

An iterative transversal CNOT decoder

Kwok Ho Wan¹, Mark Webber¹, Austin G. Fowler², and Winfried K. Hensinger^{3,1}

¹Universal Quantum Ltd, Gemini House, Mill Green Business Estate, Haywards Heath, RH16 1XQ, United Kingdom

²Google Inc., Santa Barbara, 93117 CA, USA

³Sussex Centre for Quantum Technologies, University of Sussex, Brighton, BN1 9RH, United Kingdom

Modern platforms for potential qubit candidates, such as trapped ions or neutral atoms, allow long range connectivity between distant physical qubits through shuttling. This opens up an avenue for transversal logical CNOT gates between distant logical qubits, whereby physical CNOT gates are performed between each corresponding physical qubit on the control and target logical qubits. However, the transversal CNOT can propagate errors from one logical qubit to another, leading to correlated errors between logical qubits. We have developed a multi-pass iterative decoder that decodes each logical qubit separately to deal with this correlated error. We show that under circuit-level noise and only $\mathcal{O}(1)$ code cycles, a threshold can still persist, and the logical error rate will not be significantly degraded, matching the sub-threshold logical error rate scaling of $p^{\lfloor \frac{d}{2} \rfloor}$ for a distance d rotated surface code.

1 Introduction

Quantum error correction (QEC) is essential for realising fault-tolerant quantum computation. The threshold theorem states that if the error rate per quantum gate is below a certain threshold, arbitrarily long quantum computations become possible [1]. The surface code, known for its high error threshold and ability to correct both bit-flip and phase-flip errors through local interactions, is a leading QEC candidate. It arranges qubits on a two-dimensional lattice and uses stabilizer measurements for error detection. The minimum weight perfect matching (MWPM) algorithm is an effective decoding method for the surface code [2–4].

Quantum computing hardware varies in con-

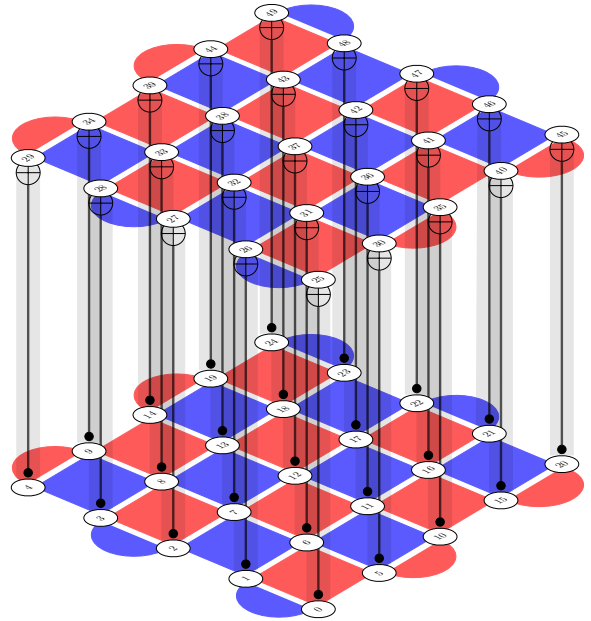


Figure 1: Two patches/copies of distance 5 rotated surface codes. A transversal logical CNOT operation is performed by apply physical CNOT gates between the corresponding physical qubits of the two patches.

nectivity capabilities. Superconducting qubits typically have fixed, limited connectivity which is imposed by the chip topology, while platforms like trapped ions, neutral atoms, and photonics allow for long-range connectivity. For example, in the trapped ion paradigm, ion shuttling operations enable arbitrary connectivity [5]. The variability of connectivity influences QEC implementation and efficiency across different systems.

Lattice surgery is a common technique for performing logical operations in the surface code while relying only on nearest neighbour operations. It involves merging and splitting logical qubits to facilitate operations such as the logical CNOT gate. The time complexity of lattice surgery scales with the code distance, typically requiring $\mathcal{O}(d^3)$ volume, where d is the code distance. One possible implementation requires a

time complexity of $2d$ and three logical qubits, one of which is an ancilla patch for mediating the merging and splitting operations [6, 7]. Note that all Clifford operations can be done in 0 time given sufficient qubits [8].

The transversal CNOT gate offers an alternative approach. It requires performing a physical CNOT operation between every corresponding physical qubit of the logical qubits. For N data qubits per logical qubit, N CNOT operations are needed, often involving long-range interactions. The transversal CNOT deterministically maps Pauli errors between patches: X errors are mapped from control to target, and Z errors are mapped from target to control. This mapping, if uncorrected, leads to spurious detection events which will greatly degrade logical qubit performance if standard decoding methods like MWPM are used.

Recent research has focused on developing specialized decoders for transversal CNOT gates [9–16]. Our work significantly expands upon a concept briefly mentioned¹ in [9], by developing a multi-pass iterative decoder that addresses the correlated errors spread by the transversal CNOT. This decoder processes each logical patch separately in multiple passes to correct these correlated errors.

Modern qubit platforms supporting long-range connectivity make transversal CNOT gates feasible, potentially improving resource estimates for quantum computing. Our multi-pass iterative decoder demonstrates that, even under circuit-level noise and only $\mathcal{O}(1)$ code cycles, a threshold can persist without significant degradation of the logical error rate. This matches the sub-threshold logical error rate scaling of $p^{[d/2]}$ for a distance d rotated surface code.

This paper presents an analysis of our multi-pass iterative decoder for transversal CNOT gates, including its design, implementation, and performance characteristics.

Specifically, we illustrate a simple method to separately decode two different patches of surface codes that had undergone transversal CNOT operations, we assume:

1. error-free and instantaneous transversal CNOT operations and

2. standard circuit-level depolarising noise model on all other gates - SD6 [17].

