ as

$$\overline{L_c}(z, \kappa, \lambda) = \overline{L}(z) \oplus \overline{F}(\kappa) \oplus \overline{F}(\lambda), \tag{8}$$

where $\overline{F}(\kappa)$, $\overline{F}(\lambda)$ indicates the parity flip of the logical observable due to the operator κ, λ.

In the noiseless case, the raw logical measurement result is equivalent to the ideal measurement result that one would obtain if one had perfectly prepared the ideal code state $|\overline{\overline{0}}\rangle$, up to the application of $\overline{F}(g \oplus \lambda)$ on the initial state. However, $g \oplus \lambda$ consists of physical Z operations only and commutes with all stabilizers, so it must act as a combination of Z stabilizers and logical Z operator on $|\overline{\overline{0}}\rangle$. Therefore, it does not change the distribution of measurement results, although it can change the interpretation of individual shots. The procedure in the noisy case can be reduced to the noiseless case after applying the MLE recovery operator κ, with a suitable modification to the repair operation λ to account for fault configurations that terminate on initialization boundaries and therefore forward-propagated to flip some stabilizers on the relevant logical measurement (ED Fig. 2(c)).

Decoding strategy. A key component of our FT construction is the decoding strategy. In our setting with transversal Clifford gates only, classical decoding only becomes necessary when we need to interpret logical measurement results. We sort the set of logical measurements into an ordering $\{\bar{L}_1, \bar{L}_2, \bar{L}_3, ..., \bar{L}_M\}$ based on the time they occur, and then decode and commit to their results in this order.

For the jth logical measurement $\bar{L}_j$, we first apply the most-likely-error (MLE) decoder to the available detector data $\mathcal{D}|_j$ and the detector error model $\Gamma|_j$, where $|_j$ denotes that this information is restricted to information up to the jth logical measurement. Note that since we allow feed-forward operations, the decoding hypergraph may differ in each repetition of the circuit (shot). After this first step, we will have obtained an inferred recovery operator κ, similar to standard decoding approaches.

The second step is to apply frame logical variables g_l such that previously-committed logical measurement results retain the same measurement result. It may not be clear a priori that this is always possible, but we prove that below a certain error threshold p_{th}, the probability of a failure decays to zero exponentially in the code distance. This guarantees that we are always consistently assigning the same results to the same measurement in each round of decoding. The assignment of frame logical variables can be solved efficiently using a linear system of equations.

Proof Sketch

In this section, we provide a sketch of our FT proof, using the concepts introduced above. Our reasoning follows three main steps:

1. We show that the transversal realization reproduces the logical measurement result distribution of the ideal circuit, regardless of the reference frame we initially projected into.

2. We obtain perfect syndrome information on the logical qubits via transversal measurements, which we then combine with correlated decoding to handle errors throughout the circuit and guarantee that any logical error must be caused by a high-weight physical error cluster.

3. By counting the number of such high-weight error clusters, we show that when the physical error probability is sufficiently low, the growth in the number of error clusters as the distance increases is slower than the decay of probability of high-weight clusters, thereby establishing an error threshold and exponential sub-threshold error suppression.

We now explain a set of useful lemmas that lead to our main theorem.

Frame variables g do not affect the logical measurement distribution. We show that the choice of frame variables g does not affect the logical measurement distribution $f_{\tilde{C}}$. Intuitively, this is because different choices of frame variables are equivalent up to the application of $\bar{Z}$ logicals on $|\overline{\overline{0}}\rangle$, which does not affect the logical measurement distribution. Indeed, as long as we are able to keep track of which subspace of random stabilizer values we are in, achieved via the transversal measurement, the measurement result distribution should not be affected.

$f_{\mathcal{C}} = f_{\tilde{\mathcal{C}}_0}$. In other words, the noiseless transversal realization $\tilde{\mathcal{C}}_0$ produces the same distribution of logical bit strings as the ideal quantum circuit $\mathcal{C}$. This can be seen from the previous statement by choosing all frame variables to be zero and invoking standard definitions of logical qubits and operations.

