Entangling logical qubits with lattice surgery

The development of quantum computing architectures from early designs and current noisy devices to fully fledged quantum computers hinges on achieving fault tolerance using quantum error correction. However, these correction capabilities come with an overhead for performing the necessary fault-tolerant logical operations on logical qubits (qubits that are encoded in ensembles of physical qubits and protected by error-correction codes). One of the most resource-efficient ways to implement logical operations is lattice surgery, where groups of physical qubits, arranged on lattices, can be merged and split to realize entangling gates and teleport logical information. Here we report the experimental realization of lattice surgery between two qubits protected via a topological error-correction code in a ten-qubit ion-trap quantum information processor. In this system, we can carry out the necessary quantum non-demolition measurements through a series of local and entangling gates, as well as measurements on auxiliary qubits. In particular, we demonstrate entanglement between two logical qubits and we implement logical state teleportation between them. The demonstration of these operations—fundamental building blocks for quantum computation—through lattice surgery represents a step towards the efficient realization of fault-tolerant quantum computation.

Surface code

One of the most prominent examples of a QEC code is the surface code, which has error thresholds of up to 1%. The surface code has a simple description within the stabilizer formalism, as we discuss in the following. Note that we work with qubits, that is, two-level quantum systems whose state space is defined by Pauli operators. The surface code as the central component of our experimental implementation. The code can be represented graphically, where the physical qubits are the vertices of a 2 x 2 bicolourable lattice, as shown in Fig. 1A, a (row 'Schematic', column 'Encoded') for two initially separate logical qubits labelled A and B. Depending on the colour, faces are associated with products of either Pauli-X, Pauli-Y, or Pauli-Z operators on the adjacent physical qubits. In Fig. 1A, a (Schematic, Encoded)
for example, the central, orange plaquettes can be associated with operators $X_iX_jX_k$ and $X_iX_jX_l$. The resulting operators are called stabilizers and form a set (group) of operations—the stabilizer code $S^{AB}$—under multiplication,

\[
S^A = (S_{1A}^A, S_{2A}^A, S_{3A}^A) = (-Z_iZ_j, -Z_iZ_k + X_iX_jX_k),
S^B = (S_{1B}^B, S_{2B}^B, S_{3B}^B) = (-Z_iZ_l, -Z_iZ_k + X_iX_jX_l).
\]

Note that we choose a negative sign for some stabilizers because this is advantageous for our implementation. The logical states $|\psi_{AB}^1\rangle$ spanning the respective code spaces for $A$ and $B$ are defined as the simultaneous $\pm 1$ eigenstates of all stabilizers, that is, $S_{1A}^A|\psi_{AB}^1\rangle = |\psi_{AB}^1\rangle$, $S_{2B}^B|\psi_{AB}^1\rangle = |\psi_{AB}^1\rangle$, $\forall i \in \{1, 2, 3\}$. Therefore, we find that each surface encodes a single logical qubit in two codewords, $|\psi_{AB}^1\rangle = \frac{1}{\sqrt{2}}(|1010\rangle + |1010\rangle)$ and $|\psi_{AB}^2\rangle = \frac{1}{\sqrt{2}}(|1010\rangle - |1010\rangle)$.

Logical operators map codewords to codewords. For instance, a logical bit-flip operator $X^i_0$ maps $|0\rangle^i$ to $|1\rangle^i$. In the stabilizer formalism, logical $X$- and $Z$-operators anti-commute with each other (that is, they obey the Pauli commutation relations) and commute with all stabilizers. For example, $X^i_0$-anti-commutes with the logical phase-flip operator $Z^i_0 = Z_iZ$ but commutes with $S^i_0$. These operators are defined up to multiplication with other logical operators, stabilizers and the imaginary unit $i$. That is, the sets of logical operators are defined as

\[
L^A = \{i, Z^i_0, X^i_0\}/S^A = \{i, Z_iZ_j, X_iX_j\}/S^A,
L^B = \{i, Z^i_0, X^i_0\}/S^B = \{i, Z_iZ_j, X_iX_j\}/S^B,
\]

where $\langle R \rangle/S$ indicates that logical Pauli operators $P_i$ form equivalence classes defined up to multiplication with stabilizers (see Methods). The logical $Y$-operator is determined as $Y_i = iZ_iX_i$ and $Y^i_0 = iX_iZ_i$.

