Error Correction of Transversal CNOT Gates for Scalable Surface-Code Computation

Kaavya Sahay,[1,2,*] Yingjia Lin,[3,4] Shilin Huang,[1,2] Kenneth R. Brown,[3,4,5,6] and Shruti Puri[1,2]

[1] *Yale Quantum Institute, Yale University, New Haven, Connecticut 06511, USA*
[2] *Department of Applied Physics, Yale University, New Haven, Connecticut 06520, USA*
[3] *Duke Quantum Center, Duke University, Durham, North Carolina 27701, USA*
[4] *Department of Physics, Duke University, Durham, North Carolina 27708, USA*
[5] *Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina 27708, USA*
[6] *Department of Chemistry, Duke University, Durham, North Carolina 27708, USA*

(Received 14 September 2024; revised 13 January 2025; accepted 4 April 2025; published 7 May 2025)

Recent experimental advances have made it possible to implement logical multiqubit transversal gates on surface codes in a multitude of platforms. A transversal controlled-NOT (tCNOT) gate on two surface codes introduces correlated errors across the code blocks and thus requires modified decoding compared to established methods of decoding surface-code quantum memory (SCQM) or lattice-surgery operations. In this work, we examine and benchmark the performance of three different decoding strategies for the tCNOT for scalable fault-tolerant quantum computation. In particular, we present a low-complexity decoder based on minimum-weight perfect matching (MWPM) that achieves the same threshold as the SCQM MWPM decoder. We extend our analysis with a study of tailored decoding of a transversal-teleportation circuit, along with a comparison between the performance of lattice-surgery and transversal operations under Pauli- and erasure-noise models. Our investigation builds toward systematic estimation of the cost of implementing large-scale quantum algorithms based on transversal gates in the surface code.

DOI: 10.1103/PRXQuantum.6.020326

I. INTRODUCTION

Quantum error correction (QEC) protects encoded logical quantum information from decoherence on the underlying physical qubits [1,2]. Recent experimental progress has led to landmark demonstrations of fault-tolerant (FT) state preparation [3–6], repeated error correction [7–10], and state teleportation [11] of encoded logical states. In a multitude of platforms, high-fidelity two-qubit operations are no longer strictly confined to two-dimensional (2D) nearest-neighbor interactions [12,13], presenting new implementation possibilities. In particular, nontrivial connectivity can be employed for logical operations in the widely studied surface code (SC), a leading candidate for practical QEC [14,15].

In fixed-qubit architectures, the prominence of the surface code can be attributed to its 2D planar layout, nearest-neighbor connectivity, low-depth syndrome-measurement circuits, and high error-tolerance threshold [16–18]. Efficient graph-based decoders, such as minimum-weight perfect matching (MWPM), perform well at correcting common circuit-level errors [19,20]. Further, logical gates on surface codes are well understood and easy to implement via braiding [16,21,22] or lattice surgery [23,24].

With nonlocal connectivity, it is possible to implement transversal logical gates, such as the transversal logical CNOT (tCNOT), between any pair of surface codes. As shown in Fig. 1, this requires applying physical CNOT gates between every corresponding data qubit of the control and target SC states [23]. A tCNOT creates correlated errors. For example, a bit-flip error on a data qubit in the control may be copied over to the corresponding data qubit in the target. One method to account for these error correlations is by appropriately adding syndrome history from rounds following the CNOT gate from one SC to another [25,26]. Decoding using the resultant syndromes is suboptimal since the combined syndromes are twice as noisy as their individual components. An alternative decoding strategy is to directly use all the measured syndromes of the two SCs without addition. In this case, decoding based on graph algorithms cannot be used and previous works thus resort to relatively slower hypergraph decoding [27,28].

*Contact author: kaavya.sahay@yale.edu

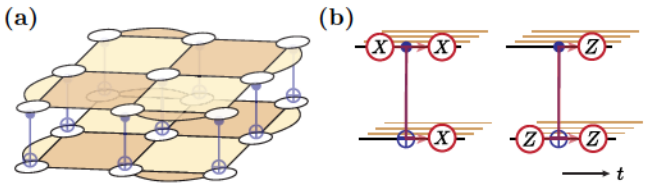

FIG. 1. (a) A logical transversal CNOT operation between two rotated surface codes is performed by applying physical CNOT gates between each corresponding pair of data qubits of the SC states. (b) The transversal CNOT creates correlated errors between surface codes. Each SC is shown as a qubit set on which an error on a physical qubit can propagate to the other SC through the tCNOT.

In this work, we benchmark the performance of the transversal CNOT gate for scalable quantum computing using three decoding methods. In addition to the approaches described above, we study a thus-far uncharacterized strategy that we refer to as *ordered decoding*. In this approach, we first decode those errors in one SC state that may be copied over to the other SC state [29]. We then correct for the identified errors on the second SC state before independently decoding the residual errors. Any decoder that independently locates bit- and phase-flip errors can be employed in this overall strategy; here, we use MWPM. We find that ordered decoding using MWPM is highly effective at correcting noise correlations introduced by the transversal CNOT on surface codes, outperforming previous tCNOT decoders in terms of thresholds. Our analysis extends beyond previously studied constant-depth circuits [11,27,28,30,31].

We additionally study decoding for transversal teleportation circuits in which one of the code blocks is measured soon after a tCNOT gate. Such teleportation circuits comprise a high fraction of two-qubit-gate usage in quantum algorithms. We find that such teleportation operations can be inherently decoded using graph-based methods. We also provide a comparison for logical operations performed transversally versus via joint-parity (JP) measurements, i.e., lattice surgery, thus far the more widely studied method for surface-code logical-gate operations. This analysis is presented for Pauli noise and a mix of erasure and Pauli noise. The latter noise model is motivated by recent studies showing that qubits with dominant erasure noise exhibit high thresholds and improved error-correction properties [32–37].

Our work is structured as follows. In Sec. II, we provide a brief introduction to the surface code when used as a quantum memory (SCQM), followed by an analysis of methods to decode and correct errors in this system. We then move on to logical computation using surface codes. We set out definitions for tCNOT circuits and argue for individually fault-tolerant tCNOT gadgets in logical algorithms in Sec. III. In Sec. IV, we discuss different tCNOT decoding strategies. In Sec. V, we provide an analysis of gate teleportation, with a focus on decoding optimizations and a brief comparison with lattice surgery. Finally, in Sec. VI, we discuss the performance of transversal and lattice-surgery-based logical gates for erasure-based noise models. We conclude in Sec. VII.

II. THE SURFACE CODE AS AN ERROR-CORRECTED MEMORY

The rotated surface code [38,39] is a stabilizer error-correcting code [40] that uses d^2 physical qubits arranged on the vertices of a $d \times d$ square lattice to encode one logical qubit. The length of the smallest logical operator or, equivalently, the minimum number of Pauli errors that cause an undetectable change in the logical state, is d. Here, we take d to be odd. The stabilizer group $\mathcal{S}$ of this code is generated by X and Z type *checks* S_X (S_Z) on alternating faces of this lattice, as illustrated in Fig. 2(a). Each X (Z) check is a product of Pauli X (Z) operators on the qubits around the face. The X- and Z-type logical operators, $\overline{X}$ and $\overline{Z}$, consist of Pauli X and Z operators on qubits lying on strings connecting the boundaries of the lattice such that $\{\overline{X}, \overline{Z}\} = 0$, $[\overline{X}, S] = 0$ and $[\overline{Z}, S] = 0$ for all checks $S \in \mathcal{S}$. A $d = 5$ surface code with a logical operator $\overline{X}$ [marked (i)] is shown in Fig. 2(a).

In practice, each X (Z) check of the rotated surface code is measured using a depth-4 circuit of CNOT (CZ) gates to entangle the relevant physical data qubits with

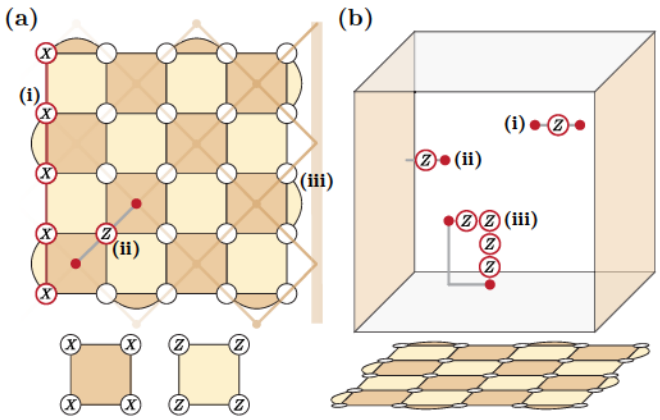

FIG. 2. (a) A $d = 5$ rotated surface code: (i) an X-logical operator, (ii) a single-qubit Z error, with the corresponding anticommuting stabilizer measurements highlighted, and (iii) the X-decoding graph G_X used to correct for Z errors for one stabilizer-measurement round. (b) A representation of G_X generated by using d rounds of stabilizer measurements on the underlying surface code: (i) errors in the bulk create two defects to be matched together, (ii) an error at the boundary creates a single defect, and (iii) a string of data-qubit and measurement errors (red) and its corresponding matching-obtained correction (gray)—the correction restores the original logical state up to code stabilizers.

an additional ancilla qubit that is then measured out [17], giving rise to a set of measurement outcomes referred to as the *syndrome*. Errors can occur within this circuit at any point. Detectable errors anticommute with a subset of checks and flip the corresponding measurement outcomes, creating *defects*. For example, in Fig. 2(a)(ii) we show a Z error on a data qubit that creates two adjacent defects. In the presence of faulty measurements, each stabilizer is generally measured $O(d)$ times [25]. This error-correction protocol implements the identity channel on the encoded qubit—as a result, an isolated surface code acts as a quantum memory.

A. Efficiently decoding errors on the SCQM

Given an error syndrome σ, optimal decoding involves finding a correction that maximizes the probability of restoration to the original code state. For general systems, this can be a computationally hard problem [41]. In this section, we give an overview of simpler polynomial-time decoders used for surface-code error correction.

We begin by defining a decoding hypergraph $G = (V, E)$ on a surface code with errors $\mathcal{E}$. Each vertex $v_S^t \in V$ corresponds to a *detector*, where a detector refers to the parity between the measurement outcomes of a check at time $t-1$ and t. In other words, $v_S^t = S^{t-1} \oplus S^t$ for $S \in \mathcal{S}$ [42]. The *defects* $\mathcal{D}$ generated by $\mathcal{E}$ are detectors with odd parity under $\mathcal{E}$. Every hyperedge $e \in E$ is a set of detectors, and is assigned a weight proportional to the logarithm of the total probability of an independent error that causes that set of detectors to take the value 1. Given $\{G, \mathcal{D}\}$, the task of a graph-based decoder is to find the most probable physical error that created $\mathcal{D}$.

For general hypergraphs where $|e| \in \mathbb{N}$, this is a computationally hard problem [43]. By making the simplification of finding a locally—as opposed to globally—optimal solution, it is possible to efficiently find a correction operator $\mathcal{C}$ of which the syndrome matches $\mathcal{D}$; one such strategy is the hypergraph union-find (HUF) decoder [44] .

Another simplification is to reduce G to a graph where $|e| \leq 2$, $\forall e$. For the surface code, this is possible for all single-qubit X (Z) errors, since these create independent pairs of defects (or a single defect at the boundaries) on the Z- (X-) type stabilizer set, seen in Fig. 2(a)(ii). As a result, the decoding hypergraph G can be split into two disjoint graphs, $\{G_X, \mathcal{D}_X\}$ and $\{G_Z, \mathcal{D}_Z\}$, that satisfy $|e| \leq 2$ $\forall e \in E$. An example of G_X for one measurement round is shown in Fig. 2(a)(iii) and the box in Fig. 2(b) represents an example of G_X for d measurement rounds that we refer to as the spacetime-decoding volume. Y noise is decomposed into G_X and G_Z as uncorrelated X and Z errors. A decoder can now identify the most probable physical error in polynomial time by mapping σ to minimum-weight matching problems on G_X and G_Z.

