

Addressable gate-based logical computation with quantum LDPC codes

Laura Pecorari,^{1,*} Francesco Paolo Guerri,^{1,*} Hugo Perrin,^{1,2,*} and Guido Pupillo^{1,3,†}

¹*University of Strasbourg and CNRS, CESQ and ISIS (UMR 7006), aQCESS, 67000 Strasbourg, France*

²*QPerfect, 23 Rue du Loess, 67200 Strasbourg, France*

³*Institut Universitaire de France (IUF), 75000 Paris, France*

(Dated: November 11, 2025)

Quantum computing relies on quantum error correction for high-fidelity logical operations, but scaling to achieve near-term quantum utility is highly resource-intensive. High-rate quantum LDPC codes can reduce error correction overhead, yet realizing high-rate fault-tolerant computation with these codes remains a central challenge. Apart of the lattice surgery approach, standard schemes for realizing logical gates have so far been restricted to performing global operations on all logical qubits at the same time. Another approach relies on low-rate code switching methods. In this work, we introduce a gate-based protocol for addressable single- and multi-qubit Clifford operations on individual logical qubits encoded within one or more quantum LDPC codes. Our scheme leverages logical transversal operations via an auxiliary Bacon-Shor code to perform logical operations with constant time overhead enabled by teleportation. We demonstrate the implementation of an overcomplete logical Clifford gate set and perform numerical simulations to evaluate the error-correction performance of our protocol. Finally, we observe that our scheme can be integrated with magic state cultivation protocols to achieve universal, gate-based, and fully addressable quantum computation.

Executing parallel logical operations on error-corrected logical qubits is key to enable quantum computation. Quantum error correction (QEC) uses redundancy to delocalize the state of individual logical qubits over several physical qubits to increase protection against errors [1]. This comes at the price of large qubit and gate overheads, which ultimately make QEC highly resource-intensive and pose challenges to scalability.

High-rate quantum Low-Density Parity-Check (LDPC) codes [2] offer a promising route toward low-overhead, error-corrected quantum memories. By exploiting long-range connectivity, these codes can compactly encode a large number of logical qubits within a single quantum register and have been shown to outperform standard locally connected codes, such as surface codes [3, 4], under circuit-level noise models [5–7]. However, this dense information encoding poses a major challenge for realizing high-rate quantum computation on near-term quantum processors.

The standard theoretical framework to date for performing quantum computation with high-rate quantum LDPC codes uses lattice surgery to measure logical Pauli operators [8–17]. This paradigm consists in filling the region between distant code patches with additional ancilla qubits to enable local connectivity and entangle distant logical qubits by measuring the ancilla qubits. This comes at the price of a $\mathcal{O}(d)$ time overhead and a $\mathcal{O}(d^2)$ or $\mathcal{O}(d)$ space – i.e. qubit – overhead, depending on the protocols, for a distance d quantum LDPC code. The lattice surgery approach is particularly favorable for superconducting platforms, because of the fast measurements and native local connectivity. It is however inconvenient for other platforms such as neutral atoms and trapped ions, where measurements are typically the slowest operations [18–20]. Alternative approaches for gate-based comput-

ing are instead based on code-switching from an LDPC code to lower-rate codes such as the surface code [6] or on performing global transversal operations on all logical qubits at the same time [21]. However, code-switching results in large qubit overheads – thus reducing the appeal of LDPC memories –, while restriction to global logical operations is incompatible with efficient compilation of quantum algorithms. For the latter, it would be highly desirable to invent protocols to perform logical operations on individual logical qubits within a given LDPC code or across different LDPC codes.

In this work, we propose a gate-based protocol to realize an overcomplete set of logical single- and multi-qubit Clifford gates on individual logical qubits within a LDPC code or across different LDPC codes. Our scheme employs transversal entangling gates between a subset of physical qubits of the quantum LDPC code and all physical qubits of an auxiliary smaller Bacon-Shor code [22], enabling parallel and fully addressable logical operations via teleportation. As an example, we present supporting numerical simulations to show the error correction performance of our Hadamard rotation protocol compared to a memory on the same code. As evidence of the versatility of our protocol, we also discuss how the logical Clifford gate set enabled by our gadget can be augmented with the Pauli-based computation framework by performing joint measurements of arbitrary logical operators of the targeted code. Finally, we briefly comment on how to integrate our protocol with magic state cultivation methods [23–26] to enable universal addressable quantum computation in the quantum LDPC code.

Our protocol requires a $\mathcal{O}(d^2)$ qubit overhead with only a $\mathcal{O}(1)$ time overhead [27] for a distance d quantum LDPC code, independent of whether the logical gate is single- or multi-qubit. Hence, it outperforms

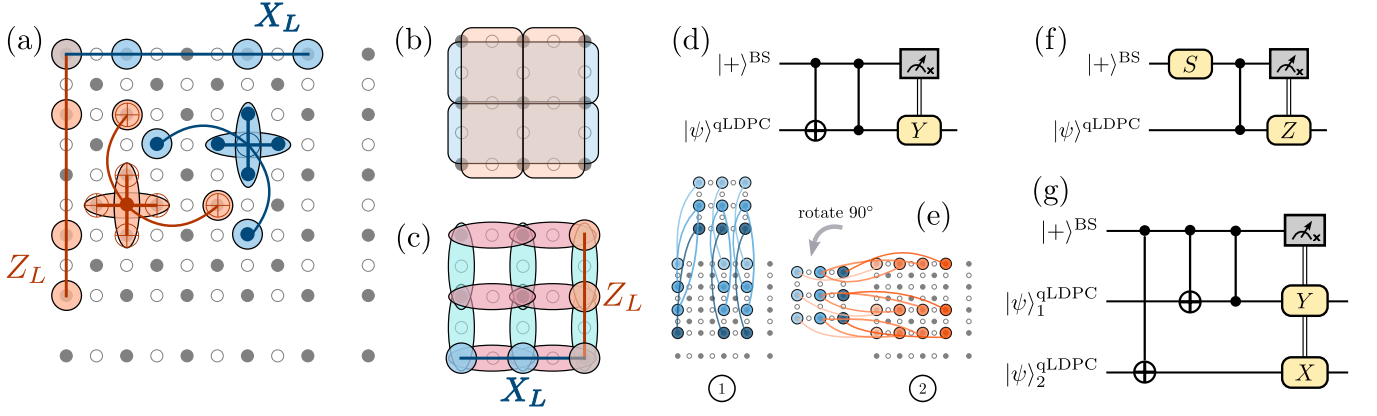


FIG. 1. (a) Example of La-cross code with next-to-nearest neighbor connectivity (labeled as $k = 2$). The code is defined by a $n \times n$ lattice with a $(n - k) \times (n - k)$ sublattice with open boundary conditions. A X - (blue) and a Z -stabilizer (red) are depicted, along with one pair of logical X_L (blue) and Z_L (red) operators. (b) Bacon-Shor code with non-local X - (blue) and Z -stabilizers (red) spanning along two consecutive rows and columns of the array, respectively. (c) X -type (cyan) and Z -type (pink) two-body gauge operators of the Bacon-Shor code and its two logical X_L (blue) and Z_L (red) operators. (d) Circuit implementing a logical $\mathcal{R}_y(\pi/2)$ rotation on a single quantum LDPC logical qubit via teleportation and (e) example of implementation via qubit shuttling. Different equivalent representatives of the same logical operator are colored with different gradients of the same color (blue for X_L and red for Z_L). (f) Circuit implementing a logical S gate via teleportation. (g) Circuit implementing a logical two-qubit entangling operation between two logical qubits via teleportation.

known surgery-based schemes in terms of time overhead, while offering a comparable or better spacetime overhead. Our protocol is well-suited for near-term implementation on platforms with non-local connectivity, such as neutral atoms [20] or trapped ions [28], using qubit shuttling [29] or native long-range Rydberg [7, 30, 31] or Coulomb interactions, respectively. However, the protocol is hardware-agnostic and can be applied to other platforms with sufficient long-range connectivity. In particular, superconducting qubits with non-local couplers [17, 32, 33] are also a promising candidate to perform, e.g., Pauli-based computation with our gadget.