In the stabilizer formalism, errors can be associated with finding an eigenvalue $-1$ when measuring a stabilizer operator. To see this, consider a bit-flip error represented by $X_i$ which can be detected by extracting the eigenvalue of $S^i_0$, $S^i_0|\psi_{AB}^1\rangle = -X_i|\psi_{AB}^1\rangle$. This eigenvalue, or error syndrome, can be obtained by measuring an auxiliary qubit (or syndrome qubit) encoding the $\pm 1$ eigenvalue of the associated stabilizer. Scaling the surface code to detect and correct more errors is, in theory, as simple as scaling the lattice (see Methods).

**Lattice surgery**

LS is a fault-tolerant protocol for entangling QEC codes that is ideally suited to the geometry of 2D topological codes such as the surface code. This is because LS between topological codes requires only local, few-body interactions. LS was introduced as a method to project two surface codes $S^A$ and $S^B$ with logical operators $X^i_A, Z^i_A$ and $X^i_B, Z^i_B$ respectively, onto joint eigenstates of either $X^i_A$ or $Z^i_B$, referred to as $Z$-type and $X$-type LS, respectively. These projections are entangling operations and can be used to construct entangling gates. To clarify the notation, note that the lattice boundaries of surface codes can be distinguished by their associated stabilizers: $Z$-type stabilizers along
the boundary define one type of boundary while \(X\)-type stabilizers define another. Similarly, we label different types of LS by the boundary type they operate on. Here, we proceed by describing \(Z\)-type LS for the minimal \(2 \times 2\) surface code discussed before, and refer to Methods for a more general introduction and details.

In order to project onto a logical eigenstate of \(X^l_1X^l_2\), we perform a logical joint measurement \(M^\text{SS}_{XX} = (1 \pm X^l_1X^l_2)/2\), which can be used to entangle two logical qubits, and where \(\mathcal{L}\) is the logical two-qubit identity matrix. To achieve this, LS proceeds in two steps: merging and splitting. This procedure is illustrated in Fig. 1B, a (Schematic, Merged) and Fig. 1C, a (Schematic, Split) for two \(2 \times 2\) surface codes \(S^A\) and \(S^B\). We first merge the two separate codes \(S^A, S^B\) into a new stabilizer code \(S^M\) by measuring merging stabilizers \(S^m^A = X^L_1X^L_2\) and \(S^m^B = X^L_2X^L_1\) between the boundaries. These stabilizers commute with all stabilizers of the original codes except \(S^{L_2}_{1} \) and \(S^{L_2}_{1}\), and are chosen such that their joint measurement corresponds to the joint logical measurement \(M^\text{SS}_{XX}\), that is, \(S^m^A S^m^B = X^L_1X^L_2\). As a result, we obtain the new code by discarding all stabilizers that anti-commute with the merging stabilizers, depicted in Fig. 1B, a (Schematic, Merged),

\[
S^M = (S^A_1, S^A_3, S^A_5, S^B_4, S^B_6), \quad (S^A_2, S^A_4, S^A_6, S^B_2, S^B_4), \quad X^L_1X^L_2, \quad X^L_2X^L_1.
\]

Note that this code already encodes the desired joint eigenstate since \(X^l_1X^l_2\) is included as a stabilizer in the merged code \(S^M\). In fact, the measurement outcomes \(m, m' \in \{0, 1\}\) of \(S^A_1, S^A_3, S^A_5\), respectively, are random such that \(m = m'\) specifies the eigenvalue associated with \(X^l_1X^l_2\) as \((-1)^{m_1}\). The merged code is an asymmetric \(2 \times 4\) surface code encoding a single logical qubit, that is,

\[
L^M = (i, Z^L_1X^L_1)/S^M = (i, Z^L_1Z^L_2, X^L_1)/S^M,
\]

and \(y^M = Y^L_1X^L_2\).

With the \(Z\)-type merge we effectively merged the logical \(Z\)-operators and performed the desired logical joint measurement \(M^\text{SS}_{XX}\). Its expectation value \(\pm 1\) is given by the product of the expectation values of merging stabilizers \(S^m_1, S^m_2\). Now, we must recover the two initial logical qubits while keeping the previously obtained expectation value of \(X^l_1X^l_2\). To this end, we split the merged code by measuring \(Z\)-stabilizers \(S^Z_2\) or \(S^Z_3\) along the merged boundaries as depicted in Fig. 1C, a (Schematic, Split). These operators commute with all stabilizers in \(S^M\) that define the separated logical qubits \(S^A, S^B\). In particular, the measured stabilizers all commute with \(X^l_1, X^l_2\), that is, the code remains in an eigenstate of \(X^l_1X^l_2\). After splitting, measurement outcomes \(m'\) and \(m''\) in \(\{0, 1\}\) of stabilizers \(S^Z_2, S^Z_3\), respectively, are random but can be tracked as errors. In conclusion, we have effectively performed a logical entangling operation, \(M^\text{SS}_{XX}\), which can be used to entangle logical qubits and teleport information. Note that the scheme presented here differs from the original proposal in that it removes the necessity of additional data qubits along the boundary. Although this change generally does not affect fault tolerance and code distance, it might alter the QEC capabilities of the underlying code by effectively changing the lattice on which the surface code is defined.