We argue that the error-free transversal operation should not significantly impact the overall behavior of the logical error rates, as the realistic transversal CNOT error is expected to contribute only a marginally higher circuit-level noise error rate [2] for some hardware platforms [5]. Furthermore, the idealized transversal CNOT operation suffices for the primary purposes of our investigation here, i.e. the performance of a new approach to decoding. A detailed analysis into hardware realistic connectivity error models and its impact on the performance of the transversal CNOT operation relative to lattice surgery is in preparation [18].

Our work has significant implications for resource estimation in quantum advantage applications, where there exists a trade-off space between run times and the number of magic state distillation factories. Previous work [19–21] has assumed lattice surgery operations and demonstrated the benefits of fast code cycle times when other factors are held constant. For instance, trapped ion hardware, with code cycle times potentially three orders of magnitude slower than superconducting qubits, would traditionally require substantially more physical qubits to achieve comparable quantum advantage run times [22]. This comparison assumes the same QEC methods and physical error rates across both platforms.

However, transversal operations offer potential improvements in both the volume scaling of magic state factories and the overhead associated with parallelized magic state consumption. By leveraging these advantages along with enhanced connectivity and higher baseline fidelities on physical gates, slower hardware architectures may be able to bridge the gap in physical qubit requirements to reach runtime parity with faster but locally-connected systems. Our findings suggest that the integration of transversal CNOT gates and specialized decoding techniques could significantly impact the resource landscape across diverse quantum computing platforms, potentially altering the balance between speed and qubit count in the pursuit of quantum advantage.

¹As a long footnote.

2 The repetition code

In the following sections, we illustrate our iterative transversal CNOT decoding method with the repetition code. Due to the surface code's symmetry in suppressing Pauli-X and -Z error, it is sufficient to study the repetition code for the intuition behind our method. For simplicity, we also assume no mid-circuit errors for the visualisation of this decoding procedure. However, it should be noted that this method easily generalises to accommodate mid-circuit errors.

A distance d repetition code has code words $|\bar{0}\rangle = |0\rangle^{\otimes d}$, $|\bar{1}\rangle = |1\rangle^{\otimes d}$ and can be represented with the following stabiliser generators, \mathcal{G} , and logical operators \bar{X} and \bar{Z} :

$$\mathcal{G} = \left\langle \bigcup_{j=0}^{d-2} Z_j Z_{j+1} \right\rangle, \quad \bar{X} = \prod_{i=0}^{d-1} X_i, \quad \bar{Z} = Z_k, \quad (1)$$

where $Z_j(X_j)$ are Pauli-Z(X) operator acting on physical qubit j . If we were to repeatedly measure the stabiliser generators of this code, we can detect and correct for Pauli-X errors only. In the particular example in figure 2, we initialise the repetition code and measure the parity checks for 3 rounds before subsequently measuring all the data qubits. We can then construct the syndrome of this quantum memory experiment [19] (in this case a classical code). For $t = 1, 2$, the syndrome data/detector clicks can be constructed by the calculating the modulo 2 difference (\oplus) of the same parity check measurement separate by one step in time.

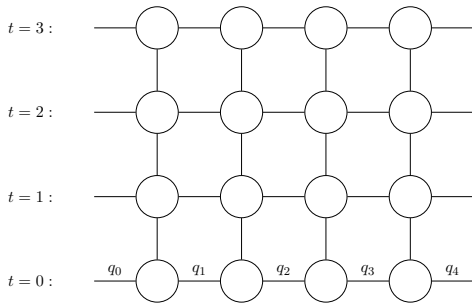


Figure 2: Distance 5 repetition code (can tolerate $\lfloor d/2 \rfloor = 2$ errors before failing) decoding graph for 4 time steps, assuming no mid-circuit errors (no diagonal edges in the syndrome graph).

Figure 2 is a syndrome graph for this quantum memory experiment with 3 rounds of syndrome extractions. The horizontal edges at each time

slices of this graph represents the qubits and vertical edges between time slices correspond to the possibility of measurement errors. Detectors are the nodes of this graph, carrying the syndrome information and will be coloured if the parity of the measurements associated with the detected differs from that expected.

3 Correlated errors

A CNOT operation inherently spread errors between the control and target modes. X errors will flow from the control to target modes (shown in equation 2) and Z errors will flow from the target to control mode (see equation 3).

$$\begin{array}{c} \boxed{X} \text{---} \bullet \text{---} \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \text{---} \boxed{X} \\ | \\ \oplus \end{array} \quad (2)$$

$$\begin{array}{c} \bullet \text{---} \boxed{X} \oplus \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \text{---} \\ | \\ \oplus \end{array} \boxed{X}$$

$$\begin{array}{c} \bullet \text{---} \boxed{Z} \\ | \\ \oplus \end{array} = \begin{array}{c} \bullet \text{---} \\ | \\ \oplus \end{array} \boxed{Z}$$

$$\begin{array}{c} \bullet \text{---} \\ | \\ \oplus \end{array} \boxed{Z} = \begin{array}{c} \bullet \text{---} \boxed{Z} \\ | \\ \oplus \end{array} \boxed{Z} \quad (3)$$

For simplicity, in this discussion we focus on the X errors only and note that by symmetry, the argument is valid for the Z errors with the control and target modes reversed. The transversal CNOT will deterministically map X errors from the control to target patch. If this propagated error is not corrected for, it will cause spurious detection events appearing on the target node right after the transversal CNOT. Ultimately, this has the overall effect of a degraded logical error rate. However, if we can first decode the control patch, and have a suitable guess for the locations of the X errors, we can use that to cancel out the propagated X error on the target patch. The motivation behind our decoder is to use a conventional memory experiment decoder for a single logical qubit memory experiment, which decodes each logical qubit separately, updates the syndrome data and Pauli-frame of each individual logical qubit and then propagates that information before decoding again iteratively. This process repeats until the Pauli-frames converge to a stable state. The number of iterations required is a function of the circuit structure, where simple circuits only ever require one iteration.

