Magic state injection on the rotated surface code

Lingling Lao*
laolinglingrolls@gmail.com
University College London
London, United Kingdom

Ben Criger
ben.criger@cambridgequantum.com
Cambridge Quantum
Cambridge, United Kingdom

ABSTRACT

Fault-tolerant quantum computing based on the surface code requires magic states to achieve universality. The initially prepared magic states have low fidelity and need to be purified by a costly procedure called magic state distillation. The high spatial-temporal cost for distillation can be reduced by improving the fidelity of initial magic states. Prior works have investigated state injection approaches on the regular surface code. In this work, we propose a magic state injection method for the rotated surface code that requires fewer physical qubits to encode one logical qubit. Analytical results show that our method could achieve lower logical error rates than the state-of-the-art approach for the regular surface code in [1]. When both the two-qubit gate and single-qubit operation error rates are p, the new method results in a logical error rate to $\frac{34p}{15}$ compared to $\frac{46p}{15}$ in [1]. This fidelity improvement and fewer qubit encoding requirement would help reduce the overhead of fault-tolerant quantum computation, which is crucial for practical implementation in the near term.

CCS CONCEPTS

• **Hardware** → **Quantum error correction and fault tolerance**.

KEYWORDS

Quantum computing, quantum error correction, fault tolerance, surface code, magic state

ACM Reference Format:
Lingling Lao and Ben Criger. 2022. Magic state injection on the rotated surface code. In *19th ACM International Conference on Computing Frontiers (CF'22), May 17–19, 2022, Torino, Italy.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3528416.3530237

1 INTRODUCTION

Quantum computing can solve a large and growing class of problems in polynomial time which are not known to be polynomially-solvable using classical computers [2–6]. The main obstacle for reliably implementing large-scale quantum computation is noise, which is continuous. Quantum error correction (QEC) can reduce the strength of this noise, leaving us with a discrete process at the logical level. Nevertheless, the discreteness of logical operations makes it difficult to do universal computation. For example, it is impossible to carry out universal computation at the logical level using only transversal gates [7, 8]. To achieve universality, one can apply gate teleportation in which ancillary logical qubits are prepared in specific states called magic states (see Figure 1). However, the preparation of magic states, also called *state injection*, is not fault-tolerant and needs to be purified by a state distillation procedure [9–15]. To circumvent the potential high overhead of magic state distillation, alternative approaches have been proposed to implement universal fault-tolerant (FT) quantum computation [16–26].

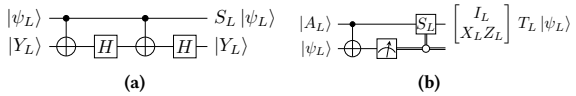


Figure 1: Circuits to teleport (a) the S and (b) the T gates, where the magic states $|Y_L\rangle = (|0\rangle + i|1\rangle)/\sqrt{2}$ and $|A_L\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$.

The surface code is one of the most promising QEC codes due to its high tolerance for errors (its error threshold is $\sim$ 1% [27]) and compatibility with the 2D nearest-neighbour (NN) connectivity constraint in many current quantum technologies [28, 29]. In planar surface codes, Pauli gates are implemented transversally and a CNOT gate in a 2D architecture can be realised by lattice surgery [30, 31]. The S and T gates on surface codes can be implemented using magic states that can be injected by measurement-based operations [1, 30–32]. Figure 2 (a) shows an injection approach where all the stabilisers involving the injected data qubit will give random measurement results, and a single X or Z error on this qubit is undetectable and causes a logical error [1].

In order to ensure fault tolerance, one can use distillation to purify the injected magic states [10, 11, 27], that is, distilling one higher-fidelity magic state from many lower-fidelity states. Moreover, magic state distillation is a non-deterministic procedure, it must be repeated until the measurement results indicate that a state is successfully purified. The success probability of distillation depends on the logical error rate p_L on input states. Once it succeeds, the infidelity of the accepted state will be suppressed. For example, the magic state infidelity is suppressed to $O(p_L^3)$ by using the 15-to-1 distillation protocol in [10]. Multiple rounds of distillation may be required to achieve the desired infidelity $O(p_L^n)$, so magic state distillation can be extremely qubit- and time-consuming.

Improvements in the fidelity of magic state injection will help reduce the spatial-temporal cost for distillation [1, 33], decreasing the overhead for fault-tolerant quantum computing. Prior works

*Corresponding author

CF'22, May 17–19, 2022, Torino, Italy

ACM ISBN 978-1-4503-9338-6/22/05…$15.00
https://doi.org/10.1145/3528416.3530237

have proposed magic state injection methods on the regular surface code [1, 30, 31]. The rotated surface code requires around half of the amount of physical qubits to encode one logical qubit but its injection method has not been well investigated. To this end, this work presents an injection approach for the rotated surface code and analyzes its logical error rates to leading order in the physical error rate. Analytical results show that our injection approach on the rotated surface code can achieve lower logical error rates than the state-of-the-art approach for the regular surface code in [1] under depolarising errors. We further numerically simulate the proposed approach using the stabiliser formalism [34] to verify the analytical results.

This paper is organised as follows. We first introduce the surface code in Section 2. Then we present the magic state injection approach for the regular surface code [1] and the proposed injection approach for the rotated surface code in Section 3. We discuss the analytical and numerical results in Section 4 and conclude the paper in Section 5.

