

Scalable Decoding Protocols for Fast Transversal Logic in the Surface Code

Mark L. Turner^{1,*}, Earl T. Campbell,^{1,2} Ophelia Crawford,¹ Neil I. Gillespie,¹ and Joan Camps¹

¹*Riverlane, Cambridge CB2 3BZ, United Kingdom*

²*School of Mathematical and Physical Sciences, University of Sheffield, Sheffield S3 7RH, United Kingdom*

 (Received 29 May 2025; revised 9 October 2025; accepted 15 December 2025; published 29 January 2026)

Atomic, molecular and optical (AMO) approaches to quantum computing are promising due to their increased connectivity, long coherence times and apparent scalability. However, they have a significantly reduced cadence of syndrome extraction compared to superconducting devices, a potentially crippling slow-down given the substantial logical gate counts required for quantum advantage. Transversal logic, which exploits higher connectivity, has the potential to significantly speed up the logical clock rate by reducing the number of syndrome extraction rounds required, but current decoders for fast transversal logic are not scalable. This is not just because existing decoders are too slow to handle the large decoding volumes resulting from fast logic; transversal logic breaks the key structural properties that make real-time decoding of lattice surgery efficient. We introduce two windowed decoding protocols for transversal logic in the surface code that restore modularity and locality to the decoding problem. Using our protocols, we show that, with a very small space overhead, our scalable decoders unlock an order of magnitude speedup for transversal logic compared to lattice surgery. Taken together, our results provide key evidence for the viability of large-scale algorithms on AMO qubits.

DOI: [10.1103/nx6p-hjqy](https://doi.org/10.1103/nx6p-hjqy)

I. INTRODUCTION

Atomic, molecular and optical (AMO) approaches to quantum computing, notably neutral atom and trapped ion devices, have seen impressive experimental progress, culminating in demonstrations of foundational primitives of logic across numerous logical qubits [1–4]. However, despite many favorable characteristics, slower physical gates in AMO systems lead to quantum error correction (QEC) cycle times in the order of ms [5,6], compared to μ s in their superconducting counterparts [7–11]. Without significant progress in accelerating AMO systems, the tractability, and therefore viability, of large-scale fault-tolerant algorithms on such hardware is at risk. While improvements are to be expected in physical gate times, they need to be complemented by QEC strategies that decisively improve the logical clock speed of the quantum computer.

Faster QEC strategies may be possible as AMO systems offer qubit shuttling and connectivity that is daunting to replicate in superconducting systems. Transversal gates,

which exploit increased connectivity to operate directly on data qubits to implement a fault-tolerant logical gate, are a promising way to accelerate these platforms. For example, Cain *et al.* [12] show that consecutive Clifford gates can be executed with $O(1)$ rounds of syndrome extraction between each gate by configuring a hypergraph decoder [13] with visibility across all logical qubits in the system. In an analogous lattice surgery context, these same circuits would be substantially more costly, because each gate layer would require $O(d)$ rounds of syndrome extraction [14].

However, decoding fast transversal logic is a fundamentally different problem to lattice surgery where the decoding problem is well understood [15]. The scaling of the decoding volume under fast transversal logic risks overwhelming the decoder and effectively halting logical computation [16], ruling out previously proposed global hypergraph decoding techniques. In part, this problem results from the inapplicability of standard windowing schemes to the spatial dimension of fast logic, pointing to a need for new windowing protocols tailored to the unique setting of fast transversal logic. Additionally, improving decoding throughput is not the only area where progress is needed: the modular structural properties of lattice surgery that respond more favorably to real-time decoding at scale need to be recovered with innovations at the decoding level in the very different context of fast transversal logic. Finally, every time a new decoding scheme is proposed, its impact on the time complexity of transversal logic needs

*Contact author: mark.turner@riverlane.com

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

to be assessed. Where previous work has focused on Clifford logic, this analysis must extend into the non-Clifford setting, and it remains the case that sustaining and scaling error correction in this context is challenging.

In this work, we address these decoding challenges in the following ways. We introduce the *ghost protocol*—a decoding protocol that guarantees small, modular decoding volumes in the context of fast transversal logic in the surface code by reformulating the spatial windowing problem as a sparse message passing algorithm. Equipped with a scalable decoder, we study its performance in the context of both deep Clifford and non-Clifford logic, focusing on the structural properties of the T gate realized via teleportation. We argue that in the context of fast, windowed transversal Cliffords, the time overhead of state teleportation is fundamentally a problem of window buffer size. From here we illuminate the surprising inherent resilience of state teleportation to small, constant size buffer regions when constituted via transversal entangling gates. We characterize how this resilience interacts with the necessity of spatial windowing and extend it with a further decoding protocol we call *patience* to improve accuracy close to what is observed with global decoding. In conclusion, our proposal makes correlated decoding a practical possibility for large-scale applications; see the discussion section for details. This delivers at least an order of magnitude speedup in the logical clock rate of Clifford+ T under standard lattice surgery assumptions [17] and can be achieved with a far smaller space overhead than would be encountered with time-optimal lattice surgery [18].

As we were finalizing our paper, related work was published by the Delft [19] group and a collaboration between Harvard, Quera and Yale [20]. We discuss how they compare in the conclusion and outlook section.

II. DECODING VOLUMES UNDER FAST LOGIC

Transversal gates are not universal [21] but nevertheless emerge as useful constructs for executing logic due to their inherent fault tolerance and low depth. A gate is *transversal* if, after a partitioning of all physical qubits into disjoint parts, the physical gates used to implement the transversal gate have a tensor product structure with respect to these parts [22]. The transversal logical Clifford gate set was recently completed in the rotated surface code with the observation that a fold-transversal S gate [23,24] could be implemented on the half-cycle state while the code's instantaneous stabilizers resemble those of the unrotated surface code [25]. The transversal H can be implemented by applying physical Hadamards on all data qubits followed by a 90° code block rotation, escaping the need for a more involved circuit [26] when qubits cannot be freely shuttled. Finally, two-qubit entanglement can be achieved via physical two-qubit gates between data qubits in each

respective part of two code blocks. The transversal CNOT has been demonstrated by performing a parallel move to interlace the data qubits of two logical qubits and executing physical CNOTs between each pair of data qubits that span code blocks [1].

In order to perform error correction, we require a decoder. *Decoders* are classical algorithms that use measurement data derived from *syndrome extraction* circuitry to identify errors that occurred during a quantum computation. Syndrome extraction involves partitioning qubits into data and ancillary resources that are repeatedly entangled, enabling the measuring of Pauli operators called *stabilizers*. Transversal gates interleave with this circuitry. Syndrome extraction enables error detection: in the simplest case the deviation of a stabilizer measurement from its previous value can be indicative of an error. Decoders typically consume measurement data in the form of *syndromes*—sets of *detector* [27] values. Detectors are parities of measurements that are deterministic under noiseless execution and get flipped from their expected values by errors. Detectors with odd parity are commonly referred to as *defects*. The number of detectors flipped by each error is referred to as the *order* of the error mechanism.

To predict which errors occurred, the decoder leverages syndrome data and a *detector error model* (DEM): a representation of the defect patterns associated with each separate error and their effects on the logical state [27]. These representations often begin as *hypergraphs* where certain single errors flip more than two detectors, though some codes and circuitry may allow for decomposition of these hypergraphs into a *graphlike* representation over which the minimum weight perfect matching problem can be solved exactly in polynomial time [28]. The Union-Find decoder [29] approximates the matching algorithm and has seen hardware implementations with throughputs that scale sublinearly in the code distance [30,31].

In practice, it is often sufficient for the decoder to determine the effect of the errors on the logical observables of interest, rather than the errors themselves. In doing so, it is useful to define the *logical sheet*, the set of errors in the decoding graph which cause the value of the logical observable to flip.

The *decoding volume* is defined as the number of detectors party to a given decoding problem. When operating at scale and in real-time, the decoding problem must be divisible into smaller parts using a *windowing protocol*, so it can be solved efficiently in response to a continuous stream of measurement data [32]. Crucially, the decoder reconfiguration overhead implicit in this operation must be manageable within reasonable time frames appropriate to the qubit modality. The impact of transversal entangling gates on the decoding volume is a by-product of measurement errors that immediately precede the round in which a transversal CNOT is performed. These errors flip three detectors and the resultant hyperedges fuse the

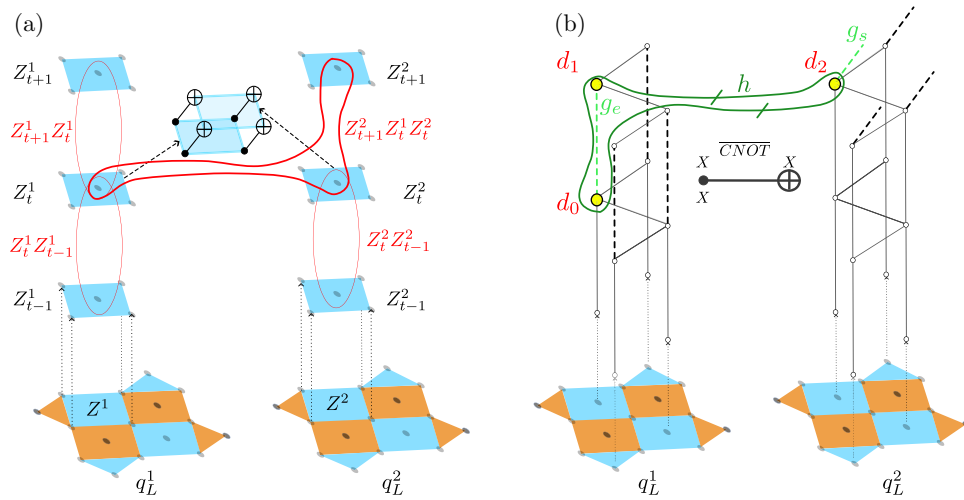


FIG. 1. Construction of detectors for transversal entangling gates. (a) Here we show Z stabilizer plaquettes from two separate logical qubits being measured at three time steps with a transversal CNOT immediately after time step t . Combinations of stabilizer measurements (labeled in black) that form detectors are circled in red. The transversal CNOT copies errors between logical qubits— X (Z) errors from control (target) to target (control)—broadening the sensitivity of certain stabilizer measurements to errors from across the system. To maintain the determinism of detector definitions and spacetime locality in error detection, the detector defined at time $t + 1$ on Z^2 is computed to be the parity of three measurements: $Z_{t+1}^2 Z_t^1 Z_t^2$ [27]. This accounts for the possibility of X errors on q_L^1 propagating through the transversal CNOT and flipping Z_{t+1}^2 . (b) A measurement error affecting the stabilizer measurement at Z_t^1 will therefore flip three detectors (labeled in red), $d_0 = Z_t^1 Z_{t-1}^1$, $d_1 = Z_{t+1}^1 Z_t^1$, $d_2 = Z_{t+1}^2 Z_t^1 Z_t^2$, yielding the hyperedge $h = \{d_0, d_1, d_2\}$ circled in green. Note, h is not trivially decomposable into a graphlike representation [12]. Decomposing h into two ghost edges (g_e and g_s visible as dashed green lines), edges that are fragments of higher-order hyperedges and unsupported by lower-order terms in the detector error model, is an effective way to break down error mechanisms of this type into a more favorable form once combined with the new correlated decoding techniques described in this paper.

components of the DEM associated with independent logical qubits into interdependent decoding problems (Fig. 1). While transversal gates preserve the locality of errors within surface code patches—and therefore fault tolerance—they also delocalize syndrome information across space, increasing the spatial decoding volume that needs to be decoded simultaneously.

