Lattice Surgery and Magic State Injection in Surface Codes

Abstract:

Contents

Contents

1 Surface code

The planar surface code is a topological stabilizer code designed for fault-tolerant quantum computation in architectures limited to two-dimensional, nearest-neighbor interactions [1]. Its structure is not merely a mathematical convenience but a direct implementation of topological order, first conceptualized by Kitaev, where quantum information is encoded non-locally in the collective state of many physical qubits [2]. This non-local encoding provides inherent protection against local errors, forming the basis of its high error threshold.

1.1 Stabilizer formalism on a 2D lattice

The code is defined on a two-dimensional square lattice, $\mathcal{L}$, which is a graph composed of a set of vertices V, edges E, and faces (or plaquettes) F. In the standard formulation, physical data qubits are associated with the edges of this lattice, $e \in E$. The total Hilbert space of the system is therefore $\mathcal{H} = (\mathbb{C}^2)^{\otimes |E|}$. The code space is defined as the ground state subspace of a local Hamiltonian, which is constructed from two types of commuting stabilizer generators.

For any vertex $v \in V$ in the bulk of the lattice, we define a **vertex operator** (or star operator), A_v, as the tensor product of Pauli-X operators on all edges incident to it:

$$A_v = \bigotimes_{e \in \mathrm{star}(v)} X_e, \tag{1.1}$$

where $\mathrm{star}(v)$ denotes the set of edges connected to vertex v. For any face $p \in F$ in the bulk, we define a **plaquette operator**, B_p, as the tensor product of Pauli-Z operators on the edges forming its boundary:

$$B_p = \bigotimes_{e \in \partial p} Z_e, \tag{1.2}$$

where ∂p is the set of edges bounding face p. A fundamental property of these generators is their mutual commutativity. For any pair of generators A_v and B_p, their commutator is zero:

$$[A_v, B_p] = 0 \quad \forall v \in V, p \in F. \tag{1.3}$$

This holds because the support of a vertex operator and a plaquette operator, $\mathrm{supp}(A_v)$ and $\mathrm{supp}(B_p)$, intersect on either zero or two edges. As Pauli operators on different qubits commute and $\{X, Z\} = XZ - ZX = -2iY$, the commutator $[X_e, Z_e]$ anti-commutes. The overall sign of the commutator is determined by $(-1)^k$, where k is the number of qubits on which they both act non-trivially. Since k is always even (0 or 2), the operators commute.

The stabilizer group $\mathcal{S}$ is the Abelian group generated by all such vertex and plaquette operators, $\mathcal{S} = \langle \{A_v\}_{v \in V}, \{B_p\}_{p \in F} \rangle$. The code space $\mathcal{C}$ is the subspace of states stabilized by this group, i.e., the simultaneous $+1$ eigenspace of all generators:

$$\mathcal{C} = \{ |\psi\rangle \in \mathcal{H} \mid S|\psi\rangle = |\psi\rangle, \forall S \in \mathcal{S} \}. \tag{1.4}$$

This algebraic definition is physically grounded in the local Hamiltonian:

$$H = -\sum_{v \in V} A_v - \sum_{p \in F} B_p. \tag{1.5}$$

Since all terms in H commute, it is frustration-free. The ground state is the state that minimizes the energy, which occurs when each term contributes its lowest eigenvalue (-1). This corresponds to the state being a $+1$ eigenstate of every A_v and B_p, which is precisely the definition of the code space $\mathcal{C}$.

Errors manifest as excitations above this ground state. A physical Pauli-Z error on an edge e, for instance, anti-commutes with the two vertex operators A_{v_1} and A_{v_2} that share edge e. This flips their eigenvalues from $+1$ to -1, creating a pair of localized, high-energy excitations. In the language of topological field theory, these excitations are anyons. The set of violated stabilizers, known as the error syndrome, marks the locations of these anyons. The goal of decoding is to identify the error chain that created the observed anyon pattern and apply a correction that annihilates them, returning the system to the ground state code space.

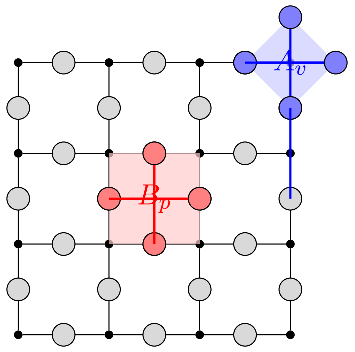

Figure 1. Bulk stabilizers of the standard surface code. Data qubits (gray circles) reside on the edges of a square lattice. A plaquette operator B_p (red) is the product of Z operators on the four qubits bounding a face. A vertex operator A_v (blue) is the product of X operators on the four qubits incident to a vertex.

1.2 Encoding a qubit: boundaries and logical operators

The toric code, defined on a torus, has a four-fold degenerate ground state, encoding two logical qubits ($k = 2$) corresponding to the two independent non-contractible loops (homology cycles) of the torus [2]. To create the planar surface code with a single logical qubit ($k = 1$), the topology is altered by introducing boundaries [3]. These boundaries are not mere terminations of the lattice; they are distinct topological features that define the logical qubit.

Boundaries are created by truncating the set of stabilizer generators at the periphery. This results in stabilizers of lower weight (e.g., weight-2 or 3) than the bulk weight-4 operators. Two primary types of boundaries are defined:

- **Smooth (Z-type) Boundary:** Characterized by terminating the lattice with complete plaquette (Z) stabilizers. The vertex (X) operators along this edge are truncated.

- **Rough (X-type) Boundary:** Characterized by terminating the lattice with complete vertex (X) stabilizers. The plaquette (Z) operators along this edge are truncated.

A single logical qubit is encoded on a rectangular patch with two opposing smooth boundaries and two opposing rough boundaries. The logical operators are non-trivial operators that commute with all elements of the stabilizer group $\mathcal{S}$ but are not themselves in $\mathcal{S}$. They are elements of the normalizer $\mathcal{N}(\mathcal{S})$ that are not in the stabilizer group itself, forming the coset $\mathcal{N}(\mathcal{S})/\mathcal{S}$.

For a patch with rough boundaries on the left/right and smooth boundaries on the top/bottom:

- The logical Z operator, Z_L, is a string of Pauli-Z operators on a path of edges connecting the two rough boundaries: $Z_L = \bigotimes_{e\in\mathcal{P}_Z} Z_e$.
- The logical X operator, X_L, is a string of Pauli-X operators on a path of edges connecting the two smooth boundaries: $X_L = \bigotimes_{e\in\mathcal{P}_X} X_e$.

These operators have three essential properties. First, they commute with all stabilizers, $[S, X_L] = 0$ and $[S, Z_L] = 0$ for all $S \in \mathcal{S}$, because any such logical string crosses the support of any stabilizer an even number of times. Second, they anti-commute with each other, $\{X_L, Z_L\} = 0$, because their respective paths must cross at an odd number of qubits. Third, they are defined by their topology. Any two paths $\mathcal{P}_Z$ and $\mathcal{P}'_Z$ connecting the same boundaries define logical operators Z_L and Z'_L that are equivalent up to a stabilizer. The operator $Z_L(Z'_L)^\dagger$ is a product of Zs on a closed, contractible loop, which is equivalent to a product of the plaquette operators it encloses. Thus, Z_L and Z'_L represent the same logical operation. This topological equivalence is the source of the code's protection: a local error cannot deform a trivial operator (a stabilizer) into a non-trivial logical operator.

1.3 Example: the rotated surface code

A widely used variant is the rotated surface code, which offers superior qubit efficiency and is often preferred in experimental implementations [1]. In this formulation, the lattice is conceptually rotated by 45 degrees, and the data qubits are placed on the *vertices* of the square lattice, rather than the edges.