For a given correction $\mathcal{C}$ found by a decoder, a corresponding update is applied to the surface code, ideally restoring it to the original state [as in Fig. 2(b)(iii)] but potentially causing a logical error if the correction proposed is logically inequivalent to the original error. For an MWPM-based decoder applied to a circuit-level noise model with two-qubit-gate errors, the threshold error rate p_t, below which the logical error rate decreases with increasing system size, is 1% [45,46]. We find the corresponding HUF threshold to be 0.89%.

III. A SCALABLE TRANSVERSAL CNOT

In a transversal multiqubit logical operation, a physical qubit of one logical block interacts with at most one physical qubit of another logical block. This approach naturally preserves the effective code distance. Here, we focus on a logical transversal CNOT (tCNOT) between a control surface-code block C and a target block T, implemented with physical CNOT gates applied between qubit q_C in C and q_T in T, for every physical qubit q.

A tCNOT can introduce correlated errors between C and T via two mechanisms. First, two-qubit errors can occur after each physical CNOT gate. Further, errors prior to the tCNOT can propagate from one code block to another. Specifically, $Z(X)$ errors on qubit q_T (q_C) existing prior to the tCNOT are copied onto q_C (q_T). Importantly, the number of these errors copied over by the tCNOT scales linearly with the number of operations between the last applied correction and the tCNOT. For example, if r stabilizer-measurement rounds precede the tCNOT before a correction is applied, then the number of copied errors on each physical qubit grows as $O(r)$. A successful decoder must be able to decode such correlated errors across the logical code blocks.

With tCNOTs, in principle it is possible to decode over an entire algorithm using $O(1)$ rounds of syndrome extraction per tCNOT with well-prepared logical ancilla states [28,47–49]. However, here we consider decoding for tCNOT gates at scale. In particular, we focus on quantum algorithms with the number of gates scaling exponentially with d, that require a distance d fault-tolerant gate set for successful implementation [50,51].

For such a circuit, we define a tCNOT *gadget* to comprise of $g = O(1)$ tCNOTs in a known configuration. We take each gadget to be followed by r rounds of stabilizer measurements on involved code blocks. For simplicity, we consider $g = 1$, i.e., a gadget consists of a single tCNOT, and we ignore single-qubit gates. In the following, we discuss the requirements for r in the context of scalable quantum computing. Note that in this deep circuit, information from final transversal logical measurements is not readily accessible for use in decoding [28].

We exemplify our arguments using a "binary-tree" circuit of logical gadget depth M involving 2^M qubits. An example circuit where $M = 3$ is shown in Fig. 3(a). In this circuit, a single bit-flip error X_q on a qubit q in the control

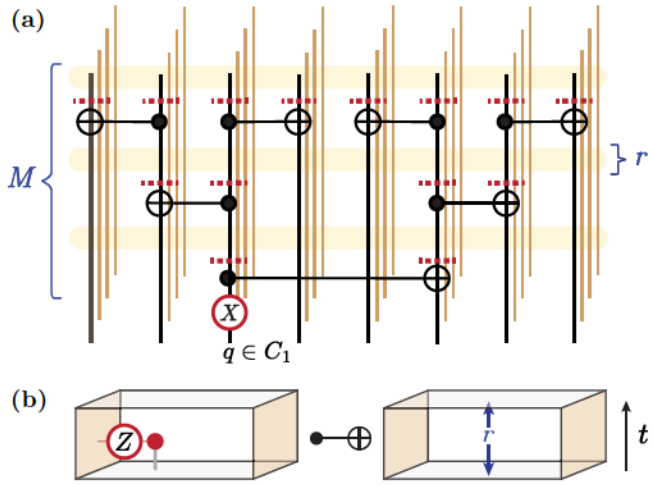

FIG. 3. (a) A "binary-tree" tCNOT circuit of logical depth M. Each tCNOT is followed by r rounds of stabilizer measurements (shaded yellow boxes). In this case a single error (e.g., an X error on qubit q) can induce correlated errors that grow exponentially with circuit depth (marked by dashed red lines). (b) If $r \ll O(d)$, each tCNOT cannot be decoded independently, since weight-$O(r)$ data errors can be misidentified as measurement errors, creating logical failures.

SC of the first tCNOT, which we term C_1, can propagate to the corresponding qubits of all SCs. In a similar fashion, Z errors on target SCs will flow to C_1. Correction of this first tCNOT should address both X and Z errors.

The surface code does not satisfy single-shot code properties for local check measurements [52–54], and decoding errors at a specific location requires roughly $O(d)$ subsequent rounds of stabilizer measurements [25,53,55]. We denote $O(d)$ by W in the following. Thus, if we set $r = W$, existing errors at each tCNOT can naturally be corrected before the application of subsequent gates. When $r \ll W$, however, one cannot decode each tCNOT independently. To see this, consider Fig. 3(b), with an open Z-error string $\mathcal{E}$ on C right before the tCNOT connecting a spatial boundary to a single defect. Given a decoding-window depth of r, a decoder can misinterpret $\mathcal{E}$ as a single length-r measurement error string if $|\mathcal{E}| > r$.

When $r < W$, instead of attempting to decode individual tCNOTs, it may be possible to use a correlated-decoding approach. Here, we use an expanded depth-m ($m \leq M$) window of the circuit extending over 2^m qubits. Syndromes in the entire window are used to decode errors at the beginning of the window [27,28]. This procedure is an extension of the overlapping- or sliding-window approach used for preserving a quantum state but applied to a quantum circuit block [56–59]. Note that in a general circuit, some qubits may be idle in the depth-m window and these can be decoded separately as conventional quantum memories. In the rest of our discussion we will neglect these idle qubits.

TABLE I. Constraints limiting tCNOT gadget separation r and decoding-window depth m while decoding an exponentially growing tCNOT circuit. The decoding volume is reported in units of d^2 and we assume that $O(1) \ll d$.

r	m	tCNOTs decoded independently	C_1 is FT	Decoding volume
$O(1)$	$O(1)$	N	N	$O(1)$
$O(1)$	$O(d)$	N	Y	$O(e^d)$
$O(d)$	$O(1)$	Y	Y	$O(d)$

Let us examine correction of X and Z errors on C_1 in this window. Since C_1 is always the control of the tCNOTs, any error X_q is copied from C_1 onto 2^m qubits, each of which may provide syndrome information about this propagated error. Conversely, Z errors propagate *onto* C_1 from m logical gates. Decoding of Z errors on C_1 at the first tCNOT is dependent on decoding of the 2^m SCs from which these errors can originate. The total decoding volume for C_1, which determines the complexity of decoding in the correlated-decoding approach, thus scales exponentially with m. If the decoding volume becomes too large, the backlog problem may be amplified [20,60]. Thus, we need to determine how this volume scales based on an (r, m) choice that ensures that long-lived errors are prevented.

To this end, first consider $r, m \ll W$ for which, following previous arguments, we find that a decoder can fail to correct a Z-error string $\mathcal{E}$ of length $O(mr)$ on C_1 at the beginning of the window i.e., the first tCNOT. Thus, $r, m \ll W$ is not sufficient to prevent a long-lived Z error. On the other hand, setting $mr = W$ is sufficient to prevent Z errors at the beginning of the window from surviving for a long time. As expected, this condition cannot be satisfied if both $m, r = O(1)$. It can instead be satisfied by setting $r = O(1)$ and $m = O(W)$. In this case, the decoding volume increases exponentially with m, leading to an undesirable slow-down in decoding. Clearly, the simplest way to achieve $mr = W$ without increasing the decoding volume exponentially is by choosing $r = O(W)$ and $m = O(1)$. These conditions are summarized in Table I.

Hence we find that even in the expanded-window approach, $r = O(W)$ is desirable. Moreover, when $r = W$ and $m = 1$, each tCNOT gadget is effectively decoded independently. The advantage of these modular gadgets is that one can benchmark their individual failure rates and then estimate algorithmic performance for a wide variety of algorithms. For increasingly larger gadgets, the internal structure of the circuit determines how errors spread between different code blocks, complicating later circuit analysis. To predict algorithmic performance, a defined set of gadgets would need to be benchmarked and the errors carefully composed.

This argument indicates that, in the general case in which errors may spread exponentially within a logical circuit, with decoders acting on volumes independent of circuit size, decoding over gadgets separated by $O(1)$ rounds will result in inferior performance compared to separation by $O(d)$ rounds. This is not the case when the whole circuit can be captured in an achievable decoding volume. Having addressed the effectiveness of $r = W$, we now discuss three decoding procedures that can be used to decode an individual tCNOT. Specifically, we use the simplifying assumption of $W = d$ since measurement errors in our numerically simulated noise model are roughly as likely as data-qubit errors.

IV. DECODERS FOR A TRANSVERSAL CNOT

Here, we analyze decoding of a tCNOT. We consider a decoding volume comprising of the d rounds of stabilizer measurements following the tCNOT along with the d rounds of stabilizer measurements that followed the preceding gate. We focus on correcting Z-type errors using X checks, using the notation S_{CX} (S_{TX}) to refer to X checks on the control (target); by symmetry, correction of X-type errors using Z checks follows the same principles.

For our decoding analysis, we introduce the unifying perspective of check *frames* for SCs in the tCNOT. In what we term the *dynamic frame*, each X check used for correcting Z errors evolves through the tCNOT as

$$\mathbf{S}^i_{\mathrm{CX}} = S^i_{\mathrm{CX}} S^i_{\mathrm{TX}} \quad \forall i \in \{d+1, d+2, \ldots, 2d\}. \tag{1}$$

This reflects the internal evolution of the checks caused by the tCNOT. $\mathbf{S}^i_{\mathrm{CX}} = S^i_{\mathrm{CX}}, \forall i \leq d$ and $\mathbf{S}_{\mathrm{TX}} = S_{\mathrm{TX}}$, i.e., other X checks remain unchanged in the dynamic frame. In contrast, we define a *static frame*, where all checks are left in their original values.

For any given frame, we can define the X-decoding graph of the system G as laid out in Sec. II. For convenience in the chosen frame, we break up G_X into *subgraphs* G_{CX} and G_{TX}, where G_{CX} (G_{TX}) contains the frame-defined nodes related to the checks of C (T). We term G_{CX} a *dependent subgraph* since its checks change due to the tCNOT in the dynamic frame. From a complementary perspective, the control can be regarded as dependent because Z errors are copied over to C by the tCNOT. Unlike G_{CX}, the nodes of G_{TX} are unaffected by the tCNOT; it is thus termed an *independent subgraph*.

Note that in experiment, we always only measure checks in the static frame. Checks of the dynamic frame can be inferred by combining the outcomes of the static X checks at the cost of making the inferred dynamic checks more unreliable than the measured static checks in the presence of measurement errors.

A. Single-update decoder

In the single-update decoding strategy, we operate in the dynamic frame. Post-tCNOT, since we update the S_{CX} check-measurement outcomes according to Eq. (1), the detectors $v^i_{\mathrm{CX}} = \mathbf{S}^{i-1}_{\mathrm{CX}} \oplus \mathbf{S}^i_{\mathrm{CX}}$ correctly track checks through the tCNOT. We can now apply MWPM to decode the defects created in the new dynamic frame detector set, similar to that of an SCQM (see Algorithm 1). Note that the use of the dynamic frame doubles the defect rate on G_{CX} post-tCNOT. We provide examples of handling of relevant errors by the single-update decoder in Appendix A.