Strong numerical evidence has emerged that deep Clifford logic can be sustained with a very small, constant number of syndrome extraction rounds between logical gates [12]. This result has assumed correlated hypergraph decoding with full visibility across all logical qubits, an approach that neglects the necessity of windowing. The impact of fast transversal entangling gates on the decoding volume is substantial and leads Sahay *et al.* to reject the viability of $O(1)$ rounds of syndrome extraction between transversal gates in cases that involve deep and wide logical computation [16]. Since windowed decoders require $O(d)$ rounds of syndrome information to decode errors of weight $\lfloor d/2 \rfloor$, they observe that transversal entangling gates separated by $O(1) \ll d$ rounds of syndrome extraction yields a decoding volume scaling as $2^{O(d)}$. This upper bound is derived from a worst-case “binary-tree” circuit that spreads an error across logical qubits exponentially in the logical circuit’s depth. They argue that this exponential decoding volume could overwhelm

the decoder, risking the backlog problem [33] where the computation grinds to a halt. We note that large, albeit not exponential, spatial volumes can also be accumulated in lattice surgery via multiqubit measurements between spatially estranged patches. However, unlike in lattice surgery, fast transversal logic provides no guarantee of locality in interactions between qubits and undermines the notion of a contained buffer region, meaning that existing windowing schemes [15,32,34,35] used to break down decoding problems in space do not apply [Figs. 2(a) and 2(b)].

To solve this, Sahay *et al.* advocate enforcing d rounds of syndrome extraction between transversal entangling gates so that each logical operation can be decoded in independent windows of duration $O(d)$ —a time cost that slow AMO platforms are unlikely to be able to afford in the large-scale fault-tolerant regime. Alternatively, one could stick to the global correlated decoding strategies of Cain *et al.* and argue that binary-tree circuits do not characterize the kind of logical subroutines that might be employed in a realistic quantum algorithm. Indeed, gadgets such as Toffoli ladders, prominent in quantum arithmetic [36], do not grow the decoding volume exponentially in the duration of the window; instead, the decoding volume would grow linearly across qubits in the temporal depth of the window given $O(1)$ transversal logic.

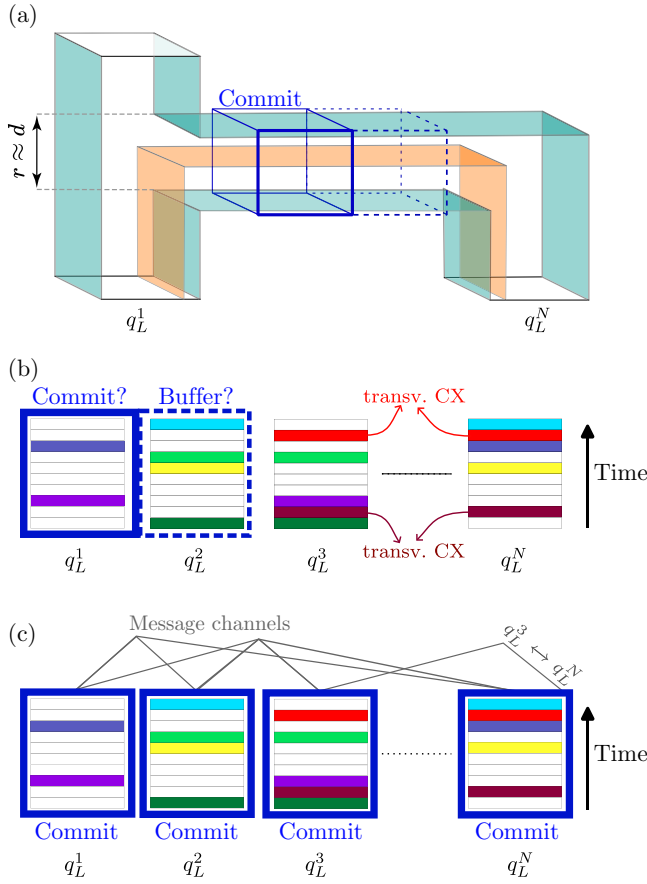


FIG. 2. Decoding volumes and the spatial windowing problem. Windowing schemes typically work by complementing a commit region with a locally adjacent buffer region [32]. The assumption is, therefore, that there is a coherent notion of spatially local syndrome information that enables successful partial decoding. Since lattice surgery conforms to two-dimensional nearest neighbor connectivity, this assumption holds and commit-buffer windowing techniques apply naturally [15]. (a) Consequently, we can slide a decoding window (solid and dashed blue boxes represent the commit and buffer regions, respectively) across a merge patch of arbitrary size, thereby dividing the merge patch into smaller subproblems. Conversely, fast transversal logic denies us locality in the detector error model. (b) and (c) Here we represent N logical qubits with time progressing vertically; each nested rectangle represents a round of syndrome extraction and two logical qubits sharing a rectangle of the same color denotes that they have undergone a transversal entangling gate during this round (for example, in red we see a transversal CNOT between q_L^3 and q_L^N in the penultimate round of the window). $O(1)$ logic is in tension with the fact decoders need $O(d)$ rounds of syndrome information to infer error patterns reliably, because it leads commit-buffer decoding to fail: as shown in (b), the solution to q_L^1 either directly or indirectly depends on all other qubits in the system and a buffer region around q_L^2 (dashed) provides insufficient additional information to reliably commit to a solution on q_L^1 . (c) We propose that logical qubits can be decoded independently if spatial windowing is rephrased as a sparse message passing problem (see Sec. III).

While such a decoding problem where decoding volumes do not scale exponentially might be solvable with the kHz throughput necessary for AMO (from a QEC cycle duration of around 1 ms), it is less obvious that this would work in a streaming context since DEMs would have limited commonality between windows, requiring the decoding problem to be reformulated and the decoder to be completely reconfigured between windows. These are additional costs independent of the decoder's throughput and could be substantial. The reason for this is that fast transversal logic forgoes the modular structure [15] that characterizes decoding problems based on lattice surgery—and it is this modular structure that minimizes the reconfiguration overhead that lattice surgery computation necessitates. It follows that we need to rethink how, in the context of fast transversal logic, we maintain a constant and manageable decoding volume while minimizing reconfiguration between windows. This means minimizing the extent to which the decoding problem changes in relation to the underlying logical circuit being decoded. With this in place, we will be better equipped to present an analysis of the time complexities of transversal gates.

III. ITERATIVE GHOST-EDGE DECODING

Our forthcoming analysis of Clifford and non-Clifford logic is founded on the observation that the decoding volume under fast transversal logic can be managed via decomposition of hyperedges that span logical qubits. Formally, these hyperedges are undecomposable since they are unsupported by existing lower-order errors in the DEM [12] (i.e., errors that flip fewer detectors than the original hyperedge and can be summed together modulo two to form the original hyperedge). Nevertheless, decomposition becomes possible if we allow ourselves to introduce new lower order *ghost edges* [dashed edges in Fig. 1(b)] and leverage them as correlated fragments of higher order effects.

The goal of our decomposition is to enable independent decoding of logical qubits. To achieve this, the ghost edges we introduce must only flip detectors associated with one logical qubit. The hyperedges that emerge from transversal entangling gates have order three and span logical qubits so we decompose them into (i) a time/hooklike ghost edge in one patch (edges joining pairs of detectors that are defined sequentially in time on two stabilizer measurements) and (ii) a ghost singleton in the other patch (an edge to the boundary that flips just one detector). These are represented by the dashed green edges labeled g_e and g_s , respectively, in Fig. 1(b). In related work, the order-three hyperedges that emerge under the transversal S gate are decomposed in a similar way [25], though the hyperedges are intrapatch error mechanisms that span stabilizer types, not, as in our case, interpatch errors that affect

the decoding volume. Our approach maintains a spatial separation between logical qubits at the level of the decomposition of the hypergraph, not through detector definitions that break the temporal locality of defect information with respect to each independent error mechanism (see Ref. [37] for a decoder that assumes detector definitions that do not span logical qubits, leading to errors that flip detectors arbitrarily long after they occur).

The *ghost protocol* is an iterative decoding procedure where each logical qubit is decoded to correction edges (including ghost edges) independently several times, each time exchanging messages with decoders that are responsible for other logical qubits in the system. These messages handle correlations between the ghost edges g_e and g_s (and, by extension, logical qubits) by invoking syndrome refinement operations where specific detectors are flipped in response to decoding to ghost edges g_e . Concretely, since we know ahead of time that g_e is necessarily correlated with g_s , any decoders that decode to g_e will:

- (1) Refine their syndrome such that the detectors at either end of g_e are flipped
- (2) Message the decoder responsible for g_s with the instruction to flip the single detector represented by this ghost edge.

We track the effect of the changing syndrome on the logical observables that the decoder is trying to protect. Taken together, these syndrome refinement operations have the effect of committing to order-three hyperedges and eliminating them from the decoding problem before subsequent iterations in the protocol are performed (see Fig. 4 for a worked example). Initial decoding passes therefore function like a predecoder that purges order-three hyperedges from the system and the derivation of a final logical effect is postponed until the predecoding routine has completed. The correlations between ghost edges are highly structured, predictable and learnable ahead of runtime.

The inner decoder used in our protocol is two pass correlated matching [38], an implementation based on Ref. [39]. That is, we employ two levels of message passing, each of which addresses thematically distinct contributions to the error hypergraph—two passes for Y errors that correlate X and Z errors, ghost decoding for interpatch hyperedges that correlate instances of g_e and g_s . These two forms of correlated decoding are closely related: in both cases correlations between correction edges are used to emulate hypergraph decoding but use simpler graph-based primitives. However, correlated decoding for Y errors does not require ghost edges since Y errors in the surface code are trivially decomposable into lower-order X and Z terms. Further, Y errors are handled with two decoding passes, whereas the ghost protocol often needs more iterations (Appendix B): refining the syndrome can unlock the discovery of new ghost edges indicative of further

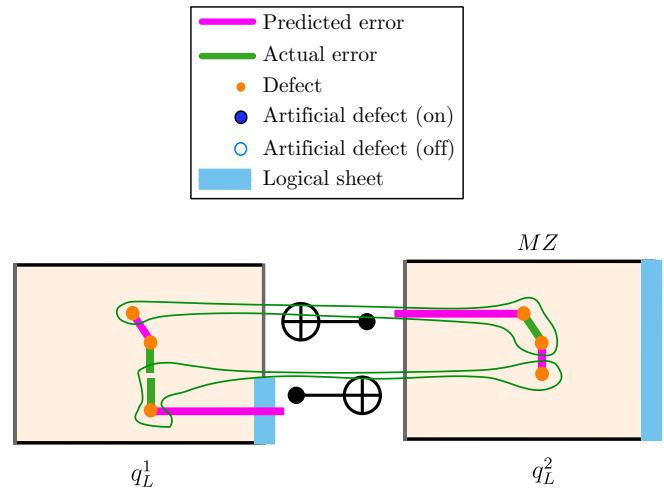


FIG. 3. A failure mode that necessitates use of g_s during iteration. When CNOTs are in opposite directions, order-three hyperedges (circled in green) form in opposite directions. This can make it hard to successfully discover g_e during iterative decoding and use this information to flip the three detectors that comprise each hyperedge. This is because logical qubits are decoded independently, so the correlation between g_s and its associated g_e can be misassociated with the g_e of another hyperedge. This can be resolved by carefully exposing all instances of g_s during some of the iterations (see Fig. 4). Note: exposing g_s during every iteration is distance destroying.

order-three hyperedges that were undiscoverable through graph-based decoding of the initial syndrome.