The stabilizer generators are now exclusively weight-4 plaquette operators, arranged in a checkerboard pattern:

- For one set of faces (e.g., "white" squares), X-type stabilizers are defined: $S_p^X = \bigotimes_{v\in\partial p} X_v$.
- For the other set of faces (e.g., "black" squares), Z-type stabilizers are defined: $S_p^Z = \bigotimes_{v\in\partial p} Z_v$.

This structure has a manifest self-duality that is only implicit in the standard code. In the standard formulation, X errors are detected by plaquette stabilizers while Z errors are detected by vertex stabilizers; the syndrome graphs for these two error types live on

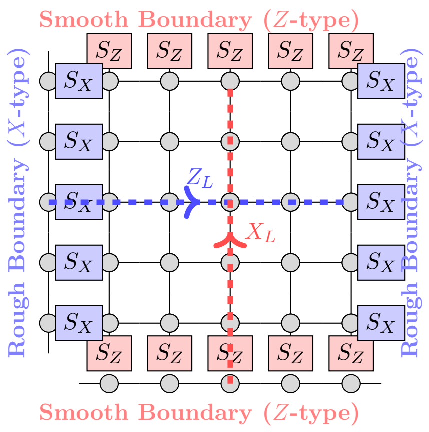
Smooth Boundary (Z-type)
Rough Boundary (X-type)
Rough Boundary (X-type)
S_Z
S_X
Z_L
X_L
Smooth Boundary (Z-type)

Figure 2. A distance-5 planar surface code patch. The logical qubit is defined by the boundary conditions. The logical Z_L operator is a string of physical Z operators connecting the two rough (X-type) boundaries. The logical X_L operator is a string of physical X operators connecting the two smooth (Z-type) boundaries. Stabilizers on the boundaries have lower weight than those in the bulk.

different geometric structures (the primal and dual lattices). In the rotated code, both X and Z errors are detected by plaquette operators. This means the syndrome graph has the same topology for both error types, simplifying the classical decoding logic.

Furthermore, the rotated code is more resource-efficient. A distance-d rotated code requires d^2 data qubits to encode one logical qubit, whereas the standard formulation requires $d^2 + (d-1)^2$ qubits. This near-halving of the physical qubit overhead for the same code distance is a significant advantage for resource-constrained quantum hardware. The logical operators are defined as strings of single-qubit Pauli operators along diagonal paths connecting the boundaries, as shown in Fig. 3.

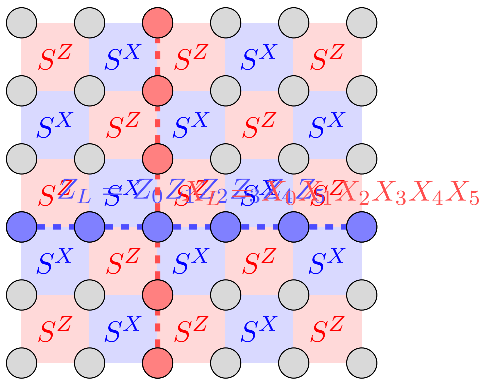
S^Z
S^X
$Z_L = Z_0 Z_1 Z_2 Z_3 Z_4 Z_5$
$X_L = X_0 X_1 X_2 X_3 X_4 X_5$

Figure 3. A distance-5 rotated surface code. Data qubits are placed on vertices. Stabilizers are weight-4 plaquette operators arranged in a checkerboard pattern (S^X in blue, S^Z in red). Logical operators are strings of single-qubit Pauli operators connecting opposite boundaries.

2 Lattice surgery

Lattice surgery provides a powerful, scalable way to perform fault-tolerant quantum computation within architectures restricted to two-dimensional nearest-neighbor (2DNN) interactions [4]. It serves as the primary alternative to transversal gates—whose fault-tolerant implementation is severely restricted by the Eastin-Knill theorem—and the braiding of topological defects, offering a more resource-efficient pathway for logical operations [1, 5].

In lattice surgery, logical operations are realized through the dynamic reconfiguration of the surface code's stabilizer group rather than the conventional circuit model of unitary gate evolution. This is achieved by a sequence of projective Pauli measurements on physical qubits, which effectively "merge" separate logical qubit patches into a single composite patch, or "split" a single patch into multiple distinct ones. All Clifford gates implemented via lattice surgery are constructed from compositions of these elementary projective measurements, framing the computation within a measurement-based model [6].

2.1 The merge operation: a detailed stabilizer analysis

The merge operation entangles two logical qubits encoded in separate surface code patches. The type of logical Pauli product operator measured ($Z_L \otimes Z_L$ or $X_L \otimes X_L$) is determined by the nature of the boundaries brought together.

2.1.1 Merging along an X-boundary (rough merge) to measure $Z_L \otimes Z_L$

Consider two adjacent, independent distance-d surface code patches, denoted A and B, encoding logical qubits $|\psi\rangle_A$ and $|\psi\rangle_B$ (See Fig. 4). Their respective stabilizer groups are $S_A = \langle\{S_{A,i}\}\rangle$ and $S_B = \langle\{S_{B,j}\}\rangle$. The logical operators are defined by strings of physical Pauli operators connecting boundaries of a specific type. For each patch, the logical Z_L is a string of Z operators connecting the two X-boundaries (rough boundaries), and the logical X_L is a string of X operators connecting the two Z-boundaries (smooth boundaries).

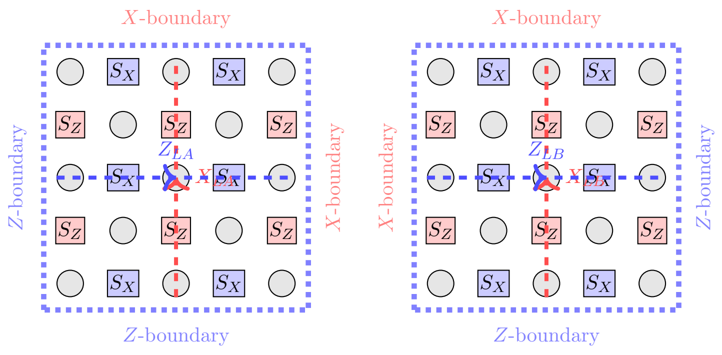

Figure 4. Initial configuration for a rough merge. Two independent distance-3 surface code patches, A and B, are positioned to merge along their X-boundaries (rough boundaries). Logical operators X_L and Z_L are shown as dashed lines connecting boundaries of the opposite type.

The merge procedure is as follows:

1. **Turn off boundary stabilizers:** The weight-2 X-stabilizers along the adjoining rough boundaries of patch A (right side) and patch B (left side) are deactivated. This means their syndromes are no longer measured.

2. **Measure merge operators:** A new set of weight-4 Z-stabilizers, denoted $\{S_{M,k}\}$, are measured across the boundary, coupling data qubits from both patches. These new stabilizers must commute with all existing bulk stabilizers of both patches. Since the bulk stabilizers are X-type plaquettes, the merge operators must be Z-type.

The result of this procedure is a single, larger surface code patch, AB, with a new stabilizer group S_{AB}. The original groups S_A and S_B are subsets of the new group, but S_{AB} contains an additional independent generator. This new generator is the product of the original logical Z operators, $Z_{LA}Z_{LB}$. To see this, consider the product of all new merge stabilizers $\prod_k S_{M,k}$. The physical Z operators on the interior data qubits of the merge region cancel in pairs, leaving only the product of physical Z operators along the path of Z_{LA} and Z_{LB}. Thus, $\prod_k S_{M,k} = Z_{LA}Z_{LB}$.

