

Universal Quantum Computation via Scalable Measurement-Free Error Correction

Stefano Veroni¹,¹ Alexandru Paler^{1,2}, and Giacomo Giudice^{1,*}

¹*PlanQC GmbH, Münchener Str. 34, 85748 Garching, Germany*

²*Department of Computer Science, Aalto University, 02150 Espoo, Finland*

 (Received 16 January 2025; revised 23 June 2025; accepted 8 October 2025; published 14 November 2025)

We show that universal quantum computation can be concretely made fault-tolerant without mid-circuit measurements. To this end, we introduce a measurement-free deformation protocol of the Bacon-Shor code to realize a logical CCZ gate. Combined with a fold-transversal logical Hadamard gate, this enables a universal set of fault-tolerant operations using only transversal gates and qubit permutations. For the purpose of benchmarking under circuit-level noise, we develop an efficient method to simulate non-Clifford circuits with a small number of Hadamard gates. Separately, we demonstrate that certain CSS codes can be concatenated without measurements or having to rely on a universal logical gate set. This is made possible by means of a resource-efficient gadget—termed the “disposable Toffoli gadget”—that realizes the error-correcting feedback. Then, under concatenation of the Bacon-Shor code, we observe a fault-tolerance threshold at a circuit-level depolarizing noise rate of approximately 0.12%. Together, the deformation and concatenation protocols outline a blueprint for a fully fault-tolerant architecture without any feed-forward operation, particularly suited to state-of-the-art neutral-atom platforms.

DOI: [10.1103/lkk1-v6wp](https://doi.org/10.1103/lkk1-v6wp)

I. INTRODUCTION

Given the inherently noisy nature of quantum hardware, it is crucial to develop approaches that suppress or limit the impact of errors. One possible pathway is quantum error correction (QEC), which provides a means to detect and correct errors during the execution of quantum algorithms [1,2]. At the heart of QEC lies the encoding of quantum information into a specific subspace within the broader physical Hilbert space—often achieved through a non-local encoding—allowing the system to be resilient against local errors. This redundancy, while effective, also significantly increases the resource overhead required for implementing QEC, both in terms of time and space. As a result, the choice of the error-correction protocol should be tailored to the specific quantum computing hardware, both to minimize overhead and to best exploit the platform’s characteristics. Despite these complexities, recent experimental breakthroughs have marked remarkable progress in

QEC across different architectures, such as superconducting qubits [3–11], trapped ions [12–21], nitrogen vacancies in diamonds [22], and neutral-atom arrays [23–26].

In traditional QEC approaches, measurements of mutually-commuting correlators—known as *error syndromes*—are used to inform a classical decoding algorithm, which then determines the corrective actions applied to the system, either through physical intervention or software-based tracking of errors [1]. These measurements often correspond to *stabilizer* operators, which commute with logical operators, such that the logical state remains unaffected [27].

In some quantum computing platforms, such as those based on neutral atoms or trapped ions, the relatively slow nature of such measurements often hinders the efficiency of this syndrome extraction [17,18,23,28–32]. An alternative approach, *measurement-free* (MF) QEC, has emerged as a promising solution [33–39], replacing classical processing with a combination of unitary dynamics and a dissipative element needed to remove the entropy introduced by noise. The latter is achieved through reset operations or a continuous supply of fresh ancilla qubits, which has recently been demonstrated in neutral-atom platforms [40–42]. In particular, the first fully *fault-tolerant* (FT) proposal correcting a single error was achieved in Ref. [38], by adapting Steane-type error-correction techniques [43]. Later, some of the authors of this work showed that by using redundant syndrome information, alternative schemes can be formulated

*Contact author: giacomo.giudice@planqc.eu

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

with a lower overhead in terms of both the number of qubits and gate count [39].

The notion of fault-tolerance is essential for the design of an error-corrected quantum computer. Broadly speaking, an FT implementation is capable of limiting error propagation, thus achieving a logical error rate that is lower than the physical error rate, as long as the latter remains below a certain threshold. This is relevant both for the QEC round (which must not increase the noise level during its noisy execution), and for logical operations. *Transversal* operations are particularly appealing, as they are straightforwardly FT [1]. Transversal operations correspond to a single layer of physical gates, each of which acts on at most one physical qubit of each logical qubit. Therefore, errors can propagate between logical qubits, but remain correctable, since QEC is applied to each logical qubit block individually.

However, it is well known that no QEC code can have a transversal universal gate set [44], and, in particular, the only transversal gates for two-dimensional codes are unitaries belonging to the Clifford group [45]. Several ways have been proposed to get around these no-go results. These include magic state distillation [46–50], code switching between appropriate codes [51–53], using non-local connectivity between Bacon-Shor codes [54], implementing pieceable constructions where operations are intertwined with intermediate error correction [55], or allowing for a loss of distance in concatenated codes [56–58].

The main theoretical challenges for MF QEC are two-fold: (i) implementing universal quantum computation at the logical level, and (ii) demonstrating scalability beyond small-distance codes. We tackle these challenges in the following way. We show (i) that a *universal gate set* is achievable with the Bacon-Shor code, by exploiting transversal operations and a MF code deformation procedure to realize a transversal CCZ gate; and (ii) that a universal gate set is *not required* to perform MF QEC cycles at higher levels of concatenation, allowing for an efficient scaling up of the code distance. During the completion of this work, the authors of Ref. [59] demonstrated MF code-switching between color codes to implement a universal set of logical gates. Equipped with this universal set, it is then possible to concatenate the code with itself to reach higher distances. In contrast, we show that the cost of concatenation can be lowered using resource-efficient gadgets, hence drastically increasing logical performance and scalability.

This work is organized as follows. In Sec. II, we review the Bacon-Shor code, which was shown to yield the best performance among currently devised MF schemes [37–39]. In Sec. III, we construct a MF procedure to *deform* the code and exploit the protocol proposed in Ref. [54] to implement a logical CCZ gate, thus enabling MF universal quantum computation. We then turn to address the second challenge in Sec. IV. While this code

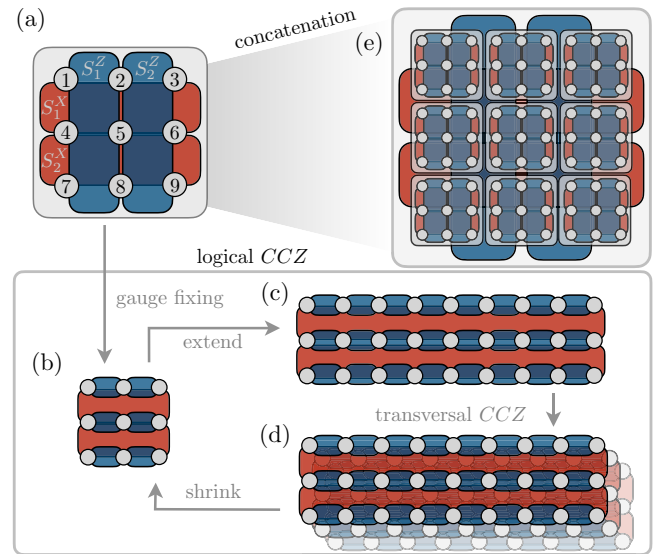


FIG. 1. (a) The 3×3 Bacon-Shor code, with red (blue) areas depicting the support of X -type (Z -type) stabilizer operators. (b) Any Bacon-Shor code has a gauge freedom. In particular, the pairwise $Z_i Z_j$ gauges along rows can be chosen to be $+1$, resulting in a Shor code. We call this the *Shor gauge*. To perform a logical CCZ , each logical qubit must start from this gauge, and then be extended to a 3×9 Bacon-Shor code, as shown in (c). The logical CCZ is then performed between logical qubits, by a transversal application of physical CCZ gates, with permutations across the columns. (d) To return to the original 3×3 configuration, a shrink move is performed. (e) To increase the tolerance to errors, *concatenation* of the same code is performed, where the data qubits of a code are composed of logical qubits at a lower level.