RESULTS

La-cross codes and structure of logical operators

We consider the Calderbank-Shor-Steane (CSS) [34, 35] La-cross code family of high-rate quantum LDPC codes, which has recently been shown to outperform the surface code in experimentally relevant regimes of physical error probability with different noise models [7, 36]. This choice is motivated by two key reasons. First, La-cross codes have moderate long-range connectivity and open boundary conditions, and thus are appealing candidates for near-term implementations in several qubit platforms, such as neutral atoms and trapped ions, with a moderate number of qubit movements or static long-range interactions [7, 37, 38]. Second, they are hypergraph product codes [39, 40] with logical operators aligned along single rows or columns of the lattice, which

can be transformed and translated using intuitive geometrical arguments inherited from the surface code, as we explain below. However, this is not restrictive for our protocol to work, and we let as an open question whether the quantum computing gadget designed in this work can enable addressable quantum computation in other high- or constant-rate quantum LDPC code families.

The La-cross code family is defined as follows [7]: Let $\mathcal{C} = [n, k, d]$ be a cyclic classical code that encodes k logical bits in n physical bits with Hamming distance d . We denote by r the number of its checks and $H \in \mathbb{F}_2^{r \times n}$ its circulant rectangular parity-check matrix, so that $k = n - \text{rank}(H)$. Let $\mathcal{C}^T$ be the transposed code of $\mathcal{C}$ with parity-check matrix $H^T \in \mathbb{F}_2^{n \times r}$, then $\mathcal{C}^T \equiv [n^T, k^T, d^T]$ with $n^T = r$, $k^T = n^T - \text{rank}(H^T)$, and d^T the corresponding Hamming distance. Using the hypergraph product (HGP) [39], we can construct a $[[N, K, D]]$ La-cross code that has a quantum parity-check matrix

$$H_Q = \begin{pmatrix} 0 & H_X \\ H_Z & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & H \otimes \mathbb{I}_n & \mathbb{I}_r \otimes H^T \\ \mathbb{I}_n \otimes H & H^T \otimes \mathbb{I}_r & 0 & 0 \end{pmatrix}. \quad (1)$$

This construction ensures that all the stabilizers commute, i.e. $H_X H_Z^T = 0$. If the cyclic classical code $\mathcal{C}$ is full-rank, i.e. if $r = n - k$, the resulting La-cross code inherits open boundary conditions and its quantum parameters read $N = n^2 + (n - k)^2$, $K = k^2$, $D = d$ [7].

It is often convenient to describe classical cyclic codes of n bits in terms of polynomials, say $h(x)$, having binary coefficients and dividing $x^n + 1$ modulo 2, i.e. $h(x) \in \mathbb{F}_2[x]/(x^n + 1)$. Then, the classical cyclic

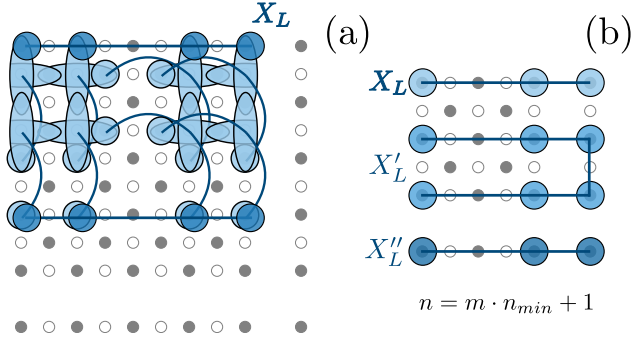


FIG. 2. (a) Logical operators in a La-cross code can be translated along the main lattice via multiplication by a subset of stabilizers. For $k = 2$, an example is illustrated: Dark blue strings are two equivalent representation of the same logical operator X_L , while in light blue we show the pattern of X -stabilizers that translates the logical operator in the first row three rows downward. (b) For $k = 2$ codes, for some lattice sizes (i.e., $n = m \cdot n_{min} + 1$ for some $m \in \mathbb{N}$) and open boundary conditions, some logical operators can be longer than the code distance d . Here we show the case of a $d = 2$ $k = 2$ La-cross code which also admits $d + 1 = 3$ -long logical operators. Crucially, it is always possible to find as many non-overlapping equivalent representatives (X_L, X'_L, X''_L logical strings colored with different gradients of blue) via multiplication by a suitable subset of stabilizers. Notably, some of these equivalent representations can have length $2d$ (e.g., X'_L in the example shown). This observation is key to enable the fault tolerance of our quantum computing gadget.

code $\mathcal{C} = [n, k, d]$ that generates a La-cross code must have a circulant rectangular parity-check matrix $H = \text{circ}(1, 1, 0, \dots, 0, 1, 0, \dots, 0) \in \mathbb{F}_2^{(n-k) \times n}$ with $H_{00} = H_{01} = H_{0k} = 1$, which corresponds to a generating polynomial $h(x) = 1 + x + x^k$. This classical code can be regarded as a higher-rate generalization of the classical repetition code, $h_{rep}(x) = 1 + x$. We also note that tensoring two copies of a classical code with weight-3 checks via HGP produces stabilizers of weight-6 in the resulting La-cross code [see Fig. 1(a)].

The logical operators of a $[[N, K, d]]$ QEC code are Pauli strings in the N -qubit Pauli group, $L \in \mathcal{P}_N$, which commute with all stabilizers without being a product of stabilizers. For CSS codes,

$$L = \begin{pmatrix} 0 & L_X \\ L_Z & 0 \end{pmatrix}. \quad (2)$$

It follows that $L_Z \in \ker(H_X)$, but $L_Z \notin \text{im}(H_Z^T)$, and $L_X \in \ker(H_Z)$, but $L_X \notin \text{im}(H_X^T)$. It is then easy to verify that the shortest logical operators of any La-cross code are horizontal and vertical Pauli strings that always align along a single row or column of the lattice [see Fig. 1(a) for an example] [7].

Consider a $[[N, K, d]]$ La-cross code with open boundaries and let X_L be one of its $K = k^2$ logical operators of X -type. To perform addressable logical operations on a specific logical qubit encoded in a La-cross code in a

fault-tolerant manner, our protocol requires finding the shortest d non-overlapping equivalent representatives of X_L , that is, the shortest X -strings that are equivalent to X_L up to multiplication by some X -stabilizers.

Without loss of generality, we assume that X_L has support over d physical qubits in the first row of the lattice. To translate a logical operator defined on the first row of the main lattice downward while preserving its length and pattern, we proceed as follows – the generalization to logical operators that do not have support on the first row is straightforward and follows from similar considerations. For the sake of clarity, we restrict the discussion to the case of $k = 2$ La-cross codes and refer to the Methods section for a general discussion. First, we identify the stabilizers that overlap on one qubit with the logical operator X_L . Then, we multiply X_L by the two subsequent rows of X -stabilizers with the pattern just identified [for an example of such stabilizer pattern, see the light blue operators in Fig. 2(a)]. This effectively translates the X_L logical operator three rows downward in the main lattice. From this new equivalent representation of X_L , we repeat the same process several times and stop when the boundary of the lattice is reached. In this way, $d' < d$ equivalent d -long representatives are found. Each of these strings, say $X_L, X'_L, X''_L, \dots$, is then multiplied by a single row of X -stabilizers which are just below each string and overlap on one qubit with it. Such a row of stabilizers always exists except if the logical Pauli string we are considering is close to the boundary of the lattice. In this way, we find the remaining $d - d'$ non-overlapping equivalent representatives of the original X_L logical operator. We observe that these last $d - d'$ representatives are $2d$ -long X -strings with the same pattern as the original X_L , but spanning across two rows of the main lattice [see, for example, Fig. 2(b), where different equivalent representations of the same logical operator are represented as strings colored with different gradients of blue].

For higher-rate, i.e. $k \geq 4$, La-cross codes and certain values of n , we find that some of the shortest, non-overlapping, and equivalent representatives of a logical operator can span across ℓ rows, where ℓ is upper bounded by $k - 1$. We comment more in details on those cases in the Methods section.

