Accelerating Fault-Tolerant Quantum Computation with Good qLDPC Codes

Guo Zhang,[1] Yuanye Zhu,[1] and Ying Li[1,*]

[1] *Graduate School of China Academy of Engineering Physics, Beijing 100193, China*

We propose a fault-tolerant quantum computation scheme that is broadly applicable to quantum low-density parity-check (qLDPC) codes. The scheme achieves constant qubit overhead and a time overhead of $O(d^{a+o(1)})$ for any $[[n,k,d]]$ qLDPC code with constant encoding rate and distance $d=\Omega(n^{1/a})$. For good qLDPC codes, the time overhead is minimized and reaches $O(d^{1+o(1)})$. In contrast, code surgery based on gauging measurement and brute-force branching requires a time overhead of $O(dw^{1+o(1)})$, where $d\leq w\leq n$. Thus, our scheme is asymptotically faster for all codes with $a<2$. This speedup is achieved by developing techniques that enable parallelized code surgery under constant qubit overhead and leverage classical locally testable codes for efficient resource state preparation. These results establish a new paradigm for accelerating fault-tolerant quantum computation on qLDPC codes, while maintaining low overhead and broad applicability.

I. INTRODUCTION

Quantum error correction is essential to realizing large-scale quantum computation [1–4]. According to the fault-tolerance theorem, when the physical error rate is below a constant threshold, arbitrarily high computational accuracy can be achieved by choosing a quantum error correction code with sufficiently large code distance d [5, 6]. However, fault-tolerant quantum computation (FTQC) inevitably introduces additional resource overheads: each logical qubit requires multiple physical qubits for encoding and for supporting its computational operations, referred to as *qubit overhead*; each round of logical-qubit operations must be implemented through multiple rounds of physical-qubit operations, referred to as *time overhead*. For instance, when computation is performed using the surface code via lattice surgery, the qubit overhead scales as $O(d^2)$, while the time overhead scales as $O(d)$ [7, 8].

Substantial effort has been devoted to reducing the resource overhead of FTQC. A major advance in this direction is the development of low-overhead quantum low-density parity-check (qLDPC) codes, such as hypergraph product (HGP) codes, which achieve a constant encoding rate [9]. This property ensures that the logical qubit number k scales linearly with the physical qubit number n, i.e. $k=\Theta(n)$. Recent breakthroughs have led to the discovery of good qLDPC codes, which further improve the code distance from $d=\Theta(n^{1/2})$ to $d=\Theta(n)$ while maintaining a constant encoding rate [10–14].

To enable logical computational operations on qLDPC codes, two main strategies have been developed: the concatenated codes combined with gate teleportation (CC+GT) approach [15–18] and the code surgery approach (a generalization of lattice surgery) [8, 19–29]. For general qLDPC codes, the most efficient known method is the gauging measurement combined with brute-force branching (GM+BFB)—a form of code surgery that achieves constant qubit overhead and a time overhead scaling as $O(d^{2+o(1)})$ [assuming the logical operator weight $w=O(d)$] [22–24, 28]. A lower time overhead of $O(d^{1+o(1)})$ can be attained using a refined variant of the CC+GT scheme [18]; however, its applicability is restricted to (almost) good quantum locally testable codes (qLTCs)—also known as c^3-qLTCs—which, beyond having constant encoding rate and constant relative distance, must also possess constant soundness [18, 30].

In this work, we propose a new scheme applicable to general qLDPC codes, achieving lower resource overhead than the GM+BFB protocol and extending the $O(d^{1+o(1)})$ time overhead to a broader code families. For qLDPC codes with a constant encoding rate and code distance $d=\Omega(n^{1/a})$, our scheme attains a time overhead of $O(d^{a+o(1)})$ while maintaining constant qubit overhead. Compared with GM+BFB, our method achieves a smaller time overhead for all constant-rate qLDPC codes whose relative distance exceeds that of HGP codes (i.e., $a<2$). In particular, for good qLDPC codes, the time overhead is further reduced to $O(d^{1+o(1)})$. Although this minimal overhead matches the best performance of the CC+GT approach, our scheme achieves it under weaker conditions—requiring only a constant encoding rate and a constant relative distance.

Our scheme integrates code surgery with gate teleportation by performing the parity-check measurements required in code surgery through gate teleportation. To minimize resource overhead, we introduce two key techniques: parallelized code surgery (PCS) and locally-testable state preparation (LTSP). In PCS, multiple code blocks share a common ancilla system (a set of mutually coupled qubits) to implement computational operations, thereby reducing the qubit overhead of code surgery to a constant. In LTSP, the resource states required for gate teleportation are prepared with constant qubit and time overhead, employing a classical locally testable code (LTC) that requires only two constants—constant encoding rate and constant soundness [13, 31, 32]. Leveraging local testability, we can effectively control measurement errors in parity-check measurements: when performing code surgery with resource states prepared via LTSP, only a single round of parity-check measurements

* yli@gscaep.ac.cn

is needed, reducing the time overhead of code surgery from $O(d)$ to $O(1)$. The introduction of PCS and LTSP inevitably leads to serialization of logical operations, resulting in a final time overhead of $O(d^{a+o(1)})$. As general overhead-reduction techniques, PCS and LTSP not only improve the asymptotic scaling of FTQC but also offer practical means to lower overhead in near-term, finite-size implementations.

II. RELATED WORKS

Two general approaches have been developed for fault-tolerant quantum computation with qLDPC codes: CC+GT and code surgery.

Gate teleportation implements logical gates by consuming pre-prepared resource states [15, 17, 18]. Gottesman proposed using gate teleportation to operate on logical qubits encoded in qLDPC codes, with resource states generated via concatenated codes [15]. This approach is made possible by the existence of well-established fault-tolerant quantum computation protocols in certain concatenated codes, such as the concatenated Steane codes [16]. In Gottesman's approach, the resource state is prepared by simulating its preparation circuit using fault-tolerant operations of the concatenated code, producing an encoded resource state. This is then decoded using the decoding circuit of the concatenated code to yield the unencoded resource state needed for gate teleportation. Since the simulation is fault-tolerant and the decoding circuit introduces only local stochastic errors at an asymptotically constant rate, the final resource state can be made sufficiently accurate to enable reliable gate teleportation.

Several variants of the CC+GT approach have been developed, all achieving constant qubit overhead [17, 18]. Their primary difference lies in the time overhead. Consider a quantum computer consisting of M memory blocks, each encoded using an $[[n, k, d]]$ qLDPC code. To implement a quantum circuit of size $|C|$ on the $K = Mk$ logical qubits with a target failure rate ϵ, the logical error rate per operation must be reduced to $p_L = O(\epsilon/|C|)$. Achieving this requires the concatenated code to have sufficiently large distance, ensuring that the encoded resource states are prepared with error rate $O(p_L)$. This requirement imposes a significant qubit cost on resource state production. To maintain a constant total qubit overhead, resource states are provided to only a subset of memory blocks at any given time. As a result, each layer of logical operations must be executed over multiple time steps. The overall time overhead is determined by both the cost of preparing resource states and the complexity due to serialization. In Gottesman's original protocol, and its subsequent improvement [15, 16], this overhead scales as $O(\mathrm{poly}(K))$. A more recent advancement [17] reduces the time overhead to

$$O\left(\log^{2a+\gamma_1+\gamma_2}\left(\frac{|C|}{\epsilon}\right)\right) = O\left(d^{2a+\gamma_1+\gamma_2}\right),$$

where γ_1 and γ_2 are parameters characterizing the concatenated code; note that $|C|/\epsilon = O(\mathrm{poly}(K))$ in typical scenarios. Here, we have re-expressed the time overhead by assuming an exponential suppression of the logical error rate with code distance, i.e., $p_L = e^{-\Omega(d)}$, allowing for a direct comparison with code-surgery protocols.

A further breakthrough was reported in Ref. [18], which introduced state distillation protocols capable of achieving an almost constant spacetime overhead. Before distillation, a concatenated code with modest distance is used to produce resource states with a sufficiently small constant error rate. These are subsequently distilled to achieve the target error rate of $O(p_L)$. Due to the efficiency of the distillation process, the time cost of preparing resource states becomes negligible, making serialization cost the dominant contribution to the overall time overhead. Consequently, the total time overhead is reduced to

$$O\left(\log^{1+o(1)}\left(\frac{|C|}{\epsilon}\right)\right) = O(d^{1+o(1)}).$$

However, since the efficient distillation relies on local testability, achieving this $O(d^{1+o(1)})$ overhead requires the underlying code to be an (almost) good qLTC, characterized by triple constants: constant encoding rate, constant relative distance, and constant soundness.

Code surgery is a generalization of lattice surgery, originally developed for implementing logical operations on surface codes [8, 19, 20, 22–26, 28, 29]. In code surgery, the original code is temporarily deformed by coupling it to an ancilla system, enabling the measurement of one or more selected logical Pauli operators. These logical measurements can then be used to implement Clifford gates. Code surgery for qLDPC codes was first proposed in Ref. [19], known as the CKBB protocol, named after its authors. In this protocol, a logical operator is measured using an ancilla system of $O(dw)$ physical qubits, where w is the weight of the logical operator. Since $w \geq d$, the required ancilla size is comparable to that of a surface code. Subsequent works introduced gauging measurements, which reduced the ancilla size to $O(w^{1+o(1)})$ physical qubits [20, 23, 24]. Gauging measurements achieve this by transforming the relevant checks into those of a repetition code and constructing the ancilla system using an expander graph, thereby achieving robustness against certain errors that the CKBB protocol mitigates using additional qubits.

Code surgery faces an inherent limitation in time efficiency: the parity-check measurements of the deformed code must be repeated for $\Theta(d)$ rounds to reliably correct measurement errors. As a result, lattice surgery on surface codes incurs a time overhead of $O(d)$. In the CKBB and gauging measurement protocols, when logical operators are measured sequentially—one in each block at a time—the total time overhead for measuring $\Theta(k)$ logical operators (a layer of operations on k logical qubits) is $O(kd) = O(d^{1+a})$.

To improve time efficiency, general techniques have

Code	Protocol	Qubit overhead exponent	Time overhead exponent	Refs.
qLDPC codes with $k=\Theta(n)$ and $d=\Omega(n^{1/a})$	GM+BFB	0	≥ 2	[25, 28]
	DS	1	1	[22]
	polylog CC+GT	0	$\geq 2a$	[17]
	PCS+LTSP+GT	0	a	This work
Good qLTCs	log CC+GT	0	1	[18]
	PCS+LTSP+GT	0	1	This work
Surface codes	LS	2	1	[33]

TABLE I. Qubit and time overheads in fault-tolerant schemes using quantum low-density parity-check (qLDPC) codes. The exponent b in the overhead expression $O(d^{b+o(1)})$ is referred to as the overhead exponent. Techniques developed for code surgery include gauging measurement (GM) [23, 24], brute-force branching (BFB) [22], and devised sticking (DS) [22]. Typical protocols based on gate teleportation (GT) utilize concatenated codes (CC) [15, 17, 18]. We note that the overhead estimates for the polylog CC+GT [17] and log CC+GT [18] protocols rely on the assumption of a single-shot decoder and good quantum locally testable codes, respectively. Lattice surgery (LS) [33] is a method for quantum computation on the surface code (SC). This work introduces two new techniques for reducing time overhead: parallelized code surgery (PCS) and locally-testable state preparation (LTSP).

been developed to construct deformed codes that support the simultaneous measurement of multiple logical operators. Two main approaches are devised sticking and brute-force branching [22]. In devised sticking, an arbitrary number of logical operators can be measured in parallel using a single ancilla system of $O(nd)$ physical qubits [34]. This approach has a time overhead of $O(d)$ and a qubit overhead of $O(d)$, assuming the underlying code has constant encoding rate. In brute-force branching, an intermediate deformed code is constructed to distribute the logical operators across disjoint supports, enabling their measurements to be performed in parallel. A protocol that combines gauging measurements with brute-force branching (GM+BFB) [28] achieves a time overhead of $O(dw^{1+o(1)})$ while reducing the qubit overhead to constant.

Note that code surgery operations alone can only implement Clifford logical gates. A parallelized magic state injection protocol enables the preparation of magic states with (almost) the same resource scaling as code surgery [25].

In addition to typical CC+GT and code surgery protocols, Ref. [17] proposes preparing gate-teleportation resource states using surface codes (or alternatives such as color codes, rather than concatenated codes), and then implementing gate teleportation on high-rate qLDPC code blocks via code surgery.

III. SCHEME

We now sketch our scheme for FTQC. In our scheme, we adopt the code-surgery approach and propose to implement code surgery via gate teleportation. We introduce two key techniques to minimize qubit and time overheads: PCS and LTSP; see Fig. 1 and Sec. IV. In PCS, we construct deformed codes that enable simultaneous logical operations on multiple code blocks and achieve constant qubit overhead. In LTSP, we generate gate-teleportation resource states exploring classical LTCs, minimizing the time cost while maintaining constant qubit overhead. As a result, our scheme achieves a time overhead of $O(d^{a+o(1)})$ and constant qubit overhead, as summarized in Theorem 1.

Theorem 1. *Consider a classical-input/classical-output quantum circuit C composed of Clifford and T gates, with size $|C| = WD$, where W denotes the width and $D = \mathrm{poly}(W)$ denotes the depth. There exists a constant $\epsilon_* \in (0,1)$ such that for any $\epsilon_L \in (0,1)$, one can efficiently construct a fault-tolerant version C_{FT} with width W_{FT} and depth D_{FT} satisfying*

$$\frac{W_{\mathrm{FT}}}{W} = O_{W\to\infty}(1), \tag{1}$$

$$\frac{D_{\mathrm{FT}}}{D} = O_{W\to\infty}(\log^{a+o(1)}(\frac{|C|}{\epsilon_L})), \tag{2}$$

where a is a constant associated with the memory code block. If C_{FT} operates under a local stochastic σ-error model with $\sigma \in \{X, Z\}$ and physical error rate $p_{\mathrm{phy}} \leq \epsilon_*$ at every location, then the total variation distance between the output distributions of C and C_{FT} is at most ϵ_L.

A detailed proof is provided in Appendix H 4.

A. Codes

Our scheme employs a set of error correction codes, as listed in Table II. A qLDPC code, referred to as the memory code, is used to encode logical qubits. We perform operations on these logical qubits via code surgery, which requires the construction of corresponding deformed codes. A classical LDPC code, called the R code,

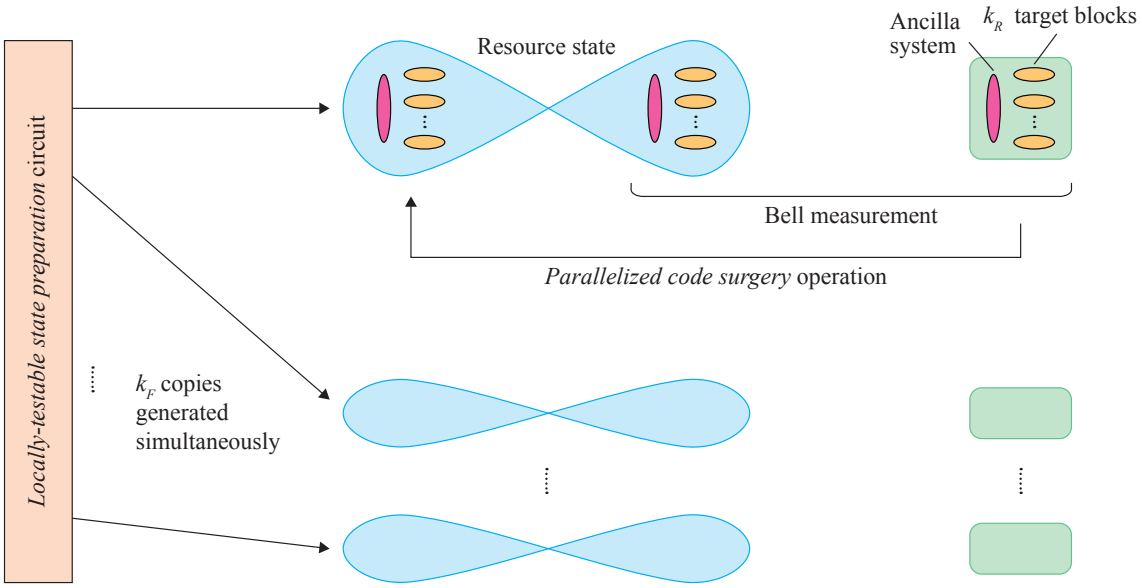

FIG. 1. Schematic illustration of our scheme. Each execution of the locally-testable state-preparation circuit produces k_F copies of the resource state, which are then used in gate teleportation to implement parallelized code surgery (PCS). PCS applies logical operations simultaneously on k_R target code blocks, using a single ancilla system shared across the k_R blocks. The locally-testable state-preparation circuit has almost constant qubit and time overhead, while gate-teleportation-based PCS also maintains constant qubit and time overhead.

Codes	Parameters	Examples
Memory code (Q)	$k = \Theta(n)$ and $d = \Omega(n^{1/a})$	Hypergraph product, $a = 2$ [9] Quantum Tanner, $a = 1$ [35] Expander LP , $a = 1 + \epsilon$ [13] DLV, $a = 1 + \epsilon$ [30] LH, $a = 1$ [36] DHLV, $a = 1$ [14]
R code (C)	$k_R = \Theta(n_R)$ and $d_R = \Omega\left(n_R^{1/a_R}\right)$	Expander, $a_R = 1$ [37] Tanner, $a_R = 1$ [38]
F code (C)	$k_F = \Theta(n_F)$, $d_F = \Omega\left(n_F^{1/a_F}\right)$, and $s = \Theta(1)$	PK, $a_F = 1$ [13] Lossless-expander LTCs, $a_F = 1$ [32]

TABLE II. Quantum (Q) and classical (C) codes utilized in our protocol. The quantum computer is structured into multiple blocks, each encoded in a qLDPC code, called memory code. The quantum computation is performed on logical qubits within these memory-code blocks via code surgery, where each operation corresponds to a specific deformed code. The deformed code is generated utilizing a classical linear code, called R code. To implement the parity-check measurements of a deformed code (or the memory code), we prepare the corresponding gate-teleportation resource states using another classical linear code, referred to as the F code. This code has to be a locally testable code, and s denotes its soundness. In examples, ϵ denotes an arbitrary positive number. Additionally, we employ a low-distance surface code to suppress errors during resource state preparation and to inject T-gate magic states, and a high-distance color code to prepare S-gate magic states. We use d_S and d_C to denote distances of the surface and color codes, respectively.

is used in constructing the deformed codes, replacing the role of the repetition code in the CKBB protocol. Another classical LDPC code, referred to as the F code, serves as the core component of the resource-state factory, which produces the resource states required for gate teleportation. The requirements for these codes, along with representative examples, are presented in Table II. Additionally, our scheme also employs a surface code and a color code for various auxiliary tasks.

B. Quantum computer

As depicted in Fig. 2, the quantum computer comprises three main components: the memory, the resource-state factory, and the magic-state factory. The memory consists of multiple memory-code blocks. M of the blocks store the logical information, giving a total of $K = Mk$ data logical qubits. Here, k is the logical dimension of the memory code. The remaining memory-code blocks serve as ancillas. The resource-state factory produces the resource states required for gate teleportation, while the magic-state factory prepares encoded magic states.

C. Logical measurements—Parallelized code surgery

We achieve universal quantum computation using measurements of logical Pauli operators, supplemented with encoded magic states. Table III summarizes all logical measurements used in our scheme. These measurements are realized through PCS; see Fig. 3 for the circuit. Each logical measurement corresponds to a specifically tailored deformed code. The construction of the deformed codes is detailed in Sec. IV A. A key feature of PCS is that the deformed codes require only a constant qubit overhead.

We refer to the collection of code blocks involved in a logical measurement as a target block. For example, the measurement $\bar{Z}_j \otimes \bar{Z}_1 \otimes \bar{X}_1$ acts on a target block composed of three memory-code blocks. In conventional code surgery, a logical measurement is performed on a single target block at a time. In contrast, PCS performs the same logical measurement simultaneously on k_R identical target blocks, where k_R is the logical dimension of the R code; see Fig. 1.

D. Gate teleportation—Locally-testable state preparation

The fundamental operations in error correction and code surgery are parity-check measurements. In our scheme, we perform parity-check measurements on various quantum codes: the memory code, surface code, color code, and deformed codes. In the standard CSS code formalism, each code corresponds to two sets of stabilizer operators—X-type and Z-type—represented by the corresponding check matrices H_X and H_Z. We measure the X and Z generators separately. For each of H_X and H_Z, the parity-check measurements are implemented via gate teleportation, each consuming a dedicated resource state; see Fig. 4 for the circuit.

We prepare the gate-teleportation resource states using LTSP. Each type of resource state is prepared by a tailored state-preparation circuit. Each execution of such a circuit generates k_F identical copies of the corresponding resource state, where k_F denotes the logical dimension of the F code; see Figs. 1 and 5. The detailed construction of these circuits is provided in Sec. IV B.

A key feature of LTSP is that, by encoding with an LTC, the prepared resource states possess a crucial property: the parity-check measurements performed using these resource states are effectively free of measurement errors. Consequently, only a single round of measurements is required in PCS (and other code-surgery protocols).

E. Universal quantum computation

We adopt the following universal set of logical gates: Hadamard gate, S gate, T gate, and controlled-NOT gate. Using logical measurements, we can realize controlled-NOT and Hadamard gates directly, while S and T gates are implemented via logical measurements assisted by their corresponding magic states. In addition to the logical gates, initialization and measurement of memory-code logical qubits are also required. Table IV summarizes all logical operations, with relevant circuits given in Appendix B.

For the S-gate magic state, we prepare it using a color code with a sufficiently large code distance d_C. The state is first created on a color-code block using the transversal S gate and then transferred to a memory-code block via logical measurements. Because the operations are fault-tolerant, the resulting magic state attains fault-tolerant–level fidelity, eliminating the need for further magic-state distillation.

For the T-gate magic state, we prepare it using a surface code with a small code distance d_S. A noisy magic state is first generated on a surface-code block following the protocol in Ref. [39], and then transferred to a memory-code block. Since the prepared state is noisy, further distillation is required, which is performed using a constant-spacetime-overhead protocol [18].

Each logical S gate requires one S-gate magic state as input. Since the magic state is preserved after use, it can be reused. To exploit this, we prepare M copies of the S-gate magic state prior to computation, with each copy stored in an ancilla memory-code block. These ancilla blocks serve S gates for data memory-code blocks throughout the entire computation.

In contrast, each T gate consumes one copy of the T-gate magic state. Therefore, these magic states must be generated on demand during computation.

F. Cost analysis

We now present a scheme for qubit allocation and logical circuit compilation, followed by an analysis of the associated resource overhead. While the scheme leaves significant room for further optimization, it is sufficient to establish our main conclusions: a constant qubit overhead and a time overhead of $O(d^{a+o(1)})$.

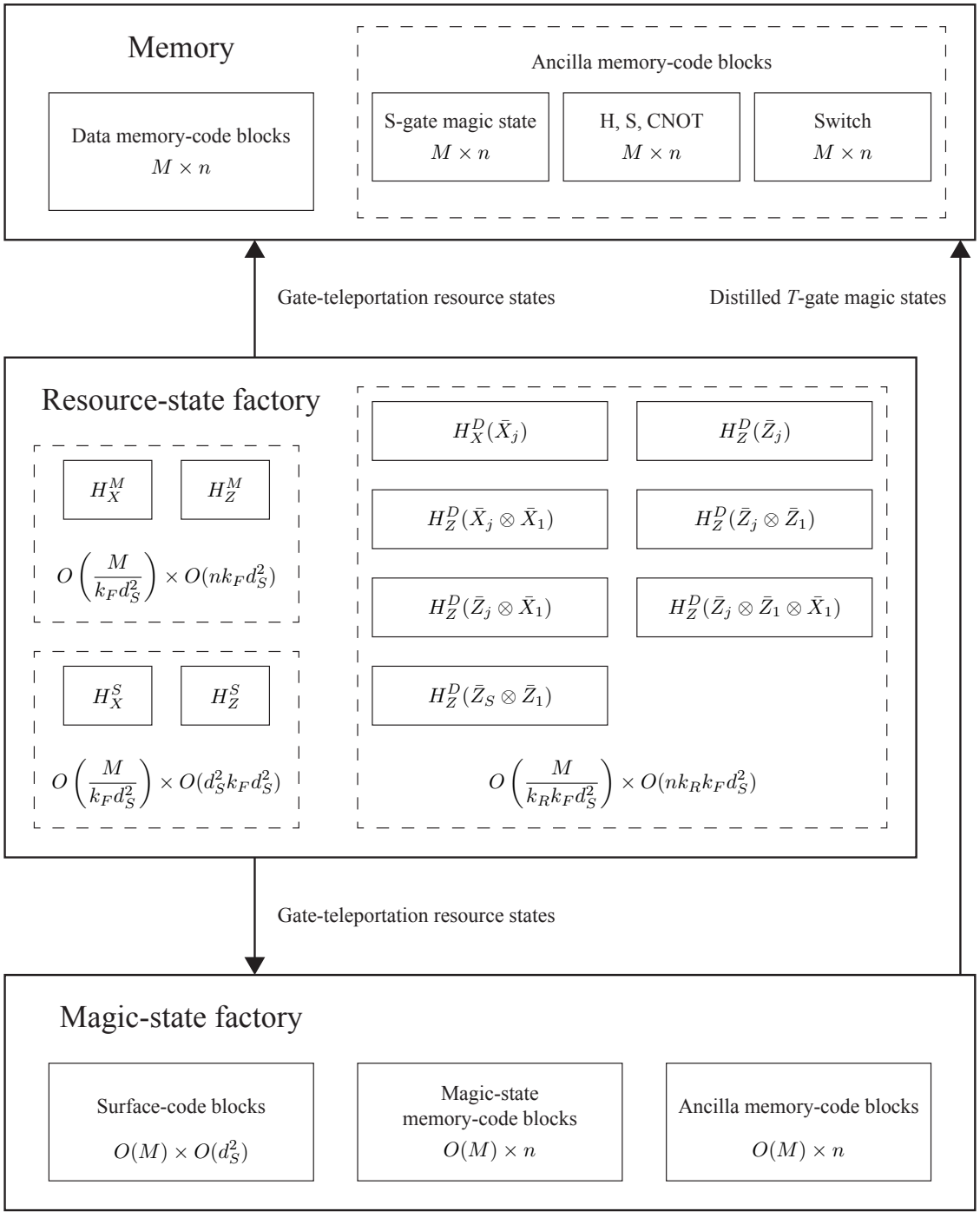

FIG. 2. Quantum computer and allocation of physical qubits. In the memory, the memory-code blocks are divided into four groups according to their functions. Each group contains M blocks, and each block occupies n physical qubits. The resource-state factory produces resource states specified by the corresponding check matrices, which are organized into three families (indicated by the dashed boxes). In each dashed box, the notation $O(A) \times O(B)$ denotes the total number of physical qubits used for generating resource states within that family: each type of resource state is prepared by a tolerated locally-testable state-preparation circuit requiring $O(B)$ physical qubits, and $O(A)$ such circuits are run in parallel. In the magic-state factory, $O(M)$ surface-code blocks are used to inject T-gate magic states into $O(M)$ memory-code blocks, while an additional $O(M)$ ancillary memory-code blocks are employed for magic-state distillation.

Logical operator (set) σ	Resource states
$\bar{X}_j$	$H_Z^D(\bar{X}_j)$, $H_Z^M \times k_R$
$\bar{Z}_j$	$H_Z^D(\bar{Z}_j)$, $H_X^M \times k_R$
$\bar{X}_j \otimes \bar{X}_1$	$H_Z^D(\bar{X}_j \otimes \bar{X}_1)$, $H_Z^M \times 2k_R$
$\bar{Z}_j \otimes \bar{Z}_1$	$H_Z^D(\bar{Z}_j \otimes \bar{Z}_1)$, $H_X^M \times 2k_R$
$\bar{Z}_j \otimes \bar{X}_1$	$H_Z^D(\bar{Z}_j \otimes \bar{X}_1)$, $H_X^M \times k_R$, $H_Z^M \times k_R$
$\bar{Z}_j \otimes \bar{Z}_1 \otimes \bar{X}_1$	$H_Z^D(\bar{Z}_j \otimes \bar{Z}_1 \otimes \bar{X}_1)$, $H_X^M \times 2k_R$, $H_Z^M \times 2k_R$
$\bar{Z}_S \otimes \bar{Z}_1$	$H_Z^D(\bar{Z}_S \otimes \bar{Z}_1)$, $H_X^S \times k_R$, $H_X^M \times k_R$
$\bar{Z}_C \otimes \bar{Z}_1$	$H_Z^D(\bar{Z}_C \otimes \bar{Z}_1)$, $H_X^C \times k_R$, $H_X^M \times k_R$

TABLE III. List of logical measurements. $\bar{X}_j$ ($\bar{Z}_j$) denotes the X (Z) logical operator of the jth logical qubit in a memory-code block; $\bar{Z}_S$ is the Z logical operator of a distance-d_S surface-code block; and $\bar{Z}_C$ is the Z logical operator of a distance-d_C color-code block. A resource state is specified by the corresponding check matrix H: the state H is used to measure Z stabilizer operators represented by H. The notation $H \times m$ denotes m copies of the resource state. Measurements of X stabilizer operators are obtained by applying transversal Hadamard gates to rotate the basis. We use H_P^c with $c = M, D, S, C$ to represent the $P \in \{X, Z\}$ check matrix of the memory, deformed, surface, and color codes, respectively; and $H_Z^D(\sigma)$ denotes the Z check matrix of the deformed code tailored to the measurement of logical operator(s) σ. Each set of resource states enables the simultaneous application of the same logical measurement on k_R identical systems (target blocks).

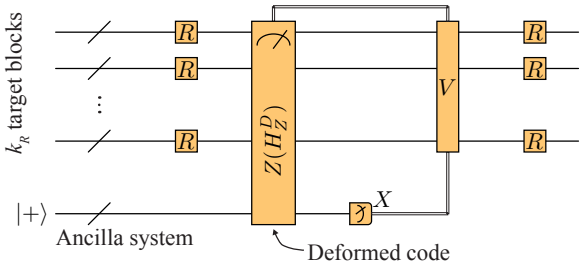

FIG. 3. Circuit for parallelized code surgery. The ancilla system is transversally initialized in the state $|+\rangle$ and measured in the X basis. The R gate rotates the logical measurement basis to Z: for each memory-code block on which the measured Pauli operator acts as X, a transversal Hadamard gate is applied. The central step of code surgery is a single round of parity-check measurements of the deformed code, specifically measuring Z stabilizer operators $Z(H_Z^D)$. Implemented via gate teleportation with a resource state from locally-testable state preparation, this single round suffices to guarantee fault tolerance. The feedback gate V is a Pauli gate.

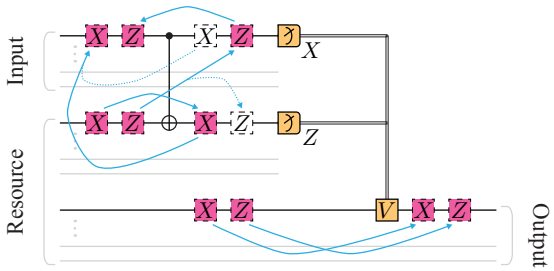

FIG. 4. Gate teleportation. The circuit involves three sets of qubits: the first set carries the input state, while the second and third sets encode the resource state. The diagram illustrates operations on a representative qubit from each set (black lines), with the same operations applied in parallel to the remaining qubits (gray lines). Depending on measurement outcomes, a Pauli gate V is applied on the output qubit. Dashed boxes indicate Pauli errors, and white boxes denote trivial errors. Arrows trace the paths of error propagation through the circuit.

The allocation of qubits in the quantum computer is illustrated in Fig. 1. In addition to data memory-code blocks, the memory includes ancilla memory-code blocks organized into three sectors: one stores S-gate magic states, another supports the implementation of H, S, and controlled-NOT gates, and the third manages slot switching in PCS. For the latter, when logical operations are performed on $q < k_R$ target blocks, ancilla blocks are used to fill the remaining $k_R - q$ slots. Each sector contains M ancilla memory-code blocks. Since the memory code has constant encoding rate, the total number of physical qubits in memory is $O(Mk)$, where Mk is the total number of logical qubits in the data memory-code blocks. Both the resource-state factory and the magic-state factory also require $O(Mk)$ physical qubits. Altogether, the quantum computer contains $O(Mk)$ physical qubits, yielding a constant qubit overhead.

We now turn to the time overhead. Consider compiling a single layer of logical operations acting on disjoint data logical qubits. These operations are serialized into multiple sub-layers so that, in each sub-layer, every data memory-code block performs at most one logical operation. Consequently, the number of operations per sub-layer is at most M. This serialization reduces to a graph coloring problem, and a full layer of operations can be scheduled within $O(k)$ sub-layers; see Ref. [18] and Appendix C. If the time overhead of each sub-layer—given the constant qubit overhead—is T, then the total time overhead is $O(kT)$. Next, we analyze the time overhead

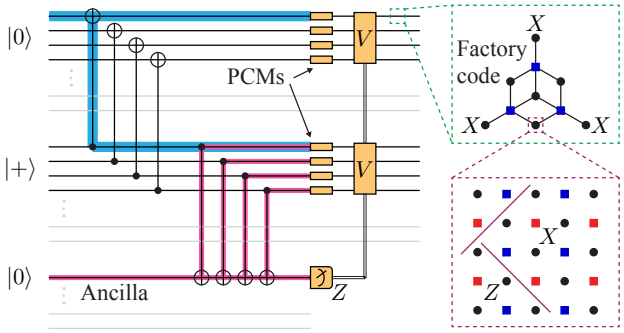

FIG. 5. Circuit for generating resource states used to measure Z stabilizers. The diagram illustrates operations on selected representative qubits (black lines), with similar operations applied in parallel to the remaining qubits (gray lines). The procedure begins by applying transversal controlled-NOT gates between two sets of qubits to prepare Bell states. An ancilla qubit is then used to measure each Z-stabilizer operator of the code, with the illustrated example assuming a weight-four stabilizer for the representative ancilla qubit. Measurement outcomes of the ancilla qubits determine Pauli gates V, which are subsequently applied. Each qubit in this circuit is encoded using a F code, which has only Z-type stabilizers. An example Tanner graph is shown, where blue (red) squares represent Z (X) stabilizers. Note that the depicted graph is from the $[7, 4, 3]$ Hamming code; in practice, the F code should be an LTC with large enough distance. Furthermore, each qubit of the F code is itself encoded in a low-distance surface code, forming a two-level encoding scheme. All operations are performed via transversal gates within both the factory and surface codes. After preparing the encoded resource states, decoding is performed to obtain unencoded resource states. This is achieved by measuring selected qubits in the X and Z bases as illustrated in the figure and applying Pauli gates to the remaining qubits based on the outcomes. The measurement pattern is determined by the standard form of the generator matrices. The entire decoding process can be completed in $O(1)$ time, producing k_F copies of unencoded resource states per circuit run. To correct errors, parity-check measurements (PCMs) of the F code are performed before decoding. Since the F code is an LTC, measurement errors in PCMs are equivalent to low-weight Pauli errors, and a single round of PCMs suffices to reliably detect and correct X errors (including measurement errors on ancilla qubits) by exploiting the propagation of F-code Z stabilizers (highlighted in bold cyan and magenta lines). As a result, each resource state copy has only low-weight residual errors.

of each sub-layer.