deformation allows for a universal logical gates set, we take a different approach and implement MF QEC on concatenated codes *without* the need for a universal gate set. Inspired by Ref. [33], we implement a feedback operation which we dub the *disposable Toffoli gadget* and is significantly more efficient than using the aforementioned logical CCZ construction. The essence of this gadget is to *unencode* ancilla qubits—storing stabilizer information—into a repetition code and then exploiting the (partial) transversality between repetition codes and a target code from the Calderbank-Steane-Shor (CSS) [60,61] family. Finally, in Sec. V we discuss potential advantages of this measurement-free approach, and how they are particularly suitable for neutral-atom platforms.

II. THE BACON-SHOR CODE

In this section, we introduce the Bacon-Shor code (Sec. II A), we summarize the construction for logical CCZ used in Ref. [54] (Sec. II B), and introduce some properties relating the Bacon-Shor code and the repetition code (Sec. II C), which we will use throughout this paper.

A. Notation

The main properties of a QEC code are typically described by the triplet $[[n, k, d]]$, where n is the number of physical qubits, k is the number of logical qubits, and d is the code distance, i.e. the minimum number of single-qubit operations required to get from one codeword to another. Additionally, we define t as the number of *tolerable* errors—i.e. all errors of weight smaller than or equal to t can be corrected—satisfying $t \leq \lfloor (d-1)/2 \rfloor$.

In this work, we will be focusing on the Bacon-Shor code [62,63]. This code essentially combines an n_1 -qubit repetition code against Z errors with an n_2 -qubit repetition code against X errors in an $n_1 \times n_2$ array of data qubits. Labeling the qubits in this array with (i, j) , the stabilizers for the $[[n_1 n_2, 1, \min(n_1, n_2)]]$ Bacon-Shor code are generated from

$$S_i^X = \prod_{j=1}^{n_2} X_{i,j} X_{i+1,j}, \quad S_i^Z = \prod_{j=1}^{n_2} Z_{i,j} Z_{i,j+1}. \quad (1)$$

These are depicted in Fig. 1(a) for the minimal error-correcting instance $[[9, 1, 3]]$. This specific instance is the main focus of this paper, as its MF QEC implementation, proposed in Ref. [39], is particularly simple.

The logical operators are $X_L = \prod_j X_{i,j}$ and $Z_L = \prod_i Z_{i,j}$ for any value of i or j respectively, as shown in Fig. 2(a). As with all CSS codes, all logical Pauli operations are trivially transversal. Additionally, the logical CNOT can be implemented transversally, by performing a CNOT between pairs of data qubits. Additionally, the logical Hadamard gate is fold-transversal—it corresponds to a Hadamard gate on all data qubits and a relabeling of the physical qubits, which can be achieved with a reflection along the diagonal to swap X and Z stabilizers [63].

The Bacon-Shor code is a subsystem code [1,62], such that multiple states encode the same logical codeword. These states are equivalent up to multiplication with *gauge operators*, which commute with the stabilizers and logical operators, thus not affecting the logical information. The gauge group is generated by the pairs $X_{i,j} X_{i+1,j}$ ($Z_{i,j} Z_{i,j+1}$) acting on qubits in the same column (row). In what follows, we will be mainly interested in the “Shor gauge” of the code, where all $Z_{i,j} Z_{i,j+1}$ gauge operators have eigenvalue $+1$. In this gauge, the codewords are equivalent to those of Shor’s code [64]. In particular, in this gauge certain logical states become a tensor product of cat states

$$|\pm\rangle_L^{(Z)} = \frac{1}{2\sqrt{2}} (|000\rangle \pm |111\rangle)^{\otimes 3}. \quad (2)$$

These states are particularly important, since they can be prepared fault-tolerantly without additional overhead [12]. We will denote the opposite gauge, in which all $X_{i,j} X_{i+1,j}$ have eigenvalue $+1$, as the “anti-Shor” gauge. In this

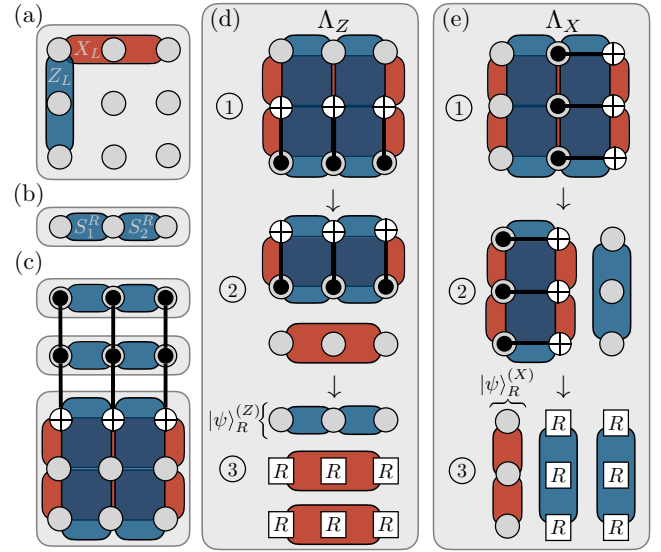


FIG. 2. (a) The logical operators of a Bacon-Shor code correspond to a single row (X_L) or column (Z_L). (b) A bit-flip repetition code is formed by pairwise $S_i^R = Z_i Z_{i+1}$ stabilizers along a chain. (c) A transversal CCZ between a three-qubit repetition code and $d = 3$ Bacon-Shor code. Note that this gate is only unidirectional. (d)–(e) The unencoding operation Λ_Z (Λ_X) maps a Bacon-Shor code down to a bit-flip (phase-flip) repetition code. Intuitively, each column (row), supporting a separate representation of the relevant logical operator, is mapped to a qubit of the resulting repetition code. The stabilizers after each layer of gates are depicted.

gauge, the states $|0\rangle_L$ and $|1\rangle_L$ have a similar structure as Eq. (2), this time as a tensor product of cat states along the columns instead of rows. A mixed gauge can be chosen as well, as long as the gauge operators commute. In particular, the rotated surface is a specific gauge choice of the Bacon-Shor code [65].