To benchmark the performance of the single-update decoder, we compare the numerically calculated logical error rate for the tCNOT decoded in this fashion using MWPM, to that of two independent disjoint SCQMs for $2d$ rounds of check measurements (hereon referred to as a 2SCQM experiment). The results are shown in Figs. 4(b)

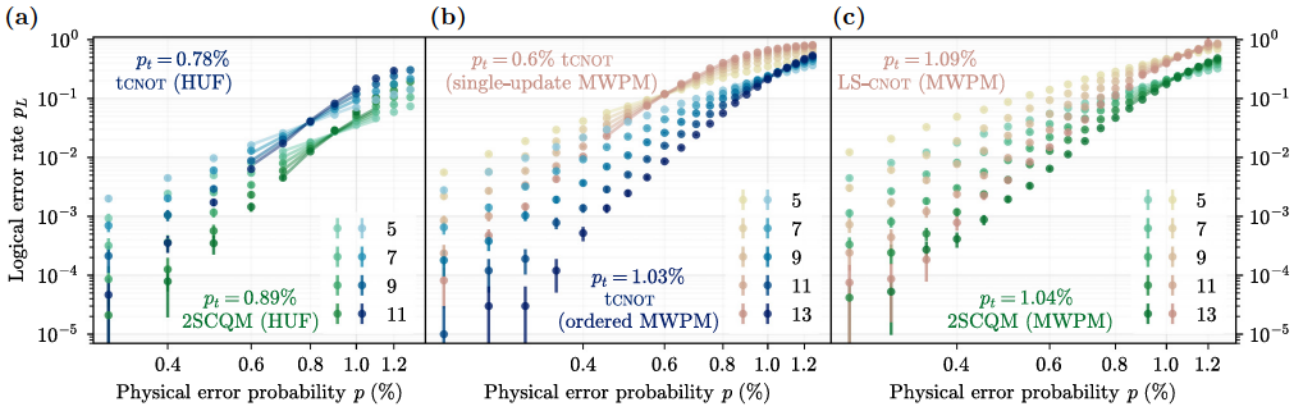

FIG. 4. Thresholds and logical failure rates under circuit-level depolarizing Pauli gate errors for SCQMs and logical CNOTs. (a) The HUF-decoder performance for a 2SCQM and tCNOT. (b) The single-update- and ordered-MWPM-decoder performance for a tCNOT. (c) The MWPM-decoder performance for a 2SCQM and lattice-surgery (LS) CNOT. The logical failure rates for different operation-decoder combinations are displayed across three panels to prevent overcrowding (for another version, see Fig. 13). Threshold values p_t are found by finite-size scaling (translucent lines).

and 4(c). In this and all simulations with Pauli noise, two-qubit-gate errors are uniformly chosen at random from $\{I,X,Y,Z\}^{\otimes 2}/\{I\otimes I\}$ at a rate p. Note that to build intuition for error handling across decoders, we assume no single-qubit gate, initialization, or measurement errors here in the main text (for further discussion and a noise model more similar to Refs. [27,31,61], see Appendix I).

For the above two-qubit-gate error model, as expected, due to the larger defect population on G_{CX}, this procedure results in a reduction in the tCNOT threshold ($p_t = 0.6\%$) compared to a 2SCQM ($p_t = 1.04\%$). We note that the single-update decoding strategy has been proposed in Ref. [25] and previously used in Ref. [26].

B. Combined hypergraph decoder

We now move onto our second decoding strategy. The dynamic frame, while natively following the tCNOT induced check evolution, increases the number of errors on the dependent subgraph. It is thus natural to attempt to use the static frame where possible. As explained in Appendix B, it is possible to *only* use the dynamic frame at the tCNOT round, with all other rounds in the static frame to define a set of detectors. We call this the *hybrid frame*. In this frame, the tCNOT creates nondecomposable *hyperedges* in the decoding graph, i.e., a single independent error mechanism creates three or more defects [27,28,62]. A naive matching decoder cannot successfully decode hyperedges; we thus use a hypergraph decoder, specifically the hypergraph union-find (HUF) decoder [44,63,64].

We compare the performance of the tCNOT decoded with HUF to that of 2SCQM in Fig. 4(a). Interestingly, we see a slight reduction in threshold for the tCNOT ($p_t = 0.78\%$) compared to the 2SCQM experiment ($p_t = 0.89\%$). This may be attributed to the locally—as opposed to globally—optimal corrections found by the HUF algorithm in combination with the increased complexity of hyperedges in the decoding graph for the tCNOT versus the 2SCQM.

Hypergraph decoding presents a potential path to decode and correct logical operations that induce hyperedges. However, as seen above, time-efficient hypergraph strategies underperform with increased hypergraph complexity and have higher run-time overheads than their graph-based counterparts. Ideally, we would like an efficient decoding algorithm for the tCNOT that scales equal to or better than the equivalent decoder applied to a 2SCQM, while at the same time preserving the SCQM threshold. Our next decoding strategy, ordered decoding, achieves this goal.

C. Ordered decoder

Let us now describe an ordered-decoding strategy. This operates entirely in the static frame. We know that Z errors occurring on the target pre-tCNOT—and *only* these Z errors—are propagated to the control. If we can, to the best of our ability, fully identify such errors via decoding the target first, we then know exactly what defects they will create on the control at round $d+1$.

Ordered decoding relies on exactly this principle: we first decode T using G_{TX} in the static frame. From the resulting solution, the decoder identifies error clusters on T that occur before the tCNOT. Detectors in G_{CX} in the static frame that correspond to the propagation of these identified errors from target to control at round $d+1$ are then flipped, keeping track of the updated logical status. This process changes the collection of apparent defects on the control. We finally simply decode and correct the control with the updated defect collection. A corresponding mechanism can also be applied to the Z-decoding graph with the control decoded before the target (see Algorithm 2 for a full description).

An ordered-decoding strategy on a tCNOT results in the preservation of the SCQM threshold [see Fig. 4(b) with $p_t = 1.03\%$], with only a marginal bounded increase in logical error rates compared to an equivalent decoder on a 2SCQM (for further analysis, see Appendix D). Note that ordered decoding doubles the decoding time, as decoding C for Z errors can only begin after the target is decoded (and vice versa for X errors). This constant-factor increase does not amplify the backlog problem.

V. LOGICAL-STATE TELEPORTATION

One of the primary uses of two-qubit gates in quantum algorithms will be for logical-state teleportation, particularly for non-Clifford gates, such as in Fig. 5. Here, we investigate these fault-tolerant teleportation circuits, focusing on the teleportation step that may be implemented transversally or by JP measurements. In a transversal-teleportation circuit using a tCNOT, where a logical measurement is to be immediately implemented following the tCNOT, we find that, unlike the general unitary case, subsequent stabilizer-measurement rounds between the tCNOT and the logical measurement on the target are not necessary to maintain code performance. We additionally compare a transversal-teleportation strategy to that using joint measurements, an approach generally considered more suitable for planar architectures.

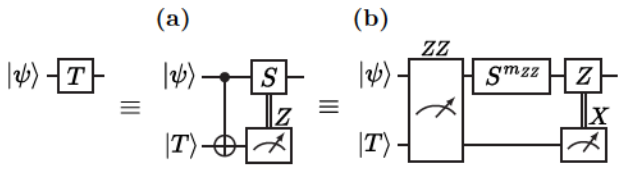

FIG. 5. Methods to perform a T gate using fault-tolerant magic state teleportation. (a) A logical state is teleported onto the original qubit using a CNOT, followed by a Clifford update. (b) In a lattice-surgery setting, the equivalent protocol in (a) can be optimized to reduce the number of joint-parity (JP) measurements [65].

A. Hypergraph reduction for transversal measurements

We first study the transversal-teleportation scheme in Fig. 5(a), where a logical measurement of one of the surface codes immediately succeeds a tCNOT. We consider d rounds of stabilizer measurements on both the control and the target before the tCNOT. Post-tCNOT, since the logical Z measurement m_{TZ_L} on T determines whether an S gate is applied to the control, it is desirable that the logical Z measurement is correctly read out. We find that the logical measurement terminates the decoding graph and reduces the hybrid-frame hyperedges across the tCNOT to edges. Therefore, MWPM decoders are naturally applicable in this scenario.

We first focus on correction of the logical Z measurement on the target. Note that m_{TZ_L} is itself FT and detects X errors on T [47]. From m_{TZ_L}, we extract an extra $(d+1)$th round of Z-stabilizer measurements. We use detectors of the hybrid frame (Appendix B). Crucially, unlike the tCNOT circuit that we study in Sec. IV, we do not include d rounds of stabilizer-measurement results on the control following the tCNOT during this correction. Therefore, there are no additional detectors v_{CZ}^{i} used for $i > d$ after the tCNOT. A measurement error on S_{CZ}^{d} thus flips only two detectors v_{CZ}^{d} and v_{CZ}^{d+1} instead of three, as described in Appendix B. Since each check is now involved in at most two detectors, this detector subset reduces hyperedges to edges. For correction of the logical Z measurement on T with an MWPM decoder, we obtain $p_t = 1.12\%$, as shown in Fig. 6.

We now move on to X error correction on C. During correction of the logical Z measurement on T, we also obtain a correction for pre-tCNOT control X errors. We first apply this correction to C in an ordered-decoding manner before continuing to measure d rounds of stabilizers, which can subsequently be treated as an SCQM. The logical error rate in this case, which overlaps with that of the target logical Z measurement, is shown in Fig. 6.

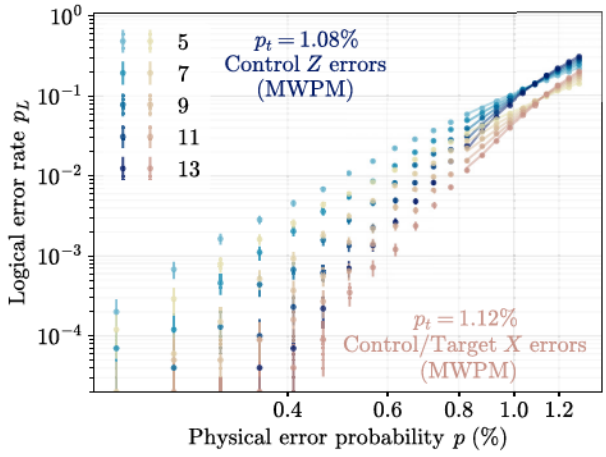

FIG. 6. Thresholds and logical error rates for a gate teleportation circuit corrected using ordered decoding under circuit-level Pauli noise. We show the performance of target X error (logical Z measurement) and control X error correction in one plot (lower points in brown) as they agree within error bars. Correction of Z errors on the control is represented by the higher points in blue. Thresholds are found by finite-size scaling close to p_t (translucent lines).

For Z error correction on C using X checks, we similarly find no hyperedges. Thus, graph-based decoding and its inherent logical error rates and thresholds are maintained when specific detectors are not present, as is in the case of transversal logical measurements immediately after a tCNOT.

B. A comparison with lattice surgery

In previous sections, we have laid out our scalable tCNOT strategy. In this section, we turn to a comparison with lattice surgery [23,24,65–67]. In lattice surgery, static logical surface-code patches are set up with bridging regions of unentangled qubits between them. Stabilizer measurements on these bridging regions are turned on and off to connect the logical operators of individual surface codes and perform joint logical Pauli measurements. Reliable measurement of these joint Pauli operators requires d rounds of stabilizer measurements. Arbitrary logical Clifford gates can be executed via combinations of these joint measurements and Pauli gates.

The decoding of these operations has been well studied, being nearly identical to decoding a SCQM. As in the case of a SCQM, a conventional MWPM decoder works well and thus we use this decoder for comparison (for details, see Appendixes E–G). In practice, a lattice-surgery-based quantum algorithm is compiled into the shortest sequence of joint measurements instead of directly using a gate set composed of CNOT gates [65,68] and a transversal implementation may use logical blocks with $m, g > 1$. In this context, we leave the somewhat artificial comparison of an isolated fault-tolerant tCNOT versus a lattice-surgery CNOT to Appendix H, showing therein that the tCNOT has both lower overheads and lower logical error rates. We display the numerical results for logical error rates in Fig. 4.