LS can further be used to realize a measurement-based scheme for local logical teleportation. In Fig. 2, we illustrate this scheme for a logical \(M^\text{SS}_{XX}\) measurement on two \(5 \times 5\) surface codes. We use \(5 \times 5\) surface codes in this theoretical example to exemplify how LS scales with increasing system size. The merging and splitting operations used to experimentally teleport quantum information between two \(2 \times 2\) surface codes are illustrated in Fig. 1. Note that a similar scheme can be used to teleport information through a logical \(M^\text{SS}_{ZZ}\) measurement (see Methods).

**Results**

We demonstrate LS in an ion-trap quantum computer, based on atomic \(^{40}\)Ca\(^+\) ions in a linear Paul trap\(^\text{a}\). Each qubit is encod in the \(|0⟩ = |4S_{\frac{1}{2}}(m_1=\mp 1/2)⟩\) and \(|1⟩ = |3D_{\frac{3}{2}}(m=\pm 1/2)⟩\) state of a single ion. Each experiment consists of (i) laser cooling and state preparation, (ii) coherent manipulation of the qubits states, and (iii) readout. (i) For
We can \( Y \) and \( X \) teleport the state from logical qubit A to logical qubit B as illustrated in Fig. 2. Logical operators (see equation (2)) in Fig. 1A (Encoded). As a first example, we choose to encode the logical qubits in the state \(|0\rangle\) = 0.669(8). Starting from the state \(|0\rangle\), the merged logical state is a +1 eigenstate of the logical \( Z^L \) operator, as can be seen in Fig. 1B, c (Logical operators, Merged). The data reveal a state fidelity of \( \mathcal{F}(|\phi_1^L\rangle) = 86.4(1.0)\% \) after merging.

Now, we split the merged logical qubit along the same boundary by mapping \( S^Z \) onto auxiliary qubit A, for the case \( m^* = 0 \). Thereby, we restore the initial code space with an average stabilizer expectation value of \( \langle S_i \rangle = 0.603(3) \), shown in Fig. 1C, b (Code stabilizers, Split). The resulting projective measurement \( (|+\rangle X^L |+\rangle L) \) maps the initial product state \(|0^L_0\rangle \) onto a maximally entangled, logical Bell state \(|\phi_1^L\rangle = \frac{1}{\sqrt{2}} (|0^L_0\rangle + |1^L_1\rangle) \). In order to deduce the fidelity of the generated state with respect to the logical Bell state, we measure the common logical stabilizers \(|Z^L_i|Z^L_i|X^L_i X^L_i|Y^L_i Y^L_i\rangle\), obtaining the fidelity \( \mathcal{F}(|\phi_1^L\rangle) = 58.0(1.6)\% \), where the raw fidelity exceeds the separability limit of 50% by 5 standard deviations. Imperfect physical gate implementations can be characterized and match our expectations, as discussed in Supplementary Information. In Supplementary Information, we also demonstrate LS for various input states in order to generate different maximally entangled Bell states.

LS enables the teleporting of quantum states from one logical qubit to another (see Fig. 2), which we demonstrate for the input states \(|0^L_0\rangle\), \(|0^L_2\rangle\) and \(|1^L_2\rangle\). After performing \( Z \)-type LS (that is, encoding, merging, splitting), we measure logical qubit A in the \( Z \) basis and apply a logical \( X \) gate on qubit B if qubit A was found in \( |1^L\rangle \) (see Fig. 3). Following the teleportation protocol, we measure logical state fidelities for qubit B of \( \mathcal{F}(|\phi_1^L\rangle) = 86.2(1)% \), \( \mathcal{F}(|\phi_2^L\rangle) = 81.2(2)% \) and \( \mathcal{F}(|\phi_4^L\rangle) = 71.1(8)% \), given the input states \(|0^L_0\rangle\), \(|0^L_2\rangle\) and \(|1^L_2\rangle\), respectively.