Transversal gates limit error propagation. One major advantage of transversal gates is that they limit error propagation [4, 7], thereby limiting the effect any given physical error event can have on any logical qubit. With the bounded cumulative partition size t defined above, one can readily show that any error e acting on at most k qubits can cause at most tk errors on a given logical qubit, when propagated to a logical measurement $P(e)|_j$.

Effect of low-weight faults on code space. Consider the syndrome adjacency graph $\Xi|_j$, which is the line graph of the detector error model $\Gamma|_j$ corresponding to the first j logical measurements, and any fault configuration $f|_j = (e \oplus \kappa)|_j$. We show that if the largest weight of any connected cluster of $f|_j$ is less than d/t, then there exists a choice of frame repair operator $\hat{\lambda}_j$, such that the forward propagation of fault configuration and frame configuration

$$P(e|_j \oplus \kappa|_j) \oplus P(g|_j \oplus \hat{\lambda}_j) \tag{9}$$

acts trivially on the first j logical measurements.

The intuition for this statement is illustrated in ED Fig. 2. Suppose without loss of generality that the logical measurement we are examining is in the Z basis, then we only need to examine errors that forward-propagate to X errors. By definition, the fault configuration $e \oplus \kappa$ and frame configuration $g \oplus \lambda$ should return things to the code space and not trigger any detectors, implying that the X basis component of $P(e \oplus \kappa \oplus g \oplus \lambda) = P(f \oplus h)$ is a product of X stabilizers and logical operators. Consider each connected component f_i of $f|_j$, then by transversality (previous lemma) and $\mathrm{wt}(f_i) < d/t$, we have $\mathrm{wt}(P(f_i)) < d$.

Case 1: If f_i does not connect to a Pauli initialization boundary (fault configurations 1 and 3 in ED Fig. 2(b)), then it is also a connected component of $f \oplus h$, since the frame configuration lives on the initialization boundary. Since $P(f_i)$ has weight less than d, it must be a stabilizer and therefore acts trivially on the logical measurement under consideration.

Note that because magic states are provided with known stabilizer values up to local stochastic noise, connected components of the fault configuration cannot terminate on them without triggering detectors. The same also holds for measurement boundaries or boundaries in which the initialization stabilizer propagates to commute with the final measurement. Only when the initialization stabilizer propagates to anti-commute can we connect to the boundary, as described in case 2, but this also then implies that the measurement is 50/50 random and can be made consistent using our methods.

Case 2: Now suppose f_i connects to an initialization boundary (fault configuration 2 in ED Fig. 2(b)) and its connected component $P(f_i \oplus h_i)$ acts as a nontrivial logical operator L, flipping the logical measurement. In this case, we can choose a different frame repair operator such that $P(\hat{\lambda}) = P(\lambda) \oplus L$, which does not flip the logical measurement. In ED Fig. 2(c,d), we can intuitively think of this as changing whether the frame repair connects in the middle or to the two boundaries. In one of these two cases, the total effect of the fault configuration and frame configuration is trivial on the logical measurements of interest (ED Fig. 2(d) in this case).

Thus, we see that when the fault configuration only involves connected clusters of limited size, its effect on the logical measurement results is very limited. This leads to a key technical lemma that lower bounds the number of faults required for a logical error to occur.

Logical errors must be composed of at least d/t faults. Due to the decoding strategy we employ, there are two types of logical errors we must account for.

First, we may have a logical error in the usual sense, where the distribution of measurement results differs from the ideal quantum circuit $f_{\tilde{\mathcal{C}}} \neq f_{\mathcal{C}}$. It is important to note here, however, that this deviation is in the distribution sense. Thus, if a measurement outcome that was 50/50 random was flipped, it does not cause a logical error yet, as the outcome is still random. In this case, it is only when the joint distribution with other logical measurements is modified that we say a logical error has occurred. When analyzing a new measurement result with some previously committed results, we analyze the distribution conditional on these previously committed results.

Second, there may be a heralded logical error, in which no valid choice of frame repair operation λ exists in the second step of our decoding strategy. More specifically, there is no λ that makes all logical measurement results identical to their previously-committed values.