In Fig. 5(b), we show the JP version of the teleportation circuit in Fig. 5(a). Since only a ZZ logical measurement between the two surface codes is required, this operation takes d rounds and requires no additional logical ancilla patches, which matches the overhead of an isolated fault-tolerant tCNOT within a logical algorithm. This suggests that in certain settings, the two strategies may use equivalent resources. We leave an extended overhead analysis to future work, hereon focusing on fault-tolerant thresholds for both approaches.

VI. LOGICAL GATES FOR ERASURE QUBITS

Until now, we have analyzed logical gates under independent identically distributed Pauli noise. We now move

onto a corresponding analysis for qubits where the dominant errors include a form of structured noise called *erasures* [32,34]. Erasures consist of detectable errors that result in the affected qubit being projected into the maximally mixed state. We further investigate a more tailored model consisting of *biased erasures* [37], where detectable errors only happen from one half of the computational subspace. We review the biased-erasure-noise model in Appendix J.

It is advantageous to engineer qubits the dominant noise of which is erasures [34,69]. In practice, not all noise can be converted to erasures; here, we assume the remainder to be depolarizing Pauli noise. This motivates us to define an erasure fraction R_e, i.e., given errors occurring on gates at a rate p, pR_e of them are converted to erasures and the rest, $p(1-R_e)$, are Pauli errors. As different qubits operate at different erasure fractions, is instructive to see how the error-correction properties of a code vary with changes in R_e.

We analyze the change in MWPM-based thresholds with R_e for logical operations in Fig. 7. Results for conventional erasures are shown in Fig. 7(a) and for biased erasures in Fig. 7(b). We now briefly summarize the results for biased erasures: the threshold for a tCNOT corrected using ordered decoding increases from 1.03% at $R_e = 0$ to 8.3% at $R_e \approx 1$. The thresholds for joint measurements (denoted by LS-CNOTs in Fig. 7) exactly coincide with a SCQM up to error bars, increasing from 1.04% at $R_e = 0$ to 10.3% at $R_e \approx 1$. These values *also* match with those of ordered decoding on a tCNOT except at extremely high erasure fractions $R_e > 0.95$. Concretely, at $R_e = 0.98$, we see

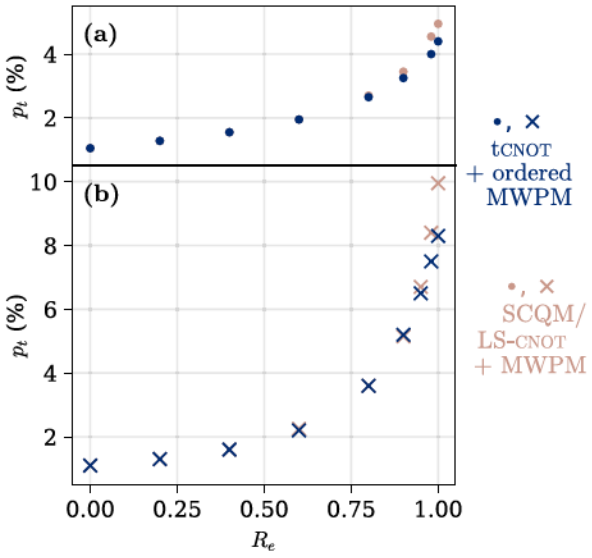

FIG. 7. MWPM thresholds under circuit-level noise while varying the erasure fraction R_e for a lattice-surgery CNOT (browns) and a tCNOT with ordered decoding (blues): (a) conventional erasures; (b) biased erasures.

that $p_t = 7.5\%$ for ordered decoding, while $p_t = 8.3\%$ for regular decoding used during lattice surgery. A brief explanation for this phenomenon is given in Appendix D 1. The fact that tCNOT thresholds for dominant erasure noise are strictly lower than the corresponding joint-measurement thresholds will ultimately affect the relative error rates of erasure-dominated transversal logical operations.

Similar behavior is seen for conventional erasures, where $p_t = 4.4\%$ ($p_t = 5.0\%$) at $R_e \approx 1$ for ordered decoding on tCNOTs (decoding an SCQM). Note that conventional erasure thresholds at high R_e are approximately half those of biased erasures. The significant increase in threshold at high R_e for all logical operations considered, and the improved performance of biased erasures in comparison to other structured-noise models, demonstrates that the Pauli < erasure < biased erasure hierarchy of suppressed logical error rates and reduced hardware requirements demonstrated for SCQMs [37] are retained for logical operations.

VII. CONCLUSIONS

In this work, we have performed an analysis of error correction of transversal CNOTs (tCNOTs) on surface codes in the context of scalable quantum computation. We have highlighted that individually fault-tolerant gadgets with $O(d)$ separation prevent possible exponentially growing decoding time in large-scale quantum algorithms. In this context, we have presented a unified framework to describe various decoding strategies that may be used to correct tCNOT operations, focusing on an intuitive strategy, ordered decoding, that uses the deterministic propagation of Pauli errors through the logical gate. This strategy has been compared with previous proposals to correct transversal logical operations—combined hypergraph decoding, and single-update decoding, showing that ordered decoding maintains the graph-based error-correction advantages and thresholds of surface-code memory experiments.

We have extended our analysis in several ways: we have studied the special case of tCNOTs used for logical-state teleportation, showing that the resultant hypergraph reduces to a graph in this instance, allowing regular surface-code decoders to be used. We have next performed a comparison of the transversal approach versus the joint-measurement-based lattice-surgery strategy, noting the possible reduced overhead of the former, with the caveat that the provided analysis is somewhat artificial given the differing optimal compilation schemes of the two strategies. Finally, we have presented an analysis of error correction of CNOTs under erasure-based noise models, showing a resultant increase in thresholds for dominant erasure noise for transversal- and lattice-surgery-based operations.

We now briefly turn to a discussion focusing on hardware limitations. Most metrics that we have studied

indicate that, where permitted by the architecture, a transversal strategy is advantageous: tCNOTs corrected using ordered decoding have similar thresholds and lower logical error rates compared to lattice-surgery-based approaches. There are, however, some subtle caveats. Transversal CNOT gates intrinsically involve nonlocal connectivity. Realistically, long-range interactions for a particular hardware may be slower and/or exhibit higher error rates than nearest-neighbor interactions. Specifically, for the example of reconfigurable neutral atoms, the corresponding error contribution arises from qubit movement across distances scaling with the surface-code size [13,70]. As another example, for static neutral-atom systems with long-range interactions realized via the Rydberg blockade, two-qubit-gate fidelity decays quasilinearly with the gate range [71,72], setting a maximum radius over which the use of such nonlocal connectivity is practical. Thus, the relative performance of these gate strategies will vary significantly based on hardware constraints.

Our work thus builds toward an analysis of fault-tolerant transversal logical operations for use in quantum algorithms to be implemented in hardware. Further study of transversal non-Clifford gates [73,74], decoders for logical gadgets, and optimal compilation schemes [68] that use a unified circuit-decoder perspective will indicate the performance of these strategies as a whole.

Note added. Before publication, we became aware of a related decoder implementation investigating low-depth circuits that uses the same underlying principle as ordered decoding [31].

ACKNOWLEDGMENTS

We are grateful to Yue Wu, Shraddha Singh, Pei-Kai Tsai, Qile Su, and Aleksander Kubica for helpful discussions. In particular, we thank Jahan Claes for insightful contributions during early stages of this project. We acknowledge the Yale Center for Research Computing for use of the Grace cluster. This work was supported by the National Science Foundation (NSF) through Quantum Leap Challenge Institutes (QLCI) Grant No. OMA-2120757. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF.

APPENDIX A: PROPAGATED DEFECTS IN THE DYNAMIC FRAME

Here, we expand on the functioning of the single-update decoder. Recall that it operates in the dynamic frame. In terms of the measured static checks, we have the dynamic frame detectors

$$v_{\mathrm{CX}}^{d} = S_{\mathrm{CX}}^{d-1} \oplus S_{\mathrm{CX}}^{d}, \tag{A1a}$$

$$v_{\mathrm{CX}}^{d+1} = S_{\mathrm{CX}}^{d} \oplus S_{\mathrm{CX}}^{d+1} \oplus S_{\mathrm{TX}}^{d+1}, \quad \text{and} \tag{A1b}$$

$$v_{\mathrm{CX}}^{d+2} = S_{\mathrm{CX}}^{d+1} \oplus S_{\mathrm{TX}}^{d+1} \oplus S_{\mathrm{TX}}^{d+2} \oplus S_{\mathrm{CX}}^{d+2}. \tag{A1c}$$

We illustrate the behavior of the decoder using an instance of a data-qubit error $Z_{T,q}$ on a qubit q in T that creates defects on T at round k:

(1) If $k \leq d$: after round d, $Z_{T,q}$ is copied over to the control, creating $Z_{C,q}$. Since $\left[Z_{T,q}Z_{C,q}, \mathbf{S}_{\mathrm{CX}} = S_{\mathrm{CX}}S_{\mathrm{TX}}\right] = 0$, no defect is created in the dynamic frame after the tCNOT despite the actual error being present. This behavior differs from the static frame.

(2) If $k > d$: $Z_{T,q}$ is not copied over to C but creates new defects on the updated C due to the updated checks of the dynamic frame: $\left[Z_{T,q}, \mathbf{S}_{\mathrm{CX}}\right] \neq 0$. These errors are not truly on the control and the resulting "false defects" simply mirror defect patterns on the target. Note that these false defects appear in addition to defects created by errors on the target itself, resulting in an effective doubling of the defect rate post-tCNOT on T.

We observe that use of the dynamic frame prevents propagated errors from creating defects on C, while creating false defects from errors that did not propagate. After independently decoding and correcting both subgraphs in the dynamic frame, we use knowledge of all the corrections applied to T to apply a final logical update to C and restore the code states (Algorithm 1).

APPENDIX B: HYPEREDGES IN THE tCNOT

Having discussed how the tCNOT copies over Z errors from T to C in the main text, here we address how, in a certain frame, the tCNOT induces nondecomposable hyperedges, i.e., leads to certain independent error mechanisms

ALGORITHM 1. Single-update decoder.

```
   /* Change dependent graphs to dynamic frame */
 1 foreach S^i_dep ∈ G_dep | i > d do
 2 |  S^i_dep ← S^i_dep ⊕ S^i_ind
 3 end
 4 foreach v^i_S ∈ G_dep do
 5 |  v^i_S ← S^{i-1}_dep ⊕ S^i_dep
 6 end
   /* Updating dependent subgraph edge weights */
 7 foreach e = (v^i_{S_k}, v^j_{S_l}) ∈ G_dep do
 8 |  if ∃ v^i_{S_k} ∈ e | i > d then
 9 |  |  p_e ← p_e + p_{eG_ind} − 2 p_e p_{e_{G_ind}}
10 |  end
11 end
12 Decode {G_ind, G_dep} independently
   /* Update G_dep's logical status */
13 L^{G_dep} ← L^{G_dep} ⊕ L^{G_ind}
```

creating more than two defects in the decoding graph G, such that they cannot be expressed as products of regular edges. Our discussion is framed in the language of a Stim circuit intended for PyMatching [20,46] that uses a decoding graph with components in both the dynamic and static frames, i.e., the graph exists in a *hybrid* frame. This versatility is facilitated by the ability of users to define detectors in Stim. We use the notation of Sec. IV, focusing on Z errors and X checks. Our system has the following three types of user-defined detectors:

(1) Until the tCNOT, the detectors are $v^i_{\mathrm{CX}} = S^{i-1}_{\mathrm{CX}} \oplus S^i_{\mathrm{CX}}$ and $v^i_{\mathrm{TX}} = S^{i-1}_{\mathrm{TX}} \oplus S^i_{\mathrm{TX}}$ for $i \in \{1, 2, \ldots, d\}$. These detectors apply to both the static and dynamic frames.