The physical circuit depth of all logical operations is $O(1)$, assuming access to sufficient gate-teleportation resource states and distilled magic states. The circuits of logical operations are detailed in Appendix B, where logical measurements are implemented using the PCS circuit shown in Fig. 3, and the parity-check measurements in PCS are realized via the gate-teleportation circuit in Fig. 4. Consequently, the time overhead T is entirely determined by the efficiency of the two state factories.

Gate-teleportation resource states used in logical operations fall into two categories: those for parity checks of the memory code and those for parity checks of deformed codes. In PCS, a deformed code can simultaneously act on k_R code blocks. Accordingly, to implement up to M operations in a sub-layer, the number of required memory-code resource states is $O(M)$, while that of deformed-code resource states is $O(M/k_R)$. In LTSP, each execution of the preparation circuit produces k_F identical copies of a resource state. For memory-code resource states, the preparation circuit requires $O(kk_Fd_S^2)$ physical qubits; for deformed-code resource states, it requires $O(kk_Rk_Fd_S^2)$ qubits. Here, $d_S = \mathrm{polylog}(d)$ is the surface code distance, which will be justified later. In both cases, the circuit depth is $O(d_S)$. Therefore, with $O(Mk)$ physical qubits allocated to the resource-state factory, one can generate all gate-teleportation resource states needed for a sub-layer within time $O(d_S^3)$.

For each sub-layer of logical operations, at most M copies of distilled T-gate magic states are required. The preparation of these states proceeds in two stages. In the first stage, we prepare $O(M)$ copies of noisy magic states on distance-d_S surface-code blocks, which are subsequently transferred to memory-code blocks—each copy is transferred from a surface-code block to the first logical qubit of a memory-code block. The noisy-state preparation circuit requires $O(Mk)$ physical qubits and has a depth of $O(1)$, while the necessary resource states are generated in the resource-state factory with a time overhead of $O(d_S^3)$. In the second stage, we distill the noisy magic states. Using a constant-overhead distillation protocol [18], we can obtain M copies of distilled magic states from the noisy inputs. The distillation stage incurs a time overhead of $O(d^{o(1)}d_S^3)$, where $d^{o(1)}$ originates from the distillation circuit, and the d_S^3 factor arises from the resource-state generation.

Based on the above analysis, the total time overhead for each sub-layer is $O(d^{o(1)}d_S^3)$, resulting in a time overhead of $O(kd^{o(1)}d_S^3) = O(d^{a+o(1)})$ for an entire layer. Note that $d_S = \mathrm{polylog}(d)$. In addition to executing logical operations, the computation also requires the preparation of S-gate magic states and the initialization of data memory-code blocks prior to computation. These tasks can be completed in time $O(kd_S^3)$—the same order as that of an operation layer—given $O(Mk)$ physical qubits. A detailed analysis is provided in Appendix D.

IV. LOW-OVERHEAD CODE SURGERY

In the resource-efficient code surgery protocol, specifically devised sticking, the overall spacetime overhead scales as $O(d^2)$. This overhead arises from two sources. One factor of d comes from the qubit cost of constructing the ancilla system in code surgery. In devised sticking, the ancilla size scales as $O(kd)$ to measure $\Theta(k)$ logical operators simultaneously. The other factor of d reflects the inherent time-efficiency limitation of conven-

Logical operations	Logical measurements	Extra resource states
Initialization		H_X^M
Mea_j	$\bar{Z}_j$	
H_j	$\bar{Z}_j \otimes \bar{Z}_1$, $\bar{X}_j$, $\bar{Z}_j \otimes \bar{X}_1$, $\bar{Z}_1$	H_Z^M
S_j	$\bar{Z}_1 \otimes \bar{Z}_1$, $\bar{Z}_j \otimes \bar{Z}_1 \otimes \bar{X}_1$, $\bar{X}_1$	H_Z^M
T_j	$\bar{Z}_j \otimes \bar{Z}_1$, $\bar{X}_1 \times 2$, $\bar{Z}_1 \otimes \bar{Z}_1$, $\bar{Z}_j \otimes \bar{Z}_1 \otimes \bar{X}_1$	T-gate magic state, H_Z^M
$CNOT_{a,b}$	$\bar{Z}_a \otimes \bar{Z}_1$, $\bar{X}_b \otimes \bar{X}_1$, $\bar{Z}_1$	H_Z^M
Noisy T-gate magic state preparation	$\bar{Z}_S \otimes \bar{Z}_1$	H_X^S, H_Z^S, H_Z^M
Fault-tolerant S-state magic state preparation	$\bar{Z}_C \otimes \bar{Z}_1$	H_X^C, H_Z^C, H_Z^M

TABLE IV. Universal set of logical operations employed in our scheme. Initialization prepares all logical qubits of a memory-code block transversally in the state $|0\rangle$. Mea_j, H_j, S_j, and T_j denote, respectively, the Z-basis measurement, Hadamard gate, S gate, and T gate acting on the jth logical qubit of a memory-code block. Note that Mea_j can be used to reinitialize a logical qubit. $CNOT_{a,b}$ is the controlled-NOT gate with control the ath logical qubit of block A and target the bth logical qubit of block B, where A and B may be the same or different blocks. Each logical operation is implemented through a set of logical measurements and therefore consumes corresponding resource states; the table also lists any additional resource states consumed in addition to those used in the logical measurements. A T gate consumes one copy of the T-gate magic state, whereas an S gate requires one copy of the S-gate magic state as input but does not consume it. In addition to the universal operation set, we also list the magic-state preparation operations.

tional code surgery, which requires $\Theta(d)$ rounds of parity-check measurements to suppress measurement errors. In this work, we develop an approach that eliminates both d-scaling factors, enabling a constant-spacetime-overhead implementation of code surgery. However, it imposes constraints on the operation set, resulting in an overall time overhead of $O(d^{a+o(1)})$ when compiling a general logical circuit.

We address the qubit cost with PCS. In this approach, code surgery is applied simultaneously across multiple memory blocks using a single ancilla system. Consequently, with an ancilla of size $O(k\mathrm{poly}(d))$, we can operate on $\Theta(k\mathrm{poly}(d))$—the same polynomial—logical qubits in parallel, resulting in a qubit overhead that remains constant.

We overcome the inherent time-efficiency limitation of code surgery through the combination of gate teleportation and LTSP. The central challenge is to reduce the time overhead without compromising the constant qubit overhead [15, 28]. To this end, we utilize gate teleportation to perform parity-check measurements, and propose generating the required resource states using a classical LTC. Leveraging the local testability property, we demonstrate that the overhead from state preparation contributes only an almost constant factor to the qubit cost, while reducing the time overhead of code surgery to almost constant.

Although the d^2 factor is eliminated, the use of PCS and LTSP imposes constraints on the compilation of logical circuits. In particular, only an operation set of size $O(1)$ can be applied to each memory block, similar to the CC+GT protocol using state distillation [18]. This limited operation set restricts intra-block parallelization and necessitates serialization of logical operations. As a result, the overall time overhead scales as $O(kd^{o(1)}) = O(d^{a+o(1)})$.

A. Parallelized code surgery

Before presenting our scheme, we first review the qubit overhead in several existing code surgery protocols; see Fig. 6. In the CKBB protocol, the ancilla system is constructed as a hypergraph product code, which is a product of two linear codes, referred to as the glue code and the R code throughout this paper. To measure a logical operator via code surgery, a layer of qubits along the glue-code direction [horizontal direction in Fig. 6(a)] is coupled to the support of the operator to extract its eigenvalue. Consequently, the glue code has size $\Theta(w)$, where w is the weight of the logical operator. The R code is chosen as a repetition code of length $\Theta(d)$ to suppress errors in the ancilla, thereby setting the size of the ancilla in the other direction [vertical direction in Fig. 6(a)]. Together, these yield an overall ancilla size of $O(wd)$ per measured logical operator. In the gauging measurement protocol, the R-code dimension is minimized by using an expander-based construction that controls errors through graph-theoretic properties, reducing the ancilla size to $O(w^{1+o(1)})$ per measured operator. In devised sticking, the ancilla is also constructed as a hypergraph product code. However, the glue-code layer is coupled to multiple logical operators simultaneously. By carefully designing the glue code, the ancilla can be programmed to measure an arbitrary subset of logical operators. When measuring up to k logical operators in parallel, the glue code has size $O(k)$, resulting in an overall ancilla size of $O(kd)$. Thus, the qubit overhead per logical operator is reduced to $O(d)$.

In our approach, we further reduce the qubit overhead

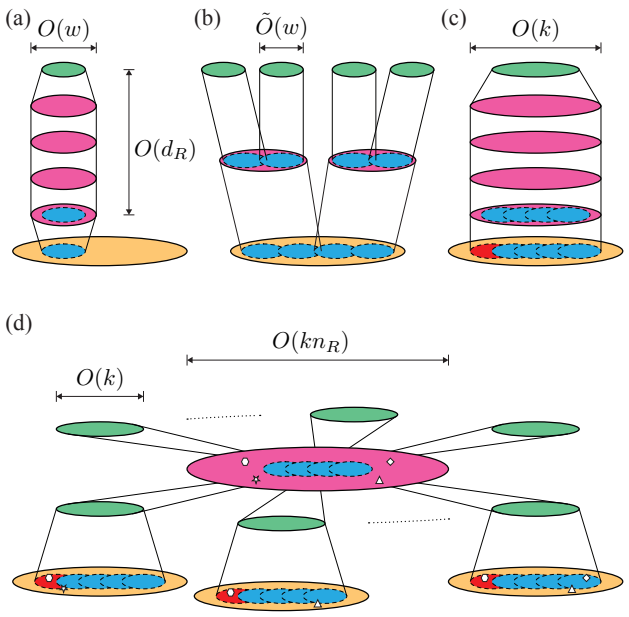

FIG. 6. Code surgery schemes. Each orange circle represents a memory block. Magenta and green circles denote two types of subsystems that constitute the ancilla system. Dashed cyan circles indicate logical operators to be measured. (a) CKBB protocol. The ancilla system comprises multiple subsystems and is coupled to the support of the target logical operator. (b) GM+BFB protocol. The ancilla system consists of two components: branch-sticker subsystems (magenta), which directly couple to memory blocks to isolate logical operators, and gauging-measurement subsystems (green), which couple to the branch stickers to facilitate readout. Although only one layer of branch stickers is shown, measuring q logical operators in parallel requires $O(\log d)$ layers in practice. (c) Devised sticking. The ancilla system is coupled to a subset of the memory block that may overlap with logical operators beyond the targeted ones. By appropriately designing the glue code, any subset of logical operators can be selectively measured, while unmeasured logical operators (dashed red circles) are preserved. (d) Parallelized code surgery. Logical operators on multiple memory blocks are measured simultaneously. The ancilla system comprises two parts: the large magenta circle corresponds to the second columns of H_X^D and H_Z^D in Eqs. (3) and (4); the green circles correspond to the third columns. There are n_R green-circle subsystems in total, with k_R of them coupled to memory blocks. Errors on the magenta subsystem (indicated by white-face markers) are equivalent to errors on the memory blocks.

by coupling a single ancilla system to multiple memory blocks; see Fig. 6(d). Specifically, we select the R code to be a constant-rate LDPC code (see Table II). We require the R code to have distance $d_R = \Omega\left(n_R^{1/a_R}\right)$ for some finite constant a_R, ensuring that its block length satisfies $n_R = \mathrm{poly}(d)$ when $d_R = \Theta(d)$. Each independent codeword of the R code functions as a dedicated channel for reading out the eigenvalues from a corresponding memory block. Consequently, a R code of length n_R enables simultaneous measurement of $\Theta(n_R)$ memory blocks and $\Theta(kn_R)$ logical operators. This construction reduces the eventual qubit overhead per logical operator to $O(1)$.

The idea of coupling an ancilla system to the memory multiple times may have broader applications. It has been discussed as a potential strategy for measuring high-weight logical operators [40]. This technique may also offer a pathway to reducing qubit overhead when measuring multiple operators within a single code block. However, such applications remain underdeveloped due to the absence of a rigorous justification of fault tolerance—specifically, there is no guarantee that the resulting deformed code maintains a well-lower-bounded distance. In contrast, the approach presented in this work provides a systematic framework for constructing the ancilla system and designing its coupling to memory blocks, ensuring a provable code distance while faithfully implementing the desired logical operations.

We now present the scheme of PCS. Following the standard formalism for CSS codes, we represent the target code by the tuple (H_X, H_Z, J_X, J_Z). This corresponds to the code of the target block on which the logical measurement is applied. The target block could be a single memory-code block or a composite of multiple blocks. To simplify notation, we use $[[n, k, d]]$ to denote the parameters of the target code, although this notation has also been used for the memory code. Here, H_X and H_Z (J_X and J_Z) are the check (generator) matrices, representing X and Z stabilizer (logical) operators of the code, respectively. Formal definitions of these matrices and notation can be found in Appendix A 1. These matrices satisfy the standard conditions: $H_X H_Z^{\mathrm{T}} = H_X J_Z^{\mathrm{T}} = H_Z J_X^{\mathrm{T}} = 0$ and $J_X J_Z^{\mathrm{T}} = E_k$, where E_k is the $k \times k$ identity matrix.

In PCS, we perform simultaneous logical measurements on multiple target blocks. On each block, we measure q independent Z logical operators, which can be compactly represented by the matrix αJ_Z. Here, $\alpha \in \mathbb{F}_2^{q\times k}$ is a full-rank matrix, with each row specifying a Z logical operator. Given a R code with logical dimension k_R, the same single-block measurement specified by αJ_Z can be applied in parallel to k_R target blocks. While we focus on single-block measurements of Z logical operators, the generalization to multi-block measurements and X logical operators is straightforward.

To realize the parallelized measurement, we employ a deformed code defined by the following check matrices:

$$H_X^D = \begin{pmatrix} E_{k_R} \otimes H_X & 0 & {G_R^{\mathrm{r}}}^{\mathrm{T}} \otimes T \\ 0 & E_{r_R} \otimes H_G & H_R \otimes E_{r_X} \end{pmatrix} \quad (3)$$

and

$$H_Z^D = \begin{pmatrix} E_{k_R} \otimes H_Z & 0 & 0 \\ G_R^{\mathrm{r}} \otimes S & H_R^{\mathrm{T}} \otimes E_{n_G} & E_{n_R} \otimes H_G^{\mathrm{T}} \end{pmatrix}. \quad (4)$$

Here, r_X is the number of rows in the matrix H_X, and the superscript 'r' denotes the right inverse of a matrix. The matrices H_R, G_R, H_G, S, and T will be explained later.

In each check matrix, the first column corresponds to the k_R target blocks, while the second and third columns represent the ancilla system. In the code surgery procedure, the ancilla-system qubits are initialized in the $|+\rangle$ state. Parity-check measurements associated with the deformed code are then performed, followed by X-basis measurements on the ancilla-system qubits. This protocol implements the desired Z logical measurements across the target blocks.

The matrices H_R and G_R define the R code, where H_R is the check matrix and G_R is the generator matrix. We choose G_R in its standard form, so that its right inverse takes the form ${G_R^{\mathrm{r}}}^{\mathrm{T}} = \begin{pmatrix} E_{k_R} & 0 \end{pmatrix}$, which is sparse. The ancilla system used in the protocol is a hypergraph product code generated from the R code and the glue code, the latter specified by the check matrix H_G. The X and Z check matrices of the hypergraph product code are given by the last two columns in the second row of H_X^D and H_Z^D, respectively.

The matrices H_G, S, and T are constructed following the framework of devised sticking [22], such that they satisfy the following conditions: i) $H_X S^{\mathrm{T}} = T H_G$; ii) there exists a matrix R such that $\alpha J_Z R S = \alpha J_Z$ and $H_G(\alpha J_Z R)^{\mathrm{T}} = 0$; and iii) there exists a matrix β such that $\alpha_\perp J_X S^{\mathrm{T}} = \beta H_G$. Here, $\alpha_\perp \in \mathbb{F}_2^{(k-q)\times k}$ is a full-rank matrix satisfying $\alpha_\perp \alpha^{\mathrm{T}} = 0$. These conditions ensure that the ancilla system is correctly programmed and coupled to the target blocks: the intended logical operators are measured, while the unmeasured operators are preserved throughout the code surgery procedure. Furthermore, the matrices H_G, S, and T are constructed to be sparse: their row and column weights are bounded above by a constant, ensuring that the resulting deformed code remains a qLDPC code. In particular, S is designed to have both row and column weights exactly equal to one, a property that contributes to maintaining a favorable distance in the deformed code. Finally, when the generator matrices J_X and J_Z are expressed in standard form, and assuming that the logical operators selected for measurement act on mutually disjoint sets of logical qubits (i.e. the logical thickness is one), the number of rows and columns in H_G scales as $O(k)$. This ensures that the qubit overhead of code surgery is constant.

We can measure the target logical operators because they belong to the stabilizer group of the deformed code. This is demonstrated by the relation

$$\begin{pmatrix} E_{k_R} \otimes (\alpha J_Z) & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & G_R \otimes (\alpha J_Z R) \end{pmatrix} H_Z^D \quad (5)$$

This equation also specifies how to extract the eigenvalues of the target logical operators from the measurement outcomes of the stabilizers.

Next, we sketch the proof for a lower bound on the distance of the deformed code. The key idea is to use the technique that establishes an equivalence between errors on the ancilla system and errors on the target blocks; see Fig. 6(d). On each target block, the weight of the equivalent error is upper bounded by the weight of the original error on the ancilla system. Consequently, the distance of the deformed code is lower bounded by the distance of the target code. We note, however, that an error on the ancilla system may be equivalent to multiple distinct errors on different target blocks, resulting in error amplification. In the present protocol, such amplification is tolerable because error correction is performed independently on each target block. Finally, we emphasize that the validity of the error equivalence argument assumes that the error weight is below the distance of the R code. Therefore, the R code must have sufficiently large distance to ensure fault tolerance.

Lemma 1. *Suppose the target code is a qLDPC code with constant encoding rate, and the R code is an LDPC code also with constant encoding rate. Then, the resulting deformed code supports PCS with constant qubit overhead. In particular, the deformed code defined by Eqs. (3) and (4) is a qLDPC code whose size—that is, the number of rows and columns in its check matrices—scales as $O(k_R k)$. This code enables the intended operation: the parallel measurement of an identical set of up to k logical operators, specified by αJ_Z, across k_R target blocks. Moreover, if H_R is full-rank, the distance of the deformed code satisfies $d_D \geq \min\{d, d_R\}$.*

See Appendix E for the proof.

B. Locally-testable state preparation

The inherent time-efficiency limitation of code surgery arises from the need to correct measurement errors. In conventional code surgery, a subset of logical operators is measured by coupling the target block to an ancilla system, resulting in a deformed code whose stabilizer group includes the logical operators to be measured. Eigenvalues of these operators are then extracted by performing parity-check measurements of the deformed code, i.e. measuring its stabilizer operators. However, due to the presence of measurement errors, these measurements must be repeated for $\Theta(d)$ rounds to ensure accurate error correction and reliable eigenvalue extraction. Consequently, if parity-check measurements could be performed without measurement errors, this would eliminate the $\Theta(d)$ repetition factor and significantly reduce the overall time overhead.

Parity-check measurements free of measurement errors can be effectively realized through gate teleportation. The key insight is that, in gate teleportation, all errors can be interpreted as acting either on the input or output state, as illustrated in Fig. 4. This leads to the following effective picture: the input state may carry some error, followed by a single round of error-free parity-check measurements, after which additional errors may occur on the output state. Although the measurements are effectively error-free in this picture, it remains crucial to ensure that the effective errors on both the input and output states are low-weight and therefore correctable.

In particular, this requires that the resource state used in gate teleportation contains only low-weight errors.

Errors in the resource state can be managed either by using a high-distance concatenated code or by applying state distillation [15–18]. While high-distance concatenated codes incur substantial spacetime overhead, constant-overhead state distillation requires the quantum memory to be encoded in a good qLTC. Since our objective is a general method of reducing spacetime overhead for qLDPC codes, an alternative approach is necessary. Furthermore, even if the memory were encoded in a good qLTC, constructing a deformed code that also satisfies local testability is a highly nontrivial task, further limiting the practical applicability of constant-overhead state distillation in our context.

Our approach, LTSP, offers a generic method to control measurement errors in qLDPC-code quantum error correction, while requiring only almost constant overhead; see Fig. 5. Central to this approach is generating resource states using a classical LTC, which we refer to as the F code because it serves as the core component of the resource state factory (see Table II). In our scheme, each resource state is dedicated to measuring only Z stabilizer operators—X stabilizer operators can be measured by rotating the basis. When measuring Z stabilizers, we simulate the corresponding measurement circuit by encoding logical qubits in the F code. Specifically, we treat the classical code as a quantum code that protects only Z operators. Then, the measurement circuit is simulated with transversal operations. This simulation produces an encoded resource state, which can be efficiently transformed into the eventual unencoded resource state. Our method reliably produces resource states meeting the requirement that errors remain low-weight, as summarized in Lemma 2. Importantly, this approach is broadly applicable to any qLDPC code.

Lemma 2. *Consider the generation of gate-teleportation resource states for measuring Z stabilizer operators of a deformed code. Suppose the F code is an LDPC code with constant soundness. Let $|\bar{e}_X^{sp}|$ and $|\bar{e}_Z^{sp}|$ denote the weights of X and Z errors occurring during the state preparation circuit, respectively. Let $|e_X^{RS}|$ and $|e_Z^{RS}|$ denote the weights of X and Z errors present on one copy of the prepared resource state. If these errors are undetectable and satisfy $|\bar{e}_X^{sp}| < d_F/C_1$, then the following bounds hold: $|e_X^{RS}| \le C_2|\bar{e}_X^{sp}|$ and $|e_Z^{RS}| \le C_3|\bar{e}_Z^{sp}|$, where C_1, C_2, and C_3 are constants determined by the weight and soundness parameters of the involved codes. The state preparation circuit uses $O(n_D n_F d_S^2)$ physical qubits, has depth $O(d_S)$, and produces n_F copies of the resource state. While d_S can be arbitrary positive integer, in practice it is chosen to scale as* $\mathrm{polylog}(n_D n_F)$. *These conclusions naturally generalize from deformed codes to arbitrary CSS-type qLDPC codes.*

See Appendix F for a detailed explanation of the protocol and the corresponding proof.

Now, we explain why errors on the resource state remain low-weight. Recall that our goal is to measure stabilizer operators of a qLDPC code (the deformed code), and the F code used for simulation is a classical LDPC code. This ensures that the entire circuit has low depth and a low-density structure, so Pauli errors generated directly by circuit operations are inherently low-weight. The challenge arises in the feedback operations: to prepare the resource state within the logical subspace of the deformed code, we apply Pauli gates conditioned on measurement outcomes identifying the subspace sectors. However, these measurements can be faulty. Since these feedback Pauli gates may have large weight, even a low-weight measurement error can propagate into a high-weight Pauli error on the output resource state. We address these measurement errors by simulating the circuit using an LTC (the F code): due to the local testability, measurement errors can be corrected using only a single round of parity-check measurements.

Next, we analyze the resource costs. To reduce the qubit overhead for generating resource states, we propose using a double-constant classical LTC with constant encoding rate and constant soundness [13, 31, 32]. Similar to the R code, we require the F code to have distance $d_F = \Omega\left(n_F^{1/a_F}\right)$ for some finite constant a_F, ensuring that its block length satisfies $n_F = \mathrm{poly}(d)$ when $d_F = \Theta(d)$. Because the encoding rate is constant, we can produce $\Theta(n_F)$ copies of the resource state simultaneously, ensuring constant qubit overhead. Since both the F code and the deformed code are (q)LDPC codes, the circuit depth remains constant.

Finally, we address a critical issue neglected in the above discussion. Because the F code is classical, it can only protect against one type of error, allowing the other type to accumulate. In fact, it cannot correct either error type to the fault-tolerant level up to our circuit design. This error accumulation typically results in a vanishing fault-tolerant threshold. To overcome this, we further encode each physical qubit of the F code into a logical qubit of a low-distance surface code (or any other low-distance code supporting a certain logical operation set). The measurement circuit has size $O(n_D)$, where n_D is the length of the deformed code. When simulating this circuit with the F code, the circuit size becomes $O(n_D n_F)$. To control error accumulation, it suffices to choose a surface code distance $d_S = \mathrm{polylog}(n_D n_F)$. Incorporating the surface code increases the circuit size to $O(n_D n_F \mathrm{poly}(d_S))$, thus contributing only a polylogarithmic overhead factor to the spacetime cost.

V. CONCLUSIONS

We have presented a general scheme for FTQC on qLDPC codes that combines code surgery with gate teleportation and introduces optimizations in deformed-code construction and resource-state preparation. Our approach achieves constant qubit overhead while signifi-

cantly reducing time overhead. For all constant-rate qLDPC codes, the time overhead is lower than that of the GM+BFB protocol when the code distance grows faster than HGP codes; for good qLDPC codes, it reaches the $O(d^{1+o(1)})$ scaling. These results improve the asymptotic resource efficiency of FTQC and provide a general framework for lowering overhead in both theoretical and near-term implementations.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant Nos. 12225507, 12088101) and NSAF (Grant No. U1930403).

Appendix A: Preliminaries

1. Notations

We use $|\bullet|$ and $\|\bullet\|$ to denote the Hamming weigh and the matrix norm induced by Hamming weight, respectively, i.e. the norm of a matrix $A \in \mathbb{F}_2^{m\times n}$ is

$$\|A\| = \max\left\{\frac{|uA|}{|u|} \ : \ u \in \mathbb{F}_2^m - \{0\}\right\}. \tag{A1}$$

We denote by $A_{j,\bullet}$ and $A_{\bullet,j}$ the j-th row and j-th column of the matrix A, respectively, and we denote by A^{r} the right inverse.

Let $b = (b_1, b_2, \ldots, b_n) \in \mathbb{F}_2^n$. We use

$$\sigma(b) \ = \ \sigma^{b_1} \otimes \sigma^{b_2} \otimes \cdots \otimes \sigma^{b_n} \tag{A2}$$

to represent n-qubit $\sigma = X, Z$ Pauli operators. When $B \in \mathbb{F}_2^{m\times n}$,

$$\sigma(B) \ = \ \begin{pmatrix} \sigma(B_{1,\bullet}) \\ \sigma(B_{2,\bullet}) \\ \vdots \\ \sigma(B_{m,\bullet}) \end{pmatrix} \tag{A3}$$

is a list of Pauli operators.

An $[[n,k,d]]$ CSS code is represented by four matrices $H_X \in \mathbb{F}_2^{r_X\times n}$, $H_Z \in \mathbb{F}_2^{r_Z\times n}$ and $J_X, J_Z \in \mathbb{F}_2^{k\times n}$ that satisfy conditions $H_X H_Z^{\mathrm{T}} = H_X J_Z^{\mathrm{T}} = J_X H_Z^{\mathrm{T}} = 0$, $J_X J_Z^{\mathrm{T}} = E_k$. The stabilizer of the code is generated by operators $X(H_X)$ and $Z(H_Z)$. The X and Z logical operators of the j-th logical qubit are $X((J_X)_{j,\bullet})$ and $Z((J_Z)_{j,\bullet})$, respectively.

We use $(H_X^c, H_Z^c, J_X^c, J_Z^c)$ with $c = M, D, S, C$ to represent the memory, deformed, surface, and color codes, respectively. Regarding parameters of a quantum code, we use notations n_c, k_c, d_c, r_X^c, and r_Z^c. Check and generator matrices of the R and F codes are $H_c \in \mathbb{F}_2^{r_c\times n_c}$ and $G_c \in \mathbb{F}_2^{k_c\times n_c}$ with $c = R, F$, respectively, which satisfy $H_c G_c^{\mathrm{T}} = 0$; and we also use d_c to denote the distance of a classical linear code.

2. Primitive operations, spacetime locations, and error models

The primitive operations are illustrated in Fig. 7. Given a quantum circuit, there are two sets of spacetime locations. Quantum locations are the qubits during time intervals between primitive operations; see Figs. 11, 13, and 14 for examples, in which red boxes represent the quantum locations. Classical locations are outcomes of projective measurements: In the measurement of $\sigma(A)$, the outcome is a vector $\nu \in \mathbb{F}_2^m$, and $(-1)^{\nu_j}$ is the eigenvalue of $\sigma\left(A_{j,\bullet}\right)$, i.e. each element corresponds to an operator; such a measurement introduces m locations, and each location corresponds to one element in the outcome.

We assume that errors can occur independently at any location. We now justify the error model. For initializations and gates with errors, each of them can be modeled as an ideal operation with errors occurring at the locations immediately after the operation. For qubit measurements with errors, they can be modeled as ideal measurements with errors occurring at locations immediately before the measurements: these errors effectively flip measurement outcomes. For a projective measurement with errors, it can be modeled as a measurement that projects the state

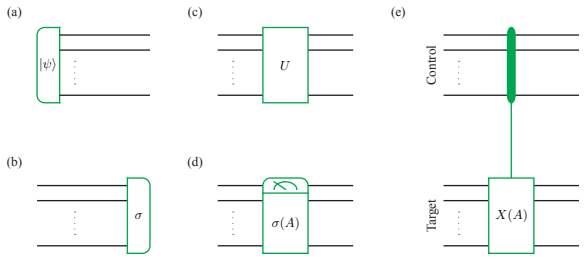

FIG. 7. (a) Transversal initialization that prepares each qubit in the state $|\psi\rangle$. (b) Transversal measurement that measures each qubit in the basis $\sigma = X, Z$. (c) Gate U. Here, U is a layer of single-qubit gates, i.e. $U = U_1 \otimes U_2 \otimes \cdots$, where $U_1, U_2, \ldots$ are single-qubit gates on qubits $1, 2, \ldots$, respectively. (d) Projective measurement on n qubits that measures Pauli operators $\sigma(A)$, where $A \in \mathbb{F}_2^{m\times n}$. (e) Generalized transversal controlled-NOT gate. The control (target) is a set of m (n) qubits. The transformation $X(A_{j,\bullet})$ is (not) applied on the target when the j-th control qubit is in the state $|1\rangle$ ($|0\rangle$). We assume that the matrix A has rows and columns of weight at most ω_{max}, where ω_{max} is a positive constant.

onto the correct subspace with errors occurring in the measurement outcome and at locations immediately after the measurement: the errors at locations after the measurement may induce an effective incorrect projection. Compared with taking spacetime locations in circuits that projective measurements and generalized transversal controlled-NOT gates are decomposed into standard single-qubit and two-qubit operations (see Lemma 3), our choice of spacetime locations could overestimate effective circuit distances and the fault-tolerance threshold by a constant factor, which does not change our conclusions.

Lemma 3. *Projective measurements and generalized transversal controlled-NOT gates specified by a matrix A can be implemented using a circuit composed of single-qubit and two-qubit operations. If the row and column weights of A are bounded above by a positive constant ω_{max}, then the circuit has depth at most $\omega_{max} + 2$. Moreover, an error occurring at a single location in this circuit may propagate to at most $\omega_{\max}$ locations.*

Proof. We only need to consider generalized transversal controlled-NOT gates because projective measurements can be implemented using them. Specifically, to measure the operators $X(A)$, we initialize the control qubits in the $|+\rangle$ state, apply the generalized transversal controlled-NOT gate defined by the matrix A, and then measure the control qubits in the X basis. Similarly, to measure $Z(A^{\mathrm{T}})$, we initialize the target qubits in the $|0\rangle$ state, apply the same gate, and then measure the target qubits in the Z basis.

A generalized transversal controlled-NOT gate can be implemented by a circuit of depth at most $\omega_{\max}$, since scheduling the controlled-NOT gates corresponds to an edge-coloring problem on a bipartite graph [41, 42].

Each qubit in the circuit is involved in at most $\omega_{\max}$ controlled-NOT gates. On a control (target) qubit, an X (Z) error can propagate to the corresponding target (control) qubits, while Z (X) errors do not propagate. Therefore, a single-qubit error may only spread to qubits within the corresponding support set. As a result, the weight of the propagated error is upper bounded by $\omega_{\max}$. □

3. Locally testable codes

Definition 1. (ω, s)-locally testable codes [31]. A linear code $C \subset \mathbb{F}_2^n$ is called (ω, s)-locally testable if it has a parity-check matrix $H \in \mathbb{F}_2^{r\times n}$ with rows of weight at most ω such that for any vector $u \in \mathbb{F}_2^n$ we have

$$\frac{1}{r}|Hu^{\mathrm{T}}| \;\geq\; \frac{s}{n}\min_{c\in C}|u-c|. \tag{A4}$$

The parameters ω and s are positive real numbers called the locality and soundness, respectively.

Lemma 4. *If $H \in \mathbb{F}_2^{r\times n}$ is the check matrix of an (ω, s)-locally testable code, for every vector $v^{\mathrm{T}} \in \mathrm{colsp}H$, there exists a vector $u_\star \in \mathbb{F}_2^n$ such that $v^{\mathrm{T}} = Hu_\star^{\mathrm{T}}$ and*

$$|u_\star| \;\leq\; \frac{n}{rs}|v|. \tag{A5}$$

Proof. Because $v^{\mathrm{T}} \in \mathrm{colsp}H$, there exists $w \in \mathbb{F}_2^n$ such that $v^{\mathrm{T}} = Hw^{\mathrm{T}}$. Then, there exists $c_\star \in \ker H$ such that

$$\frac{1}{r}|v| \;\geq\; \frac{s}{n}|w - c_\star|. \tag{A6}$$

The lemma is proved by taking $u_\star = w - c_\star$. □

Appendix B: Logical operations

The circuits implementing logical operations are shown in Fig. 8, while the circuits for magic-state preparation are shown in Fig. 9.

