

Fold-transversal surface code cultivation

Kaavya Sahay^{✉,*†}, Pei-Kai Tsai^{✉,*}, Kathleen (Katie) Chang,^{*} Qile Su[✉], Thomas B. Smith[✉],
Shraddha Singh^{✉,‡} and Shruti Puri

*Departments of Applied Physics and Physics, Yale University, New Haven, Connecticut 06520, USA
and Yale Quantum Institute, Yale University, New Haven, Connecticut 06511, USA*



(Received 6 October 2025; revised 4 March 2026; accepted 17 April 2026; published 7 July 2026)

Magic state cultivation is a state-of-the-art protocol to prepare ultra-high fidelity non-Clifford resource states for universal quantum computation. It offers a significant reduction in spacetime overhead compared to traditional magic state distillation techniques. Cultivation protocols involve measuring a transversal logical Clifford operator on an initial small-distance code and then rapidly growing to a larger-distance code. In this work, we present a new cultivation scheme in which we measure the fold-transversal Hadamard of the unrotated surface code, and leverage unitary techniques to grow within the surface code family. Using both stabilizer and state vector simulations we find that this approach achieves the lowest known spacetime overhead for magic state cultivation. Practical implementation of our protocol is best suited to architectures with nonlocal connectivity, showing the strength of architectures where such connectivity is readily available.

DOI: [10.1103/gpvl-lg4c](https://doi.org/10.1103/gpvl-lg4c)

I. INTRODUCTION

Surface codes are a popular choice for quantum error-correction due to their high thresholds and local, low-weight stabilizers [1–4]. A major bottleneck for scalable quantum computation with the surface code is the cost to implement fault-tolerant non-Clifford operations [5]. The most effective way to realize non-Clifford operations with these codes is by teleporting in high-fidelity magic states—that is, an eigenstate of a Clifford operator [6]. The logical error rate (LER) required for these magic states depends on the target quantum algorithm. For example, recent resource estimates for factoring large integers require input magic states with an infidelity of roughly 10^{-7} [7,8]. The spacetime cost to reach such low LERs with surface codes can be debilitatingly large.

Magic state cultivation (MSC) has recently emerged as an extremely resource-efficient method for preparing high-quality magic states [9–12]. However, it requires a code that has a transversal implementation of a Clifford operator. Since the rotated surface code does not have access to

a transversal Clifford gate, MSC with the surface code needs to start in a different code that can later be morphed or grafted into a rotated surface code.

MSC protocols proceed as follows. Initially, the eigenstate of a Clifford operator is injected into a small-distance code. This injection step is inherently noisy. Subsequently, the logical Clifford operator is measured transversally multiple times in order to cultivate a higher-fidelity magic state in the small code via postselection on non-trivial measurement outcomes. Finally, the small code is rapidly grown to a larger-distance rotated surface code for later use in a logical circuit. It is imperative that this final growth step does not significantly add to the LER of the protocol. Therefore, further postselection is used in this step.

Note that the size of the initial code and number of logical Clifford measurements fixes the protocol’s fault distance f . This is the minimum number of physical errors that may cause a logical error without introducing non-trivial syndromes. A protocol in which the Clifford operator is measured on a larger code has a higher achievable fault distance and is thus capable of producing a higher-fidelity magic state. However, this introduces increased spacetime overhead (STO) due to the use of additional qubits, more measurement rounds, and ultimately, more postselection.

In this work, we present a new method for cultivating a high-fidelity $H_{XY} = (X + Y)/\sqrt{2}$ state in the rotated surface code. Our protocol is summarized in Fig. 1. We observe that the unrotated surface code has a fold-transversal logical H_{XY} gate [13,14]. We leverage this property for cultivation in the unrotated surface code, along with a method for unitarily transforming between

*These authors contributed equally to this work.

†Contact author: kaavya.sahay@yale.edu

‡Present address: IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

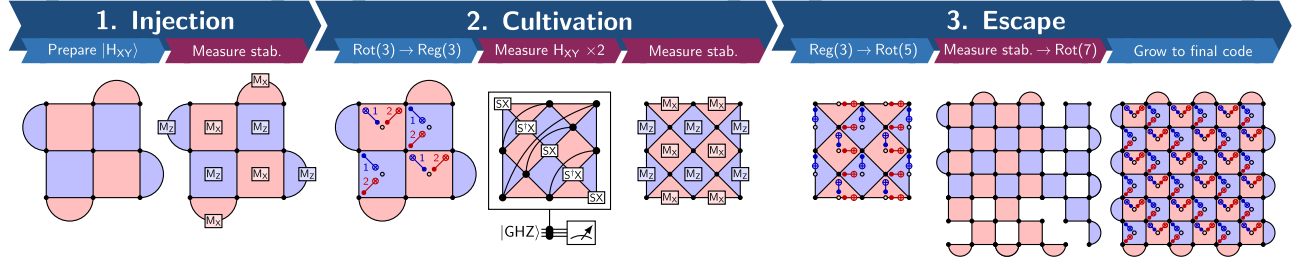


FIG. 1. Magic state cultivation with a fault distance $f = 3$. The procedure interleaves unitary steps (light blue labels) and measurements on which postselection occurs (maroon labels). 1. *Injection*: prepare an eigenstate of $H_{XY} = (X + Y)/\sqrt{2}$ on the distance-three rotated surface code $\text{Rot}(3)$, and measure its stabilizers. 2. *Cultivation*: transform to the distance-three regular surface code $\text{Reg}(3)$, and measure the fold-transversal H_{XY} operator twice via a GHZ ancilla. Then, measure the stabilizers of $\text{Reg}(3)$. 3. *Escape*: transform to $\text{Rot}(5)$ via unitary growth, then use stabilizer measurements to grow to $\text{Rot}(7)$. Decode, and post-select on the associated complementary gap. If desired, unitarily grow to a larger final code.

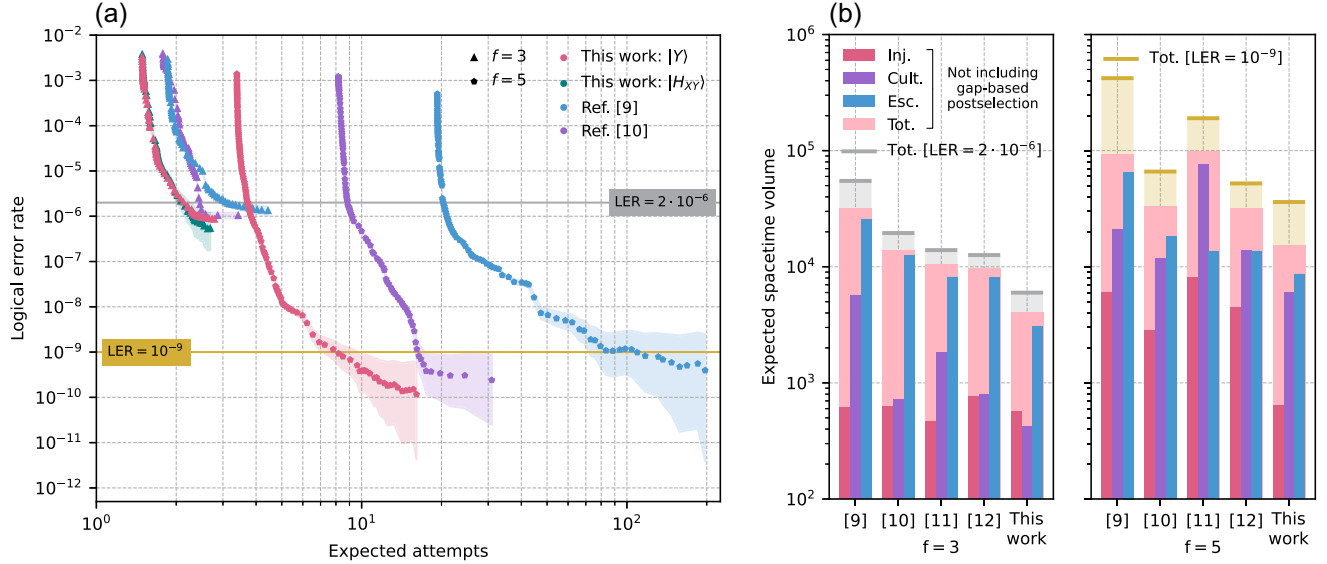


FIG. 2. Cultivation under uniform depolarizing noise. (a) Curves show the expected number of cultivation attempts of fault distances $f = 3$ (triangles) and $f = 5$ (pentagons) to prepare a state with a particular logical error rate. The $f = 3$ protocol is described in Sec. II, and its extension to $f = 5$ in Appendix C. Our scheme for $|Y\rangle$ state cultivation (dark pink) requires a lower number of expected attempts for any target LER compared to Refs. [9,10] (blue and purple, respectively). The exact simulation of $|H_{XY}\rangle$ magic state cultivation for $f = 3$ (green) is in good agreement with $|Y\rangle$ state results. (b) Expected spacetime volume (qubits $\times$ gate count $\times$ expected attempts) of $|Y\rangle$ state cultivation for fault distances $f = 3$ and $f = 5$, compared to Refs. [9–12]. For a fair comparison, the $f = 3$ protocols are normalized to the same final code distance $d_{\text{fin}} = 13$, and the $f = 5$ protocols to $d_{\text{fin}} = 15$. The expected volume in the absence of gap-based postselection (light pink) is shown, along with the component expected volumes for injection (dark pink), cultivation (purple) and escape (blue). Additional gap-based postselection is required to cultivate states with logical error rates of 2×10^{-6} for $f = 3$ (silver) and 10^{-9} for $f = 5$ (gold). Our scheme has a significantly lower overhead, owing to the efficient cultivation and escape steps that use nonlocal operations.

rotated and unrotated surface code variants [15]. Our choice of initial code, magic state, and simplified growth technique allows us to achieve lower logical error rates and significantly lower spacetime overheads compared to previous schemes.

Importantly, our scheme is most naturally implemented with nonlocal gates. These are native to quantum computing platforms such as neutral atoms [16] and trapped ions [17]. More recently, high-fidelity nonlocal gates have also

been demonstrated in superconducting circuits [18,19]. Our work demonstrates the dramatic reduction in requirements for scalable quantum computing with surface codes when nonlocal gates are available.

The performance of our magic state cultivation scheme is summarized by Fig. 2(a). This is quantified by the expected attempts required to obtain a magic state of a particular LER. Here, our results are compared to existing cultivation protocols for which the equivalent data were

available [9,10]. We find that our procedure requires a lower number of expected attempts for any target LER. The overhead relative to all current protocols [9–12] is shown in Fig. 2(b). This is measured by the expected spacetime volume. We calculate the “baseline expected volume not accounting for the additional postselection after the escape stage—as well as the expected volume required to obtain a magic state with an error rate of 2×10^{-6} and 10^{-9} . The use of nonlocal operations allows us to reduce the expected volume of injection, cultivation, and escape. As such, we obtain a much lower STO compared to existing work.

The remainder of this paper is structured as follows. In Sec. II we describe our MSC protocol in detail. In Sec. III we benchmark our protocol against previous work by simulating cultivation of the logical Y eigenstate, $|Y\rangle = S|+\rangle$. In addition, we use a combination of state vector and stabilizer simulations to accurately benchmark cultivation for the logical magic state, $|H_{XY}\rangle = T|+\rangle$. We also investigate the performance of our protocol under physically motivated noise models. We conclude in Sec. IV.

II. PROTOCOL

Our MSC protocol prepares a high-fidelity eigenstate of the logical operator $H_{XY} = (X + Y)/\sqrt{2}$ in a rotated surface code. We label magic states by the operator for which they are a $+1$ eigenstate. Here, the magic state is $|H_{XY}\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$, which is used to implement the logical $T = e^{-i\pi Z/8}$ gate via teleportation. Figure 1 illustrates our protocol for cultivating the $|H_{XY}\rangle$ state with a fault distance $f = 3$. The protocol has three main stages: injection, cultivation and escape.

Injection—In this stage, we prepare a logical magic state in a small-distance rotated surface code, $\text{Rot}(3)$. While there are several ways to achieve this, here we use a unitary encoding circuit, followed by a round of stabilizer measurements. The encoding circuit, which uses one physical non-Clifford T gate and nonlocal Clifford operations, is provided in Appendix A. This injection stage has a fault distance $f = 1$.

Cultivation—In this stage, another unitary circuit is used to transform $\text{Rot}(3)$ into an unrotated surface code, $\text{Reg}(3)$. This circuit is simply the first two steps of a stabilizer measurement round on $\text{Rot}(3)$ [20]. Note that this technique requires the introduction of additional qubits to act as the ancillae for $\text{Reg}(3)$. How this is achieved depends on underlying physical hardware. It is most naturally realized in architectures with reconfigurable qubits, such as neutral atom arrays [16].

After this transformation, the fold-transversal logical H_{XY} operator is measured twice using a three-qubit ancilla GHZ state. This operation uses some three-qubit gates, as shown in Appendix A. Unlike previous proposals [9,10,12], our logical Clifford measurement procedure is

designed to minimize the wait time for code qubits while operations are being carried out only on the GHZ ancillae.

Following this logical Clifford measurement, we perform a round of stabilizer measurements at $\text{Reg}(3)$. Postselecting on both the stabilizer measurements and the decoded GHZ measurements increases the fault distance of the protocol to $f = 3$. At this point, we have produced a high-fidelity magic state on a small-distance code.

Escape—In this stage, the code is grown to a larger distance in order to protect the magic state from future errors. Here, we use a three-step escape strategy. First, $\text{Reg}(3)$ is grown to $\text{Rot}(5)$ via a local unitary circuit [15]. Next, $\text{Rot}(5)$ is grown to $\text{Rot}(7)$ using a conventional stabilizer measurement approach [21]. Finally, $\text{Rot}(7)$ is unitarily grown to a final larger-distance code. The stabilizer measurements obtained from the growth to $\text{Rot}(7)$ are used for additional postselection. Shots are discarded depending on the likelihood of a logical error as determined by a complementary-gap-based decoder [22–25].

This three-stage escape allows us to do most of our growth unitarily, while still extracting the stabilizer information necessary for gap-based postselection. The reliance on primarily unitary circuits is crucial for maintaining a low spacetime overhead compared to using possibly slower and lower-fidelity stabilizer measurements on the larger-distance final code.

We provide further details about the protocol in the appendices. In Appendix A we provide circuits for key steps, and in Appendix B we discuss alternative choices for each stage. These choices include the magic state, injection strategy, GHZ ancilla size, pre-escape stabilizer measurement code, and escape strategy. In Appendix C we outline the additional steps required for protocols with a higher fault distance. These are a straightforward extension of the $f = 3$ protocol that we have described above. Note that we must measure the logical H_{XY} operator additional times to achieve a higher fault distance. Other modifications to the $f = 5$ scheme include a modified logical Clifford measurement circuit, and the use of a local, unitary circuit [26] for escape.

A. Comparison with other schemes

A summary of the differences between our protocol and previous schemes is provided in Table I. Here we highlight the constraints of previous schemes that limit their relative LER and STO.

Reference [9] uses a technique called grafting to convert between a color code and the rotated surface code in the escape stage [27]. This introduces additional postselection overhead that we bypass by remaining in the Rot – Reg surface code family. References [10,11] use morphing circuits to convert, respectively, between $\mathbb{RP}^2$ and Reg codes and their self-dual variants. This enables a strictly transversal logical Clifford measurement using two-qubit

TABLE I. Comparison of cultivation schemes for $|H_{XY}\rangle$ with fault distance $f = 5$. *Initial code*: the code chosen for state injection. *Clifford check code*: the code in which logical Clifford checks are performed. *Clifford check ancilla*: the largest ancillary system used to perform the logical Clifford check, where $\text{GHZ}(n) = (|0\rangle^{\otimes n} + |1\rangle^{\otimes n})/\sqrt{2}$. *Discard rate*: the expected discard rate required to cultivate a $|Y\rangle$ state with $\text{LER} = 10^{-9}$, where discard rate $1 - 1/(\text{expected attempts})$. *Performance constraints*: aspects of each cultivation scheme that limit their performance relative to ours. Note that all schemes end in a rotated surface code.