(2) Since the tCNOT copies over errors from T to C, it is natural to assume that if no error occurs between measurement rounds d and $d+1$, then $S^{d+1}_{\mathrm{CX}} = S^d_{\mathrm{CX}} \oplus S^d_{\mathrm{TX}}$. These three measurement outcomes can thus be used to define a detector $v^{d+1}_{\mathrm{CX}} = S^{d+1}_{\mathrm{CX}} \oplus S^d_{\mathrm{CX}} \oplus S^d_{\mathrm{TX}}$. Detectors on T remain unchanged: $v^{d+1}_{\mathrm{TX}} = S^{d+1}_{\mathrm{TX}} \oplus S^d_{\mathrm{TX}}$. These detectors exist in a dynamic frame that is defined in a slightly different basis to Sec. IV.

(3) After the tCNOT, in each individual surface code, stabilizer measurements should remain unchanged under error-free execution. This allows us to propose the following: $v^i_{\mathrm{CX}} = S^{i-1}_{\mathrm{CX}} \oplus S^i_{\mathrm{CX}}$ and $v^i_{\mathrm{TX}} = S^{i-1}_{\mathrm{TX}} \oplus S^i_{\mathrm{TX}}$ for $i \in \{d+2, d+3, \ldots, 2d\}$. Note that for C, these detectors are different from those defined by check evolution in the dynamic frame and are instead in the static frame.

With this set of detectors in the hybrid frame, it is possible to extract independent error mechanisms that create three defects. For example, consider a measurement error on T at round d. Assuming no other errors, $S^d_{\mathrm{TX}} = 1$ and all other outcomes are 0, meaning that the set of detectors $\{v^d_{\mathrm{TX}}, v^{d+1}_{\mathrm{CX}}, v^{d+1}_{\mathrm{TX}}\} = 1$ produces defects. This three-component term is a hyperedge.

APPENDIX C: GRAPH DECODERS FOR THE tCNOT

We provide a brief summary of the decoding algorithms based on independent handling of X and Z errors described in Sec. IV. By abuse of notation, we define $G_{\mathrm{ind}} = \{G_{\mathrm{TX}}, G_{\mathrm{CZ}}\}$ and $G_{\mathrm{dep}} = \{G_{\mathrm{CX}}, G_{\mathrm{TZ}}\}$. For every edge e of weight w in a graph G, we define $p_e = \exp(-w)/(1+\exp(-w))$

We note a fundamental conceptual similarity between ordered decoding and the two-pass or iterative MWPM-decoding methods [68,75–77]. These approaches have been used to improve SCQM-decoding accuracy by leveraging probabilistic intra-SC correlations. In contrast, for ordered decoding, we make use of deterministic inter-SC error correlations induced by the tCNOT. It may be possible to combine the two approaches, along with other subroutines that make use of soft information such as belief propagation (BP) [61], to further improve the performance of ordered decoding.

ALGORITHM 2. Ordered decoder.

Decode G_{ind} independently, obtaining the set of matched defect pairs $\mathcal{E}^{G_{ind}}$
$L' = 0$
`/* Flip nodes in dependent subgraphs */`
foreach unordered pair $e = (v^i_{S_k}, v^j_{S_l}) \in \mathcal{E}^{G_{ind}}$ **do**
  if $\exists v^i_{S_k} \in e \mid i \le d$ **then**
    $v^{d+1}_{S_k,dep} \leftarrow v^{d+1}_{S_k,dep} \oplus 1$
    $v^{d+1}_{S_l,dep} \leftarrow v^{d+1}_{S_l,dep} \oplus 1$
    $L' \leftarrow L' + L(e)$
  end
end
Decode G_{dep} independently
`/* Update` G_{dep}`'s logical status */`
$L^{G_{dep}} \leftarrow L^{G_{dep}} \oplus L'$

1. A phenomenological threshold comparison

Hypergraph decoders on surface codes show an advantage over MWPM in the presence of Y errors. This is because they can directly treat Y errors using hyperedges. In contrast, an MWPM decoder must decompose Y errors into independent X and Z errors and corresponding graphs. This decomposition results in the loss of initial $Y = XZ$ correlations. Here, we attempt to control for this possible Y error advantage of HUF during tCNOT decoding.

We perform a comparison between HUF and ordered-MWPM decoding for a tCNOT using a tailored phenomenological-noise model. In this model, independent bit-flip errors on data qubits and measurement errors are applied with equal probability p in each stabilizer-measurement round. This removes Y errors, allowing a direct comparison of the ability of the two decoders to correct correlated errors that arise solely from error propagation through tCNOT gates.

We study the same tCNOT system as in Sec. IV, with the results shown in Fig. 8. Similar to the circuit-level model, we find a higher threshold using an ordered-MWPM decoder ($p_t = 2.77\%$) compared to HUF ($p_t = 2.47\%$). Additionally, the logical error rate of ordered MWPM is slightly lower than that of HUF, reversing the general trend for circuit level noise that includes Y errors seen in Fig. 3.

These findings suggest that an ordered-MWPM strategy outperforms HUF when correcting correlated errors solely arising from tCNOT gates. Further, it appears that much of the logical-error-rate advantage of HUF indeed arises from its improved handling of Y errors.

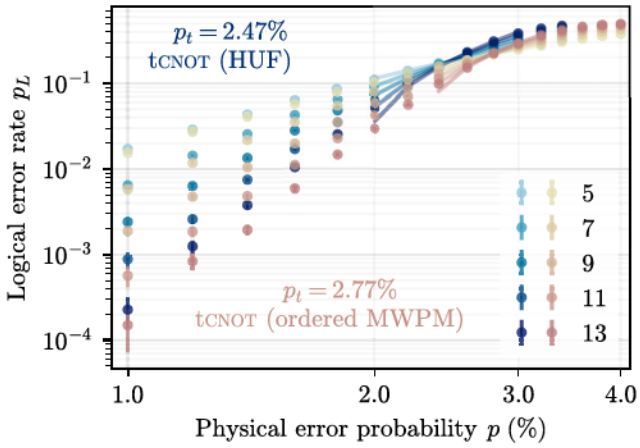

FIG. 8. Thresholds and logical error rates of a HUF decoder and an ordered-MWPM decoder for a tCNOT under biased phenomenological noise. Thresholds are found by finite-size scaling close to p_t (translucent lines).

To improve ordered MWPM, it may be useful to insert a BP subroutine prior to matching [61]. There is, however, a slight subtlety: naive BP will reweight matching edges using syndromes caused by propagated errors that do not in fact originate from the relevant code block. This may reduce the benefits of BP with ordered decoding in comparison to BP in an SCQM MWPM decoder.

APPENDIX D: UNCORRECTABLE PROPAGATED ERRORS FOR ORDERED DECODING ON THE tCNOT

Here, we discuss the logical error rates observed for ordered-MWPM decoding. For Z errors, ordered decoding uses the simple concept of a target-first–control-next decoding strategy. The first round of decoding on T pairs up defects on the X-stabilizer graph. These defect pairs correspond to errors that may or may not be copied over to C by the tCNOT. Assuming that errors are corrected up to stabilizers, these defect pairs arising from underlying physical errors and their corresponding predicted errors returned by a decoder create spacetime cycles in the decoding graph (Fig. 9). These cycles can be broadly classified into three distinct categories:

(1) Spacetime cycles entirely before the tCNOT: the error corresponding to this defect pattern can be a combination of measurement and data errors. Only the data-qubit part of this error, corresponding to its horizontal projection in spacetime, is transmitted to C. Even if the original error is corrected up to a timelike stabilizer, the dependent subgraph is correctly updated.

(2) Spacetime cycles entirely after the tCNOT: the most probable error occurs entirely after the tCNOT and hence does not propagate onto C.

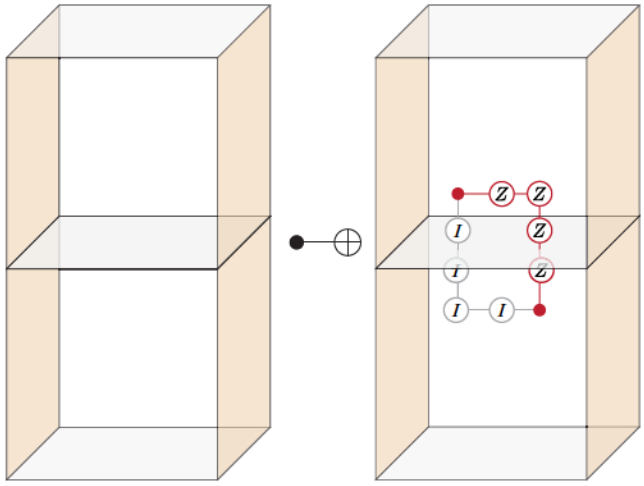

FIG. 9. Errors can span the tCNOT measurement round, creating a defect below the tCNOT paired to one above it. Depending on where the data-qubit errors actually occurred, different errors are copied over to G_{dep} for the same defect pattern. If the error in red occurred, C remains unaffected.

(3) Spacetime cycles spanning the tCNOT round: this defect pattern arises from a combination of measurement and data errors. There are multiple minimal-weight ways of projecting such a path (see, e.g., Fig. 9). Suppose that the decoder identifies the gray path with the "data-qubit projection" entirely below the tCNOT but the actual physical error was the red error with the "data-qubit projection" above the tCNOT. The former will transmit errors to the target but the latter will not. Given the choice made by the decoder, the target will be updated incorrectly. Hereon, we refer to this class of errors as *ambiguous* errors.

Ambiguous errors arising from spacetime cycles spanning the tCNOT round are effectively uncorrectable by the first stage of an ordered decoder—the decoder can only randomly choose the truly correct data-error projection. Subsequent error correction of the residual defects takes place on the dependent subgraph. As a result, ambiguous errors lead to an increase in the logical error rate of the dependent graphs. However, from Fig. 4, we observe that they do not materially degrade decoder thresholds.

1. Ordered decoding at high R_e

Here, we discuss the apparent reduction in threshold at high erasure fractions for a tCNOT with ordered decoding compared to an SCQM experiment. In essence, this is because of ambiguous errors that create cycles around the tCNOT round. For all other categories of correctable erasure

errors, the decoder can use the erasure flag on qubits suffering errors to determine the exact path of errors propagating over onto G_{dep}.

Observe the example in Fig. 9. Unlike the previous section, we envision that erasure errors have occurred along both gray and red paths. However, only one (red) path truly has Z errors due to the erasures, giving rise to two defects. Exactly like the case with Pauli errors, the decoder has no way of determining which of the two error patterns has occurred. If it guesses that the gray path has occurred, an incorrect update is applied to the control. At high p and R_e, the probabilities of such cycles increase. At sufficiently high pR_e, logical errors are dominated by contributions from these ambiguous cycles, for which the decoding ability is Pauli-like instead of erasurelike. This in turn reduces p_t for the tCNOT at high erasure fractions.

APPENDIX E: LOGICAL COMPUTATION USING LATTICE SURGERY

An SCQM requires at least d errors to create an undetectable logical operator. Ideally, this distance should be preserved while using surface codes as units of logical computation. Logical gates executed via lattice surgery achieve this in a manner compatible with planar fixed-qubit architectures [16,65].

We illustrate the process for a joint-Pauli XX measurement done via lattice surgery on two surface codes in Figs. 10(a) and 10(b). The system is set up as follows: two initially disconnected surface codes have their individual stabilizers measured for $O(d)$ rounds. A bridging region of width b between the X-logical edges of these codes is initialized with unentangled qubits in $|0\rangle$ [Fig. 10(a)]. To begin the JP measurement, surface-code checks are measured over the entire region containing the two logical qubits and the bridging region. This projects the two surface codes into a joint logical subspace.