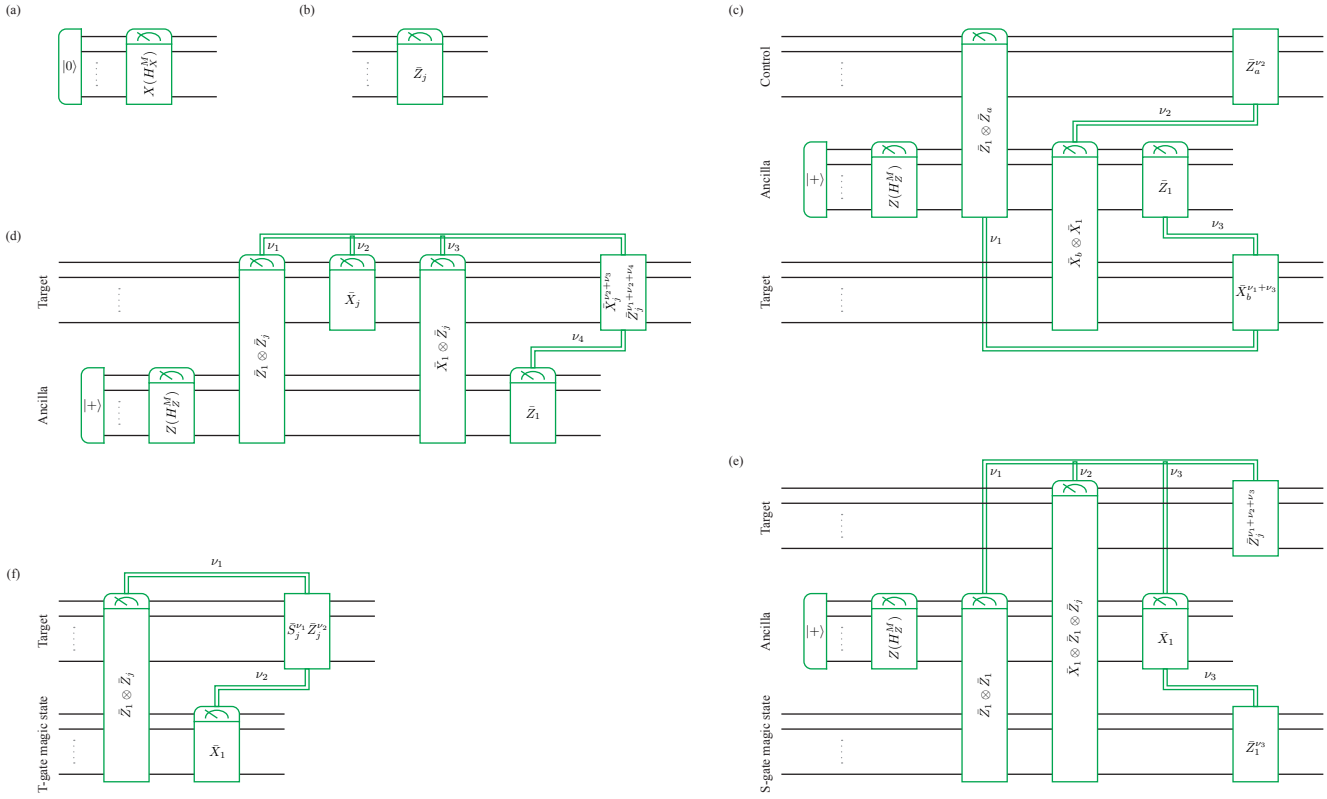

FIG. 8. Logical operations. (a) Initialization. (b) Measurement Mea_j. (c) Controlled-NOT gate $CNOT_{a,b}$. (d) Hadamard gate H_j. (e) S gate S_j. (f) T gate T_j. Each group of horizontal lines represents a memory-code block. Single-qubit gates are applied to the target memory-code block. Each of the gates $CNOT_{a,b}$, H_j, and S_j requires an ancilla memory-code block initialized in the $|+\rangle$ state. Magic states are stored in the first logical qubit of their respective memory-code blocks.

Appendix C: Serialization

We use a pair (B, F) to denote a logical operation: the operation $F \in \{Mea_j, H_j, S_j, T_j, CNOT_{a,b}\}$ applied on the block(s) B. Specifically, if $F \in \{Mea_j, H_j, S_j, T_j\}$ is applied on the u-th block, then $B = u$; if $F = CNOT_{a,b}$ is applied on the a-th logical qubit in the u-th block and the b-th logical qubit in the v-th block, then $B = (u, v)$.

We define the *qubit support* of the operation (B, F) as

$$\mathrm{supp}(B, F) = \begin{cases} \{(u, j)\}, & \text{if } B = u,\ F \in \{Mea_j, H_j, S_j, T_j\}, \\ \{(u, a), (v, b)\}, & \text{if } B = (u, v),\ F = CNOT_{a,b}. \end{cases} \tag{C1}$$

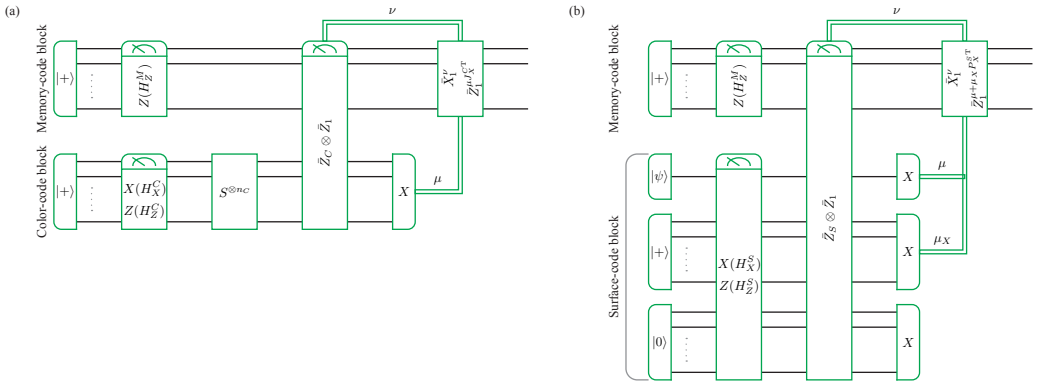

FIG. 9. (a) Preparation of the S-gate magic state: a fault-tolerant copy is prepared on the first logical qubit of the memory-code block. (b) Preparation of the T-gate magic state: a noisy copy is prepared on the first logical qubit of the memory-code block.

Similarly, we define the *block support* of (B, F) as

$$\mathrm{supp}(B) = \begin{cases} \{u\}, & \text{if } B = u, \\ \{u, v\}, & \text{if } B = (u, v). \end{cases} \tag{C2}$$

Consider an operation set $OS = \{(B_l, F_l) \mid l = 1, 2, \ldots, L\}$. We say that OS is *qubit-disjoint* if and only if the qubit supports are disjoint, i.e.,

$$\mathrm{supp}(B_{l_1}, F_{l_1}) \cap \mathrm{supp}(B_{l_2}, F_{l_2}) = \emptyset \quad \forall\, l_1 \neq l_2. \tag{C3}$$

Similarly, OS is *block-disjoint* if and only if the block supports are disjoint, i.e.,

$$\mathrm{supp}(B_{l_1}) \cap \mathrm{supp}(B_{l_2}) = \emptyset \quad \forall\, l_1 \neq l_2. \tag{C4}$$

Lemma 5. *For any qubit-disjoint operation set $OS = \{(B_l, F_l) : l = 1, 2, \ldots, L\}$ (an operation layer) acting on code blocks, each encoding k logical qubits, there exists a partition P of OS such that $|P| = O(k)$, and every element of P (a sub-layer) is block-disjoint.*

Proof. We construct the partition as follows (see also Ref. [17, 18]).

First, we map the operation set OS to a multigraph with loops $G = (V, E, r)$. Suppose the operation set acts on M code blocks; then the vertex set is $V = \{1, 2, \ldots, M\}$. The edge set $E = \{1, 2, \ldots, L\}$ has the same cardinality as the operation set, and $r(l) = \mathrm{supp}(B_l)$ specifies the endpoint vertices of the l-th edge.

Next, we find a proper edge coloring of G, i.e., a mapping $f : E \to C$, where C is the set of colors, such that any two edges incident to the same vertex receive different colors. Since OS is qubit-disjoint, the maximum degree of G is k. Then, the minimum number of colors required for the coloring, the chromatic index of G, is $O(k)$ [18, 43].

Next, we find a proper edge coloring of G, that is, a mapping $f : E \to C$, where C is the set of colors, such that any two edges incident to the same vertex are assigned different colors. Because OS is qubit-disjoint, the maximum degree of G is k. Hence, the minimum number of colors required for such a coloring—the chromatic index of G—is $O(k)$ [18].

Finally, we construct the desired partition as $P = \{U_c : c \in C\}$, where $U_c = \{(B_l, F_l) : f(l) = c\}$. Each U_c is block-disjoint by construction, and $|P| = O(k)$, completing the proof. □

Appendix D: Detailed cost analysis

In this section, we analyze the time overhead under the constraint of a constant qubit overhead. We assume that the number of logical qubits k is small compared to M. Specifically, we assume that the memory code distance scales as $d = O(\mathrm{polylog}(Mk))$, which determines both the block length n and the corresponding logical qubit number k. This distance is sufficient to achieve a logical error rate per operation p_L satisfying $1/p_L = O(\mathrm{poly}(Mk))$. If the logical circuit size is polynomial, i.e. $|C| = O(\mathrm{poly}(Mk))$, then this choice of code distance is also sufficient to achieve any target total logical error rate ϵ satisfying $1/\epsilon = O(\mathrm{poly}(Mk))$.

Lemma 6. *Suppose $d, d_R, d_F = O(\mathrm{polylog}(Mk))$. Assuming a sufficient supply of distilled magic states, any block-disjoint operation set (sub-layer) OS can be implemented in time $O(d_S^3)$, requiring $O(Mk)$ physical qubits in the resource-state factory.*

Proof. Let $\mathrm{num}(F)$ denote the number of occurrences of operation $F \in Mea_j, H_j, S_j, T_j, CNOT_{a,b}$ in OS. Because OS is block-disjoint, $\sum_F \mathrm{num}(F) = |OS| \le M$. We define

$$\mathrm{batch}(F) = \left\lceil \frac{1}{d_S^2} \left\lceil \frac{1}{k_F} \left\lceil \frac{1}{k_R} \mathrm{num}(F) \right\rceil \right\rceil \right\rceil. \tag{D1}$$

Since

$$\mathrm{batch}(F) \le \frac{1}{d_S^2} \left[\frac{1}{k_F} \left(\frac{1}{k_R} \mathrm{num}(F) + 1 \right) + 1 \right] + 1, \tag{D2}$$

we obtain

$$\sum_F \mathrm{batch}(F) \le \frac{M + \Theta(k^2)(k_R + k_R k_F + k_R k_F d_S^2)}{k_R k_F d_S^2}, \tag{D3}$$

where we have used that $|Mea_j, H_j, S_j, T_j, CNOT_{a,b}| = \Theta(k^2)$.

The operation set OS is implemented in d_S^2 steps, where each step contains at most $\mathrm{batch}(F) k_R k_F$ copies of operation F acting on different code blocks.

To illustrate, consider $F = Mea_j$. This operation requires two types of resource states: $H_Z^D(\bar{Z}_j)$ and H_X^M. For each step, at most $\mathrm{batch}(F) k_F$ copies of $H_Z^D(\bar{Z}_j)$ are needed. These can be generated by running $\mathrm{batch}(F)$ copies of the corresponding LTSP circuit in parallel, requiring at most $\mathrm{batch}(F) \times O(n k_R k_F d_S^2)$ physical qubits. Similarly, up to $\mathrm{batch}(F) k_R k_F$ copies of H_X^M are needed, which can be produced by running $\mathrm{batch}(F) k_R$ copies of the corresponding LTSP circuit in parallel, requiring at most $\mathrm{batch}(F) k_R \times O(n k_F d_S^2)$ physical qubits. Therefore, the total number of physical qubits required for operation F is $\mathrm{batch}(F) O(n k_R k_F d_S^2)$.

The same scaling applies to all operations, yielding

$$N_{RSF,1} \le \sum_F \mathrm{batch}(F) O(n k_R k_F d_S^2) \tag{D4}$$

$$\le O(Mk) + O(k^3 k_R k_F d_S^2). \tag{D5}$$

Assuming $d, d_R, d_F = O(\mathrm{polylog}(Mk))$, we have $k^3 k_R k_F d_S^2 = O(\mathrm{polylog}(Mk))$, under assumptions on code rate and distance. Consequently, $N_{RSF,1} = O(Mk)$, i.e., the total number of physical qubits required in the resource-state factory to implement a block-disjoint operation set scales linearly with the logical qubit count.

Each step involves running LTSP circuits in parallel, which requires a time overhead of $O(d_S)$. Since there are d_S^2 steps, the total time overhead for implementing a block-disjoint operation set is $O(d_S^3)$. □

Lemma 7. *Suppose $d, d_R, d_F = O(\mathrm{polylog}(Mk))$ and $d_S = O(\mathrm{polylog}(d))$. Then M copies of distilled T-gate magic states can be generated in time $O(d^{o(1)} d_S^3)$, requiring $O(Mk)$ physical qubits in the resource-state and magic-state factories.*

Proof. The generation of distilled magic states consists of two stages. First, we prepare noisy magic states, and then we distill them.

The circuit for noisy magic-state preparation is shown in Fig. 9(b). We begin by preparing a noisy magic state on a surface-code block, which is subsequently transferred to the first logical qubit of a memory-code block. Using the constant spacetime-overhead distillation protocol, to obtain M copies of distilled magic states, we require $O(M)$ copies of noisy magic states [39]. Consequently, $O(M)$ surface-code blocks and $O(M)$ memory-code blocks are needed to store these states, corresponding to $O(M d_S^2)$ and $O(Mk)$ physical qubits in the magic-state factory, respectively.

The $M' = O(M)$ copies of noisy magic states are prepared over d_S^2 steps, where each step produces at most $m' k_R k_F$ copies, with

$$m' = \left\lceil \frac{1}{d_S^2} \left\lceil \frac{1}{k_F} \left\lceil \frac{1}{k_R} M' \right\rceil \right\rceil \right\rceil. \tag{D6}$$

We require four types of gate-teleportation resource states: H_Z^M, H_X^S, H_Z^S, and $H_Z^D(\bar{Z}_S \otimes \bar{Z}_1)$. In each step, up to $m' k_R k_F$ ($m' k_R k_F$, $m' k_R k_F$, and $m' k_F$) copies of H_Z^M (H_X^S, H_Z^S, and $H_Z^D(\bar{Z}_S \otimes \bar{Z}_1)$) are required. These resource

states can be generated by running $m'k_R$ ($m'k_R$, $m'k_R$, and m') copies of the corresponding LTSP circuits in parallel, which demand at most $m'k_R \times O(nk_F d_S^2)$ ($m'k_R \times O(d_S^2 k_F d_S^2)$, $m'k_R \times O(d_S^2 k_F d_S^2)$, and $m' \times O(nk_R k_F d_S^2)$) physical qubits. Since $d_S = O(\text{polylog}(n))$, the total number of physical qubits is $O(m'nk_R k_F d_S^2) = O(M'n) = O(Mk)$. The time cost of each step is $O(d_S)$, therefore, the overall time overhead of preparing noisy magic states is $O(d_S^3)$.

To prepare M copies, the distillation circuit requires $O(M)$ logical qubits, has a depth of $O(\text{polylog}(d))$, and is repeated $O(d^{o(1)})$ times [18]. We propose to utilize only the first logical qubit of each memory-code block for distillation. Consequently, $O(M)$ memory-code blocks are required in the magic-state factory, corresponding to $O(Mk)$ physical qubits.

Each layer of the distillation circuit constitutes a block-disjoint operation set, as operations are restricted to the first logical qubit of each block. Therefore, the entire circuit can be executed in time $O(d^{o(1)} d_S^3)$, requiring $O(Mk)$ physical qubits in the resource-state factory (Lemma 6).

Note that the implementation of the distillation circuits also requires ancillary memory-code blocks; however, this contributes only a constant factor to the overall qubit overhead. □

Lemma 8. *Suppose $d, d_R, d_F = O(\text{polylog}(Mk))$, $d_C = O(d)$, and $d_S = O(\text{polylog}(d))$. The quantum computer can be initialized in time $O(kd_S^3)$, requiring $O(Mk)$ physical qubits.*

Proof. There are two tasks required to initialize the quantum computer: preparing M copies of high-fidelity S-gate magic states and storing them in M memory-code blocks—each stored in the first logical qubit of a block; and initializing the Mk logical qubits in the data memory-code blocks in the state $|0\rangle$.

As shown in Fig. 9(a), a high-fidelity S-gate magic state is first prepared on a color-code block, which is subsequently transferred to the first logical qubit of a memory-code block. We prepare the M copies in k steps, with at most $M'' = \lceil M/k \rceil$ copies prepared in each step. Thus, we require M'' color-code blocks, corresponding to at most $M'' \times O(d_C^2) = O(Mk)$ physical qubits, noting that $d_C = O(k)$.

As shown in Fig. 9(a), a high-fidelity S-gate magic state is first prepared on a color code block, which is subsequently transferred to the first logical qubit of a memory-code block. We prepare the M copies in k steps with at most $M'' = \lceil M/k \rceil\}$ copies prepared in each step; then, we need M'' color-code blocks, i.e. at most $M'' \times O(d_C^2) = O(Mk)$ physical qubits. Note that $d_C = O(k)$.

In each step, we prepare the M'' copies over d_S^2 sub-steps, where each sub-step produces at most $m''k_R k_F$ copies, with

$$m'' = \left\lceil \frac{1}{d_S^2} \left\lceil \frac{1}{k_F} \left\lceil \frac{1}{k_R} M'' \right\rceil \right\rceil \right\rceil. \tag{D7}$$

Similar to the preparation of noisy T-gate magic states, we require four types of gate-teleportation resource states: H_Z^M, H_X^C, H_Z^C, and $H_Z^D(\bar{Z}_C \otimes \bar{Z}_1)$. In each step, up to $m''k_R k_F$ ($m''k_R k_F$, $m''k_R k_F$, and $m''k_F$) copies of H_Z^M (H_X^C, H_Z^C, and $H_Z^D(\bar{Z}_C \otimes \bar{Z}_1)$) are required. These resource states can be generated by running $m''k_R$ ($m''k_R$, $m''k_R$, and m'') copies of the corresponding LTSP circuits in parallel, requiring at most $m''k_R \times O(nk_F d_S^2)$ ($m''k_R \times O(d_C^2 k_F d_S^2)$, $m''k_R \times O(d_C^2 k_F d_S^2)$, and $m'' \times O((n + d_C^2)k_R k_F d_S^2)$) physical qubits. Since $d_S = O(\text{polylog}(n))$, the total number of physical qubits is $O(m''n^2 k_R k_F d_S^2) = O(M''n^2) = O(Mk)$, noting that $d_C = O(n)$.

According to the above analysis, the overall time overhead for preparing high-fidelity S-gate magic states is $O(kd_S^3)$.

The initialization of the data memory-code blocks is similar to a block-disjoint operation set and can therefore be accomplished in time $O(d_S^3)$ with $O(Mk)$ physical qubits. □

Appendix E: Parallelized code surgery - Proof of Lemma 1

Lemma 9. *[25] Consider a CSS code (H_X, H_Z, J_X, J_Z). Let $Z(\alpha J_Z)$ be the set of logical operators to be measured, where $\alpha \in \mathbb{F}_2^{q \times k}$ is a full-rank matrix, and $q \leq k$. We can realize the measurement using a deformed code, which is a CSS code with check matrices*

$$H_X^D = \begin{pmatrix} H_X & T \\ 0 & H_M \end{pmatrix} \tag{E1}$$

and

$$H_Z^D = \begin{pmatrix} H_Z & 0 \\ S & H_G^{\mathrm{T}} \end{pmatrix}, \tag{E2}$$

where $S \in \mathbb{F}_2^{n_G \times n}$, $T \in \mathbb{F}_2^{r_X \times r_G}$, $H_G \in \mathbb{F}_2^{r_G \times n_G}$ and $H_G \in \mathbb{F}_2^{r_M \times r_G}$ satisfy the following conditions:

i) $H_X S^{\mathrm{T}} = T H_G$;

ii) *There exists a matrix* $R \in \mathbb{F}_2^{n\times n_G}$ *such that* $\alpha J_Z R S = \alpha J_Z$ *and* $H_G(\alpha J_Z R)^{\mathrm{T}} = 0$;

iii) *There exists a matrix* $\beta \in \mathbb{F}_2^{(k-q)\times r_G}$ *such that* $\alpha_\perp J_X S^{\mathrm{T}} = \beta H_G$, *where* $\alpha_\perp \in \mathbb{F}_2^{(k-q)\times k}$ *is a full-rank matrix satisfying* $\alpha_\perp \alpha^{\mathrm{T}} = 0$; *and*

iv) $H_M H_G = 0$.

The unmeasured X *and* Z *logical operators are* $X(\alpha_\perp J_X)$ *and* $Z({\alpha_\perp^{\mathrm{r}}}^{\mathrm{T}} J_Z)$, *respectively.*

Lemma 10. *[25] Given an arbitrary set of* Z *logical operators* $Z(\alpha J_Z)$ *to be measured, we can generate matrices* S, T, *and* H_G *such that code surgery conditions i), ii), and iii) are satisfied, following the devised sticking approach. Furthermore, row and column weights of the matrices* H_G, S, *and* T *are bounded above by a constant, and* $\|S\| = 1$. *When the generator matrices* J_X *and* J_Z *are expressed in standard form, and assuming that the logical operators selected for measurement act on mutually disjoint sets of logical qubits (i.e. the logical thickness is one), the number of rows and columns in* H_G *scales as* $O(k)$.

According to Lemma 10, the check matrices H_X^D and H_Z^D given by Eqs. (3) and (4) has the following properties:

i) Row and column weights are bounded above by a constant, i.e. the deformed code is a qLDPC code;

ii) The number of rows and columns scales as $O(k_R k)$.

1. Logical measurements associated with the deformed codes

Now, we verify that the deformed code given by Eqs. (3) and (4) is valid for realizing the desired measurement. Let's define matrices

$$\tilde{H}_X = E_{k_R} \otimes H_X, \tag{E3}$$
$$\tilde{H}_Z = E_{k_R} \otimes H_Z, \tag{E4}$$
$$\tilde{J}_X = E_{k_R} \otimes J_X, \tag{E5}$$
$$\tilde{J}_Z = E_{k_R} \otimes J_Z, \tag{E6}$$
$$\tilde{\alpha} = E_{k_R} \otimes \alpha, \tag{E7}$$
$$\tilde{\alpha}_\perp = E_{k_R} \otimes \alpha_\perp, \tag{E8}$$
$$\tag{E9}$$

and

$$\tilde{S} = G_R^{\mathrm{r}} \otimes S, \tag{E10}$$
$$\tilde{T} = \begin{pmatrix} 0 & {G_R^{\mathrm{r}}}^{\mathrm{T}} \otimes T \end{pmatrix}, \tag{E11}$$
$$\tilde{H}_G = \begin{pmatrix} H_R \otimes E_{n_G} \\ E_{n_R} \otimes H_G \end{pmatrix}, \tag{E12}$$
$$\tilde{H}_M = \begin{pmatrix} E_{r_R} \otimes H_G & H_R \otimes E_{r_X} \end{pmatrix}, \tag{E13}$$
$$\tilde{R} = G_R \otimes R, \tag{E14}$$
$$\tilde{\beta} = \begin{pmatrix} 0 & {G_R^{\mathrm{r}}}^{\mathrm{T}} \otimes \beta \end{pmatrix}. \tag{E15}$$

Then, we can find that check matrices in Eqs. (3) and (4) are consistent with Eqs. (E1) and (E2), and the matrices satisfy the code surgery conditions i), ii), iii), and iv), i.e.

$$\tilde{H}_X \tilde{S}^{\mathrm{T}} = \tilde{T} \tilde{H}_G, \tag{E16}$$
$$\tilde{\alpha} \tilde{J}_Z \tilde{R} \tilde{S} = \tilde{\alpha} \tilde{J}_Z, \tag{E17}$$
$$\tilde{H}_G (\tilde{\alpha} \tilde{J}_Z \tilde{R})^{\mathrm{T}} = 0, \tag{E18}$$
$$\tilde{\alpha}_\perp \tilde{J}_X \tilde{S}^{\mathrm{T}} = \tilde{\beta} \tilde{H}_G, \tag{E19}$$
$$\tilde{H}_M \tilde{H}_G = 0. \tag{E20}$$

Therefore, according to Lemma 9, with this deformed code, we can simultaneously measure logical operators $Z(\tilde{\alpha} \tilde{J}_Z)$.

2. Distances of deformed codes

Without loss of generality, we consider the R-code generator matrix in the form

$$G_R \;=\; \begin{pmatrix} E_{k_R} & P_R \end{pmatrix}, \tag{E21}$$

and then

$$G_R^{\mathrm{r}\,\mathrm{T}} \;=\; \begin{pmatrix} E_{k_R} & 0 \end{pmatrix}. \tag{E22}$$

Lemma 11. *Suppose H_R is full-rank. For every vector $v \in \mathbb{F}_2^{r_R n_G + n_R r_X}$ satisfying $\tilde{H}_M v^{\mathrm{T}} = 0$ and $|v| < d_R$, there exists a vector $u_\star \in \mathbb{F}_2^{k_R n_G}$ such that $v^{\mathrm{T}} = \tilde{H}_G u_\star^{\mathrm{T}}$ and*

$$|u_\star[(G_R^{\mathrm{r}})_{\bullet,l} \otimes S]| \le \|S\| \times |v| \tag{E23}$$

for all $l = 1, 2, \ldots, k_R$.

Proof. Let $v = \begin{pmatrix} a & b \end{pmatrix}$. Then, the condition $\tilde{H}_M v^{\mathrm{T}} = 0$ becomes

$$(E_{r_R} \otimes H_G) a^{\mathrm{T}} \;=\; (H_R \otimes E_{r_X}) b^{\mathrm{T}}. \tag{E24}$$

Let's introduce matrices A, B, and U such that

$$\mathrm{vec}(A) \;=\; a^{\mathrm{T}}, \tag{E25}$$
$$\mathrm{vec}(B) \;=\; b^{\mathrm{T}}, \tag{E26}$$
$$\mathrm{vec}(U) \;=\; u_\star^{\mathrm{T}}. \tag{E27}$$

We can rewrite the condition as

$$H_G A \;=\; B H_R^{\mathrm{T}}. \tag{E28}$$

Similarly, we can rewrite $v^{\mathrm{T}} = \tilde{H}_G u_\star^{\mathrm{T}}$ as

$$A \;=\; U H_R^{\mathrm{T}}, \tag{E29}$$
$$B \;=\; H_G U. \tag{E30}$$

Since H_R is full-rank, there exists $H_R^{\mathrm{r}} \in \mathbb{F}_2^{n_R \times r_R}$ such that $H_R H_R^{\mathrm{r}} = E_{r_R}$: We remark that $n_R > r_R$ when H_R is full-rank and k_R is not zero. Then, we can take

$$U \;=\; A H_R^{\mathrm{r}\,\mathrm{T}} \tag{E31}$$

to satisfy Eq. (E29). Such a matrix U also satisfies the inequality (E23). Note that

$$[(G_R^{\mathrm{r}})_{\bullet,l} \otimes S]^{\mathrm{T}} u_\star^{\mathrm{T}} \;=\; \mathrm{vec}\left(S^{\mathrm{T}} U (G_R^{\mathrm{r}})_{\bullet,l}\right) = (S^{\mathrm{T}} U)_{\bullet,l} = S^{\mathrm{T}} U_{\bullet,l} = S^{\mathrm{T}} A (H_R^{\mathrm{r}\,\mathrm{T}})_{\bullet,l}. \tag{E32}$$

Therefore,

$$\left|[(G_R^{\mathrm{r}})_{\bullet,l} \otimes S]^{\mathrm{T}} u_\star^{\mathrm{T}}\right| \le \|S\| \times |A (H_R^{\mathrm{r}\,\mathrm{T}})_{\bullet,l}| \le \|S\| \times |a| \le \|S\| \times |v|. \tag{E33}$$

Now, we prove that the matrix U in Eq. (E31) also satisfies Eq. (E30) when $|v| < d_R$. Using the expression of U, we have

$$H_G U \;=\; H_G A H_R^{\mathrm{r}\,\mathrm{T}} = B H_R^{\mathrm{T}} H_R^{\mathrm{r}\,\mathrm{T}}. \tag{E34}$$

Here, we have used Eq. (E28). If

$$B = B H_R^{\mathrm{T}} H_R^{\mathrm{r}\,\mathrm{T}}, \tag{E35}$$

Eq. (E31) holds. Note that the right inverse matrix of H_R is not unique. Next, we will prove that there always exists a right inverse matrix of H_R such that Eq. (E35) is true.

Because $|v| < d_R$, the matrix B has at most $d_R - 1$ non-zero columns. Suppose the indexes of the $d_R - 1$ columns are $i_1, i_2, \ldots, i_{d_R-1}$ and all other columns are zero. We introduce the matrices

$$B^{(0)} = \begin{pmatrix} B_{\bullet,i_1} & B_{\bullet,i_2} & \cdots & B_{\bullet,i_{d_R-1}} \end{pmatrix}, \tag{E36}$$

$$H_R^{(0)} = \begin{pmatrix} (H_R)_{\bullet,i_1} & (H_R)_{\bullet,i_2} & \cdots & (H_R)_{\bullet,i_{d_R-1}} \end{pmatrix}. \tag{E37}$$

Later, we will prove that the $d_R - 1$ columns in $H_R^{(0)}$ are linearly independent, and then $r_R \geq d_R - 1$. Because H_R is full-rank and has r_R linearly-independent columns, we can find additional $r_R - (d_R - 1)$ columns to constitute a complete set together with columns in $H_R^{(0)}$. Let $i_{d_R}, i_{d_R+1}, \ldots, i_{r_R}$ be indexes of the additional $r_R - (d_R - 1)$ columns, we define

$$H_R^{(1)} = \begin{pmatrix} (H_R)_{\bullet,i_{d_R}} & (H_R)_{\bullet,i_{d_R+1}} & \cdots & (H_R)_{\bullet,i_{r_R}} \end{pmatrix}. \tag{E38}$$

Then, there exists a permutation matrix $\pi \in \mathbb{F}_2^{n_R \times n_R}$ such that matrices B and H_R are in the forms

$$B = \begin{pmatrix} B^{(0)} & 0 & 0 \end{pmatrix} \pi, \tag{E39}$$

$$H_R = \begin{pmatrix} H_R^{(0)} & H_R^{(1)} & H_R^{(2)} \end{pmatrix} \pi. \tag{E40}$$

Because $H_R^{(0,1)} = \begin{pmatrix} H_R^{(0)} & H_R^{(1)} \end{pmatrix}$ is invertible, a right inverse matrix of H_R is

$$H_R^{\mathrm{r}} = \pi^{-1} \begin{pmatrix} {H_R^{(0,1)}}^{-1} \\ 0 \end{pmatrix} = \pi^{-1} \begin{pmatrix} ({H_R^{(0,1)}}^{-1})_{1:d_R-1,\bullet} \\ ({H_R^{(0,1)}}^{-1})_{d_R:r_R,\bullet} \\ 0 \end{pmatrix}. \tag{E41}$$

Note that $\pi^{-1} = \pi^{\mathrm{T}}$. Taking this right inverse matrix, we have

$$B H_R^{\mathrm{T}} {H_R^{\mathrm{r}}}^{\mathrm{T}} = \begin{pmatrix} B^{(0)} & 0 & 0 \end{pmatrix} \begin{pmatrix} E_{d_R-1} & 0 & 0 \\ 0 & E_{r_R-(d_R-1)} & 0 \\ {H_R^{(2)}}^{\mathrm{T}} [({H_R^{(0,1)}}^{-1})_{1:d_R-1,\bullet}]^{\mathrm{T}} & {H_R^{(2)}}^{\mathrm{T}} [({H_R^{(0,1)}}^{-1})_{d_R:r_R,\bullet}]^{\mathrm{T}} & 0 \end{pmatrix} \pi = B. \tag{E42}$$

Eq. (E35) has been proved.

Lastly, we prove that columns in $H_R^{(0)}$ are linearly independent. Suppose columns in $H_R^{(0)}$ are not linearly independent, there exists a vector $x \in \mathbb{F}_2^{d_R-1} - \{0\}$ such that $H_R^{(0)} x^{\mathrm{T}} = 0$. Then, we can construct a vector $y \in \mathbb{F}_2^{n_R}$,

$$y = \begin{pmatrix} x & 0 & 0 \end{pmatrix} \pi. \tag{E43}$$

We have $H_R y^{\mathrm{T}} = 0$, $y \neq 0$, and $|y| < d_R$. This is in contradiction with the code distance of the R code. The lemma has been proved. □

Lemma 12. *For the deformed code with check matrices in Eqs. (3) and (4), the generator matrices of X and Z logical operators are*

$$J_X^D = \begin{pmatrix} \tilde{\alpha}_\perp \tilde{J}_X & \tilde{\beta} \end{pmatrix} \tag{E44}$$

and

$$J_Z^D = \begin{pmatrix} {\tilde{\alpha}_\perp^{\mathrm{r}}}^{\mathrm{T}} \tilde{J}_Z & 0 \end{pmatrix}, \tag{E45}$$

respectively. Let d be the distance of the code (H_X, H_Z, J_X, J_Z). If H_R is full-rank, the distance of the deformed code satisfies

$$d_D \geq \min\{d/\|S\|, d_R\}. \tag{E46}$$

Proof. The generator matrices satisfy conditions $H_X^D J_Z^{D\mathrm{T}} = H_Z^D J_X^{D\mathrm{T}} = 0$ and $J_X^D J_Z^{D\mathrm{T}} = E_{k_R(k-q)}$, i.e. they are valid generator matrices of the deformed code.