We show that when the largest weight of any connected cluster in the fault configuration is less than d/t, neither type of logical error can occur. The absence of unheralded logical errors can be readily seen from the above characterization of the effect of low-weight faults on the code space. To study heralded errors, we make slight modifications to analyze the consistency of multiple rounds of decoding, and find that heralded errors require one of the two rounds of decoding that cannot be consistently assigned to have a fault configuration with weight $\geq d/t$, thereby leading to the desired result.

Counting lemma. The counting lemma is a useful fact that bounds the number of connected clusters of a given size within a graph, with many previous uses in the QEC context [5, 25, 28, 76, 77]. It shows that for a graph with bounded vertex degree v and n vertices, as is the case for the syndrome adjacency graph Ξ of qLDPC codes, the total number of clusters of size s is at most $n(ve)^{s-1}$. This bounds the number of large connected clusters. When the error rate is low enough, the growth of the "entropy" factor associated with the number of clusters will be slower than the growth of the "energy" penalty associated with the probability, and thus the logical error rate will exponentially decrease as the system size is increased, allowing us to prove the existence of a threshold and exponential sub-threshold suppression.

Theorem 1: Threshold theorem for transversal realization $\tilde{\mathcal{C}}$ with any CSS QLDPC code, with reliable magic state inputs and feed-forward. With the preceding lemmas, we can prove the existence of a threshold under the local stochastic noise model. Using the counting lemma, we can constrain the number of connected clusters N_s of a given size s on the syndrome adjacency graph Ξ. For a connected cluster of size s, MLE decoding implies that at least $s/2$ errors must have occurred, which has bounded probability scaling as $p^{s/2}$ under the local stochastic noise model. Our characterization of logical errors implies that a logical error can only occur when $s \geq d/t$. For each round of logical measurements, the probability of a logical error is then bounded by a geometric series summation over cluster sizes s, with an entropy factor from cluster number counting and an energy factor from the exponentially decreasing proba-

bility of each error event:

$$P_{err} \propto M \sum_{s=\frac{d}{t}}^{\infty} N_s 2^s p^{s/2}$$
$$\propto (2v\mathfrak{e}\sqrt{p})^{d/t} = \left(\frac{p}{1/(2v\mathfrak{e})^2}\right)^{d/2t}, \qquad (10)$$

where v is a bound on the vertex degree of the syndrome adjacency graph and is dependent on the degrees r and c of the QLDPC code. When the error probability p in the local stochastic noise model is sufficiently small, the latter factor outweighs the former, and the logical error rate decays exponential to zero as the code distance increases, with an exponent $p^{d/2t}$. We can then take the union bound over rounds of logical measurements to bound the total logical error probability.

While our theorem assumes reliable magic state inputs with local stochastic data qubit noise only, we expect our results to readily generalize to magic state distillation factories (see next section and discussion in main text), thereby enabling a $\Theta(d)$ saving for universal quantum computing.

Note that to prove a threshold theorem for FT simulating the ideal circuit $\mathcal{C}$, we need a family of codes $\{\mathcal{Q}\}$ with growing size that provide a transversal realization of $\mathcal{C}$. For general high-rate QLDPC codes, this may be challenging, as the set of transversal gates is highly constrained [69, 70]. However, we will now show that the surface code provides the required code family.

Theorem 2: Fault tolerance for arbitrary Clifford circuits with reliable magic state inputs and feed-forward, using a transversal realization with the surface code. We can further specialize the preceding results to the case of the surface code. With the transversal gate implementations of H, S and $CNOT$, we can implement arbitrary Clifford operations with cumulative partition size $t = 2$. Note that with more detailed analysis of the error events and gate design, it may be possible to recover the full code distance d (instead of the $d/2$ proven here), which we leave for future work. Our threshold and error suppression results are independent of the circuits implemented, e.g. whether the circuit has a large depth or width. The resulting logical error rate scales linearly with the circuit space-time volume and number of logical measurements, and is exponentially suppressed in the code distance, similar to the usual threshold theorems.

A straight-forward application of the previous theorem shows the existence of a threshold and exponential sub-threshold error suppression. Importantly, the surface code provides all elementary Clifford operations, thereby giving a concrete code family for the FT simulation of any ideal circuit $\mathcal{C}$, as long as we are provided with the appropriate magic state inputs, which can in turn be obtained in the same way via magic state distillation.