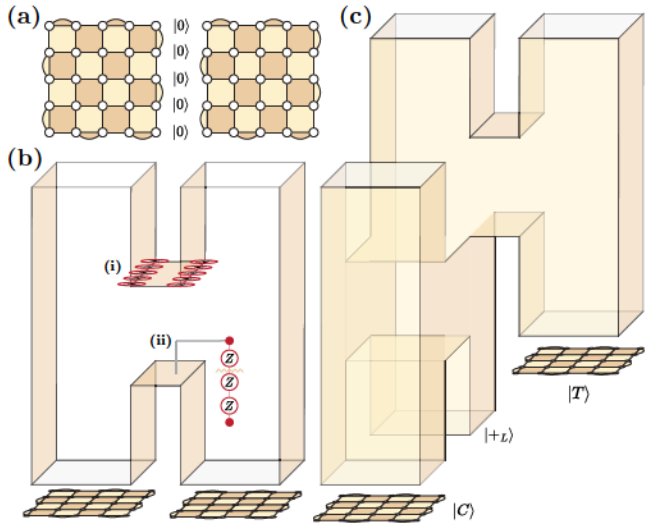

FIG. 10. The (a) setup and (b) spacetime volume of an XX logical measurement conducted using lattice surgery, with (i) the product of stabilizers denoting the XX-measurement outcome and (ii) an error that will be misidentified without the buffer measurement rounds. (c) The spacetime-decoding volume for a logical CNOT done with lattice surgery.

In the absence of errors, all check measurements in the logical region and Z-stabilizer measurements in the bridging region return $+1$ outcomes. X stabilizers in the bridging region are initially unfixed and yield random outcomes but their combined product is the XX logical-measurement outcome [Fig. 10(b)(i)]. To be robust against measurement errors that may otherwise lead to misidentification of the XX-measurement result, JP measurements are done for $O(d)$ rounds. After these JP rounds, the bridging-region qubits are measured in the Z basis and the surface codes are once again measured independently. The spacetime-decoding volume of this process is shown in Fig. 10(b).

Extending this strategy, a logical CNOT gate between a control surface code C and a target surface code T can be performed using a similar JP-measurement-based circuit. This operation uses a logical ZZ measurement on C and a logical ancilla patch, followed by a logical XX measurement on the ancilla and T. In practice, the spacetime volume of this set of operations looks like Fig. 10(c), where $O(2d)$ round of gates along with one logical ancilla are used to perform the complete gate. Note that single-control n-target CNOT gates can be carried out simultaneously by using a large ancillary region that can support $X^{\otimes n}$ measurements [24,66]. This CNOT operation has the same threshold as a SCQM and the same logical error rate scaled up to known constant prefactors (discussed in Appendix F). This is verified via numerical simulations in Fig. 4.

APPENDIX F: ERROR CORRECTION IN LATTICE SURGERY

We expand on decoding and correction of errors during the XX JP measurement discussed in Appendix E. Close observation of the "legs" of the H shape in Fig. 10(b)—representing pre-JP stabilizer measurement rounds on the individual surface codes, hereon referred to as *buffer rounds*—shows that they are equivalent to the spacetime volume of individual SCQM experiments (up to time boundaries that connect them to the JP rounds). The bridging region in the JP rounds forms a stability experiment [78]. In this stability experiment, the product of X stabilizers across the width of the bridging region is tracked. This product can be thought of as a horizontal surface in spacetime, as seen in Fig. 10. Recall that an SCQM tracks the status of a logical operator across time. Thus, the corresponding product tracked is a surface perpendicular to the stability experiment, i.e., the side of the cube in Fig. 2(b).

As a result, the stability experiment can be interpreted as a spacetime-rotated version of the SCQM.

As a combination of differently orientated SCQMs, we can observe that the complete decoding graph of the logical XX measurement is essentially a trivially extended SCQM. As a result, in lattice surgery, we preserve the threshold of an equivalent SCQM. This has previously been demonstrated in Ref. [67] using a matching decoder. Further, the total logical error rate of a lattice-surgery operation can be expressed in terms of its logical spacetime surface area (LSSA) in comparison to the LSSA of an SCQM, a concept that we discuss in Appendix G.

An additional note of interest is the choice to use buffer rounds at all; we connect this to the notion of W in the main text. Technically, these legs of the H-shaped spacetime volume are not part of the JP measurement itself. Indeed, in the ideal scenario of perfect state preparation of the disjoint surface codes, or surface-code readout immediately after the JP rounds, they are not required. However, measurement errors in the original logical patches can create defects that can be misidentified as faulty JP measurement errors had measurement data from the buffer rounds not been used. This is exemplified in Fig. 10(b)(ii): if the decoding window does not extend below the first JP round, the lone defect above the window boundary is misidentified as arising from a JP X-stabilizer measurement error. Thus, we use W buffer rounds of decoding data for fault tolerance against these errors. Like the tCNOT, these buffer rounds can be part of other noncommuting JP measurements—inclusion of their measurement data is only necessary for decoding and they do not actually need to be perfectly disjoint rounds on individual surface codes [see Fig. 10(c)].

APPENDIX G: LOGICAL SURFACE AREAS FOR SC OPERATIONS

For the operations that we study, the failure rates of different logical-gate strategies at low physical error rates can be approximately mapped to the spacetime surface-area ratios of their minimum-weight logical operators. This is because the presence or absence of logical errors is directly indicated by the parity of a complete set of logical operators. We specifically choose an overcomplete minimum-weight set to maintain equivalence across different logical operations.

As an example, for a single SCQM, we use four logical operators of length d (two X logicals on opposite boundaries of the surface code and two Z logicals). Each logical is measured for d rounds, so the total logical spacetime surface area is $4 \times d \times d = 4d^2$. We count in units of d^2 and so this is simply 4 in our chosen unit system.

We now describe a 2SCQM, i.e., a memory experiment on two surface codes for $2d$ rounds, the equivalent logical spacetime surface area (LSSA) of which is.

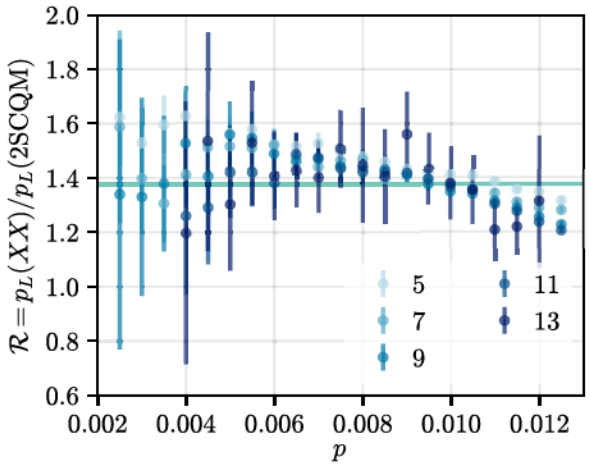

FIG. 11. The ratio of the logical failure rates for a XX measurement using lattice surgery with $b = 1$ to a 2SCQM. Each point uses 10^5 samples, discarding points with failure rates $\leq 10^{-4}$. The error bars represent the 95% confidence interval. A horizontal line is drawn at the lower bound of 1.375.

$$I_{2\text{SCQM}} = \overbrace{2}^{\text{SCs}} \times \overbrace{2}^{\text{rounds}/d^2} \times \overbrace{4}^{\text{SA per SC}} = 16.$$

Let us next examine an XX logical measurement on two surface codes. The bridging region consists of a width-b strip that uses bd qubits. Depending on the architecture, b can be chosen to be constant or scale linearly with d. The two surface codes are measured independently for d rounds, have projective measurements onto the XX basis using the bridging region for d rounds, and are then split and measured again for d rounds. The LSSA XX_{2SC+3d} for this operation is given by

$$2 \times \overbrace{\left(\overbrace{2 \times 4}^{\text{split rounds}} + \overbrace{3}^{\text{merge rounds}} \right)}^{\text{SCs for}} + \overbrace{4 \times b/d}^{\text{bridge}} = 22 + 4b/d.$$

We can thus find the logical error rate ratio of a lattice-surgery XX measurement versus a memory experiment as

$$\frac{XX_{2SC+3d}}{I_{2SCQM}} = 1.375 + \frac{b}{4d}. \tag{G1}$$

This is verified numerically in Fig. 11, which uses $b = 1$.

We can similarly calculate the LSSA for a logical CNOT between two surface codes using lattice surgery. As described in Appendix F, this operation takes a total of $4d$

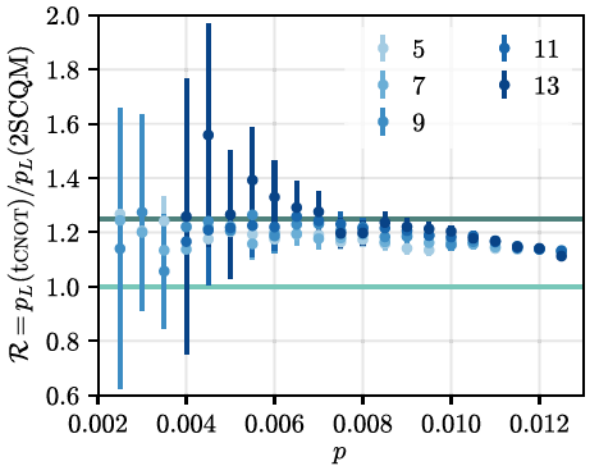

FIG. 12. The ratio of the logical failure rates for a tCNOT with ordered decoding to a 2SCQM. Each point uses 10^5 samples, discarding points with failure rates $\leq 10^{-4}$. The error bars represent the 95% confidence interval. The horizontal lines are drawn at 1 and 1.25.

rounds. Its LSSA CX_{2SC+4d} is

$$\overbrace{3\times 2\times 4}^{\text{independent SCs}} + \overbrace{3\times 4}^{\text{SCs while merging}} + \overbrace{4\times 2\times b/d}^{\text{bridge}} = 36 + 8b/d.$$

We can thus verify the logical error rate ratio of a lattice surgery CNOT versus a memory experiment as

$$\frac{\text{CX}_{2SC+4d}}{I_{\text{2SCQM}}} = 2.25 + \frac{b}{2d}. \tag{G2}$$

1. LSSAs for tCNOTs

We now calculate the magnification of error rates induced by ordered decoding at low p. The overall failure rate of the tCNOT is the probability that the decoder applies a logically incorrect update to any of the four $\{G_{\text{CX}}, G_{\text{CZ}}, G_{\text{TX}}, G_{\text{TX}}\}$ decoding subgraphs. For a 2SCQM, these four rates are equivalent. For a tCNOT with ordered decoding, we take the dependent subgraphs to have a 50% increase in logical failure rates compared to a 2SCQM because they use their own LSSA combined with half the LSSA of their corresponding independent subgraph. The independent subgraphs remain unaffected. Thus, the logical rate amplifier is $(2 + 2 \times 1.5)/(2 + 2) = 1.25$.

We verify these calculations in Fig. 12. We find the ratio $\mathcal{R}$ of the numerically obtained logical failure rates for a tCNOT corrected using ordered decoding to that of a 2SCQM for various system sizes. By virtue of incorrectly propagated logical updates, ambiguous errors, and errors introduced within the faulty physical CNOT gates that implement the tCNOT, we observe that $\mathcal{R}$ is consistently higher than 1 and seems to saturate below threshold. By fitting to a saturation function, we find a saturation $\mathcal{R} \approx 1.25$ at low p, represented by the horizontal dark green line. This agrees with the preliminary analytical prediction above.