4 Iterative decoder for the repetition code

We shall illustrate our approach with two examples concerning the repetition code. The first example (Example 1) involves a single transversal CNOT, requiring only a single-pass decoder, whereas the multiple CNOT version (Example 2) requires multiple rounds of iterative decoding.

We use the following notation and conventions to separate different detectors and edges by colour.

- “Natural” errors (edges) and detectors (nodes) are coloured **red**, $\text{---}\bullet\text{---}$, they are the errored edges and detectors that occur genuinely, and are not due to the propagation effects of the CNOT.
- Propagated errors (edges) and detectors (nodes) are coloured **brown**, $\text{---}\bullet\text{---}$, they are the errored edges and detectors that are caused solely by the propagation of errors due to the CNOT.
- Decoded matched edges are coloured **blue**, they are the edges returned by the MWPM decoding algorithm, $\text{---}\bullet\text{---}$, in this case the blue decoded edge had correctly corrected for the red corrupted edge.
- The stored propagated edges are coloured **green**, they are the recorded edges *perceived* to be propagated by the error spreading effects of the CNOT, $\text{---}\bullet\text{---}$, in this case the green propagated edge had correctly corrected for the brown propagated edge.

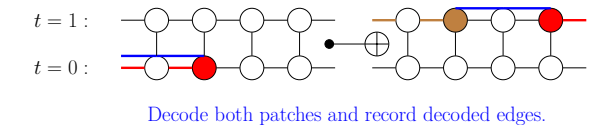
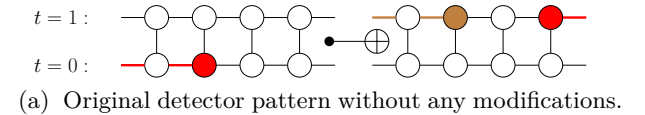
The core part of the method relies on propagating Pauli errors and detection events between the patches. For example with X errors, the Pauli frame of the control patch directly before the CNOT operation is determined, and propagated in the round following the CNOT into a separate stored list corresponding to the target patch. Next the associated detectors of the control Pauli Patch are propagated modulo 2 into the detector pattern of the target patch. The procedure acts to recover the detector pattern corresponding to only the natural errors while the Pauli errors determined to be propagated are stored separately. The logical observable is determined by combining the detector pattern decoded Pauli frame with the propagated Pauli frame.

Each CNOT is addressed in this manner sequentially. For more complex circuits, like those with alternating directions of CNOT operations, more than one iteration may be required, where each iteration involves addressing every CNOT in the circuit, first undoing the impact of this particular CNOT from the prior iteration, and then again performing the propagation of Pauli errors and detectors. The termination clause is when the combined Pauli frame for each logical qubit converges to a stable state, i.e. equality between this iteration and the prior. For circuits with two alternating direction CNOT operations, at most two iterations are ever required. A flowchart illustrating the iterative method is found in appendix A.

4.1 Example 1: Single pass decoder

In figure 3s and 4s, chronologically in time from figure 3a to figure 4c, we show a method to modify syndromes and record propagated edges to decode a single logical transversal CNOT acting between two logical qubits (left and right).

Firstly, in figure 3a, we see a detector pattern, and have annotated the propagated detectors and edges with the brown nodes and edges respectively, and also labelled the errored qubits with red edges for illustrative purposes. However in an actual experiment, this information is not available.

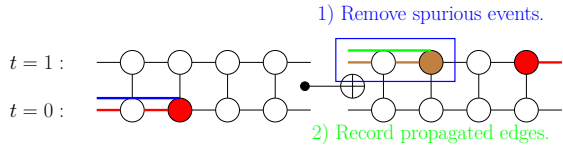


(b) Decode both logical qubits and record the blue decoded edges, we can see that we have arrived at incorrectly decoded edges.

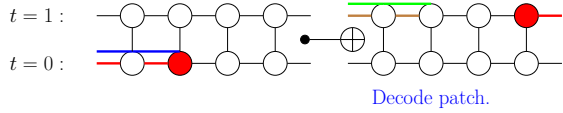
Figure 3: Figures 3a and 3b show the decoding procedure before the propagation of the Pauli-frame from the left to the right logical qubit in order to reduce the number of spurious detectors resulting from the transversal logical CNOT. We can observe that in this case, we have decoded and arrived at incorrect edges.

In figure 3b, we decode the left and right logical qubits and record the blue decoded edges in figure 3b. In this case, decoding has chosen some edges

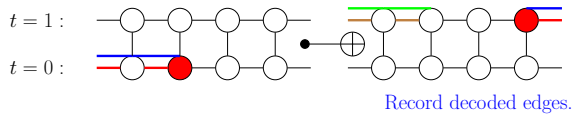
incorrectly.



(a) The Pauli-frame is propagated over to the right logical qubit, removing spurious events and with green propagated edges recorded.



(b) The right logical qubit is then decoded via MWPM.



(c) The decoding from MWPM, leading to the blue coloured edges on the right logical qubit.

Figure 4: Steps in figure 4a to figure 4c show the decoding procedure required to decode and propagate the Pauli-frame from the left logical qubit to the right logical qubit.

In order to decode correctly, we reset the blue decoded edges on the right patch, then, the predicted Pauli-frame for the left logical qubit is propagated to the right logical qubit in figure 4b, removing spurious detection events and recording the propagated edges in green at $t = 1$. We then proceed to decode the right logical qubit with MWPM and record its decoded edges in figure 4c.

With this syndrome modification and Pauli-frame propagation procedure, we are able to correctly identify the “natural errors” and the propagated errors with all the spurious detection events removed. For an instantaneous and error-free transversal CNOT operation, the logical error rate of failure after the application of a transversal CNOT should be as close as possible to the memory experiment case as possible. This can be interpreted as an enhanced memory experiment with a transversal CNOT sandwiched in between patches.