Measuring these new stabilizers projects the combined state $|\Psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B$ into an eigenstate of $Z_{LA}Z_{LB}$. The measurement outcome $m_Z \in \{+1, -1\}$ is the eigenvalue of the new stabilizer $Z_{LA}Z_{LB}$. The post-merge state is:

$$|\Psi'\rangle_{AB} = \frac{1}{\sqrt{2}}(I + m_Z Z_{LA}Z_{LB})|\Psi\rangle_{AB} \tag{2.1}$$

The logical operators of the new, larger patch are transformed. For example, $X'_L = X_{LA}$ and $Z'_L = Z_{LA}Z_{LB}$ can form a valid set, though other choices are possible. The crucial outcome is that a logical $Z_L \otimes Z_L$ measurement has been performed. The final configuration is shown in Fig. 5.

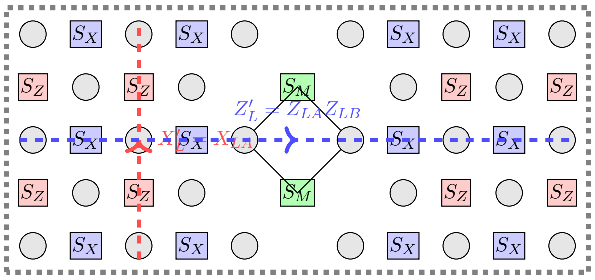

Figure 5. Final configuration after a rough merge. The two patches are now a single logical entity. New Z-type merge stabilizers (S_M) have been measured across the boundary. The product of the original logical operators, $Z_{LA}Z_{LB}$, is now a stabilizer of the merged code, effectively measuring its eigenvalue m_Z.

2.1.2 Merging along a Z-boundary (smooth merge) to measure $X_L \otimes X_L$

The dual process occurs when merging along Z-boundaries (smooth boundaries). The procedure is analogous:

1. **Turn off boundary stabilizers:** The weight-2 Z-stabilizers along the adjoining smooth boundaries are deactivated.

2. **Measure merge operators:** New weight-4 X-stabilizers, $\{S'_{M,k}\}$, are measured across the boundary.

In this case, the product of the new merge operators yields the product of the original logical X operators: $\prod_k S'_{M,k} = X_{LA}X_{LB}$. The measurement projects the combined state into an eigenstate of $X_{LA}X_{LB}$ with eigenvalue $m_X \in \{+1, -1\}$.

2.2 The split operation: a quantitative stabilizer analysis

The split operation is the inverse of merging. It divides a single logical patch into two, and in doing so, performs a projective measurement of a joint logical Pauli operator on the resulting two-qubit state. The following provides a rigorous derivation for splitting via physical X-basis measurements, which enacts a logical $Z_L \otimes Z_L$ measurement [4].

2.2.1 Initial State: The Elongated Patch and its Operator Algebra

We begin with a single, elongated surface code patch, denoted P_{AB}, encoding one logical qubit $|\psi\rangle_L$. This patch is defined on a set of data qubits $Q = Q_A \cup Q_S \cup Q_B$, where $Q_S = \{q_1, \ldots, q_d\}$ is the set of d data qubits along the vertical line where the split will occur. The initial stabilizer group is $\mathcal{S}_{AB} = \langle \{S_i\} \rangle$. The logical operators for this patch are Z'_L, a string of Pauli-Z operators on a path of qubits connecting the two rough (X-type) boundaries, and X'_L, a string of Pauli-X operators connecting the two smooth (Z-type) boundaries. The initial configuration is depicted in Fig. 6(a).

2.2.2 The Measurement Protocol and Stabilizer Group Transformation

The split is initiated by performing a projective measurement of the Pauli-X operator on each data qubit $q_i \in Q_S$.

1. **Measure data qubits:** For each $q_i \in Q_S$, the observable X_i is measured, yielding a classical outcome $m_i \in \{+1, -1\}$. The product of these outcomes is the classical result of the logical measurement:

$$m_Z = \prod_{i=1}^{d} m_i. \tag{2.2}$$

   The collective measurement operator is $M_S = \bigotimes_{i=1}^{d} X_i$.

2. **Reconfigure stabilizers:** The stabilizer group is updated according to the measurement. Any stabilizer $S \in \mathcal{S}_{AB}$ that anti-commutes with any of the measured operators $\{X_i\}$ is removed from the generator set. These are precisely the Z-type plaquette stabilizers whose support includes qubits in Q_S. The remaining stabilizers are partitioned into those with support entirely on patch A ($\mathcal{S}'_A$) and those on patch B ($\mathcal{S}'_B$). New, weight-2 boundary stabilizers are then measured along the newly created rough boundaries, completing the new, independent stabilizer groups $\mathcal{S}_A$ and $\mathcal{S}_B$.

The result is two disjoint surface code patches, A and B, encoding a two-qubit state.

2.2.3 Transformation of Logical Operators and the Projective Measurement Identity

The logical operators of the initial patch are transformed into a new set of logical operators for the two resulting patches.

- The original vertical logical operator X'_L is severed by the measurement. Its segments on the left and right of the split line become the new logical operators for patches A and B, respectively: X_{LA} and X_{LB}.
- The original horizontal logical operator Z'_L can be chosen without loss of generality to have a path that lies entirely on patch A. This operator commutes with all stabilizers in the new group $\mathcal{S}_A$ and anti-commutes with X_{LA}. It therefore becomes the logical Z for patch A: $Z_{LA} = Z'_L$.
- A new logical operator Z_{LB} for patch B is defined by a string of Pauli-Z operators connecting the rough boundaries of patch B.

The central result of the split operation is that the product of the classical measurement outcomes, m_Z, is equal to the eigenvalue of the joint logical operator $Z_{LA} \otimes Z_{LB}$ on the final two-qubit state. This can be demonstrated through the following operator identity:

$$Z_{LA} \otimes Z_{LB} = \left(\bigotimes_{i=1}^{d} X_i \right) \cdot \left(\prod_k S_{X,k} \right) = M_S \cdot \prod_k S_{X,k}, \tag{2.3}$$

where $\{S_{X,k}\}$ is a specific set of X-type stabilizers from the *original* stabilizer group $\mathcal{S}_{AB}$. This identity is most easily understood graphically, as shown in Fig. 6(c). The product of the two logical operators $Z_{LA} \otimes Z_{LB}$ can be viewed as a single operator whose support consists of two disjoint strings. By multiplying this operator by a suitable chain of the original X-stabilizers, it can be deformed until its support is precisely the set of physical X operators on the split line, M_S.

Since the initial state $|\psi\rangle_L$ is, by definition, a $+1$ eigenstate of all stabilizers $S_{X,k} \in \mathcal{S}_{AB}$, its expectation value for their product is also $+1$. Therefore, the expectation value of the joint logical operator on the final state is determined entirely by the expectation value of the measurement operator M_S:

$$\langle Z_{LA} \otimes Z_{LB} \rangle_{\text{final}} = \langle M_S \cdot \prod_k S_{X,k} \rangle_{\text{initial}} = \langle M_S \rangle \cdot \prod_k \langle S_{X,k} \rangle = m_Z \cdot \prod_k (+1) = m_Z. \tag{2.4}$$

This confirms that the physical act of splitting the patch via local X-basis measurements projects the final two-qubit state into the m_Z-eigenspace of the non-local $Z_L \otimes Z_L$ operator. The dual operation, splitting via Z-basis measurement, similarly projects the system into an eigenstate of $X_L \otimes X_L$.