### Conclusion

We have demonstrated entanglement generation and teleportation via LS between two logical qubits, each encoded in a four-qubit surface code, on a 10-qubit ion-trap quantum information processor. We have implemented both the \( Z \)- and \( X \)-type variants of LS, a technique that is considered key for operating future fault-tolerant quantum computers. For current NISQ (noisy intermediate-scale quantum)-era devices, certification of logical entanglement generated via LS can provide the means of benchmarking. Besides increasing the numbers of physical and logical qubits, future challenges lie in the implementation of LS between arbitrary topological codes to exploit different features, such as transversal gate implementation or high noise tolerance of the respective codes. In this way, LS can function as a fault-tolerant interface between quantum memories and quantum processors.
22. Reed, M. D. et al. Realization of three-qubit quantum error correction with
23. Waldherr, G. et al. Quantum error correction in a solid-state hybrid spin register.
20. Andersen, C. K. et al. Repeated quantum error detection in a surface code.
17. Takita, M., Cross, A. W., Córcoles, A. D., Chow, J. M. & Gambetta, J. M. Experimental
16. Bell, B. A. et al. Experimental demonstration of a graph state quantum error-correction
15. Wootton, J. R. & Loss, D. Repetition code of 15 qubits.
13. Boulant, N., Viola, L., Fortunato, E. M. & Cory, D. G. Experimental implementation of a
12. Chiaverini, J. et al. Realization of quantum error correction.
11. Horsman, C., Fowler, A. G., Devitt, S. & Van Meter, R. Surface code quantum computing by
10. Gutiérrez, M., Müller, M. & Bermúdez, A. Transversality and lattice surgery: exploring
9. Poulsen Nautrup, H., Friis, N. & Briegel, H. J. Fault-tolerant interface between quantum
8. Eisert, J., Jacobs, K., Papadopoulos, P. & Plenio, M. B. Optimal local implementation of
7. Terhal, B. M. Quantum error correction for quantum memories.
6. Bell, B. A. et al. Quantum error correction beyond qubits. Nat. Phys. 5, 541–546 (2009).
5. Gottesman, D. Stabilizer Codes and Quantum Error Correction. Ph.D. thesis, Caltech (1997).
4. Campbell, E. T., Terhal, B. M. & Vuillot, C. Roads towards fault-tolerant universal quantum
3. Terhal, B. M. Quantum error correction for quantum memories.
2. Devitt, S. J., Munro, W. J. & Nemoto, K. Quantum error correction for beginners.
**Methods**

**Stabilizer quantum error correction**

Quantum error correction (QEC) deals with the encoding and protection of quantum information stored in quantum systems. The simplest such systems are qubits, two-level (for example, spin-$\frac{1}{2}$) systems, whose degrees of freedom can be represented by the Pauli matrices

\[
\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.
\]

Pauli matrices, in conjunction with the identity $I$, form a basis of the vector space of $2 \times 2$ Hermitian matrices, and so an arbitrary single-qubit density matrix $\rho$ can be parameterized as $\rho = \frac{1}{2} (I + \alpha \cdot \sigma)$ where $\alpha \in \mathbb{C}^3$ is the so-called Bloch vector and $\alpha = (\alpha_x, \alpha_y, \alpha_z)^T$. Qubits can therefore be described by Pauli observables. For example, instead of writing $|0\rangle \langle 0|$, we may write $(I + Z)/2$ where we use $X, Y, Z$ as shorthand for Pauli matrices $\sigma_x, \sigma_y, \sigma_z$ and $|0\rangle, |1\rangle$ are the two eigenstates of $Z$ with eigenvalues $\pm 1$, respectively.

Interestingly, there are various ways of encoding a logical qubit in composite systems. Specifically, we may encode a qubit in either single- or multi-qubit observables. To see this, consider a pure two-qubit system $C^2 \otimes C^2 = \text{span}(|00\rangle, |01\rangle, |10\rangle, |11\rangle)$. Let us now fix the second qubit to the +1 eigenstate of $Z_2$, that is, $|0\rangle_2$. Since this is a composite system, the eigenvalue is degenerate and we can define a single qubit by span$(|0\rangle_2, |1\rangle_2)$ where subscripts label qubits. However, we may also fix the composite system to the +1 eigenstate of $Z_2$. We then can still define a qubit by span$(|0\rangle, |1\rangle)$.

This qubit is associated with a new set of logical Pauli observables $X_L, Y_L, Z_L = [X_1X_2, Y_1X_2, Z_1X_2]$, which obey the same commutation relation as $X_1, Y_1, Z_1$. This notion of encoding qubits in composite systems is used in QEC to protect quantum information.