In order for the ghost protocol to be successful, we must handle g_s with care. These edges to the boundary are inevitable by-products of our decomposition, but destroy the effective distance if exposed to the decoder continuously. This is because, being edges to the boundary across time slices of the DEM with transversal entangling gates, g_s can break the topology of the system. To avoid such distance reduction, one might be tempted to completely hide g_s from the decoder. However, during initial decoding iterations these singletons can have an important role to play: consecutive transversal entangling gates in opposite directions separated by $O(1)$ rounds of syndrome extraction have nearby g_e and g_s that can conflict with one another, obscuring the discovery of g_e , crucial for successful syndrome refinement (Fig. 3). By retaining all g_s during a carefully selected subset of decoding passes, we can break these conflicts and reliably correlate errors across logical qubits (see Fig. 4 for a worked example). This works because the passes in which we expose g_s are used to clean up the syndrome, not derive an overall logical effect where the shortest undetectable logical operator in the DEM is crucial. This addresses a weakness of existing graph-based decoders where an inability to decode even modestly deep circuits composed of fast transversal Clifford logic has been reported [40].

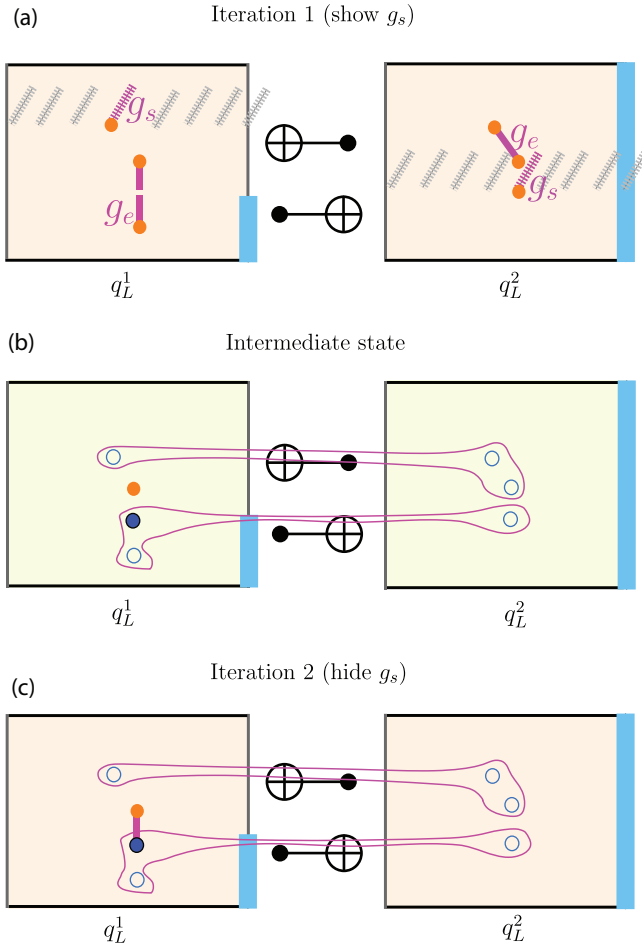


FIG. 4. Leveraging g_s during iterative ghost-edge decoding to resolve the failure mode shown in Fig. 3. (a) Decoder iteration 1. Dashed gray lines represent instances of g_s , edges to the boundary which we expose to the decoder during this iteration. On q_L^1 these edges to the boundary function to break g_s away from conflicting with the string of measurement errors below enabling the discovery of g_e . Similarly, on q_L^2 the introduction of edges to the boundary also means we decode to a separate hooklike instance of g_e . Recall that without the incorporation of g_s into the decoding problem, g_e would not have been detectable (see Fig. 3), causing a logical error. (b) Having found two instances of g_e we flip all three detectors associated with two separate order-three hyperedges. On q_L^1 this introduces a new artificial defect highlighted in blue, while the remaining five detectors are turned off (hollow blue circles). (c) Decoder iteration 2. We decode again, but from now the edges to the boundary denoted by g_s are hidden from the decoder to protect the distance of the code. Decoding proceeds to a final logical effect, a correction that is equivalent to the actual error shown in green in Fig. 3.

Appendix B describes the exact number of passes and passes with singletons used to configure the decoder for the data in this work; we treat these parameters empirically and leave optimizing them to future work. The number of iterations required is fewer than d for all our simulations. Since AMO qubits will require a decoding response

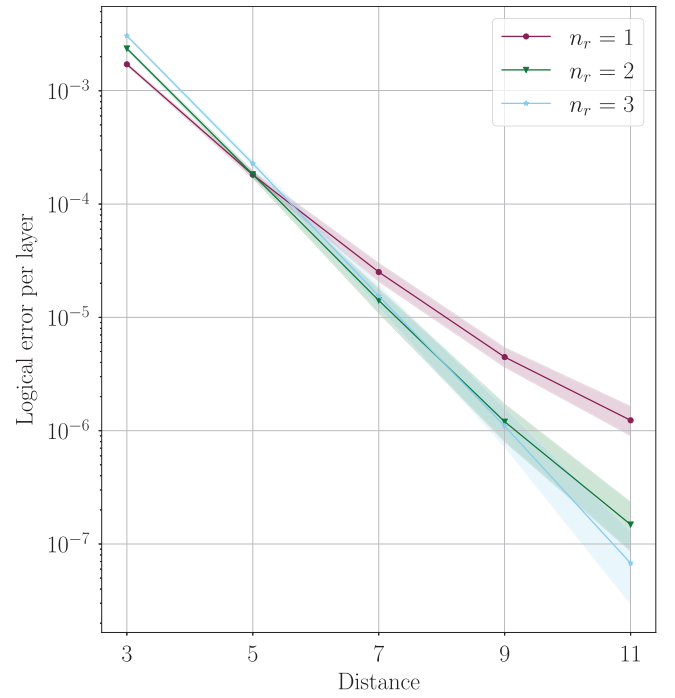


FIG. 5. Deep logical Clifford circuits. We show the subthreshold scaling of ghost decoding of deep Clifford logic between four rotated surface codes under circuit-level noise ($p = 0.1\%$). Following [12], each logical qubit is prepared in $|+\rangle$ and subject to 32 layers of transversal gates, followed by perfect measurements of the logical stabilizers of the state. A gate layer is composed of n_r rounds of syndrome extraction, preceded by a randomly chosen single-qubit transversal gate $\{H, X, Y, Z\}$ on each qubit then two transversal CNOTs that entangle random pairs of qubits. Increasing n_r can steepen our protocol's gradient of error suppression in the distance, but the spacetime volume computed as $(n_r + 1)d^2$ is minimized at $n_r = 1$ for per layer error rates of 10^{-6} . See Appendix B for the parameters that determine the decoding passes used in these numerics and Appendix A for details of the noise model.

time of ~ 1 ms, such a problem appears comfortably within reach given our protocol generates decoding windows that are compatible with existing field programmable gate array architectures for surface code decoding [30,31], where per round decoding times far below $1 \mu\text{s}$ have been observed with sublinear scaling in the distance.

Figure 5 shows the subthreshold scaling of ghost decoding of deep Clifford logic between four rotated surface codes under circuit-level noise ($p = 0.1\%$)—a benchmark first proposed in [12]. In our numerics, the best accuracy is seen with $n_r = 3$. However, in the regimes we study, the spacetime volume of the computation computed as $(n_r + 1)d^2$ is minimized at $n_r = 1$ for target per layer error rates of 10^{-6} , results that are consistent with the previously reported performance of a global correlated decoder [12]. Our results are the first to show deep transversal Clifford logic where the spatial component of the decoding volume is kept constant while decoding a larger logical circuit.

By decomposing order-three hyperedges such that they do not span logical qubits, spatial windowing emerges naturally from the act of sparse global message passing [Fig. 2(c)] and the decoding volume—both in size and content—reflects a windowed single-qubit memory experiment on each decoder core regardless of the number of rounds of syndrome extraction between transversal gates. That is, the need to manage the decoding volume motivates making the problem graphlike. Further, data locality, crucial in distributed computing, is ensured despite high qubit connectivity, with each decoder taking ownership of decoding a single logical qubit. Above all, in a streaming setting our architecture minimizes the amount of reconfiguration work necessary on each decoder core between windows: from the decoder’s perspective the decoding problem is simply memorylike and logic is handled via syndrome refinement gadgets that can be abstracted into a representation that is removed from the exact context in which a gadget is deployed. Consequently, the window configuration is divorced from the specific details of the logical circuitry. There is a structural predictability inherent in windows of this form that resembles the favorable modularity of real-time error correction in a lattice surgery context [15].

It remains to study the relationship between the ghost protocol and the time complexities of *non*-Clifford logic.

IV. FAST TELEPORTATION IS A BUFFERING PROBLEM

Transversal gates are not universal [21], centralizing the importance of gate teleportation for universal computation in the surface code. Here we focus on the T gate which is performed by consuming $|T\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle)$ states by teleportation. With reference to Fig. 6, two quantities allow us to quantify the time overhead of a T gate realized with transversal gates:

- (1) n_{buf} captures the number of rounds of syndrome extraction between a transversal CNOT that creates the entanglement necessary for teleportation and the logical measurement that informs a classically controlled S gate.
- (2) n_{sep} defines the number of rounds that separate consecutive T gates.