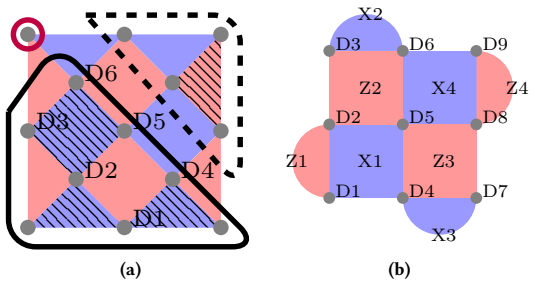


Figure 2: Qubit layouts of the surface code. Data qubits (gray dots) and ancillas (not shown) are on the vertices and the faces of the plaquettes. Blue and red plaquettes represent X and Z stabilisers, respectively. (a) A distance-3 logical qubit based on the regular surface code. For the state injection approach in [1], the circled qubit in the upper left corner is prepared in the magic state and the data qubits within the solid (dashed) outline are prepared in the $|+\rangle$ ($|0\rangle$) state. The hatched stabilisers are valid both before and after the first round of stabiliser measurements. (b) A distance-3 logical qubit based on the rotated surface code. This qubit labeling applies to all distance-3 rotated lattices in this manuscript.

2 SURFACE CODE

In this section, we briefly introduce the basics of the surface code. We refer to [27, 30–32, 35] for more details on the fault-tolerant implementation of different logical operations on the surface code.

The surface code is a stabiliser code [36]. A stabiliser code is defined using a subgroup S of the Pauli group $P_n = \{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n}$ and can encode k logical qubits into n physical qubits. The states of all data qubits are stabilised by S and form a codespace $T = \{|\psi\rangle \mid M|\psi\rangle = |\psi\rangle, |\psi\rangle \in (\mathbb{C}^2)^{\otimes n}, \forall M \in S\}$. S must be Abelian, so that any two operators in S commute with each other and $-I$ must not be in S. Moreover, the logical X and Z operators anti-commute with each other, but commute with all the stabilisers. A stabiliser group can be described by only $n-k$ linearly independent generators which are a subset of stabilisers that can generate any stabiliser in S by multiplication. The errors on data qubits can be detected by measuring stabilisers, that is, if an error anti-commutes with a stabiliser, then the measurement syndrome of this stabiliser will be -1. The number of errors that can be corrected by a QEC code is determined by the *code distance* d. If the physical error probability is below an error threshold, then the logical error rate can be suppressed to arbitrarily small levels by increasing the code distance.

Figure 2 shows the qubit layouts of two types of surface codes, which are the regular surface code and the rotated surface code. Each lattice encodes one logical qubit. In all qubit layouts in this manuscript, data qubits encoding logical information and ancilla qubits used for measuring stabilisers are located on the vertices and faces of each plaquette, respectively. Ancilla qubits will not be shown for readability. Blue plaquettes represent X stabilisers for detecting Z errors and red plaquettes represent Z stabilisers for detecting X errors. For example, in Figure 2 (a), a single-qubit Z error on data qubit D3 will be detected when measuring the stabiliser $X_{\mathrm{D2}}X_{\mathrm{D3}}X_{\mathrm{D5}}X_{\mathrm{D6}}$. The qubit labeling in Figure 2 (b) applies to all other distance-3 rotated lattices in this manuscript. For square lattices, a distance-d regular surface code requires $2d^2 - 2d + 1$ data qubits to encode one logical qubit. In comparison, a distance-d rotated surface code only needs d^2 data qubits. The circuits that simultaneously measure X and Z stabilisers for both codes are shown in Figure 4 (For brevity, we only show the measurement circuits of a few stabilisers). In principle, the order of CNOT gates in each circuit would work for both codes. Here different orders are chosen to minimise logical errors.

Besides quantum memory, a set of logical operations need to be implemented to perform universal computation. One popular gate set consists of the Hadamard (H), S, T, and CNOT gates. In the surface code, the H and CNOT gates can be implemented fault-tolerantly by using techniques such as lattice surgery and code deformation [30, 31, 35]. However, the S and T gates cannot be directly realised in a fault-tolerant way and require magic states as shown in Figure 1. As mentioned in the previous section, magic state preparation is very expensive in terms of both spatial and temporal costs. The required resource depends on the fidelity of initially injected states. Improving magic state injection will help increase the success probability, reduce the number of state distillation rounds and in turn lower the overall cost, which is the focus of this work.

3 STATE INJECTION APPROACHES

In this section, we first introduce the state injection approach for the regular surface code proposed in [1]. We then adapt it to the rotated surface code and analyse its logical error rate. Afterwards, we present our injection method on the rotated surface code which can achieve higher fidelity with fewer data qubits.

3.1 Post-selection protocol

As mentioned previously, a single gate error could lead to a logical error on the injected state. Previous work proposed by Li [1] has shown that some of these errors can be detected at runtime, and

postselected out, suppressing logical error rates. The post-selection protocol in [1] will be used for all injection approaches in this manuscript, it works as follows:

(1) **Initialisation:** First, all the data qubits on the distance-d lattice are initialised in the required states ($|+\rangle$, $|0\rangle$, or the magic state). Figure 2 (a) shows an example of the injection approach in [1], where the circled data qubit is prepared in the magic state. The data qubits in the solid and dashed blocks are initialised in the $|+\rangle$ and $|0\rangle$ states, respectively.
(2) **Stabiliser measurement:** Two rounds of stabiliser measurement are performed on the lattice. We note that some stabilisers of the code are already satisfied in the initialisation step (e.g., the hatched stabilisers in Figure 2 (a)). In the absence of errors, the measurement outcomes of these initially satisfied stabilisers should be +1. In contrast, the other stabilisers will have random measurement outcomes in the first round, because they anti-commute with the initialisation step. Furthermore, both rounds should have the same outcome for each stabiliser.
(3) **Post-selection:** However, errors will affect the stabiliser measurement outcomes. One can post-select the injected states to suppress logical error rates. An injected logical qubit will be discarded if any of the following measurement mismatches occur: 1) The outcomes of the first round of stabiliser measurements do not coincide with the satisfied stabilisers in the initialisation step; 2) the outcomes of the second round of stabiliser measurements are different from those in the first round.