In QEC our aim is to encode a few logical qubits into many physical qubits such that redundancies can be exploited to detect and correct errors. That is, we replace single-qubit basis states $|0\rangle, |1\rangle$ by encoded logical states $|0_L\rangle, |1_L\rangle$. Here we consider a general construction of surface codes in the stabilizer formalism.

**Surface code**

Here we consider a general construction of surface codes in the stabilizer formalism. Consider $n$ qubits laid out on the vertices $V$ of a bicoloured square lattice as displayed in Extended Data Fig. 1. Let us associate a stabilizer with each coloured plaquette $p \in P$ as follows,

\[
S_p^X = \prod_{v \in \Lambda(p)} X_v,
\]

\[
S_p^Z = \prod_{v \in \Lambda(p)} Z_v,
\]

which can be used to generate a group $S$ under multiplication, that is, $S = \{S_p^X, S_p^Z\} = \{1\}$. This group is called the stabilizer group and contains all stabilizers for this codespace. Since the code subspace is an eigenspace of these operators, we can simultaneously measure all stabilizers without disturbing the logical information. Without errors, measuring stabilizers will always result in the same outcome, namely $+1$. However, were an error $X_1$ to occur, the measurement outcome of stabilizer $S_1$, its so-called syndrome $s_1$, would change sign since errors anti-commute with stabilizers, that is, $[S_1, X_1] = 0$. The only other combination of $X$-errors that could possibly lead to the syndromes $s_1 = -1$ and $s_2 = +1$ is a two-qubit error $X_1X_2$. In practice, we collect these syndromes by measuring auxiliary qubits, so-called syndrome qubits, which encode the eigenvalues of associated stabilizers. That is, projective stabilizer measurement can be performed by entangling data qubits with a syndrome qubit and measuring the latter. As a result, we end up with the same majority vote as before but without measuring the logical state of the encoded qubit. This is the convenience of the stabilizer formalism.

In this formalism, logical operations take a simple form as the normalizer $N(S) \subseteq P_3$ of the stabilizer group which is the group of operators that leaves the stabilizer group invariant. We are only considering Pauli operators, and hence the normalizer is also the centralizer $C(S)$ of $S$ which is the group of operators that commutes with all stabilizers. Since this definition includes stabilizers themselves, we define the group of logical operators as a quotient group $L = N(S)/S$ such that logical operators form equivalence classes under multiplication with stabilizers. In our case, the equivalence classes are $[I], [X], [Y], [Z], [X_1X_2], [X_1Z_2], [X_2Z_1], [Z_1Z_2]$ that is, one for each logical operation.

QEC is done to protect encoded information from non-trivial logical errors in $Z$. Since we are only considering products of Pauli operators, elements of $L$ are also just products of Pauli operators. This allows us to infer the minimum number of single-qubit errors composing a logical error, that is, its distance $d$. To see this, consider the non-trivial operator $Z_1 \in L$ and its weight $w(Z_1)$ which is the number of non-trivial terms in the product of Pauli operators. In our example, $Z_1 = Z_1$, that is, its weight is 1 and a single-qubit $Z$-error can cause a logical $Z$-error. In other words, the above code can tolerate no $Z$-errors and its distance is therefore $d = 1$. However, with respect to logical $X$-operators $[X_1X_2]$, the minimum weight of any logical $X$-operator is 3 such that the code can correct 1 and detect 2 $X$-errors. Since the correction procedure is based on majority voting, a code with distance $d$ can generally correct up to $(d - 1)/2$ errors and detect up to $d - 1$ errors. The distance of the code is also the minimal Hamming distance between codewords, that is, the minimum required number of single-qubit Pauli operators mapping any one codeword to another.