Notably, n_{buf} functions as a buffer region with respect to the decoding problem centered on the logical measurements in the circuit, presenting new structural challenges for decoding. This is because the logical measurement can be flipped by errors before the transversal CNOT on both qubits, but only on the measured qubit after the transversal CNOT. The result is a decoding problem where an open temporal boundary (“severance”) has been imposed across the surviving logical qubit, a disruptive severance that is

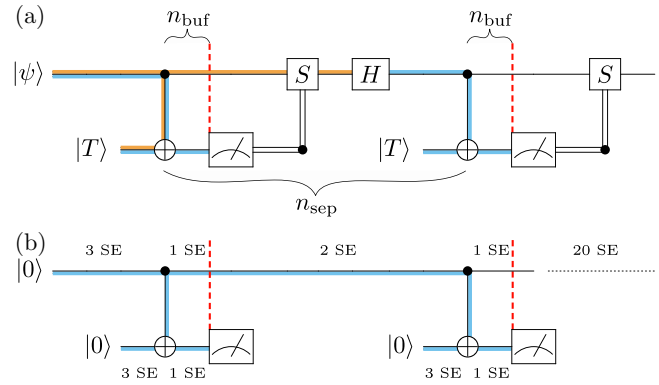


FIG. 6. T gates and measures of time overhead. (a) Two consecutive T gates, implemented via teleportation circuits, separated by a Hadamard. The X (Z) logical sheet is shown in orange (blue), capturing where a logical observable is sensitive to physical Z (X) errors. Logical sheets that do not affect a measurement are ignored for clarity. Logical measurements are made in the Z basis, and each informs a classically controlled S gate. During each T gate, a logical Z basis measurement is performed that only terminates one of the qubits—the qubit carrying the state we wish to teleport. Consequently, measuring a subset of qubits in the system imposes an artificial *severance* (dashed red line) across the decoding problem associated with the top logical qubit. A severance manifests as an open temporal boundary across the error model prior and can disrupt decoding by introducing new locations for error strings to terminate without guarantees of topological protection. With n_{buf} we quantify how many rounds of syndrome extraction separate this severance from the logical sheet (shown in blue) that needs to be decoded to close the T gate. Separately, the quantity n_{sep} captures how many rounds must separate transversal CNOTs in the context of contiguous T gates while preserving robust and scalable decoding. (b) A proxy of the T gate circuit that is simulable in the Clifford simulator, Stim [27]. All qubits are prepared in $|0\rangle$ with n SE describing how many rounds of syndrome extraction occur on each qubit at each point in the circuit. This circuit gives us an experiment to quantify n_{buf} and n_{sep} —the quantities that determine the time complexity of T gates—in the context of severances and complete windowing of fast transversal logic.

only n_{buf} rounds from the logical sheet we are trying to protect.

Buffer regions are typically required to be of size $O(d)$ rounds of syndrome extraction [15,34], suggesting n_{buf} should scale with the code distance irrespective of the time complexity of the transversal CNOT when considered in isolation. Consequently, it would seem that in breaking free of lattice surgery, a paradigm that necessitates measuring stabilizers for $O(d)$ rounds, we are quickly forced back into such a regime by the buffering conditions of windowed error correction itself. To emphasize the point, there is a striking tension here between the practicalities of error correction and the theoretical logical clock speed. Therefore, the key question is: how small can we make n_{buf} while still decoding effectively?

To study the behavior of the decoding protocol as a function of n_{buf} and n_{sep} , we simulate a proxy [Fig. 6(b)] of the repeated teleportation circuit shown in Fig. 6(a). Our goal here is to use a logical circuit that captures the key structural properties of teleportation circuitry while simultaneously being efficiently simulable within the Clifford simulator, Stim [27]. To achieve this we strip out classically controlled logical gates since they are non-Clifford. Further, we modify the circuit such that our observables are deterministic. As such, we prepare all qubits in $|\bar{0}\rangle$ and remove the H gate between the T gates in Fig. 6(a). In the ideal circuit H enables consecutive T gates to evolve the state toward an arbitrary state, however, in our context it would make the second logical Z measurement non-deterministic and resistant to simulation in Stim. Instead, we leave two rounds of syndrome extraction between the measurement of a logical qubit and the subsequent transversal CNOT in the expectation that a classically controlled S gate would be applied midcircuit during the first round [25] and the transversal H would be applied at the end of that same round of syndrome extraction. This then leaves a further empty round of syndrome extraction before the next CNOT which begins another T gate. Ultimately, this gives us a fast T gate proxy where $n_{\text{sep}} = 3$ and a benchmark that focuses on the open temporal boundaries we call “severances” (dashed red lines in Fig. 6) that challenge the decoder when logical gates are executed in $O(1)$ time.

Our proxy does not address the time complexities of state preparation around which there is an extensive literature [32,41–48]. Instead, our focus is the time overhead of teleportation in the bulk of the computation. In Fig. 7 we show the surprising result that $n_{\text{buf}} = 1$ and $n_{\text{sep}} = 3$ yield error suppression even with full temporal and spatial windowing. This suppression scales with $\lceil d/2 \rceil$ at small distances, and $\lceil d/4 \rceil$ at higher distances. In Fig. 7(a), this can be seen by the fact that the separation between lines of consecutive distances ($d, d+2$) is roughly double at small d compared to large d . In Fig. 7(b) the transition from $d/2$ to $d/4$ suppression features in the change of slope of the windowed curve (GP), that follows the global curve at small distances, and the windowed error (TW) at large distances, that has roughly half the slope. This is also the point where errors stemming from temporal windowing dominate. Temporal windowing (“TW”) errors are shown as the dotted line in Fig. 7(b), capturing the probability of the temporally windowed ghost protocol disagreeing with the global ghost protocol (note, this metric is equivalent to first step heralded errors in Ref. [49]). In the next section, we illuminate why temporal windowing errors come to dominate and characterize transition points in the observed gradients of error suppression. We relate this to the unique structural properties of the post-transversal-entangling-gate setting. This analysis culminates in a new decoding strategy—*patient decoding*—that recovers the

accuracy lost at higher distances under aggressive temporal windowing with negligible time overhead.

V. ERROR SUPPRESSION DEPENDENCE ON CIRCUIT AND BUFFER SIZE

In this section we will elucidate the error patterns that determine the characteristics of the error suppression observed in Fig. 7(b) and explain why it persists despite our aggressive temporal windowing of the decoding problem.

First, however, we reflect on the behavior of error suppression in a quantum memory context under various windowing protocols—a frame of reference for the problem to hand. A typical temporal window contains a “commit” and “buffer” region. The decoder addresses the whole window, but it only decides firmly on outcomes inside the commit regions. The buffer region expands the view of the decoder beyond the commit region and generally includes around d rounds of syndrome information. This enables reliable correction of all errors up to weight $\lfloor d/2 \rfloor$ [Fig. 8(a)] and the expected logical error suppression behavior, $P_L \propto \Lambda^{-\lceil d/2 \rceil}$. Importantly, if we utilize the sliding window protocol described in Ref. [32], error suppression in the distance persists even if no buffer region is used. In this context, the minimum weight of uncorrectable errors is diminished from $\approx d/2$ to $\approx d/4$ [Fig. 8(b)] halving the gradient of error suppression in the distance. Error suppression is nevertheless possible here, because the sliding window protocol leverages artificial defects to postpone decisions around the logical effect of “ambiguous” error patterns into the future—an effect that functions to imitate something of a buffer region even where, formally, no such buffer region exists.

Conversely, a sliding window protocol that does not utilize artificial defects will, in the absence of a buffer region, fail to suppress errors in the distance [Fig. 8(c)]. In other words, in the memory setting, it is the interdependence between windows shown in Fig. 8(b) that sustains moderate error suppression without buffer regions. As soon as this interdependence is removed, as in Fig. 8(c), buffer regions of size $O(d)$ become indispensable.

Fast T gate teleportation must survive in a paradigm that is analogous to that shown in Fig. 8(c): we need buffer regions to be small and cannot allow ourselves to benefit from the use of artificial defects that bind a firm decoding decision to a future decoding problem since a final logical effect is needed as soon as possible to apply a Clifford correction. To do otherwise would return the time complexity of the T gate to $O(d)$. Therefore, against the backdrop of windowed quantum memory where independence between decoding windows necessitates substantial buffering overheads, a surprising feature of our data in Fig. 7(b) is that strong error suppression nevertheless persists despite $n_{\text{buf}} = 1$. So what is different here and how is

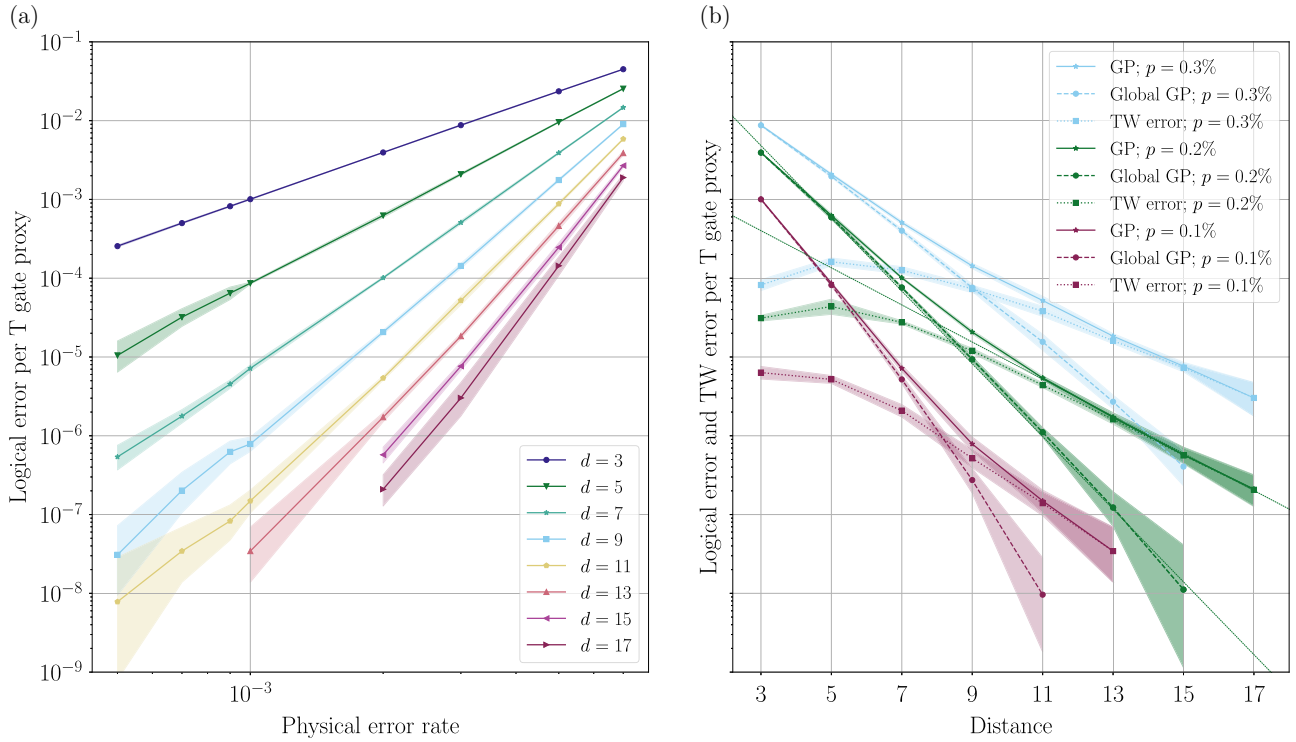


FIG. 7. Ghost decoding of T gate proxies ($n_{\text{buf}} = 1$ and $n_{\text{sep}} = 3$). (a) Logical error per T gate proxy of fully windowed (i.e., both spatially and temporally) ghost decoding as a function of the physical error rate p . Straight lines indicate power laws between the logical error and p , with the power increasing with the distance, although not quite linearly for our range of distances—note, the error mechanisms responsible for this are described in Sec. V and further decoding protocols that overcome them are proposed in Sec. VI. (b) Gradients of error suppression in the distance for fully windowed ghost decoding (solid lines labeled “GP”), ghost decoding without temporal windowing (dashed lines labeled “Global GP”, though note spatial windowing is still operational) and the temporal windowing error (dotted lines labeled “TW error”). TW error describes the probability that Global GP disagrees with the decoding result of GP, capturing the contribution of errors originating from our aggressive temporal windowing. We note a p -dependent transition point (at roughly $d = 11$ for our range of p) where the gradient of the fully windowed decoder converges on the TW error rate with a gradient of error suppression that is roughly half that of the global decoder (fits for the TW error and the global GP error at $p = 0.2\%$ are plotted as green continuous lines and have slopes of 0.24 and 0.45 respectively). This highlights an asymptotic regime where the temporally windowed decoder has failures of half the weight compared to the global decoder ($d/4$ vs $d/2$), emerging from the relationship between n_{buf} and our spatial windowing protocol (again, this distance-halving effect is reversed in Sec. VI with additional decoding).

this possible given the $O(1)$ time complexity of our T gate proxy?