Let $e = \begin{pmatrix} u & v \end{pmatrix}$ be an X logical error, i.e. $H_Z^D e^{\mathrm{T}} = 0$ and $J_Z^D e^{\mathrm{T}} \neq 0$. Then, we have

$$\tilde{H}_Z u^{\mathrm{T}} \;=\; 0 \tag{E47}$$

and

$$\tilde{\alpha}_\perp^{\mathrm{r}\,\mathrm{T}} \tilde{J}_Z u^{\mathrm{T}} \;\neq\; 0. \tag{E48}$$

Let's introduce the matrix U such that

$$\mathrm{vec}(U) \;=\; u^{\mathrm{T}}. \tag{E49}$$

Then,

$$H_Z U \;=\; 0 \tag{E50}$$

and

$$\alpha_\perp^{\mathrm{r}\,\mathrm{T}} J_Z U \;\neq\; 0. \tag{E51}$$

Suppose the m-th column in $\alpha_\perp^{\mathrm{r}\,\mathrm{T}} J_Z U$ is not zero, we have

$$H_Z U_{\bullet,m} \;=\; 0 \tag{E52}$$

and

$$\alpha_\perp^{\mathrm{r}\,\mathrm{T}} J_Z U_{\bullet,m} \;\neq\; 0. \tag{E53}$$

According to the code distance d,

$$|e| \;\geq\; |U_{\bullet,m}| \geq d. \tag{E54}$$

Let $e = \begin{pmatrix} u & v \end{pmatrix}$ be a Z logical error, i.e. $H_X^D e^{\mathrm{T}} = 0$ and $J_X^D e^{\mathrm{T}} \neq 0$. Then, we have

$$\tilde{H}_X u^{\mathrm{T}} \;=\; \tilde{T} v^{\mathrm{T}}, \tag{E55}$$

$$\tilde{H}_M v^{\mathrm{T}} \;=\; 0, \tag{E56}$$

and

$$\tilde{\alpha}_\perp \tilde{J}_X u^{\mathrm{T}} + \tilde{\beta} v^{\mathrm{T}} \;\neq\; 0. \tag{E57}$$

If $|v| < d_R$, there exists a vector $u_\star \in \mathbb{F}_2^{k_R n_G}$ such that $v^{\mathrm{T}} = \tilde{H}_G u_\star^{\mathrm{T}}$ and the inequality (E23) holds. Then,

$$\tilde{H}_X u^{\mathrm{T}} \;=\; \tilde{T} \tilde{H}_G u_\star^{\mathrm{T}} \tag{E58}$$

and

$$\tilde{\alpha}_\perp \tilde{J}_X u^{\mathrm{T}} + \tilde{\beta} \tilde{H}_G u_\star^{\mathrm{T}} \;\neq\; 0. \tag{E59}$$

Furthermore, we have

$$\tilde{H}_X (u + u_\star \tilde{S})^{\mathrm{T}} \;=\; 0 \tag{E60}$$

and

$$\tilde{\alpha}_\perp \tilde{J}_X (u + u_\star \tilde{S})^{\mathrm{T}} \;\neq\; 0. \tag{E61}$$

Here, we have used Eqs. (E16) and (E19). Let's introduce the matrix U such that

$$\mathrm{vec}(U) \;=\; (u + u_\star \tilde{S})^{\mathrm{T}}. \tag{E62}$$

Then,

$$H_X U \;=\; 0 \tag{E63}$$

and

$$\alpha_\perp J_X U \;\neq\; 0. \tag{E64}$$

Suppose the m-th column in $\alpha_\perp J_X U$ is not zero, we have

$$H_X U_{\bullet,m} \;=\; 0 \tag{E65}$$

and

$$\alpha_\perp J_X U_{\bullet,m} \;\neq\; 0. \tag{E66}$$

According to the code distance d,

$$|U_{\bullet,m}| \geq d. \tag{E67}$$

Note that

$$\begin{aligned} U_{\bullet,m} \;&=\; U(E_{k_R})_{\bullet,m} = [(E_{k_R})_{m,\bullet} \otimes E_n](u + u_\star \tilde{S})^{\mathrm{T}} \\ &=\; [(E_{k_R})_{m,\bullet} \otimes E_n]u^{\mathrm{T}} + [(G_R^{\mathrm{r}})_{\bullet,m} \otimes S]^{\mathrm{T}} u_\star^{\mathrm{T}}, \end{aligned} \tag{E68}$$

therefore,

$$\|S\| \times |e| \;\geq\; |u| + \|S\| \times |v| \geq |[(E_{k_R})_{m,\bullet} \otimes E_n]u^{\mathrm{T}}| + |[(G_R^{\mathrm{r}})_{\bullet,m} \otimes S]^{\mathrm{T}} u_\star^{\mathrm{T}}| \geq d. \tag{E69}$$

□

Appendix F: Locally-testable state preparation - Proof of Lemma 2

In LTSP, we generate resource states used to measure one type of stabilizer operator—either X or Z—of a CSS code. Without loss of generality, we focus on the measurement of Z stabilizer operators, specifically the operators $Z(H_Z^D)$ in the deformed code. The conclusions are straightforwardly applicable to X stabilizer operators or to other CSS codes, such as the memory code.

1. Stabilizer of the resource state

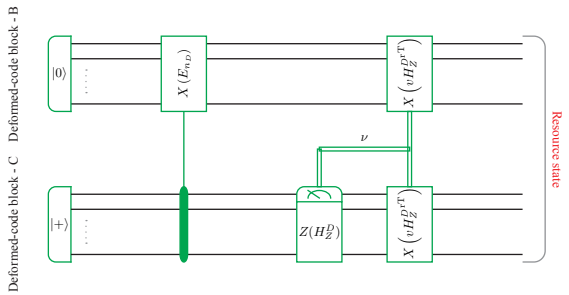

FIG. 10. Circuit that defines the resource state for the $Z(H_Z^D)$ measurement. Outcome of the $Z(H_Z^D)$ measurement in the circuit is $\nu \in \mathbb{F}_2^{r_Z^D}$.

To implement the measurement of $Z(H_Z^D)$ using gate teleportation, we need to prepare a resource state encoded in two blocks of the deformed code, denoted by B and C. Each block contains n_D qubits. Ideally, the resource state

can be prepared as follows; see Fig. 10. First, we generate n_D copies of the Bell state between the two blocks, which constitute a $2n_D$-qubit stabilizer state, and the stabilizer is generated by operators $X(H_X^{BS})$ and $Z(H_Z^{BS})$, where

$$H_X^{BS} \;=\; H_Z^{BS} = \begin{pmatrix} E_{n_D} & E_{n_D} \end{pmatrix}. \tag{F1}$$

We can rewrite the matrices as

$$\tilde{H}_X^{BS} \;=\; \begin{pmatrix} H_X^D & H_X^D \\ J_X^D & J_X^D \\ H_Z^{D^{\mathrm{r}\mathrm{T}}} & H_Z^{D^{\mathrm{r}\mathrm{T}}} \end{pmatrix} \tag{F2}$$

and

$$\tilde{H}_Z^{BS} \;=\; \begin{pmatrix} H_X^{D^{\mathrm{r}\mathrm{T}}} & H_X^{D^{\mathrm{r}\mathrm{T}}} \\ J_Z^D & J_Z^D \\ H_Z^D & H_Z^D \end{pmatrix}, \tag{F3}$$

such that $\mathrm{rowsp} H_{X/Z}^{BS} = \mathrm{rowsp} \tilde{H}_{X/Z}^{BS}$. Without loss generality, we can choose $H_X^{D^{\mathrm{r}}}$ and $H_Z^{D^{\mathrm{r}}}$ such that $J_X^D H_X^{D^{\mathrm{r}}} = J_Z^D H_Z^{D^{\mathrm{r}}} = H_X^{D^{\mathrm{r}\mathrm{T}}} H_Z^{D^{\mathrm{r}}} = 0$. Then, we apply the measurement of operators $Z(H_Z^D)$ on block-C, followed by a feedback Pauli gate. Note that $X\left(\nu H_Z^{D^{\mathrm{r}\mathrm{T}}}\right) Z\left((H_Z^D)_{j,\bullet}\right) X\left(\nu H_Z^{D^{\mathrm{r}\mathrm{T}}}\right) = (-1)^{\nu_j} Z\left((H_Z^D)_{j,\bullet}\right)$. These operations transform Bell states into another stabilizer sate with the stabilizer generated by operators $X(H_X^{RS})$ and $Z(H_Z^{RS})$, where

$$H_X^{RS} \;=\; \begin{pmatrix} H_X^D & H_X^D \\ J_X^D & J_X^D \end{pmatrix} \tag{F4}$$

and

$$H_Z^{RS} \;=\; \begin{pmatrix} E_{n_D} & E_{n_D} \\ 0 & H_Z^D \end{pmatrix}. \tag{F5}$$

The resource state to be prepared is the stabilizer state defined by the generators $X(H_X^{RS})$ and $Z(H_Z^{RS})$.

2. State preparation using a classical locally testable code

We prepare the resource state with the circuit shown in Fig. 11. The circuit constitutes of three sets of F-code blocks, denoted by B, C, and D, containing n_D, n_D, and r_Z^D blocks, respectively. In each F-code block, we encode k_F logical qubits according to the following CSS code: Stabilizer operators are $X(0)$ and $Z(H_F)$, and logical operators are $X(G_F)$ and $Z({G_F^{\mathrm{r}}}^{\mathrm{T}})$. Without loss of generality, we consider the F-code generator matrix in the form

$$G_F \;=\; \begin{pmatrix} E_{k_F} & P_F \end{pmatrix}, \tag{F6}$$

and then

$${G_F^{\mathrm{r}}}^{\mathrm{T}} \;=\; \begin{pmatrix} E_{k_F} & 0 \end{pmatrix}. \tag{F7}$$

This circuit yields n_F copies of the resource state and is supported by the following lemmas.

Lemma 13. *Let $|e_Z^{sp}|$ denote the total weight of Z errors occurring at the spacetime locations in the state preparation circuit shown in Fig. 11. Then, each copy of the prepared resource state contains Z errors with total weight at most $|e_Z^{RS}| \le |e_Z^{sp}|$.*

Lemma 14. *Let $|e_X^{sp}|$ denote the total weight of X errors (including incorrect outcomes in projective measurements of Z operators) occurring at the spacetime locations in the state preparation circuit shown in Fig. 11. Suppose the F code is (ω, s)-locally testable, and that H_Z^D has rows and columns of weights at most ω_Z^D. If the spacetime X errors are undetectable by the two sets of checks illustrated in Fig. 5, and*

$$|e_X^{sp}| \;<\; \frac{d_F}{\omega_Z^D \max\left\{1, \frac{n_F}{r_F s}\right\}}, \tag{F8}$$

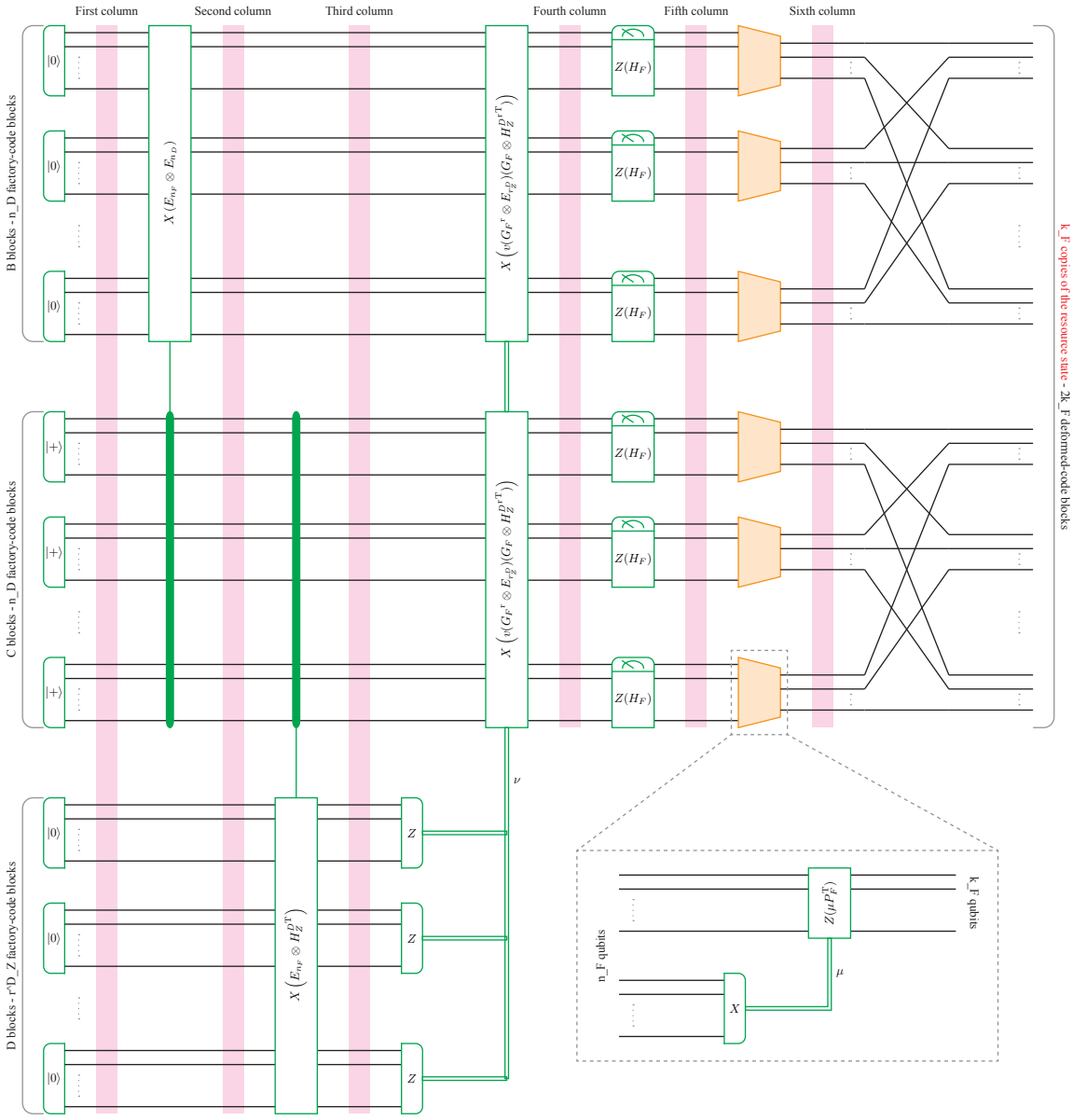

FIG. 11. Locally-testable state preparation. The outcome of the transversal measurement on D blocks is $\nu = \mathrm{vec}(V)^{\mathrm{T}}$, where $V \in \mathbb{F}_2^{r_Z^D \times n_F}$, and $V_{i,j}$ is the outcome of the measurement on the j-th qubit in the i-th block. The outcome of the partial transversal measurement on one of B and C blocks in decoding is $\mu \in \mathbb{F}_2^{n_F - k_F}$, and μ_j is the outcome of the measurement on the $(k_F + j)$-th qubit in the block.

then each copy of the prepared resource state contains X errors with total weight at most

$$|e_X^{RS}| \;\leq\; \max\left\{1, \frac{n_F}{r_F s}\right\} \times |e_X^{sp}|. \tag{F9}$$

a. Proof of Lemma 13 - X operators and Z errors

Focusing on the j-th copy of the prepared resource state, i.e. j-th logical qubits in B and C F-code blocks, the propagation of X operators in spacetime is described by the generator matrix

$$J^{sp}_{X,j} = \begin{pmatrix} J^{sp}_{X,B,j} & J^{sp}_{X,C,j} & J^{sp}_{X,D,j} \end{pmatrix}, \tag{F10}$$

where

$$J^{sp}_{X,B,j} = \begin{pmatrix} 0 & (G_F)_{j,\bullet} \otimes H^D_X & (G_F)_{j,\bullet} \otimes H^D_X & (G_F)_{j,\bullet} \otimes H^D_X & (G_F)_{j,\bullet} \otimes H^D_X & (E_{k_F})_{j,\bullet} \otimes H^D_X \\ 0 & (G_F)_{j,\bullet} \otimes J^D_X & (G_F)_{j,\bullet} \otimes J^D_X & (G_F)_{j,\bullet} \otimes J^D_X & (G_F)_{j,\bullet} \otimes J^D_X & (E_{k_F})_{j,\bullet} \otimes J^D_X \end{pmatrix}, \tag{F11}$$

$$J^{sp}_{X,C,j} = \begin{pmatrix} (G_F)_{j,\bullet} \otimes H^D_X & (G_F)_{j,\bullet} \otimes H^D_X & (G_F)_{j,\bullet} \otimes H^D_X & (G_F)_{j,\bullet} \otimes H^D_X & (G_F)_{j,\bullet} \otimes H^D_X & (E_{k_F})_{j,\bullet} \otimes H^D_X \\ (G_F)_{j,\bullet} \otimes J^D_X & (G_F)_{j,\bullet} \otimes J^D_X & (G_F)_{j,\bullet} \otimes J^D_X & (G_F)_{j,\bullet} \otimes J^D_X & (G_F)_{j,\bullet} \otimes J^D_X & (E_{k_F})_{j,\bullet} \otimes J^D_X \end{pmatrix}, \tag{F12}$$

and

$$J^{sp}_{X,D,j} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{F13}$$

correspond to spacetime locations on B, C, and D blocks, respectively: The six columns correspond to spacetime locations as illustrated in Fig. 11. We remark that operators on C blocks do not propagate to D blocks by the second transversal controlled-NOT gate because of $[(G_F)_{j,\bullet} \otimes H^D_X][E_{n_F} \otimes {H^D_Z}^{\mathrm{T}}] = [(G_F)_{j,\bullet} \otimes J^D_X][E_{n_F} \otimes {H^D_Z}^{\mathrm{T}}] = 0$. This generator matrix describes that operators $X\left((G_F)_{j,\bullet} \otimes H^D_X\right)$ and $X\left((G_F)_{j,\bullet} \otimes J^D_X\right)$ on C blocks at the beginning are transformed into operators $X\left((G_F)_{j,\bullet} \otimes H^D_X\right) \otimes X\left((G_F)_{j,\bullet} \otimes H^D_X\right)$ and $X\left((G_F)_{j,\bullet} \otimes J^D_X\right) \otimes X\left((G_F)_{j,\bullet} \otimes J^D_X\right)$ on B and C blocks at the end, respectively. Because qubits in C blocks are initialized in the state $|+\rangle$, the final state is an eigenstate of operators $X\left((G_F)_{j,\bullet} \otimes H^D_X\right) \otimes X\left((G_F)_{j,\bullet} \otimes H^D_X\right)$ and $X\left((G_F)_{j,\bullet} \otimes J^D_X\right) \otimes X\left((G_F)_{j,\bullet} \otimes J^D_X\right)$ on B and C blocks with the eigenvalue $+1$.

We represent Z errors in spacetime with a vector

$$e^{sp}_Z = \begin{pmatrix} e^{sp}_{Z,B} & e^{sp}_{Z,C} & e^{sp}_{Z,D} \end{pmatrix}, \tag{F14}$$

where

$$e^{sp}_{Z,B/C} = \begin{pmatrix} u_{B/C,1} & u_{B/C,2} & u_{B/C,3} & u_{B/C,4} & u_{B/C,5} & u_{B/C,6} \end{pmatrix} \tag{F15}$$

and

$$e^{sp}_{Z,D} = \begin{pmatrix} u_{D,1} & u_{D,2} & u_{D,3} \end{pmatrix}. \tag{F16}$$

The errors flip operators during the propagation according to

$$\begin{aligned} J^{sp}_{X,j} {e^{sp}_Z}^{\mathrm{T}} &= \begin{pmatrix} \mathrm{vec}\left(H^D_X U[(G_F)_{j,\bullet}]^{\mathrm{T}}\right) \\ \mathrm{vec}\left(J^D_X U[(G_F)_{j,\bullet}]^{\mathrm{T}}\right) \end{pmatrix} + \begin{pmatrix} \mathrm{vec}\left(H^D_X U'[(E_{k_F})_{j,\bullet}]^{\mathrm{T}}\right) \\ \mathrm{vec}\left(J^D_X U'[(E_{k_F})_{j,\bullet}]^{\mathrm{T}}\right) \end{pmatrix} \\ &= \begin{pmatrix} H^D_X u^{\mathrm{T}}_{eff} \\ J^D_X u^{\mathrm{T}}_{eff} \end{pmatrix} = H^{RS}_X \begin{pmatrix} 0 \\ u^{\mathrm{T}}_{eff} \end{pmatrix}, \end{aligned} \tag{F17}$$

where

$$\mathrm{vec}(U) = (u_{B,2} + u_{B,3} + u_{B,4} + u_{B,5} + u_{C,1} + u_{C,2} + u_{C,3} + u_{C,3} + u_{C,4} + u_{C,5})^{\mathrm{T}}, \tag{F18}$$

$$\mathrm{vec}(U') = (u_{B,6} + u_{C,6})^{\mathrm{T}}, \tag{F19}$$

$$u^{\mathrm{T}}_{eff} = U[(G_F)_{j,\bullet}]^{\mathrm{T}} + U'[(E_{k_F})_{j,\bullet}]^{\mathrm{T}}. \tag{F20}$$

We can find that the vector

$$e^{RS}_Z = \begin{pmatrix} 0 & u_{eff} \end{pmatrix} \in \mathbb{F}_2^{2n_D} \tag{F21}$$

satisfies $H^{RS}_X {e^{RS}_Z}^{\mathrm{T}} = J^{sp}_X {e^{sp}_Z}^{\mathrm{T}}$ as illustrated in Eq. (F17). Therefore, e^{RS}_Z represents Z errors on the prepared resource state. According to Eq. (F20), it satisfies the inequality $|e^{RS}_Z| \leq |e^{sp}_Z|$.

b. Proof of Lemma 14 - Z operators and X errors

Focusing on the j-th copy of the prepared resource state, i.e. j-th logical qubits in B and C F-code blocks, the propagation of Z operators in spacetime is described by the generator matrix

$$J^{sp}_{Z,j} = \begin{pmatrix} J^{sp}_{Z,B,j} & J^{sp}_{Z,C,j} & J^{sp}_{Z,D,j} & J^{sp}_{Z,mea,j} \end{pmatrix}, \tag{F22}$$

where

$$J^{sp}_{Z,B,j} = \begin{pmatrix} (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (E_{k_F})_{j,\bullet} \otimes E_{n_D} \\ 0 & 0 & 0 & 0 & & \end{pmatrix}, \tag{F23}$$

$$J^{sp}_{Z,C,j} = \begin{pmatrix} 0 & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D} & (E_{k_F})_{j,\bullet} \otimes E_{n_D} \\ 0 & 0 & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes H_Z^D & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes H_Z^D & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes H_Z^D & (E_{k_F})_{j,\bullet} \otimes H_Z^D \end{pmatrix}, \tag{F24}$$

and

$$J^{sp}_{Z,D,j} = \begin{pmatrix} 0 & 0 & 0 \\ (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{r_Z^D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{r_Z^D} & (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{r_Z^D} \end{pmatrix}, \tag{F25}$$

correspond to spacetime locations on B, C, and D blocks, respectively, and

$$J^{sp}_{Z,mea,j} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{F26}$$

corresponds to measurements of $Z(H_F)$: The two columns correspond to measurements on B and C blocks, respectively. Note that $[(G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes H_Z^D][(G_F^{\mathrm{r}} \otimes E_{r_Z^D})(G_F \otimes H_Z^{D^{\mathrm{r}\,\mathrm{T}}})]^{\mathrm{T}} = (G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{r_Z^D}$, which justifies the propagation through the feedback gate depending on the outcome ν. This generator matrix describes that operators $Z\left((G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{n_D}\right)$ on B blocks and and $Z\left((G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet} \otimes E_{r_Z^D}\right)$ on D blocks at the beginning are transformed into operators $Z\left((E_{k_F})_{j,\bullet} \otimes E_{n_D}\right) \otimes Z\left((E_{k_F})_{j,\bullet} \otimes E_{n_D}\right)$ on B and C blocks and $Z\left((E_{k_F})_{j,\bullet} \otimes H_Z^D\right)$ on C blocks at the end, respectively. Because qubits in C and D blocks are initialized in the state $|0\rangle$, the final state is an eigenstate of operators $Z\left((E_{k_F})_{j,\bullet} \otimes E_{n_D}\right) \otimes Z\left((E_{k_F})_{j,\bullet} \otimes E_{n_D}\right)$ on B and C blocks and $Z\left((E_{k_F})_{j,\bullet} \otimes H_Z^D\right)$ on C blocks with the eigenvalue $+1$.

We detect errors with two sets of checks. First, let $\eta \in \mathbb{F}_2^{r_F}$ be the outcome of a $Z(H_F)$ measurement, it satisfies $\eta \in \mathrm{colsp} H_F$ if the circuit is error-free. Therefore, we can detect errors that flip the measurement outcome through $H_M^F \eta$, where H_M^F is the check matrix of the code $\mathrm{colsp} H_F$. Second, because qubits in C and D blocks are initialized in the state $|0\rangle$, outcomes of $Z(H_F)$ measurements on B and C blocks as well as transversal Z measurements on D blocks satisfy certain equations if the circuit is error-free. Accordingly, the check matrix is

$$H^{sp}_Z = \begin{pmatrix} H^{sp}_{Z,B} & H^{sp}_{Z,C} & H^{sp}_{Z,D} & H^{sp}_{Z,mea} \end{pmatrix}, \tag{F27}$$

where

$$H^{sp}_{Z,B} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ H_F \otimes E_{n_D} & H_F \otimes E_{n_D} & H_F \otimes E_{n_D} & H_F \otimes E_{n_D} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{F28}$$

$$H^{sp}_{Z,C} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & H_F \otimes E_{n_D} & H_F \otimes E_{n_D} & H_F \otimes E_{n_D} & 0 & 0 \\ 0 & 0 & H_F \otimes H_Z^D & H_F \otimes H_Z^D & 0 & 0 \end{pmatrix}, \tag{F29}$$

$$H^{sp}_{Z,D} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ H_F \otimes E_{r_Z^D} & H_F \otimes E_{r_Z^D} & H_F \otimes E_{r_Z^D} \end{pmatrix}, \tag{F30}$$

and

$$H^{sp}_{Z,mea} = \begin{pmatrix} H^F_M \otimes E_{n_D} & 0 \\ 0 & H^F_M \otimes E_{n_D} \\ E_{r_F} \otimes E_{n_D} & E_{r_F} \otimes E_{n_D} \\ 0 & E_{r_F} \otimes H^D_Z \end{pmatrix}. \tag{F31}$$

We represent X errors in spacetime with a vector

$$e^{sp}_X = \begin{pmatrix} e^{sp}_{X,B} & e^{sp}_{X,C} & e^{sp}_{X,D} & e^{sp}_{X,mea} \end{pmatrix}, \tag{F32}$$

where

$$e^{sp}_{X,B/C} = \begin{pmatrix} u_{B/C,1} & u_{B/C,2} & u_{B/C,3} & u_{B/C,4} & u_{B/C,5} & u_{B/C,6} \end{pmatrix}, \tag{F33}$$

$$e^{sp}_{X,D} = \begin{pmatrix} u_{D,1} & u_{D,2} & u_{D,3} \end{pmatrix}. \tag{F34}$$

and

$$e^{sp}_{X,mea} = \begin{pmatrix} v_B & v_C \end{pmatrix}. \tag{F35}$$

The errors flip operators during the propagation according to

$$J^{sp}_{Z,j} e^{sp\,\mathrm{T}}_X = \begin{pmatrix} \mathrm{vec}\left((U_B + U'_B + U_C + U'_C)[(G^{\mathrm{r}}_F{}^{\mathrm{T}})_{j,\bullet}]^{\mathrm{T}}\right) \\ \mathrm{vec}\left((H^D_Z U_C + H^D_Z U'_C + U_D)[(G^{\mathrm{r}}_F{}^{\mathrm{T}})_{j,\bullet}]^{\mathrm{T}}\right) \end{pmatrix} + \begin{pmatrix} \mathrm{vec}\left((U''_B + U''_C)[(E_{k_F})_{j,\bullet}]^{\mathrm{T}}\right) \\ \mathrm{vec}\left(H^D_Z U''_C[(E_{k_F})_{j,\bullet}]^{\mathrm{T}}\right) \end{pmatrix}, \tag{F36}$$

where

$$\begin{aligned} \mathrm{vec}(U_B) &= (u_{B,1} + u_{B,2} + u_{B,3} + u_{B,4} + u_{C,2})^{\mathrm{T}}, && \text{(F37)} \\ \mathrm{vec}(U'_B) &= u^{\mathrm{T}}_{B,5}, && \text{(F38)} \\ \mathrm{vec}(U''_B) &= u^{\mathrm{T}}_{B,6}, && \text{(F39)} \\ \mathrm{vec}(U_C) &= (u_{C,3} + u_{C,4})^{\mathrm{T}}, && \text{(F40)} \\ \mathrm{vec}(U'_C) &= u^{\mathrm{T}}_{C,5}, && \text{(F41)} \\ \mathrm{vec}(U''_C) &= u^{\mathrm{T}}_{C,6}, && \text{(F42)} \\ \mathrm{vec}(U_D) &= (u_{D,1} + u_{D,2} + u_{D,3})^{\mathrm{T}}. && \text{(F43)} \\ & && \text{(F44)} \end{aligned}$$

If the spacetime error e^{sp}_X is undetectable by H^{sp}_Z, it satisfies the condition $H^{sp}_Z e^{sp\,\mathrm{T}}_X = 0$. Let

$$\mathrm{vec}(V_{B/C}) = v^{\mathrm{T}}_{B/C}, \tag{F45}$$

the condition can be rewritten as

$$\begin{aligned} V_{B/C} H^F_M{}^{\mathrm{T}} &= 0, && \text{(F46)} \\ (U_B + U_C) H^{\mathrm{T}}_F &= V_B + V_C && \text{(F47)} \\ H^D_Z U_C H^{\mathrm{T}}_F + U_D H^{\mathrm{T}}_F &= H^D_Z V_C. && \text{(F48)} \end{aligned}$$

When Eq. (F46) holds, there exists $W_{B/C} \in \mathbb{F}_2^{n_D \times n_F}$ such that $V_{B/C} = W_{B/C} H^{\mathrm{T}}_F$, and

$$|(W_{B/C})_{a,\bullet}| \leq \frac{n_F}{r_F s} |(V_{B/C})_{a,\bullet}| \tag{F49}$$

for all $a = 1, 2, \ldots, n_D$; see Lemma 4. Then, we have

$$\begin{aligned} (U_B + U_C) H^{\mathrm{T}}_F &= (W_B + W_C) H^{\mathrm{T}}_F, && \text{(F50)} \\ H^D_Z U_C H^{\mathrm{T}}_F + U_D H^{\mathrm{T}}_F &= H^D_Z W_C H^{\mathrm{T}}_F. && \text{(F51)} \end{aligned}$$

If the inequality (F8) is true,

$$|(U_B+U_C)_{a,\bullet}+(W_B+W_C)_{a,\bullet}| \;\le\; \max\left\{1,\frac{n_F}{r_F s}\right\}\times|e_Z^{sp}|\le d_F \tag{F52}$$

and

$$|(H_Z^D U_C+U_D)_{b,\bullet}+(H_Z^D W_C)_{a,\bullet}| \;\le\; \omega_Z^D\max\left\{1,\frac{n_F}{r_F s}\right\}\times|e_Z^{sp}|\le d_F \tag{F53}$$

hold for all $a=1,2,\ldots,n_D$ and $b=1,2,\ldots,r_Z^D$. Then, according to the F-code distance, we have

$$U_B+U_C \;=\; W_B+W_C, \tag{F54}$$
$$H_Z^D U_C+U_D \;=\; H_Z^D W_C. \tag{F55}$$

Substitute the above equations into Eq. (F36), we obtain

$$\begin{aligned} J_{Z,j}^{sp}e_X^{sp\mathrm{T}} &= \begin{pmatrix} \mathrm{vec}\left((W_B+W_C+U_B'+U_C')[(G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet}]^{\mathrm{T}}\right) \\ \mathrm{vec}\left((H_Z^D W_C+H_Z^D U_C')[(G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet}]^{\mathrm{T}}\right)\end{pmatrix} + \begin{pmatrix} \mathrm{vec}\left((U_B''+U_C'')[(E_{k_F})_{j,\bullet}]^{\mathrm{T}}\right) \\ \mathrm{vec}\left(H_Z^D U_C''[(E_{k_F})_{j,\bullet}]^{\mathrm{T}}\right)\end{pmatrix} \\ &= H_Z^{RS}\begin{pmatrix} u_{eff,B}^{\mathrm{T}} \\ u_{eff,C}^{\mathrm{T}}\end{pmatrix}, \end{aligned} \tag{F56}$$

where

$$u_{eff,B/C}^{\mathrm{T}} \;=\; W_{B/C}[(G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet}]^{\mathrm{T}}+U_{B/C}'[(G_F^{\mathrm{r}\,\mathrm{T}})_{j,\bullet}]^{\mathrm{T}}+U_{B/C}''[(E_{k_F})_{j,\bullet}]^{\mathrm{T}}. \tag{F57}$$

Accordingly, the X error on the prepared resource state is

$$e_X^{RS} \;=\; \begin{pmatrix} u_{eff,B} & u_{eff,C}\end{pmatrix}, \tag{F58}$$

such that $H_Z^{RS}e_X^{RS\mathrm{T}}=J_{Z,j}^{sp}e_X^{sp\mathrm{T}}$. Note that

$$|u_{eff,B}|+|u_{eff,C}| \;\le\; \max\left\{1,\frac{n_F}{r_F s}\right\}\times|e_X^{sp}|. \tag{F59}$$

The lemma has been proved.

3. Surface code in the state preparation

If we prepare resource states with the circuit in Fig. 11, the F code can only correct X errors, and Z errors may accumulate and damage the prepared resource states. To solve this problem, we protect each qubit in the circuit with a surface code.

We simulate the circuit in Fig. 11 using the surface code. Now, each qubit in the circuit becomes a surface-code logical qubit. The circuit involves the following operations: initializations in states $|0\rangle$ and $|+\rangle$, measurements in the X and Z bases, Pauli gates, and controlled-NOT gate. We can implement these operations on surface-code logical qubits in the transversal way; see Figs. 12(a-d). For initializations and the controlled-NOT gate, we measure surface-code stabilizer operators for $d_T=\Theta(d_S)$ rounds after each logical operation to correct the errors. With these logical operations, we can simulate the circuit in Fig. 11 and prepare resource states encoded in surface-code logical qubits.

Next, to obtain resource states in physical qubits, we have to decode the surface-code logical qubits. This operation can be realized by measuring physical qubits in a surface-code block in X and Z bases, leaving only one physical qubit unmeasured; see Fig. 12(e). After decoding each surface-code logical qubit in the resource state, we obtain the eventual physical-qubit resource state; see Fig. 12(f).