B. Logical CCZ

It was shown in Ref. [54] that a $m \times m^k$ Bacon-Shor code in the Shor gauge supports a transversal logical $C^k Z_L$, up to a qubit permutation. In particular, as shown in Fig. 3, the logical CCZ_L between three 3×9 logical qubits is implemented by a *single layer* of 27 physical CCZ gates

$$CCZ_L = \prod_{j=0}^8 \prod_{i=0}^2 CCZ_{(i,j), (i \oplus \lfloor j/3 \rfloor, j), (i \oplus j, j)}, \quad (3)$$

where it is assumed that (i, j) indices start from 0 for ease of notation, and $\oplus$ represents addition modulo 3.

Since the Hadamard and CCZ gates together define a universal gate set, it is then sufficient to demonstrate that these two gates can be performed fault-tolerantly. As our starting point, we choose the 3×3 Bacon-Shor code, since

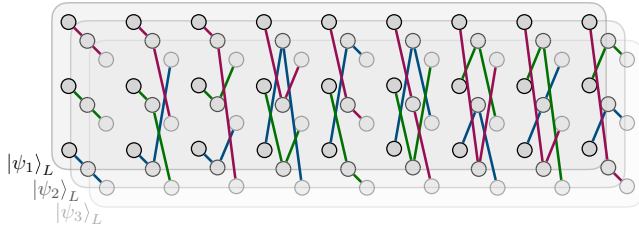


FIG. 3. Graphical depiction of the connectivity required for the transversal CCZ_L $|\psi_1\psi_2\psi_3\rangle_L$ between three logical states $|\psi_k\rangle_L$, $k = 1, 2, 3$, each encoded in the 3×9 the Bacon-Shor codes, in the Shor gauge. Different colors correspond to different CCZ gates between rows, and highlight that the permutations occur only across columns.

it has a MF QEC implementation [39], as well as a straightforward fold-transversal Hadamard gate. However, certain technicalities need to be addressed. The transversal realization of the logical Hadamard gate H_L is only supported on a $m \times m$ patch of physical qubits, while the CCZ_L gate in Eq. (3) requires a $m \times m^2$ patch. Furthermore, the latter requires to be in the Shor gauge, which is not preserved under the application of H_L . To address these issues, in Sec. III we will demonstrate a fully MF protocol to implement the logical gate set: first, the code is mapped to the Shor gauge, by means of a *gauge fixing* procedure [Fig. 1(b)], then extended to a 3×9 code [Fig. 1(c)]; now the logical CCZ can be performed using Eq. (3), cf. Fig. 1(d). To return to the original code, we then shrink it down to a 3×3 code [Fig. 1(b)].

C. Relationship with the repetition code

Before continuing with the logical protocol, we wish to outline some properties connecting CSS codes with repetition codes, which we will exploit later on. The bit-flip (phase-flip) repetition code is defined on n qubits by pairwise stabilizers $S_i^R = Z_i Z_{i+1}$ ($S_i^Z = X_i X_{i+1}$), cf. Fig. 2(b), and protects against X -type (Z -type) errors. The respective logical codewords are $|0/1\rangle_R^{(Z)} = |0/1\rangle^{\otimes n}$ and $|\pm\rangle_R^{(X)} = |\pm\rangle^{\otimes n}$. Generalizing the constructions proposed in Ref. [33], throughout this work we exploit the fact that *unidirectional* transversal gates exist between repetition codes and CSS codes. In particular, a logical $C^k X$ ($C^k Z$) gate on a distance- d code can be controlled from k length- d bit-flip repetition codes, as illustrated in Fig. 2(c) for the Toffoli gate. This gate is transversal, as it is composed of d physical $C^k X$ ($C^k Z$) gates targeting d qubits supporting the X_L (Z_L) operator of the code. As explained in Appendix A, the stabilizer structure is preserved.

Additionally, we introduce a gadget to convert a logical codeword to a codeword of the repetition code. We call this an *unencoding* gadget, and can be used when partial protection against either bit flips or phase flips is sufficient. For an arbitrary logical state $|\psi\rangle_L = \alpha |0\rangle_L + \beta |1\rangle_L$,

$|\alpha|^2 + |\beta|^2 = 1$, we introduce the unencoding operations

$$\Lambda_{X/Z} |\psi\rangle_L = |\psi\rangle_R^{(X/Z)} \otimes |0\rangle^{\otimes 6}, \quad (4)$$

which converts it to $|\psi\rangle_R^{(X/Z)} = \alpha |0\rangle_R^{(X/Z)} + \beta |1\rangle_R^{(X/Z)}$. Notice that these gadgets are not unitary, since information about the gauge choice is removed by the reset operations R . The implementation of the unencoding gadgets for the 3×3 Bacon-Shor code is illustrated in Figs. 2(d) and 2(e), and is similar to the reverse of the encoding circuit. This is somewhat reminiscent of code morphing [66], albeit differing both conceptually and implementation-wise. Effectively, the unencoding gadget Λ_Z (Λ_X) maps the value of each column's Z_L (row's X_L) on a qubit of the resulting repetition code—thus it is applicable to any gauge choice of the Bacon-Shor code, e.g. the rotated surface code. For generic CSS codes, such gadgets can be implemented, albeit with a larger overhead, by repeatedly performing Hadamard tests on different logical strings of the logical operators, and performing a majority vote, as was proposed in Ref. [33].

By combining the unencoding gadget with the property in Fig. 2(c), we can engineer a FT feedback operation without measurements: a logical operation can be conditioned on another logical state, at the price of having to unencode the latter to a repetition code. This can still be useful in many situations, such as when the control state is later discarded. For instance, this will be used later on to perform the gauge fixing procedure detailed in Sec. III, and in Sec. IV to construct an efficient feedback operation for concatenated codes.

III. MEASUREMENT-FREE FAULT-TOLERANT LOGICAL CCZ

In this section, we discuss in detail the protocol for a fault-tolerant CCZ_L , schematically illustrated in Fig. 1. For the distance-3 Bacon-Shor code, this corresponds to the protocol in Fig. 4. We describe in more detail this protocol in Sec. III A and benchmark it under depolarizing noise in Sec. III B.