In summary, QEC in the stabilizer formalism is active in the sense that we are required to measure stabilizers and extract syndromes throughout a quantum computation. The syndromes can then be analysed to determine by a majority vote the errors that have occurred. Logical operators are operators that commute with all stabilizers but are not stabilizers themselves.
where \( \delta(p) \subseteq \mathcal{V} \) is the set of vertices neighbouring a plaquette \( p \) and \( \mathcal{V} \) is the set of faces. \( X \)-stabilizers \( S^X \) are placed on orange plaquettes while \( Z \)-stabilizers \( S^Z \) are placed on aquamarine plaquettes. Since neighbouring plaquettes always share two vertices, stabilizers commute for all \( p \in \mathcal{P} \). For the lattice under consideration, there are \( s = n - 1 \) independent, commuting stabilizers. Therefore, the Hilbert space, which is the simultaneous +1 eigenstate of all stabilizers, has \( n - s = 1 \) degree of freedom. This degree of freedom is a qubit since we can define logical \( X_L \) and \( Z_L \) Pauli operators. In the case of the surface code, logical operators are products of Pauli operators connecting opposite boundaries of the lattice. To see this, consider a line drawn on the lattice connecting top and bottom boundaries, as indicated by dashed frames in Extended Data Fig. 1. Placing \( X \)-operators on vertices enclosed by this frame, we obtain an operator commuting with all stabilizers but which is not a stabilizer itself. Therefore, this operator corresponds to a logical operator \( X_L \). At the same time, we can analogously draw a line connecting left and right boundaries. Placing \( Z \)-operators along this line, we obtain an operator commuting with all stabilizers but anti-commuting with \( X_L \). Therefore, this product of Pauli-\( Z \)-operators defines the logical \( Z \)-operator \( Z_L \). Note that the shortest line connecting opposite boundaries crosses three vertices. Therefore, the code can correct up to one single-qubit error and has distance \( d = 3 \).

In order to perform QEC, we continuously measure the code stabilizers. Whenever a stabilizer measurement result, that is, its syndrome, changes sign from +1 to -1, we have detected an error. Assuming that less than \((d - 1)/2 \) errors have occurred, we can associate with each syndrome a correction procedure which recovers the state of all +1 stabilizers from the erroneous state without causing a logical error.

**Lattice surgery**

Here, we consider lattice surgery (LS) in general as a method to project onto a joint eigenstate of logical Pauli operators. That is, LS maps two stabilizer QEC codes \( S^X, S^Z \) into a joint eigenstate \( P^X_L \otimes P^Z_L \) of two logical Pauli operators of the codes. This is achieved through a joint measurement \( M^X_{PP} = (1 \pm P^X_A \otimes P^X_B) \) which can be implemented fault-tolerantly. Note that an operation is called fault-tolerant if errors during the operation can only map to a constant number of physical qubits in the encoding independent of the code size. In Extended Data Fig. 2, we illustrate how joint Pauli measurements as described above can be used to implement logical CNOT and Hadamard \( H \) gates as well as code teleportation through a measurement-based scheme.

LS itself proceeds in two steps: merging and splitting. In order to initialize a measurement \( M^X_{PP} \), we first merge the two separated codes \( S^X, S^Z \) into a new stabilizer code \( S^M \) by projecting onto a joint eigenstate \( P^X_L \otimes P^Z_L \). In order for this to be fault-tolerant, we measure a number of so-called merging stabilizers \( \{ S^M \} \) across the boundary such that \( \frac{1}{\sqrt{2}} S^M = P^X_L \otimes P^Z_L \). This is displayed for the surface code in Extended Data Fig. 3 where we consider \( S^X, S^Z \) to be \( 2 \times 2 \) surface codes and \( R = R_L \otimes R_R \). Then, the merged code is just a new surface code on an asymmetric lattice and the merging stabilizers are just surface code stabilizers at the interface between the two codes. Stabilizers at the boundary that do not commute with the merging stabilizers are discarded from the stabilizer group and only the product of boundary operators remain since they commute. Notably, the merged code encodes only a single logical qubit and \( P^X_L \otimes P^Z_L \) is contained as a stabilizer. That is, this procedure projects onto an eigenstate of \( P^X_L \otimes P^Z_L \), its eigenvalue \( \pm 1 \) is determined by the measurement outcome of the product of merging stabilizers. In order to correct for measurement errors, we need to measure \( \{ S^M \} \) times. These errors correspond to failed measurements that yield a syndrome \( s \) although its expectation value is \( -s \). Such measurement errors can be identified by comparing measurement results at different times.

Now, we want to recover the two initial logical qubits while remaining in an eigenstate of \( P^X_L \otimes P^Z_L \). To this end, we split the merged code by measuring stabilizers of the separated codes \( S^X, S^Z \) along the aligned boundaries as illustrated in Extended Data Fig. 3C rightmost column (Split). Since these stabilizers anti-commute with merging stabilizers, the set \( \{ S^M \} \) is discarded from the stabilizer groups and we recover the original two codes. However, since all stabilizers always commute with the logical operators, the resulting state remains an eigenstate of \( P^X_L \otimes P^Z_L \). At the end, QEC is required to ensure full fault-tolerance. Surface code LS usually distinguishes \( Z \)-type and \( X \)-type LS by association with the respective boundaries along which LS is performed. Note that \( Z \)-type and \( X \)-type LS therefore refer to a projection onto an \( X^M \) or \( Z^M \) eigenstate, respectively. Let us further point out that one may encounter a different terminology in terms of rough and smooth LS in the literature\(^a\). That notation is due to a different surface code representation and may be associated with what we call \( Z \) - and \( X \)-type LS, respectively. Generally however, we do not restrict to \( Z \)-type and \( X \)-type LS alone since a projection onto \( Z^M \) can be used to generate a logical Hadamard as shown in Extended Data Fig. 2.