Scheme	Initial code	Clifford check code	Clifford check ancilla state	Discard rate of $ Y\rangle$ with $\text{LER} = 10^{-9}$	Performance constraints
Ref. [9]	Color(3)	Color	$ +\rangle^{\otimes 19}$	99%	Planar connectivity, grafting
Ref. [10]	$\mathbb{RP}^2 - 3$	SRP	$ +\rangle^{\otimes 25}$	93%	Inefficient code morphing
Ref. [11]	Rot(3)	$S - \text{Reg}$	$ +\rangle^{\otimes 9}$	99%	Restricted connectivity, ≤ 2 -qubit gates
Ref. [12]	Rot(3)	Reg	$ +\rangle^{\otimes 21}$	96%	High-overhead logical check
This work	Rot(3)	Reg	GHZ(5)	90%	

gates. In contrast, we relax the requirement of morphing to a self-dual code and allow for the use of three-qubit gates. Our code conversion within the surface code family is comparatively more efficient. Reference [11] only allows operations that can be efficiently realized on a neutral-atom platform under reconfigurability constraints. While we consider this as a design principle, we leave exact optimization to future work.

Figure 2(b) shows that we are able to obtain the lowest expected spacetime volume for all stages: injection, cultivation, and escape. We attribute most of our improvements to the use of a more efficient check circuit, and our three-stage escape strategy. These improvements are discussed further throughout Appendix B.

III. RESULTS

In this section, we quantify the performance of our protocol numerically. In Sec. III A, we use stabilizer simulations of logical $|Y\rangle$ state preparation to benchmark against previous works. Using the $|Y\rangle$ state as a proxy for the $|H_{XY}\rangle$ state avoids simulating non-Clifford circuits. However, it is yet unclear if this provides robust estimates of the LER of magic state cultivation. Motivated by this, in Sec. III B we introduce a *handoff* approach, where outputs from a state vector simulator are passed to a stabilizer simulator. This allows us to study exact $|H_{XY}\rangle$ state cultivation for the first time. Finally, in Sec. III C, we examine the performance of our scheme in an experimentally motivated setting.

A. $|Y\rangle$ state cultivation

To adapt our scheme for the logical $|Y\rangle$ state, we make two changes to the protocol described in Sec. II. First, injection is modified to use an S gate instead of a T gate [10,28]. Second, we measure the logical Y operator in place of measuring the logical H_{XY} operator. We provide the relevant circuits in Appendix A. In this section, we compare our scheme against other cultivation schemes in

terms of LER vs expected attempts, and expected space-time volume [9,10]. Circuits for all schemes are simulated under a uniform depolarizing noise model (SD6) with physical error rate $p = 10^{-3}$.

Figure 2(a) shows the expected attempts to achieve a particular LER. For each curve, the leftmost data point reflects the performance without any complementary-gap-based postselection during escape. As one progresses right, increasing amounts of postselection via the complementary gap drives the LER lower, until the error floor is reached by fully post-selecting on any nontrivial syndrome. For any given LER, our protocol consistently requires lower expected attempts than existing proposals. Additionally, our scheme is the only one so far for which $f = 3$ cultivation has a floor below 10^{-6} . This brings us slightly closer to the 7×10^{-7} LER required for 2048 bit integer factorization [8]. For our $f = 5$ cultivation, 10 expected attempts are required to reach an LER of 10^{-9} , fewer than the 15–90 attempts required by previous proposals.

Figure 2(b) highlights the advantage of our approach in terms of overhead. We calculate the expected spacetime volume required for target LERs of 2×10^{-6} and 10^{-9} , for protocols with fault distances $f = 3$ and $f = 5$, respectively. The spacetime volume is defined as the qubit-gate product (the number of active qubits times the number of gate steps). This metric naturally accounts for differences in depth and qubit count across protocols and also reflects the fact that error detection can halt a protocol mid-way rather than incurring the cost of executing the full circuit. Using this metric provides a more accurate comparison than the expected number of attempts. Our protocol demonstrates significant improvement in spacetime volume compared to Refs. [9,10]. We refer the reader to Appendix E for further details on the calculation of expected spacetime volume.

B. $|H_{XY}\rangle$ state cultivation

Here, we describe the state vector to stabilizer handoff simulation that is used to accurately estimate the LER for

$|H_{XY}\rangle$ magic state cultivation. In Appendix D we describe our handoff simulation strategy in detail. We provide a high-level summary below.

First, we use a state vector simulator from injection up to the completion of the logical Clifford measurements. This part of the circuit involves non-Clifford physical gates but only involves a limited number of qubits. After the logical measurements, the logical qubit state vector and the code's stabilizer eigenvalues are extracted. This logical qubit state vector is stored offline, and the stabilizer eigenvalues are passed onto the stabilizer simulator.

The stabilizer simulator then simulates the rest of the protocol. The decoder is only given stabilizer measurement information post-handoff and predicts Pauli errors that have occurred. If the predicted errors differ from the actual errors by logical Pauli operators, then the corresponding logical Pauli operator is applied to the stored pre-handoff logical qubit state vector. We compare this final noisy state to the ideal magic state to determine if the protocol has succeeded.

Figure 2(a) contains the results for $|H_{XY}\rangle$ cultivation with $f = 3$. Interestingly, the LERs for $|H_{XY}\rangle$ cultivation are virtually indistinguishable from those for $|Y\rangle$ state cultivation. Reference [9] observed a difference between the LERs of $|Y\rangle$ and $|H_{XY}\rangle$ cultivation when simulating a truncated protocol, where the escape stage is excluded. We also observe a difference for truncated simulations, but this effect does not carry over to our full simulations that include escape. We discuss $|Y\rangle$ vs $|H_{XY}\rangle$ simulations further in Appendix D.

The state vector simulations are only performed up to the logical checks in the cultivation stage, which involve a small number of qubits. This renders handoff tractable for the $f = 3$ protocol. However, our handoff strategy is too expensive to simulate $f = 5$, which uses roughly 47 qubits during cultivation. Nevertheless, our demonstration for $f = 3$ establishes it as a useful tool for estimating the performance of protocols where the system is limited to a small number of qubits during non-Clifford operations in a circuit.

C. Tailoring cultivation to experiments

Finally, we examine the performance of our protocol in settings inspired by nonlocal architectures such as neutral atoms. Broadly speaking, in quantum systems with non-local operations, single-qubit gates are less noisy than multi-qubit gates, and local gates are less noisy than nonlocal gates. Moreover, for neutral atoms, idle errors are an insignificant portion of the error budget. Note that the measurement of H_{XY} uses physical three-qubit CCZ gates. This may be challenging to realize natively in some hardware platforms. Consequently, in Ref. [29] we investigate $|H_{XY}\rangle$ cultivation in a setting where CCZ gate is decomposed into two-qubit gates [30]. For the remainder

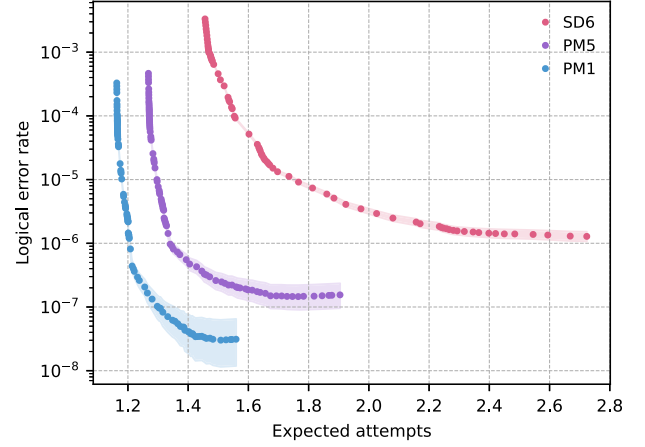


FIG. 3. Physically motivated noise. Logical error rate of $|Y\rangle$ state cultivation under three different noise models. A standard depolarizing noise model (SD6) is compared to two physically motivated noise models (PM1 and PM5). In PM1 the noise strength of single-qubit gates is ten times lower than that of multi-qubit gates. The same is true for PM5, but the noise strength of nonlocal multi-qubit gates is five times that of local multi-qubit gates. PM1 and PM5 have no idle errors.

of this section, we simulate $|Y\rangle$ preparation as, based on our previous discussion, we expect its trends to extend to $|H_{XY}\rangle$ cultivation.

We consider two physically motivated noise models, denoted PM1 and PM5. In these noise models, there are no idle errors, single-qubit gates have uniform depolarizing noise at a rate $p/10$, and local multi-qubit gates have uniform depolarizing noise at a rate p . The two models differ in the noise strength assigned to nonlocal multi-qubit gates. In PM1 these gates experience uniform depolarizing noise at a rate p (the same as local multi-qubit gates), and in PM5 they have a higher noise rate $5p$. We also use a more natural gate set for neutral atom qubits, and decompose CX gates into CZ gates and local single-qubit rotations.

Note that cultivation requires the state to be in the $+1$ eigenspace of all stabilizers. If using measurement-based injection, this necessitates fast measurement-conditioned feedback, which may not be experimentally desirable. Here, we use optimized unitary state preparation followed by a subset of stabilizer measurements. Importantly, fast feedback based on the stabilizer measurements is not required in this injection strategy, which is described in more detail in Appendix B.

Figure 3 shows the performance of our protocol under these noise models with $p = 10^{-3}$, compared to the standard depolarizing noise model (SD6) used in previous sections. We observe that PM noise models lead to reduced LERs compared to SD6 even when nonlocal gates are heavily penalized. This indicates that the protocol is largely limited by single-qubit gates and idling errors.

Furthermore, our protocol requires less expected attempts for a given LER compared to Ref. [12].

IV. DISCUSSION AND CONCLUSION

In this work, we have presented a fold-transversal cultivation scheme for the surface code. We have benchmarked its performance by evaluating $|Y\rangle$ state cultivation, and for the first time have studied exact $|H_{XY}\rangle$ magic state cultivation. Our protocol achieves the lowest known spacetime overhead for MSC. This improved performance is due to efficient unitary circuits that allow us to stay entirely within the surface code family throughout, and utilize the native fold-transversal H_{XY} operator. Both of these tasks require nonlocal operations. There are some natural extensions of our work.

The first arises from the reduced sampling cost of our cultivation protocol. As our scheme reaches lower LERs with far fewer expected attempts, it is conceivable to simulate fault distances higher than $f = 5$. This may enable us to determine the necessary spacetime overhead to produce a magic state with much lower LER, perhaps low enough that further distillation is not even required.

It is also important to study the performance of MSC under realistic noise models. While we only consider Pauli errors in our work, real hardware experiences a variety of other error mechanisms. It would be informative to analyze performance of our scheme in the presence of erasure errors [12,31], which is relevant to neutral atom architectures [32–35]. Due to the ability to post-select on erasure, we anticipate further improved fidelity in this setting. Several platforms also experience leakage or coherent errors. However, these are more challenging to simulate.

Throughout this work we used a nonlocal unitary circuit to grow the surface code, and convert between rotated and unrotated variants. This contributes to our lower spacetime overheads. However, nonlocal unitary growth is not compatible with all qubit platforms. It is worthwhile to consider low-overhead growth using alternative schemes such as the local unitary circuits of Refs. [26,36]. These circuits have an inherent tradeoff between gate depth and idling time. If an overhead reduction is possible via their use, then our protocol may be better applied to fixed-qubit architectures with limited nonlocal connectivity.

Beyond these immediate extensions of our work, an avenue of future research is to extend magic state cultivation to other quantum low-density parity check (LDPC) codes hosting fold-transversal gates [37] and families with morphing circuits [38].

We expect our work to inspire new surface code protocols that leverage experimental advances in nonlocal operations for overhead reduction. At the very least, these

advances call for a more fair comparison between the surface code and other quantum LDPC codes that have nonlocal stabilizers. Now equipped with a full computational suite of accurate decoders, efficient Clifford operations, and low-overhead magic state generation in nonlocal architectures, surface codes remain a leading candidate for practical quantum computation.

Note added. The authors were made aware of two related publications in the same arXiv posting: an updated version of Ref. [12] with an investigation of $|H\rangle$ state cultivation on the surface code, and surface code cultivation using two-qubit gates from Claes [11].

ACKNOWLEDGMENTS

We thank Ludwig Schmid for modifying the unitary encoding optimizers of Ref. [39], which led to the identification of the optimized unitary injection scheme. We would also like to thank Craig Gidney, Jeff D. Thompson, and Jahan Claes for helpful discussions, and the authors of Refs. [9,22,40], for releasing their code, which has been used extensively in this work. This work was supported under ARO (Grant No. W911NF-23-1-0051), DARPA MeasQuIT (Grant No. HR00112490363), NSF QLCI (Grant No. OMA2120757), and DOE C²QA (Contract No. DE-SC0012704).

DATA AVAILABILITY

Sample code and data for this work are provided at Ref. [41].

APPENDIX A: CIRCUITS

In this section, we provide circuit diagrams for key stages of our protocol. The circuits used to simulate fault distance $f = 3|Y\rangle$ state cultivation (Fig. 2) are shown in Figs. 4, 5, and 7–9 (generated in *Stim*). Red regions in these figures represent X detectors (which flip in the presence of Z errors), and blue regions represent Z detectors (which are affected by X errors). The H_{XY} check circuit for **Reg(3)** is shown in Fig. 6. This circuit is used to simulate $|H_{XY}\rangle$ state cultivation (Fig. 2).

In Appendix C we describe how these circuits are modified for state cultivation with a higher fault distance $f = 5$. A notable modification is the design of the logical check circuit for **Reg(5)**. In Fig. 10 we show the H_{XY} check circuit, and in Fig. 11 we show the equivalent Y check circuit.

Finally, an alternative unitary encoding circuit is shown in Fig. 12. This is used to simulate $|Y\rangle$ state cultivation under physically motivated constraints (Fig. 3).

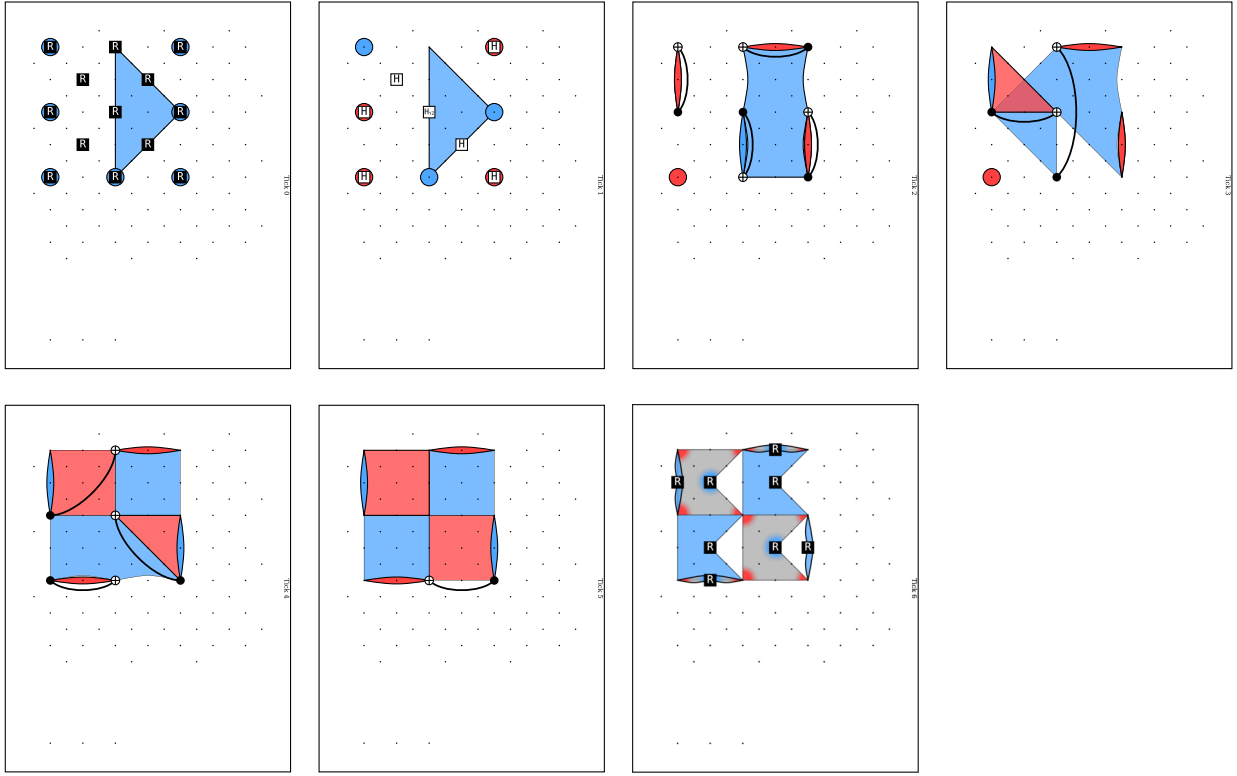


FIG. 4. Unitary encoder for $\text{Rot}(3)$. The rotation on the central qubit in the second step produces a physical $|Y\rangle$ state. In subsequent steps this is used to unitarily prepare the logical $|Y\rangle$ state. A subsequent round of stabilizer measurements (not pictured) is used to detect weight-1 and 2 errors.