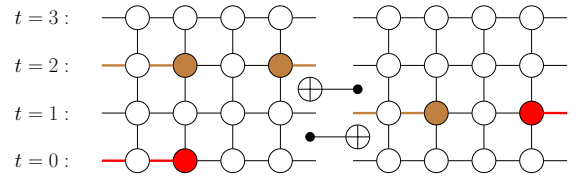
4.2 Example 2: Multi-pass decoder

We will now show a more complicated example which includes two alternating transversal

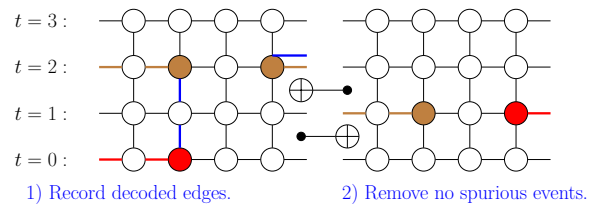
CNOTs. Multiple iterations of frame propagation are required, and we see that this naturally follows from the mapping behaviour of the transversal CNOT. In the case of two alternating transversal CNOT’s, we observe that an X error spreading from control to target in the first instance will be reflected back after the second transversal CNOT. If not correctly decoded, this would lead to an amplification of errors that will severely affect the overall logical error rate.

4.2.1 1st iteration

In the first iteration, we follow similar methods outlined in Example 1. The detector pattern from experiment can be seen in figure 5a, the series of two alternating transversal CNOTs with a round of syndrome extraction in between leads to the brown propagated edges and nodes. We first decode the left logical qubit and record its blue decoded edges in 5b. We then propagate the Pauli-frame at $t = 0$ in the left patch to the right patch to remove spurious events, in this case no data (spatial) edges are matched in the left patch at $t = 0$, hence no spurious events are propagated over to the right. These are the incorrectly decoded edges on the left, we shall see how iterating the process in the 2nd iteration solves this.



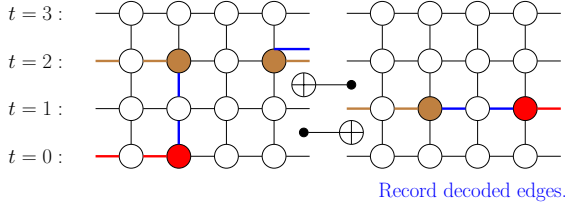
(a) Original syndrome detector signal without any modifications for Example 2 with two alternating CNOTs.



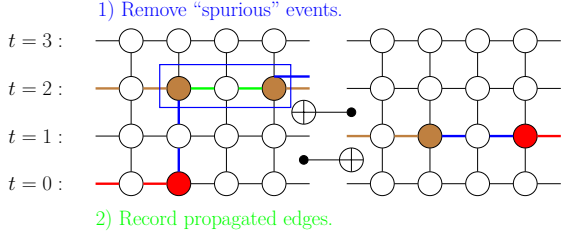
(b) We decode the left patch and propagate the Pauli-frame to the right patch, in this case, no errors were detected, so no spurious events or propagation of green edges were formed on the right patch.

Figure 5: The original syndrome detection pattern of the two alternating CNOT example, following by the first iteration’s first Pauli-frame propagation through the first CNOT.

Then, we decode the logical qubit on the right, leading to the blue decoded edges in figure 6a.



(a) Decode the logical qubit on the right and record its decoded blue coloured edges.



(b) Using the Pauli-frame of the right patch up to $t = 1$, propagated the frame over to the left, record the propagated edges and remove the spurious events. Note that the propagated edges and the removed spurious event are incorrect and will be fixed in the second iteration.

Figure 6: The procedure outlining the first iteration's second Pauli-frame propagation through the second CNOT.

Again these matched edges are incorrect, we will still propagate the frame over from the right at $t = 1$ to the left at $t = 2$, removing spurious events and recording the green decoded edges (figure 6b). At the end of the first iteration, we arrive at a series of matched edges as shown in figure 7. The pattern of detection events has changed, and this triggers another round of decoding.

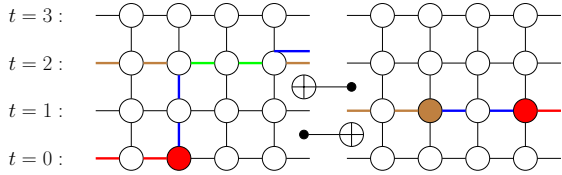
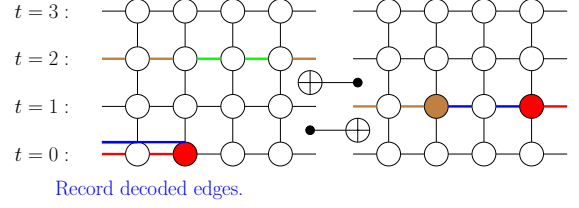


Figure 7: The decoded and propagated edge record after the first iteration, note that we have removed two spurious detection events but at this stage introduced a logical error. This will be rectified in the second iteration.

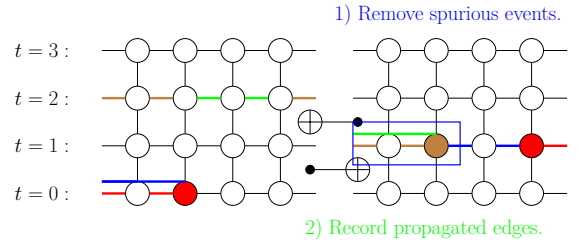
4.2.2 2nd iteration

It's clear that a single iteration of frame propagation will not be sufficient to find good decoded edges in this case. In the second iteration, we reset the magneta edges on the left qubit and decode again leading to new blue edges at $t = 0$

(figure 8a). Propagate the Pauli-frame over to $t = 1$ of the logical qubit on the right, removing the spurious events and record the newly propagated green edges (figure 8b).



(a) Reset the decoded blue edges on the left logical qubit and then re-decode.



(b) Propagate the Pauli-frame up till $t = 0$ from the left logical qubit from figure 8a to the right logical qubit at $t = 1$.