**Results of \( X \)-type LS**

\( X \)-type LS differs from \( Z \)-type LS as described in the main text only in so far that both codes are considered to be rotated by 90° before LS. Equivalently, one can understand \( X \)-type LS as merging and splitting along the upper/lower instead of the left/right boundaries, as illustrated in Extended Data Fig. 3b (bottom row; \( X \)-type). In the case of two \( 2 \times 2 \) surface codes, measuring the merging stabilizer \( S^M_i \) at \( Z_i \) yields an asymmetric surface code

\[
S^M = \{ S^M_1, S^M_2, S^M_3, S^M_4 \} = X^M \otimes X^M \otimes \overline{Z^M} \otimes \overline{Z^M}.
\]

which can be split by discarding the merging stabilizer. It is worth noting that the merging operation already concludes the LS since the merging stabilizer is in fact \( Z^M \otimes \overline{Z^M} \). This is an artefact of reducing the general LS procedure as exemplified for a larger surface code in Fig. 2 to distance-two surface codes.

We present the results for the \( X \)-type LS in Extended Data Fig. 4 and report a Bell state fidelity of \( F(\phi^+;\phi^-) = 63.9(2.8)/78(2.7)% \). Further measurement results for various input states can be found in Supplementary Information.

**Data availability**

The data that support the findings of this study are available at https://doi.org/10.5281/zenodo.4081412.

**Code availability**

All codes used for data analysis are available from the corresponding authors upon reasonable request.

---

\( a \) Note that a different terminology in terms of rough and smooth LS in the literature is used, which may be associated with what we call \( Z \)- and \( X \)-type LS, respectively. Generally, however, we do not restrict to \( Z \)-type and \( X \)-type LS alone since a projection onto \( Z^M \) can be used to generate a logical Hadamard as shown in Extended Data Fig. 2.
**Author contributions** A.E., H.P.N., P.S. and N.F. wrote the manuscript and A.E., H.P.N., M.M., L.P., R.S., M.R., P.S., H.J.B., R.B., N.F. and T.M. provided revisions. A.E., H.P.N., P.S. and N.F. developed the research based on discussions with H.J.B., R.B. and T.M. H.P.N. and N.F. developed the theory. A.E. and P.S. performed the experiments. A.E., M.M., L.P., R.S., M.S., V.N., M.R., P.S., R.B. and T.M. contributed to the experimental setup. A.E., H.P.N., M.M., L.P., R.S., M.R., P.S., H.J.B., R.B., N.F. and T.M. contributed to discussions of the results and the manuscript.

**Competing interests** T.M. and R.B. are founding members of Alpine Quantum Technologies GmbH.

**Additional information**
Supplementary information is available for this paper at https://doi.org/10.1038/s41586-020-03079-6.

**Correspondence and requests for materials** should be addressed to N.F. or T.M.

**Peer review information** Nature thanks Rodney Van Meter, Theodore Yoder and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.

**Reprints and permissions information** is available at http://www.nature.com/reprints.
Extended Data Fig. 1 | Standard surface code of distance 3. The standard surface code is defined on a square lattice with (data) qubits located on vertices. Stabilizers are associated with faces and boundaries. Aquamarine faces and boundaries indicate $Z$-type stabilizers, as in equation (6). Red faces and boundaries indicate $X$-type stabilizers, as in equation (5). The surface code with boundaries encodes a single logical qubit defined by its logical Pauli-$X$ and Pauli-$Z$ operators. These operators are defined on paths connecting opposite boundaries of the lattice and act as products of $X$- and $Z$-operators, respectively, along the paths. Here, two representative logical operators are drawn as products of Pauli-operators within the dashed rectangles. Red indicates Pauli-$X$ operators and green indicates Pauli-$Z$ operators. The two operators anti-commute at the crossing drawn in yellow.
Lattice surgery (LS) enables measurement-based implementations of logic gates and logical state teleportation. LS operations are logical joint measurements of the form \( M_{PP} = (\pm \hat{P} \hat{P})/2 \) where \( \hat{P} \) are Pauli operators. Moreover, the protocols make use of single-qubit measurements of the form \( M_{P} = (I \pm \hat{P})/2 \). Thick lines indicate logical qubits in the circuit model and double lines represent classical bits indicating measurement outcomes \( m_i = 0, 1 \). Pauli corrections need to be applied which are conditioned on the measurement outcomes as \( \hat{P}_m \). The symbol \( \oplus \) represents an XOR gate between classical bits.