Finally, we analyze the impact of using the surface code on conclusions in Lemmas 13 and 14. On surface-code logical qubits, we implement a logical operation using a set of physical operations. The logical operation has errors only if physical operations have errors: Let w_L and w_P be the weights of logical errors and physical errors, respectively; then $w_L=0$ if $w_P=0$. For single-qubit logical operations, i.e. initializations, measurements, and Pauli gates, we have $w_L\le w_P$: Note that w_L is either zero or one. For a logical controlled-NOT gate, because it acts on two

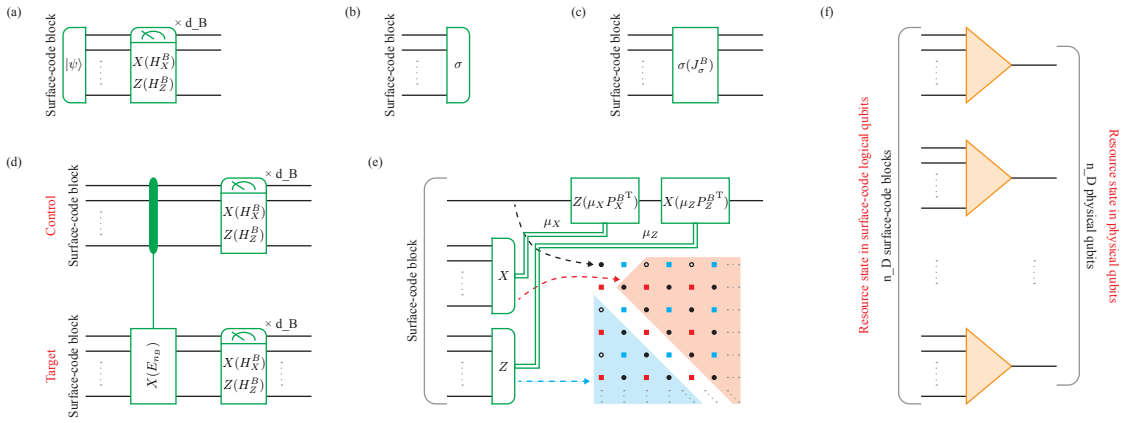

FIG. 12. Operations on surface-code logical qubits. (a) Initialization in the state $|\psi\rangle = |0\rangle, |+\rangle$. (b) Measurement in the basis $\sigma = X, Z$. (c) Logical Pauli gate $\sigma = X, Z$. Vectors J_X^B and J_Z^B represent X and Z logical operators of the surface-code logical qubit. (d) Logical controlled-NOT gate. (e) Decoding on a surface-code logical qubit. Black circles denote qubits. Red and blue squares denote X and Z checks, respectively; edges of the Tanner graph have been neglected. Qubits in the area marked in red (blue) are measured in the X (Z) basis, and the measurement outcome is the vector μ_X (μ_Z). In feedback gates, P_X^B and P_X^B are vectors, and their supports are marked with open circles: In the vector P_X^B (P_Z^B), elements corresponding to open circles in the red (blue) area are ones, and other elements are zeros. (f) Decoding of the resource state. Triangles denote decoding operations on surface-code logical qubits, and each of them is realized with the circuit in (e).

surface-code logical qubits, $w_L \leq 2$, and we have $w_L \leq 2w_P$. Multiple logical controlled-NOT gates constitute the generalized transversal controlled-NOT gates in Fig. 11. The error propagation in the second generalized transversal controlled-NOT gate may transform an error on one surface-code logical qubit into an error on multiple surface-code logical qubits, i.e. increase the weight of logical errors. Because H_Z^D has rows and columns of weight at most ω_Z^D, the logical-error weight is amplified by a factor of at most ω_Z^D. Therefore, for all these logical operations, we have $w_L \leq 2\omega_Z^D w_P$. Decoding operations may also cause errors, which are the same as single-qubit logical operations; they cause extra errors, and we can take into account decoding errors in the sixth column in Fig. 11.

Lemma 15 (Formal version of Lemma 2)**.** *Suppose the F code is (ω, s)-locally testable, and H_Z^D has rows and columns of weight at most ω_Z^D. Let $\bar{e}_X^{sp}$ and $\bar{e}_Z^{sp}$ be the X and Z spacetime (physical) errors in the state preparation circuit shown in Fig. 11, when simulated using a surface code of arbitrary code distance. If the spacetime errors are undetectable and satisfy*

$$|\bar{e}_X^{sp}| \;<\; \frac{d_F}{2{\omega_Z^D}^2 \max\left\{1, \frac{n_F}{r_F s}\right\}}, \tag{F60}$$

the following inequalities hold:

$$|e_X^{RS}| \;\leq\; 2\omega_Z^D \max\left\{1, \frac{n_F}{r_F s}\right\} \times |\bar{e}_X^{sp}|, \tag{F61}$$

$$|e_Z^{RS}| \;\leq\; 2\omega_Z^D |\bar{e}_Z^{sp}|. \tag{F62}$$

Here, $\frac{n_F}{r_F s} \leq \frac{w}{s}$, since F is a good LDPC code, i.e., $wr_F \leq n_F$.

4. Overhead in the state preparation

The state preparation circuit shown in Fig. 11 involves $2n_D + r_Z^D$ F-code blocks, so the number of data qubits is $(2n_D + r_Z^D)n_F$. Additionally, we require r_F ancilla qubits to implement parity-check measurements on each F-code block, and such measurements are applied to $2n_D$ blocks. Thus, the total number of qubits is $(2n_D+r_Z^D)n_F+2n_D r_F = O(n_D n_F)$. In this circuit, all operation layers have depth one, except for two layers of parity-check measurements: the layer between the second and third columns, and the layer between the fourth and fifth columns. According to Lemma 3, each of these layers has depth $O(1)$ because the deformed code is a qLDPC code and the F code is an LDPC code. Therefore, the overall circuit depth is $O(1)$.

When simulating the state preparation circuit using a surface code, each qubit in the circuit is encoded in a surface code of distance d_S, requiring $(2d_S-1)^2 = O(d_S^2)$ physical qubits. Since the circuit involves $O(n_D n_F)$ qubits, the total number of physical qubits is $O(n_D n_F d_S^2)$. Each operation on surface-code logical qubits is implemented using a physical circuit of depth $O(d_T)$. Therefore, the overall depth of the physical circuit for state preparation is $O(d_T) = O(d_S)$.

Appendix G: Parallelized code surgery realized through locally-testable state preparation

1. Measurements

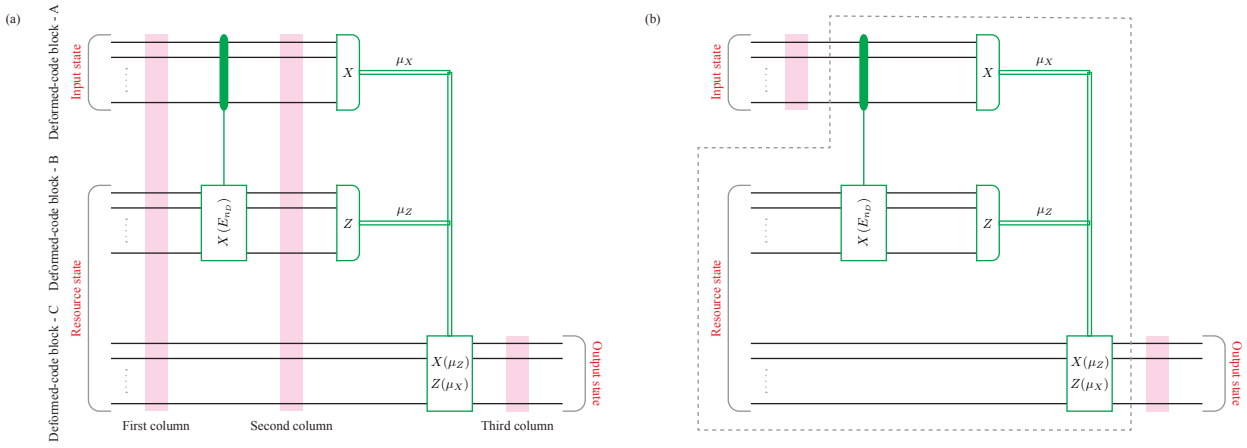

FIG. 13. (a) Measurement of $Z(H_Z^D)$ based on the resource state. Outcomes of transversal measurements on blocks A and B are $\mu_X, \mu_Z \in \mathbb{F}_2^{n_D}$, respectively. Outcome of the $Z(H_Z^D)$ measurement is $H_Z^D \mu_Z$. (b) Effective error in the measurement of $Z(H_Z^D)$ based on the resource state. Operations in the dashed box are effectively error-free.

With the resource state prepared on blocks B and C, we can transfer the state of block-A to block-C and effectively realize the $Z(H_Z^D)$ measurement; see Fig. 13.

a. X operators and Z errors

The propagation of X operators in spacetime is described by the generator matrix

$$J_X^m = \begin{pmatrix} J_{X,A}^m & J_{X,B}^m & J_{X,C}^m \end{pmatrix}, \tag{G1}$$

where

$$J_{X,A}^m = \begin{pmatrix} H_X^D & H_X^D \\ J_X^D & J_X^D \end{pmatrix}, \tag{G2}$$

$$J_{X,B}^m = \begin{pmatrix} H_X^D & 0 \\ J_X^D & 0 \end{pmatrix}, \tag{G3}$$

and

$$J_{X,C}^m = \begin{pmatrix} H_X^D & H_X^D & H_X^D \\ J_X^D & J_X^D & J_X^D \end{pmatrix}, \tag{G4}$$

correspond to spacetime locations on A, B, and C blocks, respectively: The three columns correspond to spacetime locations as illustrated in Fig. 13. This generator matrix describes that operators $X(H_X^D) \otimes X(H_X^D) \otimes X(H_X^D)$ and

$X(J_X^D) \otimes X(J_X^D) \otimes X(J_X^D)$ on three blocks at the beginning are transformed into operators $X(H_X^D)$ and $X(J_X^D)$ on block-C at the end, respectively. Note that blocks B and C are prepared in the resource state, which is an eigenstate of operators $X(H_X^D) \otimes X(H_X^D)$ and $X(J_X^D) \otimes X(J_X^D)$ with the eigenvalue $+1$. Therefore, such a process effectively transforms operators $X(H_X^D)$ and $X(J_X^D)$ on block-A into operators $X(H_X^D)$ and $X(J_X^D)$ on block-C, respectively.

We represent Z errors in spacetime with a vector

$$e_Z^m = \begin{pmatrix} e_{Z,A}^m & e_{Z,B}^m & e_{Z,C}^m \end{pmatrix} \tag{G5}$$

where

$$e_{Z,A/B}^m = \begin{pmatrix} u_{A/B,1} & u_{A/B,2} \end{pmatrix} \tag{G6}$$

and

$$e_{Z,C}^m = \begin{pmatrix} u_{C,1} & u_{C,2} & u_{C,3} \end{pmatrix}. \tag{G7}$$

The errors flip operators during the propagation according to

$$J_X^m e_Z^{m\mathrm{T}} = \begin{pmatrix} H_X^D \\ J_X^D \end{pmatrix} u_{eff}^{\mathrm{T}}, \tag{G8}$$

where

$$u_{eff} = u_{A,1} + u_{A,2} + u_{B,1} + u_{C,1} + u_{C,2} + u_{C,3}. \tag{G9}$$

Lemma 16. *Let e_Z^m be the Z spacetime error in the $Z(H_Z^D)$ measurement based on the resource state. There exists an effective spacetime error in the form*

$$e_{Z,eff}^m = \begin{pmatrix} 0 & 0 & e_{Z,C,eff}^m \end{pmatrix} \tag{G10}$$

where

$$e_{Z,C,eff}^m = \begin{pmatrix} 0 & 0 & u_{eff} \end{pmatrix}, \tag{G11}$$

such that the effective error $e_{Z,eff}^m$ is equivalent to the error e_Z^m, i.e. $J_X^m {e_{Z,eff}^m}^{\mathrm{T}} = J_X^m e_Z^{m\mathrm{T}}$, and $|e_{Z,eff}^m| \leq |e_Z^m|$. In the effective spacetime error, errors only occur on block-C at the end.

Proof. In the effective spacetime error, we take u_{eff} according to Eq. (G9). Then, $J_X^m {e_{Z,eff}^m}^{\mathrm{T}} = J_X^m e_Z^{m\mathrm{T}}$ is true, and $|e_{Z,eff}^m| \leq |e_Z^m|$. □

b. Z operators and X errors

The propagation of unmeasured Z operators in spacetime is described by the generator matrix

$$J_Z^m = \begin{pmatrix} J_{Z,A}^m & J_{Z,B}^m & J_{Z,C}^m \end{pmatrix}, \tag{G12}$$

where

$$J_{Z,A}^m = \begin{pmatrix} (H_X^D)^{\mathrm{r}\mathrm{T}} & 0 \\ J_Z^D & 0 \end{pmatrix}, \tag{G13}$$

$$J_{Z,B}^m = \begin{pmatrix} (H_X^D)^{\mathrm{r}\mathrm{T}} & (H_X^D)^{\mathrm{r}\mathrm{T}} \\ J_Z^D & J_Z^D \end{pmatrix}, \tag{G14}$$

and

$$J_{Z,C}^m = \begin{pmatrix} (H_X^D)^{\mathrm{r}\mathrm{T}} & (H_X^D)^{\mathrm{r}\mathrm{T}} & (H_X^D)^{\mathrm{r}\mathrm{T}} \\ J_Z^D & J_Z^D & J_Z^D \end{pmatrix}. \tag{G15}$$

Similar to J_X^m, the generator matrix J_Z^m describes the transformation of operators $Z\left((H_X^D)^{\mathrm{r\,T}}\right)$ and $Z(J_Z^D)$ on block-A into operators $Z\left((H_X^D)^{\mathrm{r\,T}}\right)$ and $Z(J_Z^D)$ on block-C, respectively.

The propagation of measured Z operators in spacetime is described by the generator matrix

$$J_{mz}^m \;=\; \begin{pmatrix} J_{mz,A}^m & J_{mz,B}^m & J_{mz,C}^m \end{pmatrix} \tag{G16}$$

where

$$J_{mz,A}^m \;=\; \begin{pmatrix} H_Z^D & 0 \end{pmatrix}, \tag{G17}$$

$$J_{mz,B}^m \;=\; \begin{pmatrix} H_Z^D & H_Z^D \end{pmatrix}, \tag{G18}$$

and

$$J_{mz,C}^m \;=\; \begin{pmatrix} H_Z^D & H_Z^D & H_Z^D \end{pmatrix}. \tag{G19}$$

This generator matrix describes the transformation of operators $Z(H_Z^D)$ on block-A into operators $Z(H_Z^D)$ on block-C.

Outcome of the $Z(H_Z^D)$ measurement is justified by the following operator propagation,

$$J_{oc}^m \;=\; \begin{pmatrix} J_{oc}^m & J_{oc,B}^m & J_{oc,C}^m \end{pmatrix} \tag{G20}$$

where

$$J_{oc,A}^m \;=\; \begin{pmatrix} H_Z^D & 0 \end{pmatrix}, \tag{G21}$$

$$J_{oc,B}^m \;=\; \begin{pmatrix} H_Z^D & H_Z^D \end{pmatrix}, \tag{G22}$$

and

$$J_{oc,C}^m \;=\; \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}. \tag{G23}$$

Note that the resource state is an eigenstate of operators $Z(H_Z^D)$ on block-B. This generator matrix describes the transformation of operators $Z(H_Z^D)$ on block-A into operators $Z(H_Z^D)$ on block-B, which is subsequently measured in the transversal measurement on block-B.

We represent X errors in spacetime with a vector

$$e_X^m \;=\; \begin{pmatrix} e_{X,A}^m & e_{X,B}^m & e_{X,C}^m \end{pmatrix} \tag{G24}$$

where

$$e_{X,A/B}^m \;=\; \begin{pmatrix} u_{A/B,1} & u_{A/B,2} \end{pmatrix} \tag{G25}$$

and

$$e_{X,C}^m \;=\; \begin{pmatrix} u_{C,1} & u_{C,2} & u_{C,3} \end{pmatrix}. \tag{G26}$$

The errors flip operators and measurement outcomes during the propagation according to

$$J_Z^m e_X^{m\,\mathrm{T}} \;=\; \begin{pmatrix} (H_X^D)^{\mathrm{r\,T}} \\ J_Z^D \end{pmatrix} (u_{A,eff} + u_{C,eff})^{\mathrm{T}}, \tag{G27}$$

$$J_{mz}^m e_X^{m\,\mathrm{T}} \;=\; H_Z^D (u_{A,eff} + u_{C,eff})^{\mathrm{T}}, \tag{G28}$$

$$J_{oc}^m e_X^{m\,\mathrm{T}} \;=\; H_Z^D u_{A,eff}^{\mathrm{T}}, \tag{G29}$$

where

$$u_{A,eff} \;=\; u_{A,1} + u_{B,1} + u_{B,2}, \tag{G30}$$

$$u_{C,eff} \;=\; u_{C,1} + u_{C,2} + u_{C,3}. \tag{G31}$$

Lemma 17. *Let e_X^m be the X spacetime error in the $Z(H_Z^D)$ measurement based on the resource state. There exists an effective spacetime error in the form*

$$e_{X,eff}^m = \begin{pmatrix} e_{X,A,eff}^m & 0 & e_{X,C,eff}^m \end{pmatrix} \tag{G32}$$

where

$$e_{X,A,eff}^m = \begin{pmatrix} u_{A,eff} & 0 \end{pmatrix}, \tag{G33}$$

$$e_{X,C,eff}^m = \begin{pmatrix} 0 & 0 & u_{C,eff} \end{pmatrix}, \tag{G34}$$

such that the effective error $e_{X,eff}^m$ is equivalent to the error e_X^m, i.e.

$$\begin{pmatrix} J_Z^m \\ J_{mz}^m \\ J_{oc}^m \end{pmatrix} {e_{X,eff}^m}^{\mathrm{T}} = \begin{pmatrix} J_Z^m \\ J_{mz}^m \\ J_{oc}^m \end{pmatrix} {e_X^m}^{\mathrm{T}}, \tag{G35}$$

and $|e_{X,eff}^m| \leq |e_X^m|$. In the effective spacetime error, errors only occur on block-A at the beginning and block-C at the end.

Proof. In the effective spacetime error, we take $u_{A,eff}$ and $u_{C,eff}$ according to Eqs. (G30) and (G31). Then, Eq. (G35) is true, and $|e_{X,eff}^m| \leq |e_X^m|$. □

2. Code surgery

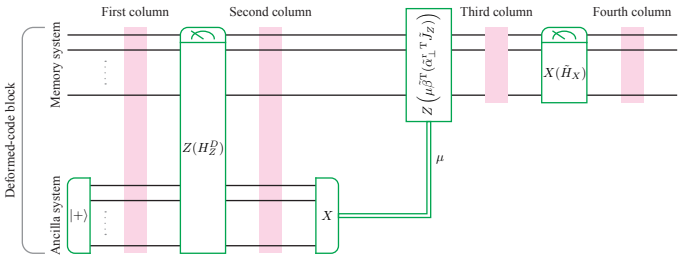

FIG. 14. Code surgery.

The circuit of code surgery is illustrated in Fig. 14, which is applied on k_F blocks of the code (H_X, H_Z, J_X, J_Z), called memory system. Check and generator matrices of the memory system are $(\tilde{H}_X, \tilde{H}_Z, \tilde{J}_X, \tilde{J}_Z)$. This code surgery realizes the measurement of $Z(\tilde{\alpha}\tilde{J}_Z)$.

The input state of the memory system may carry errors. We model them as errors occurring at locations on the memory system in the first column; see Fig. 14. Accordingly, the effective input state (the state of the memory system before the first column) is in the codeword subspace defined by stabilizer operators $X(\tilde{H}_X)$ and $Z(\tilde{H}_Z)$.

We realize the measurements of $Z(H_Z^D)$ and $X(\tilde{H}_X)$ with corresponding resource states. According to Lemmas 16 and 17, errors effectively occur on the input and output states of these measurements, i.e. outcomes of these measurements are effectively error-free; see Fig. 13(b).

a. X operators and Z errors

The propagation of X operators in spacetime is described by the generator matrix

$$J_X^{ls} = \begin{pmatrix} J_{X,M}^{ls} & J_{X,A}^{ls} & J_{X,mea}^{ls} \end{pmatrix}, \tag{G36}$$

where

$$J^{ls}_{X,M} \;=\; \begin{pmatrix} \tilde{\alpha}_\perp \tilde{J}_X & \tilde{\alpha}_\perp \tilde{J}_X & \tilde{\alpha}_\perp \tilde{J}_X & \tilde{\alpha}_\perp \tilde{J}_X \end{pmatrix} \tag{G37}$$

and

$$J^{ls}_{X,A} \;=\; \begin{pmatrix} \tilde{\beta} & \tilde{\beta} \end{pmatrix} \tag{G38}$$

correspond to spacetime locations on the memory and ancilla systems, respectively, and

$$J^{ls}_{X,mea} \;=\; 0 \tag{G39}$$

corresponds to the measurement of $X(\tilde{H}_X)$. The columns in $J^{ls}_{X,M}$ and $J^{ls}_{X,A}$ correspond to spacetime locations as illustrated in Fig. 14. This generator matrix describes that operators $X(\tilde{\alpha}_\perp \tilde{J}_X) \otimes X(\tilde{\beta})$ on two systems at the beginning are transformed into operators $X(\tilde{\alpha}_\perp \tilde{J}_X)$ on the memory system at the end. Note that operators $X(\tilde{\alpha}_\perp \tilde{J}_X) \otimes X(\tilde{\beta}) = X(J^D_X)$ commute with the measurement of $Z(H^D_Z)$. Because qubits in the ancilla system are prepared in the state $|+\rangle$, such a process effectively implements a trivial transformation on operators $X(\tilde{\alpha}_\perp \tilde{J}_X)$.

We detect errors in the following way. Let $\tilde{\nu} \in \mathbb{F}_2^{k_F r_X}$ be the measurement outcome of $X(\tilde{H}_X)$. We introduce the vector

$$\nu \;=\; \begin{pmatrix} \tilde{\nu} + \mu \tilde{T}^{\mathrm{T}} & \mu \tilde{H}_M^{\mathrm{T}} \end{pmatrix}. \tag{G40}$$

Then, $(-1)^{\nu_j}$ is the eigenvalue of the operator $X\left((H^D_X)_{j,\bullet}\right)$. Note that operators $X(\tilde{H}_X)$ $[X(H^D_X)]$ commute with the feedback gate [measurement of $Z(H^D_X)$]. Therefore, we have $\nu = 0$ if the circuit is error-free. Accordingly, the check matrix is

$$H^{ls}_X \;=\; \begin{pmatrix} H^{ls}_{X,M} & H^{ls}_{X,A} & H^{ls}_{X,mea} \end{pmatrix}, \tag{G41}$$

where

$$H^{ls}_{X,M} \;=\; \begin{pmatrix} \tilde{H}_X & \tilde{H}_X & \tilde{H}_X & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{G42}$$

$$H^{ls}_{X,A} \;=\; \begin{pmatrix} \tilde{T} & \tilde{T} \\ \tilde{H}_M & \tilde{H}_M \end{pmatrix}, \tag{G43}$$

and

$$H^{ls}_{X,mea} \;=\; \begin{pmatrix} E_{k_F r_X} \\ 0 \end{pmatrix}. \tag{G44}$$

We represent Z errors in spacetime with the vector

$$e^{ls}_Z \;=\; \begin{pmatrix} e^{ls}_{Z,M} & e^{ls}_{Z,A} & 0 \end{pmatrix} \tag{G45}$$

represents errors before the $X(\tilde{H}_X)$ measurement, where

$$e^{ls}_{Z,M} \;=\; \begin{pmatrix} u_{M,1} & u_{M,2} & u_{M,3} & u_{M,4} \end{pmatrix} \tag{G46}$$

and

$$e^{ls}_{Z,A} \;=\; \begin{pmatrix} u_{A,1} & u_{A,2} \end{pmatrix}. \tag{G47}$$

We decompose this spacetime error into two components:

$$e^{ls-b}_Z \;=\; \begin{pmatrix} u_{M,1} & u_{M,2} & u_{M,3} & 0 & u_{A,1} & u_{A,2} & 0 \end{pmatrix} \tag{G48}$$

represents errors occurring before the $X(\tilde{H}_X)$ measurement, and

$$e_Z^{ls-a} \;=\; \begin{pmatrix} 0 & 0 & 0 & u_{M,4} & 0 & 0 & 0 \end{pmatrix} \tag{G49}$$

represents errors occurring after the $X(\tilde{H}_X)$ measurement. Note that $e_Z^{ls} = e_Z^{ls-b} + e_Z^{ls-a}$. The errors flip operators during the propagation according to

$$J_X^{ls} {e_Z^{ls}}^{\mathrm{T}} \;=\; J_X^D u_{eff}^{\mathrm{T}} + (\tilde{\alpha}_\perp \tilde{J}_X) u_{res}^{\mathrm{T}}, \tag{G50}$$

where

$$u_{eff} \;=\; \begin{pmatrix} u_{M,1} + u_{M,2} + u_{M,3} & u_{A,1} + u_{A,2} \end{pmatrix}, \tag{G51}$$

$$u_{res} \;=\; u_{M,4}. \tag{G52}$$

If the spacetime error e_Z^{ls} is undetectable through H_X^{ls}, it satisfies the condition $H_X^{ls} {e_Z^{ls}}^{\mathrm{T}} = 0$. This condition can be rewritten as $H_X^D u_{eff}^{\mathrm{T}} = 0$.

Lemma 18. *Let e_Z^{ls-b} and e_Z^{ls-a} be the spacetime error vectors representing Z errors that occur before and after the $X(\tilde{H}_X)$ measurement, respectively, during code surgery. If errors are undetectable and satisfy $|e_Z^{ls-b}| < d_D$, then the weight of the residual Z errors on the output state of the memory system is upper bounded by $|e_Z^{ls-p}|$.*

Proof. Note that d_D is the distance of the deformed code. Because $|u_{eff}| \leq |e_Z^{ls-b}| < d_D$ and $H_X^D u_{eff}^{\mathrm{T}} = 0$, we have $J_X^D u_{eff}^{\mathrm{T}} = 0$. Then,

$$J_X^{ls} {e_Z^{ls}}^{\mathrm{T}} \;=\; (\tilde{\alpha}_\perp \tilde{J}_X) u_{res}^{\mathrm{T}}, \tag{G53}$$

i.e. the vector u_{res} represents the residual errors, and $|u_{res}| = |e_Z^{ls-a}|$. □

b. Z operators and X errors

The propagation of unmeasured Z operators in spacetime is described by the generator matrix

$$J_Z^{ls} \;=\; \begin{pmatrix} J_{Z,M}^{ls} & J_{Z,A}^{ls} & J_{Z,mea}^{ls} \end{pmatrix}, \tag{G54}$$

where

$$J_{Z,M}^{ls} \;=\; \begin{pmatrix} {\tilde{\alpha}_\perp^{\mathrm{r}}}^{\mathrm{T}} \tilde{J}_Z & {\tilde{\alpha}_\perp^{\mathrm{r}}}^{\mathrm{T}} \tilde{J}_Z & {\tilde{\alpha}_\perp^{\mathrm{r}}}^{\mathrm{T}} \tilde{J}_Z & {\tilde{\alpha}_\perp^{\mathrm{r}}}^{\mathrm{T}} \tilde{J}_Z \end{pmatrix} \tag{G55}$$

and

$$J_{Z,A}^{ls} \;=\; \begin{pmatrix} 0 & 0 \end{pmatrix} \tag{G56}$$

correspond to spacetime locations on the memory and ancilla systems, respectively, and

$$J_{Z,mea}^{ls} \;=\; 0 \tag{G57}$$

corresponds to the measurement of $Z(H_Z^D)$. Note that operators $Z({\tilde{\alpha}_\perp^{\mathrm{r}}}^{\mathrm{T}} \tilde{J}_Z)$ commute with the measurement of $X(\tilde{H}_X)$.

Similarly, the propagation of measured Z operators in spacetime is described by the generator matrix

$$J_{mz}^{ls} \;=\; \begin{pmatrix} J_{mz,M}^{ls} & J_{mz,A}^{ls} & J_{mz,mea}^{ls} \end{pmatrix}, \tag{G58}$$

where

$$J_{mz,M}^{ls} \;=\; \begin{pmatrix} \tilde{\alpha} \tilde{J}_Z & \tilde{\alpha} \tilde{J}_Z & \tilde{\alpha} \tilde{J}_Z & \tilde{\alpha} \tilde{J}_Z \end{pmatrix}, \tag{G59}$$

$$J^{ls}_{mz,A} \;=\; \begin{pmatrix} 0 & 0 \end{pmatrix}, \tag{G60}$$

and

$$J^{ls}_{mz,mea} \;=\; 0. \tag{G61}$$

Outcome of the $Z(\tilde{\alpha}\tilde{J}_Z)$ measurement is described by the following generator matrix,

$$J^{ls}_{oc} \;=\; \begin{pmatrix} J^{ls}_{oc,M} & J^{ls}_{oc,A} & J^{ls}_{oc,mea} \end{pmatrix}, \tag{G62}$$

where

$$J^{ls}_{oc,M} \;=\; \begin{pmatrix} \tilde{\alpha}\tilde{J}_Z & 0 & 0 & 0 \end{pmatrix}, \tag{G63}$$

$$J^{ls}_{oc,A} \;=\; \begin{pmatrix} 0 & 0 \end{pmatrix}, \tag{G64}$$

and

$$J^{ls}_{oc,mea} \;=\; \tilde{\alpha}\tilde{J}_Z\tilde{R}\gamma_2. \tag{G65}$$

Here,

$$\gamma_1 = \begin{pmatrix} E_{k_R r_Z} & 0 \end{pmatrix} \tag{G66}$$

and

$$\gamma_2 = \begin{pmatrix} 0 & E_{r^D_Z - k_R r_Z} \end{pmatrix}. \tag{G67}$$

Note that the measurement of $Z(H^D_Z)$ measures operators $Z(\tilde{\alpha}\tilde{J}_Z)$ because

$$\tilde{\alpha}\tilde{J}_Z\tilde{R}\gamma_2 H^D_Z = \begin{pmatrix} \tilde{\alpha}\tilde{J}_Z & 0 \end{pmatrix}. \tag{G68}$$

Let $\nu \in \mathbb{F}_2^{r^D_Z}$ be the outcome of the $Z(H^D_Z)$ measurement. Then, $\nu\gamma^{\mathrm{T}}_{1/2}$ is the outcome of $Z(\gamma_{1/2}H^D_Z)$ operators, and $\nu(\tilde{\alpha}\tilde{J}_Z\tilde{R}\gamma_2)^{\mathrm{T}}$ is the outcome of $Z(\tilde{\alpha}\tilde{J}_Z)$ operators.

If the circuit is error-free, we have $\nu\gamma^{\mathrm{T}}_1 = 1$. Accordingly, the check matrix is

$$H^{ls}_Z \;=\; \begin{pmatrix} H^{ls}_{Z,M} & H^{ls}_{Z,A} & H^{ls}_{Z,mea} \end{pmatrix}, \tag{G69}$$

where

$$H^{ls}_{Z,M} \;=\; \begin{pmatrix} \tilde{H}_Z & 0 & 0 & 0 \end{pmatrix}, \tag{G70}$$

$$H^{ls}_{Z,A} \;=\; \begin{pmatrix} 0 & 0 \end{pmatrix}, \tag{G71}$$

and

$$H^{ls}_{Z,mea} \;=\; \gamma_1. \tag{G72}$$

We represent X errors in spacetime with the vector

$$e^{ls}_X \;=\; \begin{pmatrix} e^{ls}_{X,M} & e^{ls}_{X,A} & 0 \end{pmatrix} \tag{G73}$$

represents errors before the $X(\tilde{H}_X)$ measurement, where

$$e^{ls}_{X,M} \;=\; \begin{pmatrix} u_{M,1} & u_{M,2} & u_{M,3} & u_{M,4} \end{pmatrix} \tag{G74}$$

and

$$e^{ls}_{X,A} = \begin{pmatrix} u_{A,1} & u_{A,2} \end{pmatrix}. \tag{G75}$$

Similar to Z errors, we decompose this spacetime error into two components:

$$e^{ls-b}_{X} = \begin{pmatrix} u_{M,1} & 0 & 0 & 0 & u_{A,1} & 0 & 0 \end{pmatrix} \tag{G76}$$

represents errors occurring before the $Z(H^D_Z)$ measurement, and

$$e^{ls-a}_{X} = \begin{pmatrix} 0 & u_{M,2} & u_{M,3} & u_{M,4} & 0 & u_{A,2} & 0 \end{pmatrix} \tag{G77}$$

represents errors occurring after the $Z(H^D_Z)$ measurement. The errors flip operators and measurement outcomes during the propagation according to

$$\begin{aligned} J^{ls}_Z {e^{ls}_X}^{\mathrm{T}} &= (\tilde{\alpha}^{\mathrm{r}}_{\perp}{}^{\mathrm{T}} \tilde{J}_Z) u^{\mathrm{T}}_{eff} + (\tilde{\alpha}^{\mathrm{r}}_{\perp}{}^{\mathrm{T}} \tilde{J}_Z) u^{\mathrm{T}}_{res}, && \text{(G78)} \\ J^{ls}_{mz} {e^{ls}_X}^{\mathrm{T}} &= (\tilde{\alpha} \tilde{J}_Z) u^{\mathrm{T}}_{eff} + (\tilde{\alpha} \tilde{J}_Z) u^{\mathrm{T}}_{res}, && \text{(G79)} \\ J^{ls}_{oc} {e^{ls}_X}^{\mathrm{T}} &= (\tilde{\alpha} \tilde{J}_Z) u^{\mathrm{T}}_{eff}, && \text{(G80)} \\ & && \text{(G81)} \end{aligned}$$

where where

$$\begin{aligned} u_{eff} &= u_{M,1}, && \text{(G82)} \\ u_{res} &= u_{M,2} + u_{M,3} + u_{M,4}. && \text{(G83)} \end{aligned}$$

If the spacetime error e^{ls}_X is undetectable through H^{ls}_Z, it satisfies the condition $H^{ls}_Z {e^{ls}_X}^{\mathrm{T}} = 0$. This condition can be rewritten as $\tilde{H}_Z u^{\mathrm{T}}_{eff} = 0$.

Lemma 19. *Let e^{ls-b}_X and e^{ls-a}_X be the spacetime error vectors representing X errors that occur before and after the $Z(H^D_Z)$ measurement, respectively, during code surgery. If errors are undetectable and satisfy $|e^{ls-b}_X| < d$, then the measurement outcome of $Z(\tilde{\alpha}\tilde{J}_Z)$ is correct, and the weight of the residual X errors on the output state of the memory system is upper bounded by $|e^{ls-p}_X|$.*

Proof. Note that d is the distance of the code (H_X, H_Z, J_X, J_Z), which is also the distance of the code $(\tilde{H}_X, \tilde{H}_Z, \tilde{J}_X, \tilde{J}_Z)$. Because $|u_{eff}| \leq |e^{ls-b}_X| < d$ and $\tilde{H}_Z u^{\mathrm{T}}_{eff} = 0$, we have $\tilde{J}_Z u^{\mathrm{T}}_{eff} = 0$. Then,

$$\begin{aligned} J^{ls}_Z {e^{ls}_X}^{\mathrm{T}} &= (\tilde{\alpha}^{\mathrm{r}}_{\perp}{}^{\mathrm{T}} \tilde{J}_Z) u^{\mathrm{T}}_{res}, && \text{(G84)} \\ J^{ls}_{mz} {e^{ls}_X}^{\mathrm{T}} &= (\tilde{\alpha} \tilde{J}_Z) u^{\mathrm{T}}_{res}, && \text{(G85)} \\ J^{ls}_{oc} {e^{ls}_X}^{\mathrm{T}} &= 0, && \text{(G86)} \\ & && \text{(G87)} \end{aligned}$$

i.e. the measurement outcome is correct, and the vector u_{res} represents the residual errors. Regarding the weight, $|u_{res}| \leq |e^{ls-a}_X|$. □

Appendix H: THRESHOLD THEOREM

In this section, we prove the threshold theorem for our protocol. We begin by presenting the noise model used throughout this work, in which every physical location in the circuit is subject to potential errors. Next, we perform partial circuit reduction to obtain simplified error models in which errors occur only at the boundaries of subcircuits, as described in Appendices F and G. These reduced models are then used to analyze the effective errors, as illustrated in Figs.11, 13 and 14. Finally, based on the lemmas from Appendices F and G, we establish and prove the threshold theorem for our protocol.