Single-shot code patch growth. To further extend the applicability of our results, we also analyze a setting in which reliable magic states are provided at a code distance d_1 smaller than the full distance d of the main computation. This is relevant, for example, to multi-stage magic state distillation procedures that are commonly employed to improve the quality of noisy injected magic state inputs. Lower levels of magic state distillation are typically performed at a reduced code distance, due to the less stringent error rate requirements, before they are grown into larger distance for further distillation, as is the case in Fig. 4.

By analyzing which stabilizers are deterministic during the code patch growth process, we find that a strip of width d_1 has deterministic values. A fault configuration that causes a logical error must span across this region, and thus have weight at least d_1. Therefore, in this case we still have fault tolerance and exponential error suppression, but with an effective distance now modified to scale as d_1 instead of d, set by the smaller patch size of the magic state input as expected.

State Distillation Factories

In this section, we provide more details on state distillation factories. First, we derive the output fidelity of the $|\overline{Y}\rangle$ state distillation factory described in the main text, as a function of input $|\overline{Y}\rangle$ state fidelity and assuming ideal Clifford operations within the factory. Second, we illustrate the 15-to-1 $|\overline{T}\rangle$ magic state distillation factory and comment on a few simplifications that our decoding strategy enables in executing this factory.

The $|\overline{Y}\rangle$ state distillation factory described in the main text prepares a Bell pair between a single logical qubit and seven logical qubits further encoded into the $[[7, 1, 3]]$ Steane code. Applying a transversal $\overline{S}$ gate on the Steane code then leads to a $\overline{S}$ gate on the output logical qubit due to the Bell pair. Error detection on the Steane code further allows one to distill a higher-fidelity logical state. For this distillation factory, we can directly count the error cases for the magic state input that lead to a logical error, conditional on post-selection results. For example, there are seven logical $\overline{Z}$ representatives of weight three and one logical representative of weight seven, and the application of a logical representative leads to an undetectable error. Counting all possible combinations, we arrive at the following formula for noisy magic state inputs and ideal Clifford operations

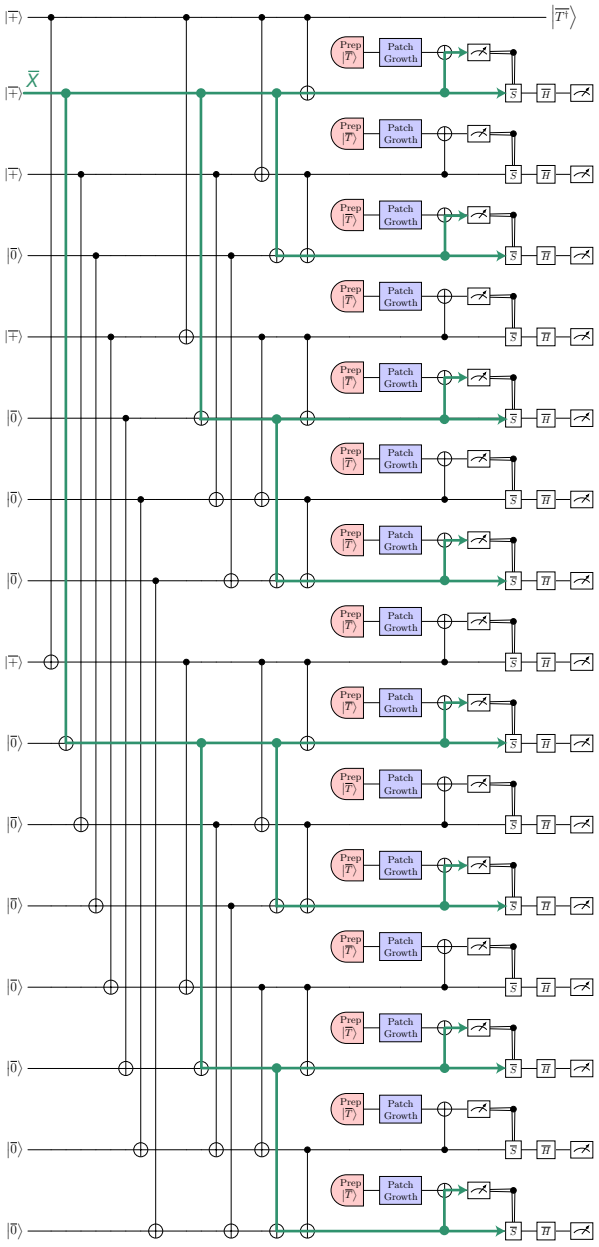
$$P_{out} = \frac{7P_{in}^3(1-P_{in})^4 + P_{in}^7}{(1-P_{in})^7 + 7P_{in}^3(1-P_{in})^4 + 7P_{in}^4(1-P_{in})^3 + P_{in}^7}$$
$$\approx 7P_{in}^3, \qquad (11)$$