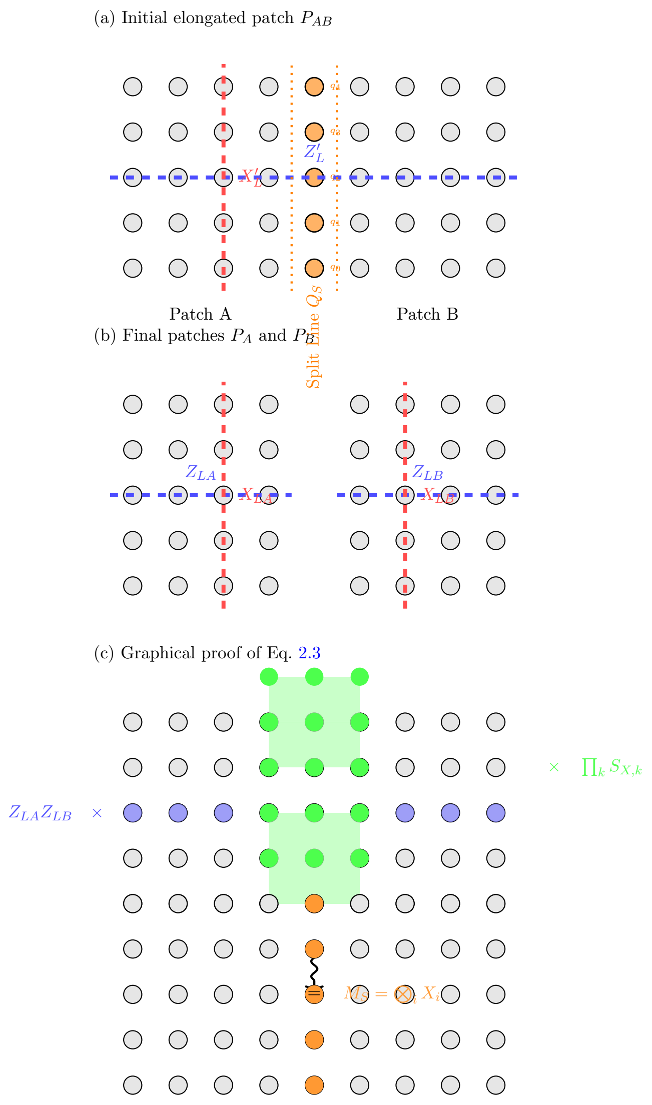

Figure 6. Quantitative analysis of the split operation. (a) An initial elongated patch P_{AB} encodes one logical qubit. The logical operators Z'_L and X'_L span the patch. A vertical column of data qubits Q_S is designated for measurement. (b) After measuring qubits in Q_S in the X-basis, the patch is divided into two independent patches, P_A and P_B, with new sets of logical operators. (c) A graphical representation of the identity in Eq. 2.3. The product of the logical operators $Z_{LA}Z_{LB}$ (blue) and a chain of original X-stabilizers (green) is topologically equivalent to the product of physical X operators on the split line, M_S (orange).

2.3 Application: a fault-tolerant CNOT gate

The elementary merge and split operations are composed to implement universal Clifford gates. The logical CNOT gate provides a canonical example of the technique's utility, realizing a non-local two-qubit gate through a sequence of local measurements and dynamic stabilizer reconfigurations [4, 5].

2.3.1 Protocol, State Space, and Circuit Identity

The CNOT gate from a control logical qubit C to a target logical qubit T is implemented using a third ancilla logical qubit A. The protocol realizes the following sequence of projective measurements:

1. Prepare the ancilla in the logical $|+\rangle_L$ state.
2. Measure the joint Pauli operator $M_1 = Z_{LC} \otimes Z_{LA}$. Let the outcome be $m_1 \in \{+1, -1\}$.
3. Measure the joint Pauli operator $M_2 = X_{LA} \otimes X_{LT}$. Let the outcome be $m_2 \in \{+1, -1\}$.
4. Apply Pauli corrections to C and T conditioned on the outcomes m_1, m_2.

The initial state of the three-qubit system is $|\Psi_0\rangle = |\psi\rangle_C \otimes |+\rangle_A \otimes |\phi\rangle_T$, where $|\psi\rangle_C = \alpha|0\rangle_L + \beta|1\rangle_L$ is an arbitrary control state, $|\phi\rangle_T$ is an arbitrary target state, and $|+\rangle_A = \frac{1}{\sqrt{2}}(|0\rangle_L + |1\rangle_L)_A$. The target unitary operation is $U_{\text{CNOT}} = |0\rangle_L\langle 0|_C \otimes I_T + |1\rangle_L\langle 1|_C \otimes X_{LT}$.

2.3.2 Spacetime Evolution: Parallel Merges and State Transformation

The two measurements are performed in parallel over a single fault-tolerant time step of duration d (the code distance) via two simultaneous merge operations.

- A **rough merge** between patches C and A projects the state with the operator $P_1(m_1) = \frac{1}{2}(I + m_1 Z_{LC} Z_{LA})$.
- A **smooth merge** between patches A and T projects the state with the operator $P_2(m_2) = \frac{1}{2}(I + m_2 X_{LA} X_{LT})$.

The state of the system after this phase, $|\Psi_1\rangle$, is given by the application of these projectors to the initial state $|\Psi_0\rangle$. Since the projectors commute, their order does not matter.

$$\begin{aligned} |\Psi_1\rangle &= P_2(m_2)P_1(m_1)|\Psi_0\rangle \\ &= \frac{1}{2}(I + m_2 X_{LA} X_{LT})(I + m_1 Z_{LC} Z_{LA}) \left(|\psi\rangle_C \otimes \frac{1}{\sqrt{2}}(|0\rangle_A + |1\rangle_A) \otimes |\phi\rangle_T \right). \end{aligned} \tag{2.5}$$

Expanding this expression and using the relations $Z_A|0\rangle_A = |0\rangle_A$, $Z_A|1\rangle_A = -|1\rangle_A$, $X_A|0\rangle_A = |1\rangle_A$, and $X_A|1\rangle_A = |0\rangle_A$ allows for a full derivation of the resulting entangled state across the three merged patches.

2.3.3 Split Operations and Final State Derivation

Following the merge phase, a time step of duration d is used to perform split operations that restore the three independent patches C, A, and T. This process is equivalent to measuring out the ancilla qubit A. To find the final state of the control and target qubits, we project $|\Psi_1\rangle$ onto the logical basis of A and trace it out. The final state of the C-T system, $|\Psi_f\rangle_{CT}$, is found to be proportional to:

$$|\Psi_f\rangle_{CT} \propto X_{LC}^{m_2'} Z_{LT}^{m_1'} \cdot U_{\text{CNOT}} \cdot (|\psi\rangle_C \otimes |\phi\rangle_T), \tag{2.6}$$

where the classical bits $m'_1, m'_2 \in \{0,1\}$ are functions of the measurement outcomes $m_1, m_2 \in \{+1,-1\}$. For instance, $m'_1 = (1-m_1)/2$.

2.3.4 Pauli Frame Corrections

To recover the desired state $U_{\mathrm{CNOT}}(|\psi\rangle_C \otimes |\phi\rangle_T)$, a correction operator $U_{\mathrm{corr}} = (Z_{LT}^{m'_1})^\dagger (X_{LC}^{m'_2})^\dagger = Z_{LT}^{m'_1} X_{LC}^{m'_2}$ must be applied. In a fault-tolerant architecture, these logical Pauli operations are not implemented with additional physical gates. Instead, they are tracked in a classical data structure known as the Pauli frame. The effect of these operators is propagated classically through subsequent Clifford gates, and they are only physically realized when a non-Clifford operation is required. A summary of the protocol is provided in Table 1.

Table 1. Summary of the fault-tolerant CNOT protocol via lattice surgery.