A. Circuit design

To apply Eq. (3), the Shor gauge must be first enforced on the code. In general, this is necessary in a FT quantum computation even if we initialized logical states in the correct gauge, since the H gate exchanges the Shor and anti-Shor gauge. Furthermore, the QEC circuit proposed in Ref. [39], and later adapted in Sec. IV, corrects every single-qubit error up to a gauge operator. From a measurement-free perspective, fixing the gauge is somewhat more challenging than correcting single errors, as we must account for both gauge flips as well as possible single-qubit errors, so the heuristics developed in Ref. [39]

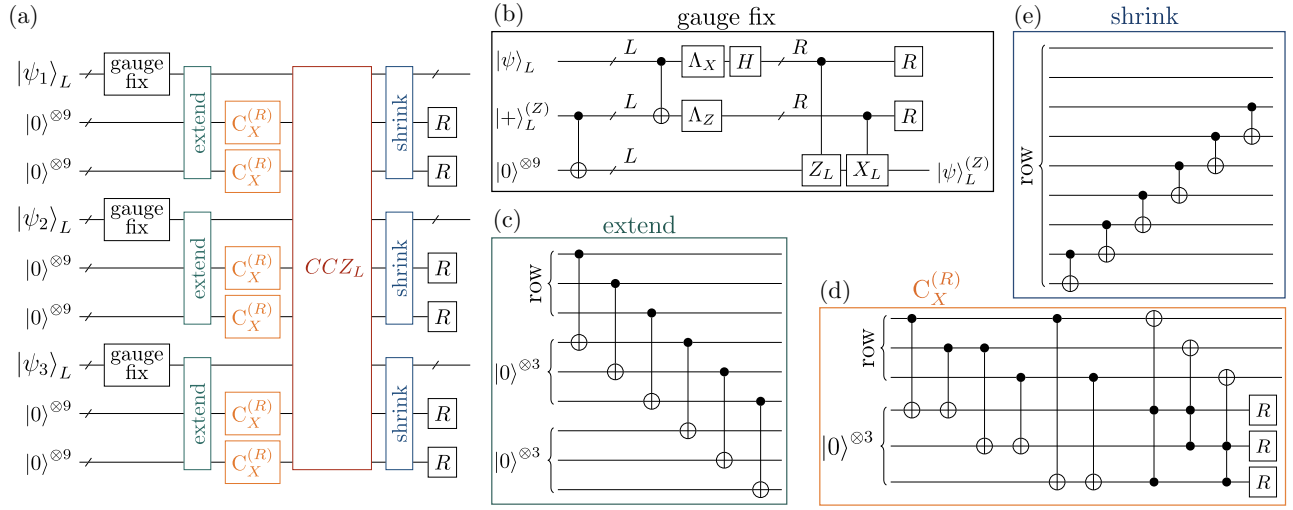


FIG. 4. (a) Components of the protocol for the CCZ_L gate between 3×3 Bacon-Shor codes. (b) Measurement-free gauge fixing by teleportation. A logical Bell state $(|00\rangle_L + |11\rangle_L)/\sqrt{2}$ is first prepared in the Shor gauge. Note that, because the $|+\rangle_L$ is prepared in the Shor gauge, the last logical qubit does not need to be prepared in $|0\rangle_L$. This auxiliary state is then entangled with the logical state, and then MF teleportation using CX and CZ gates is performed, by first unencoding to the repetition code, cf. Sec. II C. (c) The extend gadget deforms a 3×3 Bacon-Shor code to its 3×9 variant, by exploiting the Shor gauge in the previous step. (d) A measurement-free error-correction round for the repetition code, used to correct for single bit flips on each triplet of each row of the logical states. (e) The extend gadget is reversed by a shrink gadget, bringing the logical state back to a 3×3 Bacon-Shor code.

are not applicable. To gauge a logical state, we propose two different alternatives: (i) a *teleportation* protocol and (ii) *Steane-type gauge fixing*. Both have similar performances under depolarizing noise, so we focus on the former, while the latter is relegated to Appendix B.

In the teleportation protocol, the logical state is moved to a fresh register, which is initialized in the correct gauge. The gauge information of the original state is removed by the unencoding gadget. Its MF variant is illustrated in Fig. 4(b), and requires two additional logical registers. Note that it can also correct single bit-flip errors, and has the added benefit of converting potential leakage errors into computational errors.

To extend a Bacon-Shor code in the Shor gauge from 3×3 to 3×9 , it is sufficient to perform two consecutive transversal CNOTs from the code to a $|0\rangle^{\otimes 9}$, cf. Fig. 4(c). However, this may propagate bit flips that may lead to uncorrectable errors, so to make the procedure FT we introduce bit-flip corrections $C_X^{(R)}$ along the rows, cf. Fig. 4(d). To shrink back a logical state to the 3×3 code, it is then sufficient to perform a series of CNOTs, illustrated in Fig. 4(e). Since the circuits act independently on each row, phase flips cannot propagate to different rows and remain correctable. In principle, such schemes can be generalized to any 3×3^k Bacon-Shor code, to directly perform transversal $C^k Z$ gates [54].

B. Numerical results

We benchmark the scheme in Fig. 4 against the hardware-agnostic depolarizing noise, where after each

gate, a random Pauli error is applied with probability p . That is, given the set of Pauli strings of length ℓ , i.e. $\mathcal{P}_\ell = \{I, X, Y, Z\}^{\otimes \ell}$, any ℓ -qubit gate is followed by the channel

$$\mathcal{E}(\rho) = (1 - p)\rho + \frac{p}{|\mathcal{P}_\ell| - 1} \sum_{\substack{P \in \mathcal{P}_\ell \\ P \neq I^{\otimes \ell}}} P \rho P^\dagger. \quad (5)$$

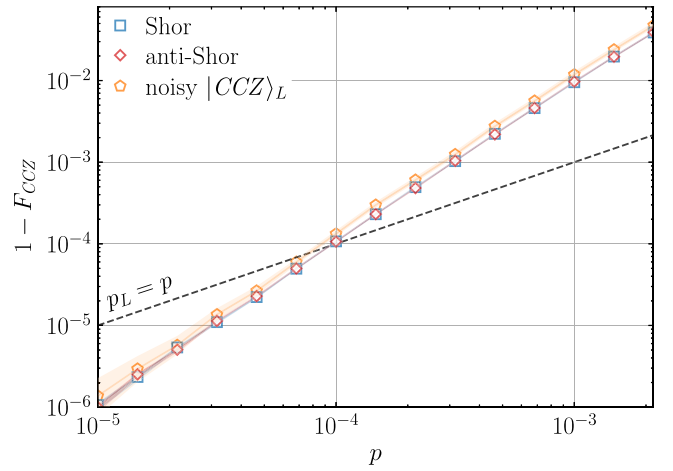


FIG. 5. Average error rate of the logical CCZ gate under depolarizing noise, computed using Eq. (6). The performance is benchmarked for initial states encoded without errors in the Shor (blue) or anti-Shor (red) gauges. Starting from $|+++\rangle_L$, we can prepare the non-Clifford resource state $|CCZ\rangle_L$. We therefore show the fidelity (orange) of its preparation using a noisy FT preparation in the Shor gauge, and without any gauge fixing. Shaded regions correspond to 99% confidence intervals.

Considering that the circuits contain more than 80 qubits and 27 physical CCZ gates, both state vector simulators as well as Clifford simulators are not suitable. Nevertheless, the logical states contain very few non-zero amplitudes in the computational basis, so the state vector is extremely *sparse*. In particular, the worst-case scenario is $|\pm\rangle_L$ in the anti-Shor gauge, since it is the eigenstate of six independent XX gauge operators and X_L , leading to $2^{6+1} = 128$ non-zero amplitudes. If one simply keeps track of these non-zero entries—much as one would do with “pen and paper” calculation—the simulations can be performed with very moderate computational resources.