**a** Controlled-Not. Measurement-based implementation of a logical CNOT-gate between arbitrary control and target qubits requiring an auxiliary qubit in \( |+\rangle \) state. **b** X-type teleportation. Measurement-based teleportation protocol for state teleportation between two logical qubits using X-type LS. **c** Hadamard. Measurement-based implementation of a logical Hadamard gate \( H \) based on the teleportation protocol.
Extended Data Fig. 3 | Surface code lattice surgery in theory. Surface code LS between $Z$-type and $X$-type boundaries implementing logical joint measurements $M_{XX} = (\pm) /2$ and $M_{ZZ} = (\pm) /2$, respectively. 

A. Encoded. The two initial surface codes are defined on $2 \times 2$ lattices where $X$-stabilizers are associated with orange faces and $Z$-stabilizers with aquamarine faces in accordance with equation (1). Logical operators are products of Pauli operators connecting opposite boundaries as in equation (2). 

B. Merged. Treating the two codes as a single (asymmetric) surface code, (merging) stabilizers along the boundaries are measured. The merged code encodes a single logical qubit corresponding to the logical Pauli operators $X_L^M, Z_L^M$. 

C. Split. In order to split the merged code while preserving the eigenstate of the joint logical operator, the boundary stabilizers of the original code are measured. These operators anti-commute with the merging stabilizers and thus project onto the individual codes. Since the boundary operators commute with individual logical operators, the resulting state remains an eigenstate of the joint logical operator. 

a. $Z$-type encoded. The two initial surface codes are defined on $2 \times 2$ lattices where $X$-stabilizers are associated with orange faces and $Z$-stabilizers with aquamarine faces in accordance with equation (1). Logical operators are products of Pauli operators connecting opposite boundaries as in equation (2). 

b. $X$-type encoded. The surface codes are aligned along their $Z$-type boundary. 

b. $X$-type merged. Merging stabilizers (indicated in green) are chosen such that their product is $X_L^M X_L^M$. 

b. $X$-type split. Measuring $X$-stabilizers along the boundary (indicated in red) preserves the eigenstate of $Z_L^M Z_L^M$ while projecting onto the individual codes.
Extended Data Fig. 4 | Experimental X-type surface code lattice surgery.

Bell state generation via lattice surgery along the X-type boundary between two surface code qubits through a logical joint measurement $M_{I_M} = I + Z_A^I Z_B^I$.

Post-selected measurements are presented in light coloured bars. A, Encoded. Two logical qubits (a) are encoded with average stabilizer values (b) of $\langle |S_i|\rangle = 0.813(4)$. We observe raw and post-selected state fidelities (c) of $\mathcal{F}(|0\rangle) = 93.3(5)\%$ for logical qubit A and $\mathcal{F}(|0\rangle) = 92.4(5)\%$ for logical qubit B.

B, Merged. The two separated logical qubits are merged (a) into a single logical qubit by measuring the stabilizer $S_7^M$ using auxiliary qubit $A_1$ as syndrome qubit. Thereby, the code space is extended in the vertical direction and the new logical operator $X_{X^X} = X_A^I X_B^I$ is formed. As the data show, the stabilizer $S_7^M$ is indeed created. The average stabilizer values (b) are $\langle |S_i|\rangle = 0.719(5)$ and logical state fidelities (c) are $\mathcal{F}(|+\rangle) = 76.2(8)\%$ for logical qubit A, $\mathcal{F}(|+\rangle) = 78.0(2.7)\%$ for logical qubit B. C, Split. The single logical qubit is again split into two logical qubits (a) along the same boundary they have been initially merged through. We measure the stabilizer $S_6^M$ by using auxiliary qubit $A_2$ as syndrome qubit to perform the splitting and obtain average stabilizer values (b) of $\langle |S_i|\rangle = 0.763(5)$. The fidelity (c) of the generated state to the logical Bell state is $\mathcal{F}(|\psi^+\rangle) = 63.9(2.8)\%$. Note that measuring the merging stabilizer $S_7^M = Z_A^I Z_B^I$ directly projects onto a joint eigenstate of the logical Z-operators such that the splitting becomes redundant. Nevertheless, the general procedure as described in Methods requires the measurement of X-stabilizers along the boundary which is why it is still included here.