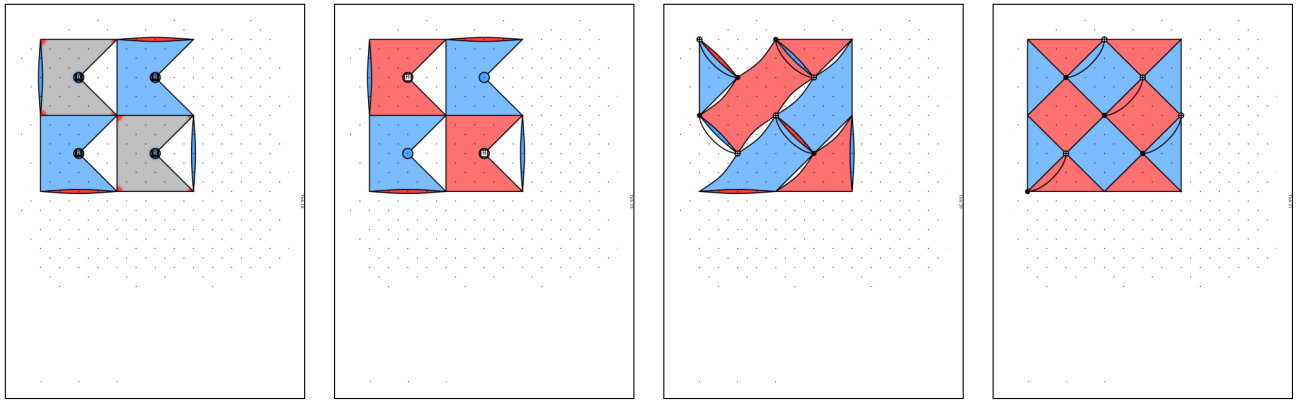


FIG. 5. Growth from $\text{Rot}(3)$ to $\text{Reg}(3)$. This is simply the first two steps of the stabilizer measurement circuit for $\text{Rot}(3)$. The ancilla qubits of $\text{Rot}(3)$ become data qubits of $\text{Reg}(3)$. Additional qubits are required for the $\text{Reg}(3)$ code for stabilizer measurements. Depending on the hardware, these ancilla may be present beforehand (static architectures with nonlocal connectivity), or freshly initialized qubits may be introduced to the lattice (reconfigurable architectures).

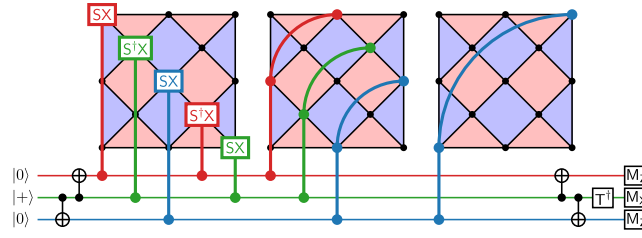


FIG. 6. Logical H_{XY} check for $\text{Reg}(3)$. The ancilla GHZ(3) state is supported on the bottom wires. The check is decomposed into controlled-SX gates involving the diagonal qubits, and CCZ gates involving pairs of off-diagonal qubits.

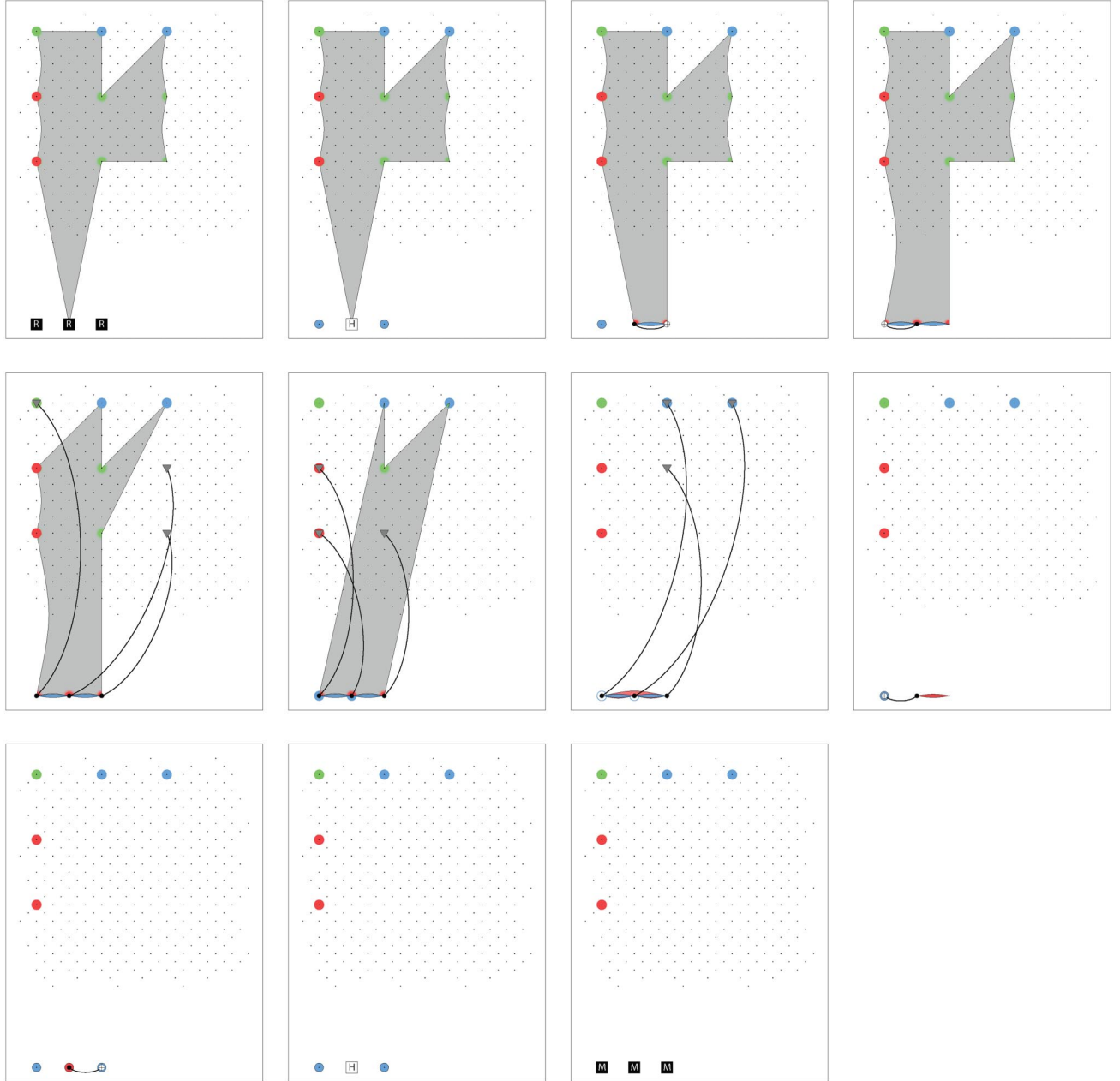


FIG. 7. Logical Y check for $\text{Reg}(3)$. The logical observable is highlighted. Note that since the logical Y operator is transversal on $\text{Reg}(3)$ (compared to the fold-transversal logical H_{XY} operator) we only need two-qubit gates.

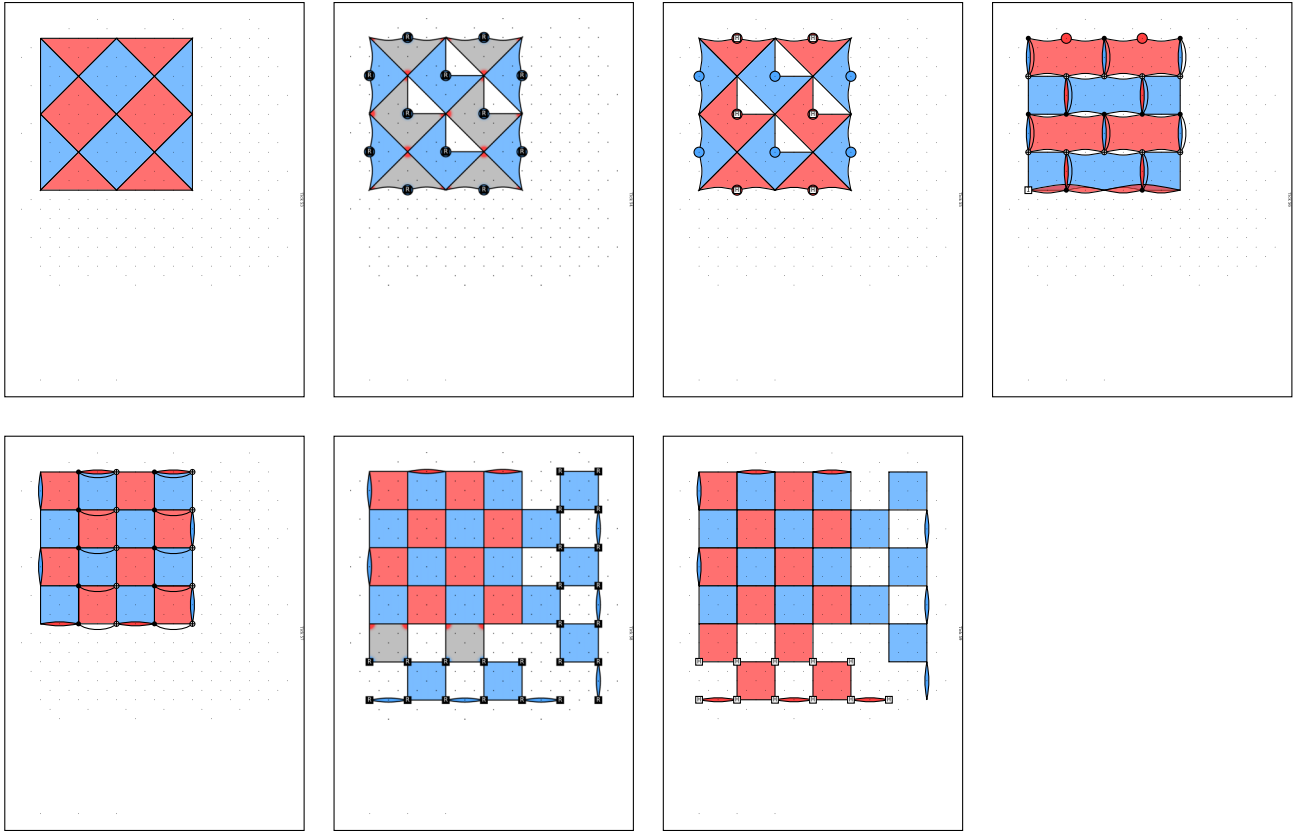


FIG. 8. Unitary growth from $\text{Reg}(3)$ to $\text{Rot}(5)$. This is the first stage of our three-stage escape sequence. The first two steps of stabilizer measurement circuit for $\text{Reg}(3)$ are used to grow and rotate aka *growtate* the code to $\text{Rot}(5)$. As in Fig. 5, the ancilla qubits of $\text{Reg}(3)$ become data qubits of $\text{Rot}(5)$, and additional ancilla qubits need to be introduced. Furthermore, two patches of freshly initialized qubits, used for the second stage of escape, are shown. This second stage involves subsequent rounds of stabilizer measurements (not pictured) that induce growth from $\text{Rot}(5)$ to $\text{Rot}(7)$.

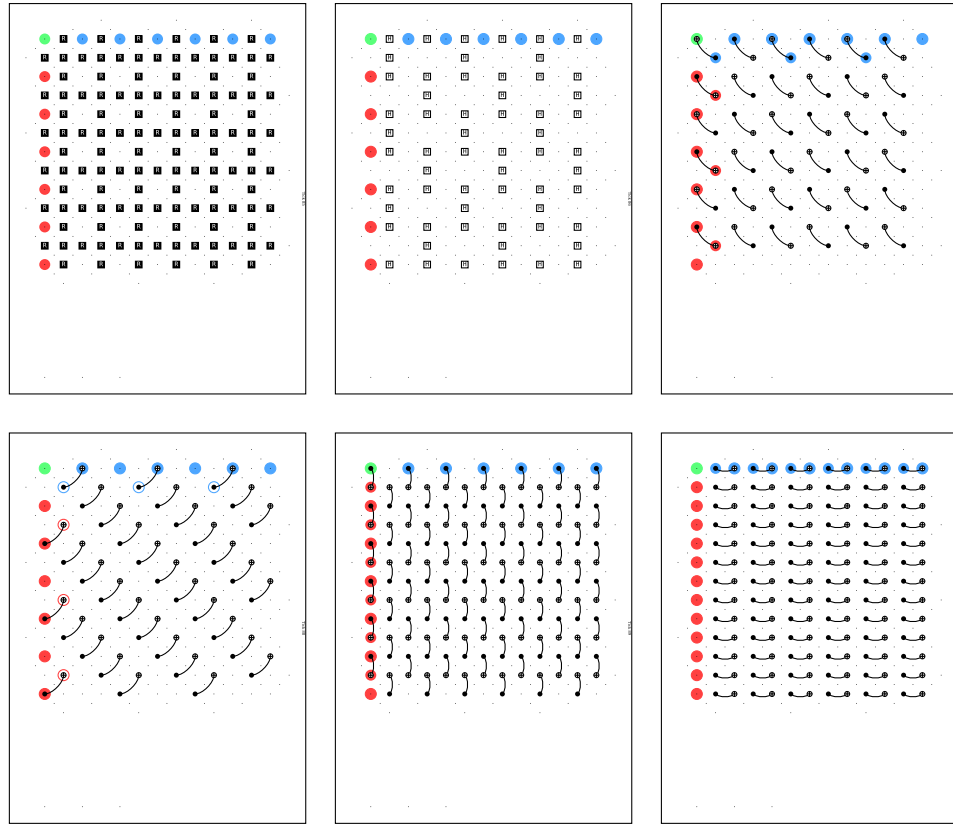


FIG. 9. Unitary growth from Rot(7) to Rot(13). Detectors are omitted for visual clarity. This is the final stage of our three-stage escape sequence. The logical Y observable is highlighted, showing the increase in its support.

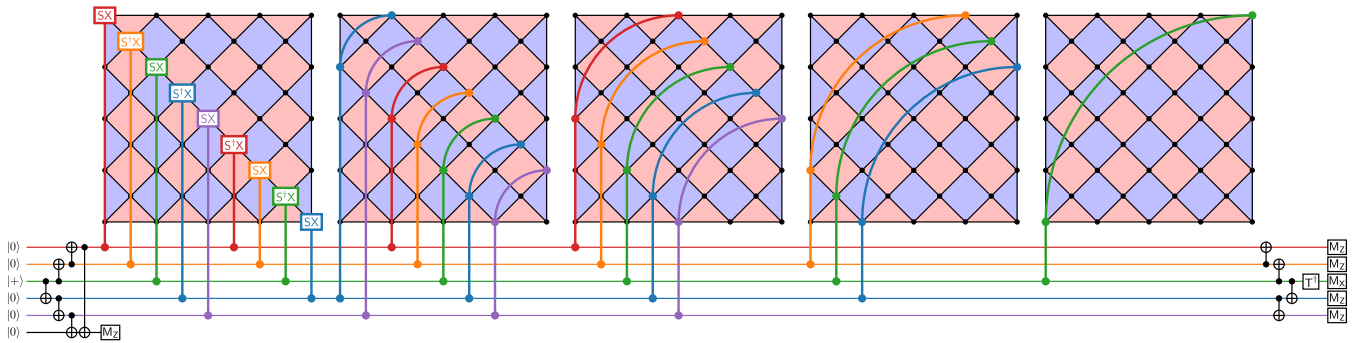


FIG. 10. Logical Y check for Reg(5). The ancilla GHZ(5) state is supported on the bottom wires, with an additional qubit used as a flag for fault-tolerant preparation. This circuit ensures that four or less errors on the GHZ state do not propagate to an undetectable logical error on the code.