For this task, we develop `SparseStates.jl` [67] a simulation package in Julia which can efficiently simulate circuits with few *branching* gates, i.e. gates that create superpositions in the computational basis (which are only H gates in our case). This allows us to compute thousands of circuit samples per second with circuits involving more than 100 qubits and hundreds of Toffoli-like gates. To the best of our knowledge, only Ref. [68] implements a similar method, based on hash maps, and with further gate rearrangements for the optimized simulation of quantum algorithms. Instead, our state-vector implementation is array-based, which allows us to perform most operations in place—avoid unnecessary memory allocations—and opening up the possibility to exploit vectorized instructions available on most general-purpose processors.

To assess the performance of a protocol ideally realizing an operation U at the logical level, we compute the average logical fidelity [69], by averaging over all different combinations of Pauli eigenstates $\mathcal{S}_\ell = \{|\pm X\rangle, |\pm Y\rangle, |\pm Z\rangle\}^{\otimes \ell}$ as input states [70]. To account for the degeneracy of the gauge choice, we expand the overlap between each target state and the resulting state ρ_σ as a sum over expectation values of logical Pauli strings

$$F_U = \frac{1}{|\mathcal{S}_\ell|} \sum_{|\sigma\rangle \in \mathcal{S}_\ell} \langle \sigma | U^\dagger \rho_\sigma U | \sigma \rangle_L = \sum_{\substack{|\sigma\rangle \in \mathcal{S}_\ell \\ P \in \mathcal{P}_\ell}} c_\sigma^{(P)} \text{Tr}(P_L \rho_\sigma), \quad (6)$$

where ℓ is the number of logical qubits U acts on and $c_\sigma^{(P)} = (1/|\mathcal{S}_\ell|) \langle \sigma | U^\dagger P U | \sigma \rangle_L$. Different Kraus operators are sampled according to Eq. (5), simulating up to 2^{24} realizations or 2^{10} failures per state, whichever comes first.

The numerical results for the CCZ_L gate are presented in Fig. 5, for two different gauge choices. However, these choices have minimal impact on the results, owing to the gauge fixing procedure (see also Appendix B). The quadratic scaling law of the logical error confirms the fault-tolerance of the scheme. Additionally, we show the performance of preparing the $|CCZ\rangle_L = CCZ_L |++\rangle_L$ magic state, with a noisy FT initialization of the Shor gauge, and hence not requiring any gauge fixing. This state can serve as a non-Clifford resource, and may be useful

for alternative constructions, for instance as the starting point of a distillation protocol to obtain better logical fidelities. Interestingly, the preparation of $|CCZ\rangle_L$ exhibits a modest performance penalty, despite the absence of gauge fixing. This can be attributed to the noisy preparation and the inherent asymmetry in the CCZ_L protocol. States like $|++\rangle_L$ are maximally vulnerable to phase errors, and most subroutines in the protocol let single phase flips propagate, as they do not lead to uncorrectable errors. However, at higher orders in p , multiple phase flips can accumulate, leading to logical phase errors.

Finally, we note that a MF code switching protocol was recently proposed by Ref. [59], to realize a logical T gate and achieving a breakeven fidelity of 2.6×10^{-4} . This figure is, broadly speaking, comparable to the numerical results presented here. Notably, decomposing a CCZ gate requires six CX gates and seven T gates [71,72]. The relative utility of these non-Clifford gates depends on the algorithmic context.

IV. MEASUREMENT-FREE CONCATENATION

Having demonstrated all the ingredients for FT and MF universal computation, in this section we show that the Bacon-Shor code can be scaled up by means of concatenation, thus achieving exponential suppression of noise. In Sec. IV A we introduce code concatenation and how it applies to MF QEC. Then, in Sec. IV B we present an optimized QEC protocol for the concatenated Bacon-Shor code. We do not exploit the deformation procedure developed in Sec. III, but rather introduce a gadget to perform MF feedback operations. We call this the *disposable* Toffoli gadget, since it effectively acts as a transversal gate as long as the control qubits do not need to be protected or preserved, such as the case of ancilla qubits. Numerical benchmarks are presented in Sec. IV C.

A. Code concatenation

We explore repeated *concatenation* of the same code, which we denote as

$$\mathcal{C}_N = \underbrace{\mathcal{C} \circ \dots \circ \mathcal{C}}_{N \text{ times}}, \quad (7)$$

such that the first layer of concatenation is $\mathcal{C}_1 \equiv \mathcal{C}$. The concatenation operation $\circ$ is a way of combining two codes: the logical qubits of the *inner* code are used as physical qubits of the *outer* one, cf. Fig. 1(e). By recursively applying this operation, we can then construct a N -layer concatenated code.

Concatenation is a modular way of increasing the code distance. Indeed, by concatenating N times a CSS code of distance d , the resulting distance is $d_N = d^N$ [1]. The number of errors t_N that we can correct depends on the

TABLE I. Key properties of the first layers of concatenation, the effective distance d^{eff} , the number of correctable faults t ($d^{\text{eff}} = 2t + 1$) and pseudothresholds obtained from numerical simulations with depolarizing noise.

Code	$[[n, k, d]]$	d^{eff}	t	Pseudothreshold
$\mathcal{C}_1$	$[[9, 1, 3]]$	3	1	3.84×10^{-3}
$\mathcal{C}_2$	$[[81, 1, 9]]$	7	3	1.95×10^{-3}
$\mathcal{C}_3$	$[[729, 1, 27]]$	15	7	1.50×10^{-3}

decoding strategy. Since in MF QEC decoding and corrections are operated by noisy quantum circuits rather than via classical processing, the complexity of the available feedback operations is in practice limited. In the simplest decoding strategy each unit operates autonomously, decoding each inner code individually, *layer by layer*. This still achieves an exponential suppression, albeit slower, of errors at the cost of a polynomial increase in the size of the error-correction circuit. In a FT protocol adopting this strategy, any t faults at the lower level of concatenation are correctable, hence we have the recurrence relation $t_N = (t + 1)(t_{N-1} + 1) - 1$, which yields

$$t_N = (t + 1)^N - 1. \quad (8)$$

For a base code correcting a single error ($t = 1$), we obtain the sequence $t_N = 1, 3, 7, 15, \dots$ for the first layers of concatenation. This corresponds to an *effective distance* $d_N^{\text{eff}} = 2t_N + 1$, exponentially worse than the best case d_N , cf. Table I. Nevertheless, one still has an exponential suppression of errors by increasing t_N , guaranteeing that a threshold exists [73–75]. To understand this, we consider a circuit-level error model, where errors occur at every gate independently with probability p . Defining the logical failure rate at the N th level of concatenation as p_N , with $p_0 \equiv p$, we have $p_{N+1} \sim C p_N^{t+1}$, where $C > 1$ is a constant, corresponding to the number of combinations of $t + 1$ faults leading to a logical error. Therefore, if $p < p_{\text{th}} = C^{-1/t}$, then recursively $p_{N+1} < p_N \forall N$, hence the suppression of the error rate is faster than the circuit growth and a threshold is attained.