As an interesting observation, we find that, when open boundary conditions are enforced, the stabilizer pattern of La-cross codes breaks the standard notion of translational invariance across the lattice. As a consequence, a distance- d La-cross code has at least one pair of d -long (X_L, Z_L) logical operators, but it also admits longer logical operators. This can be easily understood as follows. Consider a $k = 2$ La-cross code, which has stabilizers of Z -type (the discussion is symmetric for X) acting on three qubits aligned in a row, say, $q_0 q_1 q_2$. Therefore, $\bar{X}_0 = X_0 X_1$, $\bar{X}_1 = X_0 X_2$, and $\bar{X}_2 = X_1 X_2 = \bar{X}_0 \bar{X}_1$ all commute with the stabilizers. Additionally, $k = 2$ La-cross codes enjoy translational invariance over $n_{min} = 3$

lattice sites. The logical operators can be easily identified by repeating the (not all linearly independent) generating patterns $\bar{X}_0, \bar{X}_1, \bar{X}_2$ along one row of the $n \times n$ principal lattice. When $n = m \cdot n_{\min} + 1$, for some $m \in \mathbb{N}$, half of the four logical operators will then have length d and half will have length $d + 1$. Nevertheless, it is always possible to find as many non-overlapping equivalent representatives as the length of the longest logical operator with the iterative procedures explained above [see Fig. 2(b)]. Crucially, this can be leveraged to engineer a fault-tolerant protocol that preserves the circuit-level distance of each distinct logical operator, even of those that are longer than the actual distance of the code. Preserving the circuit-level distance of each individual logical operator – especially those that are longer than d – is highly relevant, e.g., for the compilation of algorithms that only use a subset of the logical qubits encoded in a quantum LDPC code. This would allow one to select only the longest logical operators and hence improve robustness against errors.

Finally, we observe that similar considerations can be generalized to higher-rate, i.e. $k > 2$, La-cross codes with open boundaries that break translational invariance on a larger scale. For example, by inspection, we find that $n_{\min} = 7$ for $k = 3$ and $n_{\min} = 15$ for $k = 4$. It follows that these codes more often admit logical operators which are longer than d physical qubits.

Auxiliary Bacon-Shor code

The quantum computing gadget that we design to enable addressable quantum computation with La-cross quantum LDPC codes leverages gate teleportation via an auxiliary Bacon-Shor code, which we here briefly review.

The Bacon-Shor (BS) code [22] $[[N_{BS}, 1, d_{BS}]]$ is a subsystem stabilizer code that encodes one logical qubit into a $n \times m$ lattice of physical qubits. The number of independent stabilizers is strictly smaller than $N_{BS} - 1$, hence certain degrees of freedom do not store logical information and can be freely modified without affecting the encoded state. Let $X_{i,j}^{\text{BS}}$ ($Z_{i,j}^{\text{BS}}$) be a X (Z) operator acting on the qubit in the i th row and j th column of the BS lattice. The BS code is defined by $n + m - 2$ stabilizers

$$S_{z,j}^{\text{BS}} = \prod_{i=1}^n Z_{i,j}^{\text{BS}} Z_{i,j+1}^{\text{BS}}, \quad S_{x,i}^{\text{BS}} = \prod_{j=1}^m X_{i,j}^{\text{BS}} X_{i+1,j}^{\text{BS}} \quad (3)$$

which span two entire rows or columns and are then highly non-local [see Fig. 1(b)]. The remaining degrees of freedom can be used to define local gauge operators [see Fig. 1(c)],

$$G_{z,i,j}^{\text{BS}} = Z_{i,j}^{\text{BS}} Z_{i,j+1}^{\text{BS}}, \quad G_{x,i,j}^{\text{BS}} = X_{i,j}^{\text{BS}} X_{i+1,j}^{\text{BS}}, \quad (4)$$

such that the original non-local stabilizers can be recov-

ered as the product of weight-2 gauge operators,

$$S_{z,i}^{\text{BS}} = \prod_{j=1}^m G_{z,i,j}^{\text{BS}}, \quad S_{x,j}^{\text{BS}} = \prod_{i=1}^n G_{x,i,j}^{\text{BS}}. \quad (5)$$

Thus, the BS code has $n(n-1) + m(m-1)$ non-mutually commuting gauge operators. A gauge is fixed by promoting a maximal commuting subset of gauge operators to stabilizers. For example, in the so-called *Z-gauge* all Z -type gauge operators are chosen as Z -stabilizers, $Z S_{z,i,j}^{\text{BS}} = G_{z,i,j}^{\text{BS}}$, while in the *X-gauge* all the X -type gauge operators are chosen as X -stabilizers, $X S_{x,i,j}^{\text{BS}} = G_{x,i,j}^{\text{BS}}$. Additionally, we observe that the X_L (Z_L) logical operator of the BS code is an X string acting on a single row (column) of the lattice [see Fig. 1(c)].

Finally, we note that there exists another possible gauge fixing choice for the BS code, which we call the *surface code gauge*, that effectively morphs the BS code into a rotated surface code [41]. This is convenient as it allows one to use the full surface code toolbox and then morph the code back to a BS code. In the following, we will use this possibility to implement a S gate directly in the BS code without extra qubit overhead.

Addressable Hadamard rotation

We start by presenting the protocol to perform a Hadamard (H) gate, i.e., a logical single-qubit rotation $\mathcal{R}_y(\pi/2)$, on a selected logical qubit encoded in the La-cross code [Fig. 1(d)].

We consider a d -long X_L logical operator of the La-cross code with support on the first row of the lattice, together with its d non-overlapping, shortest, and equivalent representatives, as discussed before. The collection of these equivalent logical operators effectively realizes a partition of the La-cross code lattice. In this way, the only physical qubits involved in the operations are those that uniquely correspond to the single La-cross logical qubit we aim to address.

The protocol for performing a Hadamard rotation is as follows. (i) We first consider a $d \times d$ square BS code and initialize it in its $|+\rangle^{\text{BS}}$ logical state. For the state preparation and all the controlled operations to be fault-tolerant, we use the Z -gauge for the BS code. Next, (ii) we apply a logical transversal CNOT (tCNOT) gate controlled by the rows of the BS code and targeting all the equivalent representatives of the selected X_L logical operator of the La-cross code, that is

$$|+\rangle^{\text{BS}} |\psi\rangle^{\text{qLDPC}} \xrightarrow{\text{tCNOT}} \frac{1}{\sqrt{2}} \left(|0\rangle^{\text{BS}} + |1\rangle^{\text{BS}} X \right) |\psi\rangle^{\text{qLDPC}}.$$

Then, (iii) we apply a logical transversal CZ (tCZ) gate controlled by the rows of the BS code and that targets all the equivalent representatives of the selected Z_L logical

operator of the La-cross code (that is, the Z_L logical operators corresponding to the initial X_L logical operator),

$$\begin{aligned} & \xrightarrow{\text{tCZ}} \frac{1}{\sqrt{2}} \left(|0\rangle^{\text{BS}} + |1\rangle^{\text{BS}} Z.X \right) |\psi\rangle^{\text{qLDPC}} \\ &= \frac{|+\rangle^{\text{BS}}}{\sqrt{2}} \frac{1+iY}{\sqrt{2}} |\psi\rangle^{\text{qLDPC}} + \frac{|-\rangle^{\text{BS}}}{\sqrt{2}} \frac{1-iY}{\sqrt{2}} |\psi\rangle^{\text{qLDPC}}. \end{aligned}$$

We show in Fig. 1(e) a possible implementation scheme for these two transversal controlled operations by using either qubit shuttling or long range couplers. We also note that a round of QEC stabilizer readout has to be inserted on both codes between the two controlled logical operations to ensure fault tolerance by preventing errors from propagating to multiple physical qubits without being detected. Finally, (iv) we measure the BS code in the X -basis. Thus, if the BS code is measured in the $|+\rangle^{\text{BS}}$ state, the protocol correctly implements a $\mathcal{R}_y(\pi/2)$ rotation of the selected La-cross logical qubit. Instead, if the BS code is in state $|-\rangle^{\text{BS}}$ before the measurement, a conditional logical Y correction must be applied to the targeted La-cross logical qubit. Notably, such a correction can be tracked classically throughout the computation when the remaining circuit consists only of Clifford gates, and when non-Clifford states are prepared offline and injected via teleportation protocols.