2. Multitarget CNOTs

We can now perform the same analysis for an n-target lattice-surgery CNOT, which can be executed in $2d$ rounds. Note that here the required number of extra bridging qubits is lower bounded at $\lceil n/2 \rceil bd$. We obtain the LSSA $\text{CX}_{\text{LS-n}}$:

$$\overbrace{3\times(n+1)\times 4}^{\text{independent SCs}} + \overbrace{3\times(2+(n+1))}^{\text{SCs while merging}} + \overbrace{4\times 2\times\lceil n/2\rceil bd}^{\text{bridge}}.$$

Comparing it to the identity operation,

$$\frac{\text{CX}_{\text{LS-n}}}{I_{(n+1)SC+2d}} = \frac{21 + 15n + \lceil n/2\rceil 8bd}{8(n+1)}. \tag{G3}$$

Similarly, we evaluate the LSSA of an n target tCNOT with ordered decoding. Note that this can either be executed natively in hardware or using consecutive single-target tCNOTs, with the block then requiring a subsequent W buffer rounds, without exacerbating the decoding complexity. We can then compare it to the identity operation,

$$\frac{\text{CX}_{\text{t-n}}}{I_{(n+1)SC+2d}} = \frac{(n+1) + n\times 1.5 + (1+0.5n)}{2(n+1)}. \tag{G4}$$

APPENDIX H: AN ISOLATED LS-CNOT VERSUS A tCNOT

Here, we discuss our results benchmarking the performance of a fault-tolerant tCNOT against a lattice-surgery CNOT, in a setting in which state preparation and measurement do not immediately occur before and after the logical operation. We highlight that this comparison is quite artificial, given that (1) in practice for transversal circuits, it may be more advantageous to use $m, g > 1$ tCNOTs in a logical-decoding block (see Sec. III), and (2) in practice for joint-measurement based circuits, Clifford gates can be time-efficiently compiled [79]. A more realistic analysis is left to future work.

We find that for a tCNOT, the total logical error rate is only marginally amplified compared to a 2SCQM experiment. However, it more than doubles for a LS-CNOT on account of that fact that it needs two logical Pauli measurements (for an analysis, see Appendix. G). Further, the lattice-surgery approach requires an additional ancilla surface-code patch, as well as bridging regions of uninitialized qubits. This results in a total additional qubit overhead of $d^2 + 2bd$, where b is the width of the bridging region of qubits. b depends on the architecture chosen: for movable architectures, this can be $O(1)$, but for fixed architectures this is generally the separation of the surface-code patches. These extra qubits are not needed for a tCNOT.

MWPM on a sparse surface-code graph $G = (V, E)$ generally takes $O(V^\alpha)$ time, where $\alpha \leq 3$ depending on

TABLE II. A preliminary overhead comparison for an isolated fault-tolerant logical CNOT performed via lattice surgery versus transversally with ordered decoding. We take a MWPM-decoding subroutine that scales as $O(V^\alpha)$ in the number of qubits V, resulting in both CNOT strategies having a threshold of 1%. We benchmark the number of additional qubits and measurement rounds needed for both strategies, along with the decoding-time complexity and total logical error rate in comparison to a 2SCQM.

Method of conducting logical CNOT	Number of additional qubits	Number of measurement rounds	Decoding-time complexity	Minimal patch movement	Amplification of logical error rate over 2SCQM
Lattice surgery	d^2+2bd	$2d$	$O((10d^3+2bd^2)^\alpha)$	Y	$2.25+b/2d$ [Appendix G]
tCNOT + ordered decoding	0	d	$2\times O((2d^3)^\alpha)$	N	~1.25 [Appendix G]

the MWPM implementation [19,80,81]. The time overhead for ordered-MWPM decoding of the tCNOT scales as $2\times O((d^3)^\alpha)$. The factor of 2 arises since the independent and dependent graphs are decoded serially. The same quantity is $O((10d^3+2d^2b)^\alpha)$ for lattice surgery, implying that despite the inherent latency of ordered decoding, a correction $\mathcal{C}$ can be identified faster than an equivalent lattice-surgery instance. The results of this comparison are summarized in Table II.

APPENDIX I: DETAILS OF NUMERICAL SIMULATIONS

The simulations based on MWPM decoders for the SCQM and transversal CNOT operations have been implemented in C++ using the BLOSSOM algorithm [80,82], combined with Djikstra to account for adjusted edge weights arising from erasures. The code for these is available at Ref. [83]. The corresponding HUF-based simulations have been implemented in Stim [46] and decoded using the MWPF implementation of HUF [64]. A second implementation of ordered decoding in Stim using PyMatching [20,46] has additionally been developed for further testing.

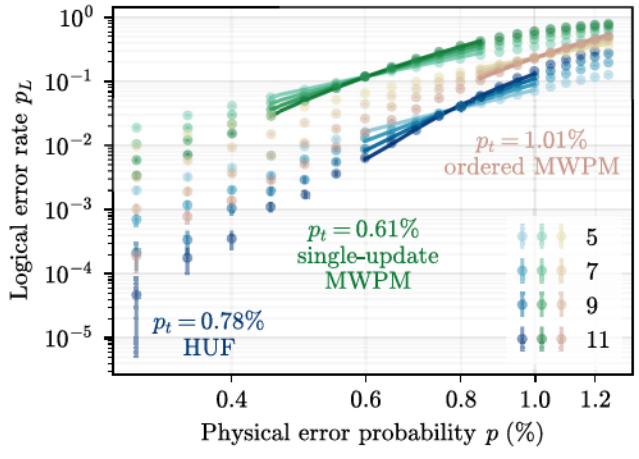

FIG. 13. Thresholds and logical error rates of a HUF decoder, a single-update decoder, and an ordered-MWPM decoder for a tCNOT under 2Q noise. The data are a subset of that in Fig. 4.

For the lattice-surgery analysis, simulations and measurements of the logical error rate have been conducted for a logical XX-measurement instance using the procedure detailed in Ref. [67] in C++, using BLOSSOM and Djikstra. The results reported for the CNOT have then been calculated by scaling according to the respective LSSAs of the operations.

For data displayed in Figs. 4 and 6, individual points are collected using 10^5 Monte Carlo samples; error bars are found by jackknife resampling and represent the 95% confidence interval. Note that we report the total logical error rate per operation, i.e., over $2d$ measurement rounds for tCNOTs, as opposed to the error rate per syndrome extraction round.

1. 2Q noise model

In this noise model, used for all simulations in the main text, two-qubit-gate errors are uniformly chosen at random from $\{I,X,Y,Z\}^{\otimes 2}/\{I\otimes I\}$ at a rate p. This principle also applies to the physical CNOT gates used in the logical tCNOT. In order to simplify analysis, other noise sources such as state preparation, measurement, idling, and single-qubit gate errors are not included. This particular noise model has previously been studied in Ref. [16] (Fig. 7 of this work displays per-round thresholds), showing that the excluded classes of errors contribute significantly less to logical error rates. In Fig. 13, we additionally show a subplot of Fig. 4 that only displays a comparison between tCNOT decoders under the 2Q noise model.

2. SD6 noise model

Several recent works have used a more general noise model that includes measurement, single-qubit gate, and

TABLE III. The errors per operation in terms of a global depolarizing-noise probability p for the 2Q and SD6 noise models.

	$\{I,H\}$	$\{\mathrm{CX},\mathrm{CZ}\}$	M_z	R_z
2Q	0	p	0	0
SD6	p	p	p	p

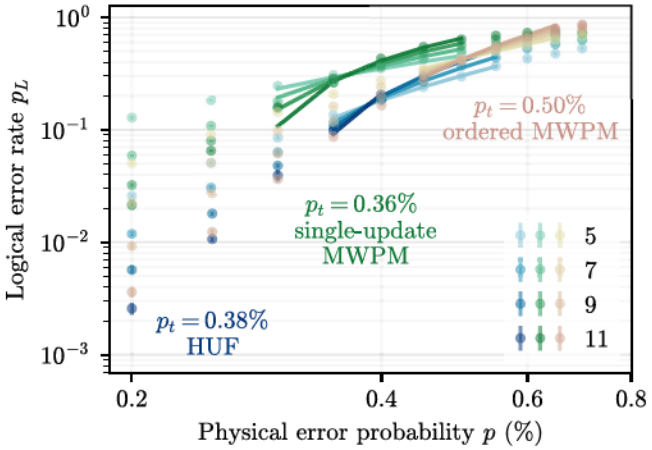

FIG. 14. Thresholds for tCNOT decoders under SD6 noise. Note the lower thresholds and higher logical error rates compared to Fig. 13.

idling errors [27,31]. In this section, we briefly turn to a similar noise model that we term SD6 [61]. For a global parameter p that denotes the probability of a depolarizing n-qubit noise channel following a specific physical operation on n qubits, we define and compare the SD6 and 2Q noise models in Table. III.

We show the thresholds and logical error rates for tCNOT decoders under the SD6 noise model in Fig. 14. Due to the inclusion of single-qubit error mechanisms, the SD6 thresholds are universally lower, and the logical error rates are universally higher, than their equivalents for 2Q noise shown in Fig. 13.

APPENDIX J: BIASED ERASURES

A biased erasure is defined as a heralded exit from one half of the computational subspace [32,37]. The effective Kraus operators for this channel can be written as

$$W_0 = |0\rangle\langle 0| + \sqrt{1-2p_e}\,|1\rangle\langle 1| \tag{J1}$$

$$W_e = \sqrt{2p_e}\,|1\rangle\langle 1|\,. \tag{J2}$$

This error channel is motivated by the metastable ^{171}Yb atom qubit, where the dominant decay mechanism is decay from the Rydberg level $|r\rangle$ to a set of external states during two-qubit gates. Since the qubit is only excited to $|r\rangle$ from the $|1\rangle$ state during these gates, detection of population in the external states is akin to measurement in the Z basis with a known outcome. The qubit can be reinitialized by replacement with a fresh atom in $|1\rangle$.

There are dual advantages to engineering qubits to experience dominant biased erasures as opposed to Pauli noise. First, biased erasures are heralded errors, which means that the logical failure rate scales as $p_L \propto p^d$. In contrast, the only information available to identify qubits that have experienced Pauli errors within the computational subspace is their syndrome, leading to a scaling of $p_L \propto p^{(d+1)/2}$. Second, one benefits from the *biased* nature of the erasure. If the error occurred during a CZ gate (or at the control of a CNOT gate), one can prove that the effective Kraus operator on the qubit after atom replacement and Pauli twirling is $(I\rho I + Z\rho Z)/2$. If it has occurred to the qubit at the target of a CNOT gate, the effective error channel is $(I\rho I + X\rho X)/2$. Biased erasures thus allow identification of the apparent error (X or Z) at a known location, lending to easier error correction and high thresholds.

For the metastable Yb, the theoretical maximum erasure fraction is $R_e = 98\%$ [34]. $R_e = 33\%$ has recently been achieved experimentally via detection of leakage to the ground state [84].

[1] P. W. Shor, in *Proceedings of 37th Conference on Foundations of Computer Science* (IEEE, Burlington, VT, USA, 1996).

[2] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, Phys. Rev. A **54**, 1098 (1996).

[3] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, M. Cetina, and C. Monroe, Fault-tolerant control of an error-corrected qubit, Nature **598**, 281 (2021).

[4] R. Acharya, *et al.*, Google Quantum AI, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676 (2023).

[5] V. V. Sivak, A. Eickbusch, B. Royer, S. Singh, I. Tsioutsios, S. Ganjam, A. Miano, B. L. Brock, A. Z. Ding, L. Frunzio, S. M. Girvin, R. J. Schoelkopf, and M. H. Devoret, Real-time quantum error correction beyond break-even, Nature **616**, 50 (2023).

[6] Google Quantum AI and Collaborators, Quantum error correction below the surface code threshold, Nature **638**, 920 (2025).

[7] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, and R. Blatt, Experimental repetitive quantum error correction, Science **332**, 1059 (2011).

[8] Z. Chen, K. J. Satzinger, J. Atalaya, A. N. Korotkov, A. Dunsworth, D. Sank, C. Quintana, M. McEwen, R. Barends, P. V. Klimov, S. Hong, C. Jones, A. Petukhov, D. Kafri, S. Demura, B. Burkett, C. Gidney, A. G. Fowler, A. Paler, and H. Putterman, Google Quantum AI, Exponential suppression of bit or phase errors with cyclic error correction, Nature **595**, 383 (2021).