Step	Time	Operation	Patches	Logical Operator Measured	Outcome
1	$t_1 \to t_1 + d$	Rough Merge	C, A	$Z_{LC} \otimes Z_{LA}$	$m_1 \in \{+1,-1\}$
2	$t_1 \to t_1 + d$	Smooth Merge	A, T	$X_{LA} \otimes X_{LT}$	$m_2 \in \{+1,-1\}$
3	$t_2 \to t_2 + d$	Split	(CA), (AT)	(Discards Ancilla)	-
4	Post-proc.	Pauli Correction	C, T	(Classical processing)	Apply $X_{LC}^{m'_2}, Z_{LT}^{m'_1}$

2.3.5 The CNOT as a Topological Spacetime Event

The entire CNOT procedure is best understood not as a sequence of discrete 2D operations, but as a single, continuous (2+1)-dimensional spacetime manifold, as depicted in Fig. 7. The logical qubits trace out 2D world-sheets that evolve along the time axis. The merge operations correspond to the fusion of these world-sheets, creating a topologically non-trivial interaction region. Within this merged volume, the product of logical operators that were formerly independent strings becomes a contractible loop, which is equivalent to a product of local stabilizers. For example, the product of the string-like operators Z_{LC} and Z_{LA} forms a closed loop on the merged C-A world-sheet, demonstrating geometrically why its joint eigenvalue is measured. The subsequent split operation is the fission of this manifold back into independent world-sheets. The inherent fault tolerance of the gate arises from the robustness of this spacetime topology: local physical errors create small "punctures" or "bumps" on the world-sheets but do not alter the global topology of their interaction, thus preserving the logical outcome of the gate.

2.4 Relation to magic state injection

The fault-tolerant logical CNOT gate constructed via lattice surgery is the critical entangling resource required for a fully robust magic state injection protocol. The procedure outlined in Section 3 describes a CNOT between a logical data qubit and a single *physical* ancilla qubit prepared in the magic state $|T\rangle$. While this protocol is fault-tolerant, its primary vulnerability lies in errors affecting the unprotected physical ancilla during the entangling operation.

Lattice surgery enables a superior implementation. The protocol can be upgraded by first encoding the physical magic state $|T\rangle_A$ into its own small, logical surface code patch,

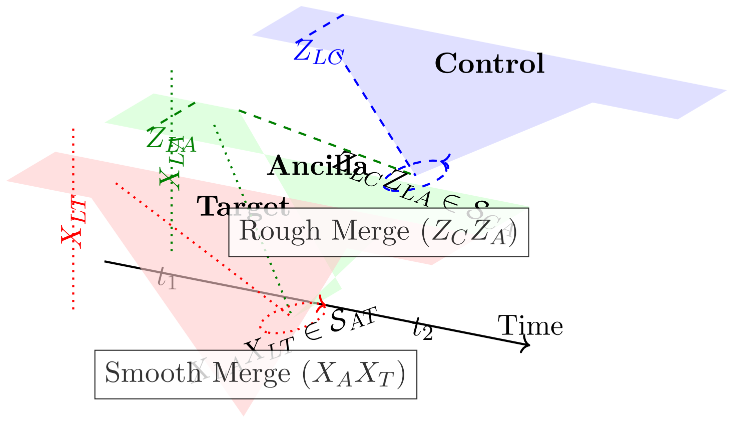

Figure 7. Enhanced spacetime diagram of a logical CNOT gate. The world-sheets of the Control, Ancilla, and Target qubits fuse between times t_1 and t_2. The dashed (blue/green) lines trace the paths of the Z_L operators, which form a contractible loop on the merged C-A manifold, making their product $Z_{LC}Z_{LA}$ a stabilizer. Similarly, the dotted (green/red) lines trace the X_L operators, whose product $X_{LA}X_{LT}$ becomes a stabilizer on the A-T manifold. This geometric representation illustrates how the gate's logic is encoded in the spacetime topology.

creating a logical magic state $|T\rangle_L$. The entangling operation is then a logical-logical CNOT, $\text{CNOT}_{L,L}$, precisely of the form detailed in Section 2.3. This procedure involves merging the logical data patch with the logical magic state patch. By protecting the magic state within its own code patch throughout the interaction, this method significantly reduces the probability of uncorrectable errors during the entangling step.

Therefore, lattice surgery is not merely an alternative method for implementing Clifford gates; it is the foundational technology that enables the robust and scalable implementation of non-Clifford gates within a 2D architecture. It provides the necessary tools to interface logical data qubits with logical resource states (like $|T\rangle_L$), which is the cornerstone of universal, fault-tolerant quantum computation with surface codes.

3 Magic state injection for universal computation

3.1 The necessity of non-Clifford gates in a Clifford-dominated architecture

The surface code provides a remarkably robust framework for fault-tolerant quantum computation, primarily due to its high error threshold and its natural compatibility with a 2D planar layout of qubits with only nearest-neighbor interactions [1]. Within this architecture, the Clifford group, can be implemented fault-tolerantly with relative efficiency through techniques like lattice surgery [5]. The Clifford group, generated by gates such as Hadamard (H), Phase (S), and CNOT, is powerful yet insufficient for universal quantum computation. The *Gottesman-Knill theorem* establishes that any quantum circuit composed exclusively of Clifford gates, stabilizer state preparations, and Pauli measurements can be efficiently simulated on a classical computer [7]. To unlock the full potential of quantum computation and perform tasks intractable for classical machines, it is essential to supplement the Clifford gate set with at least one non-Clifford gate.

The canonical choice for this role is the T-gate, defined by the matrix $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ [1, 7, 8]. The addition of the T-gate to the Clifford set renders the gate set universal. However, the T-gate presents a significant challenge in the surface code architecture. The fault tolerance of logical Clifford gates often arises from their transversal implementation, where the logical gate is realized by applying the corresponding physical gate to each data qubit in the code block. This approach fails for the T-gate since applying a physical T-gate transversally to the data qubits of a surface code patch does not produce a valid logical T-gate. Instead, it spreads errors in a way that the code's stabilizers cannot detect or correct, thereby destroying the encoded information [1]. This fundamental limitation necessitates an indirect, yet fault-tolerant, method to achieve the effect of a T-gate, a problem elegantly solved by the technique of magic state injection.

3.2 The magic state: a consumable quantum resource

The solution to implementing non-Clifford gates is to treat them not as direct operations, but as the result of consuming a special quantum resource. This resource is an ancillary qubit prepared in a specific non-stabilizer state known as a "magic state."

Definition 3.1:

For the T-gate, the corresponding ***magic state*** is typically denoted as $|T\rangle$ or $|A\rangle$ and is defined as:

$$|T\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle). \tag{3.1}$$

Crucially, it is not an eigenstate of any Pauli operator and thus lies outside the set of stabilizer states that are efficiently simulable classically. It is precisely this non-stabilizer character that provides the "magic" required to perform a non-Clifford operation [1].

3.3 The injection protocol

The standard protocol for using a magic state to apply a logical T-gate is a form of gate teleportation. It involves a logical data qubit, which we wish to act upon, and a physical ancilla qubit prepared in the magic state. The procedure unfolds in three stages:

$\boxed{\text{Step 1}}$ **Initialization**

The system begins with a logical data qubit in an arbitrary state $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$ and a single physical ancilla qubit prepared in the magic state $|T\rangle_A = \frac{1}{\sqrt{2}}(|0\rangle_A + e^{i\pi/4}|1\rangle_A)$. The combined initial state of the system is the tensor product

$$|\Psi_0\rangle = |\psi\rangle_L \otimes |T\rangle_A = \frac{1}{\sqrt{2}}(\alpha|0\rangle_L + \beta|1\rangle_L) \otimes (|0\rangle_A + e^{i\pi/4}|1\rangle_A) \tag{3.2}$$