We note that our protocol only requires measuring the Z -stabilizers of the BS code and never the X -stabilizers, which we then only measure during the final X -measurement. This is due to the fact that Z -errors in the BS code can never spread to the targeted logical La-cross qubit via the two controlled operations, which reduces the gate cost of the protocol. We also observe that the same protocol can be used to implement the same gate with opposite angle, that is, a $R_y(-\pi/2)$ rotation, either by reversing the order of the controlled operations or by classically controlling the Y -correction with the $|+\rangle^{\text{BS}}$ logical state of the BS code.

In the next section, we discuss the fault tolerance of our Hadamard rotation protocol and show QEC numerical simulations to assess its logical error-correction performance.

On the fault tolerance of the BS gadget

We now prove that the transversal controlled operations between the BS code and the selected qubit partition of the La-cross code correctly implement the desired operations in a fault-tolerant manner.

We first observe that fault tolerance straightforwardly derives from the transversality of the protocol, which individually couples each of the d rows of the BS code with each of the d different representatives of the selected logical operator in the La-cross code [see also Fig. 1(e)]. Crucially, the identified equivalent representatives of the

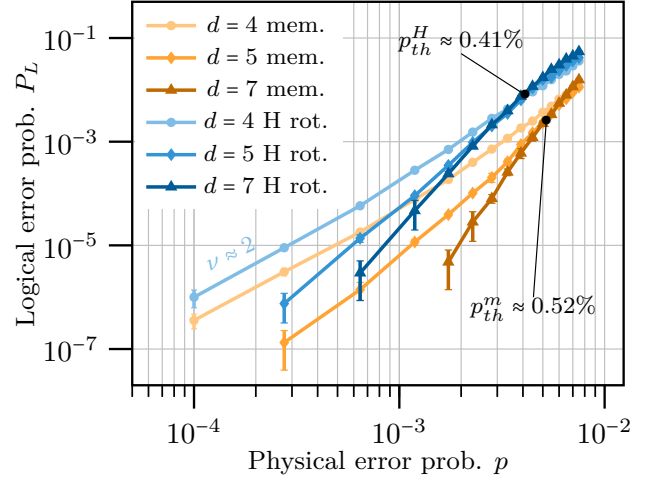


FIG. 3. Quantum error correction simulations for performing a logical Hadamard (H) rotation on a single logical qubit encoded in a $k = 2$ La-cross code. The performance for the Hadamard rotation (blue) is compared against the performance for a memory experiment (orange), showing a moderate increase of the logical error probability and decrease of the physical error threshold. All the results are for a single La-cross logical qubit and for d rounds of error correction. The logical error probability is normalized by the number of rounds and the error bars are standard deviations from the Monte Carlo simulations. The La-cross codes that we have simulated are $[[52, 4, 4]]$, $[[100, 4, 5]]$, and $[[202, 4, 7]]$. BP+OSD decoder was used (see Methods for details).

same logical operator must have non-overlapping support for the protocol to be fault-tolerant. We also observe that the longer, namely $2d$ -long (or, generally, ℓd -long), representatives can be fault-tolerantly controlled with a d -long row of the BS code provided that the operation is realized in two (ℓ) steps, one for each of the two (ℓ) d -long rows of the logical representative. That is because an X error on one of the control qubits of the BS code can only propagate through the CNOT gates into vertical pairs (ℓ -tuple) of X errors on the La-cross code. Since the logical X_L operators are all horizontal in the La-cross code, these vertical X errors do not align with the logical operator, and hence do not degrade the circuit-level distance of the code, thus ensuring fault tolerance. The reasoning is similar for CZ gates.

To support our argument, we show in Fig. 3 the QEC simulations for the logical error probability of our Hadamard rotation protocol (blue) compared to a memory experiment (orange) for the $k = 2$ La-cross code family. The three code instances shown are $[[52, 4, 4]]$, $[[100, 4, 5]]$, and $[[202, 4, 7]]$.

We use `Stim` [42] to perform the QEC numerical simulations. For both logical rotation and memory, only one logical operator is considered. Motivated by experiments (see Methods), we choose a noise model that consists of two-qubit depolarizing errors for two-qubit gates

and phase-flip errors for measurements and resets in the X -basis with equal error strength p . Single-qubit and idle-errors are neglected. For the Hadamard rotation, we employ a correlated decoding strategy [27, 43, 44] that allows us to insert a single round of error correction after each logical gate on both codes. More specifically, we initialize the LDPC code in the $|0\rangle_L$ by performing one round of error correction, apply another round between the two logical controlled gates and append the remaining $d - 2$ rounds at the end for comparison with the memory experiments, where we perform d rounds of QEC. Data are normalized by the number of rounds as $P_L = 1 - (1 - p_L)^{1/d}$, where p_L is the ratio between the total number of errors and the number of shots. The error bars are standard deviations associated with the Monte Carlo simulations. The syndrome information is decoded using Belief Propagation with Ordered Statistics Decoder (BP+OSD) [45, 46]. More details on circuit implementation and decoding are given in the Methods section.

Our numerical results show a small decrease in the physical error threshold – approximately from $p_{th}^m \approx 0.52\%$ to $p_{th}^H \approx 0.41\%$ – and an increase in the logical error probability for the rotation protocol compared to the memory, due to the larger number of operations and hence error locations. Crucially, the scaling of the logical error probabilities for the rotation protocol compared to the memory shows that our scheme is fault-tolerant – i.e., the corresponding curves for the rotation and memory are parallel and with the expected scaling for fault tolerance. For example, for the $d = 4$ La-cross code shown in Fig. 3, we find that the logical error probability correctly scales as $P_L \propto p^\nu$, with $\nu = \lceil d/2 \rceil \approx 2$ in the deep sub-threshold regime.

In addition to the above proof of fault tolerance of the protocol, we can also prove that the protocol implements the desired controlled operation, for example, a tCNOT (the argument is symmetric for tCZ), and that the stabilizers of both BS and La-cross codes commute through the tCNOT up to multiplications by other stabilizers. That is, formally, $\text{tCNOT} \cdot S_i \cdot \text{tCNOT} = S_i \prod_{j \neq i} S_j$, $\forall i$, where S_i is a stabilizer of either code. This is done as follows: The logical state $|0\rangle^{\text{BS}}$ of the BS code is a superposition of all states with an even number of rows in the state $|\bar{1}\rangle = |1 \cdots 1\rangle$. In contrast, its logical state $|1\rangle^{\text{BS}}$ is a superposition of all states with an odd number of such rows. Therefore, when the BS code is in $|0\rangle^{\text{BS}}$ ($|1\rangle^{\text{BS}}$), the X logical operator of the targeted La-cross logical qubit is applied an even (odd) number of times, thereby acting as the identity (single Pauli X) on the targeted qubit. This correctly implements a logical tCNOT on the selected La-cross logical qubit controlled by the BS code. Similar considerations apply to the tCZ gate.

Let us now consider how the stabilizers of each code propagate through the tCNOT. First, we observe that, due to transversality, the Z -stabilizers of the BS code and

the X -stabilizers of the La-cross code commute through the tCNOT, and are therefore left unchanged. A Z -stabilizer of the La-cross code propagates through the tCNOT into a product of Z Pauli operators on the BS code. Furthermore, since the X_L logical operator and the Z -stabilizers of the La-cross code must commute, they always overlap on an even number of qubits in the main lattice. Consequently, after conjugation by tCNOT, the Z -stabilizers of the La-cross code transform as

$$\text{tCNOT} \cdot S_{z,i}^{\text{qLDPC}} \cdot \text{tCNOT} = S_{z,i}^{\text{qLDPC}} \cdot Z_{k_i,j_1}^{\text{BS}} \cdot Z_{k_i,j_2}^{\text{BS}} \quad (6)$$

where $S_{z,i}^{\text{qLDPC}}$ denotes the i -th Z -stabilizer of the La-cross code and $Z_{k_i,j}^{\text{BS}}$ is the Pauli Z operator acting on the data qubit located in the k_i -th row and j -th column of the BS code. We consider the Z -gauge of the BS code, where we promote to stabilizers the two-body gauge operators $Z_{z,i,j}^{\text{BS}} = Z_{i,j}^{\text{BS}} \cdot Z_{i,j+1}^{\text{BS}}$. Therefore, the product $Z_{k_i,j_1}^{\text{BS}} \cdot Z_{k_i,j_2}^{\text{BS}}$ in Eq. (6) can be expressed as a product of BS Z -stabilizers as

$$Z_{k_i,j_1}^{\text{BS}} \cdot Z_{k_i,j_2}^{\text{BS}} = \prod_{j'=j_1}^{j_2-1} Z_{z,i,j'}^{\text{BS}}.$$