Here we have to return to what is special about the DEMs that underpin transversal entangling gates [recall Fig. 1(b)]. The core difference in comparison to the memory setting is that transversal entangling gates create *more* defect information than standard errors in the surface code. In particular, error strings that begin before a transversal entangling gate and terminate after it deterministically spread defect information onto other qubits due to the subsumption of timelike errors into order-three hyperedges that link the otherwise detached error models of two logical qubits.

Figure 9 gives a topological perspective on why this increased syndrome information sustains error correction even when this same syndrome information is incomplete in proximity to a commit region. We show how the copying of errors through transversal gates opens up a wealth of

logically trivial loops in the hypergraph that the ghost protocol is able to exploit when decoding to logical effects. Notably, failure to discover a ghost edge and trigger syndrome refinement [Figs. 9(a) and 9(c)]—or conversely, erroneous discovery of a ghost edge [Fig. 9(b)]—is *not* inevitably a failure mode. This underlies the viability of fast T gates with aggressive windowing.

These unusual topological structures create redundancy and enable the error suppression in the distance shown numerically in Fig. 7. However, there are limits to this resilience that fall short of the ordinary level of protection we expect from our code distance. We find that the ghost protocol is subject to logical errors of weight (see Fig. 10),

$$w \geq \frac{n_{\text{buf}} + 1 + \lceil d/2 \rceil}{2}. \quad (1)$$

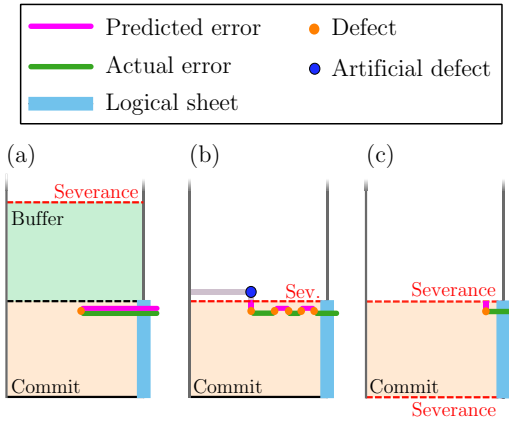


FIG. 8. Basics of buffer size for quantum memories. (a) The maximum weight ($w = \lfloor d/2 \rfloor$) adversarial error string that is correctable in a sliding window setting [32], given a buffer region of size $\approx d$. (b) Given a buffer region of size 0, the maximum weight correctable adversarial error shrinks to $w = \lfloor d/4 \rfloor$. Here we show an error pattern described in Ref. [15] that can occur when this is exceeded. Note, error suppression is still possible in absence of a buffer region, because the effect of a buffer region is imitated by the act of pushing artificial defects into the future. (c) Conversely, if forgoing the use of artificial defects but retaining a buffer region of size 0, the decoder would be susceptible to small, constant weight logical errors. That is, increased independence between decoding windows is incompatible with minimal buffer region sizing.

We have verified by search that the smallest error patterns leading to failure in our decoder saturate this inequality.

To derive Eq. (1), look at Fig. 10(b). In the figure, an error string (green) of length w starts at an offset c_h from a boundary in the control logical qubit q_L^1 and travels across and up through a transversal CNOT. The ghost decoder introduces a logical error when the concatenation of the error (w) and correction (c_h) strings extends past the midpoint of q_L^2 , $w + c_h \geq \lceil d/2 \rceil$, because, unlike in Fig. 10(a), the defects in q_L^1 and q_L^2 are matched to opposite boundaries in either logical qubit. We can rewrite this condition as $w + c \geq n_{\text{buf}} + 1 + \lceil d/2 \rceil$ using that the total correction weight is $c = c_h + n_{\text{buf}} + 1$ (+1 for the edge to the open temporal boundary). Now, the ghost decoder will pick this purple correction in q_L^1 when $w \geq c$. Adding the two inequalities, c cancels and we deduce Eq. (1). Crucially, logical errors are suppressed in the distance despite $n_{\text{buf}} = 1$, a significant difference to the aforementioned windowing of memory without artificial defects. However, there is a halving of the effective distance that is implied by this inequality. This halving is slow to affect the gradients of error suppression we showed numerically in Fig. 7(b), because $n_{\text{buf}} + 1$ acts as a constant in Eq. (1) that inflates w , creating the illusion of error suppression scaling with $\lceil d/2 \rceil$ at low d . At intermediate d the impact

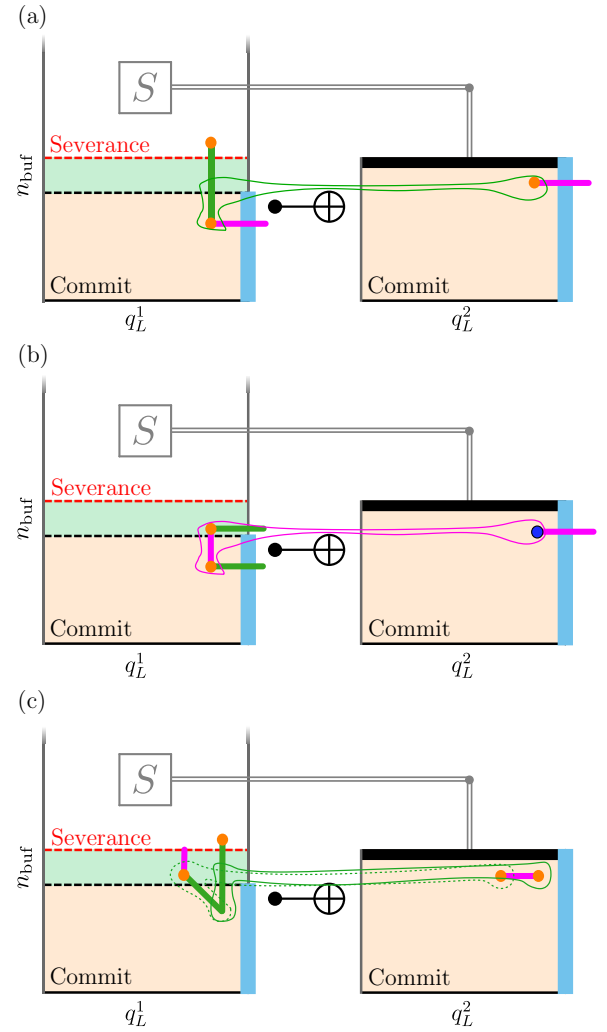


FIG. 9. Resilience of the ghost protocol to T gates with $n_{\text{buf}} = 1$. In this figure we show syndromes for errors (highlighted in green) and their correction (purple) under iterative ghost decoding. We assume a simple configuration with two iterations of the ghost protocol. (a) An error string in q_L^1 (green) pierces through a transversal CNOT, lighting a detector in q_L^2 (orange dot, explained by the green hyperedge across the two logical qubits), and terminates after a severance in q_L^1 (top orange dot in q_L^1). Because of the severance in q_L^1 , the decoder never infers that an order-three hyperedge occurs, but nevertheless arrives at a logically equivalent correction (purple). This correction crosses the logical sheet that tracks the observable measured in q_L^2 (light blue, on both qubits before the CNOT, and only on q_L^2 after) separately on both q_L^1 and q_L^2 . (b) Similarly, two spatially oriented error strings (green) are decoded as an order-three hyperedge (purple), leading to an apparently wrong logical effect in q_L^1 , but also introducing a new artificial defect in q_L^2 (blue dot). The correction in q_L^2 (purple) restores the correct logical effect. (c) More than one order-three hyperedge (circled with dashed and solid lines) can occur and be undiscovered without jeopardizing the logical information.

of this constant is diminished and error suppression scales with $\lceil d/4 \rceil$, evidencing the halving of the effective distance anticipated in Eq. (1).

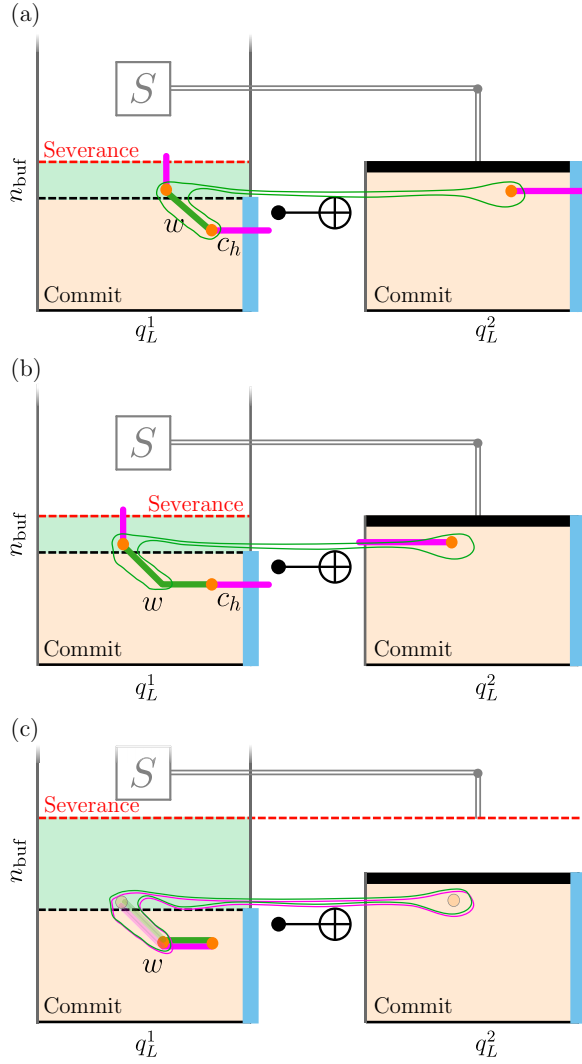


FIG. 10. Limits of the ghost protocol's resilience given T gates with $n_{\text{buf}} = 1$. (a) Here we show an error of weight w and a windowed correction of weight $c = c_h + n_{\text{buf}} + 1$ in qubit q_L^1 . A logical error is avoided but only because w is small. (b) A larger horizontal component of w turns the error of (a) into a failure mode. Here a logical error is introduced because the actual error does not flip the logical, but unlike before, the effect of c_h crossing the logical in q_L^1 is no longer neutralized in q_L^2 due to the increase in w . (c) The naive fix to the aforementioned failure mode is to enforce $n_{\text{buf}} \approx d/2$. Section VI improves upon this.