$\boxed{\text{Step 2}}$ **Entangling operation**

A fault-tolerant logical-physical CNOT gate, denoted,

$$\text{CNOT}_{L,A} : |\alpha\rangle_L \otimes |\beta\rangle_A \;\mapsto\; |\alpha\rangle_L \otimes X_A^{\alpha}|\beta\rangle_A \tag{3.3}$$

is applied. The logical data qubit serves as the control, and the physical ancilla qubit is the target. The state of the system evolves to

$$\begin{aligned}|\Psi_1\rangle &= \mathrm{CNOT}_{L,A}|\Psi_0\rangle \\ &= \frac{1}{\sqrt{2}}\Big(\alpha|0\rangle_L \otimes \big(|0\rangle_A + e^{i\pi/4}|1\rangle_A\big) + \beta|1\rangle_L \otimes X_A\big(|0\rangle_A + e^{i\pi/4}|1\rangle_A\big)\Big) \\ &= \frac{1}{\sqrt{2}}\Big(\alpha|0\rangle_L\big(|0\rangle_A + e^{i\pi/4}|1\rangle_A\big) + \beta|1\rangle_L\big(|1\rangle_A + e^{i\pi/4}|0\rangle_A\big)\Big) \\ &= \frac{1}{\sqrt{2}}\Big(\big(\alpha|0\rangle_L + e^{i\pi/4}\beta|1\rangle_L\big) \otimes |0\rangle_A + \big(e^{i\pi/4}\alpha|0\rangle_L + \beta|1\rangle_L\big) \otimes |1\rangle_A\Big)\end{aligned} \tag{3.4}$$

$\boxed{\text{Step 3}}$ **Ancilla measurement and conditional correction**

The ancilla qubit is measured in Z basis $|0\rangle_A$, $|1\rangle_A$. Let the measurement outcome be $m \in \{0, 1\}$, corresponding to projections onto $|0\rangle_A$ and $|1\rangle_A$ respectively.

- If the outcome is $m = 0$ (projection onto ${}_A\langle 0|$):

$$\begin{aligned}|\Psi_{m=0}\rangle &= {}_A\langle 0|\Psi_1\rangle \\ &= \frac{1}{\sqrt{2}}\big(\alpha|0\rangle_L + e^{i\pi/4}\beta|1\rangle_L\big) \\ &\propto T_L\,|\psi\rangle_L\,.\end{aligned} \tag{3.5}$$

- If the outcome is $m = 1$ (projection onto ${}_A\langle 1|$):

$$\begin{aligned}|\Psi_{m=1}\rangle &= {}_A\langle 1|\Psi_1\rangle \\ &= \frac{1}{\sqrt{2}}\big(e^{i\pi/4}\alpha|0\rangle_L + \beta|1\rangle_L\big) \\ &\propto S_L^\dagger T_L\,|\psi\rangle_L\,.\end{aligned} \tag{3.6}$$

To obtain the desired state $T_L\,|\psi\rangle_L$, it suffices to apply the logical phase gate S_L. Since S_L is a Clifford gate, this correction can be implemented fault-tolerantly.

The entire procedure is depicted in Fig. 8. It shows the standard quantum circuit diagram for magic state injection. The top wire represents the logical data qubit, while the bottom wire represents the physical ancilla qubit. The protocol begins by preparing the ancilla in the $|T\rangle$ state. A controlled-NOT operation entangles the two qubits. The classical measurement outcome on the ancilla, m, then controls the application of a corrective logical gate, S_L^m, to the data qubit.

Fig. 9 provides a conceptual illustration of how this protocol might be physically realized within a surface code architecture. There, a distance-3 surface code patch, comprising data qubits and ancilla qubits for stabilizer measurements, encodes the logical qubit $|\psi\rangle_L$. A single, physically separate ancilla qubit is prepared in the $|T\rangle$ state. The logical CNOT operation is not a single physical gate but a complex sequence of operations. This could be implemented, for example, by deforming the code patch to move a logical operator string across the physical ancilla (a process known as braiding) or by using lattice surgery to temporarily merge the ancilla into the code structure to perform a joint measurement [1, 5].

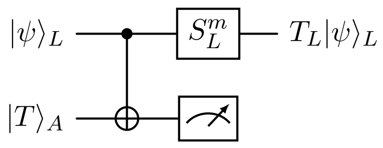

Figure 8. Quantum circuit for T-gate magic state injection. A logical data qubit $|\psi\rangle_L$ is entangled with a physical ancilla prepared in the magic state $|T\rangle_A$. The ancilla is then measured in the Hadamard basis, and the classical outcome m determines the corrective logical Clifford gate applied to the data qubit.

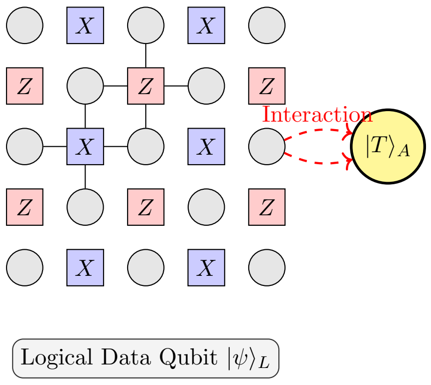

Figure 9. Conceptual layout of magic state injection on a distance-3 surface code patch. The logical qubit is encoded in the grid of data qubits (gray circles). Stabilizer measurements are performed using ancillas (X-type in blue, Z-type in red). A separate physical ancilla is prepared in the $|T\rangle$ state (yellow). The fault-tolerant CNOT involves a complex interaction between the logical qubit and this physical ancilla.

3.4 Fault tolerance and error propagation analysis

The elegance of the magic state injection protocol lies in its inherent fault tolerance. The most vulnerable component is the single physical ancilla qubit $|T\rangle_A$, which is not protected by the surface code during its preparation and initial interaction. The following analysis provides a precise, step-by-step derivation of how errors on this fragile ancilla are transformed into manageable, correctable errors on the robust logical qubit. We consider the effect of a single physical Pauli error $E_A \in \{X_A, Y_A, Z_A\}$ occurring on the ancilla qubit *before* the entangling $\text{CNOT}_{L,A}$ gate.

- Z_A **Error:** If a Z error occurs, the ancilla state becomes $Z_A|T\rangle_A = \frac{1}{\sqrt{2}}(|0\rangle_A - e^{i\pi/4}|1\rangle_A)$. This error commutes through the target of the $\text{CNOT}_{L,A}$ gate. The commutation relation for a CNOT gate where the control is logical qubit L and the target is ancilla A is $\text{CNOT}_{L,A}(I_L \otimes Z_A) = (Z_L \otimes Z_A)\text{CNOT}_{L,A}$. Applying this to the initial state $|\psi\rangle_L \otimes |T\rangle_A$:

$$\text{CNOT}_{L,A}(|\psi\rangle_L \otimes Z_A|T\rangle_A) = (Z_L \otimes Z_A)\text{CNOT}_{L,A}(|\psi\rangle_L \otimes |T\rangle_A). \tag{3.7}$$

  The Z_A error on the ancilla after the CNOT does not affect the measurement outcome m in the Z-basis. However, the error has propagated to the logical qubit as a Z_L

error. This logical error is detectable and correctable by subsequent rounds of surface code stabilizer measurements.

- X_A **Error:** The original analysis of this error was imprecise. A rigorous derivation follows. An X_A error transforms the ancilla state to $X_A|T\rangle_A = \frac{1}{\sqrt{2}}(|1\rangle_A + e^{i\pi/4}|0\rangle_A)$. The initial state of the system with this error is $|\Psi_{err,0}\rangle = |\psi\rangle_L \otimes X_A|T\rangle_A$. Let $|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L$. The state after the CNOT$_{L,A}$ gate is:

$$\begin{aligned} |\Psi_{err,1}\rangle &= \mathrm{CNOT}_{L,A}|\Psi_{err,0}\rangle \\ &= \frac{1}{\sqrt{2}}\Big(\alpha|0\rangle_L \otimes \big(|1\rangle_A + e^{i\pi/4}|0\rangle_A\big) + \beta|1\rangle_L \otimes X_A\big(|1\rangle_A + e^{i\pi/4}|0\rangle_A\big)\Big) \\ &= \frac{1}{\sqrt{2}}\Big(\alpha|0\rangle_L\big(|1\rangle_A + e^{i\pi/4}|0\rangle_A\big) + \beta|1\rangle_L\big(|0\rangle_A + e^{i\pi/4}|1\rangle_A\big)\Big). \end{aligned} \tag{3.8}$$

  To analyze the measurement outcome, we regroup the terms according to the ancilla's computational basis states:

$$|\Psi_{err,1}\rangle = \frac{1}{\sqrt{2}}\Big(\underbrace{\big(e^{i\pi/4}\alpha|0\rangle_L + \beta|1\rangle_L\big)}_{\propto S_L^\dagger T_L|\psi\rangle_L} \otimes|0\rangle_A + \underbrace{\big(\alpha|0\rangle_L + e^{i\pi/4}\beta|1\rangle_L\big)}_{\propto T_L|\psi\rangle_L} \otimes|1\rangle_A\Big). \tag{3.9}$$

  Comparing this state to the error-free state in Eq. (3.4), the logical state components associated with $|0\rangle_A$ and $|1\rangle_A$ have been swapped. This directly implies that the measurement outcome m will be flipped relative to the error-free case.