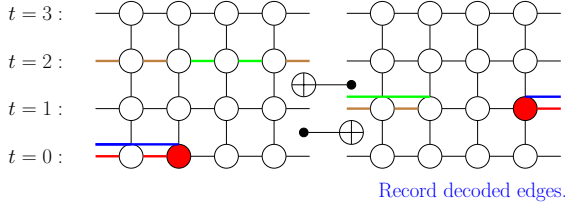
Figure 8: In the second iteration of the Pauli-frame through the first CNOT, we can see that we have correctly identified some more of the errored edges.

Next, the right logical qubit is decoded, leading to blue edges at $t = 1$ in figure 9a. Before propagating the frame from $t = 1$ on the right to left, we need to remove the previously propagated edges and spurious events at $t = 2$ on the left from the first iteration (see figure 9b). We then reset the decoded blue edges on the left, before following the conventional frame propagation shown in figure 10a to 10b.

After the second iteration, we have correctly identified all the propagated edges and errored edges. Please refer to figure 15 in appendix A for an outline of the process.

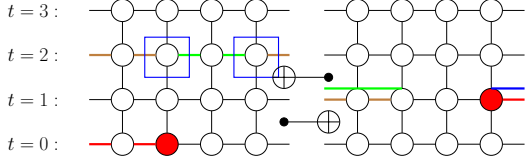
5 Surface code simulations

The procedure illustrated in the previous sections can be easily generalised to patches of surface code after applications of transversal CNOT operations. We aim to study and simulate the minimally interesting cases. Firstly, we characterise a transversal CNOT operation memory experiment with one round of syndrome extraction before and after it. Then we study the next minimally interesting case with 2 alternating transversal CNOTs,



(a) Reset the decoded blue edges on the right patch and re-decode.

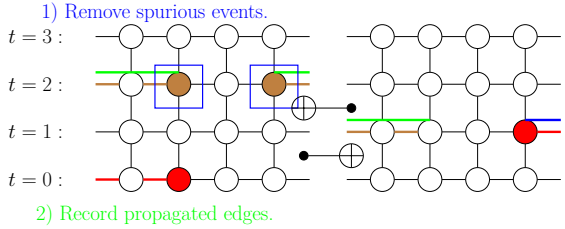
- 1) Remove previously propagated edges at $t = 2$.
- 2) Remove previously modified spurious events.



- 3) Reset decoded blue edges.

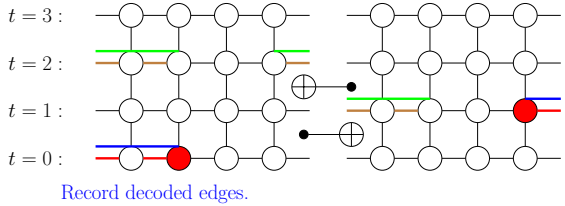
(b) Firstly, we remove the previously modified spurious events and propagated green edge at $t = 2$ on the left logical qubit from the previous iteration.

Figure 9: Crucial step in the second iteration of Pauli-frame propagation through the second CNOT, we need to undo detector modifications and propagated edges done in the first iteration.



- 2) Record propagated edges.

(a) Re-propagate the green edges from the right patch to the left patch at $t = 2$, removing new spurious events and record new propagated edges.



- Record decoded edges.

(b) Finally, re-decode the left logical qubit and we have finally arrived at a pattern of detectors and set of Pauli frames that is stable warranting no further iteration.

Figure 10: Final frame propagation leading to all the correct corrupted edges being identified.

with the iterative procedure and varying number of rounds of syndrome extraction between the CNOTs. All the logical error rates show close agreement with the memory experiment logical error rates.

We use Stim [23] to construct the detector er-

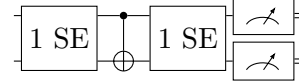
ror model and sample from the syndrome extraction circuits undergoing the SD6 circuit-level noise model. We then subsequently use PyMatching to decode each logical qubits' syndromes separately with an equivalent single patch memory experiment matching graph.

We represent arbitrary number of rounds of syndrome extraction that is performed as follows:

$$|\bar{0}\rangle \xrightarrow{x \text{ SE}} \hat{U} \xrightarrow{y \text{ SE}} \text{Measurement}, \quad (4)$$

the quantum circuit in equation 4 shows one logical qubit undergoing x rounds of syndrome extraction(s), before a unitary operation \hat{U} , followed by another y rounds of syndrome extraction(s) before reading it out in the computational basis.

5.1 Single transversal CNOT



A single CNOT circuit with one round of syndrome extraction before and after, followed by measurement of the logical observables.

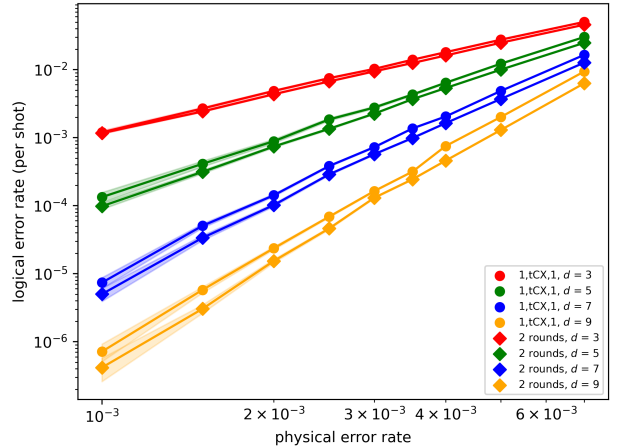
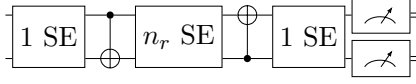


Figure 11: Surface code simulations with a single transversal CNOT sandwich between one round of syndrome extraction before and after before reading out all the data qubits. At every distance, the transversal CNOT logical error rate matches the memory experiment results closely.