1. Noise Model

Definition 2. Quantum LDPC codes. A quantum CSS code $Q = \mathrm{CSS}(H_X, H_Z)$ is said to be an (r, c)-qLDPC code if the row weight of both H_X and H_Z is at most r, and the column weight of both is at most c.

To protect an original quantum circuit C against noise during implementation, we encode it into a fault-tolerant circuit C^{FT} using quantum LDPC (qLDPC) codes. We model a fault-tolerant circuit as a sequence of physical operations, each occurring at a distinct *location*. A location may correspond to a single-qubit gate, a two-qubit gate, the preparation or measurement of a single physical qubit, or simply a wait operation.

In our analysis, we must account for errors that can occur at the physical level. A widely adopted and practically relevant noise model for fault-tolerant quantum circuits is the *local stochastic Pauli error model*. We define a *faulty location* as one at which a local stochastic Pauli error occurs. Locations without faults perform their intended operations correctly, while faulty locations behave incorrectly due to the presence of errors.

Definition 3. Local Stochastic Error Model. Let C be a quantum circuit (and, by abuse of notation, also the set of its physical locations). A set of faulty locations is specified by a random variable $F \subseteq C$, where arbitrary physical errors may occur at these locations (such a set F is referred to as a *fault path*). The local stochastic error model specifies that for any subset $S \subseteq C$, the probability that S is contained in the random fault set F satisfies:

$$\Pr[S \subseteq F] \leq \prod_{i \in S} p_i, \tag{H1}$$

where each physical location $i \in C$ has an associated error parameter p_i with $p_i \in (0, 1]$.

Since any Pauli error can be decomposed into X and Z components, it suffices to consider X-type and Z-type errors in our analysis. A Y-type error is equivalent to the simultaneous occurrence of an X-type and a Z-type error at the same location, and is therefore covered by analyzing these two components.

Assumption 1. *Local Stochastic σ Error Model.* *We assume that C^{FT} is subject to a Local Stochastic σ Error Model with error probability p_i for all locations $i \in C^{\mathrm{FT}}$, where $\sigma = \{X, Z\}$. This means that the error model can be regarded as the combination of two independent models: a Local Stochastic X Error Model and a Local Stochastic Z Error Model. In the first model, each physical location $i \in C^{\mathrm{FT}}$ suffers an X-type error with probability p_i, and in the second model, each physical location suffers a Z-type error with the same probability p_i. The two types of errors occur independently, and a simultaneous occurrence corresponds to a Y-type error at that location.*

We use $(C, \{p_i\}_{i \in C})$ to indicate that the physical errors in the circuit C are distributed according to the locally stochastic σ error model. This model can be straightforwardly generalized to allow for different error probabilities at different types of locations or at different time steps in the circuit C_{FT}. In the following, we assume that the error parameter at each location is upper bounded by a uniform parameter $p_{\mathrm{phy}} \in (0, 1]$, i.e., $p_i \leq p_{\mathrm{phy}}$ for all $i \in C$.

2. Partial circuit reduction

Partial circuit reduction is based on propagating Pauli errors through each layer of the Clifford-gate circuit, thereby pushing these errors toward the layer boundaries. This procedure enables the replacement of certain faulty portions of the circuit with equivalent error-free subcircuits, as is shown in FIG. 15. In what follows, we introduce the concept of *circuit reducibility*, which forms the basis for *partial circuit reduction* used in our proofs. Circuit reducibility permits the simplification of an *incomplete circuit*, defined as a contiguous segment of the full fault-tolerant circuit. We define this simplification method as *partial circuit reduction*.

For clarity, we first establish notation. For any set C, we denote its *power set*—the set of all subsets of C—by 2^C. We further define a binary-valued function $m : 2^C \to \{0, 1\}$, which serves as a *failure flag*. If $m(F) = 1$, this signals a decoding failure or a similar error, in which case the circuit's output may follow an arbitrary probability distribution.

Definition 4. (Definition. 13 in [17]) We say that a pair $(C, \{p_i\}_{i \in C})$ of an incomplete circuit C and a set $\{p_i\}_{i \in C}$ of error parameters for C is δ-reducible to another pair $(C', \{q_j\}_{j \in C'})$ of an incomplete circuit C' and a set $\{q_j\}_{j \in C'}$ of error parameters for C' if there exist maps $\Gamma : 2^C \to 2^{C'}$ and $m : 2^C \to \{0, 1\}$ satisfying the following conditions.

1. For any probability distribution of $F \in 2^C$ satisfying

$$\forall A \subseteq C, \mathbb{P}[F \supseteq A] \leq \prod_{i \in A} p_i, \tag{H2}$$

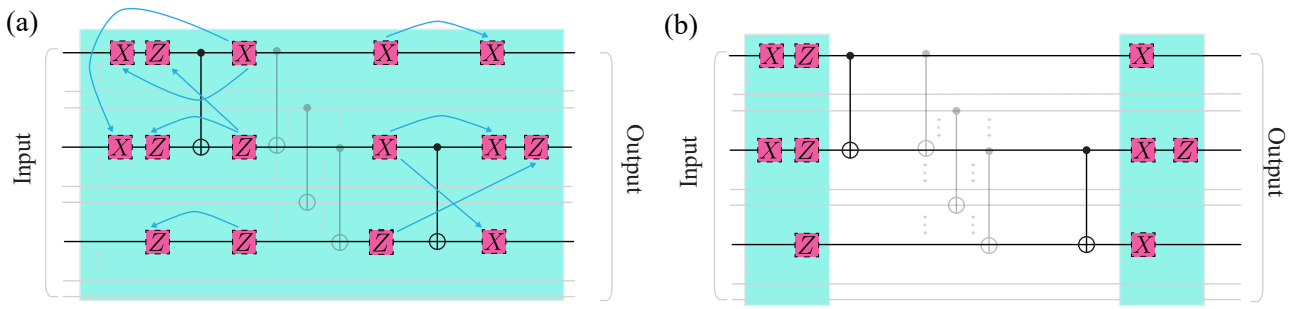

FIG. 15. Illustration of partial circuit reduction. The green shaded regions indicate locations in the circuit where faults may occur. (a) Error propagation through the Clifford-gate circuit: the arrows indicate how Pauli errors move after passing through CNOT gates or idle (identity) operations. (b) After applying partial circuit reduction, faults are confined to the boundaries of the circuit, i.e., only at the beginning and end of the circuit, while the intermediate operations are effectively noisy-free.

it hold that

$$\mathbb{P}[m[F] = 0] \geq 1 - \delta, \tag{H3}$$

and

$$\forall A' \subseteq C', \mathbb{P}[m(F) = 0 \text{ and } \Gamma(F) \supseteq A'] \leq \prod_{j \in A'} q_j. \tag{H4}$$

2. For any set $F \subseteq C$ of faulty locations in C with $m(F) = 0$, and any assignment of Pauli errors in F, there exists an assignment of Pauli errors in the set $\Gamma(F) \subseteq C'$ of faulty locations in C' such that C with the errors in F and C' with the errors in $\Gamma(F)$ are equivalent.

Lemma 20. *(Lemma. 15 in [17]) If $(C, \{p_i\}_{i\in C})$ is δ-reducible to $(C', \{q_j\}_{j\in C'})$, then for any $(C_0, \{r_k\}_{k\in C_0})$, $(C \cup C_0, \{p_i\}_{i\in C} \cup \{r_k\}_{k\in C_0})$ is δ-reducible to $(C' \cup C_0, \{q_j\}_{j\in C'} \cup \{r_k\}_{k\in C_0})$.*

Lemma 21. *(Lemma. 16 in [17]) Consider a pair $(C, \{p_i\}_{i\in C})$ for which C is partitioned into two disjoint subsets C_1 and C_2, and*

$$\forall i \in C_1, \quad p_i \leq p^{(1)} \in [0, 1], \tag{H5}$$
$$\forall i \in C_2, \quad p_i \leq p^{(2)} \in [0, 1] \tag{H6}$$

holds. The pair $(C, \{p_i\}_{i\in C})$ is reducible (i.e., δ-reducible with $\delta = 0$) to another pair $(C', \{q_j\}_{j\in C'})$ if there exists a map $\Gamma : 2^C \to 2^{C'}$ and positive integers $c_{fw}, c_{bw}^{(1)}$, and $c_{bw}^{(2)}$ satisfying the following conditions.

1. It holds that

$$\Gamma(A) = \bigcup_{i\in A} \Gamma(\{i\}), \tag{H7}$$
$$\Gamma(\emptyset) = \emptyset. \tag{H8}$$

2. For all $i \in C$,

$$|\Gamma(\{i\})| \leq c_{fw}. \tag{H9}$$

3. For all $j \in C'$,

$$|\{i \in C_1 : j \in \Gamma(\{i\})\}| \leq c_{bw}^{(1)}, \tag{H10}$$
$$|\{i \in C_2 : j \in \Gamma(\{i\})\}| \leq c_{bw}^{(2)}. \tag{H11}$$

4. *Define*

$$C'_{faulty} := \bigcup_{i \in C : p_i \neq 0} \Gamma(\{i\}). \tag{H12}$$

For all $j \in C'_{faulty}$,

$$q_j \geq 2^{c_{bw}^{(2)}} \left(1 + \frac{p^{(1)}}{p^{(2)}}\right)^{c_{bw}^{(1)}} (p^{(2)})^{1/c_{fw}} + 2^{c_{bw}^{(1)}} (p^{(1)})^{1/c_{fw}}; \tag{H13}$$

and for all $j \in C' \setminus C'_{faulty}$,

$$q_j = 0. \tag{H14}$$

5. *For any set* $A \subseteq C$ *of faulty locations in* C *and any assignment of Pauli errors in* A*, there exists an assignment of Pauli errors in the set* $\Gamma(A) \subseteq C'$ *such that* C *with the errors in* A *and* C' *with the errors in* $\Gamma(A)$ *are equivalent.*

Lemma 22. Wait operation. *In a quantum circuit* C*, two sequential idle operation layers, each exhibiting a local stochastic Pauli error with parameters bounded by* $p_1 \in [0, 1)$ *and* $p_2 \in [0, 1)$*, respectively, are equivalent to a single idle operation layer with a local stochastic Pauli error parameter bounded by* $p_1 + p_2 + p_1 p_2 \leq \min\{2p_1 + p_2,\, p_1 + 2p_2\}$.

Proof. Let A be a subset of k qubits in the circuit, with $k \leq m$ where m is the total number of physical locations in an idle layer. Under the local stochastic Pauli error model (Definition 3), the probability that all k qubits in A are faulty after two sequential idle layers can be bounded by considering three disjoint cases for each qubit:

1. errors in both layers: probability $\leq p_1 p_2$,
2. errors only in the first layer: probability $\leq p_1$,
3. errors only in the second layer: probability $\leq p_2$.

If i qubits fall into case (1), j into case (2), and $k - i - j$ into case (3), then

$$\Pr[S \subseteq F] \leq \prod_{l \in S} p_l \leq (p_1 p_2)^i \, p_1^j \, p_2^{k-i-j} = p_1^{i+j} p_2^{k-j}. \tag{H15}$$

Summing over all possible such configurations, we obtain

$$\Pr[S \subseteq F] \leq \sum_{i=0}^{k} \sum_{j=0}^{k-i} \binom{k}{i} \binom{k-i}{j} p_1^{i+j} p_2^{k-j} \tag{H16}$$

$$= (p_1 + p_2 + p_1 p_2)^k. \tag{H17}$$

which corresponds to a single idle layer with local stochastic error parameter $p_{\text{eq}} = p_1 + p_2 + p_1 p_2$. Finally, since $p_1 p_2 \leq \min\{p_1, p_2\}$ for $p_1, p_2 \in [0, 1)$, we have $p_{\text{eq}} \leq \min\{2p_1 + p_2,\, p_1 + 2p_2\}$.

□

3. Applications of circuit reducibility

The circuit primarily consists of two main components. The first is the preparation of resource states, which serve as essential inputs for both logical operations and error correction. These resource states are subsequently consumed during the second component, *gate teleportation*, which uses them to implement logical gates and perform error correction. Next, we perform partial circuit reduction on the resource state preparation and gate teleportation stages.

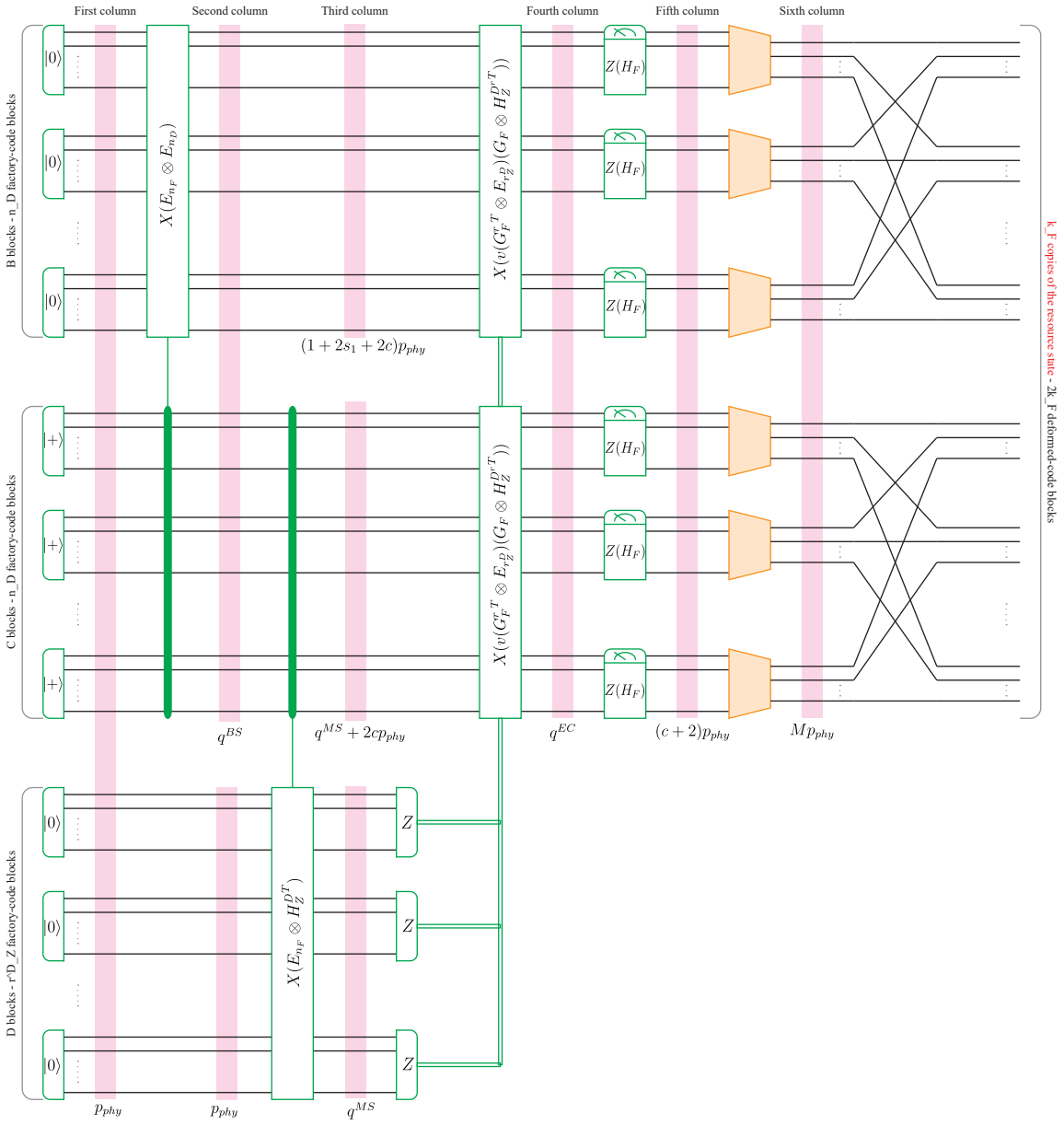

FIG. 16. Circuit reducibility applied to the resource state preparation circuit. The figure shows the equivalent circuit obtained after applying the reduction procedure described in Sec. H 2. Red shaded regions indicate locations where local stochastic σ errors occur, and the parameters shown below each region give upper bounds on the error rates for those locations. These bounds are derived using the partial circuit reduction technique, which propagates Pauli errors through Clifford gates and merges multiple wait operations into equivalent wait operations. A detailed derivation of these bounds is presented in the subsequent analysis.

a. Resource state prepration

As shown in Fig. 16, the resource state preparation circuit can be transformed, via the circuit reducibility procedure described in Sec. H 2, into an equivalent form in which all faults are confined to specific red shaded regions. The parameters shown beneath each red region represent upper bounds on the local stochastic Pauli error rates for the corresponding faulty locations. These bounds are obtained by propagating Pauli errors through Clifford gates and merging idle periods into single wait operations, following the partial circuit reduction method. A detailed derivation of these bounds will be provided in the subsequent analysis.

We first analyze the reducibility of the $Z(H_Z^D)$ measurement operation within the resource state preparation circuit,

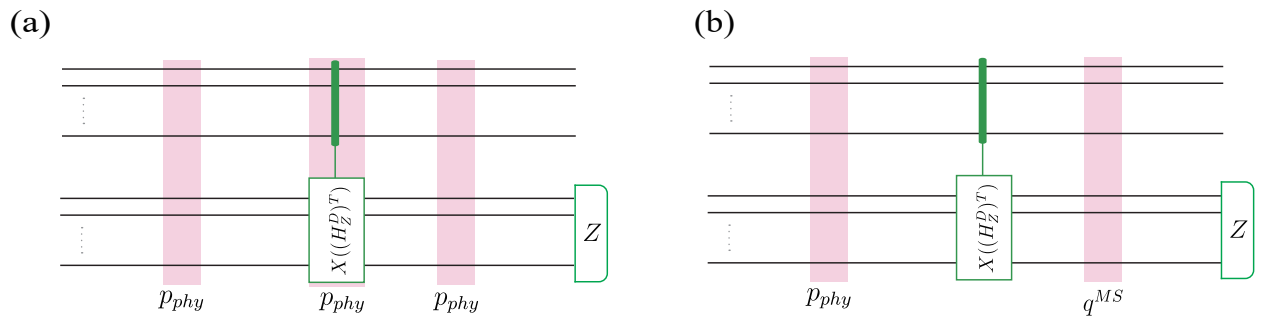

FIG. 17. Reducibility of the $Z(H_Z^D)$ measurement circuit in resource state preparation. (a) Original gadget $C^{Z(H_Z^D)}$ with initial/final wait layers and a syndrome measurement subcircuit of depth s; red regions mark locations where faults may occur. (b) Equivalent reduced circuit via Lemma 21, propagation errors from faulty locations in C^{synd} to those in C_2^{wait}.

as shown in Fig. 17.

Proof. Let $C^{Z(H_Z^D)}$ be a measurement gadget of total depth $2+s_1$, composed of:

1. an initial wait layer C_1^{wait} of depth 1,
2. a syndrome measurement circuit C^{synd} of depth at most s_1, where $s_1 \geq w_{\max}$ and $w_{\max}$ denotes the maximum row (or column) weight of the parity-check matrix H_Z^D.
3. a final wait layer C_2^{wait} of depth 1.

Thus,

$$C^{Z(H_Z^D)} = C_1^{\text{wait}} \cup C^{\text{synd}} \cup C_2^{\text{wait}}, \tag{H18}$$

For all locations $i \in C_1^{\text{wait}} \cup C^{\text{synd}} \cup C_2^{\text{wait}}$, the physical error rate satisfies $p_i \leq p_{\text{phy}}$. We consider Pauli error propagation from faulty locations in C^{synd} to those in C_2^{wait}. To apply Lemma 21, we set $C_1 = C^{\text{synd}}$ and $C_2 = C_2^{\text{wait}}$. After propagation, we partition $C^{\text{synd}} \cup C_2^{\text{wait}}$ into the set of non-faulty locations $C_{\text{nonfaulty}}^{\text{synd}}$ and the set of faulty locations $C_{\text{faulty}}^{\text{synd}}$:

$$C^{\text{synd}} \cup C_2^{\text{wait}} = C_{\text{nonfaulty}}^{\text{synd}} \cup C_{\text{faulty}}^{\text{synd}}. \tag{H19}$$

This error propagation defines a map $\Gamma^{\text{synd}} : 2^{(C^{\text{synd}} \cup C_2^{\text{wait}})} \to 2^{(C_{\text{faulty}}^{\text{synd}})}$. A single-qubit Pauli error propagating through a single-depth layer can expand to at most a two-qubit Pauli error, since each physical gate acts on at most two qubits. Therefore, a Pauli error originating in C^{synd} may propagate to at most 2^{s_1} locations, while an error in C^{wait} does not propagate. Thus,

$$|\Gamma^{\text{synd}}(\{i\})| \leq c_{\text{fw}} = 2^{s_1} \quad \text{for } i \in C^{\text{synd}} \cup C_2^{\text{wait}}. \tag{H20}$$

Moreover, each location in $C_{\text{faulty}}^{\text{synd}}$ can be affected by errors from at most $s_1 2^{s_1}$ locations in C^{synd} and at most $c_{\text{bw}}^{(2)} = 1$ location in C_2^{wait}:

$$|\{i \in C^{\text{synd}} : j \in \Gamma^{\text{synd}}(\{i\})\}| \leq c_{\text{bw}}^{(1)} = s_1 2^{s_1}, \tag{H21}$$

$$|\{i \in C_2^{\text{wait}} : j \in \Gamma^{\text{synd}}(\{i\})\}| \leq c_{\text{bw}}^{(2)} = 1. \tag{H22}$$

Furthermore, under this partition, $p_i \leq p^{(1)} = p_{\text{phy}}$ for all $i \in C^{\text{synd}}$ and $p_i \leq p^{(2)} = p_{\text{phy}}$ for all $i \in C_2^{\text{wait}}$. Therefore, if we choose q_j for all $j \in C_{\text{faulty}}^{\text{synd}}$ as

$$q_j \geq 2^{c_{\text{bw}}^{(2)}} \left(1 + \frac{p^{(1)}}{p^{(2)}}\right)^{c_{\text{bw}}^{(1)}} (p^{(2)})^{1/c_{\text{fw}}} + 2^{c_{\text{bw}}^{(1)}} (p^{(1)})^{1/c_{\text{fw}}} \tag{H23}$$

$$= 2 \cdot 2^{s_1 2^{s_1}} (2p_{\text{phy}})^{1/2^{s_1}} + 2^{s_1 2^{s_1}} (p_{\text{phy}})^{1/2^{s_1}} \tag{H24}$$

$$= q^{\text{MS}}, \tag{H25}$$

then the map Γ^{synd} and the error parameter sets $\{p_i\}_{i\in C^{\text{synd}}\cup C_2^{\text{wait}}}$ and $\{q_j\}_{j\in C^{\text{synd}}_{\text{faulty}}}$ satisfy all conditions in Lemma 21. Thus, $(C^{\text{synd}}\cup C_2^{\text{wait}}, \{p_i\})$ is reducible to $(C^{\text{synd}}_{\text{faulty}}, \{q_j\}_{j\in C^{\text{synd}}_{\text{faulty}}})$ and $(C^{\text{synd}}_{\text{nonfaulty}}, \{0\}_{j\in C^{\text{synd}}_{\text{nonfaulty}}})$. □

After performing the $Z(H_Z^D)$ measurement, Pauli gates are applied based on the measurement outcomes. We can assume each location has an error parameter $p_i \le cp_{\text{phy}}$, and that these errors do not propagate.

Then, we analyze the circuit reducibility of the Bell state preparation operation within the resource state preparation, as shown in Fig. 18.

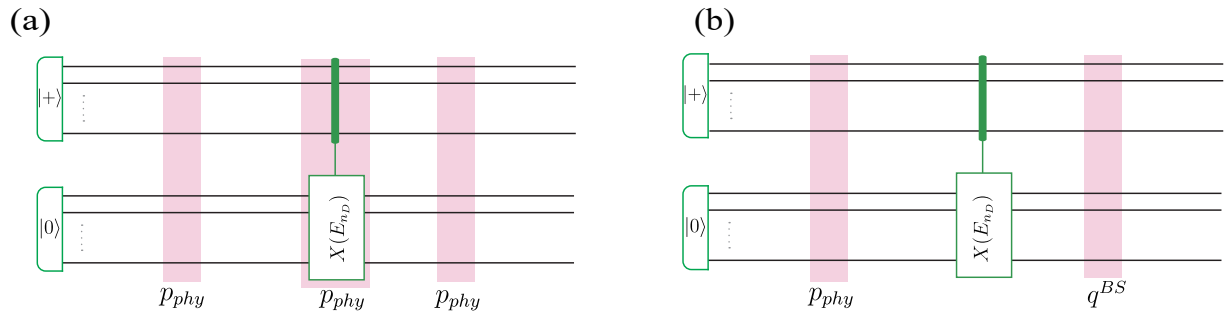

FIG. 18. Reducibility of the Bell state preparation circuit in resource state preparation. (a) Original Bell state gadget with red regions indicating fault locations, each having physical error rate p_{phy}. (b) Equivalent reduced circuit obtained via Lemma 21, mapping faults to $C^{\text{synd}}_{\text{faulty}} = C'_{\text{faulty}}$ with uniform effective error rate q^{BS}.

Proof. The proof follows similarly to the reducibility analysis of the $Z(H_Z^D)$ measurement circuit. For the Bell state preparation, the syndrome subcircuit has depth $s_1 = 1$. We set $p^{(1)} = p_{\text{phy}}$, $p^{(2)} = p_{\text{phy}}$. The forward and backward bounds are $c_{\text{fw}} = 2$, $c_{\text{bw}}^{(1)} = 2$, $c_{\text{bw}}^{(2)} = 1$. Applying Lemma 21, for all $j \in C^{\text{synd}}_{\text{faulty}}$,

$$q_j \ge 2^{c_{\text{bw}}^{(2)}} \left(1 + \frac{p^{(1)}}{p^{(2)}}\right)^{c_{\text{bw}}^{(1)}} (p^{(2)})^{1/c_{\text{fw}}} + 2^{c_{\text{bw}}^{(1)}} (p^{(1)})^{1/c_{\text{fw}}} \tag{H26}$$

$$= 2 \cdot (1+1)^2 (p_{\text{phy}})^{1/2} + 4(p_{\text{phy}})^{1/2} \tag{H27}$$

$$= 12(p_{\text{phy}})^{1/2} \tag{H28}$$

$$= q^{\text{BS}}. \tag{H29}$$

□

We analyze the reducibility of the $Z(H_Z^F)$ measurement operation within the resource state preparation circuit, shown in Fig. 19. In contrast to the reducibility of $C^{Z(H_Z^D)}$, we will achieve the equivalent circuit by backpropagation of errors.

Proof. Let $C^{Z(H_F)}$ be a measurement gadget of total depth $2 + s_2$, composed of:

1. an initial wait layer C_1^{wait} of depth 1,
2. a syndrome measurement circuit C_1^{synd} of depth at most s_2,
3. a final wait layer C_2^{wait} of depth 1.

Thus,

$$C^{Z(H_F)} = C_1^{\text{wait}} \cup C_1^{\text{synd}} \cup C_2^{\text{wait}}. \tag{H30}$$

For all $i \in C_1^{\text{wait}} \cup C_1^{\text{synd}} \cup C_2^{\text{wait}}$, the physical error rate satisfies $p_i \le p_{\text{phy}}$.

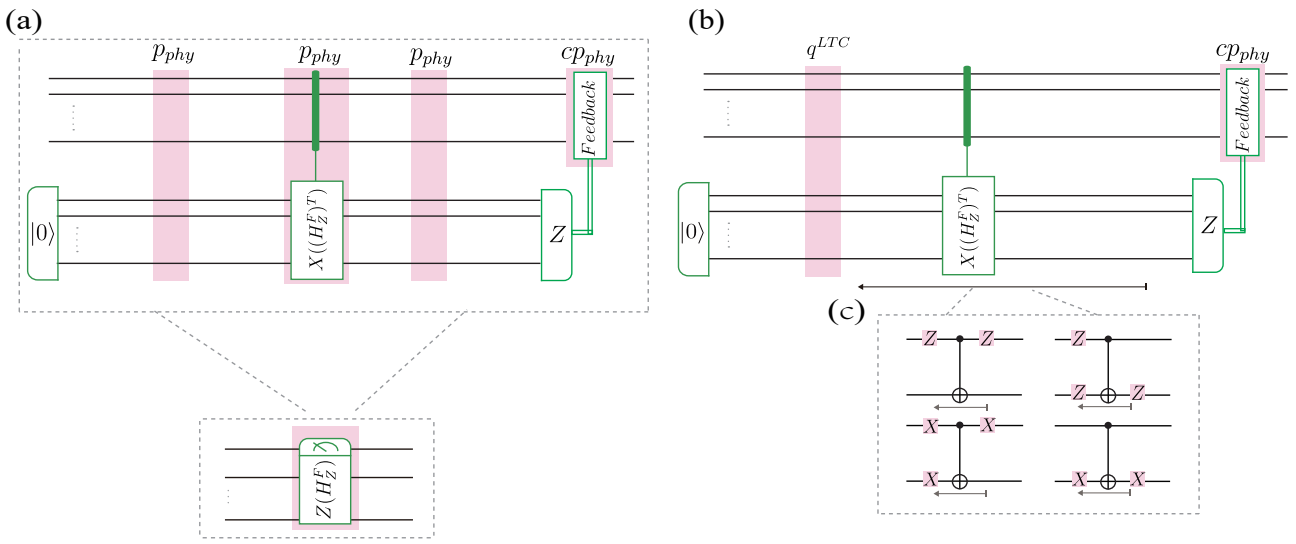

FIG. 19. Reducibility of the error correction gadget $C^{Z(H_F)}$ for the local testable code.(a) Original the error correction gadget $C^{Z(H_F)}$ (b) Equivalent reduced circuit obtained.(c) Illustration of backward propagation of X and Z errors.

We consider Pauli error propagation from faulty locations in $C_1^{\text{synd}} \cup C_2^{\text{wait}}$ to those in C_1^{wait}. In Lemma 21, we set $C_1 = C_1^{\text{synd}} \cup C_2^{\text{wait}}$, $C_2 = C_1^{\text{wait}}$. After propagation, we partition

$$C^{Z(H_F)} = C_{\text{faulty}}^{\text{synd}} \cup C_{\text{nonfaulty}}^{\text{synd}}, \tag{H31}$$

where $C_{\text{faulty}}^{\text{synd}}$ contains all locations reached by propagated errors.

This defines the map $\Gamma^{\text{synd}} : 2^{(C_1^{\text{wait}} \cup C_1^{\text{synd}} \cup C_2^{\text{wait}})} \to 2^{(C_{\text{faulty}}^{\text{synd}})}$. A single-qubit Pauli error in one layer can affect at most two qubits in the next layer. Thus, an error originating in $C_1^{\text{synd}} \cup C_2^{\text{wait}}$ can spread to at most 2^{s_2} locations, giving $c_{\text{fw}} = 2^{s_2}$. Errors in C^{wait} remain localized.

Each location in $C_{\text{faulty}}^{\text{synd}}$ can be influenced by at most $c_{\text{bw}}^{(1)} = (s_2+1)2^{s_2}$ locations in $C_1^{\text{synd}} \cup C_2^{\text{wait}}$, and at most $c_{\text{bw}}^{(2)} = 1$ location in C_1^{wait}:

$$|\{i \in C_1^{\text{synd}} \cup C_2^{\text{wait}} : j \in \Gamma^{\text{synd}}(\{i\})\}| \leq (s_2+1)2^{s_2}, \tag{H32}$$

$$|\{i \in C_1^{\text{wait}} : j \in \Gamma^{\text{synd}}(\{i\})\}| \leq 1. \tag{H33}$$

We set $p^{(1)} = p_{\text{phy}}$ for $i \in C_1^{\text{synd}} \cup C_2^{\text{wait}}$ and $p^{(2)} = p_{\text{phy}}$ for $i \in C_1^{\text{wait}}$. Then, for all $j \in C_{\text{faulty}}^{\text{synd}}$,

$$q_j \geq 2^{c_{\text{bw}}^{(2)}} \left(1 + \frac{p^{(1)}}{p^{(2)}}\right)^{c_{\text{bw}}^{(1)}} (p^{(2)})^{1/c_{\text{fw}}} + 2^{c_{\text{bw}}^{(1)}} (p^{(1)})^{1/c_{\text{fw}}} \tag{H34}$$

$$= 2 \cdot 2^{(s_2+1)2^{s_2}} (p_{\text{phy}})^{1/2^{s_2}} + 2^{(s_2+1)2^{s_2}} (p_{\text{phy}})^{1/2^{s_2}} \tag{H35}$$

$$= q^{\text{LTC}}. \tag{H36}$$

For $j \in C_{\text{nonfaulty}}^{\text{synd}}$, we have $q_j = 0$. Thus, $(C_2^{\text{wait}} \cup C_1^{\text{synd}} \cup C_2^{\text{wait}}, \{p_i\})$ is reducible to $(C_{\text{faulty}}^{\text{synd}}, \{q_j\}_{j \in C_{\text{faulty}}^{\text{synd}}})$ and $(C_{\text{nonfaulty}}^{\text{synd}}, \{0\}_{j \in C_{\text{nonfaulty}}^{\text{synd}}})$.