Notably, this failure mode is a feature of our windowing protocol where we consider logical qubits separately. Decoders that forgo spatial windowing [12,16] achieve the expected code distance as they do not suffer from this failure mode: given the error in Fig. 10(b), decoders with visibility over both q_L^1 and q_L^2 can use the defect information on q_L^2 to infer the actual error string on q_L^1 . Consequently, in a regime where decoders are not spatially windowed, the resilience property we outline here survives

without limitations even with $n_{\text{buf}} = O(1)$ —a key difference to what is possible in normal temporal windowing settings.

Instead, to guarantee ordinary protection against all errors up to weight $\lfloor d/2 \rfloor$ it follows from Eq. (1) that ghost decoding needs a buffer size of $n_{\text{buf}} \approx d/2$ [Fig. 10(c)]—a big cost for transforming the problem into the windowed form advocated for in Sec. III. Without increasing n_{buf} in this way, ghost decoding would still enable $O(1)$ logic, but with a fourfold space overhead in the number of qubits (from the $d \rightarrow 2d$ increase needed to counter the $d/2 \rightarrow d/4$ failure mode, leading to a $d^2 \rightarrow 4d^2$ increased space cost). In the following section we overcome this with a new decoding strategy that allows $O(1)$ transversal logic and windowed decoding to coexist with a significantly reduced accuracy penalty.

VI. ALL YOU NEED IS PATIENCE

Here we describe a new decoding technique—*patience*—that builds on the ghost protocol and seeks to overcome the aforementioned distance reducing failure modes with minimal impact on n_{buf} . We begin by highlighting two important properties fundamental to patience:

- (1) The freedom to postpone the application of conditional logic during teleportation. That is, we are free to accumulate more syndrome information than originally planned for via further syndrome extraction rounds before applying a conditional gate.
- (2) The inherent rarity of the distance reducing failure modes described in Sec. V. This rarity is evident in the low TW error probability observed in Fig. 7(b).

Property (1) means n_{buf} can be a runtime parameter without changing the underlying logical operation (in the absence of error, a T gate remains a T gate regardless of how many rounds separate a CNOT from the classically controlled S gate) and property (2) is suggestive of the fact that increases to n_{buf} are only necessary in a heavily suppressed number of cases.

Patience is a protocol that aims to exploit these two properties in order to escape the failure modes that afflict the ghost protocol without necessitating a significant increase in *average* n_{buf} . Specifically, patience extends the ghost protocol with the ability to herald distance reducing failure mechanisms resulting from aggressive temporal windowing. In the minority of cases where the decoder raises a herald, the protocol postpones the application of a conditional gate until n_{buf} has increased through continued syndrome extraction rounds. This means increasing n_{buf} such that $w = \lceil d/2 \rceil$ where w is defined in Eq. (1). Once further syndrome data has been generated and passed to the decoder, the decoder has a final attempt at determining a correction with more syndrome information.

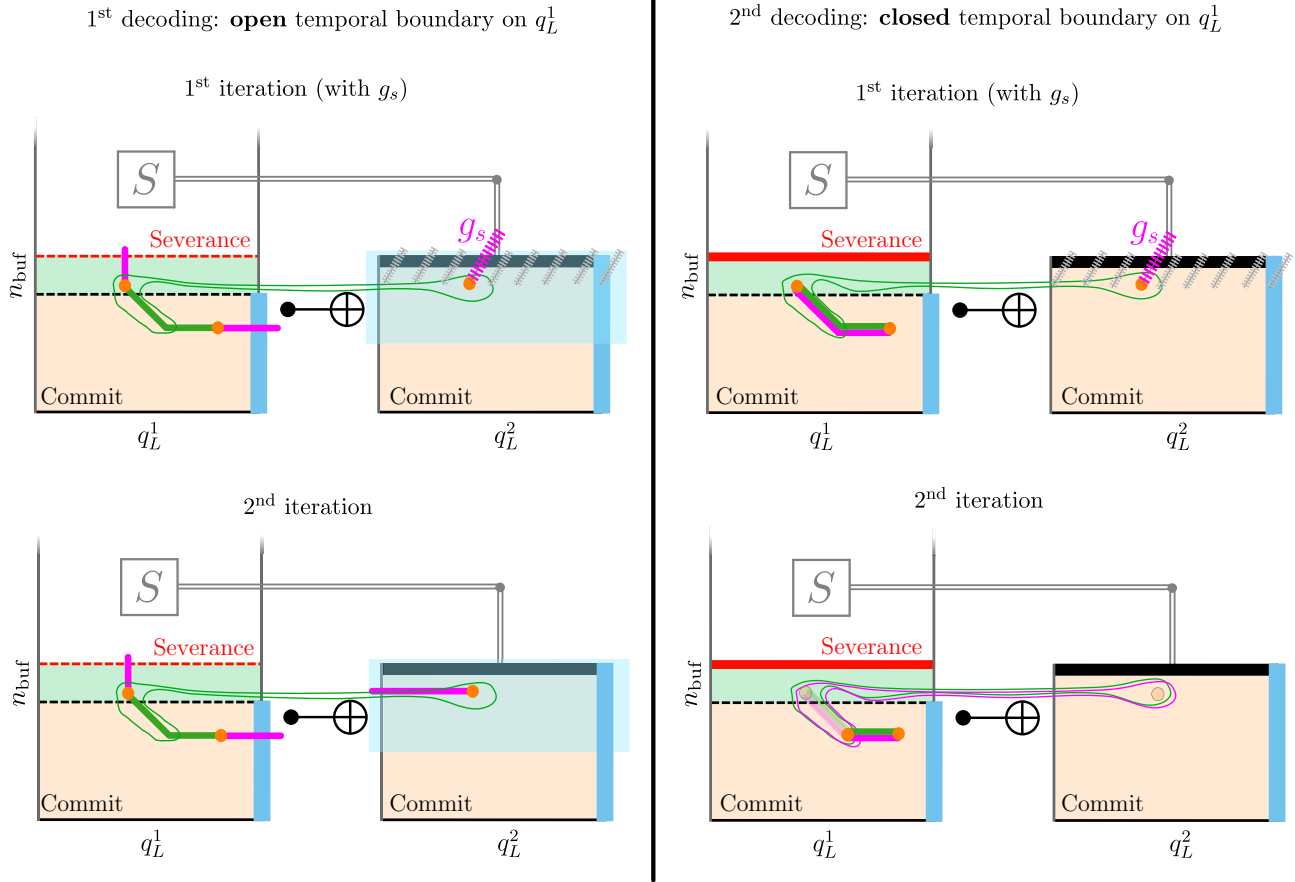


FIG. 11. Heralding failure modes with the patience protocol. Here we show how the failure mode first highlighted in Fig. 10(b) can be heralded. Heralding allows us to postpone the application of the Clifford correction, allowing more syndrome information to be collected before redecoding the logical measurement that conditions the Clifford correction with improved fidelity. (Left) We perform a decoding with the ghost protocol using an open temporal boundary (red dashed line) on q_L^1 and compare it to a further decoding outcome (right) where we decode with a closed temporal boundary (solid red line). Correction strings are free to terminate on open temporal boundaries, but not on closed temporal boundaries. Notice that after the second iteration, these two decodings have arrived at different logical predictions: this difference is a sufficient trigger of patience. Separately, while iterating during the first decoding (left) we track whether or not the weight of the correction in the blue shaded region increases between the first and final iteration of the ghost protocol. An increase, as shown here, is sufficient to herald the need to be patient.

Operationally, we trigger patience through two inference steps either of which is sufficient to immediately trigger patience if satisfied (Fig. 11 gives visual intuition behind these steps):

- (1) We run the ghost protocol as usual, but track whether the weight of the correction string increases between the first and final decoding iterations performed by the ghost protocol. This functions to detect most cases where aggressive temporal windowing spuriously triggers a syndrome refinement operation (of the form described in Sec. III) despite the fact no such order-three hyperedge occurred. In the language of Fig. 1(b), these cases are often detectable because they mistakenly involve a flip of g_s , triggering a defect that, in future decoding

passes, incurs a long set of correction edges due to the stringlike nature of errors in the surface code. The change in syndrome weight is assessed in a local region so that the analysis is not distracted by distant parts of the decoding problem. In the particular case of our T gate proxy, this means we localize this analysis on the target qubit in temporal proximity to the severance [dashed red line in Fig. 6(b)]—the place where increases to the weight of the correction string is most directly indicative of a likely temporal windowing failure.

- (2) Inspired by soft-output decoding constructions [50, 51], we perform a further complementary matching on a modified DEM, where the open temporal boundary is turned into a closed boundary, meaning error strings can no longer terminate at a temporal

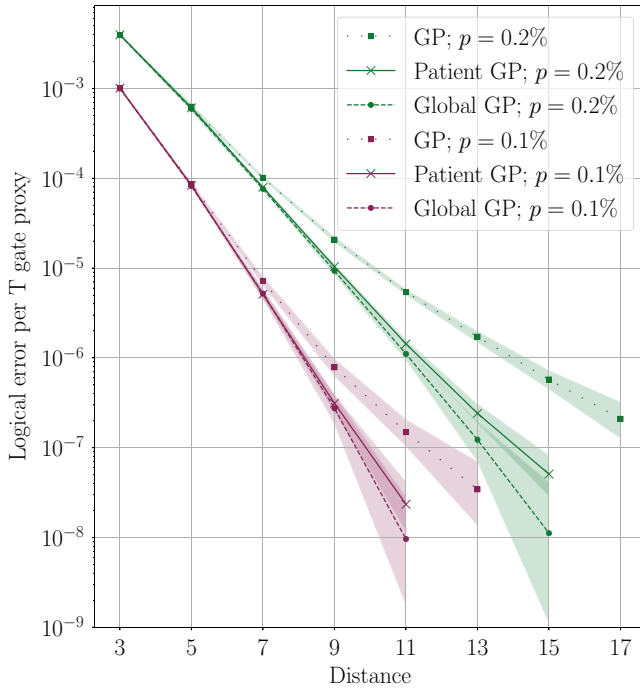


FIG. 12. Patient temporal windowing decoder. Here we overlay the accuracy data presented in Fig. 7 with further data derived from when the ghost protocol is extended with patience (“Patient GP”). When the decoder heralds the need to be patient, we collect further rounds of syndrome information before redecoding the logical measurement. In a fully windowed setting, patience overcomes the failure modes of weight $\approx d/4$ described in Sec. V. This is achieved without incurring a costly time penalty from enforcing $n_{\text{buf}} \approx d/2$ across all T gates. See Table I for data on the average time penalty of patience.

boundary. With this complementary matching, we can herald cases that yield a different correction bit to the previous decoding with the open temporal boundary. This functions to detect most cases where the ghost protocol fails to discover order-three hyperedges that occurred [for example, the case shown in Fig. 10(b)].