where P_{out} is the output logical error rate and P_{in} is the input logical error rate. For our numerical simulations, we artificially inject Z errors for the input state.

In ED Fig. 3, we illustrate the 15-to-1 $|\overline{T}\rangle$ state distillation factory, which takes 15 noisy $|\overline{T}\rangle$ states and distills a single high quality $|\overline{T}\rangle$ state. As described in Ref. [33], assuming ideal Clifford operations, the rejection probability scales linearly with the input infidelity, while the output logical error rate scales with the cube of the input infidelity. The $|\overline{T}\rangle$ factory bears a lot of similarities with the $|\overline{S}\rangle$ factory in the main text: In both cases, we start with Pauli basis states, apply parallel layers of CNOT gates, and then perform resource state teleportation using a CNOT. The resource states at the lowest level can be prepared using state injection, which is agnostic to the precise quantum state being injected and therefore should apply equally to a $|\overline{S}\rangle$ and $|\overline{T}\rangle$ state, while the resource states at the higher levels are obtained by lower levels of the same distillation factory. The main difference is that because the feed-forward operation is now a Clifford instead of a Pauli, the feed-forward gate must be executed in hardware, rather than just kept track of in software.

When performing magic state distillation and teleporting the magic state into the main computation, the first step of our protocol requires correlated decoding of the distillation factory and main computation together. It will be interesting to formally extend our threshold analysis to incorporate noisy magic state injection and state distillation procedures. As low-weight logical errors are localized around the state injection sites, we expect common arguments regarding the error scaling of distillation factories to hold, as is also supported by our numerical results. We leave a detailed proof of this to future work. In practice, to reduce the decoding cost, one can also insert $\Theta(d)$ SE rounds on the single output logical qubit of the factory, in order to separate the system into modular blocks [71]. Since we only need to insert the $\Theta(d)$ SE rounds on a single logical qubit, while a two-level distillation factory typically involves hundreds of logical qubits [47, 48], we expect that this will only cause a slight increase in the total distillation cost.

Using our decoding strategy, it is possible to reduce the number of feed-forward operations that need to be executed. As illustrated in ED Fig. 3, we can apply an $\overline{X}$ operator on the $|\overline{+}\rangle$ logical initial states, which is a logical stabilizer of the resulting quantum state. Applying this operator flips the interpreted results of some subset of logical measurements. Thus, we can always choose to not apply a feed-forward $\overline{S}$ on the first $|\overline{T}\rangle$ teleportation, but instead change what feed-forward operations are applied on the remaining $|\overline{T}\rangle$ teleportations. There are 15 $|\overline{T}\rangle$ teleportations to be implemented and 5 $|\overline{+}\rangle$ logical state initialization locations. Therefore, we expect that at most 10 feed-forward operations need to be applied. Using these techniques, the logical qubit locations where the feed-forward operations need to be applied may also be adjusted, which may be beneficial for the purpose of control parallelism [36].



Extended Data Fig. 3. Illustration of a 15-to-1 $|\overline{T}\rangle$ magic state distillation factory, adapted from Ref. [30]. The green lines illustrate the application of a logical stabilizer, which allows re-interpretation of measurement results and changes which feed-forwards should be performed.