[9] M. P. da Silva, *et al.*, Demonstration of logical qubits and repeated error correction with better-than-physical error rates, ArXiv:2404.02280.

[10] Y. Kim, M. Sevior, and M. Usman, Transversal CNOT gate with multi-cycle error correction, ArXiv:2406.12267.

[11] C. Ryan-Anderson, N. C. Brown, C. H. Baldwin, J. M. Dreiling, C. Foltz, J. P. Gaebler, T. M. Gatterman, N. Hewitt, C. Holliman, C. V. Horst, *et al.*, High-fidelity teleportation of a logical qubit using transversal gates and lattice surgery, Science **385**, 1327 (2024).

[12] S. Huang, K. R. Brown, and M. Cetina, Comparing Shor and Steane error correction using the Bacon-Shor code, Sci. Adv. **10** (2024).
[13] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, V. Vuletić, and M. D. Lukin, A quantum processor based on coherent transport of entangled atom arrays, Nature **604**, 451 (2022).
[14] C. J. Trout, M. Li, M. Gutiérrez, Y. Wu, S.-T. Wang, L. Duan, and K. R. Brown, Simulating the performance of a distance-3 surface code in a linear ion trap, New J. Phys. **20**, 043038 (2018).
[15] D. Bluvstein, *et al.*, Logical quantum processor based on reconfigurable atom arrays, Nature **626**, 58 (2024).
[16] 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).
[17] Y. Tomita and K. M. Svore, Low-distance surface codes under realistic quantum noise, Phys. Rev. A **90**, 062320 (2014).
[18] A. G. Fowler, Proof of finite surface code threshold for matching, Phys. Rev. Lett. **109**, 180502 (2012).
[19] Y. Wu and L. Zhong, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, Bellevue, WA, USA, 2023).
[20] O. Higgott and C. Gidney, Sparse blossom: Correcting a million errors per core second with minimum-weight matching, Quantum **9**, 1600 (2025).
[21] R. Raussendorf and J. Harrington, Fault-tolerant quantum computation with high threshold in two dimensions, Phys. Rev. Lett. **98**, 190504 (2007).
[22] B. J. Brown, K. Laubscher, M. S. Kesselring, and J. R. Wootton, Poking holes and cutting corners to achieve Clifford gates with the surface code, Phys. Rev. X **7**, 021029 (2017).
[23] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, New J. Phys. **14**, 123011 (2012).
[24] A. G. Fowler and C. Gidney, Low overhead quantum computation using lattice surgery, ArXiv:1808.06709.
[25] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, J. Math. Phys. **43**, 4452 (2002).
[26] M. E. Beverland, A. Kubica, and K. M. Svore, Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes, PRX Quantum **2**, 020341 (2021).
[27] M. Cain, C. Zhao, H. Zhou, N. Meister, J. P. B. Ataides, A. Jaffe, D. Bluvstein, and M. D. Lukin, Correlated decoding of logical algorithms with transversal gates, Phys. Rev. Lett. **133**, 240602 (2024).
[28] H. Zhou, C. Zhao, M. Cain, D. Bluvstein, C. Duckering, H.-Y. Hu, S.-T. Wang, A. Kubica, and M. D. Lukin, Algorithmic fault tolerance for fast quantum computing, ArXiv:2406.17653.
[29] The basic concept behind ordered decoding has been mentioned in a single sentence in Ref. [26]. However, no concrete implementation or analytical or numerical results have been provided.
[30] J. Viszlai, S. F. Lin, S. Dangwal, J. M. Baker, and F. T. Chong, An architecture for improved surface code connectivity in neutral atoms, ArXiv:2309.13507.
[31] K. H. Wan, M. Webber, A. G. Fowler, and W. K. Hensinger, An iterative transversal CNOT decoder, ArXiv:2407.20976.
[32] M. Grassl, T. Beth, and T. Pellizzari, Codes for the quantum erasure channel, Phys. Rev. A **56**, 33 (1997).
[33] T. M. Stace, S. D. Barrett, and A. C. Doherty, Thresholds for topological codes in the presence of loss, Phys. Rev. Lett. **102**, 200501 (2009).
[34] 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).
[35] M. Kang, W. C. Campbell, and K. R. Brown, Quantum error correction with metastable states of trapped ions using erasure conversion, PRX Quantum **4**, 020358 (2023).
[36] J. D. Teoh, P. Winkel, H. K. Babla, B. J. Chapman, J. Claes, S. J. De Graaf, J. W. O. Garmon, W. D. Kalfus, Y. Lu, A. Maiti, K. Sahay, N. Thakur, T. Tsunoda, S. H. Xue, L. Frunzio, S. M. Girvin, S. Puri, and R. J. Schoelkopf, Dual-rail encoding with superconducting cavities, Proc. Natl. Acad. Sci. **120**, e2221736120 (2023).
[37] K. Sahay, J. Jin, J. Claes, J. D. Thompson, and S. Puri, High-threshold codes for neutral-atom qubits with biased erasure errors, Phys. Rev. X **13**, 041013 (2023).
[38] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary, ArXiv:quant-ph/9811052.
[39] H. Bombin and M. A. Martin-Delgado, Optimal resources for topological two-dimensional stabilizer codes: Comparative study, Phys. Rev. A **76**, 012305 (2007).
[40] D. Gottesman, Stabilizer codes and quantum error correction, ArXiv:quant-ph/9705052.
[41] P. Iyer and D. Poulin, Hardness of decoding quantum stabilizer codes, IEEE Trans. Inf. Theory **61**, 5209 (2015).
[42] We consider S^0 and S^{d+1} to be initial and final perfect-check measurements, respectively.
[43] E. Berlekamp, R. McEliece, and H. Van Tilborg, On the inherent intractability of certain coding problems (Corresp.), IEEE Trans. Inf. Theory **24**, 384 (1978).
[44] N. Delfosse, V. Londe, and M. Beverland, Toward a union-find decoder for quantum LDPC codes, ArXiv:2103.08049.
[45] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Surface code quantum computing with error rates over 1%, Phys. Rev. A **83**, 020302 (2011).
[46] C. Gidney, STIM: A fast stabilizer circuit simulator, Quantum **5**, 497 (2021).
[47] A. M. Steane, Active stabilization, quantum computation, and quantum state synthesis, Phys. Rev. Lett. **78**, 2252 (1997).
[48] E. Knill, Fault-tolerant postselected quantum computation: Schemes, ArXiv:quant-ph/0402171.
[49] E. Knill, Quantum computing with realistically noisy devices, Nature **434**, 39 (2005).
[50] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, Quantum **5**, 433 (2021).

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

[52] H. Bombín, Single-shot fault-tolerant quantum error correction, Phys. Rev. X **5**, 031043 (2015).

[53] E. T. Campbell, A theory of single-shot error correction for adversarial noise, Quantum Sci. Technol. **4**, 025006 (2019).

[54] Y. Lin, S. Huang, and K. R. Brown, Single-shot error correction on toric codes with high-weight stabilizers, Phys. Rev. A **109**, 052438 (2024).

[55] N. Delfosse, B. W. Reichardt, and K. M. Svore, Beyond single-shot fault-tolerant quantum error correction, IEEE Trans. Inf. Theory **68**, 287 (2022).

[56] L. Skoric, D. E. Browne, K. M. Barnes, N. I. Gillespie, and E. T. Campbell, Parallel window decoding enables scalable fault tolerant quantum computation, Nat. Commun. **14**, 7040 (2023).

[57] H. Bombín, C. Dawson, Y.-H. Liu, N. Nickerson, F. Pastawski, and S. Roberts, Modular decoding: Parallelizable real-time decoding for quantum computers, ArXiv:2303.04846.

[58] X. Tan, F. Zhang, R. Chao, Y. Shi, and J. Chen, Scalable surface-code decoders with parallelization in time, PRX Quantum **4**, 040344 (2023).

[59] N. Delfosse and A. Paetznick, Spacetime codes of Clifford circuits, ArXiv:2304.05943.

[60] B. M. Terhal, Quantum error correction for quantum memories, Rev. Mod. Phys. **87**, 307 (2015).

[61] O. Higgott, T. C. Bohdanowicz, A. Kubica, S. T. Flammia, and E. T. Campbell, Improved decoding of circuit noise and fragile boundaries of tailored surface codes, Phys. Rev. X **13**, 031007 (2023).

[62] B. Hetényi and J. R. Wootton, Creating entangled logical qubits in the heavy-hex lattice with topological codes, PRX Quantum **5**, 040334 (2024).

[63] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, Quantum **5**, 595 (2021).

[64] Y. Wu <wuyue16pku@gmail.com>, mwpf: Hypergraph Minimum-Weight Parity Factor (MWPF) solver for quantum LDPC codes, https://hyper-mwpf.com.

[65] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, Quantum **3**, 128 (2019).

[66] D. Litinski and F. V. Oppen, Lattice surgery with a twist: Simplifying Clifford gates of surface codes, Quantum **2**, 62 (2018).

[67] C. Chamberland and E. T. Campbell, Universal quantum computing with twist-free and temporally encoded lattice surgery, PRX Quantum **3**, 010331 (2022).

[68] A. G. Fowler, Optimal complexity correction of correlated errors in the surface code, ArXiv:1310.0863.

[69] A. Kubica, A. Haim, Y. Vaknin, H. Levine, F. Brandão, and A. Retzker, Erasure qubits: Overcoming the T_1 limit in superconducting circuits, Phys. Rev. X **13**, 041022 (2023).

[70] S. J. Evered, D. Bluvstein, M. Kalinowski, S. Ebadi, T. Manovitz, H. Zhou, S. H. Li, A. A. Geim, T. T. Wang, N. Maskara, H. Levine, G. Semeghini, M. Greiner, V. Vuletić, and M. D. Lukin, High-fidelity parallel entangling gates on a neutral-atom quantum computer, Nature **622**, 268 (2023).

[71] L. Pecorari, S. Jandura, G. K. Brennen, and G. Pupillo, High-rate quantum LDPC codes for long-range-connected neutral atom registers, Nat. Commun. **16**, 1111 (2025).

[72] C. Poole, T. M. Graham, M. A. Perlin, M. Otten, and M. Saffman, Architecture for fast implementation of quantum low-density parity-check codes with optimized Rydberg gates, Phys. Rev. A **111**, 022433 (2025).

[73] J. E. Moussa, Transversal Clifford gates on folded surface codes, Phys. Rev. A **94**, 042316 (2016).

[74] B. J. Brown, A fault-tolerant non-Clifford gate for the surface code in two dimensions, Sci. Adv. **6**, eaay4929 (2020).

[75] Y. Yuan and C.-C. Lu, A modified MWPM decoding algorithm for quantum surface codes over depolarizing channels, ArXiv:2202.11239.

[76] A. Paler and A. G. Fowler, Pipelined correlated minimum weight perfect matching of the surface code, Quantum **7**, 1205 (2023).

[77] A. d. iOlius, J. E. Martinez, P. Fuentes, and P. M. Crespo, Performance enhancement of surface codes via recursive minimum-weight perfect-match decoding, Phys. Rev. A **108**, 022401 (2023).

[78] C. Gidney, Stability experiments: The overlooked dual of memory experiments, Quantum **6**, 786 (2022).

[79] A. G. Fowler, Time-optimal quantum computation, ArXiv:1210.4626.

[80] V. Kolmogorov, BLOSSOM V: A new implementation of a minimum cost perfect matching algorithm, Math. Prog. Comput. **1**, 43 (2009).

[81] A. G. Fowler, Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $O(1)$ parallel time, ArXiv:1307.1740.

[82] J. Edmonds, Paths, trees, and flowers, Can. J. Math. **17**, 449 (1965).

[83] https://github.com/kaavyas99/CSS-tCNOT-decoders

[84] 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 **622**, 279 (2023).