On the other hand, since the X -stabilizers of the BS code act on two consecutive rows of data qubits, under conjugation by tCNOT they acquire support on two consecutive, non-overlapping representatives of the same X_L logical operator of the La-cross code. Since different representatives differ by a product of X -type stabilizers of the La-cross code, it follows that

$$\text{tCNOT} \cdot S_{x,i}^{\text{BS}} \cdot \text{tCNOT} = S_{x,i}^{\text{BS}} \cdot \prod_j S_{x,j}^{\text{qLDPC}},$$

where $S_{x,i}^{\text{BS}}$ denotes the i -th X -type stabilizer of the BS code and $S_{x,j}^{\text{qLDPC}}$ are X -stabilizers of the La-cross code. This shows that our protocol implements the correct logical operation, while preserving the codespaces of both BS and La-cross codes.

Finally, we note that the BS gadget implements the desired operations on a single La-cross logical qubit, while leaving the other logical qubits unaffected. That is, similarly to the logical Pauli operators, which only affect their corresponding logical qubit, their controlled version used in the BS gadget does not affect the logical state of the surrounding logical qubits.

Addressable single qubit rotations

Similarly to the Hadamard gate, also the S gate, i.e. a logical single-qubit rotation $\mathcal{R}_z(\pi/2)$, can be implemented on any single La-cross logical qubit via gate teleportation.

We start by initializing the $d \times d$ square BS code in state $|S\rangle \equiv |+\rangle^{\text{BS}}$. Then, a logical tCZ gate between

the BS code and the targeted La-cross logical qubit is applied, and finally the BS code is measured in the X -basis. A Z -gate correction is applied, depending on the outcome of the X -measurement. Therefore, this protocol [the circuit is depicted in Fig. 1(f)] correctly implements an S gate on the targeted La-cross logical qubit. We also observe that a single-qubit rotation $\mathcal{R}_x(\pi/2)$ can be implemented in a very similar manner, using a logical tCNOT gate instead of the tCZ in the circuit described above.

The protocol we have just described exploits the fact that an $|S\rangle$ state can be fault-tolerantly prepared in the BS code by initializing it in the $|+\rangle_L$ of the surface code gauge [41], where fault-tolerant methods for implementing an S gate are known. In particular, a fold-transversal S gate can be fault-tolerantly implemented halfway through the syndrome extraction circuit leveraging the equivalence between rotated and unrotated surface codes at half-cycle [47]. After performing this operation, we morph the BS code back to the Z -gauge in order to fault-tolerantly apply the subsequent logical controlled operation.

Addressable two- and multi-qubit gates

The protocol for implementing logical multi-qubit entangling gates is a generalization of the scheme discussed above for the Hadamard rotation. Notably, the targeted logical qubits can now belong to either the same La-cross code or to two distinct codes.

We start by presenting how to realize a logical two-qubit entangling gate. The scheme is as follows: The $d \times d$ BS code is first initialized to its logical $|+\rangle^{\text{BS}}$ state and then entangled via two tCNOT gates to each La-cross logical qubit in the same way as in the previous sections. Then, a tCZ gate is applied between the BS code and the first La-cross logical qubit. Finally, the BS is measured in the X -basis. Conditionally on the BS measurement outcome, a $Y_1 X_2$ correction is applied on the two targeted logical qubits. This protocol then correctly realizes an entangling logical $e^{i\frac{\pi}{4} Y_1 X_2}$ gate [see also Fig. 1(g)].

In the same spirit, we observe that, if a tCZ is applied in place of the first tCNOT, a $e^{i\frac{\pi}{4} Y_1 Z_2}$ gate is realized. Additionally, using the S -gate protocol described above, our scheme allows one to perform any two-qubit gate of the form $e^{i\frac{\pi}{4} X_1 X_2}$, $e^{i\frac{\pi}{4} Z_1 Z_2}$, and $e^{i\frac{\pi}{4} X_1 Z_2}$. This can then be straightforwardly extended to more than two logical qubits, enabling the native implementation of any multi-qubit logical gate of the form $e^{i\frac{\pi}{4} P}$, where P is a Pauli string containing either an odd number of Pauli- Y operators or none, and any number of Pauli- X and Pauli- Z operators.

All these observations suggest that, since the Clifford gate set enabled by our BS gadget is not minimal, ef-

ficient circuit-recompilation strategies may exploit this richer native gate set to reduce the overall gate count for quantum algorithms applications.

We also note that, in contrast to the measurement-based approach [6, 9, 10, 12–15], which requires $\mathcal{O}(d)$ rounds of QEC to perform a logical operation, our scheme operates in a time that is independent of the code distance. In fact, the total duration of a logical gate of the form $e^{i\frac{\pi}{4} P}$ depends solely on the number of controlled operations needed. In turn, the number of necessary controlled operations corresponds to the number of non-trivial Pauli operators in the Pauli string P defined above, and counting each Y operator twice, since it decomposes into X and Z . Moreover, several optimizations can reduce this number by merging multiple controlled logical operations into a single one. For example, (i) if two logical operators X_{L_1}, X_{L_2} act on the same row, the corresponding controlled operation can be applied directly to $X_{L_1} \cdot X_{L_2}$, that is, to the qubits in the support of the product of both operators. This is possible if the product of two d -long logical operators is still a d -long logical Pauli string, which is exactly the case for those logical operators that are as long as the distance d of the code. Alternatively, (ii) for small- k La-cross codes, if two logical operators act on different rows but have the same pattern, it is possible in some cases to find a compact (i.e., single-row) representation of their product on another row. The protocol can then be applied to this compact representation, and the physical controlled operation performed on the different representatives — where the two original rows jointly form a $2d$ -long representative of the compact operator. Similar considerations apply to Z logical operators.

Finally, as an example of use-case, we observe that our BS gadget can be used to implement a Greenberger–Horne–Zeilinger (GHZ) state of all the logical qubits encoded in a single La-cross code. This can be achieved by preparing one logical qubit in state $|+\rangle^{\text{qLDPC}}$ and the others in state $|0\rangle^{\text{qLDPC}}$, and then performing a cascade of CNOT gates — which are available up to single-qubit logical rotation with our protocol — between any pair of logical qubits.

Alternative approach: Pauli-based quantum computation

So far, we have discussed unitary Clifford gates for digital gate-based quantum computing. However, one can adopt a different paradigm for quantum computation based on measurements of joint logical operators, which enables the implementation of any Clifford circuit. This approach, known as Pauli-based computation (PBC) [48], underlies many lattice-surgery techniques for the surface code [8] and for quantum LDPC codes [6, 9, 10, 12–15]. This mechanism enables the

generation of entanglement across arbitrary subsets of qubits. Together with the standard single-qubit Clifford gates, any of these entangling operations suffices to generate the full Clifford group.

This additional framework can be used to augment the native Clifford gate set enabled by our protocol. Using our BS gadget, it is possible to measure the product of any joint logical operators composed of Pauli- X , and Pauli- Z operators. This is achieved by initializing the BS code in the logical state $|+\rangle^{\text{BS}}$ and applying a logical CNOT (resp. CZ) gate to a target logical qubit whenever a X_L (resp. Z_L) logical operator appears in the joint product. By subsequently measuring the BS code in the X -basis, one effectively implements a projective measurement $\frac{1 \pm P}{2}$, where P denotes the product of the logical operators. The sign of the projector is intrinsically random and a corresponding Pauli correction may be required; however, such corrections can be tracked and handled in software. Additionally, we observe that performing measurement-based quantum computation with our gadget uses few transversal gates – namely, only one for each pair of logical operators – and does not require preparing a $|S\rangle$ state in the BS code, compared to the previous gate-based protocols.