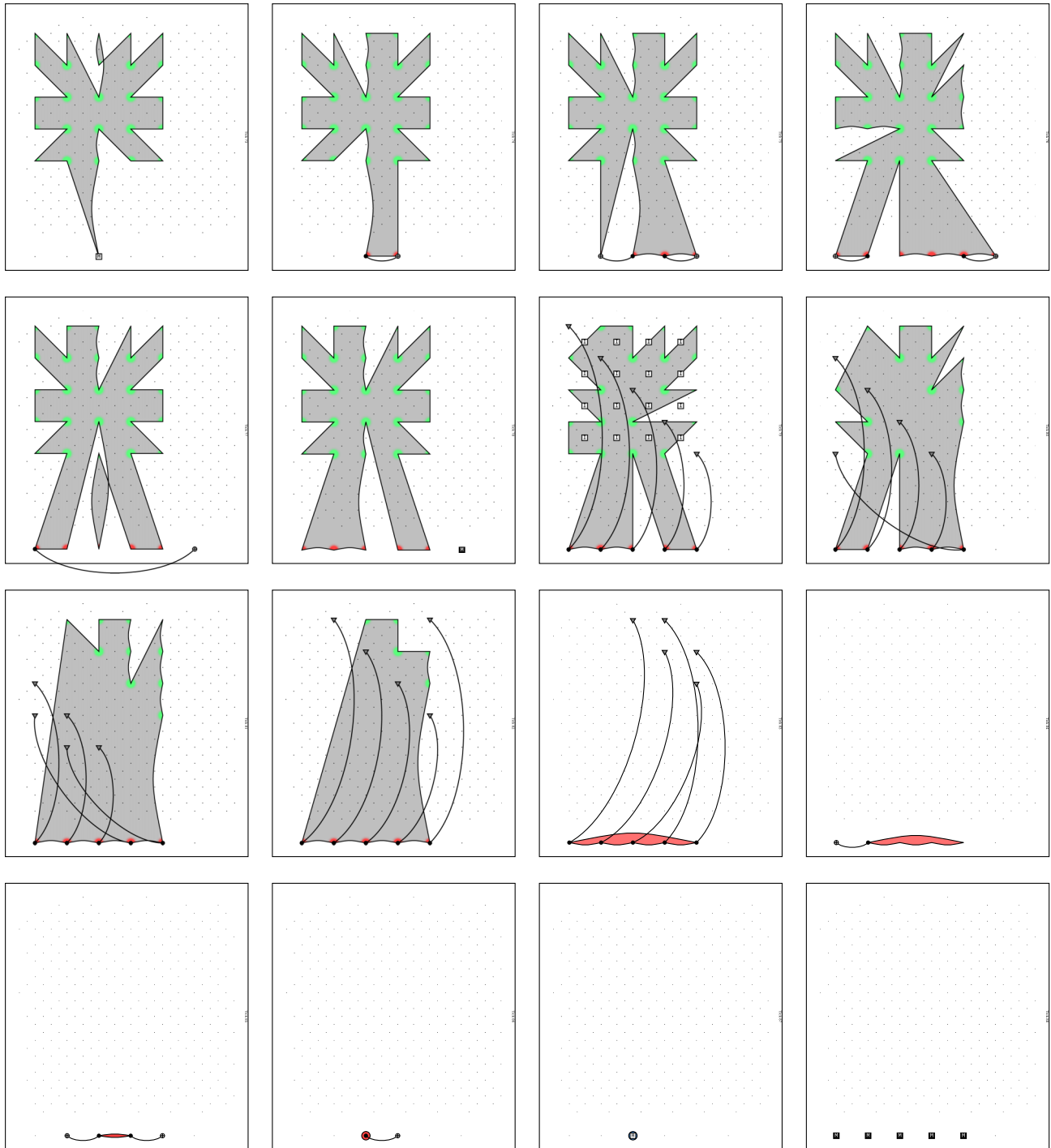


FIG. 11. Logical Y check for Reg(5).

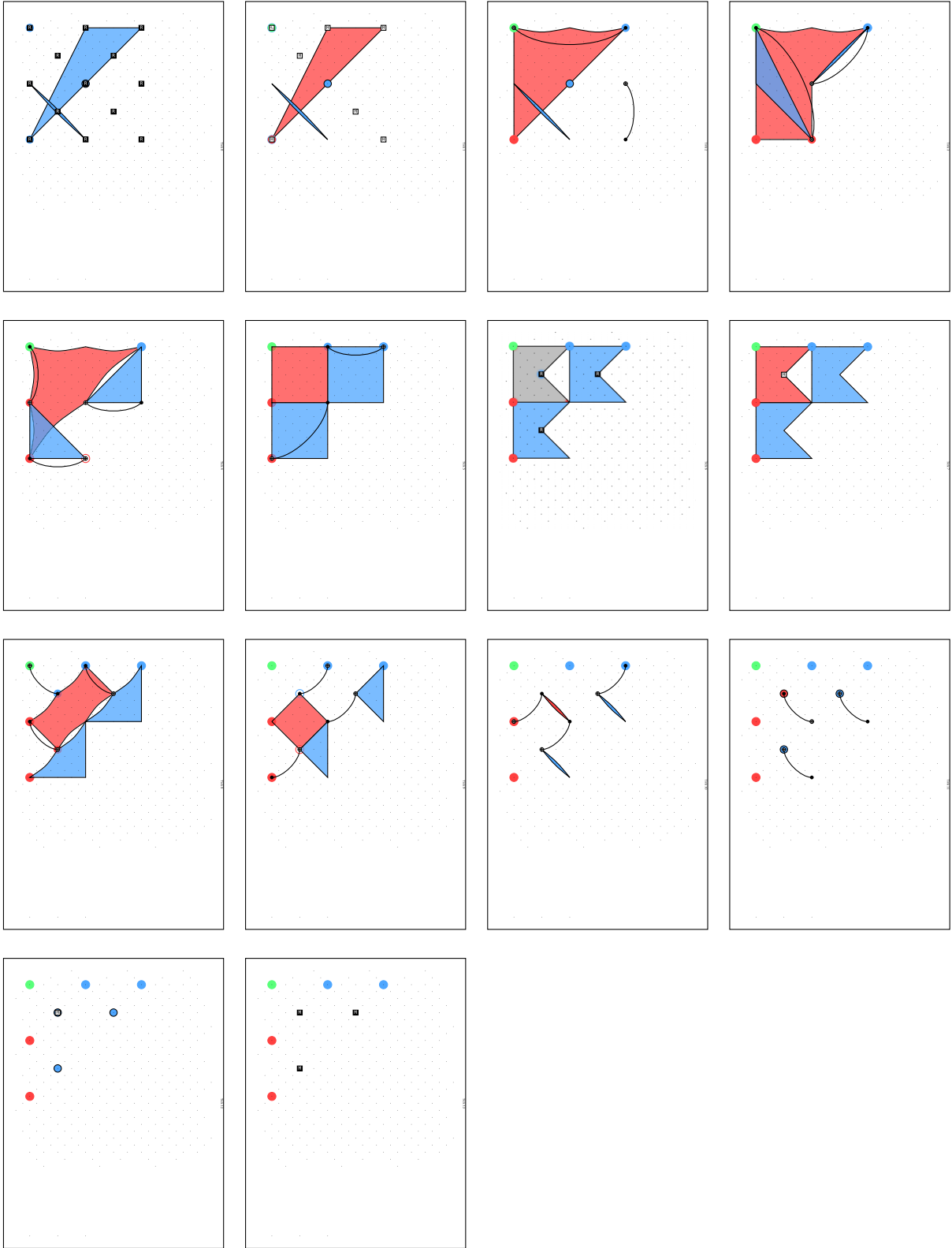


FIG. 12. Alternative unitary encoder for $\text{Rot}(3)$. This circuit only requires the subsequent measurement of three code stabilizers (detectors shown) to maintain fault distance. It is therefore particularly suited to architectures where measurements are more costly than gates in terms of speed, fidelity or both.

APPENDIX B: POTENTIAL MODIFICATIONS

In Sec. II, we described a cultivation protocol that achieved both low logical error rates and reduced space-time overhead. This protocol is highly modular, where any given stage can be modified to use different codes or subroutines. Here, we describe several alternatives, showing the resultant performance tradeoffs. This discussion is structured by going through the procedure for cultivation from beginning to end and considering points where modifications may be made. In this section and the remainder of the appendix, logical operators $\bar{O}$ are denoted with a bar, to differentiate them from physical operators O .

1. Logical state

In this section, we discuss the logical non-Clifford state used for magic state cultivation. We restrict ourselves to considering protocols where the logical state is checked at $\text{Reg}(3)$. In this setting, checking a Hadamard-like operator presents a natural choice due to its fold transversality on $\text{Reg}(3)$. Here, we briefly digress to first prove the correctness of our implementation of the logical $\bar{H}_{XY}$ on $\text{Reg}(3)$ and then show that $\bar{H}_{XY}$ requires less physical gates compared to $\bar{H}_{XZ}$.

In unrotated surface codes, the logical $\bar{S}$ can be implemented fold transversally [14,37]. For a $d \times d$ unrotated surface code, let D_i be the i th qubit along the main diagonal where $0 \leq i \leq 2d-2$, Δ be the set of qubits below the main diagonal (not included), and τ be the map that reflects a qubit across the main diagonal. For a qubit $q \in \Delta$, we call $(q, \tau(q))$ a pair of *mirrored qubits*. $\bar{S}$ can be implemented as

$$\bar{S} = (S_{D_0} S_{D_1}^\dagger \cdots S_{D_{2d-3}}^\dagger S_{D_{2d-2}}) \prod_{q \in \Delta} \text{CZ}_{q, \tau(q)}, \quad (\text{B1})$$

where S and $S^\dagger$ gates alternately act on qubits on the main diagonal, and CZs are applied to each pair of mirrored qubits.

As $\bar{X}$ can be applied fold transversally by applying X on all qubits on the main diagonal, $\bar{H}_{XY} = e^{-i\pi/4} \bar{S} \bar{X}$ is also fold transversal. Specifically, $\bar{H}_{XY}$ can be written as

$$\begin{aligned} \bar{H}_{XY} &= e^{-i\pi/4} \bar{S} \bar{X} \\ &= e^{-i\pi/4} \prod_{i \text{ even}} (SX)_{D_i} \prod_{j \text{ odd}} (S^\dagger X)_{D_j} \prod_{q \in \Delta} \text{CZ}_{q, \tau(q)} \\ &= \prod_{i \text{ even}} (TXT^\dagger)_{D_i} \prod_{j \text{ odd}} (TYT^\dagger)_{D_j} \prod_{q \in \Delta} \text{CZ}_{q, \tau(q)} \\ &= \prod_{i \text{ even}} (G_X Z G_X^\dagger)_{D_i} \prod_{j \text{ odd}} (G_Y Z G_Y^\dagger)_{D_j} \prod_{q \in \Delta} \text{CZ}_{q, \tau(q)} \end{aligned} \quad (\text{B2})$$

TABLE II. Gate count for controlled- $\bar{H}$ checks. We show the number of two-qubit gates (n_{2Q}) and three-qubit gates (n_{3Q}) for different code distances d .

d	n_{2Q} for $\bar{H}_{XY}$	n_{2Q} for $\bar{H}_{XZ}$	n_{3Q}
3	5	21	4
5	9	73	16
d	$2d-1$	$4d^2-6d+3$	$(d-1)^2$

where $G_X = TH$, $G_Y = TSH$ are single-qubit rotations, and we have used the gate identity $TXT^\dagger = e^{-i\pi/4} SX$.

To implement a controlled- $\bar{H}_{XY}$ gate for logical measurements, we need a CZ gate (conjugated by G_X or G_Y) for each qubit on the main diagonal, and a CCZ gate for each pair of mirrored qubit. In total, there are $(2d-1)$ CZ gates and $(d-1)^2$ CCZ gates.

The unrotated surface code supports a fold-transversal $\bar{H}_{XZ} = (\bar{X} + \bar{Z})/\sqrt{2}$, whose $+1$ eigenstate $|\bar{H}_{XZ}\rangle$ is also a magic state that is Clifford-equivalent to the conventional $|T\rangle$ state. It is implemented as

$$\bar{H}_{XZ} = \left(\prod_{q \in \Delta} \text{SWAP}_{q, \tau(q)} \right) H^{\otimes n}, \quad (\text{B3})$$

where $H^{\otimes n}$ denotes physical Hadamard gates on all qubits, followed by SWAP gates on each pair of mirrored qubits. To measure $\bar{H}_{XZ}$, its controlled version involves CH (controlled-Hadamard) gates acting on all data qubits, and CSWAP (controlled-SWAP) gates on each pair of mirrored qubit. Each CH can be written as $R_Y(\pi/4) \cdot \text{CZ} \cdot R_Y(-\pi/4)$, and each CSWAP gate can be decomposed into a single CCZ gate and two CZ gates. In total, there are $(d^2 + 3(d-1)^2)$ CZ gates and $(d-1)^2$ CCZ gates. The number of three-qubit gates is the same as in $\bar{H}_{XY}$, with an increase in the number of two-qubit gates, as summarized in Table II. Therefore we choose to prepare the $|\bar{H}_{XY}\rangle$ state in this work.

We implement the controlled- $\bar{H}_{XY}$ using a combination of controlled- SX and CCZ gates. Figure 6 shows the $\bar{H}_{XY}$ check circuit for $f=3$. Since the decomposition $\bar{H}_{XY} = e^{-i\pi/4} \bar{S} \bar{X}$ introduces an extra phase factor, an additional $T^\dagger$ gate must be applied to the ancilla qubit. This gate is placed immediately before the ancilla X -basis measurement. Figure 10 shows the $\bar{H}_{XY}$ check for $f=5$. In practice, one may prefer the decomposition using CZ, CCZ, and single-qubit gates as in Eq. (B2). In this form, no $T^\dagger$ gate on the ancilla is required; instead, suitable single-qubit rotations are applied before and after the CZ gates.

2. Injection scheme

In this section, we discuss different ways to inject the $|\bar{H}_{XY}\rangle$ state. Note that we restrict ourselves to injection

into $\text{Rot}(3)$ and not $\text{Reg}(3)$ due to its lower spacetime overhead. We consider the following:

1. *Hook injection* [40,42]. This injection scheme consists of two rounds of stabilizer measurements. In the first round, gates on Z-stabilizers along the diagonal are deliberately misordered such that a mid-cycle T gate on an ancilla is converted into a logical operator. We discard any shots with non-trivial measurement outcomes on fixed stabilizers—or conflicting outcomes on unfixed stabilizers across these two rounds. For a uniform depolarizing noise model with strength p , the logical error rate of this scheme is $\frac{8}{15}p$. Note that stabilizer-based injection schemes such as this require a subsequent

measurement-based-fixup operation to ensure that the codespace is the $+1$ eigenspace of all stabilizers.