These mismatches are caused by errors, therefore discarding such runs will help improve the fidelity of the injected magic states. To analyse the logical error rates of an injection approach, we only need to consider errors that will not cause syndrome mismatches but will lead to logical errors. Our analysis will be based on the circuit-level noise model as in [1]. It inserts errors with probability $O(p)$ after each operation of the preparation circuit (including data qubit initialisation and two rounds of stabiliser measurement) as follows: each single-qubit gate is followed by a X, Y, or Z with probability $p_1/3$, each two-qubit gate is followed by an element of $\{I, X, Y, Z\}^{\otimes 2}\backslash\{II\}$ with probability $p_2/15$, each qubit initialisation and measurement is flipped with probability p_{IN} and p_{M} respectively. In this analysis, we will use the stabiliser measurement circuits in [37] for the rotated surface code and only consider the single-operation errors that contribute to the logical error rates.

3.2 Injection on the regular surface code

Figure 2 (a) shows the state injection approach for the regular surface code presented in [1]. We will label this approach 'YL'. The above post-selection protocol is applied to suppress logical error rates. The order of CNOT gates in the stabiliser measurement circuit has a vital impact on the logical error rates of the surface code. Carefully ordering these CNOT gates can help post-select out some errors, further improving fidelity. The optimised circuit in Figure 4 (a) is used in this method, where the CNOT gates for measuring X and Z stabilisers are partially serialised and all CNOT gates are performed in six timesteps. Gate errors on both the circled data qubit (called the 'magic qubit') and its horizontally neighbouring data qubit could induce logical errors. Detailed error analysis can be found in [1]. Assuming the circuit-level noise model, the logical error rate of a distance-d magic state implemented by the YL approach is

$$p_{\text{YL}} = \frac{2}{5}p_2 + 2p_{\text{IN}} + \frac{2}{3}p_1 + O(p^2). \quad (1)$$

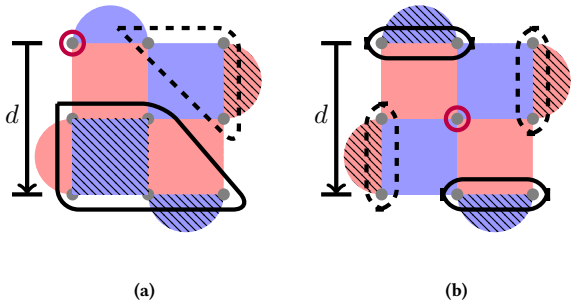
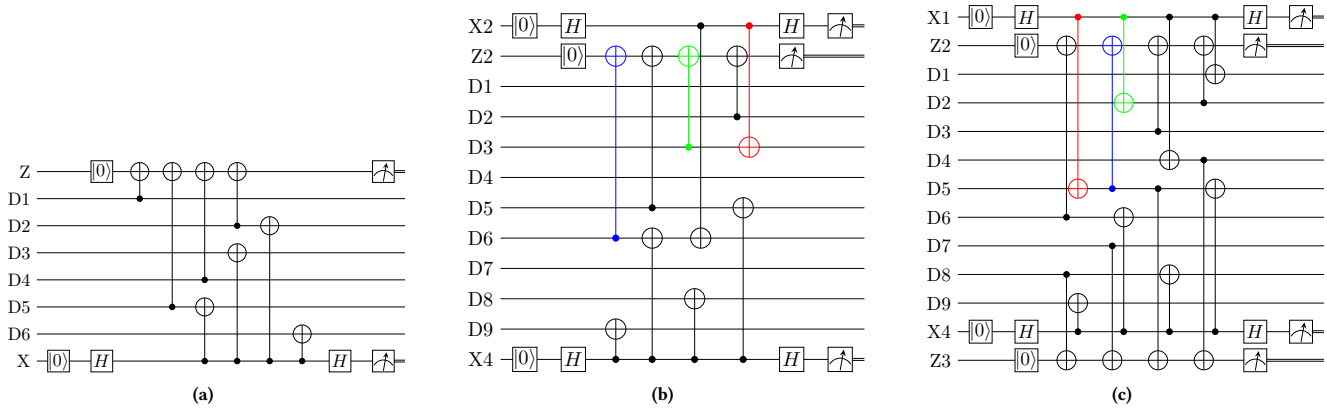


Figure 3: Qubit layouts for state injection on the rotated surface code. (a) The injection from corner (CR) approach and (b) the injection from middle (MR) approach For both approaches, first the circled qubit (magic qubit) is prepared in the magic state and the data qubits within the solid (dashed) outline are prepared in the $|+\rangle$ ($|0\rangle$) state. Then one performs two cycles of stabiliser measurements on the entire lattice and post-selects the states based on the measurement outcomes. The hatched stabilisers should be satisfied both before and after stabiliser measurements.