Together, these techniques give us a mechanism for efficiently reducing the gap between the error suppression seen under global decoding of T gates and the temporally windowed decoder (Fig. 12). Patience is effective because TW errors are rare [see Fig. 7(b)] and therefore to herald them (even with a relatively high probability of false positives) and delay the computation when these events are encountered has a minimal impact on average n_{buf} and, by extension, the logical clock rate of the quantum computer. False positive heralds are benign in terms of accuracy, though the goal is to suppress them such that n_{buf} remains small on average without increasing false negatives, which inevitably lead to logical errors. Table I confirms that the average cost of patience (the number of additional rounds

TABLE I. Average patience delay. For each distance and noise model simulated in Fig. 12, we report the number of rounds of syndrome extraction performed in n_{buf} when patience is applied (see “delay”). With these patience delays, n_{buf} is increased such that $w = \lceil d/2 \rceil$ where w is defined in Eq. (1). Without a patience delay $n_{\text{buf}} = 1$ is fixed. The average number of rounds of delay incurred per T gate proxy through patience is given for $p = 0.1\%$ and $p = 0.2\%$. Since patience is applied rarely in the subthreshold regime, its impact on average n_{buf} per T gate proxy is negligible. Recall that without patience, ghost decoding of state teleportation in the large d regime would require an overhead of $n_{\text{buf}} \approx d/2$ per T gate to reliably match the error suppression behavior of the global decoder. We note that reductions in the physical error rate p minimize the overhead of patience.

d	Delay	Average patience delay (num rounds)	
		$p = 0.1\%$	$p = 0.2\%$
3	0	0	0
5	1	$(5.227 \pm 0.051) \times 10^{-4}$	$(2.284 \pm 0.014) \times 10^{-3}$
7	2	$(1.125 \pm 0.007) \times 10^{-3}$	$(5.031 \pm 0.020) \times 10^{-3}$
9	3	$(2.324 \pm 0.005) \times 10^{-3}$	$(9.764 \pm 0.015) \times 10^{-3}$
11	4	$(4.029 \pm 0.006) \times 10^{-3}$	$(1.545 \pm 0.001) \times 10^{-2}$
13	5	—	$(2.077 \pm 0.001) \times 10^{-2}$
15	6	—	$(2.492 \pm 0.002) \times 10^{-2}$

of syndrome extraction that patience introduces per T gate) is negligible in the subthreshold regime. Together, this confirms the viability of fast T gates in a fully windowed context.

VII. DISCUSSION AND OUTLOOK

In this work we proposed two new decoding protocols—the ghost protocol and patience—that combine to enable scalable decoding of fast transversal logic. We analyzed the performance of the ghost protocol on deep logical Clifford circuits with as few as one round of syndrome extraction between logical gate layers. Coupling the ghost protocol with patience, we demonstrated scalable, modular decoding strategies for non-Clifford logic, focusing on the structural properties of the T gate realized via teleportation. Further, we characterized the topological structures that give rise to a resilience property that facilitates fast state teleportation despite aggressive temporal windowing.

By lattice surgery, a multiqubit $\pi/8$ rotation requires $2d$ rounds of syndrome extraction on a planar lattice using the Litinski architecture [52] due to the complexity of measuring arbitrary Pauli products [17]. Achieving upward of 10^9 T /Toffoli gates by lattice surgery with kHz syndrome extraction circuitry has daunting implications for the time overhead of large-scale fault-tolerant algorithms. By enabling scalable, fast transversal logic our protocols deliver an $O(d)$ speedup for qubits with high connectivity. This is achieved without the significant increases in

the space overhead of the computation incurred in Refs. [18,53]. The topic of fast transversal logic is subtle and expert opinion is divided, so future non-Clifford simulations that study the time complexity of logic in relation to specific decoding schemes may be valuable.

Since our approach is compatible with graph-based decoders with MHz throughputs (e.g., [30,31]), we can estimate the number of iterations of the ghost protocol that could be performed within the latency budget of AMO hardware (1 ms). Assuming an inner decoder with a latency of about 10 μ s for a surface code memorylike decoding volume [30], and the message passing between decoders during a single iteration of the decoding problem taking 10 μ s, 50 iterations of our decoding protocol could be performed within 1 ms. The maximum number of iterations needed in the simulations presented in this paper was 8 (see Appendix B), leaving a comfortable margin for decoding within the AMO latency budget using the ghost protocol.

Compressing the spatial dimension of a quantum computation is a developed and active area of research [5,54–61]. While our study concerns the surface code, which has a lower encoding rate and therefore greater space overhead than nonsurface quantum low density parity check (qLDPC) codes, it is important to consider the comparative value of compressing different axes of the spacetime volume in a way that accounts for the specificities of the underlying hardware. In an AMO context, the speed of syndrome extraction means we anticipate that codes with favorable characteristics in time may be preferable even over codes that reduce the spacetime volume more substantially through acting in space. Further study will be necessary to quantify this trade-off, but we suggest that the possibility of fast transversal logic in the surface code preserves its competitiveness even against the backdrop of further innovation in the qLDPC space. Regarding the applicability of our decoding protocols to the broader qLDPC context, we note that our protocols are not dependent on graph-based decoders and could be coupled with fast hypergraph decoders [62–64] that will be necessary in this area. However, the decomposition of hyperedges that span logical qubits would yield higher weight ghost *hyperedges* and further techniques for iteratively correlating these hyperedges may need to be found.

Fast transversal logic is difficult to decode not just from a decoder throughput perspective but also because it challenges key structural properties that make real-time decoding of lattice surgery efficient. In lattice surgery we have a modular, blocklike structure which gives predictability and repetitiveness to the decoding problem as well as a local memory model that benefits highly distributed computation. Under fast transversal logic it would seem like we have to completely reformulate the decoding problem from scratch in response to each logical measurement. Alternative proposals for decoding

fast transversal logic [19,20] are less heuristic in their approach, but transpose into a matchable problem where decoding throughput is considered acceptable for AMO at the expense of a potentially crippling reconfiguration overhead: each decoding problem is circuit specific and modularity is lost. This overhead is unquantified and, as discussed above, may be consequential in a streaming setting. With patient ghost decoding, fast transversal logic is windowed with minimal reconfiguration overhead. Our protocols enable fast transversal logic at scale, securing a crucial speedup for qubit types with high connectivity but slow physical gate times.

ACKNOWLEDGMENTS

We thank Ben Barber, Dan Browne and Joonas Majaniemi for feedback on a draft of this paper. We thank Luka Skoric for valuable discussions about windowing. We thank Steve Brierley and our colleagues at Riverlane for creating a stimulating environment for research.

M.L.T. led the research, developing, implementing and evaluating the decoding protocols. J.C. and M.L.T. wrote the paper with input from all authors. All authors contributed to discussions on research direction and problem-solving approaches. O.C. and M.L.T. generated the circuits simulated in this work.

DATA AVAILABILITY

The data that support the findings of this article are openly available [65].

APPENDIX A: NOISE MODEL

We adopt the circuit-level noise model proposed in Ref. [66] across all our numerical simulations. The model is parameterized by a single probability p and introduces independent channels of the following type and strength throughout our Stim circuits:

- (1) Depolarization of both qubits after each two-qubit Clifford gate with probability p .
- (2) Depolarization of each idle qubit and after each single-qubit Clifford gate, including measurement and reset operations, with probability $p/10$.
- (3) Randomly change the result of a measurement with probability p .

The only exception to this is that logical stabilizers at the end of our deep logical circuits (Fig. 5) are measured noiselessly with multiqubit Pauli product measurements. All error bars in our data represent likelihoods within a factor of 1000 of the max likelihood hypothesis, given the sampled data. All Stim circuits simulated in this paper are available on Zenodo (DOI: 10.5281/zenodo.15544576).

APPENDIX B: PARAMETERS AND DECODING PASSES

By default the ghost protocol performs four decoding passes. That is, all logical qubits get decoded independently four times, each time exchanging messages between one another and refining the syndrome (see Sec. III for a full description of the message passing). In this default configuration, the first decoding pass involves turning on ghost singletons [g_s in Fig. 1(b)] in the first pass, a fast reweighting operation that is linear in the number of ancilla qubits. This default configuration is used in the numerical simulations of the T gate proxy—i.e., the data shown in Figs. 7 and 12. We believe the default configuration to be sufficient for most logical circuits.

The number of decoding passes and the content of these passes (whether or not ghost singletons, g_s , are exposed to the decoder) is however a parameter that needs to be evaluated in relation to the structure of the logical circuit, the number of rounds of syndrome extraction between logical gates and the code distance. For the numerical simulations of deep logical circuits (Fig. 5) the default configuration underlies the data points at all values of n_r for $d = 3, 5, 7, 9$. However, at $d = 11$ and $n_r = 2, 3$ we have to increase the number of decoding passes to six, displaying ghost singletons during the first and fourth decoding pass, to maintain exponential error suppression in the distance. At $d = 11$ and $n_r = 1$ the protocol has to work harder still to purge the syndrome of interpatch, order-three hyperedges, performing eight passes, showing boundary edges on the first, fourth and sixth decoding pass. We currently determine these parameters empirically and further study is needed to work out how to set these parameters automatically in regimes that are hard to simulate. A fundamental principle that has worked well is to include ghost singletons in the decoding problem during the initial pass and then occasionally in intermediate decoding iterations; for the reasons discussed in Sec. III ghost singletons should never be shown during the final pass.

As the number of rounds between transversal CNOTs shrinks, order-three hyperedges offset in time and with opposite orientations conflict with one another and it takes more time to discover them. Each time we discover a ghost edge there is a possibility that a subsequent matching opens up the possibility of further ghost edges appearing in a subsequent correction string. Under ghost decoding this can only be solved iteratively. We have not seen the number of passes required to decode effectively exceed the distance of the code giving us confidence in the tractability of our scheme at high distances.

In practice configuring the decoding system for the ghost protocol involves:

- (1) Constructing a DEM using Stim [27].
- (2) Decomposing hyperedges that flip detectors across logical qubits into ghost edges [g_e and g_s in

ALGORITHM 1. Ghost protocol.

```

 $i \leftarrow 0$ 
while  $i < n_{iter}$  do
  if iteration  $i$  requires exposing  $g_s$  then
    for all  $d_i \in \text{decoders}$  do                                ▷ in parallel
      expose  $g_s$  on  $d_i$ 
    for all  $d_i \in \text{decoders}$  do                                ▷ in parallel
      decode  $q_L^i$  on  $d_i$ 
       $c_i \leftarrow$  correction edges from  $d_i$ 
      for all  $g_e \in c_i$  do
        update syndrome by flipping  $g_e$ 
        message  $d_j$  to flip the  $g_s$  correlated with  $g_e$ 
  if  $i < n_{iter} - 1$  then
    discard all correction edges
  else
    return the overall logical correction
   $i \leftarrow i + 1$ 

```

Fig. 1(b)]. Note, the order-three hyperedges can embed in higher weight hyperedges, meaning the splitting into g_e and g_s needs to be supplemented by more standard decomposition techniques where hyperedges are decomposed into both ghost edges and existing lower-order graphlike error mechanisms.

- (3) Recording which instances of g_e and which instances of g_s are correlated with one another via order-three hyperedges.
- (4) Assigning a decoder to each logical qubit. This spatial locality is enabled by the previously performed decomposition. Note: for the surface code, this will look like a quantum memory decoding graph and a streaming decoder will have to know how to replicate this structure through a succession of temporal windows. Similarly, an emergent property of our architecture is that a streaming decoder can store a single mapping that captures the relationship between instances of g_e and instances of g_s , avoiding the need to perform bespoke decomposition operations for full logical circuits on the fly.
- (5) Determining the number of iterations n_{iter} as well as the iterations in which instances of g_s are exposed to the decoder from empirical data.