Finally, we also comment on the relation of our results to other computational models that make use of magic state inputs and Clifford operations. In particular, Pauli-based computation [78, 79] has been shown to provide a weak simulation of universal quantum circuits using only magic state inputs, apparently removing the need of $|\overline{0}\rangle$ and $|\overline{+}\rangle$ logical states altogether, and clari-

fying the importance of $|\overline{T}\rangle$ state preparation in particular. However, this model relies on the logical measurements being non-destructive, and continues to use a given logical qubit after measurement, which is not possible for transversal measurements on logical qubits without Pauli basis initialization. Thus, in an error-corrected implementation, Pauli basis initialization is still necessary, and the use of our FT framework is necessary to achieve low time overhead. This comparison to other computational models highlights the generality of the algorithmic fault-tolerance framework, and indicates that universally across these various computational models, such techniques allow a $\Theta(d)$ saving.

Importance of Shallow Depth Algorithmic Gadgets

In this section, we discuss the importance of shallow-depth algorithmic gadgets in many practical compilations of quantum algorithms. This highlights the need for FT strategies that do not require a $\Theta(d)$ separation between initialization and measurement, as we developed in the main text.

In general, circuit components that involve an ancilla logical qubit often have a shallow depth between initialization and measurement, whether this ancilla is used for algorithmic reasons or compilation reasons. For instance, temporary ancilla registers are used in algorithmic gadgets such as adders [80, 81] or quantum read-only memories [82], where the bottom rail of a ripple carry structure is initialized, two or three operations are performed on it, and then the ancilla qubit is measured. A useful technique for performing multiple circuit operations in parallel is time-optimal quantum computation [14, 16, 83], which is also related to gate teleportation [63] and Knill error correction [84]. In this case, a pair of logical qubits are initialized in a Bell state. One qubit is then sent as the input into a circuit fragment A, while the other qubit executes a Bell basis measurement with the output of another circuit fragment B. The combined circuit is equivalent to the sequential execution of B and A. This allows the two circuit fragments to be executed in parallel, despite them originally being sequential, thereby reducing the total circuit depth and idling volume. However, to fully capitalize on this advantage, it is desirable to only have a constant number of SE rounds separating the Bell state initialization and Bell basis measurement, in order to minimize the extra circuit volume incurred by the space-time trade-off. Thus, a depth $O(1)$ separation between state initialization and measurement is again highly desirable.

Another common situation in which there is a low-depth separation between initialization and measurement is magic state distillation [33] and auto-corrected magic state teleportation [85]. Many magic state factories involve a constant-depth Clifford circuit (e.g. depth 4 for the 15-to-1 distillation factory), followed by application of non-Clifford rotations [13, 30, 33, 86]. The non-Clifford rotations are often implemented via noisy magic states and gate teleportation, which therefore require logical measurements. If the Clifford circuit depth has to be at least d to maintain FT, as is assumed in e.g. Ref. [42], the time cost of the magic state factory will be much larger than the case in which we can execute the circuit fault-tolerantly in constant depth, as we demonstrate here.

Decoding Complexity

In this section, we discuss the decoding complexity of our FT construction, and highlight important directions of future research. While a detailed analysis and high-performance implementation of large-scale decoding is beyond the scope of this work, this will be important for the large-scale practical realization of our scheme and to maximize the savings in space-time cost. We therefore sketch some key considerations and highlight important avenues of research that can address the decoding problem. We emphasize that much of our discussion is not specific to our FT strategy, and may also apply to other hypergraph decoding problems and existing discussions of single-shot QEC [25] (Supplementary Information).

Compared with usual decoding problems, there are two main aspects that may increase the complexity in our setting. First, the decoding problem is now by necessity a hypergraph decoding problem, involving hyperedges connecting more than two vertices, which are not decomposable into existing weight-two edges [15]. Second, the size of the relevant decoding problem (decoding volume) may be much larger, as one needs to jointly decode many logical qubits, in the worst-case reaching the scale of the full system.

The hypergraph decoding problem has been studied in a variety of different settings [15, 87–90], and heuristic decoders appear to handle this fairly well in the low error rate regime in practice. For example, polynomial-time decoders such as belief propagation + ordered statistic decoding (BPOSD) [91], hypergraph union find (HUF) [15, 90], and minimum-weight parity factor (MWPF) [92] have been shown to result in competitive thresholds. Decoding on hypergraphs is also often required for high-rate QLDPC codes, or to appropriately handle error correlations. Therefore, we expect that hypergraph decoding does not pose any serious challenge in practice.