Following the syndrome measurement, a Pauli feedback operation is applied. Since the exact circuit depth of this feedback stage is not specified, we assume a depth of 1, and assign an error probability of $p_{\text{err}} = c\,p_{\text{phy}}$, to each physical location, where c is a constant. This assumption is justified by the fact that Pauli gates do not propagate errors; consequently, the feedback stage can be modeled as a sequence of idle operations subject to local stochastic σ errors. By applying Lemma 22, we merge any sequence of wait operations, as well as any Pauli operation adjacent to a wait operation, into a single equivalent wait operation. In addition, the remaining Pauli feedback operations are processed in the same way, by merging any Pauli operation followed by a wait operation, as well as any sequence of wait operations, into a single equivalent wait operation. □

Finally, we consider the decoding stage within resource state preparation. Let $C_{SC}^{\mathrm{Dec}} = C^{\mathrm{Dec}} \cup C^{\mathrm{wait}}$ denote a decoding gadget based on the surface code (the factory code). This gadget is an incomplete circuit consisting of a decoding subcircuit C^{Dec} and a set of wait locations C^{wait}.

The decoding process of the surface code is illustrated in Fig. 12. Since C^{Dec} contains only X- or Z-basis measurements and Pauli gates conditioned on the measurement outcomes, any faulty decoding circuit can be transformed into an equivalent fault-free decoding circuit by propagating the resulting Pauli errors to the corresponding faulty locations in C^{wait}.

Assumption 2 (Error Model for Surface Code Decoding Gadget)**.** *Let $C_{SC}^{\mathrm{Dec}} = C^{\mathrm{Dec}} \cup C^{\mathrm{wait}}$ denote a decoding gadget based on the surface code (the underlying code). We assume that this gadget is subject to a local stochastic σ error model, with error probability for each location j given by:*

$$p_j = 0 \quad \textit{if} \quad j \in C^{\mathrm{Dec}}, \tag{H37}$$

$$p_j \leq \lambda p_{\mathrm{phy}} \quad \textit{if} \quad j \in C^{\mathrm{wait}}, \tag{H38}$$

where λ is a constant.

Similarly, for the LTC code (the factory code), let $C_{LTC}^{\mathrm{Dec}} = C^{\mathrm{Dec}} \cup C^{\mathrm{wait}}$ denote the corresponding decoding gadget. We assume:

$$p_j = 0 \quad \text{if} \quad j \in C^{\mathrm{Dec}}, \tag{H39}$$

$$p_j \leq Mp \quad \text{if} \quad j \in C^{\mathrm{wait}}, \tag{H40}$$

where M is a constant that depends on the specific construction of the factory code.

b. gate teleportation

As shown in Figs. 22(a) and 22(b), the errors are propagated from the CNOT gate to the subsequent idle operations. Furthermore, by applying Lemma 22, we obtain the desired reduced subcircuit (Fig. 13).

4. Threshold theorem analysis

In this section, we analyze the probability of logical errors and the residual errors that remain uncorrected after the error correction procedure for both logical Clifford gates (e.g., H, S, and CNOT) and non-Clifford gates (e.g., CCZ) implemented via code surgery. Before proceeding with the analysis, we introduce the notation $Q = (|\phi\rangle, t, p)$ to denote a quantum state $|\phi\rangle$ that satisfies the following two conditions: (1) it has undergone fewer than t Pauli errors, with each error acting on a distinct physical qubit; and (2) it has been subjected to a circuit of wait operations that induce local stochastic Pauli errors, with the error parameter at every location less than p, where $p \in [0, 1)$. This notation is introduced for convenience in the subsequent analysis.

a. Resource state preparation of clifford gate

To implement logical Clifford gates, we perform a sequence of joint measurements as illustrated in Fig. 8. The resource states required for these joint measurements are listed in Table III. We introduce the notation H_σ^{RS} to denote this sequence of resource states. For error correction, we define one H_σ^{RS} as n_R copies of H_Z^M or H_X^M.

Assumption 3. *Let Q be a distance-d surface code. Suppose the code suffers from local stochastic noise with error probability p_1 per qubit initially, error rate p_2 per physical qubit per syndrome extraction, and error rate p_3 for each measured syndrome bit. Assume the syndrome measurement is repeated d times. Then there exists a threshold $\epsilon_{\mathrm{th,s}}$ such that if $\min\{p_1, p_2, p_3\} \leq \epsilon_{\mathrm{th,s}}$, there exists a decoding algorithm such that, after performing logical operations (transversal Clifford gates + error correction), the logical error rate satisfies*

$$p_{\mathrm{L,S}} = \left(\frac{p}{\epsilon_{\mathrm{th,s}}}\right)^{\lfloor \frac{d}{2} \rfloor + 1}. \tag{H41}$$

Theorem 2 (Resource State Preparation for Clifford Gates)**.** *There exists a constant $\epsilon_{\mathrm{rs}} \in (0, 1)$ such that, for resource state preparation, there is a circuit $\mathcal{C}_{\mathrm{rs}}$ of depth $\Theta(d_S)$ and $A_{\mathrm{rs}} = \Theta(n_{\mathrm{RS}} k_F d_S^2)$ locations, satisfying:*

- *In the absence of noise, the circuit outputs k_F resource states H_σ^{RS}.*
- *Under noise, with physical error rate $p_{\mathrm{phy}} \le \epsilon_{\mathrm{rs}}$, the expected output is $Q_{\mathrm{rs}} = \left((H_\sigma^{\mathrm{RS}})^{\otimes k_F},\, t_L,\, \lambda p_{\mathrm{phy}}\right)$, and the failure probability satisfies $\Pr[\mathcal{C}_{\mathrm{rs}}, \mathrm{fail}] \le \frac{k_F\, k_R\, \exp(-\Omega(d))}{12}$.*

Here, n_{RS} denotes the number of physical qubits in the resource state H_σ^{RS}, and $t_L = \min\left\{\frac{d}{16}, \frac{d_F}{2w_Z^D}\right\}$.

Proof. We employ LTSP to prepare these resource states. Consider the noisy case with $p_{\mathrm{phy}} \le \epsilon_{\mathrm{th,s}}$. Each qubit in $\mathcal{C}_{\mathrm{rs}}$ is encoded using the surface code with code distance $d_S = O(\mathrm{polylog}(d))$. By Theorem 3, each logical operation induces a logical error with probability $p_{\mathrm{L,S}} = \left(\frac{p}{\epsilon_{\mathrm{th,s}}}\right)^{\lfloor \frac{d_S}{2} \rfloor + 1}$.

As shown in Fig. 16, the circuit contains $\Theta(n_{\mathrm{RS}}\, k_F)$ logical locations (logical operations implemented using the surface code). The number of fault paths producing exactly t errors is $\binom{\Theta(n_{\mathrm{RS}}\, k_F)}{t}$.

Applying the local stochastic error model (Definition 3), the probability of accumulating more than $t_L/2$ X errors or $t_L/2$ Z errors is bounded by

$$\Pr(\mathcal{C}_{\mathrm{rs}}; \mathrm{fail}) \le \binom{\Theta(n_{\mathrm{RS}} k_F)}{t_{X,M}} \left[c_1 \left(p_{\mathrm{L,S}}\right)^{1/2^{\Delta_1}}\right]^{t_{X,M}} + \binom{\Theta(n_{\mathrm{RS}} k_F)}{t_{Z,M}} \left[c_1 \left(p_{\mathrm{L,S}}\right)^{1/2^{\Delta_1}}\right]^{t_{Z,M}}. \tag{H42}$$

Using the standard binomial bound $\binom{n}{k} \le \left(\frac{ne}{k}\right)^k$, we obtain

$$\Pr(\mathcal{C}_{\mathrm{rs}}; \mathrm{fail}) \le \left(\frac{\Theta(n_{\mathrm{RS}} k_F)\, c_1 e}{\left(1/p_{\mathrm{L,S}}\right)^{1/2^{\Delta_1}}\, t_{X,M}}\right)^{t_{X,M}} + \left(\frac{\Theta(n_{\mathrm{RS}} k_F)\, c_1 e}{\left(1/p_{\mathrm{L,S}}\right)^{1/2^{\Delta_1}}\, t_{Z,M}}\right)^{t_{Z,M}}. \tag{H43}$$

Here, $t_{X,M} = \frac{t_L}{4w_Z^D \max\left\{1, \frac{n_F}{r_F s}\right\}}$ and $t_{Z,M} = \frac{t_L}{4w_Z^D}$.

Therefore, there exists a constant ϵ_{rs} for which, whenever $p_{\mathrm{phy}} \le \epsilon_{\mathrm{rs}}$, the failure probability satisfies $\Pr(\mathcal{C}_{\mathrm{rs}}; \mathrm{fail}) \le \frac{k_R k_F \exp(-\Omega(d))}{12}$.

Here, c_1 and Δ_1 are constants such that $c_1 p_{\mathrm{phy}}^{1/\Delta_1} = \max\left\{M p_{\mathrm{phy}}, q^{\mathrm{LTC}}, q^{\mathrm{BS}}, q^{\mathrm{EC}}, q^{\mathrm{MS}} + 2c\, p_{\mathrm{phy}}\right\} \in [0, 1)$.

Finally, by Assumption 2, unencoding the surface code yields k_F resource states H_σ^{RS} satisfying $Q_{\mathrm{rs}} = \left((H_\sigma^{\mathrm{RS}})^{\otimes k_F},\, t_L,\, \lambda p_{\mathrm{phy}}\right)$, namely:

1. Fewer than t_L Pauli errors, each acting on a distinct physical qubit.
2. Subjected to a wait-operation circuit inducing λp_{phy}-local stochastic Pauli errors.

□

b. *Error analysis during code surgery*

In this section, we use Lemmas 19, 18, 17, and 16 to analyze the error probability of efficient errors at every location during the code surgery process. We first analyze the $Z(H_Z^D)$ measurement circuit, and then consider the case of code surgery, which involves performing both the $Z(H_Z^D)$ and $X(\tilde{H}_X)$ measurements.

Efficient errors during $Z(H_Z^D)$ As shown in Fig. 20(a), the $Z(H_Z^D)$ measurement circuit contains multiple red-shaded regions, each annotated with the maximum error probability per location. Using circuit reducibility [Fig. 20(b)], the error model is simplified by mapping faults to the boundaries of the subcircuit. According to Lemmas 17, Z-type Pauli errors localized within distinct red regions can be equivalently mapped to corresponding faults occurring at the end of Block-C. Similarly, wait operations with local stochastic Z errors from various regions can be mapped to the end of Block-C, as shown in Fig. 20(c).

By Lemma 22, multiple idle-operation layers with local stochastic X errors are equivalent to one such layer, with each location error probability at most $q_{(3)}^{\mathrm{CS}}$, where $q_{(3)}^{\mathrm{CS}}$ is the sum of contributions from all fault regions:

$$q_{(3)}^{\mathrm{CS}} = 2p_{\mathrm{phy}} + 12p_{\mathrm{phy}}^{1/2} + 4(\lambda + 2)p_{\mathrm{phy}} + 6p_{\mathrm{phy}} + 2(c + 2)p_{\mathrm{phy}}. \tag{H44}$$

Similarly, X errors arising from various red regions can be effectively transformed into equivalent errors occurring at the end of Block-C or the begining of Block-A of the circuit. As shown in Fig. 20(d), we have

$$q_{(1)}^{CS} = 2p_{phy} + 12p_{phy}^{1/2} + 2(\lambda + 2)p_{phy} \tag{H45}$$

$$q_{(2)}^{CS} = (\lambda + 2)p_{phy} + 6p_{phy} + 2(c + 2)p_{phy}. \tag{H46}$$

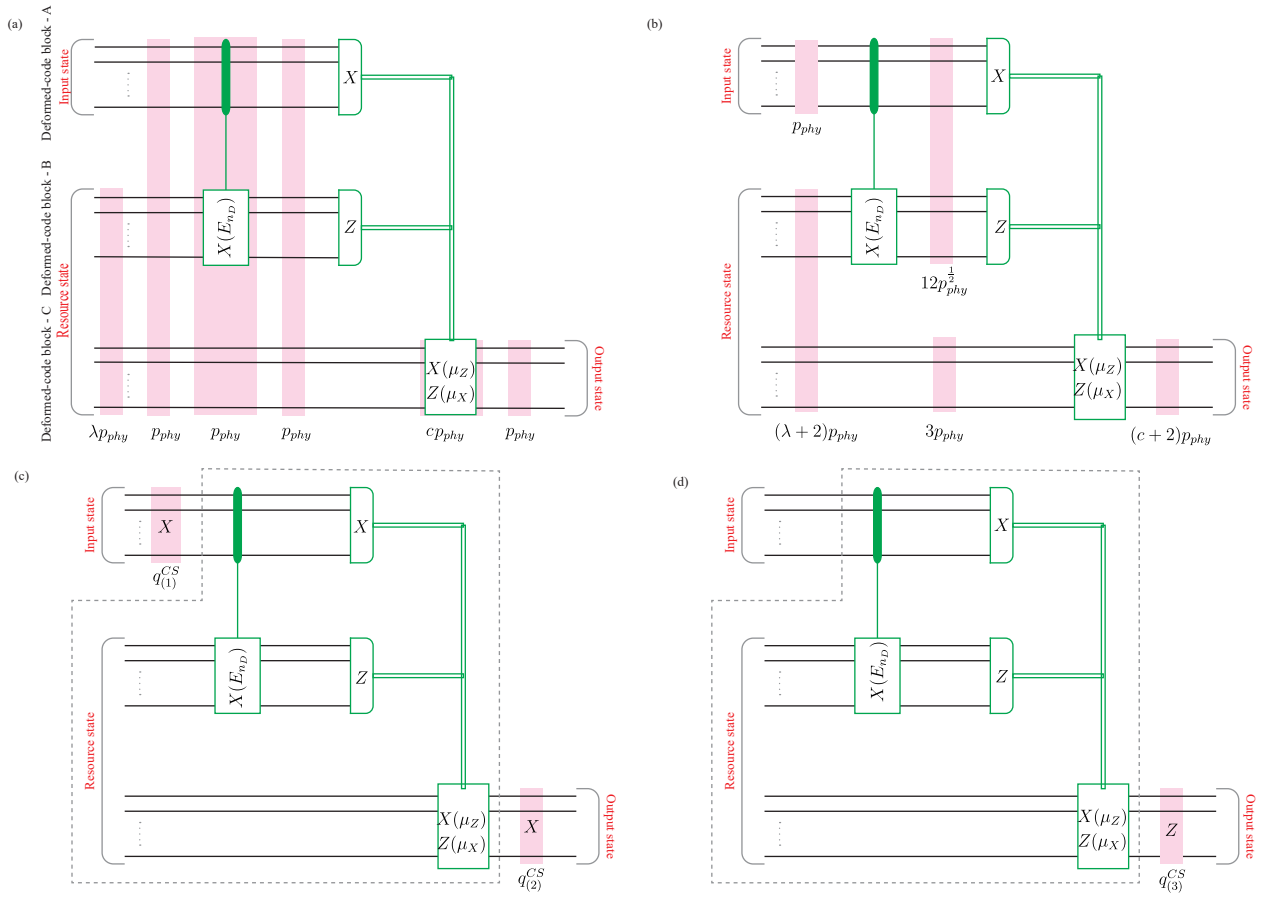

FIG. 20. Effective error models for the $Z(H_Z^D)$ measurement circuit. (a) Original $Z(H_Z^D)$ measurement circuit, where red shaded regions indicate locations subject to local stochastic $\sigma(\sigma = X, Z)$ errors. (b) Reduced form of the circuit obtained via circuit reducibility, in which faults are mapped to the boundaries of the subcircuit. (c) Effective error model for X-type Pauli errors, with $q^{\rm CS}_{(1)}$ and $q^{\rm CS}_{(2)}$ denoting the maximum probability of X errors per location at the beginning of Block A and at the end of Block C, respectively. (d) Effective error model for Z-type Pauli errors, with $q^{\rm CS}_{(3)}$ denoting the maximum probability of Z errors per location at the end of Block C.

Efficient errors during code surgery The analysis of $X(\tilde{H}_X)$ is identical to that of $Z(H_Z^D)$, except that X and Z errors are exchanged.

a. X-type errors. According to Fig. 20, X-type errors effectively occur at the beginning and end of these measurements. We use the corresponding effective X-error model as the error model for $Z(H_Z^D)$ and $X(\tilde{H}_X)$ in code surgery. By reducing adjacent circuits of wait operations into a single one, we obtain the effective error model shown in Fig. 21(b), where:

$$q^{CS}_{(4)} = q^{CS}_{(1)} + 2p_{\rm phy}, \tag{H47}$$
$$q^{CS}_{(5)} = q^{CS}_{(2)} + 2p_{\rm phy}, \tag{H48}$$
$$q^{CS}_{(6)} = q^{CS}_{(3)} + 2p_{\rm phy}. \tag{H49}$$

Using Lemmas. 19, the effective X error model for the case where X errors occur at the end of $X(H_X)$ measurement and at the begin of $Z(H_Z^D)$ measurement is shown in Fig. 21(c), with:

$$q^{CS}_{(7)} = 2q^{CS}_{(5)} + q^{CS}_{(6)} + 2(c+2)p_{\rm phy}. \tag{H50}$$

b. Z-type errors. Following a similar reduction process, we obtain the effective Z error model shown in Fig. 22(c):

$$q^{CS}_{(10)} = q^{CS}_{(2)} + 2p_{\rm phy} \tag{H51}$$
$$q^{CS}_{(11)} = q^{CS}_{(8)} + 2p_{\rm phy} + 2q^{CS}_{(8)}, \tag{H52}$$

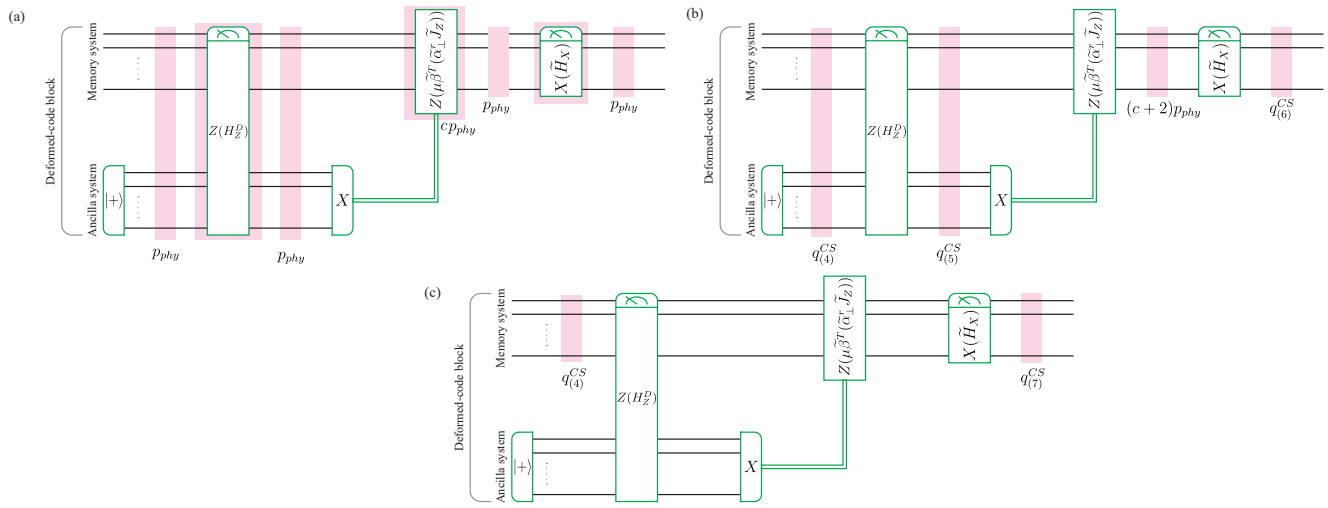

FIG. 21. Effective X-type error models during code surgery, where red-shaded regions indicate locations subject to local stochastic X errors. (a) Initial code-surgery circuit. (b) Reduced circuit obtained using the effective $Z(H_Z^D)$ and $X(\tilde{H}_X)$ error model. Here, $q_{(4)}^{\mathrm{CS}}$, $q_{(5)}^{\mathrm{CS}}$, and $q_{(6)}^{\mathrm{CS}}$ denote the maximum probability of an X error per location for the corresponding red-shaded regions. (c) Final effective X-error model for the case where X errors occur at the end of the $X(H_X)$ measurement and at the beginning of the $Z(H_Z^D)$ measurement, with $q_{(4)}^{\mathrm{CS}}$ and $q_{(7)}^{\mathrm{CS}}$ denoting the maximum probability of an X error per location for the corresponding regions.

where $q_{(8)}^{CS} = q_{(3)}^{CS} + 2p_{\mathrm{phy}}$ and $q_{(9)}^{CS} = q_{(1)}^{CS} + (2c+2)p_{\mathrm{phy}}$.

c. *Code surgry*

Then, based on the previous discussion, we consider the failure probability of code surgery. Note that it is not necessary to ensure that all errors are corrected during the logical operation; we only require that the residual errors are controlled such that they do not cause logical errors in the subsequent process.

a. Measurement protocol. During code surgery, we employ the *Roped-Devised-Sticking* protocol, in which logical operators on n_R memory blocks are measured simultaneously. For each logical operator, a measurement outcome is obtained. We define $\delta_{\mathrm{fail}} = 0$ to indicate that no logical operator has been incorrectly flipped (operation succeeds), whereas $\delta_{\mathrm{fail}} = 1$ denotes the occurrence of such an erroneous flip (operation fails). Furthermore, we use $|\bar{\phi}\rangle_{\mathrm{in}}$ to denote the input state, and $|\bar{\phi}\rangle_{\mathrm{out}}$ to denote the state after the operation (i.e., joint measurement). Both $|\bar{\phi}\rangle_{\mathrm{in}}$ and $|\bar{\phi}\rangle_{\mathrm{out}}$ represent logical states without any Pauli errors, i.e., in the ideal, noise-free setting.

Lemma 23 (Lemma 1 in [15])**.** *Consider a set S of t nodes in a graph where each node has degree at most z. Let $M_z(\mathsf{s},\mathsf{S})$ be the number of sets containing S and in total s nodes ($\mathsf{s}-t$ nodes beyond those in S), which form a union of connected clusters, each containing at least one node from S. Then $M_z(\mathsf{s},\mathsf{S}) \le e^{t-1}(ze)^{\mathsf{s}-t}$, where e is the base of the natural logarithm.*

During resource state preparation, we generate $\big((H_\sigma^{\mathrm{RS}})^{\otimes k_F}, t_L, \lambda p_{\mathrm{phy}}\big)$, which denotes k_F resource states H_σ^{RS} containing at most t_L total errors. Because these errors are not uniformly distributed, we conservatively assume that each resource state may contain up to t_L errors, i.e., $\big(H_\sigma^{\mathrm{RS}}, t_L, \lambda p_{\mathrm{phy}}\big)$. Although this assumption inflates the error count, subsequent analysis confirms that it does not affect the final result.

Theorem 3 (Code Surgery)**.** *There exists a constant $\epsilon_{\mathrm{cs}} \in [0,1)$ such that, for $p_{\mathrm{phy}} \in [0,\epsilon_{\mathrm{cs}})$, there is a gadget $\mathrm{Gad}_{\mathrm{cs}}$ of depth $\Theta(1)$ and $A_{\mathrm{cs}} = \Theta(n_D)$ locations satisfying:*

- *In the noise-free setting, $\mathrm{Gad}_{\mathrm{cs}}$ performs joint measurements on the memory codes.*
- *The input state is $Q_{\mathrm{in}} = (|\bar{\phi}\rangle_{\mathrm{in}}^{\otimes k_R}, 2t_L, c_2 p_{\mathrm{phy}}^{1/\Delta_2})$ and resource states are $Q_\sigma^{RS} = (H_\sigma^{\mathrm{RS}}, t_L, \lambda p_{\mathrm{phy}})$.*
- *Under noise, the expected output is $Q_{CS} = (|\bar{\phi}\rangle_{\mathrm{out}}^{\otimes k_R}, 2t_L, c_2 p_{\mathrm{phy}}^{1/\Delta_2})$ with $\delta_{\mathrm{fail}} = 0$, and the failure probability (i.e., no producing Q_{CS} with $\delta_{\mathrm{fail}} = 1$) satisfies $\Pr[\mathrm{Gad}_{\mathrm{CS}}, \delta_{\mathrm{fail}} = 1] \le \frac{k_R \exp(-\Omega(d))}{12}$.*

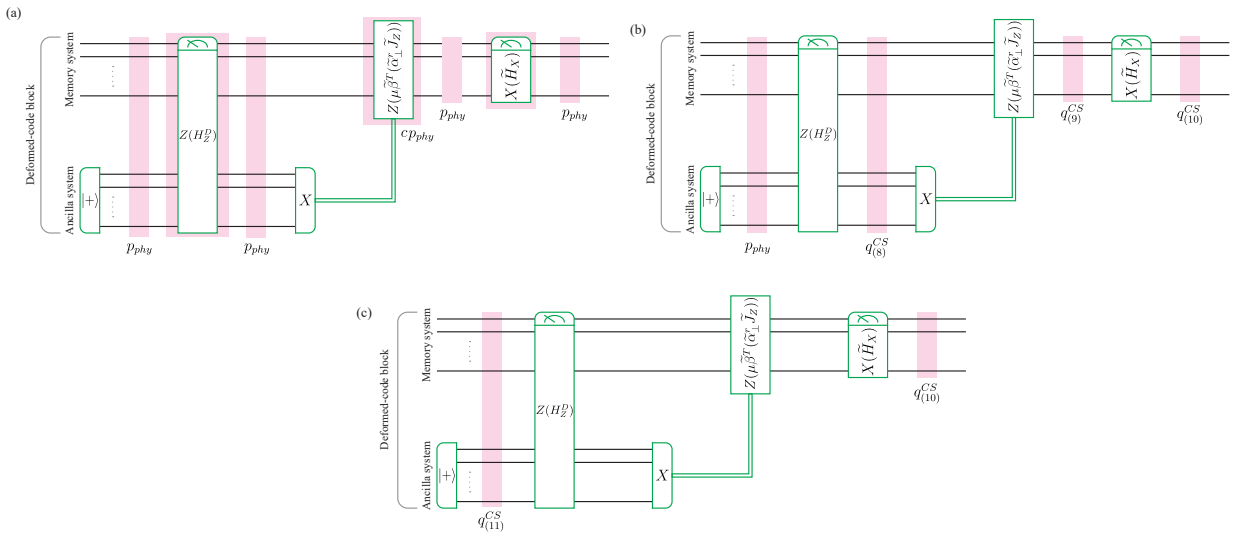

FIG. 22. Effective Z-type error models during code surgery, where red-shaded regions indicate locations subject to local stochastic Z errors. (a) Initial measurement circuit. (b) Reduced circuit using the effective $Z(H_Z^D)$ and $X(\tilde{H}_X)$ error model. (c) Final effective Z error model.

where c_2 *and* Δ_2 *are constant and statisfy* $c_2 p_{\text{phy}}^{1/\Delta_2} = \max\{q^{\text{CS}}_{(7)}, q^{\text{CS}}_{(10)}\}$.

Proof. We largely follow the proof of Theorem 2 in Ref. [15], with minor modifications. A logical error is assumed to occur if and only if there exists a connected cluster of $\mathsf{s} \geq d_D$ qubits containing at least $\lceil \mathsf{s}/2 \rceil$ actual errors in the fault path. We analyze how errors in the input and resource states propagate during the joint measurements, and determine the minimum number of actual errors required to cause a logical failure.

We separately consider two cases in code surgery: (1) a logical X error that causes the $Z(H_Z^D)$ measurement to fail, and (2) a logical Z error that causes the $X(\tilde{H}_X)$ measurement to fail. For each case, we determine the minimum number of actual errors required to produce the corresponding logical failure.

Case 1: $Z(H_Z^D)$ measurement failure due to a logical X error. The input state Q_{in} contains at most $2t_L$ errors, and the resource state contains at most t_L errors. By Lemma 17, during $Z(H_Z^D)$ an X error in the resource state is equivalent to appearing either on block-A at the beginning or on block-C at the end. Consequently, at most t_L X errors from the resource state and $2t_L$ X errors from the input state may be mapped to the beginning of $Z(H_Z^D)$. According to Lemma 18, this mapping is equivalent to the input state having $3t_L$ X errors, forming part of a connected error cluster. Therefore, a cluster must contain at least $\lceil \mathsf{s}/2 \rceil - 3t_L$ actual errors to produce a logical X error and cause the $Z(H_Z^D)$ measurement to fail.

The Z errors in Q_{in} and Q_{out} are equivalent to appearing at the end of the $Z(H_Z^D)$ measurement, with at most $3t_L$ such errors.

Case 2: $X(\tilde{H}_X)$ measurement failure due to a logical Z error. By Lemma 17, an X error in the resource state is equivalent to appearing either on block-A at the beginning or on block-C at the end. Together with the uncorrected $3t_L$ Z errors from the $Z(H_Z^D)$ measurement, there are at most $4t_L$ errors at the beginning of the $X(\tilde{H}_X)$ measurement circuit. According to Lemma 19, this is equivalent to the input state having $4t_L$ Z errors, forming part of a connected error cluster. Therefore, a cluster must contain at least $\lceil \mathsf{s}/2 \rceil - 4t_L$ actual errors to produce a logical Z error.

Using Lemma 22, we combine input-state σ errors with rate $c_2 p_{\text{phy}}^{1/\Delta_2}$ and circuit-induced errors, as shown in Fig. 21 and Fig. 22. The probability of $\delta_{\text{fail}} = 1$ is then bounded by: (i) Z errors with rate $q^{\text{CS}}_{(11)} + 2c_2 p_{\text{phy}}^{1/\Delta_2}$, (ii) X errors with rate $q^{\text{CS}}_{(4)} + 2c_2 p_{\text{phy}}^{1/\Delta_2}$. The first term corresponds to Case 1, and the second term to Case 2. Applying Lemma 23, we

obtain:

$$\begin{aligned}\Pr[\mathrm{Gad}_{\mathrm{cs}},\delta_{\mathrm{fail}}=1] &\le \frac{n\cdot k_R}{ze}\sum_{\mathsf{s}\ge d_D}(2ze)^{\mathsf{s}}(q^{\mathrm{CS}}_{(11)}+2c_2p_{\mathrm{phy}}^{1/\Delta_2})^{\mathsf{s}/2-3t_L}+\frac{n_D}{ze}\sum_{\mathsf{s}\ge d_D}(2ze)^{\mathsf{s}}(q^{\mathrm{CS}}_{(4)}+2c_2p_{\mathrm{phy}}^{1/\Delta_2})^{\mathsf{s}/2-4t_L}\\ &\le \frac{n\cdot k_R}{ze}\frac{(2ze\sqrt{q^{\mathrm{CS}}_{(11)}+2c_2p_{\mathrm{phy}}^{1/\Delta_2}})^{d_D-6t_L}}{(1-2ze\sqrt{q^{\mathrm{CS}}_{(11)}+2c_2p_{\mathrm{phy}}^{1/\Delta_2}})}+\frac{n_D}{ze}\frac{(2ze\sqrt{q^{\mathrm{CS}}_{(4)}+2c_2p_{\mathrm{phy}}^{1/\Delta_2}})^{d_D-8t_L}}{(1-2ze\sqrt{q^{\mathrm{CS}}_{(4)}+2c_2p_{\mathrm{phy}}^{1/\Delta_2}})}\end{aligned}\tag{H53}$$

where z is a constant related to the weight of the memory code. Therefore, there exists a constant ϵ_{cs} for which, whenever $p_{\mathrm{phy}}\le\epsilon_{\mathrm{cs}}$, the failure probability satisfies

$$\Pr[\mathrm{Gad}_{\mathrm{cs}},\delta_{\mathrm{fail}}=1]\le\frac{k_R\exp(-\Omega(d))}{12}.\tag{H54}$$

After performing code surgery, as illustrated in Fig. 21 and Fig. 22, the output still needs to undergo two equivalent wait operations, with effective X- and Z-error probabilities $q^{\mathrm{CS}}_{(7)}$ and $q^{\mathrm{CS}}_{(10)}$, respectively. There exist constants c_2 and Δ_2 such that

$$c_2p_{\mathrm{phy}}^{1/\Delta_2}=\max\{q^{\mathrm{CS}}_{(10)},q^{\mathrm{CS}}_{(7)}\}.\tag{H55}$$

This inequality ensures that the output sbject to local stochastic σ errors with residual σ error probability $c_1p_{\mathrm{phy}}^{1/\Delta_1}$, upper-bound the effective X and Z error probability arising in the code surgery gadget. In addition, there are residual errors from the $Z(H_Z^D)$ and $X(\tilde{H}_X)$ measurements, with an analysis process similar to that described above. Specifically, there are at most t_L errors originating from the resource state of $Z(H_Z^D)$, of which t_L X errors are corrected during the $X(\tilde{H}_X)$ measurement; and at most t_L errors originating from the resource state of $X(\tilde{H}_X)$.

Thus, we have

$$\Pr[\mathrm{Gad}_{\mathrm{CS}},\delta_{\mathrm{fail}}=1]\le\frac{k_R\exp(-\Omega(d))}{12},$$

which ensures that code surgery succeeds while keeping the residual errors within $2t_L$. Moreover, the error parameters at every location of the wait operations are bounded by $c_1p_{\mathrm{phy}}^{1/\Delta_1}$. □

Theorem 4 (Error Correction)**.** *There exists a constant $\epsilon_{\mathrm{ec}}\in(0,1)$ such that, for $p_{\mathrm{phy}}\in[0,\epsilon_{\mathrm{ec}})$, there exists an error correction gadget $\mathrm{Gad}_{\mathrm{EC}}$ of depth $\Theta(1)$ and $A_{\mathrm{EC}}=\Theta(3n_D)$ locations satisfying:*

- *In the noise-free setting, $\mathrm{Gad}_{\mathrm{EC}}$ performs error correction on the memory code blocks.*
- *The input state is $Q_{\mathrm{in}}=(|\bar{\phi}\rangle_{\mathrm{out}},2t_L,c_1p_{\mathrm{phy}}^{1/\Delta_1})^{\otimes n_R}$ and the resource states are $Q_\sigma^{RS}=((H_\sigma^{\mathrm{M}})^{\otimes n_R},t_L,\lambda p_{\mathrm{phy}})$.*
- *Under noise, the expected output is $Q_{\mathrm{EC}}=((|\bar{\phi}\rangle_{\mathrm{out}})^{\otimes n_R},2t_L,c_2\,p_{\mathrm{phy}}^{1/\Delta_2})$, and the failure probability (i.e., no producing Q_{EC} with $\delta_{\mathrm{fail}}=1$) satisfies $\Pr[\mathrm{Gad}_{\mathrm{EC}},\delta_{\mathrm{fail}}=1]\le k_R\exp(-\Omega(d))/4$.*

Proof. We follow the same cluster-based analysis as in Theorem 3.