Similarly to the gate-based paradigm discussed in the previous section, our BS gadget also allows one to generate a logical GHZ state (up to Pauli corrections) between all the logical qubits encoded in a single La-cross code by first initializing all the logical qubits in $|+\rangle^{\text{qLDPC}}$ and then measuring the product $Z_i^L Z_{i+1}^L$ for each pair of consecutive qubits. This would require $K - 1$ BS gadgets, K being the number of logical qubits of the La-cross code.

Towards addressable non-Clifford operations

Finally, we comment on how one could complement our BS gadget with non-Clifford resources to enable universal gate-based quantum computation.

Similarly to Clifford resources, non-Clifford operations of the form $e^{i\frac{\pi}{8}P}$, where P is the same Pauli string defined above, can be prepared in the BS code and then teleported into the quantum LDPC code. In close analogy with the S -gate scheme, one could fold-transversally cultivate the $|T\rangle$ state in the surface code [26] and then morph it into the BS code by measuring its Z -gauge stabilizers. As an alternative approach, we could directly cultivate the T state in the folded BS code, which we let as an open future perspective.

DISCUSSION

In conclusion, we have demonstrated a novel gate-based scheme for performing addressable quantum computation with high-rate quantum LDPC codes with-

out relying on lattice-surgery-based techniques. We have shown how Clifford and non-Clifford resources can be fault-tolerantly prepared in an auxiliary Bacon-Shor code, also using the possibility of morphing such code into a rotated surface code, where $|S\rangle$ states can be efficiently prepared and $|T\rangle$ states cultivated in a fold-transversal manner [26, 47]. These universal resources can then be injected into a single logical qubit encoded in the quantum LDPC code via teleportation. Logical addressability is enabled by a careful qubit partition of the quantum LDPC code lattice consisting of the shortest d non-overlapping equivalent representatives of the same d -long logical operator. Crucially, our protocol relies only on transversal controlled operations between the BS and the quantum LDPC code, which can be realized in constant time. Therefore, our protocol outperforms all known lattice surgery schemes in terms of time overhead, while offering a comparable or better $\mathcal{O}(d^2)$ spacetime overhead overall. Additionally, since we do not require measuring all the stabilizers of the auxiliary Bacon-Shor code at each QEC round, the total gate count of our scheme is also substantially reduced.

The protocol we have presented in this work is hardware-agnostic, and hence suitable for implementation in several qubit platforms. Rydberg atom and trapped-ion platforms are promising for near-term experimental realizations as they display native long-range couplings and flexible connectivity due to atom shuttling [49] and long-range Rydberg and Coulomb interactions [7, 31], respectively. In addition, atomic qubits typically suffer from slow measurements. Therefore, fast schemes that do not rely on lattice surgery are particularly useful to enable error-corrected quantum computation.

In this work, we focus on La-cross quantum LDPC codes because their simple code structure allows us to identify the translation rules among individual logical operators. This, in turn, enables the development of a gate-based, fault-tolerant protocol that can operate on individual logical qubits in an addressable manner. It is an interesting open question whether our protocol can be generalized to other high-rate or constant-rate quantum LDPC code families, such as bivariate bicycle codes [5], or lifted product codes [6, 50, 51], which share similar stabilizer and logical operator structure.

Another important open direction is the development of fast decoding algorithms for quantum LDPC codes. In this work, we employed the BP+OSD decoder, which currently represents the state of the art in quantum LDPC decoding. However, despite its high accuracy, its significant time overhead renders it impractical for deep quantum computations. Our BS gadget relies on gate teleportation, and while this poses no issue for preparing Clifford resources, implementing non-Clifford gates requires real-time decoding to avoid the accumulation of unprocessed error syndromes. This can result in a backlog that would quickly degrade the protocol's performance.

Therefore, the development of fast and efficient decoding schemes remains a critical challenge for achieving universal quantum computation with quantum LDPC codes on near-term quantum processors.

ACKNOWLEDGEMENTS

We gratefully acknowledge discussions with Gavin K. Brennen. This research has received funding from the European Union’s Horizon 2020 research and innovation programme under the Horizon Europe programme HORIZON-CL4-2021-DIGITAL-EMERGING-01-30 via the project 101070144 (EuRyQa) and under the Marie Skłodowska-Curie grant agreement number 101120240 (MLQ) and from the French National Research Agency under the Investments of the Future Program projects ANR-21-ESRE-0032 (aQCess), ANR-22-CE47-0013-02 (CLIMAQS), ANR-17-EURE-0024 (QMat), and ANR-22-CMAS-0001 France 2030 (QuantEdu-France).

During the final stages of this work, a study appeared [52], which develops a different approach to perform high-rate addressable Clifford gates in CSS quantum LDPC codes.

* These authors contributed equally to this work.

† pupillo@unistra.fr

- [1] D. Gottesman, Stabilizer codes and quantum error correction, Caltech Ph.D. Thesis (1997), arXiv:quant-ph/9705052.
- [2] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, *PRX Quantum* **2**, 040101 (2021).
- [3] A. Kitaev, Fault-tolerant quantum computation by anyons, *Annals of Physics* **303**, 2–30 (2003).
- [4] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *Journal of Mathematical Physics* **43**, 4452–4505 (2002).
- [5] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, *Nature* **627**, 778 (2024).
- [6] Q. Xu, J. P. B. Ataiades, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasic, M. D. Lukin, L. Jiang, and H. Zhou, Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays, *Nature Physics* **20**, 1084 (2024).
- [7] L. Pecorari, S. Jandura, G. K. Brennen, and G. Pupillo, High-rate quantum ldpc codes for long-range-connected neutral atom registers, *Nature Communications* **16**, 1111 (2025).
- [8] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, *New Journal of Physics* **14**, 123011 (2012).
- [9] L. Z. Cohen, I. H. Kim, S. D. Bartlett, and B. J. Brown, Low-overhead fault-tolerant quantum computing using long-range connectivity, *Science Advances* **8**, eabn1717 (2022), <https://www.science.org/doi/pdf/10.1126/sciadv.abn1717>.
- [10] A. Cross, Z. He, P. Rall, and T. Yoder, Improved qldpc surgery: Logical measurements and bridging codes (2024), arXiv:2407.18393 [quant-ph].
- [11] A. Cowtan, Ssip: automated surgery with quantum ldpc codes (2024), arXiv:2407.09423 [quant-ph].
- [12] D. J. Williamson and T. J. Yoder, Low-overhead fault-tolerant quantum computation by gauging logical operators (2024), arXiv:2410.02213 [quant-ph].
- [13] A. Cowtan, Z. He, D. J. Williamson, and T. J. Yoder, Parallel logical measurements via quantum code surgery (2025), arXiv:2503.05003 [quant-ph].
- [14] B. Ide, M. G. Gowda, P. J. Nadkarni, and G. Dauphinais, Fault-tolerant logical measurements via homological measurement, *Phys. Rev. X* **15**, 021088 (2025).
- [15] E. Swaroop, T. Jochym-O’Connor, and T. J. Yoder, Universal adapters between quantum ldpc codes (2025), arXiv:2410.03628 [quant-ph].
- [16] G. Zhang and Y. Li, Time-efficient logical operations on quantum low-density parity check codes, *Physical Review Letters* **134**, 10.1103/physrevlett.134.070602 (2025).
- [17] T. J. Yoder, E. Schoute, P. Rall, E. Pritchett, J. M. Gambetta, A. W. Cross, M. Carroll, and M. E. Beverland, Tour de gross: A modular quantum computer based on bivariate bicycle codes (2025), arXiv:2506.03094 [quant-ph].
- [18] G. A. Paz-Silva, G. K. Brennen, and J. Twamley, Fault tolerance with noisy and slow measurements and preparation, *Phys. Rev. Lett.* **105**, 100501 (2010).
- [19] F. Butt, D. F. Locher, K. Brechtelsbauer, H. P. Büchler, and M. Müller, Measurement-free, scalable, and fault-tolerant universal quantum computing, *Science Advances* **11**, eadv2590 (2025).
- [20] D. Bluvstein, A. A. Geim, S. H. Li, S. J. Evered, J. P. B. Ataiades, G. Baranes, A. Gu, T. Manovitz, M. Xu, M. Kalinowski, S. Majidy, C. Kokail, N. Maskara, E. C. Trapp, L. M. Stewart, S. Hollerith, H. Zhou, M. J. Gullans, S. F. Yelin, M. Greiner, V. Vuletic, M. Cain, and M. D. Lukin, Architectural mechanisms of a universal fault-tolerant quantum computer (2025), arXiv:2506.20661 [quant-ph].
- [21] Q. Xu, H. Zhou, G. Zheng, D. Bluvstein, J. P. B. Ataiades, M. D. Lukin, and L. Jiang, Fast and parallelizable logical computation with homological product codes, *Phys. Rev. X* **15**, 021065 (2025).
- [22] D. Bacon, Operator quantum error-correcting subsystems for self-correcting quantum memories, *Phys. Rev. A* **73**, 012340 (2006).
- [23] C. Gidney, N. Shutty, and C. Jones, Magic state cultivation: growing t states as cheap as cnot gates (2024), arXiv:2409.17595 [quant-ph].
- [24] Y. Vakinin, S. Jacoby, A. Grimsmo, and A. Retzker, Efficient magic state cultivation on the surface code (2025), arXiv:2502.01743 [quant-ph].
- [25] J. Claes, Cultivating t states on the surface code with only two-qubit gates (2025), arXiv:2509.05232 [quant-ph].
- [26] K. Sahay, P.-K. Tsai, K. Chang, Q. Su, T. B. Smith, S. Singh, and S. Puri, Fold-transversal surface code cultivation (2025), arXiv:2509.05212 [quant-ph].
- [27] H. Zhou, C. Zhao, M. Cain, D. Bluvstein, N. Maskara, C. Duckering, H.-Y. Hu, S.-T. Wang, A. Kubica, and M. D. Lukin, Low-overhead transversal fault tolerance for