There are several ways in which the increased decoding volume can be dealt with. First, error inferences that are sufficiently far $\Omega(d)$ away from measurements or outgoing qubits can be committed to without affecting the logical error rate [71]. This reduces the relevant decoding volume. Moreover, for underlying codes with the single-shot QEC property [25], it may be possible to further

reduce this depth.

Second, extra QEC rounds can also be inserted to reduce the relevant decoding volume and give more time for the classical decoder to keep up with the quantum computer and avoid the backlog problem [53]. Asymptotically, this may be necessary for both our scheme and for computation schemes based on single-shot quantum error correction [25, 93], unless $O(1)$-time classical decoding is possible. In both cases, the time cost will grow from $\Theta(1)$ to $\Theta(d/C)$, where the improvement factor C over conventional schemes with d SE rounds can be made arbitrarily large as the classical computation is sped up.

Third, we expect algorithms based on cluster growth (HUF and MWPF) and belief propagation to be readily amenable to parallelization across multiple cores [94–97], with the decoding problems merging only when an error cluster spans multiple decoding cores. As an error cluster of size $\Theta(d)$ is exponentially unlikely, we expect it to be unlikely for many decoding problems to have to be merged together. Indeed, fast parallel decoders for the surface code [96, 97] and QLDPC codes [98] have been argued to achieve average runtime $O(1)$ per SE round, although they still have an $O(d)$ or $O(\log d)$ latency. Therefore, although the original decoding problem is not modular (input-level modularity) [71, 99, 100], in practice we may expect the decoder to naturally split things into modular error clusters (decoder-level modularity).

Finally, there are many additional optimizations that can be applied in practice. Because the decoding problems have substantial overlap, it may be possible to make partial use of past decoding results, particularly when using clustering decoders. The decoding and cluster growth process can also be initiated with partial syndrome information and continuously updated as more information becomes available. Decoding problems with specific structure, such as circuit fragments in which the flow of CNOTs are directional (ED Fig. 3), may also benefit from specialized decoders [30]. We also note that although the relevant decoding hypergraph for any given measurement is now larger, for a given rate of syndrome extraction on the hardware, the amount of incoming data is comparable to the usual FT setting. Although the individual correlated decoding problem is larger, we will only need to solve very few of them. In both algorithmic FT and conventional FT, we expect the total amount of classical decoding resources to scale with the number of logical qubits. When decomposing correlated decoding into individual cluster decoding problems, we therefore expect the aggregate classical decoding resources required for our protocol to still remain competitive with conventional approaches.

Hardware Considerations

In this section, we briefly comment on the hardware requirements to implement our scheme. It is worth emphasizing that these requirements may be relaxed with future improvements to our construction.

Our algorithmic FT protocol makes important use of transversal gate operations between multiple logical qubits. As such, a direct implementation likely requires two key ingredients: long-range connectivity and reconfigurability. Long-range connectivity is used to entangle physical qubits that are located at matching positions in large code patches, which are otherwise spatially-separated. Reconfigurability is useful because a given logical qubit may perform transversal gates with many other logical qubits throughout its lifetime, such that a high cumulative connectivity degree is required, or multiple swaps and routing must be used. Given that common routing techniques based on lattice surgery incur a $\Theta(d)$ time cost, it is desirable to perform direct connections via reconfigurable qubit interactions.

These considerations make dynamically-reconfigurable hardware platforms such as atomic systems [35, 36, 101, 102] particularly appealing. In particular, neutral atom arrays have demonstrated hundreds of transversal gate operations on tens of logical qubits, making use of the flexible connectivity afforded by atom moving [36]. In comparison, while systems with connections based on fixed wiring can support long-range connectivity and switching [22, 103], transversal connections between multiple logical qubits likely increases the cumulative qubit degree which may significantly increase the hardware complexity. From a clock speed perspective, for typical assumed code distances of $d \sim 30$, our techniques correspond to a 10–100$\times$ speed-up by using transversal operations in a reconfigurable architecture.