In figure 11 we analyse the single CNOT case, where only one iteration is ever required. We show close agreement between the CNOT circuit with propagating decoder versus the memory equivalent, which corresponds to two logical

qubits with no CNOT operations and the same number of total rounds, i.e. two rounds, decoded with standard independent MWPM. This represents a lower-bound on performance quality. The scaling relationship with physical error rate is preserved.

5.2 Two alternating transversal CNOTs



A two-alternating CNOT circuit with one round of syndrome extraction before, a variable number of rounds between, and one after, followed by measurement of the logical observables.

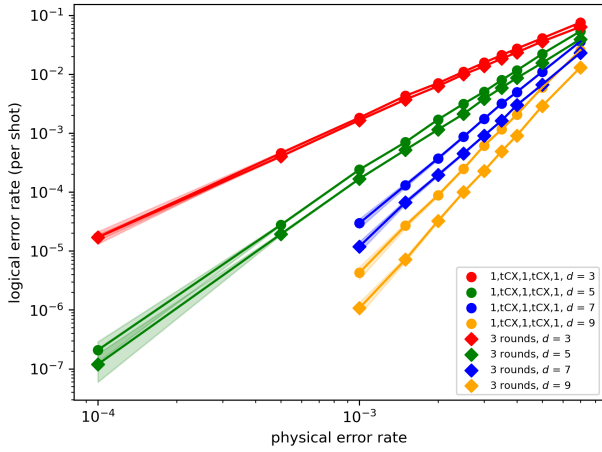


Figure 12: Surface code simulations with two alternating transversal CNOTs, one round of syndrome extraction between transversal CNOTs.

In this section we investigate the performance of the iterative CNOT decoder as a function of the number of rounds of syndrome extraction between successive alternating transversal CNOT operations. We show that in figure 12 where only a single round of error correction is performed between, we still maintain the same scaling relationship with physical error rate, and a comparable threshold. We note that the discrepancy between the transversal CNOT case versus memory appears to grow with increasing code distance.

For figure 13 we instead include two rounds of syndrome extraction, and now this growing discrepancy with increasing distance is no longer evident. This difference may be attributable to the space-time edges that span across the syndrome extraction rounds where there is a degen-

erate choice of round occurrence in relationship to the location of the CNOT operations (discovered independently by [15]). The impact of this is mitigated by moving to two rounds and beyond.

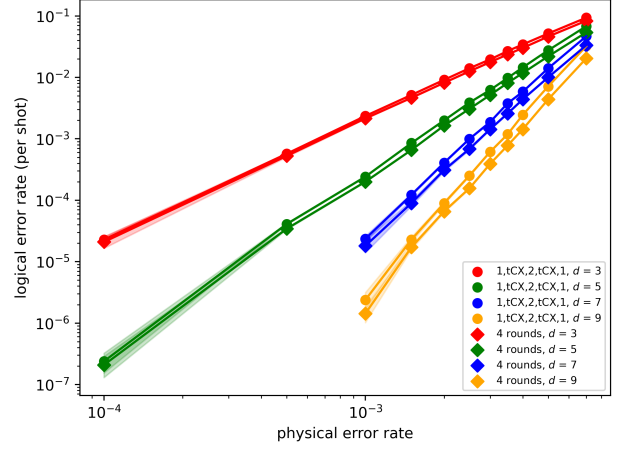


Figure 13: Surface code simulations with two alternating transversal CNOTs, two rounds of syndrome extraction between transversal CNOTs.

In figure 14 we plot for the three round case, and see the same overall scaling relationships as the two round case. Of course, the ideal performance of a transversal CNOT decoder would be the minimal time complexity with the minimal final error rate. Moving from one round to two rounds improved our final logical error rate, e.g. for distance 9, physical error rate $1E-3$, the final error rate improved from $4.3E-6$ to $2.4E-6$. In contrast, moving from two rounds to three rounds actually slightly worsened our final error rate, i.e. from $2.4E-6$ to $2.6E-6$. This is because there are opposing forces, i.e. the iterative propagating decoder has some additional resiliency to time component edges with increasing round count separating CNOT operations, but each additional round added itself has some additional contribution of error. For all of our investigations so far, the minimal error case is found at two rounds of QEC separating the transversal CNOT operations. The optimal choice will ultimately include the desired target error rate, and a minimization of total volume, i.e. the distance required, and the number of rounds between CNOT operations to reach the target.

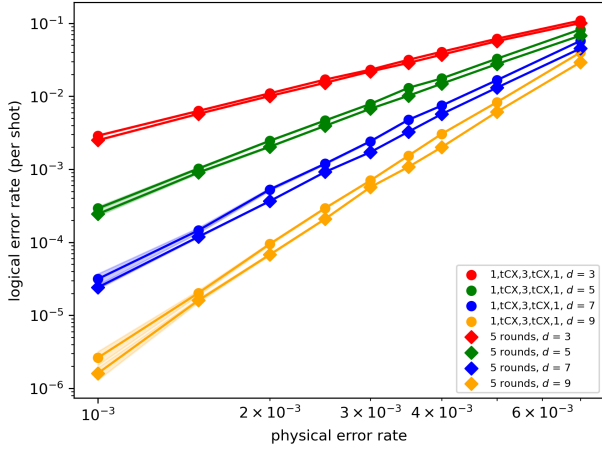


Figure 14: Surface code simulations with two alternating transversal CNOTs, three rounds of syndrome extraction between transversal CNOTs.

6 Discussions and outlook

Our iterative decoder addresses the challenge of managing correlated errors resulting from transversal CNOT operations. The relevance of this work is supported by the feasibility of long-range two-qubit gates between physical qubits, as demonstrated in hardware platforms such as trapped ions [5, 10, 24].

A key outcome of this study is the finding that two rounds of QEC between alternating transversal CNOTs lead to the optimal final error rate, independent of the code distance. This result confirms that the time complexity of the transversal CNOT is $\mathcal{O}(1)$. Our results show that the iterative decoder maintains the expected scaling of logical error rates with code distance and physical error rates under typical circuit-level noise models.