- universal quantum computation, *Nature* **646**, 303–308 (2025).
- [28] D. Kielpinski, C. Monroe, and D. J. Wineland, Architecture for a large-scale ion-trap quantum computer, *Nature* **417**, 709 (2002).
- [29] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, V. Vuletić, and M. D. Lukin, A quantum processor based on coherent transport of entangled atom arrays, *Nature* **604**, 451–456 (2022).
- [30] C. Poole, T. M. Graham, M. A. Perlin, M. Otten, and M. Saffman, Architecture for fast implementation of quantum low-density parity-check codes with optimized rydberg gates, *Phys. Rev. A* **111**, 022433 (2025).
- [31] M. Saffman, Quantum computing with atomic qubit arrays: confronting the cost of connectivity (2025), arXiv:2505.11218 [quant-ph].
- [32] M. A. Tremblay, N. Delfosse, and M. E. Beverland, Constant-overhead quantum error correction with thin planar connectivity, *Phys. Rev. Lett.* **129**, 050504 (2022).
- [33] N. Berthussen, D. Devulapalli, E. Schoute, A. M. Childs, M. J. Gullans, A. V. Gorshkov, and D. Gottesman, Toward a 2d local implementation of quantum low-density parity-check codes, *PRX Quantum* **6**, 010306 (2025).
- [34] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, *Physical Review A* **54**, 1098–1105 (1996).
- [35] A. Steane, Multiple-particle interference and quantum error correction, *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **452**, 2551–2577 (1996).
- [36] L. Pecorari and G. Pupillo, Quantum ldpc codes for erasure-biased atomic quantum processors (2025), arXiv:2502.20189 [quant-ph].
- [37] Y. Hong, M. Marinelli, A. M. Kaufman, and A. Lucas, Long-range-enhanced surface codes, *Phys. Rev. A* **110**, 022607 (2024).
- [38] Y. Hong, E. Durso-Sabina, D. Hayes, and A. Lucas, Entangling four logical qubits beyond break-even in a non-local code, *Phys. Rev. Lett.* **133**, 180601 (2024).
- [39] J.-P. Tillich and G. Zemor, Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength, *IEEE Transactions on Information Theory* **60**, 1193–1202 (2014).
- [40] A. A. Kovalev and L. P. Pryadko, Quantum kronecker sum-product low-density parity-check codes with finite rate, *Physical Review A* **88**, 10.1103/physreva.88.012311 (2013).
- [41] M. Li, D. Miller, and K. R. Brown, Direct measurement of bacon-shor code stabilizers, *Physical Review A* **98** (2018).
- [42] C. Gidney, Stim: a fast stabilizer circuit simulator, *Quantum* **5**, 497 (2021).
- [43] M. Cain, C. Zhao, H. Zhou, N. Meister, J. P. B. Ataiades, A. Jaffe, D. Bluvstein, and M. D. Lukin, Correlated decoding of logical algorithms with transversal gates (2025), arXiv:2403.03272 [quant-ph].
- [44] M. Serra-Peralta, M. H. Shaw, and B. M. Terhal, Decoding across transversal clifford gates in the surface code (2025), arXiv:2505.13599 [quant-ph].
- [45] J. Roffe, D. R. White, S. Burton, and E. Campbell, Decoding across the quantum low-density parity-check code landscape, *Phys. Rev. Res.* **2**, 043423 (2020).
- [46] J. Roffe, LDPC: Python tools for low density parity check codes (2022).
- [47] Z.-H. Chen, M.-C. Chen, C.-Y. Lu, and J.-W. Pan, Transversal logical clifford gates on rotated surface codes with reconfigurable neutral atom arrays (2024), arXiv:2412.01391 [quant-ph].
- [48] S. Bravyi, G. Smith, and J. A. Smolin, Trading classical and quantum computational resources, *Phys. Rev. X* **6**, 021043 (2016).
- [49] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataiades, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, Logical quantum processor based on reconfigurable atom arrays, *Nature* **626**, 58–65 (2023).
- [50] P. Panteleev and G. Kalachev, Quantum ldpc codes with almost linear minimum distance, *IEEE Transactions on Information Theory* **68**, 213 (2022).
- [51] J. Old, M. Rispler, and M. Müller, Lift-connected surface codes, *Quantum Science and Technology* **9**, 045012 (2024).
- [52] Q. Xu, H. Zhou, D. Bluvstein, M. Cain, M. Kalinowski, J. Preskill, M. D. Lukin, and N. Maskara, Batched high-rate logical operations for quantum ldpc codes (2025), arXiv:2510.06159 [quant-ph].
- [53] K. Sahay, Y. Lin, S. Huang, K. R. Brown, and S. Puri, Error correction of transversal cnot gates for scalable surface-code computation, *PRX Quantum* **6**, 020326 (2025).

METHODS

Deriving equivalent representations of a logical operator: an algebraic approach

In the following, we describe how to derive the different equivalent representations associated with a logical operator X_L of arbitrary k La-cross codes. The discussion is analogous for Z_L operators.

(i) *Derive the shortest logical operator.* Starting from the matrix H_X in Eq.(1), the logical operators are obtained by finding the vectors $\mathbf{x}$ that solve the problem $H_X \mathbf{x} = 0$. The vector $\mathbf{x}$ represents the support of the logical operator X_L acting on $N = n^2 + r^2$, with $r = n - k$, physical qubits:

$$X_L = \bigotimes_{l=1}^N X_l^{\mathbf{x}_l}, \quad (7)$$

where X_l is a Pauli X operator acting on the l th qubit, $\mathbf{x}_l$ is the l th component of the vector $\mathbf{x}$, and hence $X_l^{\mathbf{x}_l}$ is an operator acting at the qubit location specified by $\mathbf{x}_l$. The physical qubits of the array are indexed in increasing order from left to right and from top to bottom, starting from the $n \times n$ main lattice and then the $r \times r$ sublattice. We solve the problem numerically via Gauss elimination and select the shortest solutions defined on a single row of the main lattice. There exist k^2 independent solutions that represent the different inequivalent logical operators.