2. *Lao-Criger injection* [43]. This scheme also uses two rounds of stabilizer measurements and thus has a similar spacetime overhead to hook injection. For a uniform depolarizing noise model with strength p , the logical error rate of this scheme is $\frac{34}{15}p$. This is higher than hook injection, therefore we do not consider this injection scheme further in this work.
3. *Unitary encoder + stabilizer measurements*. Using a unitary encoding circuit should reduce the spacetime volume of injection. Unlike optimizations made in Ref. [9], here we consider a naïve unitary encoder with the physical $|T\rangle$ state as a part of the

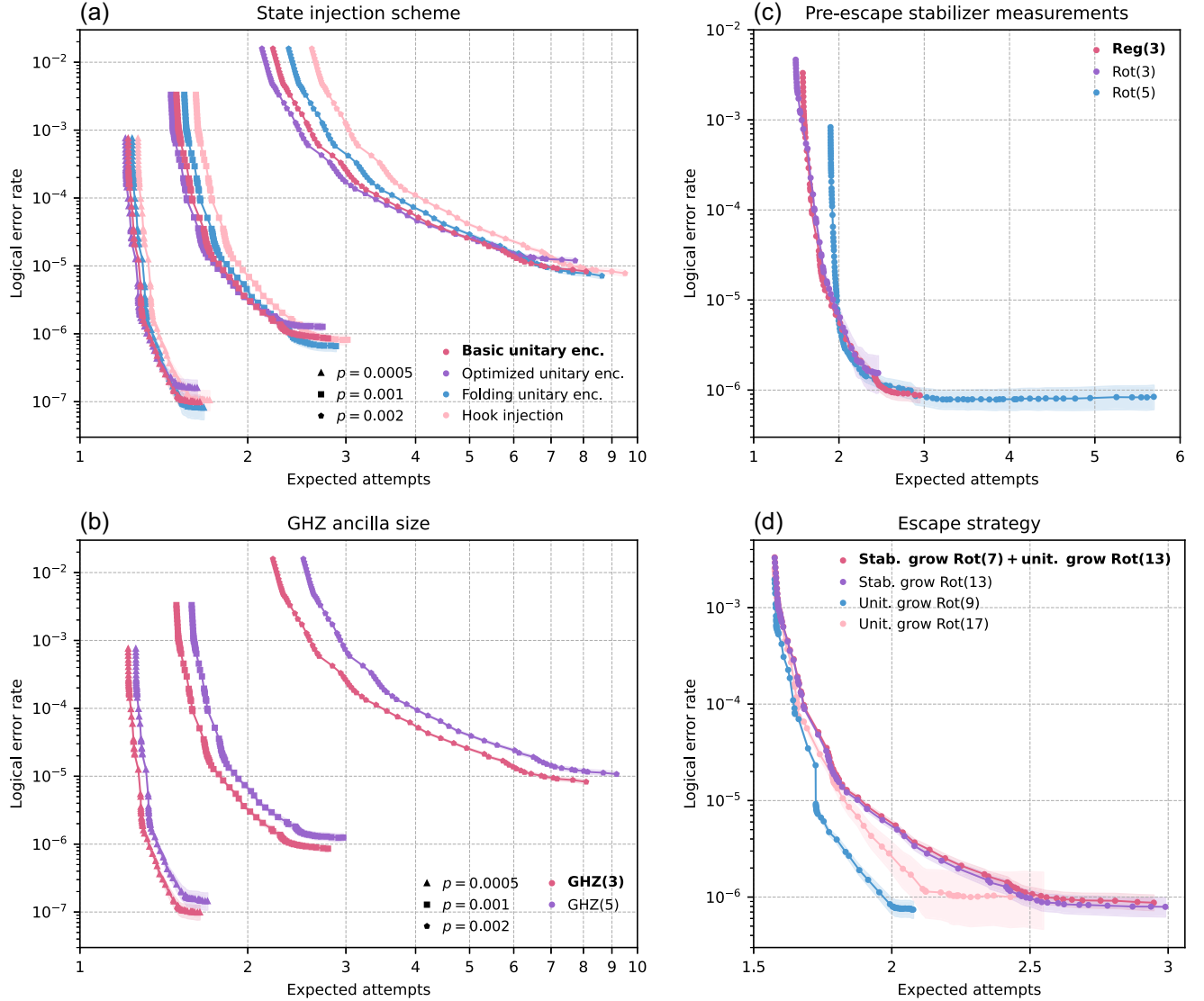


FIG. 13. Alternative choices for cultivation subroutines. Each panel shows the performance of the $f = 3|Y\rangle$ state cultivation, but with one modified step. The specific step that is modified is indicated by the title of the panels, and the modifications used are stated in the legend. Our “baseline” choice used in the main text is indicated in bold. A uniform depolarizing noise model with strength $p = 0.001$ is used throughout, unless indicated otherwise in the legend.

initial product state. Note that unitary circuits intrinsically allow weight-1 errors to spread to weight-2 along a logical. Consequently, they generally require a round of stabilizer measurements to detect and post-select out these mechanisms.

4. *“Optimized” unitary encoder + subset of stabilizer measurements.* It is possible to design an optimized unitary encoding circuit that minimizes the spread of damaging errors. This has been done in the quantum toolkit of Ref. [39]. Using this package, we obtain an optimized unitary encoder that requires us to measure only three out of the eight stabilizers of $\text{Rot}(3)$ in order to detect weight-1 errors that spread along the logical. This further reduces the spacetime volume for injection compared to a naïve unitary encoder. This injection scheme is shown in Fig. 12 and discussed in Sec. III C.
5. *Folding unitary encoder + stabilizer measurements.* Other magic state cultivation schemes use a unitary encoding circuit that first prepares a logical Pauli state, then “folds” in the logical operator, rotates it to the $|\bar{H}_{XY}\rangle$ state, and then folds it back out [9,11,12]. This preparation scheme reduces the number of undetectable logical error mechanisms, but has a higher overall circuit depth. Stabilizer measurements are also required in this case to detect data qubit errors.

Figure 13(a) provides a benchmark of the different injection schemes described above (excluding Lao-Criger injection). We plot the performance of $f = 3$ cultivation in the presence of uniform depolarizing noise with different strengths p . We observe a consistent hierarchy of performance. For example, for all p , hook injection has the highest initial expected attempts, and the optimized unitary encoder has the lowest.

Table III summarizes the relative performance of each injection scheme at $p = 0.001$. We focus on two quantities: the number of initial expected attempts and the lowest logical error rate obtained when fully post-selecting

TABLE III. Comparison of injection schemes under uniform depolarizing noise of strength $p = 0.001$. Quantified by the minimum expected attempts, and the error floors at full gap-based postselection. Logical error rates have an uncertainty of roughly 1×10^{-7} .

Injection scheme	Min. expected attempts	Min. logical error rate
Hook injection	1.62	8×10^{-7}
Basic unitary encoder	1.49	9×10^{-7}
Optimized unitary encoder	1.46	13×10^{-7}
Folding unitary encoder	1.54	7×10^{-7}

on the complementary gap. We note that that, within error bars, the folding unitary encoding circuit reaches a logical error rate of 7×10^{-7} . This is the required value provided in Ref. [8] for factoring of large integers. However, it has a higher initial expected attempts compared to other unitary encoders.

The naïve unitary encoding circuit has a balanced tradeoff between logical error rate and expected attempts. Therefore we use it for our simulations in the main text. This choice may be amended based on the constraints of a given architecture, along with logical error rates required of the given application. An injection scheme that achieves optimal performance remains the subject of further investigation. Finally, we note that unitary-based injection schemes do not require mid-cultivation classical feedback. As such, they may be more attractive to physical architectures where measurement and subsequent classical feedback is slow.

3. Logical measurement

In this section, we discuss the various choices related to measuring the $\bar{H}_{XY}$ operator after it has been injected. It is important to perform this check efficiently while maintaining the fault distance of the protocol.

In this work, logical checks are performed with ancillary GHZ states. The size of the ancilla system can impact the fault distance, as well as the expected spacetime volume. For example, using only a single ancilla qubit for logical measurements is insufficient as a single error on it may propagate to multiple errors on data qubits, leading to logical errors. On the other hand, an ancillary $\text{GHZ}(n)$ state permits fault-tolerant logical measurements in an n -qubit code because any ancilla error does not propagate to more than one data qubit. However, this requires an unacceptably large spacetime volume for the fault-tolerant preparation of the large $\text{GHZ}(n)$ state.

Here we show that a $\text{GHZ}(3)$ state is sufficient for measuring the $\bar{H}_{XY}$ operator of $\text{Reg}(3)$ with a fault distance 3. Similarly, $\text{GHZ}(5)$ states are sufficient for $\text{Reg}(5)$ with a fault distance 5.

We first prove that in the absence of errors, a GHZ state of any size can be used to measure $\bar{H}_{XY}$. From Eq. (B2), we have that a $\bar{H}_{XY}$ consists of single qubit gates $UZU^\dagger$ for some U on diagonal qubits and CZ gates on each pair of mirrored qubits. A controlled- $\bar{H}_{XY}$ gate therefore requires one CZ gate for each diagonal qubit and one CCZ gate for each pair of mirrored qubits. When an ancilla system is prepared in an l -qubit $\text{GHZ}(l)$ state stabilized by $X^{\otimes l}$, these CZ and CCZ gates can be implemented with their control on any of the ancilla qubits. This holds because, after all CZ and CCZ gates are applied, the stabilizer becomes

$$X^{\otimes l} \otimes \left(\prod_{i=0}^{2d-2} Z_{D_i} \right) \otimes \left(\prod_{q \in \Delta} \text{CZ}_{q, \tau(q)} \right), \quad (\text{B4})$$

which is independent of the particular choice of control qubits of each gate. Finally, by applying a GHZ decoding circuit on the ancilla qubits, $X^{\otimes l}$ is mapped to a single-qubit X on one ancilla qubit. Measuring this in the X basis yields the logical measurement outcome of $\bar{H}_{XY}$.

a. Data qubit partitions

To achieve a desired fault distance f , the logical measurement circuit has to be designed such that any physical error with weight less than f does not result in an undetectable logical error. This requires (i) a sufficiently large ancilla GHZ state $l \geq f$, so that each ancilla qubit couples to only a limited number of data qubits, and (ii) a carefully chosen assignment of ancilla-data couplings, which prevents harmful correlated errors on the data. We present this assignment as a *partition* of the data qubits into l disjoint subsets, with each subset coupled to a distinct qubit of an ancillary GHZ(l) state. These partitions are represented by different colors in Figs. 6 and 10.

The partitions are constructed such that no combination of $\lfloor f/2 \rfloor$ subsets fully supports a logical operator. For example, in Fig. 6 no single partition is connected to a pattern of data qubits that support a logical operator on $\text{Reg}(3)$. Otherwise, an error on that partition's ancilla could propagate to a logical error on the data, and a subsequent error on the same ancilla could cancel the first one. This combination of two errors may introduce an undetectable logical error, lowering the fault distance of the logical check circuit. In addition, it is desirable that the partition balances the number of gates on each ancilla qubit, which enables a shallow circuit and helps mitigate idle errors.

b. GHZ size

While a GHZ(3) ancilla is sufficient to achieve a logical measurement circuit with fault distance 3, we also compare its performance with that of a larger GHZ(5) ancilla in Fig. 13(b). Using a larger ancilla reduces the circuit depth required during logical measurements, but comes at the expense of increased spacetime volume to prepare the ancilla fault-tolerantly. In particular, we must include a flag qubit to ensure the fault-tolerant preparation of GHZ(5). The results in Fig. 13(b) include shots discarded purely due to flagged GHZ(5) preparation, and thus may overestimate the true overall discard rate. Compared to the smaller GHZ(3), GHZ(5) offers little advantage in terms of the expected spacetime volume. Thus, we choose GHZ(3) in our protocol due to its lower overhead.

c. “Double-check” vs “single-check”

Reference [9] uses a “double-check to measure a transversal Clifford operator twice using a single ancilla product state. This double-check provides two bits of information about the logical eigenvalue, along with error flags. Here, we simply choose to use two GHZ states to

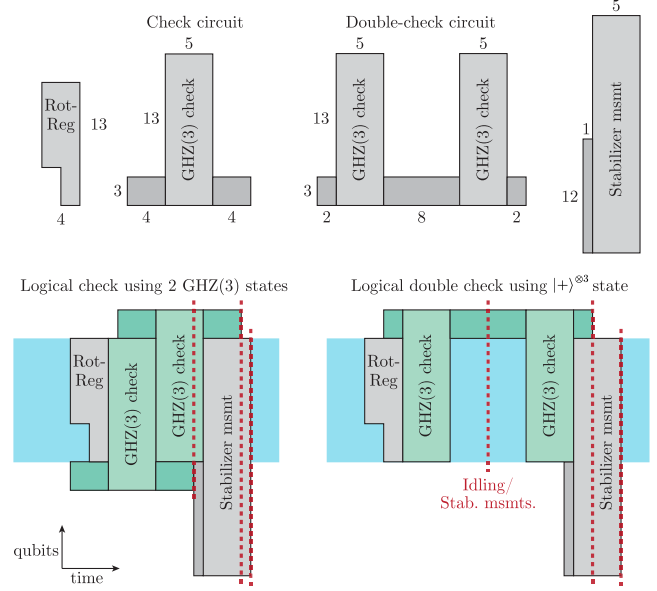


FIG. 14. Spacetime volume of logical checks. We show the space (vertical) and time (horizontal) cost of different cultivation steps. The top row shows the space and time costs of the preliminary gadgets that compose the logical checks. The lower left shows the spacetime volume of two sequential single-checks” with pre-prepared GHZ(3) states, showing efficient pipelining. The lower right shows the spacetime volume of the double-check” strategy of Ref. [9]. Red dashed lines represent postselection on measurement outcomes.

check this operator (once per GHZ state). We obtain more efficient pipelining of operations since we are not required to wait for decoding and re-encoding of the ancillary system. Consequently, we bypass the cost of either having a longer idling time on the initial code during which errors can build up, or having to do more stabilizer measurements during this time.

Figure 14 shows a spacetime volume comparison of these alternative check strategies. The prepared GHZ states allow more compact cultivation, albeit at the cost of using more qubits.

d. Higher fault distances

Figure 10 shows a partition of a $\text{Reg}(5)$ code into five subsets that enables fault-tolerant $\bar{H}_{XY}$ measurements. For the partition to be valid, it must prevent any weight-4 physical error from propagating into an undetectable logical error on the data. To guarantee $f = 5$, we impose the condition that the union of any two subsets must not contain the full support of a logical operator. Similar to the $\text{Reg}(3)$ code, the most dangerous scenario arises when two errors occur on two ancilla qubits, propagate to the data, and then two additional errors on the same ancillae cancel the earlier ones, rendering the error undetectable.

For higher fault distances, we must also increase the number of measurements of the fold-transversal

$\bar{H}_{XY}$ operator. Specifically, we must measure $\bar{H}_{XY}$ at least $f - 1$ times for a fault distance f . To see why, consider the following adversarial error:

1. A single error during injection may cause a logical error, as the injection circuit is not fault-tolerant.
2. A measurement error on the outcome of $\bar{H}_{XY}$ may hide this logical error, yielding $+1$ and leading to the belief that no error has occurred.

If we only measure the logical operator $t < f - 1$ times, then we have a logical error on the code with only $t + 1$ errors. To ensure that logical errors only occur after f errors, must measure the logical operator $f - 1$ times. For $f = 5$, we can achieve this by doing two checks at **Reg**(3), and two checks at **Reg**(5). This design reduces expected spacetime volume compared to checking four times at **Reg**(5).

4. Pre-escape stabilizer measurement

In this section, we discuss different options for extracting stabilizer information after the logical checks have been performed, but prior to the escape stage. This is used for additional postselection in order to achieve low error rates [44]. For example, in our $f = 3$ protocol a single round of stabilizer measurements is needed after the logical checks at **Reg**(3). However, it is not strictly necessary to perform this check at **Reg**(3), and unitary circuits may be used to transform to another code where stabilizer measurements improve performance.

Figure 13(c) shows the performance of cultivation for three different choices of code in which to perform these “pre-escape” stabilizer measurements. We consider the following options: (i) stabilizer measurements at **Reg**(3) using a depth-4 circuit, (ii) stabilizer measurements at **Rot**(3) using a depth-2 circuit, and (iii) growing to **Rot**(5) and then measuring stabilizer using a depth-6. Measuring at **Reg**(3) offers the best performance in terms of expected attempts and logical error rate.

5. Escape strategy

In this section, we discuss different escape strategies. The primary goal of the escape stage is to expand to a larger code of distance $d_2 > 2f$. This is done as quickly as possible in order to preserve the error scaling achieved by cultivation. A secondary goal is to collect stabilizer information from this larger code in order to post-select based on the complementary gap. Note that d_2 is the minimum code distance necessary to protect the magic state. This is often going to be different to the final code distance d_{fin} required for use in a quantum algorithm. It is most important to grow to d_2 rapidly; the subsequent growth strategy to d_{fin} is done after all postselection.