To achieve the upper bound $t_N = \lfloor (d^N - 1)/2 \rfloor$, decoding strategies exist, but require message-passing between different layers [76]. To see how layer-by-layer decoding is not optimal, consider the concatenation of a three-bit classical repetition code with itself. This is equivalent to a nine-bit repetition code, which can correct up to four errors. However, by decoding each inner code individually, we can correct only up to three errors, as two errors in two subcodes can lead to the wrong logical state.

One can see layer-by-layer MF correction as a *decoding trade-off*: we trade fault-tolerance for lower *latency*. The latter includes (1) *measurement latency*—the time to perform the quantum measurements; (2) *decoding latency*—the time to communicate the syndromes to the

decoder, decode the syndromes, and then apply or track the corrections. Feed-forward (FF) decoding uses measurements and classical computation to achieve the t_N upper bound, but MF decoding can speed up the logical performance by allowing the decoding units to operate autonomously. This addresses both measurement and decoding latencies, at the price of suboptimal decoding and lower error tolerance.

Regarding the measurement latency, MF decoding is attractive for several platforms, including trapped ions and neutral atoms, as it completely side-steps the issue of having slow measurements, which is one of the main bottlenecks in a QEC round [17,18,23,28–32]. In Appendix C we analyze the threshold measurement time for which MF decoding achieves the same fault-tolerance as a standard FF protocol, in the case of neutral-atom arrays. We estimate that the measurement latency in current state-of-the-art hardware is a significant bottleneck, such that MF schemes could outpace their conventional counterpart for small to mid-distance codes ($d^{\text{eff}} \lesssim 15$).

The MF protocol avoids the decoding latency as well. The fastest decoders have been shown to operate in real-time only for small distances [8,9]. Real-time decoding at larger distances with many logical qubits is extremely challenging, and an active field of research [77–79]. It generally requires trading tolerance for speed, too: faster decoding, such as union-find [80], has lower thresholds.

B. Circuit design

The starting point of our concatenation procedure is the MF error-correcting circuit for the 3×3 Bacon-Shor code proposed in Ref. [39]. For each subcircuit $C_X^{(m)}$ ($C_Z^{(m)}$) correcting bit flips (phase flips), Z -type (X -type) stabilizers are extracted to ancilla qubits, as shown in Fig. 6. Additionally, a third stabilizer $S_3 = S_1 S_2$ is extracted, to be FT against circuit-level noise. For a more detailed discussion on how to choose the set of redundant stabilizers and determine the circuit's design, we refer the reader to Ref. [39]. When concatenating this code with itself, we promote both physical qubits as well as ancilla qubits to logical qubits of the lower layer.

For the correction, we construct a gadget to perform the feedback efficiently, exploiting the fact that the phase information in the ancilla qubits is unimportant. We unencode the ancilla qubits to a length- d_N repetition code using the unencoding gadgets repeatedly [see Figs. 2(d) and 2(e)], to exploit the transversality of multi-controlled gates described in Sec. II C and shown in Fig. 2(c). We note that the involved physical operations either have disjoint supports or act on the same column (row). As can be seen in Fig. 2(a) these columns (rows) are orthogonal to the X_L (Z_L) operator, thus the unencoding protocol Λ_Z (Λ_X) is FT, as each error location can affect at most one qubit of the repetition code. By partially unencoding

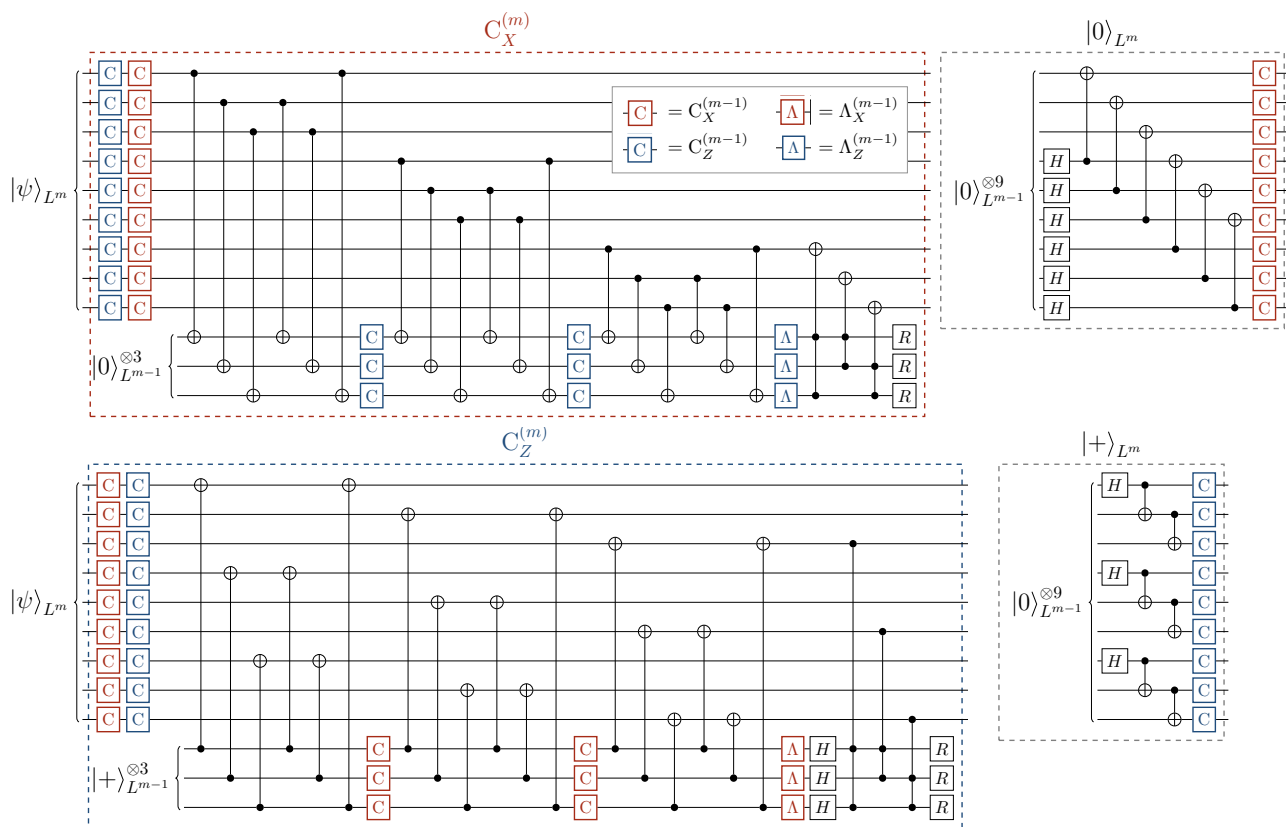


FIG. 6. Measurement-free fault-tolerant QEC implementation for the concatenated Bacon-Shor code, composed of a $C_X^{(m)}$ and a $C_Z^{(m)}$ subcircuit correcting for X and Z errors, respectively, where m is the layer of concatenation. At the lowest layer $C_1 \equiv C$, intermediate error-correcting subcircuits $C_{X/Z}^0$ and unencoding operations $\Lambda_{X/Z}^0$ are replaced by identities. The ancilla registers at higher layers of concatenation are initialized fault-tolerantly using the circuits on the right, with $|0\rangle_{+0} \equiv |0\rangle_{+}$.

the logical ancilla qubits to a repetition code, we can then perform transversal multi-controlled operations on the logical qubits. This is well-suited to the MF context, since it reduces the overhead for MF feedback operations while preserving fault-tolerance—as phase errors on the ancillas are not transferred to the data qubits, and single bit flips result in at most a correctable bit flip on the data qubits.