3.3 Injection from corner on the rotated surface code

We adapt the YL approach to the rotated planar surface code as shown in Figure 3(a). That is, it first injects a magic state to the data qubit (we call it magic qubit) in the upper-left **C**orner of the **R**otated lattice and other data qubits are prepared in either $|+\rangle$ or $|0\rangle$ states. The magic state is the +1 eigenstate of $M = \alpha X + \beta Y + \gamma Z$. Then one performs stabiliser measurement and post-selects the encoded magic states. We call this approach **CR** for short. We use the stabiliser measurement circuit for the rotated surface code presented in [37] which can reduce the final logical error rates compared to the CNOT ordering in Figure 4(a). Figure 4 (b) shows that the CNOT gates are completed in four timesteps. After the injection step, in the absence of errors, the logical qubit will be in the +1 eigenstate of $M_L = \alpha X_L + \beta Y_L + \gamma Z_L$, where $X_L = \Pi_{i\in\text{left boundary}} X_i$, $Z_L = \Pi_{j\in\text{top boundary}} Z_j$, $Y_L = iX_LZ_L$. In Appendix A, we explicitly show the stabiliser transformation during the first round of stabiliser measurement and the final logical magic state that results. We discard the runs where the hatched stabilisers have -1 measurement outcomes (i.e., they are mismatched with the initialisation step) or the two rounds of stabiliser measurements are different.

Any error that flips the sign of the operator M_L and cannot be detected by the postselection step will lead to a logical error. For CR, initialization (into the $|0\rangle$ state) errors on the magic qubit and its neighbour to the right will cause a logical error with probability $2p_{\text{IN}}$. Furthermore, the gate for rotating the magic qubit to the magic state will result in a logical error with probability $\frac{2}{3}p_1$. In addition,



(a) (b) (c)

Figure 4: (a) Segment of the depth-six (CNOT) stabiliser measurement circuit for the regular surface code when using the injection method in [1]. The depth-four circuits for performing stabiliser measurement on the sensitive qubits for (b) the CR and (c) the MR injection approaches on the rotated surface code. A single error on the colored (red, blue, green) CNOT gates may induce a logical error.

there are specific two-qubit errors occurring after the CNOT in the stabiliser measurement circuit which induce logical errors (see Figure 4 (b)). These are $X_C X_T$, $Y_C X_T$, $Z_C I_T$ after the green CNOT for measuring the $Z2$ stabiliser, $X_C X_T$, $X_C Y_T$, $Y_C X_T$, $Y_C Y_T$ after the blue CNOT (in the measurement of $Z2$), and $Z_C Z_T$, $Y_C Z_T$ after the red CNOT for measuring the $X2$ stabiliser ('C' and 'T' represent the control and the target qubits, respectively).

In the circuit-level error model, each of the above two-qubit gate errors occurs with probability $p_2/15$. Therefore, the logical error rate of a distance-d magic state implemented by this CR approach is

$$p_{\text{CR}} = \frac{3}{5}p_2 + 2p_{\text{IN}} + \frac{2}{3}p_1 + O(p^2). \tag{2}$$

3.4 Injection from middle on the rotated surface code

In the CR approach, the single-qubit errors on two data qubits (the magic qubit and its neighbour on the right) can induce logical errors. To further reduce logical error rates, we present a method named **MR**. In MR, the magic state is injected to the qubit located in the **M**iddle of the **R**otated lattice and other data qubits are initialised in the required $|+\rangle$ or $|0\rangle$ states as shown in Figure 3(b). After injection, the lattice is in the +1 eigenstate of the logical operator $\alpha M_L = X_L + \beta Y_L + \gamma Z_L$, where $X_L = \Pi_{i\in\text{middle vertical chain}} X_i$, $Z_L = \Pi_{j\in\text{middle horizontal chain}} Z_j$, $Y_L = iX_L Z_L$. By applying the post-selection protocol to MR, only the single-qubit operation errors on the magic qubit can result in logical errors, to first order. Single-qubit operation errors on other data qubits will not induce logical errors, e.g., qubit D6 (on top of the magic qubit) is initialised in the eigenstate of X operator and a Z error will be detected by the stabiliser X_{D3D6}. Similar to CR, initialization and single-qubit rotation errors on the magic qubit will cause logical errors with probabilities p_{IN} and $\frac{2}{3}p_1$ respectively. Furthermore, as shown in Figure 4 (b), the two-qubit errors on the magic qubit including $X_C X_T$ (on the blue CNOT) for measuring the $Z2$ stabiliser,

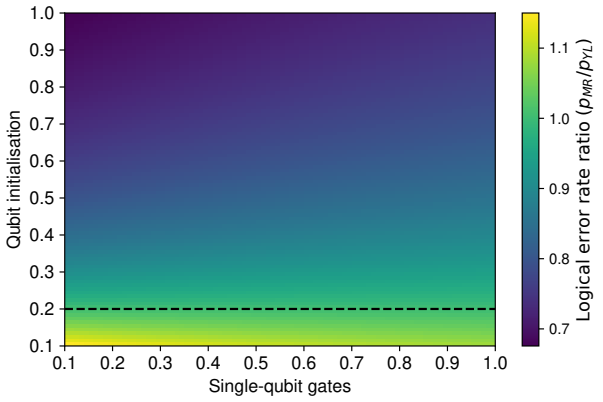


Figure 5: Comparison between the MR and YL approaches. The X axis and the Y axis show the relative error rates of single-qubit gates and qubit initialisation to two-qubit gates, which are $\frac{p_1}{p_2}$ and $\frac{p_{\text{IN}}}{p_2}$, respectively. The heatmap shows the relative logical error rates of MR to YL, that is, $\frac{p_{\text{MR}}}{p_{\text{YL}}}$ (Values smaller than 1 mean the MR approach achieves lower logical error rates than the YL approach). The dashed line marks the error rate of initialisation at $p_{\text{IN}} = \frac{1}{5}p_2$ where MR and YL have the same fidelity.