Reference [9] uses stabilizer measurement rounds to grow directly to d_{fin} . This stabilizer-based escape is

expensive in terms of spacetime volume, since several gate and measurement layers at d_{fin} are required. Ref. [10] also uses stabilizer-based escape, but only to d_2 , thus overlooking the possible overhead associated with growing to d_{fin} .

Recently, efficient unitary encoders have been proposed that enable conversion from small- to large-distance rotated surface codes in low depth [15,36]. While efficient in terms of spacetime volume, a unitary escape strategy directly to d_{fin} does not by itself provide the stabilizer information necessary for gap-based-postselection. Nevertheless, we can examine its “theoretical” performance.

Figure 13(d) provides a comparison of unitary-based and stabilizer-based escape strategies. To benchmark purely unitary escape, we use a perfect measurement round at the end of growth to compute a complementary gap and obtain a logical error rate. Though this is unphysical due to the perfect round of measurement, unitary escape has both a lower spacetime overhead and a lower error floor than stabilizer-based escape. This demonstrates the limitations imposed by escape. We also remark that in a setting where measurements are noisier than gates, such as in neutral atoms, one may further benefit from using unitary escape instead of measurement-based escape.

In this work, we utilize a *three-stage escape* strategy that is a combination of unitary- and stabilizer-based growth steps. After the logical checks in **Reg**, we unitarily grow to **Rot**. Then, we use stabilizer-based escape to grow to **Rot**(d_2), where we post-select on the complementary gap. Finally, we unitarily grow to **Rot**(d_{fin}). In doing so, we avoid the large spacetime overhead of stabilizer measurement rounds at d_{fin} .

APPENDIX C: FAULT DISTANCE 5

In Sec. II we presented a cultivation protocol for fault distance $f=3$. Here, we extend our protocol to a larger fault distance $f=5$. This variant of our protocol reaches a lower logical error rate by further fault-tolerantly checking the $\bar{H}_{XY}$ eigenstate at **Reg**(5). Similar to previous works, we find that a simple modification that changes our $f=5$ scheme to $f=4$ achieves comparable logical error rate with lower overhead [9].

We begin by outlining our end-to-end $f=5$ protocol. The initial stages are identical to that of $f=3$. Specifically, from injection up to the first two H_{XY} checks at **Reg**(3). Following this, we add the following steps:

1. Unitarily grow to **Rot**(5).
2. Perform r_1 stabilizer measurement rounds.
3. Unitarily grow to **Reg**(5) and perform two additional logical $\bar{H}_{XY}$ checks using a **GHZ**(5) ancilla.
4. Three-stage escape to **Rot**(d_{fin}) which involves r_2 stabilizer measurement rounds at **Reg**(5).

Figure 2(a) shows the logical error rate of $f = 5|Y\rangle$ state cultivation under uniform depolarizing noise model with a strength $p = 0.001$. Setting $(r_1, r_2) = (2, 0)$ we can reach a logical error rate of 10^{-9} in fewer than 10 expected attempts. This is an improvement over previous works that require approximately 20 expected attempts in order to reach this regime. In the following paragraphs, we reiterate the protocol from end-to-end for clarity.

Injection—We use a unitary encoding circuit to inject the $|\bar{H}_{XY}\rangle$ state into $\text{Rot}(3)$. This is followed by a round of stabilizer measurements. We post-select on any non-trivial stabilizer measurement outcomes.

Cultivation—We use a unitary circuit to transform to $\text{Reg}(3)$ and measure $\bar{H}_{XY}$ twice with a GHZ(3) ancilla. We then grow to $\text{Rot}(5)$ and perform r_1 rounds of stabilizer measurements.

Next, we grow to $\text{Reg}(5)$ and measure $\bar{H}_{XY}$ twice again, except this time with a GHZ(5) ancilla. We show the circuit for this check in Fig. 10 (and the equivalent $\bar{Y}$ check in Fig. 11). The GHZ(5) ancilla uses an extra flag qubit such that no single physical error leads to a weight-2 error on the GHZ(5) state.

At this stage, we may choose to measure stabilizers of $\text{Reg}(5)$ surface code r_2 times to further lower the LER before escape. Through numerical search for the minimum-weight error that causes a logical error, we observe that $r_1 \geq 3$ and $r_2 \geq 1$ are necessary to achieve $f = 5$ scaling. This is because high-weight errors may appear in $\text{Rot}(5)$ arising from a single error before growth at $\text{Reg}(3)$. These errors can remain undetected by all subsequent $\bar{H}_{XY}$ checks in $\text{Rot}(5)$. They may then grow to undetectable logical operators in future steps, reducing the fault distance when $r_1 < 3$.

Escape—We use a three-stage escape strategy. First, we unitarily grow to $\text{Rot}(9)$. Next, we use stabilizer-based growth to $\text{Rot}(d_{\text{mid}})$, using r_{mid} rounds of stabilizer information for gap-based postselection. Finally, we unitarily grow to $\text{Rot}(d_{\text{fin}})$.

For our numerical simulations, we perform $r_{\text{mid}} = 5$ rounds of stabilizer measurements on an intermediate code of distance $d_{\text{mid}} = 11$. We choose a final code distance $d_{\text{fin}} = 15$ to be consistent with the choice in Ref. [9]. For the third stage of escape we use the unitary growth circuit proposed in Ref. [26]. If we were to use the same unitary growth circuit as in the $f = 3$ protocol, the code distance would double in a single step, taking $d_{\text{mid}} = 11$ to 21, which exceeds our target final size.

Figure 15 shows the performance of $f = 5$ cultivation for different numbers of stabilizer measurement rounds (r_1, r_2) . We choose $(2, 0)$ for our protocol. It has comparable error floor to $(2, 1)$ and $(3, 1)$ schemes, but lower overhead as it involves less stabilizer measurement rounds. Through exhaustive numerical search, the comparable error floors of these alternatives are attributed to an extremely low number of weight-4 logical error

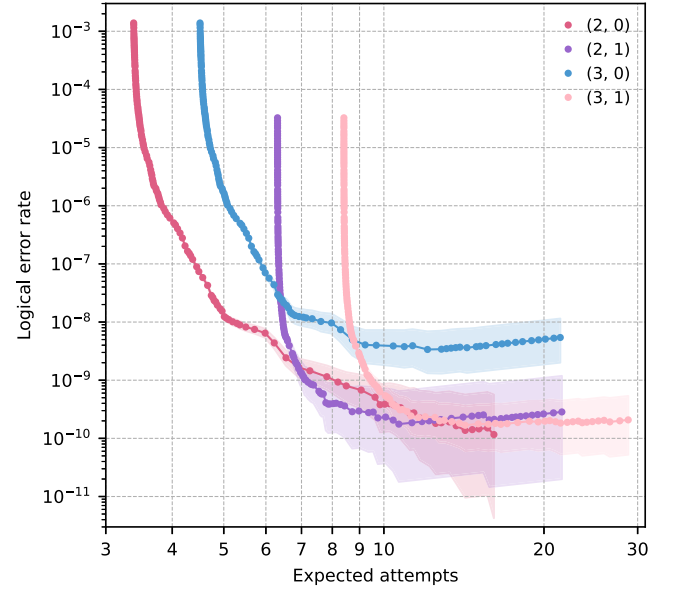


FIG. 15. Varying stabilizer measurements (r_1, r_2) for fault distance $f = 5$ cultivation. Here, r_1 is the number of stabilizer measurement rounds at $\text{Rot}(5)$, and r_2 the number of rounds at $\text{Reg}(5)$.

mechanisms in $(2, 0)$. We also disregard the $(3, 0)$ scheme, because it has a higher error floor of compared to the $(2, 0)$ scheme. Since both $(3, 0)$ and $(2, 0)$ have a fault distance of 4, we suspect that the increased error floor is due to a greater number of error mechanisms introduced by the additional $\text{Rot}(3)$ check at $(3, 0)$. Note that all data shown here include shots discarded due to flags during GHZ(5) state preparation. Thus, it may overestimate the number of shots required for cultivation using a set of pre-prepared GHZ(5) states by about $\sim 1\%$.

APPENDIX D: HANDOFF SIMULATION

In this section, we describe our handoff simulation that combines both state vector and stabilizer simulators. We use this technique to simulate the logical error rate of $|\bar{H}_{XY}\rangle$ magic state cultivation scheme. This is to be contrasted with previous work that instead simulates $|\bar{Y}\rangle$ state cultivation as a proxy [9–12]. Our handoff simulation directly samples noisy logical $|\bar{H}_{XY}\rangle$ states produced by the full protocol without approximations, while remaining tractable for fault distance $f = 3$.

We achieve this by splitting our circuit into two parts, which are simulated by a state vector simulator and a stabilizer simulator, respectively. The state vector simulation starts with the injection of $|\bar{H}_{XY}\rangle$ and ends after the completion of the $\bar{H}_{XY}$ measurements. This part of the protocol contains all the physical non-Clifford operations and is supported on a relatively small number of qubits. The stabilizer simulation starts from the stabilizer measurements after the $\bar{H}_{XY}$ measurement and ends after the completion of the escape stage. This part of the circuit

contains only Clifford gates and can be simulated efficiently with a stabilizer simulator such as `Stim`.

In Appendix D 1 we describe the steps of the handoff simulation, and how we combine the state vector and stabilizer segments. Then, in Appendix D 2 we prove that the simulation exactly samples the noisy $|\bar{H}_{XY}\rangle$ states. Finally, in Appendix D 3 we discuss the numerical results and compare the cultivation of $|\bar{Y}\rangle$ states vs $|\bar{H}_{XY}\rangle$ states. We remark here that our approach can be considered as an adaptation of the splitting technique from Ref. [45] to magic state cultivation.

1. Procedure

In this section, we describe the steps of our handoff simulation for fault distance $f = 3$ cultivation.

Each step involves noisy operations unless specified otherwise. The details of our cultivation protocol are summarized in Fig. 1 and described in Sec. II.

a. State vector simulation

First, we use a state vector simulation to extract the noisy state of `Reg(3)` immediately following $\bar{H}_{XY}$ checks. This is achieved by means of a noiseless round of stabilizer measurements. These are *not* the stabilizer measurements used for postselection. Instead, they are a means of extracting the logical state of the code, as well as the state of the data qubits. The simulation proceeds as follows.

1. Simulate the cultivation protocol up to and including the $\bar{H}_{XY}$ checks. Post-select on any nontrivial measurement outcomes. Namely, the outcomes of the stabilizer measurements during injection and the $\bar{H}_{XY}$ checks during cultivation.
2. Obtain the state vector $|\psi\rangle \in \mathbb{C}^{2^{13}}$ that describes the thirteen data qubits of `Reg(3)`.
3. Perform a round of noiseless stabilizer measurements and obtain the bit string $s \in \{0, 1\}^{12}$ corresponding to the syndromes.
4. Obtain the (normalized) state vector $|\psi_s\rangle \in \mathbb{C}^{2^{13}}$ by projecting $|\psi\rangle$ onto the subspace defined by s . This is the state of the data qubits after measuring the ancilla qubits and obtaining the measurement outcomes s .
5. Construct a *Pauli destabilizer* D_s by multiplying strings of physical X and Z operators that cancel all nontrivial syndromes in s by connecting each nontrivial syndrome to the bottom or right boundaries of the code.
6. Rewrite $|\psi_s\rangle$ in the form of a logical state vector multiplied by the destabilizer D_s :

$$|\psi_s\rangle = D_s(\alpha|0_a\rangle + \beta|1_a\rangle), \quad (D1)$$

where $|0_a\rangle$ and $|1_a\rangle$ are the logical basis states of `Reg(3)` and the coefficients are given by

$$\alpha = \langle 0_a | D_s^\dagger | \psi_s \rangle, \quad (D2)$$

$$\beta = \langle 1_a | D_s^\dagger | \psi_s \rangle. \quad (D3)$$

Remark 1: In practice, steps 2 and 3 above can be condensed. This is achieved by applying the inverse of the unitary encoding circuit U_a . Applying $U_a^\dagger$ maps a representative of the logical operators $\bar{X}_a$ and $\bar{Z}_a$ onto the X and Z operators of a single data qubit. Simultaneously, it maps the product of stabilizer generators onto single-qubit Z operators on the twelve other data qubits. Therefore, with appropriate post-processing, the syndrome bit string s and the logical coefficients α and β can be determined by measuring the thirteen data qubits.

Remark 2: The noiseless stabilizer measurement outcomes s are used only for the purposes of handoff; we do not use s for decoding.

b. Stabilizer simulation

At this point, we handoff the results to a stabilizer simulator. The purpose of the stabilizer simulator is to sample logical Pauli errors that may occur after the $\bar{H}_{XY}$ checks. As such, the stabilizer simulator only takes the destabilizer D_s as input from the state-vector simulator. The logical state vector coefficients α and β are stored in order to reconstruct the noisy magic state at the end of the protocol. The simulation proceeds as follows.

1. Prepare the logical state $|0_a\rangle \in \mathbb{C}^{2^{13}}$ of the `Reg(3)` code using a noiseless unitary encoding circuit [46]. Then, apply the destabilizer D_s obtained from the previous state vector simulation.
2. Proceed with the remainder of the protocol. Perform a noisy round of stabilizer measurements of `Reg(3)` and post-select for the trivial syndrome. If successful, proceed with escape, obtaining measurement outcomes g along the way.
3. Terminate the simulation at the end of MSC with a round of noiseless stabilizer measurements on `Rot( $d_{\text{fin}}$ )` and obtain the measurement outcomes q .
4. Propagate D_s and the physical Pauli errors that were sampled in the stabilizer simulation to the end of line on each physical qubit in software.
5. Use two bits x and z to record whether the propagated error anticommutes with the logical operators of `Rot( $d_{\text{fin}}$ )`, denoted $\bar{X}_b$ and $\bar{Z}_b$. Here, True is recorded as 1.
6. Decode using the syndrome information obtained from the stabilizer measurements made during escape, g and q . Recall that this is another point of postselection, depending on the confidence of the gap-based decoder. From this, we obtain the correction operator R .

7. Store two bits x_R and z_R to record whether the recovery R anticommutes with $\bar{X}_b$ and $\bar{Z}_b$. Again, with **True** being recorded as 1.

c. Magic state reconstruction

Finally, we combine the two simulations. The noisy magic state sampled by the handoff simulation is

$$|\tilde{H}_{XY}\rangle = \bar{X}_b^{z+x_R} \bar{Z}_b^{x+x_R} (\alpha|0_b\rangle + \beta|1_b\rangle), \quad (\text{D4})$$

where $|0_b\rangle$ and $|1_b\rangle$ are the logical basis states of $\text{Rot}(d_{\text{fin}})$. Recall that α and β are obtained from the state vector simulation, while the bits x , z , x_R , and z_R are obtained from the stabilizer simulation. We declare whether a logical error has occurred by comparing the noisy magic state to the ideal magic state $|\bar{H}_{XY}\rangle = (|0_b\rangle + e^{i\pi/4}|1_b\rangle)/\sqrt{2}$. A logical error has occurred when we find that

$$|\langle \bar{H}_{XY} | \tilde{H}_{XY} \rangle|^2 < 1. \quad (\text{D5})$$

2. Working principle

In this section we prove that our handoff simulation exactly samples the noisy $|\tilde{H}_{XY}\rangle$ states without approximation. First, we show that inserting a round of fictitious” and noiseless stabilizer measurements after the $\tilde{H}_{XY}$ checks on **Reg**(3) does not disturb the outputs of the noisy cultivation circuit. Intuitively, this is because any state vector collapse caused by the noiseless measurements happens regardless due to the *actual* noisy stabilizer measurements performed shortly thereafter.