  - If the outcome is $m = 0$, the logical state collapses to $|\Psi_{m=0}\rangle \propto S_L^\dagger T_L|\psi\rangle_L$. The protocol applies the correction $S_L^m = S_L^0 = I_L$. The final state is incorrect.
  - If the outcome is $m = 1$, the logical state collapses to $|\Psi_{m=1}\rangle \propto T_L|\psi\rangle_L$. The protocol applies the correction $S_L^m = S_L^1 = S_L$. The final state is $S_L T_L|\psi\rangle_L$, which is also incorrect.

  In both cases, the application of the wrong correction S_L^m due to the flipped measurement outcome results in a final logical Pauli error on the data qubit.

- Y_A **Error:** A Y_A error is equivalent to $iX_A Z_A$. This combines the effects of both an X_A and a Z_A error. The Z_A component propagates to a Z_L error on the logical qubit, while the X_A component flips the measurement outcome m, leading to an incorrect correction. The combined effect results in a final logical Y_L error (up to Clifford byproducts), which is also correctable.

The conclusion is that a single physical Pauli error on the magic state ancilla results in, at most, a single logical Pauli error on the data qubit. The injection protocol thus successfully transforms a local, high-probability physical error on an unprotected qubit into a non-local, correctable logical error on a protected one. This error transformation is the core of the protocol's fault tolerance.

Table 2. Revised error propagation during T-gate magic state injection. A single physical Pauli error on the ancilla qubit prior to the CNOT gate is converted into a correctable logical Pauli error on the data qubit.

Ancilla Error (E_A)	**Effect on Measurement** (m)	**Final Logical Error**
I_A (No error)	m	I_L
X_A	$m \to 1 - m$	S_L or $S_L^\dagger$
Z_A	m	Z_L
$Y_A = iX_A Z_A$	$m \to 1 - m$	Y_L (up to Clifford byproduct)

3.5 The fidelity imperative: distillation

While the injection protocol is fault-tolerant with respect to errors occurring *during* the procedure, its overall success is fundamentally limited by the initial fidelity of the magic state itself. An error in the preparation of the $|T\rangle$ state is indistinguishable from an error occurring just before the CNOT, and as shown in Table 2, this leads directly to a logical error. Therefore, the error rate of the logical T-gate is, at best, equal to the physical error rate of preparing the magic state, p_{phys}. For any algorithm of meaningful complexity, requiring millions or billions of T-gates, a logical gate error rate on the order of $p_{\text{phys}} \approx 10^{-3}$ is unacceptably high. The fidelity of the magic states must be improved by many orders of magnitude. This challenge is addressed by magic state distillation [7].

3.5.1 Magic state distillation

Magic State Distillation (***MSD***) is a quantum error detection protocol applied to the magic states themselves, rather than to the data qubits [7]. It is a probabilistic procedure that consumes N noisy input magic states to produce K (with $K < N$) output states of significantly higher fidelity. The protocol works by encoding a logical state using the noisy magic states, performing a Clifford operation, and then measuring error syndromes. If a non-trivial syndrome is detected, the entire state is discarded. If the syndrome is trivial, the state is accepted and decoded, yielding a purified magic state. This post-selection is the mechanism for error suppression. The key feature is that for an input error rate p, the output error rate p_{out} scales as $p_{out} \propto p^k$ with $k > 1$. This nonlinear error suppression allows for recursive application of the protocol to achieve arbitrarily low error rates, provided the initial error rate p is below a certain threshold.

The canonical example is the 15-to-1 Bravyi-Kitaev protocol, which we now describe in detail.

The 15-to-1 Bravyi-Kitaev Protocol. This protocol is designed to distill the $|T\rangle$ state and relies on the properties of the [] quantum Reed-Muller code.

- **Theoretical Foundation: The [] Code.** The protocol's functionality is enabled by a specific property of the 15-qubit quantum Reed-Muller code: it possesses a transversal, bitwise T-gate. This means that applying a physical T-gate to each of the 15 physical data qubits implements a valid logical T_L gate on the single encoded qubit. This property is rare among quantum codes and is not shared by the surface code. The code has distance $d = 3$, which implies it can detect any single-qubit or two-qubit error.

- **Protocol Logic.** The protocol can be understood as a four-step logical procedure, depicted conceptually in Fig. 10.

  1. **Encoding:** Prepare the logical $|+\rangle_L$ state of the [] code. This state is an eigenstate of the logical X_L operator.
  2. **Transversal Operation:** Apply a physical T-gate to each of the 15 qubits. This is the step where errors are introduced. Each physical T-gate is implemented using one of the noisy input magic states, so each gate has an independent probability p of failing. The transversal application transforms the logical state from $|+\rangle_L$ to $T_L|+\rangle_L = |T\rangle_L$.
  3. **Syndrome Measurement and Post-selection:** Measure the 14 stabilizer generators of the [] code. If any stabilizer measurement yields a -1 eigenvalue (a non-trivial syndrome), it indicates that a detectable error has occurred. In this case, the entire 15-qubit state is discarded, and the protocol fails. The protocol only succeeds if all stabilizer measurements yield $+1$.
  4. **Decoding:** If the state is accepted (trivial syndrome), a decoding circuit maps the 15-qubit logical state $|T\rangle_L$ back to a single physical qubit. This output qubit is the distilled, higher-fidelity $|T\rangle$ state.

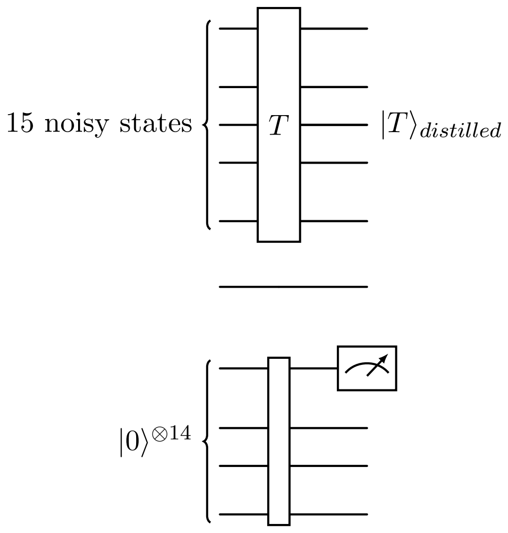

Figure 10. Conceptual circuit for the 15-to-1 magic state distillation protocol. 15 noisy T-gates are used to apply a logical T_L gate to a logical $|+\rangle_L$ state encoded in the [] code. Stabilizer measurements detect errors, and post-selection on the trivial syndrome outcome yields a single, high-fidelity distilled magic state.

- **Quantitative Error Analysis.** The output error rate p_{out} can be derived by analyzing which error patterns evade detection and corrupt the final state.