$Z_C Z_T$, $Z_C Y_T$, $I_C X_T$, $X_C I_T$, $Y_C Y_T$, $Y_C Z_T$ (on the red CNOT) for measuring the $X1$ stabiliser, and the two-qubit errors on the data qubit D2 including $X_C X_T$, $Y_C Y_T$ (on the green CNOT) for measuring the $X1$ stabiliser will induce logical errors. Therefore, the logical error rate of injecting a distance-d magic state using the MR approach is

$$p_{\text{MR}} = \frac{3}{5}p_2 + p_{\text{IN}} + \frac{2}{3}p_1 + O(p^2). \tag{3}$$

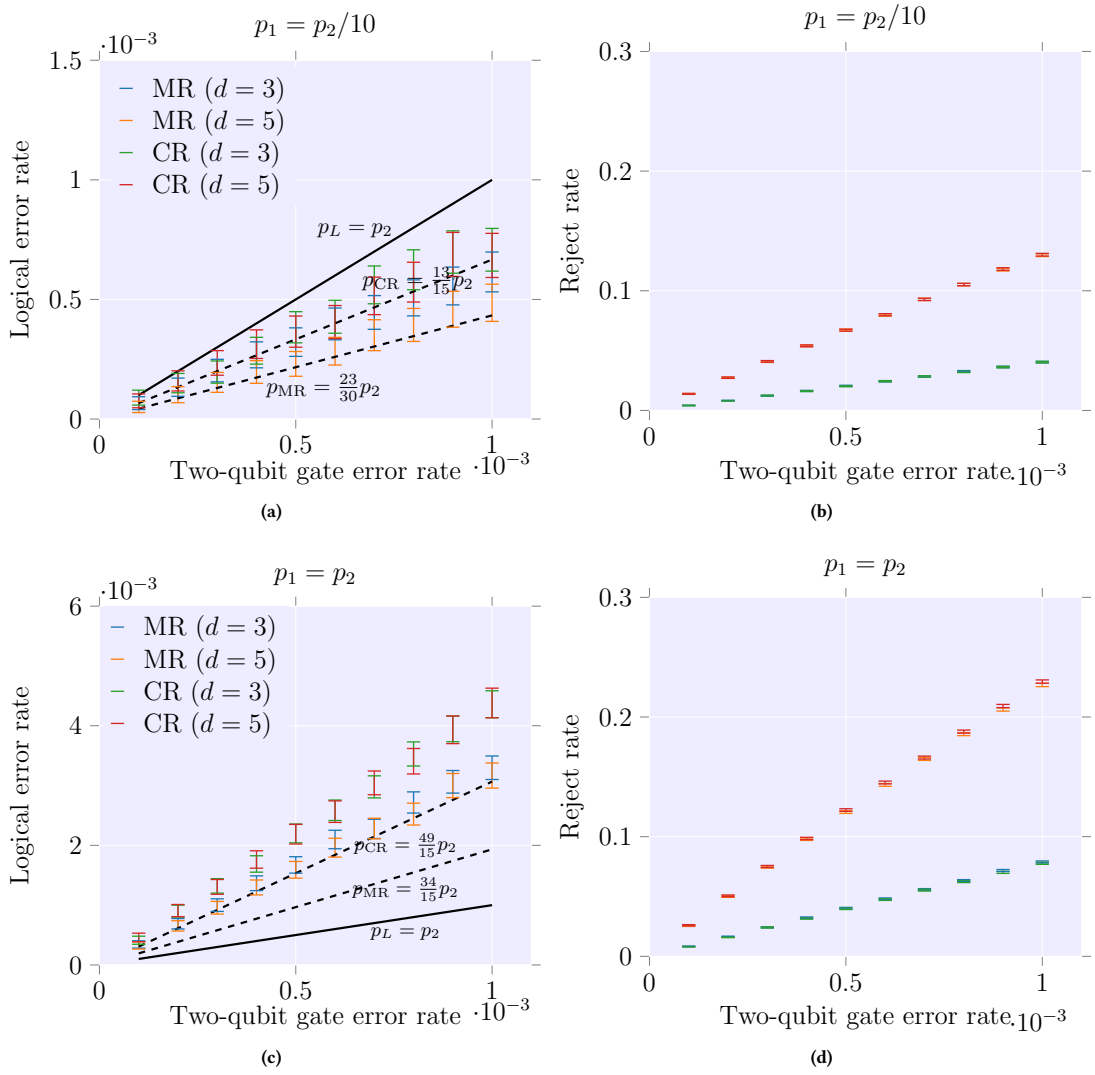


Figure 6: Numerical comparison of the CR (Section 3.3) and MR (Section 3.4) approaches in terms of their logical error rates and rejection rates of the post selection procedure. Error bars present 99.9% confidence intervals. Dashed lines are analytical values based on Equations 2 and 3 to first order. (a-b) Assuming biased errors $p_{\mathrm{IN}} = p_{\mathrm{M}} = p_1 = p_2/10$ and (c-d) assuming unbiased errors $p_{\mathrm{IN}} = p_{\mathrm{M}} = p_1 = p_2$. For the biased error model, the logical error rates converge to the analytical results and can be lower than the two-qubit gate error rate.