Consider the state $|\psi\rangle$ of the data qubits following the logical checks. It is convenient to rewrite this state as

$$|\psi\rangle = \sum_s c_s |\psi_s\rangle, \quad (\text{D6})$$

where $|\psi_s\rangle$ is the state after projection onto the subspace defined by the syndrome bit string s , and c_s is a complex coefficient. The noisy syndrome extraction circuit may be expressed in the form

$$\tilde{C} = P_{\text{dat}} P_{\text{anc}} C, \quad (\text{D7})$$

where C is the noiseless circuit, and P_{dat} and P_{anc} are Pauli errors on the data and ancilla qubits, respectively. The noiseless syndrome extraction circuit C maps the syndromes s onto the ancilla qubits:

$$C(|\psi\rangle \otimes |0\rangle^{\otimes n_{\text{anc}}}) = \sum_s c_s (|\psi_s\rangle \otimes |s\rangle). \quad (\text{D8})$$

The noisy circuit $\tilde{C}$ has the following action on the data and ancilla qubits:

$$\begin{aligned} \tilde{C}(|\psi\rangle \otimes |0\rangle^{\otimes n_{\text{anc}}}) &= P_{\text{dat}} P_{\text{anc}} \sum_s c_s (|\psi_s\rangle \otimes |s\rangle) \\ &= \sum_s c_s (P_{\text{dat}} |\psi_s\rangle \otimes P_{\text{anc}} |s\rangle). \end{aligned} \quad (\text{D9})$$

Upon measurements of the ancilla qubits, we obtain the state $P_{\text{dat}} |\psi_s\rangle \otimes P_{\text{anc}} |s\rangle$ with probability $|c_s|^2$.

Now we will show that the handoff circuit with noiseless stabilizer measurements preserves the same state and statistics as Eq. (D9). As before, the initial state is $|\psi\rangle \otimes |0\rangle^{\otimes n_{\text{anc}}}$. First, we perform the round of noiseless stabilizer measurements that are used for handoff. From Eq. (D8), measurement of the ancilla qubits projects onto the state $|\psi_s\rangle \otimes |s\rangle$ with probability $|c_s|^2$. Then, we apply the noisy syndrome extraction circuit $\tilde{C}$:

$$\begin{aligned} \tilde{C}(|\psi_s\rangle \otimes |0\rangle^{\otimes n_{\text{anc}}}) &= P_{\text{dat}} P_{\text{anc}} (|\psi_s\rangle \otimes |s\rangle) \\ &= P_{\text{dat}} |\psi_s\rangle \otimes P_{\text{anc}} |s\rangle, \end{aligned} \quad (\text{D10})$$

which is the same state as Eq. (D9). The noiseless circuit C , followed by ancilla measurement, and then the noisy circuit $\tilde{C}$ result in the same state $P_{\text{dat}} |\psi_s\rangle \otimes P_{\text{anc}} |s\rangle$ with the same probability $|c_s|^2$. Therefore, the noiseless stabilizer measurements do not disturb the outputs of the noisy stabilizer measurements in the cultivation stage, nor the outputs of the rest of the circuit.

After the stabilizer measurements we are left with a state $|\psi_s\rangle$ that is confined to a two-dimensional subspace defined by the syndrome bit string s . A basis for this two-dimensional subspace is $D_s|0_a\rangle$ and $D_s|1_a\rangle$, where D_s is the Pauli destabilizer defined in step 5 of the state vector simulation.

We can write the state as

$$|\psi_s\rangle = D_s(\alpha|0_a\rangle + \beta|1_a\rangle), \quad (\text{D11})$$

where α and β are given by Eqs. (D2) and (D3). In this way, the syndrome bit string s and the complex coefficients α and β completely determine state $|\psi_s\rangle$.

We now wish to simulate the evolution $|\psi_s\rangle$ through the rest of the cultivation protocol. While $|\psi_s\rangle$ is not guaranteed to be a stabilizer state, we will see that the measurement outcome statistics of the stabilizer simulation are unchanged if the state $|\psi_s\rangle$ is replaced by a stabilizer state $D_s|0_a\rangle$. This is because the remaining Clifford circuit is constructed to measure either stabilizers or operators that anticommute with stabilizers, and not logical operators. The sampling of measurement outcomes can be achieved in the stabilizer simulator by preparing $D_s|0_a\rangle$, or indeed any logical Pauli state.

The final noisy magic state is obtained from Eq. (D4). Note that this depends on the logical coefficients α and β (extracted from the state vector simulation), and the bits x ,

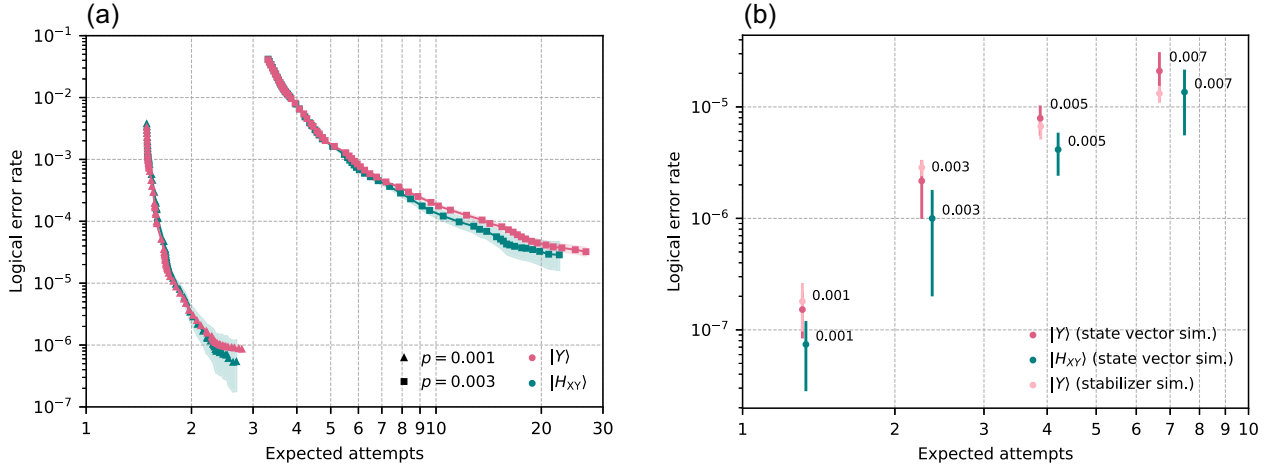


FIG. 16. Comparing $|\bar{Y}\rangle$ and $|\bar{H}_{XY}\rangle$ state cultivation (a) including the escape stage and (b) excluding the escape stage. Simulations use the fault distance $f = 3$ protocol, with uniform depolarizing noise of strength p . When we exclude escape, the two states have different logical error rates.

z , x_R and z_R (extracted from the stabilizer simulation). Only the syndrome bit string s from $\text{Reg}(3)$ is passed between the two simulations, in the form of the destabilizer D_s . Finally, we point out that if the non-Clifford gates transform Pauli errors in a simple way, then magic state preparation can be exactly simulated on a stabilizer simulator. Specifically, by rewriting the magic state density matrix into a linear combination of the density matrices of pure stabilizer states [47].

3. $|\bar{Y}\rangle$ vs $|\bar{H}_{XY}\rangle$ H cultivation

Here, we further compare $|\bar{Y}\rangle$ and $|\bar{H}_{XY}\rangle$ state cultivation for our $f = 3$ scheme. Figure 16(a) shows the logical error rate of cultivation for the two states under uniform depolarizing noise of different strengths p . We find that the relationship between the LER of cultivation and the expected attempts is very similar for the two different states.

In previous work, a truncated state vector simulation was used to estimate the relative performance of Clifford and non-Clifford state cultivation [9]. In these simulations, the injection and cultivation stages are truncated by a perfect round of stabilizer measurements and there is no escape stage. Figure 16(b) shows the results of such simulations for our protocol. We find that $|\bar{Y}\rangle$ and $|\bar{H}_{XY}\rangle$ state cultivation performs differently in this setting. Additionally, we observe that, at higher p values magic state cultivation has a lower logical error rate and correspondingly more expected attempts than the Clifford state. This behavior is in contradiction to what is seen in previous work [48]. To verify that our truncated state vector simulation is performing reliably, we have also used a stabilizer simulator for the $|\bar{Y}\rangle$ state. This stabilizer simulation agrees with the state vector simulation.

Taken as a whole, these simulations suggest that the logical error rate of the protocol is dominated by error

mechanisms during escape. It follows that the truncated simulations are thus not a reliable indicator of the true end-to-end performance of cultivation protocols dominated by error mechanisms during escape.

APPENDIX E: NUMERICAL SIMULATIONS

In this section, we present details as to how we obtain and plot the data we show in the main text. Note that sample code and circuits are included at Ref. [41]. Data were generated using `sinter` and `PyMatching` [49,50].

1. Complementary gap plots

Data from other schemes—We obtained the data for Fig. 2 from the Zenodo repositories of Refs. [9,10]. Note that Ref. [10] uses 2D soft outputs in terms of the parameters $(\phi_{\mathbb{R}P^2}, \phi_{bd})$, instead of the 1D complementary gap. For plotting purposes we convert the data to a 1D complementary gap using the function $G = \phi_{\mathbb{R}P^2} + 0.2\phi_{bd}$. To the best of our knowledge, these converted data are near-indistinguishable from the optimized points chosen from the 2D space shown in the original manuscript.

Error bars—The error bars shown are calculated using an inbuilt `sinter` function, highlighting a region within a factor of 1000 of the maximum likelihood. Note that for handoff we performed 50 fewer state vector shots, and oversampled the Clifford circuits on these outputs.

Final, perfect measurement—Our simulations include a final perfect round of measurements in an actual MSC protocol, such perfect syndrome measurement is not present, and thus cannot contribute to the computation of the complementary gap G . In order to verify that our results do not rely on this noiseless information for the computation of the complementary gap, we tested different numbers of stabilizer measurement rounds during escape. Note that the complementary gap plots for only unitary

TABLE IV. Noise models. Channels are applied after the unitary operation or reset, or before measurement.

Abbreviation	SD6	PM1 or PM5
Name	Standard depolarizing	Physically motivated
Noisy gateset	$CX(p)$	$CZ_{\text{local}}(p)$
	$CY(p)$	$CZ_{\text{nonlocal}}(p \text{ or } 5p)$
	$CCZ(p)$	$CCZ(5p)$
	$\text{AnyClifford}_1(p)$	$\text{AnyClifford}_1(p/10)$
	$\text{Init}_Z(p)$	$\text{Init}_Z(p)$
	$M_Z(p)$	$M_Z(p)$
	$\text{Idle}(p)$	$\text{Idle}(0)$

growth-based escape solely rely on this perfect round and are thus unphysical.

Baseline scheme used in Figs. 13(c) and 13(d)—In these figures, the state preparation was achieved by means of

hook injection, instead of a unitary encoder used in other simulations. Due to the modularity of MSC, we expect that the relative hierarchy of performance between the curves should not change if we were to use a unitary encoder.

2. Noise models

In Table IV, we show the noise models used in this work. A noisy unitary operation on n qubits is associated with the relevant depolarizing Pauli channel $\{I, X, Y, Z\}^{\otimes n} \setminus I^{\otimes n}$ occurring with the stated probability. For measurement and reset in the Z basis, we apply a probabilistic X error before and after the operations, respectively.

Note added: In a previous version of this manuscript, reported $f = 5$ data incorrectly did not have idling errors at a subset of circuit locations. Specifically, idles were excluded on $\text{Reg}(3)$ qubits that did not participate in

TABLE V. Spacetime volume comparison for $f = 3$ for different protocols. Each protocol is divided into multiple stages with their corresponding overhead. Dot-product notation indicates qubit $\cdot$ steps, while expressions with an explicit “ $\times$ ” indicate repeated execution of the circuit. For example, $A \times (QT)$ denotes a circuit with Q qubits and T time steps repeated A times. Note that in all circuits, the preparation of a $|+\rangle$ state is implemented as preparation of $|0\rangle$ followed by a Hadamard gate, taking two steps. Similarly, each X -basis measurement is counted as two steps.

Protocol	Stage	Spacetime volume	Description
This work	Inject	$9 \cdot 6 + 17 \cdot 8$	Unitary encoding of $\text{Rot}(3)$.
	Grow	$13 \cdot 4$	Unitary growth to $\text{Reg}(3)$.
	Check $\times 2$	$2 \times [2(3 \cdot 4) + 16 \cdot 3]$	$\bar{H}_{XY}$ check; includes encoding/decoding $\text{GHZ}(3)$.
	Stabilize	$25 \cdot 8$	Stabilizer measurements of $\text{Reg}(3)$.
	Grow	$25 \cdot 4$	Unitary growth to $\text{Rot}(5)$.
	Stabilize $\times 3$	$3 \times (97 \cdot 8)$	Stabilizer-based growth to $\text{Rot}(7)$.
	Grow	$169 \cdot 4$	Unitary growth to $\text{Rot}(13)$.
[9]	Inject	$13 \cdot 15$	Stages defined as in Fig. 15 (left) of Ref. [9].
	Stabilize	$14 \cdot 10$	
	Check	$2 \times (13 \cdot 7)$	
	Stabilize	$3 \times (337 \cdot 12)$	
	Escape and stabilize	$3 \times (337 \cdot 12)$	
[10]	Inject	$9 \cdot 3 + 19 \cdot 9$	Stages defined as in Fig. 14(a) of Ref. [10].
	Stabilize	$19 \cdot 9$	
	Morph	$15 \cdot 5$	
	Check	$2 \times (24 \cdot 9)$	
	Morph and expand	$343 \cdot 10$	
[11]	Stabilize	$3 \times (337 \cdot 8)$	Stages defined as in Fig. 1(a) of Ref. [11].
	Inject	$9 \cdot 10$	
	Stabilize	$17 \cdot 8$	
	Deform	$21 \cdot 10$	
	Check	$2 \times (26 \cdot 12)$	
	Undeform	$26 \cdot 10$	
[12]	Stabilize	$3 \times (337 \cdot 8)$	Stages defined as in Ref. [12].
	Inject	$9 \cdot 10$	
	Stabilize	$17 \cdot 8$	
	Deform	$21 \cdot 10$	
	Check	$2 \times (26 \cdot 12)$	
	Undeform	$26 \cdot 10$	

TABLE VI. Spacetime volume comparison for $f = 5$ for different protocols. Each protocol is divided into multiple stages with their corresponding overhead. Notation identical to that of Table V. (*Note that this stage uses the unitary growth circuit from Ref. [26]).