- **Error Model:** We assume a simplified depolarizing error model where each of the 15 physical T-gates fails with an independent probability p. A gate failure is modeled as the application of an additional Pauli-Z operator. This is a justified simplification because for the input state $|T\rangle$, physical X and Y errors are equivalent to coherent Z rotations, which are discretized into Z errors by the syndrome measurement process.
- **Error Detection:** The code's distance $d = 3$ ensures that all error operators of weight $w < 3$ (i.e., single-qubit and two-qubit Z errors) are detectable. An error E is detected if it anti-commutes with at least one stabilizer $S_i \in \mathcal{S}$. Such errors produce a non-trivial syndrome, leading to the state being discarded.
- **Undetectable Errors:** The protocol fails if an error occurs that is both undetectable and logically non-trivial. An error E is undetectable if $[E, S_i] = 0$ for all $S_i \in \mathcal{S}$. This means E must be in the normalizer of the stabilizer group, $\mathcal{N}(\mathcal{S})$. Such operators are either stabilizers themselves ($E \in \mathcal{S}$) or logical operators ($E \in \mathcal{N}(\mathcal{S}) \setminus \mathcal{S}$). A stabilizer error is logically trivial and does not corrupt the output state. Therefore, failure occurs only if the error E is equivalent to a logical Pauli operator.
- **Leading-Order Failure:** The lowest-weight error that can cause a logical failure is a weight-3 error. The output error rate p_{out} is dominated by the probability of such weight-3 undetectable logical errors.

$$p_{out} = N_{fail} \cdot p^3(1-p)^{12} + \mathcal{O}(p^4) \approx N_{fail}p^3, \tag{3.10}$$

  where N_{fail} is the number of distinct weight-3 Pauli-Z error patterns that are equivalent to a logical operator.
- **Combinatorial Derivation of $N_{fail} = 35$:** The number of undetectable weight-3 logical errors is a specific property of the [] code's structure.
  1. The total number of possible weight-3 error patterns is $\binom{15}{3} = \frac{15 \cdot 14 \cdot 13}{3 \cdot 2 \cdot 1} = 455$.
  2. The code is constructed such that for any pair of single-qubit errors $\{Z_i, Z_j\}$, their combined syndrome is non-zero and unique.
  3. For any such pair, there exists exactly one unique third error Z_k such that the triplet $\{Z_i, Z_j, Z_k\}$ has a trivial syndrome (i.e., it is an element of $\mathcal{N}(\mathcal{S})$).
  4. The number of distinct pairs of error locations is $\binom{15}{2} = 105$. Each pair defines a unique undetectable weight-3 error operator.
  5. However, this method counts each undetectable triplet three times. For an undetectable triplet $\{Z_i, Z_j, Z_k\}$, the pair $\{Z_i, Z_j\}$ points to Z_k, the pair $\{Z_i, Z_k\}$ points to Z_j, and the pair $\{Z_j, Z_k\}$ points to Z_i.
  6. Therefore, the total number of unique, undetectable, weight-3 error operators is $N_{fail} = \frac{\binom{15}{2}}{\binom{3}{2}} = \frac{105}{3} = 35$.

  All 35 of these operators correspond to non-trivial logical errors for the [] code. The output error probability to leading order is thus:

$$p_{out} \approx 35p^3. \tag{3.11}$$

This cubic suppression of errors demonstrates the power of distillation. If the initial error rate p is sufficiently small (below the threshold where $35p^3 < p$), this protocol can be iterated to produce magic states of arbitrarily high fidelity, albeit at an exponential cost in the number of raw noisy states. This overhead is a primary driver of the resource requirements for fault-tolerant quantum computers [1].

3.5.2 Magic state cultivation

Magic State Cultivation is a more recent paradigm that aims to reduce this overhead [9]. Instead of using large multi-qubit circuits, cultivation focuses on purifying a single magic state after it has been injected into a small-distance code (e.g., distance-3). The process involves three stages:

1. **Injection:** A noisy magic state is prepared and encoded into a small surface code patch.
2. **Cultivation:** The stabilizers of the code are measured repeatedly. Additionally, logical operators that ought to have a specific eigenvalue for the magic state (e.g., $S_L H_L$) are also measured fault-tolerantly. If any measurement gives an unexpected outcome, the state is discarded (post-selection). This process verifies the state and projects out errors.
3. **Escape:** Once the state has been verified to a sufficient degree of confidence, the code patch can be grown to a larger distance for use in the main algorithm [9].

Cultivation may offer a more resource-efficient pathway to high-fidelity magic states by avoiding the large ancillary logical qubit requirements of traditional distillation protocols.

3.6 Practical implementation and hardware considerations

The abstract circuit diagram in Fig. 8 belies the true complexity of implementing magic state injection on physical hardware. The logical-physical CNOT gate is not a primitive operation but a subroutine compiled into a sequence of physical gates, the specifics of which are highly dependent on the underlying hardware architecture.

For instance, architectures with limited qubit connectivity, such as the heavy-hexagon lattice used in some superconducting quantum processors, cannot directly perform the four-qubit parity checks required for standard surface code stabilizers. This necessitates modified syndrome extraction circuits, sometimes involving additional “flag” qubits to signal errors within the measurement cycle itself [10]. The layout of the logical CNOT for injection must be co-designed with these hardware-specific constraints. To optimize qubit usage, variants like the rotated surface code are often preferred, which further alters the specific gate sequences required for logical operations [1, 10].

An alternative to gate-based implementations of the logical CNOT is **lattice surgery**. In this paradigm, the magic state ancilla would first be encoded in its own small surface code patch. The entangling operation would then be realized by merging this ancilla patch with the data qubit patch, performing a joint Pauli operator measurement (e.g., $Z_L Z_A$), and then splitting the patches apart again [1, 5]. This approach is particularly well-suited for architectures with static qubits and local connectivity, transforming the problem of complex gate sequences into one of dynamically reconfiguring the boundaries of the error-correcting code itself. The choice between these implementation strategies depends on a complex trade-off between hardware capabilities, error rates, and the overall resource cost of the computation.

References

[1] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, *Surface codes: Towards practical large-scale quantum computation*, Phys. Rev. A **86** (2012) 032324, [arXiv:1208.0928].

[2] A. Y. Kitaev, *Fault-tolerant quantum computation by anyons*, Annals of Physics **303** (2003), no. 1 2–30, [quant-ph/9707021].

[3] S. B. Bravyi and A. Y. Kitaev, *Quantum codes on a lattice with boundary*, quant-ph/9811052.

[4] C. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, *Surface code quantum computing by lattice surgery*, New Journal of Physics **14** (2012), no. 12 123011, [arXiv:1111.4022].

[5] D. Litinski, *A game of surface codes: Performing meaningful quantum computations with available qubits*, Quantum **3** (2019) 128, [arXiv:1808.02892].

[6] N. de Beaudrap and D. Horsman, *The ZX calculus is a language for surface code lattice surgery*, Quantum **4** (2020) 218, [arXiv:1704.08670].

[7] S. Bravyi and A. Kitaev, *Universal quantum computation with ideal clifford gates and noisy ancillas*, Phys. Rev. A **71** (Feb, 2005) 022316, [quant-ph/0403025].

[8] B. J. Brown, *A fault-tolerant non-clifford gate for the surface code in two dimensions*, Science Advances **6** (2020), no. 21 eaay4929, [arXiv:1903.11634].

[9] Y. Vaknin, S. Jacoby, A. Grimsmo, and A. Retzker, *Efficient magic state cultivation on the surface code*, arXiv:2502.01743.

[10] D. H. Kim, J.-K. K. Kim, H.-S. Kim, S.-W. Lee, J. Kim, and S.-K. Kwon, *Magic state injection on ibm quantum processors above the distillation threshold*, arXiv:2412.01446.