4 NUMERICAL RESULTS

We first compare the proposed MR approach for the rotated surface code with the YL approach in [1] for the regular surface code. We consider the biased error model that is motivated by many current quantum processors where single-qubit rotation gates have around five to ten times lower infidelity than two-qubit gates [38–40]. We also vary the error rates of qubit initialisation relative to two-qubit gates. Measurement error rates may be higher than two-qubit gate error rates [38–40] but will not be considered in this work. We compute the relative logical error rates between MR and YL ($\frac{p_{\mathrm{MR}}}{p_{\mathrm{YL}}}$) based on Equation 1 and Equation 3 to first order in the physical gate error rates.

Figure 5 shows the comparison of resulting logical error rates. The proposed MR approach has lower logical error rates than YL when the error rate of qubit initialisation is larger than $\frac{1}{5}$ of the two-qubit gate error rate ($p_{\mathrm{IN}} > \frac{1}{5} p_2$). MR has slightly worse performance than YL ($p_{\mathrm{MR}} = 1.15 p_{\mathrm{YL}}$) when $p_{\mathrm{IN}} = p_1 = \frac{1}{10} p_2$. When the errors are not biased ($p_{\mathrm{IN}} = p_1 = p_2$), MR reduces the logical error rate approximately by a factor of 1.35 compared to YL. Assuming the 15-to-1 distillation protocol [10], the error reduction by MR can be by a factor of 1.35^{3n} after n rounds of distillation.

Therefore, this fidelity improvement will help reduce distillation costs. We note that errors on idling qubits (i.e., decoherence) are not considered in this analysis. As mentioned in Section 3, the stabiliser measurement circuit of YL takes six timesteps for CNOT gates while MR only requires four timesteps. This means MR could achieve more fidelity improvement when taking qubit decoherence into account. Moreover, the rotated surface code uses around half the amount of data qubits to encode one logical qubit compared to the regular surface code. The higher fidelity and lower qubit requirement of the proposed MR approach will be beneficial for practical implementation of fault-tolerant quantum computation based on the surface code.

Furthermore, we numerically simulate the proposed MR approach and the CR approach based on the circuit-level error model using the stabiliser formalism [34]. In this simulation, we assume both unbiased errors $p_{\mathrm{I}} = p_{\mathrm{M}} = p_1 = p_2$ and biased errors $p_{\mathrm{I}} = p_{\mathrm{M}} = p_1 = p_2/10$. For each point in the numerics, 10^6 iterations of injection procedure have been run and confidence intervals at 99.9% are plotted.

Figure 6 shows that the MR approach can achieve lower logical error rates than the CR approach for both biased and unbiased errors. The rejection rates for these two methods are similar. For the same method, the logical error rates for different code distances are similar, but the rejection rate of a higher-distance code is larger than a lower-distance code. This implies that one can first prepare a low-distance magic state and then grow the code to a larger distance using a fault-tolerant approach such as the one proposed in [35] (Numerical results of the lattice growing procedure are provided in Appendix B). For lower physical error rates in Figure 6, the logical error rates converge to the first-order estimated values (dashed lines) in Equations 2 and 3. However, the logical error rates in Figure 6 (c) do not converge to the dashed lines, which might be due to the high amount of weight-2 physical errors that can lead to logical errors in lower-distance codes.

5 CONCLUSION AND DISCUSSION

We have presented two methods to inject magic states on the rotated surface code. We first analysed their logical error rates and then numerically simulated these approaches to verify the analytical results. Furthermore, we compared the proposed MR approach with the previous YL approach for the regular surface code by [1], For the same code distance, MR uses around half the amount of data qubits for encoding and could achieve lower logical error rates compared to YL. These improvements could potentially pave the way towards practical fault-tolerant quantum computation based on the surface code.

ACKNOWLEDGEMENTS

We thank Barbara Terhal, Christophe Vuillot, Daniel Litinski for useful discussions and Dan E. Browne for valuable feedback on the manuscript. LL acknowledges funding from the EPSRC Prosperity Partnership in Quantum Software for Modelling and Simulation (Grant No. EP/S005021/1).