The practical implications of these findings are significant for particular quantum computing platforms. Incorporating transversal CNOT gates and the specialized decoding technique can reduce the physical qubit overhead required for fault-tolerant operations. For example, the lattice surgery CNOT implementation has a volume complexity of $\mathcal{O}(d^3)$, whereas the transversal CNOT has a volume complexity of $\mathcal{O}(d^2)$. The integration of transversal CNOTs would enhance the efficiency of magic state distillation processes and parallelized magic state consumption, thus reducing the overall resources required for quantum advantage applications in the fault-tolerant regime. This has the potential to bridge the

gap in performance and resource requirements between different hardware architectures, such as superconducting qubits and trapped ions, making it feasible for slower, more connected systems to achieve comparable operational efficiency to their faster, locally-connected counterparts.

Further exploration of hardware-specific error models and their impact on the performance of the transversal CNOT is essential. A detailed analysis of the performance of the transversal CNOT with hardware-realistic connectivity error models, and its contrast to the lattice surgery alternative, is in preparation [18]. Additionally, an extension to the quantum advantage resource estimation process to include the benefits of transversal CNOT operations is also in preparation [25]. In this latter upcoming work, physical qubit requirements to reach target runtime will be estimated for different connectivity and code cycle assumptions.

In summary, the multi-pass iterative decoder for transversal CNOT gates offers an approach to utilize the benefits associated with high-fidelity long-range qubit connectivity. By optimizing the number of QEC rounds required and the resulting error rate, this method can improve the performance of quantum hardware with such connectivity available.

7 Acknowledgments

We want to thank Niel De Beudrap for his insights into analysing the iterative nature of the transversal CNOT decoder. We also acknowledge discussions with Craig Gidney, Michael Newman and Lucy J. Robson. We also want to thank all our colleagues at Universal Quantum Ltd for their continued support. Part of the numerical results were presented at the ICAP 2024 Satellite Conference (Brighton, UK) on 22 July 2024. Winfried K. Hensinger acknowledges support from Innovate UK (project number 10004857) and the University of Sussex.

References

- [1] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation, 2009. URL <https://arxiv.org/abs/0904.2557>.

- [2] Oscar Higgott and Craig Gidney. Py-matching v2. <https://github.com/oscarhiggott/PyMatching>, 2022.
- [3] Oscar Higgott and Craig Gidney. Sparse blossom: correcting a million errors per core second with minimum-weight matching, 2023. URL <https://arxiv.org/abs/2303.15933>.
- [4] Yue Wu and Lin Zhong. Fusion blossom: Fast mwpm decoders for qec, 2023. URL <https://arxiv.org/abs/2305.08307>.
- [5] Mark Webber, Steven Herbert, Sebastian Weidt, and Winfried K. Hensinger. Efficient qubit routing for a globally connected trapped ion quantum computer, 2020. URL <https://arxiv.org/abs/2002.12782>.
- [6] Dominic Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, December 2012. ISSN 1367-2630. DOI: 10.1088/1367-2630/14/12/123011. URL <http://dx.doi.org/10.1088/1367-2630/14/12/123011>.
- [7] Avimita Chatterjee, Subrata Das, and Swaroop Ghosh. Lattice surgery for dummies, 2024. URL <https://arxiv.org/abs/2404.13202>.
- [8] Austin G. Fowler. Time-optimal quantum computation, 2013. URL <https://arxiv.org/abs/1210.4626>.
- [9] Michael E. Beverland, Aleksander Kubica, and Krysta M. Svore. Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes. *PRX Quantum*, 2:020341, Jun 2021. DOI: 10.1103/PRXQuantum.2.020341. URL <https://link.aps.org/doi/10.1103/PRXQuantum.2.020341>.
- [10] Dolev Bluvstein, Simon J. Evered, Alexandra A. Geim, Sophie H. Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, J. Pablo Bonilla Ataides, Nishad Maskara, Iris Cong, Xun Gao, Pedro Sales Rodriguez, Thomas Karolyshyn, Giulia Semeghini, Michael J. Gullans, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997):58–65, Feb 2024. ISSN 1476-4687. DOI: 10.1038/s41586-023-06927-3. URL <https://doi.org/10.1038/s41586-023-06927-3>.
- [11] Madelyn Cain, Chen Zhao, Hengyun Zhou, Nadine Meister, J. Pablo Bonilla Ataides, Arthur Jaffe, Dolev Bluvstein, and Mikhail D. Lukin. Correlated decoding of logical algorithms with transversal gates, 2024.
- [12] Hengyun Zhou, Chen Zhao, Madelyn Cain, Dolev Bluvstein, Casey Duckering, Hong-Ye Hu, Sheng-Tao Wang, Aleksander Kubica, and Mikhail D. Lukin. Algorithmic fault tolerance for fast quantum computing, 2024. URL <https://arxiv.org/abs/2406.17653>.
- [13] Dominik Hangleiter, Marcin Kalinowski, Dolev Bluvstein, Madelyn Cain, Nishad Maskara, Xun Gao, Aleksander Kubica, Mikhail D. Lukin, and Michael J. Gullans. Fault-tolerant compiling of classically hard iqp circuits on hypercubes, 2024. URL <https://arxiv.org/abs/2404.19005>.
- [14] Younghun Kim, Martin Sevior, and Muhammad Usman. Transversal cnot gate with multi-cycle error correction, 2024.
- [15] Kaavya Sahay. Fault-tolerant logical gates for neutral-atom surface codes, 2024. URL <https://www.youtube.com/watch?v=2vcjcmxSXCy>.
- [16] Bence Hetényi and James R. Wootton. Creating entangled logical qubits in the heavy-hex lattice with topological codes, 2024.
- [17] Victor V. Albert and Philippe Faist, editors. *The Error Correction Zoo*. 2024. URL <https://errorcorrectionzoo.org/>.
- [18] Mark Webber and Universal Quantum team et. al. Transversal cnot versus lattice surgery as a function of qubit connectivity costs. In preparation.
- [19] Austin G. Fowler and Craig Gidney. Low overhead quantum computation using lattice surgery, 2019. URL <https://arxiv.org/abs/1808.06709>.
- [20] Craig Gidney and Austin G. Fowler. Efficient magic state factories with a catalyzed CCZ to 2T transformation. *Quantum*, 3:135, April 2019. ISSN 2521-327X. DOI: 10.22331/q-2019-04-30-135. URL <http://dx.doi.org/10.22331/q-2019-04-30-135>.
- [21] Daniel Bochen Tan, Murphy Yuezhen Niu,