Finally, when concatenating a desired code, we need to include the error-correcting rounds for the underlying layers, which we call *subrounds*. To ensure fault-tolerance, subrounds can be introduced after every single gate [75]. However, for the Bacon-Shor code we consider, we realize that in many locations this is not necessary. In particular, we can allow errors on the ancilla qubits to propagate onto the logical qubit, as long as they correspond to a gauge of the code. For instance, in $\mathcal{C}_X$ ($\mathcal{C}_Z$), a single phase (bit) flip after initialization of the first ancilla would result in an error $Z_{1,1}Z_{1,2}$ ($X_{1,1}X_{2,1}$) on the data qubits after the first two layers of CNOTs. If we let these errors propagate further however, we would end up potentially with correlated errors along rows (columns) which are aligned with the logical X_L (Z_L) operator, and would lead to an uncorrectable error. Therefore, we just need to place a $\mathcal{C}_X$

(C_Z) subround on the ancilla qubits after having extracted a $Z_{i,j} Z_{i+1,j}$ ($X_{i,j} X_{i,j+1}$) gauge.

C. Performance

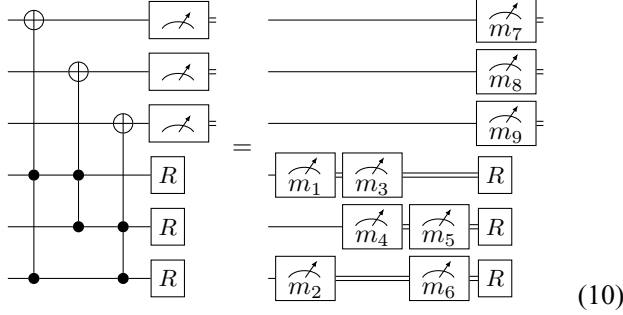
We consider the performance of the circuits presented in Sec. IV B under depolarizing noise, defined in Eq. (5). We consider the first three layers of concatenation, summarized in Table I. The circuits are simulated with `stim` [81], which allows for large system-size simulations by using the stabilizer tableau formalism.

To convert Toffoli-like gates into a circuit composed solely of Clifford gates, we note that all ancillas are reset after use, such that we can separately sample their values as either 0 or 1. We therefore convert multi-qubit gates using the principle of deferred measurement

$$\begin{array}{c} \oplus \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} \begin{array}{c} \\ R \\ \\ R \\ \end{array} = \begin{array}{c} \\ \\ \diagup \\ \diagdown \\ \diagup \\ \diagdown \end{array} \begin{array}{c} X \\ \hline \bullet \\ \hline \bullet \\ \hline \end{array} \quad (9)$$

and including the corresponding noise channels defined by Eq. (5), such that every Toffoli-like gate is followed by a three-qubit depolarizing noise channel.

Classical logic in `stim` is limited to exclusive-or, so, in order to benefit from the faster compiled samplers, we need to further delay the feedback operations. To do so, we precompute a table associating each pair of virtual measurements to a list of future measurements that need to be flipped. For example, for a quantum memory experiment with the $C_X^{(R)}$ gadget, the final part is replaced as follows:



where $m_1 \wedge m_2$ flip m_7 , $m_3 \wedge m_4$ flip m_8 and $m_5 \wedge m_6$ flip m_9 . In general, propagations of Pauli corrections are more complicated, but can still be computed efficiently, since the circuit is composed entirely of Clifford gates. For this purpose, we use `stim's FlipSimulator`. This approach eliminates entirely all mid-circuit classical logic, enabling the use of the high-performance `CompiledMeasurementsampler`. After sampling the feedback-free circuits, the measurement outcomes are sequentially updated using the precomputed table [82]. We then simulate noisy circuits stochastically for a variable number of cycles N_C , for all eigenstates of the Pauli matrices, sampling up to 2^{24} realizations or 2^{10} failures—whichever occurs first.

In Fig. 7, we show the logical error rate per cycle $p_L = 1 - F_I^{1/N_C}$, calculated by averaging over all the eigenstates of the logical Pauli operators, cf. Eq. (6). The numerical results indicate that the failure rate of C_N scales asymptotically as p^{t_N+1} , as expected for a protocol that is robust against t_N faults at the circuit level. From this data, we extract an asymptotic threshold $p_{th} \simeq 1.2 \times 10^{-3}$, and the pseud thresholds in Table I, potentially relevant for real implementations. Compared to universal concatenated constructions that rely on measurements [58], we note that our protocol has a comparable asymptotic threshold, despite having the overhead of being measurement-free. For higher levels of concatenations, we can approximate the logical error rate sufficiently below threshold as $p_L(p) \sim p_L(p_{th}) (p/p_{th})^{t_N+1}$. As shown in the inset of Fig. 7, we can then estimate the level of concatenation required to reach a target p_L .

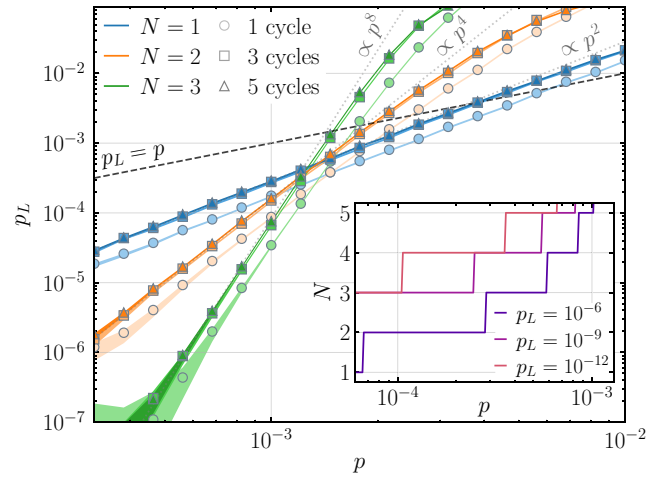


FIG. 7. Logical error rate per cycle p_L vs gate error rate p , for the first three layers of concatenation. The initial state is prepared without errors, and multiple QEC rounds are performed (markers). The solid lines correspond to the asymptotic estimates, extracted from a polynomial fit. From these estimates, we extrapolate the concatenation level N required to reach different target logical error rates (inset).