REFERENCES

[1] Ying Li. A magic state's fidelity can be superior to the operations that created it. *New Journal of Physics*, 17(2):023037, 2015.
[2] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982.
[3] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.
[4] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.
[5] Julia Kempe. Quantum random walks hit exponentially faster. *arXiv preprint quant-ph/0205083*, 2002.
[6] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.
[7] Bryan Eastin and Emanuel Knill. Restrictions on transversal encoded quantum gate sets. *Phys. Rev. Lett.*, 102:110502, Mar 2009.
[8] Bei Zeng, Andrew Cross, and Isaac L Chuang. Transversality versus universality for additive quantum codes. *IEEE Transactions on Information Theory*, 57(9):6272–6284, 2011.
[9] Emanuel Knill. Fault-tolerant postselected quantum computation: Schemes. *arXiv preprint quant-ph/0402171*, 2004.
[10] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005.
[11] Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Phys. Rev. A*, 86:052329, Nov 2012.
[12] Adam M Meier, Bryan Eastin, and Emanuel Knill. Magic-state distillation with the four-qubit code. *arXiv preprint arXiv:1204.4221*, 2012.
[13] Cody Jones. Multilevel distillation of magic states for quantum computing. *Phys. Rev. A*, 87:042305, Apr 2013.
[14] Earl T. Campbell and Mark Howard. Unifying gate synthesis and magic state distillation. *Phys. Rev. Lett.*, 118:060501, Feb 2017.
[15] Daniel Litinski. Magic state distillation: Not as costly as you think. *Quantum*, 3:205, 2019.
[16] Ashley M. Stephens, Zachary W. E. Evans, Simon J. Devitt, and Lloyd C. L. Hollenberg. Asymmetric quantum error correction via code conversion. *Phys. Rev. A*, 77:062335, Jun 2008.
[17] Charles D Hill, Austin G Fowler, David S Wang, and Lloyd CL Hollenberg. Fault-tolerant quantum error correction code conversion. *Quantum Information & Computation*, 13(5-6):439–451, 2013.
[18] Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin. Fault-tolerant conversion between the steane and reed-muller quantum codes. *Phys. Rev. Lett.*, 113:080501, Aug 2014.
[19] Theodore J. Yoder, Ryuji Takagi, and Isaac L. Chuang. Universal fault-tolerant gates on concatenated stabilizer codes. *Phys. Rev. X*, 6:031039, Sep 2016.
[20] Eesa Nikahd, Morteza Saheb Zamani, and Mehdi Sedighi. A low-overhead hybrid approach for universal fault-tolerant quantum computation. *arXiv preprint arXiv:1610.03309*, 2016.
[21] Kristina R Colladay and Erich J Mueller. Rewiring stabilizer codes. *New Journal of Physics*, 20(8):083030, 2018.
[22] Hendrik Poulsen Nautrup, Nicolai Friis, and Hans J Briegel. Fault-tolerant interface between quantum memories and quantum processors. *Nature communications*, 8(1):1–8, 2017.
[23] Sergey Bravyi and Andrew Cross. Doubled color codes. *arXiv preprint arXiv:1509.03239*, 2015.
[24] Tomas Jochym-O'Connor and Stephen D. Bartlett. Stacked codes: Universal fault-tolerant quantum computation in a two-dimensional layout. *Phys. Rev. A*, 93:022323, Feb 2016.
[25] Cody Jones, Peter Brooks, and Jim Harrington. Gauge color codes in two dimensions. *Phys. Rev. A*, 93:052332, May 2016.
[26] Benjamin J Brown. A fault-tolerant non-clifford gate for the surface code in two dimensions. *Science advances*, 6(21):eaay4929, 2020.
[27] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, Sep 2012.
[28] Rami Barends et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, 2014.
[29] Charles D Hill et al. A surface code quantum computer in silicon. *Science advances*, 1(9):e1500707, 2015.
[30] Clare Horsman et al. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, 2012.
[31] Andrew J Landahl and Ciaran Ryan-Anderson. Quantum computing by color-code lattice surgery. *arXiv:1407.5103*, 2014.
[32] Eric Dennis et al. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002.
[33] Earl T Campbell and Mark Howard. Magic state parity-checker with pre-distilled components. *Quantum*, 2:56, 2018.
[34] Daniel Gottesman. The heisenberg representation of quantum computers. *arXiv preprint quant-ph/9807006*, 1998.
[35] Christophe Vuillot, Lingling Lao, Ben Criger, Carmen García Almudéver, Koen Bertels, and Barbara M Terhal. Code deformation and lattice surgery are gauge fixing. *New Journal of Physics*, 21(3):033028, 2019.

[36] Daniel Gottesman. Stabilizer codes and quantum error correction. *arXiv preprint quant-ph/9705052*, 1997.
[37] Yu Tomita and Krysta M. Svore. Low-distance surface codes under realistic quantum noise. *Phys. Rev. A*, 90:062320, Dec 2014.
[38] IBM. IBM Quantum Experience Devices. https://quantum-computing.ibm.com/, 2020.
[39] Rigetti. Rigetti Aspen-8. https://medium.com/rigetti/rigetti-aspen-8-on-aws-236d9dc11613, 2020.
[40] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

A STABILISER TRANSFORMATION DURING STATE INJECTION

As shown in Figure 3, the stabilisers in the initialisation step of the CR approach for the distance-3 surface code are

$$\langle M_3, X_1, X_2, X_4, X_5, X_7, Z_6, Z_8, Z_9 \rangle ,$$

and the operators that will be measured during stabiliser measurements (which are the stabilisers of code) are

$$\langle S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8 \rangle ,$$

$$S_1 = Z_1Z_2, S_2 = X_1X_2X_4X_5, S_3 = Z_2Z_3Z_5Z_6,$$

$$S_4 = X_3X_6, S_5 = X_4X_7, S_6 = Z_4Z_5Z_7Z_8,$$

$$S_7 = X_5X_6X_8X_9, S_8 = Z_8Z_9.$$

In Table 1, we explicitly show how the CR approach injects a magic state stabilised by $M = \alpha X + \beta Y + \gamma Z$ into the distance-3 surface code using the Heisenberg representation [34]. The stabilisers after state injection are presented in the last column of Table 1. Some of the stabiliser measurements have random outcomes. Others have +1 outcomes in the absence of errors. The logical qubit is in the +1 eigenstate of operator $M_L = M_3Z_6Z_9X_1X_2$. Any error that flips the sign of M_L and cannot be detected by the postselection procedure will cause a logical error. For example, an X_6 error will lead to a logical error while an X_9 error will not because it can be detected by the stabiliser Z_8Z_9.

B CODE GROWING

For the aforementioned injection approaches, the logical error rates of the prepared magic states are similar for different distances. However, the post-selection rates decrease with code distance because there are more error events in larger codes. It may be preferable to first inject a small-distance code and then grow it into a larger code. The procedure of growing lattices can be implemented fault-tolerantly by using code deformation as discussed in [35]. Figure 7 shows the procedure of expanding a distance-3 surface code to a distance-5 one. Two steps are required to maintain the code distance (see [35] for more details). We numerically simulate this lattice growing procedure. A decoder using the minimum-weight perfect matching (MWPM) algorithm is used in this simulation. Figure 8 shows the fault tolerance of this step, that is, the logical error rates decrease as the code distance increases when the physical error rates are below the error threshold.