- and Craig Gidney. A SAT scalpel for lattice surgery: Representation and synthesis of subroutines for surface-code fault-tolerant quantum computing, 2024. URL <https://arxiv.org/abs/2404.18369>.
- [22] Mark Webber, Vincent Elfving, Sebastian Weidt, and Winfried K. Hensinger. The impact of hardware specifications on reaching quantum advantage in the fault tolerant regime. *AVS Quantum Science*, 4 (1), January 2022. ISSN 2639-0213. DOI: [10.1116/5.0073075](https://doi.org/10.1116/5.0073075). URL <http://dx.doi.org/10.1116/5.0073075>.
- [23] Craig Gidney. Stim: a fast stabilizer circuit simulator. *Quantum*, 5:497, July 2021. ISSN 2521-327X. DOI: [10.22331/q-2021-07-06-497](https://doi.org/10.22331/q-2021-07-06-497). URL <https://doi.org/10.22331/q-2021-07-06-497>.
- [24] 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, 2022. URL <https://arxiv.org/abs/2208.01863>.
- [25] Mark Webber and Universal Quantum team et. al. The impact of high-fidelity long-range connectivity on reaching quantum advantage in the fault tolerant regime. In preparation.

A Iterative decoding flowchart

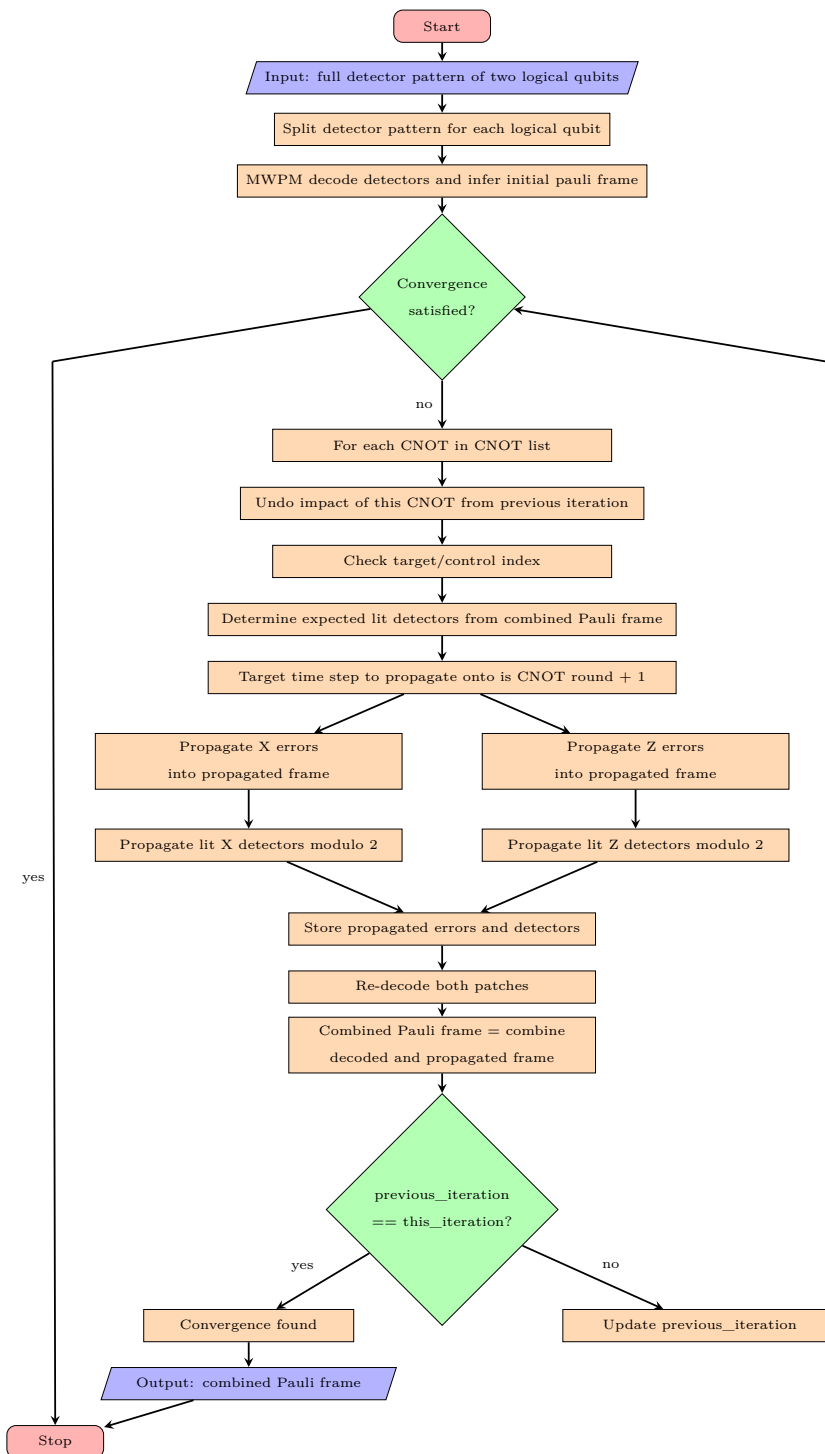


Figure 15: This flowchart outlines the procedure and termination of iteration required to iteratively decode multiple alternating transversal CNOTs between two logical qubits.