(ii) *Finding equivalent representations.* Let us consider a particular solution X_L that has support on the row i of the main lattice, characterized by the support pattern $P = \{p | \mathbf{x}_{(i-1)n+p} = 1\}_{1 \leq p \leq n}$, which in the following we index as $X_L^{(i,P)}$ and derive its different equivalent representations. Any logical operator $X_L^{(i,P)}$ can be translated vertically along the main lattice via multiplication by X -stabilizers. The logical operator $X_L^{(i,P)}$ on row $i \leq r$, can be translated into a two-row equivalent representation acting on the two underlying $(i+1)$ th and $(i+k)$ th rows as follows. Consider the set of X -stabilizers corresponding to the rows of H_X which we re-index as $\{S_{i,p}^X := S_{(i-1)n+p}^X\}_{p=1,\dots,n}^{i=1,\dots,r}$. One can show that applying the subset $\{S_{i,p}^X\}_{p \in P}$ of X -stabilizers, whose ancilla qubits lie in the row immediately below the data qubits in the support P of $X_L^{(i,P)}$, and that couple upward to them via nearest-neighbor coupling yields the desired translation

$$\prod_{p \in P} S_{i,p}^X X_L^{(i,P)} = X_L^{(i+1,P)} X_L^{(i+k,P)}, \quad i \leq r. \quad (8)$$

We note that this relation is symmetric under the permutation of the indices of the logical operators, that is, one can perform vertical translations both upward and downward. Therefore, any two-row representation acting on any given pair of rows chosen among the three rows i , $i+1$, or $i+k$ can be mapped into a single-row representation acting solely on the row that is not part of the pair.

We introduce a convenient approach to perform these calculations directly on the vectors $\mathbf{x}^{(i,P)}$: We construct the set of vectors $\{h_{i,p}^X := h_{(i-1)n+p}^X\}_{p=1,\dots,n}^{i=1,\dots,r} \in \mathbb{F}_2^N$ that correspond to the nr X -stabilizers. The product of stabilizers in Eq. (8) can be reexpressed as a macro-stabilizer vector, that is, a sum of stabilizer vectors, $M_{(i,P)}^X$, defined as

$$M_{(i,P)}^X = \sum_{p \in P} h_{i,p}^X. \quad (9)$$

With this new notation, Eq. (8) reads as a sum of the vectors $M_{(i,P)}^X$ and $\mathbf{x}^{(i,P)}$,

$$M_{(i,P)}^X + \mathbf{x}^{(i,P)} = \mathbf{x}^{(i+1,P)} + \mathbf{x}^{(i+k,P)}. \quad (10)$$

By varying i , it is possible to construct r sums of stabilizer vectors $\{M_{(i,P)}^X\}_{i=1}^r$, so that all possible translations between the rows are obtained via all possible combinations of these sums. To obtain as many representations as the length of the considered logical operator, our protocol requires taking all the possible combinations of $M_{(i,P)}^X$ acting on $X_L^{(i,P)}$ to identify all the equivalent logical operators. Then, these operators are ordered according to their length, and, finally, we retain only the shortest non-overlapping representatives.

For the logical Z operators, the discussion is analogous, but applied to the columns of the lattice.

We find by inspection that for $k = 2, 3$ La-cross codes it is always possible to find as many equivalent representations of the same logical operator as its length which span either a single row or two rows of the main lattice. Instead, for $k = 4$ we find lattice sizes n for which some equivalent representatives span three rows. It follows that for our BS gadget to be fault-tolerant with high-rate La-cross codes, $\ell \leq k - 1$ -long representations of a logical operator must be considered. The protocol remains valid, but requires more steps per logical operation. For an optimal compilation strategy, it may be advantageous to restrict to code sizes that admit small- ℓ representations for large- k La-cross codes (e.g., $\ell = 2$ for the case $k = 4$).

Details on quantum error correction simulations

In the following, we provide additional details about the QEC numerical simulations of the logical Hadamard rotation protocol discussed in the main text. Since the general structure of the protocol has already been described, we focus here primarily on the decoding strategy and noise model employed.

The logical transversal property of our scheme enables the use of a correlated decoding strategy [27, 43, 44] that significantly reduces the number of error correction cycle required. Indeed, only a single round of error correction is needed after a logical gate as errors propagate deterministically through Clifford gates and can therefore be tracked in software rather than corrected physically at each step. To implement this decoding strategy, we use the **detectors** in **Stim**, defined as the binary addition of measurement outcomes. In a standard quantum memory experiment, one typically associates a detector to each stabilizer by pairing its measurement outcomes across two consecutive rounds. The resulting detector signal is therefore, only sensitive to changes in the stabilizer value between rounds that corresponds to actual physical or measurement errors. In a logical circuit, the logical gates propagate errors between code blocks. To avoid flagging the same error multiple times, detectors must be designed to be sensitive only to newly introduced errors. This is achieved by backpropagating each stabilizer through the preceding logical gate and rewriting it as a product of stabilizers in the updated frame. This is always possible thanks to the fault-tolerance of our gadget (see main text). Detectors are then defined as the parity of the measurement outcomes associated with these stabilizers. The explicit backpropagation rules for all stabilizer types and codes are given in the main text.

The final measurement of the data qubits in both codes provides additional information that can be exploited to improve decoding performance. When the data qubits

are measured in a certain basis, one can reconstruct the value of stabilizers of the corresponding type by properly combining the measurement outcomes. These “artificial” stabilizers can then be backpropagated to the previous error correction round, in the same way as ordinary stabilizers, to define additional detectors. For the LDPC code, the end of a circuit corresponds to a memory experiment and this backpropagation is trivial. In contrast, for the BS code, the X -measurement is performed immediately after the transversal CZ operation. The backpropagated “artificial” X -stabilizer is equal to the product of a Z -stabilizer of the LDPC code and an X -stabilizer of the BS code. Since the latter is never measured before in the circuit, the corresponding detector is defined as the combination of the other stabilizers.

In addition of the detectors, we must define a **logical_observable**, that is typically the logical measurement outcome of a logical qubit. In the noiseless situation, this observable should be deterministic. When errors occur, the value of this observable might be affected and the decoder’s role is to infer the noiseless logical observable value based on detector outcomes. In our scheme, logical measurements in the BS and the LDPC code are always random when considered separately but their joint product is deterministic. We, therefore, define the latter as being the logical observable of our system in order to evaluate the logical error of our protocol.

We used a circuit-level noise model that considers errors only after two-qubit gates, measurements, and resets. Single-qubit gates and idle operations are assumed to be perfect. That is motivated by the fact that these operations typically have one order of magnitude lower infidelities than two-qubit gates and measurements in most quantum computing platforms, such as neutral atoms. For two-qubit gates we enforced two-qubit depolarizing noise, that is, we assumed uniformly distributed Pauli er-

rors drawn at random from $\{I, X, Z, Y\}^{\otimes 2} \setminus \{I \otimes I\}$ with probability $p/15$. For measurements and resets in the X -basis, we inject phase-flip errors with probability p .

From the set of detectors, the logical observable and the noise model defined above, **Stim** is able to derive a detector error model (DEM) i.e. the set of all single error mechanisms along with their associated probabilities, the flipped detectors and potentially the flipped logical observable. We decoded the syndrome information using Belief Propagation with Ordered Statistics Decoder (BP+OSD) [45, 46]. We used the minimum-sum variant of Belief Propagation, setting the number of iterations to 4 and optimizing the scaling factor, s , to obtain the lowest logical error probabilities and the correct asymptotic scaling without error floors. The optimal scaling factors were found to be $s = 0.2$ for the Hadamard rotation protocol and $s = 0.3$ for the memory experiment. OSD was used in combination sweep mode up to order 1. Monte Carlo samplings were performed using the Sinter library.

We note that for the rotation protocol we have used BP+OSD over the full decoding problem, consisting of the two code memories – BS and La-cross – along with the two transversal entangling operations between them. That is because the DEM of quantum LDPC codes naturally suffers from hyperedges [45] and the logical entangling operations between different codes introduce additional hyperedges to the decoding graph due to errors propagating from one code to the other. Although this is a legitimate decoding strategy, hypergraph complexity and time overhead increase, which can ultimately lead to under-performance of the decoder. To address this issue, more efficient decoding techniques consisting in decomposing the decoding problem into different parts have been introduced [53]. We let to future work the issue of addressing the decoding problem for our BS gadget with more refined and possibly more efficient methods.