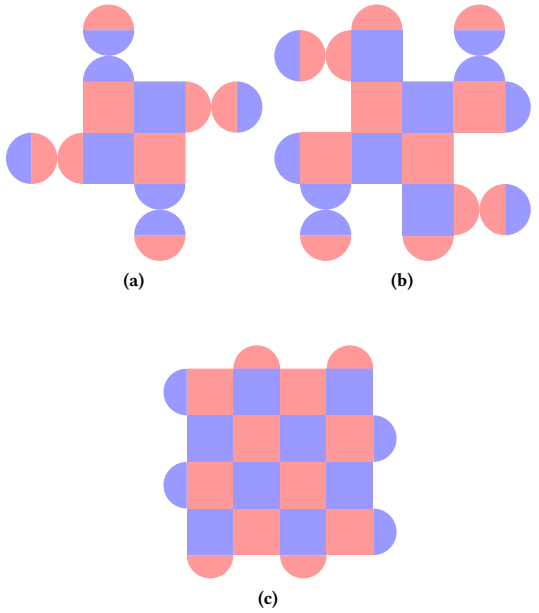


Figure 7: Fault-tolerant procedure for growing a lattice from d to $d + 2$. (a) The distance-d code and the pre-prepared EPR pairs on each side of the lattice. Growing the original code is realised by merging it with the EPR pairs, which is completed in two steps ((b) and (c)) to maintain the code distance.

Table 1: Stabiliser transformation of the CR approach. The first column shows the stabilisers in the initialisation and the last column shows the stabilisers after the first round of stabiliser measurements.

M_3		$(-1)^{S_4}X_3X_6$	$(-1)^{S_4}X_3X_6$	$(-1)^{S_4}X_3X_6$
Z_6	$\xrightarrow{M_{X_3X_6}}$	M_3Z_6	$M_3Z_6Z_9$	$M_3Z_6Z_9X_1X_2$
X_1	$\xrightarrow{M_{Z_1Z_2}}$	$(-1)^{S_1}Z_1Z_2$	$(-1)^{S_1}Z_1Z_2$	$(-1)^{S_1}Z_1Z_2$
X_2		X_1X_2	X_1X_2	$(-1)^{S_3}Z_2Z_3Z_5Z_6$
X_4	$\xrightarrow{M_{Z_4Z_5Z_7Z_8}}$	$(-1)^{S_6}Z_4Z_5Z_7Z_8$	$(-1)^{S_6}Z_4Z_5Z_7Z_8$	$(-1)^{S_6}Z_4Z_5Z_7Z_8$
X_5		X_4X_5	X_4X_5	$X_1X_2X_4X_5$
X_7		X_4X_7	X_4X_7	X_4X_7
Z_8		Z_8	Z_8Z_9	Z_8Z_9
Z_9		Z_9	$(-1)^{S_7}X_5X_6X_8X_9$	$(-1)^{S_7}X_5X_6X_8X_9$

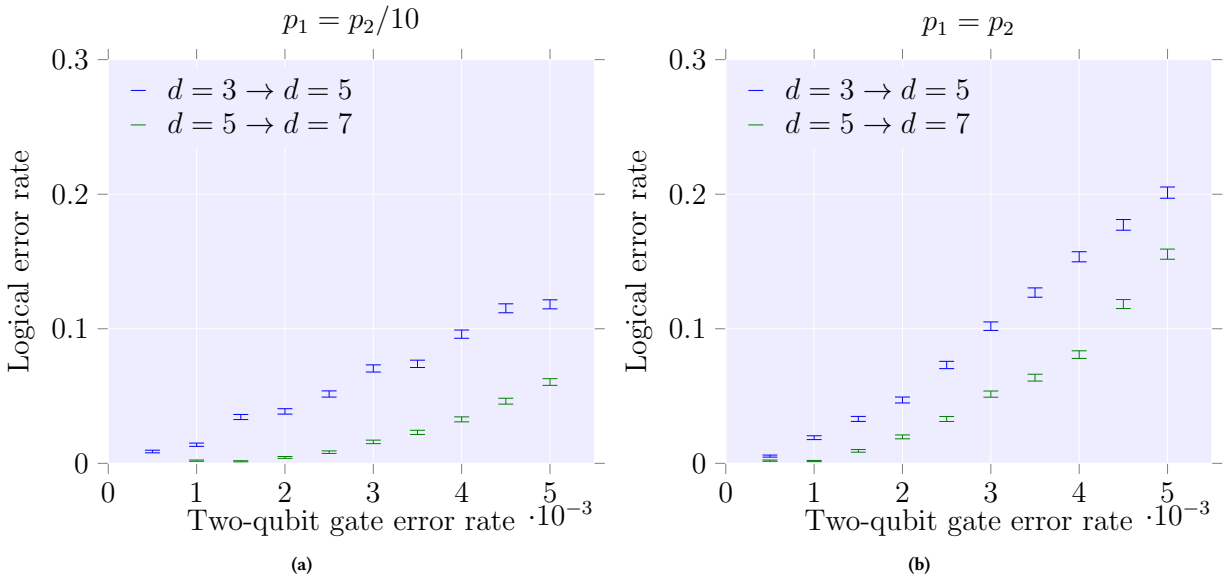


Figure 8: Numerical results of the lattice growing procedure assuming (a) biased errors $p_{\mathbf{IN}} = p_{\mathbf{M}} = p_1 = p_2/10$ and (b) unbiased errors $p_{\mathbf{IN}} = p_{\mathbf{M}} = p_1 = p_2$.