At runtime the ghost protocol proceeds as shown in Algorithm 1.

-
- [1] D. Bluvstein *et al.*, Logical quantum processor based on reconfigurable atom arrays, *Nature* **626**, 58 (2024).
 - [2] P. S. Rodriguez *et al.*, Experimental demonstration of logical magic state distillation, *Nature* **645**, 620 (2025).
 - [3] B. W. Reichardt *et al.*, Logical computation demonstrated with a neutral atom quantum processor, *arXiv:2411.11822*.

- [4] A. Paetzniak *et al.*, Demonstration of logical qubits and repeated error correction with better-than-physical error rates, [arXiv:2404.02280](#).
- [5] 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).
- [6] F. A. An, A. Ransford, A. Schaffer, L. R. Sletten, J. Gaebler, J. Hostetter, and G. Vittorini, High fidelity state preparation and measurement of ion hyperfine qubits with $I > \frac{1}{2}$, *Phys. Rev. Lett.* **129**, 130501 (2022).
- [7] Google Quantum AI and Collaborators, Quantum error correction below the surface code threshold, *Nature* **638**, 920 (2025).
- [8] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, G. J. Norris, C. K. Andersen, M. Müller, A. Blais, C. Eichler, and A. Wallraff, Realizing repeated quantum error correction in a distance-three surface code, *Nature* **605**, 669 (2022).
- [9] I. Besedin, M. Kerschbaum, J. Knoll, I. Hesner, L. Bideker, L. Colmenarez, L. Hofele, N. Lacroix, C. Hellings, F. Swiadek, A. Flasby, M. B. Panah, D. C. Zanuz, M. Müller, and A. Wallraff, Realizing lattice surgery on two distance-three repetition codes with superconducting qubits, [arXiv:2501.04612](#).
- [10] L. Caune *et al.*, Demonstrating real-time and low-latency quantum error correction with superconducting qubits, [arXiv:2410.05202](#).
- [11] H. Ali, J. Marques, O. Crawford, J. Majaniemi, M. Serraperralta, D. Byfield, B. Varbanov, B. M. Terhal, L. DiCarlo, and E. T. Campbell, Reducing the error rate of a superconducting logical qubit using analog readout information, *Phys. Rev. Appl.* **22**, 044031 (2024).
- [12] M. Cain, C. Zhao, H. Zhou, N. Meister, J. P. B. Ataiades, A. Jaffe, D. Bluvstein, and M. D. Lukin, Correlated decoding of logical algorithms with transversal gates, *Phys. Rev. Lett.* **133**, 240602 (2024).
- [13] N. Delfosse, V. Londe, and M. Beverland, Toward a union-find decoder for quantum LDPC codes, *IEEE Trans. Inf. Theory* **68**, 3187 (2022).
- [14] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, *New J. Phys.* **14**, 123011 (2012).
- [15] H. Bombin, C. Dawson, Y.-H. Liu, N. Nickerson, F. Pastawski, and S. Roberts, Modular decoding: Parallelizable real-time decoding for quantum computers, [arXiv:2303.04846](#).
- [16] K. Sahay, Y. Lin, S. Huang, K. R. Brown, and S. Puri, Error correction of transversal CNOT gates for scalable surface-code computation, *PRX Quantum* **6**, 020326 (2025).
- [17] C. Chamberland and E. T. Campbell, Universal quantum computing with twist-free and temporally encoded lattice surgery, *PRX Quantum* **3**, 010331 (2022).
- [18] A. G. Fowler, Time-optimal quantum computation, [arXiv:1210.4626](#).
- [19] M. Serraperralta, M. H. Shaw, and B. M. Terhal, Decoding across transversal Clifford gates in the surface code, [arXiv:505.13599](#).
- [20] M. Cain, D. Bluvstein, C. Zhao, S. Gu, N. Maskara, M. Kalinowski, A. A. Geim, A. Kubica, M. D. Lukin, and H. Zhou, Fast correlated decoding of transversal logical algorithms, [arXiv:2505.13587](#).
- [21] B. Eastin and E. Knill, Restrictions on transversal encoded quantum gate sets, *Phys. Rev. Lett.* **102**, 110502 (2009).
- [22] T. Jochym-O'Connor, A. Kubica, and T. J. Yoder, Disjointness of stabilizer codes and limitations on fault-tolerant logical gates, *Phys. Rev. X* **8**, 021047 (2018).
- [23] A. Kubica, B. Yoshida, and F. Pastawski, Unfolding the color code, *New J. Phys.* **17**, 083026 (2015).
- [24] J. E. Moussa, Transversal Clifford gates on folded surface codes, *Phys. Rev. A* **94**, 042316 (2016).
- [25] Z.-H. Chen, M.-C. Chen, C.-Y. Lu, and J.-W. Pan, Transversal logical Clifford gates on rotated surface codes with reconfigurable neutral atom arrays, [arXiv:2412.01391](#).
- [26] G. P. Gehér, C. McLauchlan, E. T. Campbell, A. E. Moylett, and O. Crawford, Error-corrected Hadamard gate simulated at the circuit level, *Quantum* **8**, 1394 (2024).
- [27] C. Gidney, Stim: A fast stabilizer circuit simulator, *Quantum* **5**, 497 (2021).
- [28] J. Edmonds, Paths, trees, and flowers, *Can. J. Math.* **17**, 449 (1965).
- [29] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, *Quantum* **5**, 595 (2021).
- [30] A. B. Ziad, A. Zalawadiya, C. Topal, J. Camps, G. P. Gehér, M. P. Stafford, and M. L. Turner, Local clustering decoder as a fast and adaptive hardware decoder for the surface code, *Nat. Commun.* **16**, 11048 (2025).
- [31] N. Liyanage, Y. Wu, S. Tagare, and L. Zhong, FPGA-based distributed union-find decoder for surface codes, *IEEE Trans. Quantum Eng.* **5**, 1 (2024).
- [32] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *J. Math. Phys.* **43**, 4452 (2002).
- [33] B. M. Terhal, Quantum error correction for quantum memories, *Rev. Mod. Phys.* **87**, 307 (2015).
- [34] 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).
- [35] Y. Wu and L. Zhong, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2023), pp. 928–938.
- [36] C. Gidney, Halving the cost of quantum addition, *Quantum* **2**, 74 (2018).
- [37] K. H. Wan, M. Webber, A. G. Fowler, and W. K. Hensinger, An iterative transversal CNOT decoder, [arXiv:2407.20976](#).
- [38] A. G. Fowler, Optimal complexity correction of correlated errors in the surface code, [arXiv:1310.0863](#).
- [39] O. Higgott and C. Gidney, Sparse Blossom: Correcting a million errors per core second with minimum-weight matching, *Quantum* **9**, 1600 (2025).
- [40] A. Guernut and C. Vuillot, Fault-tolerant constant-depth Clifford gates on toric codes, [arXiv:2411.18287](#).
- [41] S. Bravyi, M. B. Hastings, and F. Verstraete, Lieb-Robinson bounds and the generation of correlations and topological quantum order, *Phys. Rev. Lett.* **97**, 050401 (2006).
- [42] M. B. Hastings, Topological order at nonzero temperature, *Phys. Rev. Lett.* **107**, 210501 (2011).
- [43] C. Gidney, Inplace access to the surface Code Y basis, *Quantum* **8**, 1310 (2024).

- [44] K. H. Wan, Constant-time magic state distillation, [arXiv:2410.17992](#).
- [45] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, *Phys. Rev. A* **71**, 022316 (2005).
- [46] D. Litinski, Magic state distillation: Not as costly as you think, *Quantum* **3**, 205 (2019).
- [47] C. Gidney and A. G. Fowler, Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation, *Quantum* **3**, 135 (2019).
- [48] C. Gidney, N. Shutty, and C. Jones, Magic state cultivation: Growing T states as cheap as CNOT gates, [arXiv:2409.17595](#).
- [49] 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, *Nature* **646**, 303 (2025).
- [50] C. Gidney, M. Newman, P. Brooks, and C. Jones, Yoked surface codes, *Nat. Commun.* **16**, 4498 (2025).
- [51] N. Meister, C. A. Pattison, and J. Preskill, Efficient soft-output decoders for the surface code, [arXiv:2405.07433](#).
- [52] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, *Quantum* **3**, 128 (2019).
- [53] D. Litinski and N. Nickerson, Active volume: An architecture for efficient fault-tolerant quantum computers with limited non-local connections, [arXiv:2211.15465](#).
- [54] J.-P. Tillich and G. Zemor, in *2009 IEEE International Symposium on Information Theory*, Vol. 60 (2014), pp. 1193–1202.
- [55] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, *PRX Quantum* **2**, 040101 (2021).
- [56] P. Panteleev and G. Kalachev, Degenerate quantum LDPC codes with good finite length performance, *Quantum* **5**, 585 (2021).
- [57] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, *Nature* **627**, 778 (2024).
- [58] M. H. Shaw and B. M. Terhal, Lowering connectivity requirements for bivariate bicycle codes using morphing circuits, *Phys. Rev. Lett.* **134**, 090602 (2025).
- [59] N. Berthussen, D. Devulapalli, E. Schoute, A. M. Childs, M. J. Gullans, A. V. Gorshkov, and D. Gottesman, Toward a 2D local implementation of quantum low-density parity-check codes, *PRX Quantum* **6**, 010306 (2025).
- [60] A. J. Malcolm, A. N. Glaudell, P. Fuentes, D. Chandra, A. Schotte, C. DeLisle, R. Haenel, A. Ebrahimi, J. Roffe, A. O. Quintavalle, S. J. Beale, N. R. Lee-Hone, and S. Simmons, Computing efficiently in QLDPC codes, [arXiv:2502.07150](#).
- [61] Q. Xu, J. P. Bonilla Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasić, M. D. Lukin, L. Jiang, and H. Zhou, Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays, *Nat. Phys.* **20**, 1084 (2024).
- [62] S. Wolanski and B. Barber, Ambiguity clustering: An accurate and efficient decoder for qLDPC codes, [arXiv:2406.14527](#).
- [63] T. Hillmann, L. Berent, A. O. Quintavalle, J. Eisert, R. Wille, and J. Roffe, Localized statistics decoding: A parallel decoding algorithm for quantum low-density parity-check codes, *Nat. Commun.* **16**, 8214 (2025).
- [64] K. Yin, X. Fang, J. Ruan, H. Zhang, D. Tullsen, A. Sornborger, C. Liu, A. Li, T. Humble, and Y. Ding, SymBreak: Mitigating quantum degeneracy issues in QLDPC code decoders by breaking symmetry, [arXiv:2412.02885](#).
- [65] M. Turner, Stim circuits for paper: “Scalable decoding protocols for dense transversal logic in the surface code,” <https://zenodo.org/records/15544576> (2025).
- [66] B. Barber, K. M. Barnes, T. Bialas, O. Buğdaycı, E. T. Campbell, N. I. Gillespie, K. Johar, R. Rajan, A. W. Richardson, L. Skoric, C. Topal, M. L. Turner, and A. B. Ziad, A real-time, scalable, fast and resource-efficient decoder for a quantum computer, *Nat. Electron.* **8**, 84 (2025).