Case 1: $Z(H_Z^M)$ measurement failure due to a logical X error. The input state contains at most $2t_L$ errors and the resource state contains at most t_L errors. Thus, at most t_L X errors from the resource state and $2t_L$ X errors from the input state may be mapped to the beginning of $Z(H_Z^M)$. According to Lemma 18, this is equivalent to the input state having $3t_L$ X errors, forming part of a connected error cluster. Therefore, a cluster must contain at least $\lceil\mathsf{s}/2\rceil-3t_L$ actual X errors to produce a logical X error.

Case 2: $X(H_X^M)$ measurement failure due to a logical Z error. From Case 1, there are $3t_L$ uncorrected Z errors from the $Z(H_Z^M)$ measurement. Considering at most t_L Z errors from the resource state, there are at most $4t_L$ effective errors at the beginning of $X(H_X^M)$. According to Lemma 19, this is equivalent to the input state having $4t_L$ Z errors, forming part of a connected error cluster. Thus, a cluster must contain at least $\lceil\mathsf{s}/2\rceil-4t_L$ actual Z errors to produce a logical X error.

Using Fig. 20 and Lemma 22, since the input is $(|\bar{\phi}\rangle_{\mathrm{out}},2t_L,c_2p_{\mathrm{phy}}^{1/\Delta_2})$, we obtain $q^{\mathrm{EC}}_{(1)}=q^{\mathrm{CS}}_{(1)}+2c_2p_{\mathrm{phy}}^{1/\Delta_2}$, $q^{\mathrm{EC}}_{(2)}=q^{\mathrm{CS}}_{(3)}+2c_2p_{\mathrm{phy}}^{1/\Delta_2}$, as shown in FIG. 23.

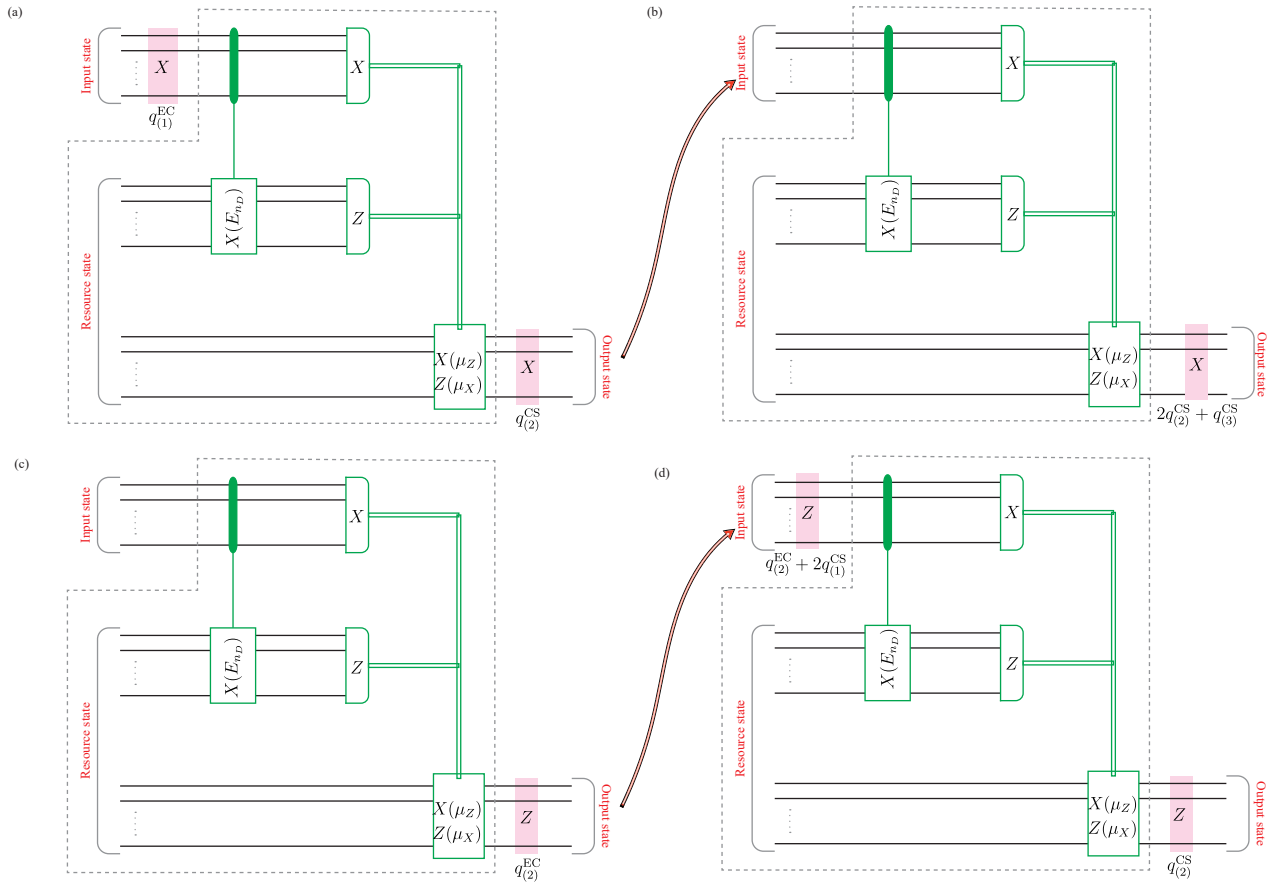

FIG. 23. Error correction involving sequential $Z(H_Z^M)$ and $X(H_X^M)$ measurements. (a) and (b) illustrate the propagation of X errors during the measurement process. (c) and (d) illustrate the propagation of Z errors during the measurement process.

The failure probability is then bounded by

$$\begin{aligned}\Pr[\mathrm{Gad}_{\mathrm{EC}}, \delta_{\mathrm{fail}} = 1] &\le n_R \left[\frac{n}{ze} \sum_{\mathsf{s} \ge d_D} (2ze)^{\mathsf{s}} (q^{\mathrm{EC}}_{(1)})^{\mathsf{s}/2 - 3t_L} + \frac{n}{ze} \sum_{\mathsf{s} \ge d_D} (2ze)^{\mathsf{s}} (q^{\mathrm{EC}}_{(2)})^{\mathsf{s}/2 - 4t_L} \right] \\ &\le n_R \left[\frac{n}{ze} \frac{(2ze\sqrt{q^{\mathrm{EC}}_{(1)}})^{d_D - 6t_L}}{(1 - 2ze\sqrt{q^{\mathrm{EC}}_{(1)}})} + \frac{n}{ze} \frac{(2ze\sqrt{q^{\mathrm{EC}}_{(2)}})^{d_D - 8t_L}}{(1 - 2ze\sqrt{q^{\mathrm{EC}}_{(2)}})} \right] \end{aligned} \tag{H56}$$

Therefore, there exists a constant ϵ_{cs} for which, whenever $p_{\mathrm{phy}} \le \epsilon_{\mathrm{cs}}$, the failure probability satisfies $\Pr[\mathrm{Gad}_{\mathrm{EC}}, \delta_{\mathrm{fail}} = 1] \le \frac{k_R \exp(-\Omega(d))}{4}$.

After error correction, as shown in Fig. 23, the output still needs to undergo two equivalent wait operations, with effective X- and Z-error probabilities $2q^{\mathrm{CS}}_{(2)} + q^{\mathrm{CS}}_{(3)}$ and $q^{\mathrm{CS}}_{(3)}$, respectively. According to Eq. H50, $c_1 p_{\mathrm{phy}}^{1/\Delta_1} \ge 2q^{\mathrm{CS}}_{(2)} + q^{\mathrm{CS}}_{(3)}$. In addition, there are residual errors between the two measurements: at most t_L X errors from Q_Z^{RS} and at most t_L errors from Q_X^{RS}.

Thus, we have $\Pr[\mathrm{Gad}_{\mathrm{EC}}, \delta_{\mathrm{fail}} = 1] \le \frac{k_R \exp(-\Omega(d))}{4}$, ensuring that error correction succeeds while keeping the residual errors within $2t_L$, and that the error parameters are bounded by $c_1 p_{\mathrm{phy}}^{1/\Delta_1}$. □

We now consider a complete joint-measurement procedure. One round of error correction is performed before the code-surgery step, and another round is performed afterwards. In this process, six resource states must be prepared.We consider performing $k_F k_R$ joint-measurement operations in parallel. Combining the above bounds for all constituent

gadgets yields the overall failure probability of the joint measurement:

$$\begin{aligned}\Pr[\mathrm{Gad}_{\mathrm{Clifford}}, \delta_{\mathrm{fail}} = 1] &= 2\, k_F \Pr[\mathrm{Gad}_{\mathrm{EC}}, \delta_{\mathrm{fail}} = 1] + k_F \Pr[\mathrm{Gad}_{\mathrm{CS}}, \delta_{\mathrm{fail}} = 1] + 6 \Pr[\mathcal{C}_{\mathrm{RS}}, \mathrm{fail}] \\ &\leq k_F\, k_R \exp(-\Omega(d)). \end{aligned} \tag{H57}$$

Thus, the average failure probability per joint measurement is $\exp(-\Omega(d))$.

d. *Resource state preparation of magic injection*

In this section, we focus on the preparation of resource states for magic state injection and the probability of successful magic injection. The resource states involved in this process include $H_Z^D(\bar{Z}_S \otimes \bar{Z}_1)$, $(H_X^S)^{\otimes n_R}$, $(H_Z^S)^{\otimes n_R}$, and $(H_Z^M)^{\otimes n_R}$. For notational convenience, we collectively denote them as $(H_\sigma^{\mathrm{S}})^{\mathrm{RS}}$.

Theorem 5 (Resource State Preparation of Magic injection)**.** *There exists a constant $\epsilon_{\mathrm{rsM}} \in (0,1)$ such that, for resource state preparation, there is a circuit $\mathcal{C}_{\mathrm{rsM}}$ of depth $\Theta(d_S)$ and $A_{\mathrm{rsM}} = \Theta(kk_Fk_Rd_S^2)$ locations, satisfying:*

- *In the noise-free setting, the circuit outputs k_F resource states $(H_\sigma^{\mathrm{S}})^{\mathrm{RS}}$.*
- *Under noise, the expected output is $Q_\sigma^{\mathrm{S}} = (((H_\sigma^{\mathrm{S}})^{\mathrm{RS}})^{\otimes k_F}, t_S, \lambda p_{\mathrm{phy}})$, and the failure probability (i.e., no producing Q_σ^{S} with $\delta_{\mathrm{fail}} = 1$) satisfies $\Pr[\mathcal{C}_{\mathrm{rsM}}, \delta_{\mathrm{fail}} = 1] \leq kk_Rk_F \exp(-\Omega(d_S))/4 \leq \frac{\epsilon_{\mathrm{noisy}}}{12}$.*

Here, $t_S = \frac{d_S}{18}$.

Proof. The proof follows the same procedure as in Lemma 2. For the resource states preparation used in magic state injection, we continue to employ LTSP . The circuit contains $\Theta(kk_Fk_R)$ logical locations (logical operations implemented using the surface code). The number of fault paths producing exactly $t_S = \Theta(d_S)$ errors is $\binom{\Theta(kk_Fk_R)}{t_S}$.

Applying the local stochastic error model (Lemma. 3), the probability of accumulating more than $t_S/2$ X errors or $t_S/2$ Z errors is bounded by

$$\begin{aligned}\Pr(\mathcal{C}_{\mathrm{rsM}}; \mathrm{fail}) &\leq \binom{\Theta(n_{\mathrm{RS}}k_F)}{t_{X,S}} \left[c_1 \left(p_{\mathrm{L,S}}\right)^{1/2^{\Delta_1}}\right]^{t_{X,S}} + \binom{\Theta(n_{\mathrm{RS}}k_F)}{t_{Z,S}} \left[c_1 \left(p_{\mathrm{L,S}}\right)^{1/2^{\Delta_1}}\right]^{t_{Z,S}} \\ &\leq \left(\frac{\Theta(n_{\mathrm{RS}}k_F)\, c_1 e}{\left(1/p_{\mathrm{L,S}}\right)^{1/2^{\Delta_1}}\, t_{X,S}}\right)^{t_{X,S}} + \left(\frac{\Theta(n_{\mathrm{RS}}k_F)\, c_1 e}{\left(1/p_{\mathrm{L,S}}\right)^{1/2^{\Delta_1}}\, t_{Z,S}}\right)^{t_{Z,S}}. \end{aligned} \tag{H58}$$

Here, $t_{X,S} = \frac{t_S}{4w_Z^D \max\left\{1, \frac{n_F}{r_F s}\right\}}$ and $t_{Z,S} = \frac{t_S}{4w_Z^D}$. Therefore, there exists a constant ϵ_{rsM} for which, whenever $p_{\mathrm{phy}} \leq \epsilon_{\mathrm{rsM}}$, the failure probability satisfies

$$\Pr(\mathcal{C}_{\mathrm{rsM}}; \mathrm{fail}) \leq kk_Rk_F \exp(-\Omega(d_S))/4 \leq \frac{\epsilon_{\mathrm{noisy}}}{12}. \tag{H59}$$

Finally, by Assumption 2, unencoding the surface code yields k_F logical states $(H_\sigma^{\mathrm{S}})^{\mathrm{RS}}$ satisfying the conditions of Q_σ^{S} :

1. Fewer than t_S Pauli errors, each acting on a distinct physical qubit.
2. Subjected to a wait-operation circuit inducing λp_{phy}-local stochastic Pauli errors.

□

For magic state injection, we require a noisy T-gate magic state $|\bar{T}_{\mathrm{noisy}}\rangle$ that has already been encoded in a surface code of distance d_{S}. Its density matrix can be expressed as

$$\rho_{T,\mathrm{S}} = (1 - p_{\mathrm{err}})|\bar{T}\rangle_{\mathrm{S}}\langle\bar{T}|_{\mathrm{S}} + \epsilon_{\mathrm{noisy}}\, \rho_{\mathrm{error,S}}. \tag{H60}$$

where:

- $|\bar{T}\rangle_{\mathrm{S}}$ is the ideal logical T-gate magic state encoded in the surface code,
- p_{err} is the probability that preparation fails to produce the desired T-gate magic state, in which case the block is in $\rho_{\mathrm{error,S}}$, a (possibly mixed) erroneous logical state encoded in the surface code.

During injection, the input state is

$$Q_{\mathrm{M,in}} = \left(|\bar{T}_{\mathrm{noisy}}\rangle_{\mathrm{S}},\ 0,\ c_3 p_{\mathrm{phy}}\right)^{\otimes n_R}, \tag{H61}$$

where c_3 is a constant and n_R is the number of resource states.

Theorem 6 (Magic injection)**.** *There exists a constant $\epsilon_{\mathrm{M}} \in (0,1)$ such that, for $p_{\mathrm{phy}} \in [0, \epsilon_{\mathrm{M}})$, there is a gadget $\mathrm{Gad}_{\mathrm{M}}$ of depth $\Theta(1)$ and $A_{\mathrm{M}} = \Theta(n_D)$ locations satisfying:*

- *In the absence of noise, $\mathrm{Gad}_{\mathrm{M}}$ prepares a memory-code block encoding a single logical qubit in the state $|\bar{T}\rangle_{\mathrm{M}}$ (ideal logical T-state in the memory code).*
- *The input state is $Q_{\mathrm{M,in}} = (|\bar{T}_{\mathrm{noisy}}\rangle_{\mathrm{S}}, 0, c_3 p_{\mathrm{phy}})^{\otimes k_R}$ and resource states are $Q^{\mathrm{S}}_{\sigma} = (((H^{\mathrm{S}}_{\sigma})^{\mathrm{RS}}), t_S, \lambda p_{\mathrm{phy}})$.*
- *Under noise, the expected output is $Q_{\mathrm{M,out}} = (|\bar{T}\rangle^{\otimes k_R}_{\mathrm{M}}, 2t_{\mathrm{S}}, c_1 p^{1/\Delta_1}_{\mathrm{phy}})$ with $\delta_{\mathrm{fail}} = 0$. And the failure probability (i.e., not producing $Q_{\mathrm{M,out}}$ as specified while $\delta_{\mathrm{fail}} = 1$) satisfies $\Pr[\mathrm{Gad}_{\mathrm{M}}, \delta_{\mathrm{fail}} = 1] \le \epsilon_{\mathrm{noisy}}$.*

Proof. The circuit of the gadget $\mathrm{Gad}_{\mathrm{M}}$ is shown in Fig. 9. We begin by analyzing the preparation of the logical state $|\bar{+}\rangle_{\mathrm{M}}$ in the memory-code block. The physical qubits are first initialized in the $|+\rangle$ state, i.e., the input is $(|+\rangle^{\otimes(n\cdot n_R)}, 0, p_{\mathrm{phy}})$. Following the reasoning in the proof of Lemma 4, we consider the measurement of $Z(H^M_Z)$. Since at most t_{S} errors originate from the resource state, a cluster must contain at least $\lceil s/2 \rceil - t_{\mathrm{S}}$ actual X-type errors to induce a logical X error. Taking into account the input state's intrinsic error probability, the effective X-error rate entering the $Z(H^M_Z)$ measurement is given by $q^{\mathrm{CS}}_{(1)} + 2p_{\mathrm{phy}}$, where $q^{\mathrm{CS}}_{(1)}$ denotes the error contribution from the measurement circuit itself (see Fig. 20). Consequently, the failure probability can be bounded as

$$\Pr[|\bar{+}\rangle_{\mathrm{M}}, \delta_{\mathrm{fail}} = 1] \le n_R \frac{n}{ze} \sum_{\mathsf{s} \ge d_D} (2ze)^{\mathsf{s}} \left(q^{\mathrm{CS}}_{(1)} + 2p_{\mathrm{phy}}\right)^{\mathsf{s}/2 - t_{\mathrm{L}}} \le \frac{k_R \cdot k\, \exp(-\Omega(d_{\mathrm{S}}))}{12}. \tag{H62}$$

Next, we consider performing the $Z(H^S_Z)$ and $X(H^S_X)$ measurements on the surface code or on physical qubits. We refer to this procedure as the gadget $\mathrm{Gad}_{\mathrm{EC},S}$, which is used either to perform error correction on the surface code or to prepare the surface code in the logical state $|\bar{+}\rangle_{\mathrm{S}}$ or $|\bar{0}\rangle_{\mathrm{S}}$. This procedure follows the same analysis as in Lemma 4, with the following key differences:

- The number of input-state errors is zero.
- The resource state contains at most t_{S} errors.
- $q^{\mathrm{EC}}_{(1)} = q^{\mathrm{CS}}_{(1)} + 2c_2 p^{1/\Delta_2}_{\mathrm{phy}}$, $q^{\mathrm{EC}}_{(2)} = q^{\mathrm{CS}}_{(3)} + 2c_2 p^{1/\Delta_2}_{\mathrm{phy}}$, with $c_2 = \max\{c_3, 1\}$ and $\Delta_2 = 0$.
- A connected cluster of size $\mathsf{s} \ge d_{\mathrm{S}}$ is required to produce a logical error.

The probability that the gadget $\mathrm{Gad}_{\mathrm{EC},S}$ fails due to a logical error is

$$\begin{aligned}
\Pr[\mathrm{Gad}_{\mathrm{EC,M}}, \delta_{\mathrm{fail}} = 1] &\le n_R \left[\frac{n}{ze} \sum_{\mathsf{s} \ge d_{\mathrm{S}}} (2ze)^{\mathsf{s}} (q^{\mathrm{EC}}_{(1)})^{\mathsf{s}/2 - 3t_{\mathrm{S}}} + \frac{n}{ze} \sum_{\mathsf{s} \ge d_{\mathrm{S}}} (2ze)^{\mathsf{s}} (q^{\mathrm{EC}}_{(2)})^{\mathsf{s}/2 - 4t_{\mathrm{S}}} \right] \\
&\le n_R \left[\frac{n}{ze} \frac{(2ze\sqrt{q^{\mathrm{EC}}_{(1)}})^{d_{\mathrm{S}} - 6t_{\mathrm{S}}}}{(1 - 2ze\sqrt{q^{\mathrm{EC}}_{(1)}})} + \frac{n}{ze} \frac{(2ze\sqrt{q^{\mathrm{EC}}_{(2)}})^{d_{\mathrm{S}} - 8t_{\mathrm{S}}}}{(1 - 2ze\sqrt{q^{\mathrm{EC}}_{(2)}})} \right] \\
&\le \frac{k_R \cdot k \exp(-\Omega(d_{\mathrm{S}}))}{6}.
\end{aligned} \tag{H63}$$

We then analyze the joint measurement $H^S_Z(\bar{Z}_S \otimes \bar{Z}_1)$ used for magic-state injection. We first consider the code-surgery procedure, i.e., $\mathrm{Gad}_{\mathrm{CS,S}}$, which follows an analysis similar to that in Theorem 3. The key difference is that the input state contains $7t_{\mathrm{S}}$ residual errors from previous operations, of which $6t_{\mathrm{S}}$ originate from the surface-code block and t_{S} from the memory-code block. In addition, the resource state itself contains at most t_{S} errors. Since the

deformed code has code distance $d_{\rm S}$, a connected cluster of size $s \geq d_{\rm S}$ is required to produce a logical error. The probability that the gadget $\mathrm{Gad}_{\rm CS,S}$ fails due to a logical error is

$$\begin{aligned}
\Pr[\mathrm{Gad}_{\rm CS,S}, \delta_{\rm fail}=1] &\leq \frac{n\cdot n_R}{ze}\sum_{\mathsf{s}\geq d_{\rm S}}(2ze)^{\mathsf{s}}(2p_{\rm phy}+c_1 p_{\rm phy}^{1/\Delta_1})^{\mathsf{s}/2-8t_{\rm S}} \\
&\quad + \frac{n_D}{ze}\sum_{\mathsf{s}\geq d_{\rm S}}(2ze)^{\mathsf{s}}(2q_{(4)}^{\rm CS}+c_1 p_{\rm phy}^{1/\Delta_1})^{\mathsf{s}/2-9t_{\rm S}} \\
&\leq \frac{n\cdot n_R}{ze}\frac{(2ze\sqrt{2p_{\rm phy}+c_1 p_{\rm phy}^{1/\Delta_1}})^{d_{\rm S}-16t_{\rm S}}}{(1-2ze\sqrt{2p_{\rm phy}+c_1 p_{\rm phy}^{1/\Delta_1}})} \\
&\quad + \frac{n_D}{ze}\frac{(2ze\sqrt{2q_{(4)}^{\rm CS}+c_1 p_{\rm phy}^{1/\Delta_1}})^{d_{\rm S}-18t_{\rm S}}}{(1-2ze\sqrt{2q_{(4)}^{\rm CS}+c_1 p_{\rm phy}^{1/\Delta_1}})} \\
&\leq \frac{k\cdot k_R\exp(-\Omega(d_{\rm S}))}{3}.
\end{aligned} \tag{H64}$$

It is clear that there exists a constant $\epsilon_{\rm M}\in(0,1)$ such that, for $p_{\rm phy}\in[0,\epsilon_{\rm M})$, the final step in Eq. H62, H63 and H64 holds. Finally, we perform error correction on the memory code using the gadget $\mathrm{Gad}_{\rm EC}$. Its failure probability is denoted by $\Pr[\mathrm{Gad}_{\rm EC},\delta_{\rm fail}=1]$. Note that the memory code contains at most $10t_{\rm S}\ll t_{\rm L}$ errors, and thus does not affect the bound in Theorem 4. Combining the above bounds for all constituent gadgets yields the overall failure probability of magic-state injection:

$$\begin{aligned}
\Pr[\mathrm{Gad}_{\rm M},\delta_{\rm fail}=1] &= \Pr[|\bar{+}\rangle_{\rm M},\delta_{\rm fail}=1] + 3\,\Pr[\mathrm{Gad}_{\rm EC,M},\delta_{\rm fail}=1] \\
&\quad + \Pr[\mathrm{Gad}_{\rm CS,S},\delta_{\rm fail}=1] + \Pr[\mathrm{Gad}_{\rm EC},\delta_{\rm fail}=1]) \\
&\leq \epsilon_{\rm noisy}.
\end{aligned} \tag{H65}$$

Thus, we have $1-\Pr[\mathrm{Gad}_{\rm M},\delta_{\rm fail}=1]\geq 1-\epsilon_{\rm noisy}$, ensuring that error correction succeeds while keeping the residual errors within $2t_{\rm S}$, and that the error parameters at every location of the wait operations are bounded by $c_1 p_{\rm phy}^{1/\Delta_1}$. □

In other words, a logical T-gate magic state can be prepared in the memory-code block with logical error probability $2\epsilon_{\rm noisy}$, matching the physical-level error rate.

Theorem 7 (Theorem 1.4 in [18]). *Let $|\mathrm{CCZ}\rangle=\mathrm{CCZ}|+\rangle^{\otimes 3}$. There exists a distillation protocol (using noiseless Clifford operations and measurements) $\mathrm{MSD}(\varepsilon)$ and a constant noise threshold $\epsilon_{\rm noisy}$ such that the following holds. Upon receiving N (independent) noisy states whose infidelity with $|\mathrm{CCZ}\rangle$ is $\varepsilon_{\rm in}/2\leq\epsilon_{\rm noisy}$, where $N>N(\varepsilon_{\rm MSD})$ is a sufficiently large number, $\mathrm{MSD}(\varepsilon)$ produces $\Theta\Big(\frac{N}{\log^{o(1)}(1/\varepsilon)}\Big)$ states each of which has infidelity at most ε with $|\mathrm{CCZ}\rangle$. Furthermore, the quantum depth and classical depth of $\mathrm{MSD}(\varepsilon)$ are both polyloglog$(1/\varepsilon)$. Here, the $o(1)$ exponent scales as $O\Big(\frac{1}{\log\log\log(1/\varepsilon)}\Big)$.*

Theorem 8 (Main result with vanishing threshold). *Consider a classical-input/classical-output quantum circuit C composed of Clifford and T gates, with size $|C|=WD$, where W denotes the width and $D=\mathrm{poly}(W)$ denotes the depth. There exists a constant $\epsilon_*\in(0,1)$ such that for any $\epsilon_L\in(0,1)$, one can efficiently construct a fault-tolerant version $C_{\rm FT}$ with width $W_{\rm FT}$ and depth $D_{\rm FT}$ satisfying*

$$\frac{W_{\rm FT}}{W} = O_{W\to\infty}(1), \tag{H66}$$

$$\frac{D_{\rm FT}}{D} = O_{W\to\infty}(\log^{a+o(1)}(\frac{|C|}{\epsilon_L})), \tag{H67}$$

where a is a constant associated with the memory code block. If $C_{\rm FT}$ operates under a local stochastic σ-error model with $\sigma\in\{X,Z\}$ and physical error rate $p_{\rm phy}\leq\epsilon_*$ at every location, then the total variation distance between the output distributions of C and $C_{\rm FT}$ is at most ϵ_L.

Proof. Let $\epsilon_*=\min\{\epsilon_{\rm M},\epsilon_{\rm rs},\epsilon_{\rm rsM},\epsilon_{\rm ec},\epsilon_{\rm cs}\}$. We employ the constant-overhead magic state distillation scheme from Ref. [18] to obtain a T-gate magic state with infidelity $\exp(-\Omega(d))$. From Eq. H57, the total variation distance

between the output distributions of C and C_{FT} is

$$\delta = O\left(\frac{|C|}{k_F k_R}\right)(k_F k_R \exp(-\Omega(d))) \tag{H68}$$
$$= O(|C|)\, O\left(\exp(-\Omega(d))\right) \tag{H69}$$
$$= O(|C|)\, O\left(\frac{\epsilon_L}{|C|}\right) \tag{H70}$$
$$\leq \epsilon_L. \tag{H71}$$

From Appendix D, the width of C_{FT} is $O(Mk)$. Since we use a quantum code with constant encoding rate, we have

$$\frac{W_{\mathrm{FT}}}{W} = \frac{O(Mk)}{O(K)} = O_{W\to\infty}(1). \tag{H72}$$

From Theorem 7, in order to prepare M copies of T-gate magic states with infidelity $k\exp(-\Omega(d))$, the required depth is $O\left((d^{o(1)})\,\mathrm{polylog}(d)\right)$. Therefore, we obtain

$$\frac{D_{\mathrm{FT}}}{D} = O(kd_{\mathrm{S}}^3)\, O\left((d^{o(1)})\,\mathrm{polylog}(d)\right) \tag{H73}$$
$$= O_{W\to\infty}(d^{a+o(1)}) = O_{W\to\infty}\left(\log^{a+o(1)}\frac{|C|}{\epsilon_L}\right), \tag{H74}$$

where $k = \Theta(d^a)$ and $o(1)$ exponent roughly scales as $O(\frac{1}{\mathrm{loglog}(d)} + \frac{\mathrm{loglog}(d)}{\log(d)})$. □

[1] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, Physical Review A **52**, R2493 (1995).
[2] A. Y. Kitaev, Quantum computations: algorithms and error correction, Russian Mathematical Surveys **52**, 1191 (1997).
[3] D. Gottesman, Theory of fault-tolerant quantum computation, Physical Review A **57**, 127 (1998).
[4] F. Gaitan, *Quantum Error Correction and Fault Tolerant Quantum Computing* (CRC Press, 2018).
[5] E. Knill, R. Laflamme, and W. H. Zurek, Resilient quantum computation, Science **279**, 342 (1998).
[6] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, SIAM Journal on Computing **38**, 1207 (2008).
[7] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Physical Review A **86**, 032324 (2012).
[8] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, New J. Phys. **14**, 123011 (2012).
[9] J.-P. Tillich and G. Zemor, Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength, IEEE Transactions on Information Theory **60**, 1193 (2014).
[10] P. Panteleev and G. Kalachev, Degenerate quantum LDPC codes with good finite length performance, Quantum **5**, 585 (2021).
[11] N. P. Breuckmann and J. N. Eberhardt, Balanced product quantum codes, IEEE Transactions on Information Theory **67**, 6653 (2021).
[12] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, PRX Quantum **2**, 040101 (2021).
[13] P. Panteleev and G. Kalachev, Asymptotically good quantum and locally testable classical ldpc codes, in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC '22 (ACM, 2022).
[14] I. Dinur, M.-H. Hsieh, T.-C. Lin, and T. Vidick, Good quantum ldpc codes with linear time decoders (2022), arXiv:2206.07750 [quant-ph].
[15] D. Gottesman, Fault-tolerant quantum computation with constant overhead, Quantum Information and Computation **14**, 1339 (2014).
[16] H. Yamasaki and M. Koashi, Time-efficient constant-space-overhead fault-tolerant quantum computation, Nature Physics **20**, 247 (2024).
[17] S. Tamiya, M. Koashi, and H. Yamasaki, Polylog-time- and constant-space-overhead fault-tolerant quantum computation with quantum low-density parity-check codes (2024), arXiv:2411.03683 [quant-ph].
[18] Q. T. Nguyen and C. A. Pattison, Quantum fault tolerance with constant-space and logarithmic-time overheads (2025), arXiv:2411.03632 [quant-ph].
[19] L. Z. Cohen, I. H. Kim, S. D. Bartlett, and B. J. Brown, Low-overhead fault-tolerant quantum computing using long-range connectivity, Science Advances **8**, 10.1126/sciadv.abn1717 (2022).
[20] A. Cross, Z. He, P. Rall, and T. Yoder, Improved qldpc surgery: Logical measurements and bridging codes

(2024), arXiv:2407.18393 [quant-ph].
[21] A. Cowtan and S. Burton, CSS code surgery as a universal construction, Quantum **8**, 1344 (2024).
[22] G. Zhang and Y. Li, Time-efficient logical operations on quantum low-density parity check codes, Physical Review Letters **134**, 070602 (2025).
[23] B. Ide, M. G. Gowda, P. J. Nadkarni, and G. Dauphinais, Fault-tolerant logical measurements via homological measurement, Physical Review X **15**, 021088 (2025).
[24] D. J. Williamson and T. J. Yoder, Low-overhead fault-tolerant quantum computation by gauging logical operators (2024), arXiv:2410.02213 [quant-ph].
[25] G. Zhang, Y. Zhu, X. Yuan, and Y. Li, Constant-overhead magic state injection into qldpc codes with error independence guarantees (2025), arXiv:2505.06981 [quant-ph].
[26] Z. He, A. Cowtan, D. J. Williamson, and T. J. Yoder, Extractors: Qldpc architectures for efficient pauli-based computation (2025), arXiv:2503.10390 [quant-ph].
[27] E. Swaroop, T. Jochym-O'Connor, and T. J. Yoder, Universal adapters between quantum ldpc codes (2025), arXiv:2410.03628 [quant-ph].
[28] A. Cowtan, Z. He, D. J. Williamson, and T. J. Yoder, Parallel logical measurements via quantum code surgery (2025), arXiv:2503.05003 [quant-ph].
[29] N. Baspin, L. Berent, and L. Z. Cohen, Fast surgery for quantum ldpc codes (2025), arXiv:2510.04521 [quant-ph].
[30] I. Dinur, T.-C. Lin, and T. Vidick, Expansion of high-dimensional cubical complexes: with application to quantum locally testable codes, in *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2024) pp. 379–385.
[31] A. Leverrier, V. Londe, and G. Zémor, Towards local testability for quantum coding, Quantum **6**, 661 (2022).
[32] T.-C. Lin and M.-H. Hsieh, c^3-locally testable codes from lossless expanders (2022), arXiv:2201.11369 [cs.IT].
[33] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, Quantum **3**, 128 (2019).
[34] Throughout this paper, when discussing devised sticking, we refer specifically to the case where the logical operators to be measured are (i) expressed in standard form, and (ii) have logical thickness one, meaning they act on mutually disjoint sets of logical qubits [22].
[35] A. Leverrier and G. Zemor, Quantum tanner codes, in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2022) pp. 872–883.
[36] T.-C. Lin and M.-H. Hsieh, Good quantum ldpc codes with linear time decoder from lossless expanders (2022), arXiv:2203.03581 [quant-ph].
[37] M. Sipser and D. Spielman, Expander codes, IEEE Transactions on Information Theory **42**, 1710 (1996).
[38] R. Tanner, A recursive approach to low complexity codes, IEEE Transactions on Information Theory **27**, 533 (1981).
[39] Y. Li, A magic state's fidelity can be superior to the operations that created it, New Journal of Physics **17** (2014).
[40] In a discussion with Armands Strikis (unpublished data, 2024).
[41] D. B. West, *Introduction to graph theory*, 2nd ed. (Prentice Hall, Upper Saddle River, NJ, 2001) includes bibliographical references (p. 533 - 568) and indexes.
[42] M. A. Tremblay, N. Delfosse, and M. E. Beverland, Constant-overhead quantum error correction with thin planar connectivity, Physical Review Letters **129**, 050504 (2022).
[43] E. Arjomandi, An efficient algorithm for colouring the edges of a graph with $\delta + 1$ colours, INFOR: Information Systems and Operational Research **20**, 82 (1982).