Protocol	Stage	Spacetime volume	Description
This work	Inject	$9 \cdot 6 + 17 \cdot 8$	Unitary encoding of $\text{Rot}(3)$.
	Grow	$13 \cdot 4$	Unitary growth to $\text{Reg}(3)$.
	Check $\times 2$	$2 \times [2(3 \cdot 4) + 16 \cdot 3]$	$\tilde{H}_{XY}$ check; includes encoding/decoding $\text{GHZ}(3)$.
	Grow	$25 \cdot 4$	Unitary growth to $\text{Rot}(5)$.
	Stabilize $\times 2$	$2 \times (49 \cdot 8)$	Stabilizer measurements of $\text{Rot}(5)$.
	Grow	$49 \cdot 4$	Unitary growth to $\text{Reg}(5)$.
	Check $\times 2$	$2 \times (6 \cdot 8 + 54 \cdot 5 + 5 \cdot 5)$	$\tilde{H}_{XY}$ check; includes encoding/decoding $\text{GHZ}(5)$.
	Grow	$81 \cdot 4$	Unitary growth to $\text{Rot}(9)$.
	Stabilize $\times 5$	$5 \times (241 \cdot 8)$	Stabilizer-based growth to $\text{Rot}(11)$.
	Grow	$225 \cdot 4$	Unitary growth* to $\text{Rot}(15)$.
[9]	Inject	$13 \cdot 15$	Stages defined as in Fig. 15 (left) of Ref. [9].
	Stabilize	$14 \cdot 10$	
	Check	$2 \times (13 \cdot 7)$	
	Stabilize	$3 \times (38 \cdot 12)$	
	Check	$2 \times (39 \cdot 12)$	
	Stabilize	$5 \times (449 \cdot 12)$	
[10]	Escape and stabilize	$5 \times (449 \cdot 12)$	Stages defined as in Fig. 14(a) of Ref. [10].
	Inject	$9 \cdot 3 + 19 \cdot 9$	
	Stabilize	$19 \cdot 9$	
	Morph	$15 \cdot 5$	
	Check	$2 \times (24 \cdot 9)$	
	Morph and grow	$57 \cdot 14$	
	Stabilize	$51 \cdot 9$	
	Morph	$35 \cdot 6$	
	Check	$2 \times (50 \cdot 12)$	
	Morph and expand	$250 \cdot 14$	
[11]	Stabilize	$5 \times (449 \cdot 8)$	Stages defined as in Fig. 1(a) of Ref. [11].
	Inject	$9 \cdot 10$	
	Stabilize	$17 \cdot 8$	
	Deform	$21 \cdot 10$	
	Check	$2 \times (26 \cdot 12)$	
	Undeform	$26 \cdot 10$	
	Expand and stabilize	$25 \cdot 4 + 49 \cdot 8$	
	Deform	$57 \cdot 10$	
	Check	$2 \times (66 \cdot 17)$	
	Undeform	$66 \cdot 10$	
[12]	Stabilize	$5 \times (449 \cdot 8)$	Stages defined as in Ref. [12].
	Inject	$17 \cdot 8$	
	Inject and stabilize	$17 \cdot 17$	
	Grow	$17 \cdot 4$	
	Check	$2 \times (24 \cdot 9)$	
	Grow	$17 \cdot 4$	
	Grow	$25 \cdot 13$	
	Stabilize	$49 \cdot 8$	
	Grow	$49 \cdot 4$	
	Check	$2 \times (62 \cdot 11)$	
	Grow	$49 \cdot 4$	
	Grow and stabilize	$5 \times (449 \cdot 8)$	

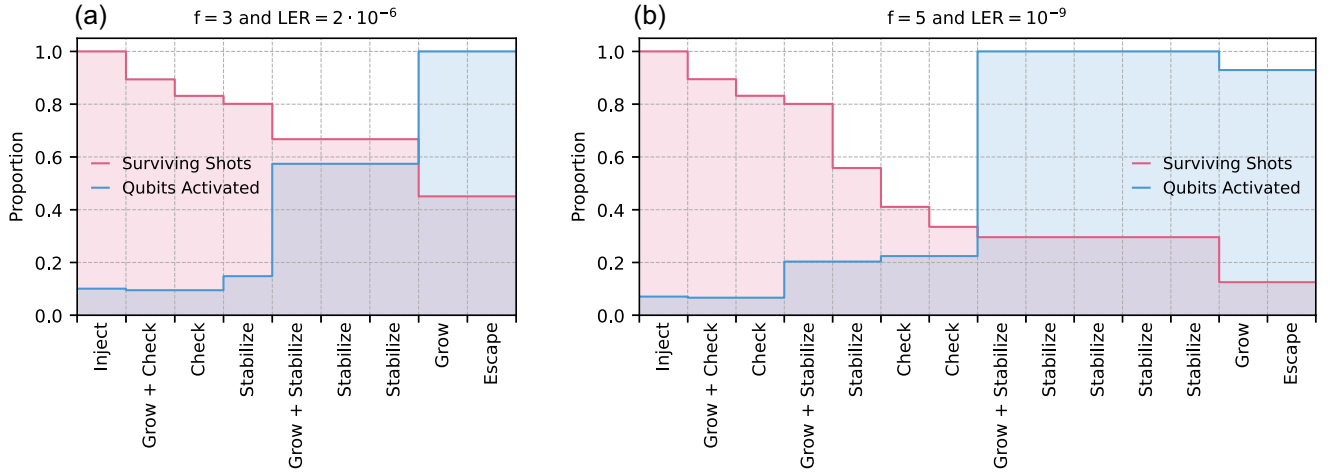


FIG. 17. Survival of shots during cultivation. Fraction of activated qubits (blue) and survival rate of shots (pink) throughout the protocol. We calculate this for a target logical error rate of (a) 2×10^{-6} with $f = 3$, and (b) 10^{-9} with $f = 5$. This includes gap-based postselection and unitary growth during the escape stage.

the logical CY check for the duration of the check. These are the same qubits absent in $\text{Rot}(3)$. This has now been corrected, leading to a slight increase in the number of expected attempts for $f = 5$ data, and a higher error floor for the $(3,0)$ scheme in Fig. 15.

3. Spacetime volume

We calculate the expected spacetime volume of protocols in the following way. For a cultivation circuit, let M denote the number of postselection stages which divide the circuit into $M + 1$ segments labeled C_i . Each segment C_i has a spacetime volume V_i , defined as the product of the number of active qubits and the number of gate steps (circuit depth). Let f_i denote the fraction of shots that proceed through the first i postselection stages, with $f_0 = 1$ by definition. At the end, a fraction f_M of shots successfully passes all postselection stages, so on average the protocol has to be repeated $1/f_M$ times to obtain a successful shot.

The expected spacetime volume per successful shot is $V = \frac{1}{f_M} \sum_{i=0}^M f_i V_i$. Each term $f_i V_i$ accounts for the expected cost of executing segment C_i , weighted by the fraction of shots that reaches it. The prefactor $1/f_M$ accounts for the fact that multiple attempts are typically required before obtaining a kept shot. Note that when there is no circuit required after the last postselection stage, such as complementary-gap-based postselection in Refs. [9–12], $V_M = 0$.

To make a fair comparison of the expected spacetime volume at a fixed final code distance in Fig. 2(b), we estimate the volume of the protocols in Refs. [9–12] at $d_{\text{fin}} = 13$ (for $f = 3$) and $d_{\text{fin}} = 15$ (for $f = 5$). All stages prior to the final escape are independent of the final distance, so only the escape stage requires adjustment. In the escape stage, the spacetime volumes of the escape circuits themselves can be straightforwardly rescaled to d_{fin} . For the final success rates, we note that in our protocol

the tradeoff between logical error rate and expected attempts, shown in Fig. 2(a), does not depend strongly on the final code distance. Guided by this observation, we adopt the reported curves from Refs. [9–12] with different final distances as estimates of the final success rates for our desired d_{fin} . For the protocols that implement escape via stabilizer measurements, we assume three rounds for $f = 3$ and five rounds for $f = 5$.

The circuit depth of each segment in different protocols depends on the specific implementation of gates and may be reduced through circuit optimization. To estimate the spacetime volumes V_i for the protocols in Refs. [9,10], we use their reported Stim circuits and count the circuit depth without any further optimization. For our protocol, we perform the same analysis using the circuits provided in Appendix A. The number of active qubits and the circuit depth of each segment for different $f = 3$ and $f = 5$ protocols are summarized in Tables V and VI, respectively.

4. Shot survival

Figure 17 shows the survival of shots for the complete cultivation process, including gap-based postselection. We do this for a target logical error rate of 2×10^{-6} ($f = 3$) and 10^{-9} ($f = 5$). The cultivation procedure is divided into multiple stages, each corresponding to one round of qubit measurements, following the convention in Ref. [9]. In our plots, we assume that the final unitary growth is applied only when the shot passes the gap-postselection. The orange curve indicates the fraction of qubits activated in each stage of the protocol, while the blue curve shows the survival rate of shots as the process progresses. The final acceptance rate is determined by the survival rate in the last stage, where the code has successfully passed all postselection checks, and has reached the target logical error rate.

- [1] S. B. Bravyi and A. Y. Kitaev, *Quantum codes on a lattice with boundary*, [arXiv:quant-ph/9811052](#).
- [2] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, *Topological quantum memory*, *J. Math. Phys.* **43**, 4452 (2002).
- [3] A. Y. Kitaev, *Fault-tolerant quantum computation by anyons*, *Ann. Phys. (N.Y.)* **303**, 2 (2003).
- [4] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, *Surface codes: Towards practical large-scale quantum computation*, *Phys. Rev. A* **86**, 032324 (2012).
- [5] A. G. Fowler, S. J. Devitt, and C. Jones, *Surface code implementation of block code state distillation*, *Sci. Rep.* **3**, 1939 (2013).
- [6] S. Bravyi and A. Kitaev, *Universal quantum computation with ideal Clifford gates and noisy ancillas*, *Phys. Rev. A* **71**, 022316 (2005).
- [7] C. Gidney, *How to factor 2048 bit RSA integers with less than a million noisy qubits*, [arXiv:2505.15917](#).
- [8] H. Zhou, C. Duckering, C. Zhao, D. Bluvstein, M. Cain, A. Kubica, S.-T. Wang, and M. D. Lukin, *Resource analysis of low-overhead transversal architectures for reconfigurable atom arrays*, in *Proceedings of the 52nd Annual International Symposium on Computer Architecture (ISCA 25)* (2025), [10.1145/3695053.373103](#).
- [9] C. Gidney, N. Shutty, and C. Jones, *Magic state cultivation: Growing T states as cheap as CNOT gates*, [arXiv:2409.17595](#).
- [10] Z.-H. Chen, M.-C. Chen, C.-Y. Lu, and J.-W. Pan, *Efficient magic state cultivation on $\mathbb{RP}^2$* , [arXiv:2503.18657](#).
- [11] J. Claes, *Cultivating T states on the surface code with only two-qubit gates*, [arXiv:2509.05232](#).
- [12] Y. Vaknin, S. Jacoby, A. Grimsmo, and A. Retzker, *Efficient magic state cultivation on the surface code*, [arXiv:2502.01743](#).
- [13] A. Kubica, B. Yoshida, and F. Pastawski, *Unfolding the color code*, *New J. Phys.* **17**, 083026 (2015).
- [14] J. E. Moussa, *Transversal Clifford gates on folded surface codes*, *Phys. Rev. A* **94**, 042316 (2016).
- [15] P.-K. Tsai and S. Puri, *A unitary encoder for surface codes*, [arXiv:2506.04084](#).
- [16] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter *et al.*, *Logical quantum processor based on reconfigurable atom arrays*, *Nature (London)* **626**, 58 (2024).
- [17] S. A. Moses, C. H. Baldwin, M. S. Allman, R. Ancona, L. Ascarrunz, C. Barnes, J. Bartolotta, B. Bjork, P. Blanchard, M. Bohn, J. G. Bohnet, N. C. Brown, N. Q. Burdick, W. C. Burton, S. L. Campbell, J. P. Campora *et al.*, *A race-track trapped-ion quantum processor*, *Phys. Rev. X* **13**, 041052 (2023).
- [18] K. Wang *et al.*, *Demonstration of low-overhead quantum error correction codes*, *Nat. Phys.* **22**, 308 (2026).
- [19] A. Cross, *A modular quantum computer based on bivariate bicycle codes*, in *Proceedings of the 7th International Conference on Quantum Error Correction* (2025), <https://yale.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=28b64187-d5f5-4eae-bbd3-b33000fc6aba>.
- [20] M. McEwen, D. Bacon, and C. Gidney, *Relaxing hardware requirements for surface code circuits using time-dynamics*, *Quantum* **7**, 1172 (2023).
- [21] Y. Li, *A magic state's fidelity can be superior to the operations that created it*, *New J. Phys.* **17**, 023037 (2015).
- [22] C. Gidney, M. Newman, P. Brooks, and C. Jones, *Yoked surface codes*, *Nat. Commun.* **16**, 4498 (2025).
- [23] H. Bombín, M. Pant, S. Roberts, and K. I. Seetharam, *Fault-tolerant postselection for low-overhead magic state preparation*, *PRX Quantum* **5**, 010302 (2024).
- [24] A. Hutter, J. R. Wootton, and D. Loss, *Efficient Markov chain Monte Carlo algorithm for the surface code*, *Phys. Rev. A* **89**, 022326 (2014).
- [25] N. Meister, C. A. Pattison, and J. Preskill, *Efficient soft-output decoders for the surface code*, [arXiv:2405.07433](#).
- [26] J. Claes, *Lower-depth local encoding circuits for the surface code*, [arXiv:2509.09779](#).
- [27] Reference [9] first escapes into a “matchable” code that aims to be as similar to Rot as possible.
- [28] C. Gidney and A. G. Fowler, *Efficient magic state factories with a catalyzed CCZ to 2 T transformation*, *Quantum* **3**, 135 (2019).
- [29] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/gpvl-lg4c> studies CCZ decomposition into 2-qubit gates.
- [30] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, *Elementary gates for quantum computation*, *Phys. Rev. A* **52**, 3457 (1995).
- [31] S. Jacoby, Y. Vaknin, A. Retzker, and A. L. Grimsmo, *Magic state injection with erasure qubits*, *PRX Quantum* **6**, 040323 (2025).
- [32] Y. Wu, S. Kolkowitz, S. Puri, and J. D. Thompson, *Erasure conversion for fault-tolerant quantum computing in alkaline earth Rydberg atom arrays*, *Nat. Commun.* **13**, 4657 (2022).
- [33] S. Ma, G. Liu, P. Peng, B. Zhang, S. Jandura, J. Claes, A. P. Burgers, G. Pupillo, S. Puri, and J. D. Thompson, *High-fidelity gates and mid-circuit erasure conversion in an atomic qubit*, *Nature (London)* **622**, 279 (2023).
- [34] P. Scholl, A. L. Shaw, R. B.-S. Tsai, R. Finkelstein, J. Choi, and M. Endres, *Erasure conversion in a high-fidelity Rydberg quantum simulator*, *Nature (London)* **622**, 273 (2023).
- [35] B. Zhang, G. Liu, G. Bornet, S. P. Horvath, P. Peng, S. Ma, S. Huang, S. Puri, and J. D. Thompson, *Leveraging erasure errors in logical qubits with metastable ^{171}Yb atoms*, [arXiv:2506.13724](#).
- [36] O. Higgott, M. Wilson, J. Hefford, J. Dborin, F. Hanif, S. Burton, and D. E. Browne, *Optimal local unitary encoding circuits for the surface code*, *Quantum* **5**, 517 (2021).
- [37] N. P. Breuckmann and S. Burton, *Fold-transversal Clifford gates for quantum codes*, *Quantum* **8**, 1372 (2024).
- [38] M. H. Shaw and B. M. Terhal, *Lowering connectivity requirements for bivariate bicycle codes using morphing circuits*, *Phys. Rev. Lett.* **134**, 090602 (2025).
- [39] R. Wille, L. Berent, T. Forster, J. Kunasaikaran, K. Mato, T. Peham, N. Quetschlich, D. Rovara, A. Sander, L. Schmid, D. Schönberger, Y. Stade, and L. Burgholzer, *The MQT handbook: A summary of design automation tools and software for quantum computing*, in *Proceedings of the 2024 IEEE International Conference on Quantum Software (QSW)* (IEEE, New York, 2024), pp. 1–8, [10.1109/QSW62656.2024.00013](#).

- [40] C. Gidney, *Cleaner magic states with hook injection*, [arXiv:2302.12292](#).
- [41] K. Sahay, P.-K. Tsai, and K. Chang, *Fold transversal surface code cultivation* (2025), https://github.com/kaavyas99/MSC_foldedH [accessed September 04, 2025].
- [42] S. Singh, A. S. Darmawan, B. J. Brown, and S. Puri, *High-fidelity magic-state preparation with a biased-noise architecture*, *Phys. Rev. A* **105**, 052410 (2022).
- [43] L. Lao and B. Criger, *Magic state injection on the rotated surface code*, in *Proceedings of the 19th ACM International Conference on Computing Frontiers* (2022), [10.1145/3528416.3530237](#).
- [44] C. Chamberland and K. Noh, *Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits*, *npj Quantum Inf.* **6**, 91 (2020).
- [45] F. Lima and A. C. Medina, *Clifford and Non-clifford splitting in quantum circuits: applications and ZX-calculus detection procedure*, [arXiv:2504.16004](#).
- [46] Any logical Pauli state is sufficient here. The choice is irrelevant because further measurements and corrections are independent of the encoded logical state.
- [47] L. Daguerre and I. H. Kim, *Code switching revisited: Low-overhead magic state preparation using color codes*, *Phys. Rev. Res.* **7**, 023080 (2025).
- [48] Fig. 13 of Ref. [9].
- [49] O. Higgott and C. Gidney, *Sparse blossom: Correcting a million errors per core second with minimum-weight matching*, *Quantum* **9**, 1600 (2025).
- [50] C. Gidney, *Sinter: Fast QEC sampling* (2025), <https://pypi.org/project/sinter/> [accessed September 09, 2025].