Overall, these results indicate that the MF error-correcting protocols can achieve below-breakeven logical error rates on near-term quantum devices, assuming physical error rates below 0.1%, generally considered within reach. In particular, state-of-the-art trapped ion platforms have already demonstrated comparable error rates [83–85] and neutral atom arrays are making rapid progress towards this goal [86–90]. Furthermore, a few layers of concatenation ($N \leq 3$) are sufficient to achieve a strong suppression of logical errors, within the regimes of interest for next-generation devices [91].

V. OUTLOOK

We have constructed a fully fault-tolerant architecture for universal and scalable quantum computation that does not require any measurements.

First, by performing a measurement-free gauge fixing and code deformation, we have demonstrated a fault-tolerant protocol for the realization of a logical CCZ gate in the Bacon-Shor code. Together with the code's fault-tolerant initialization and transversal Hadamard gate, this enables measurement-free, fault-tolerant and universal operations. These ideas of gauge fixing are very powerful, as well-known error-correction concepts such as lattice surgery and code deformation have been recast as a gauge fixing procedure [92]. It is likely that in the future such concepts can be exploited in the measurement-free context, to devise more efficient universal logical operations.

Second, we propose to concatenate measurement-free implementations of small codes to realize higher-distance

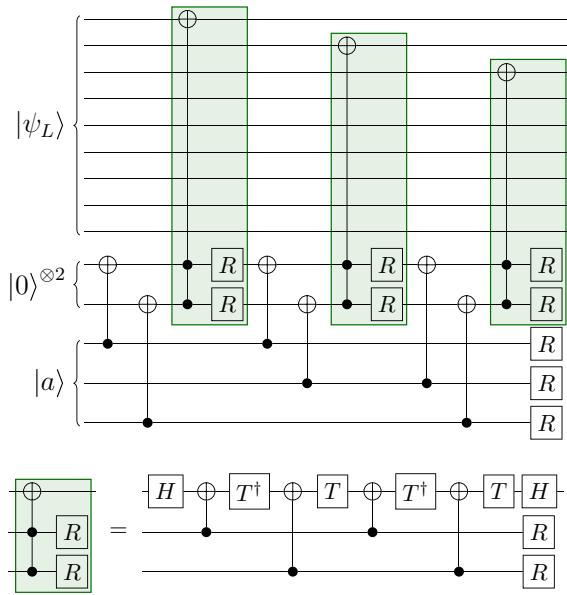


FIG. 8. Decomposition of the correction step for the 3×3 Bacon-Shor code in Fig. 6 in terms of two-qubit gates. By copying the control qubits in the ancilla register $|a\rangle$, we can then use a Toffoli-reset gadget with fewer gates than the full decomposition, as already recognized in Ref. [38].

codes and thus achieve protection against an arbitrary number of faults. Remarkably, these concatenated protocols do not require a universal gate set, as we use a gadget to perform the feedback operation in a hardware-efficient way. We stress that such constructions are not unique to the Bacon-Shor code, but could be adapted to other codes of the CSS family as well. In particular, we note that the disposable Toffoli gadget is also compatible with the $d = 3$ rotated surface code, without any modifications. By performing numerical simulations, we find that the performance is competitive and can be considered for error-correcting experiments on near-term devices. Furthermore, it can be expected that the protocols can be further tailored to a real hardware device. For example, by exploiting biased noise, the performance of measurement-free protocols can be significantly enhanced [39]. Furthermore, neutral atoms provide an ideal platform for the implementation of such protocols, as native CCZ operations have been demonstrated [87]. In the case where such operations are not available, the protocols can be adapted without spoiling fault-tolerance. As an example, as shown in Fig. 8, the correction operations can always be decomposed fault-tolerantly, by copying the control qubits to an auxiliary register, and performing a simplified decomposition of the feedback operation [38].

Altogether, this architecture requires non-local, but heavily parallelizable operations. This is an ideal setting for neutral-atom platforms, which have demonstrated parallel Clifford operations and parallel shuttling of multiple

registers for the realization of transversal entangling operations [23,24]. Furthermore, continuous loading of atoms provides a reservoir of fresh qubits, which can emulate reset operations in a scalable way [40–42]. Despite a potential performance overhead, measurement-free protocols can be a valid alternative to significantly speed up the *logical clock rate*. In Appendix C, we perform “back-of-the-envelope” calculations that suggest that for fault-tolerant experiments in the foreseeable future ($d \lesssim 15$), measurement times would need to significantly decrease to match the clock times of their measurement-free counterpart. This further motivates measurement-free error correction as a viable and scalable pathway towards fault-tolerant quantum computation.

ACKNOWLEDGMENTS

We acknowledge inspiring discussions with Gavin Brennen and Johannes Zeiher. A.P. acknowledges partial funding from the Defense Advanced Research Projects Agency [under the Quantum Benchmarking (QB) program under Awards no. HR00112230006 and No. HR001121S0026 contracts], and was supported by the QuantERA grant EQUIP through the Academy of Finland, decision number 352188. The views, opinions and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

DATA AVAILABILITY

The data that support the findings of this article are openly available [82].

APPENDIX A: TRANSVERSALITY OF CONTROLLED GATES BETWEEN REPETITION CODES AND CSS CODES

In this Appendix, we show that a logical $C^k X$ ($C^k Z$) gate on a distance- d CSS code can be controlled from k length- d bit-flip repetition codes, without affecting the stabilizer structure of the logical qubits involved. These consist of d physical $C^k X$ ($C^k Z$) gates targeting the d qubits of the code supporting the X_L (Z_L) operator.

For the CX gate, using the well-known propagation rules

$$\begin{aligned} \begin{array}{c} \boxed{X} \\ \oplus \end{array} &= \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \boxed{X} \\ \oplus \end{array}, & \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \boxed{X} \\ \oplus \end{array} &= \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \boxed{X} \\ \oplus \end{array}, \\ \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \boxed{Z} \\ \oplus \end{array} &= \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \boxed{Z} \\ \oplus \end{array}, & \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \boxed{Z} \\ \oplus \end{array} &= \begin{array}{c} \bullet \\ \oplus \end{array} \begin{array}{c} \boxed{Z} \\ \oplus \end{array}, \end{aligned} \quad (A1)$$

we realize that the only non-trivial propagation are the S^Z stabilizers of the code overlapping with X_L , which

get mapped to the stabilizers of the repetition code. By definition, X_L commutes with the stabilizers, meaning every overlap with S^Z stabilizers involves an even number of qubits. But an even product of distinct Z operators can always be absorbed by the stabilizer generators of the repetition code. In the case of a CZ gate, the targets support the Z_L operator, and equivalent results are obtained by using the identity $CZ = (I \otimes H)CX(I \otimes H)$. By symmetry these arguments also hold for unidirectional gates on length d phase-flip repetition codes controlled by distance- d CSS codes.

We can now extend these result to three-qubit gates as well. Again, using the propagation rules for the Toffoli gate

$$\begin{aligned}
 & \begin{array}{c} \boxed{X} \\ \text{---} \end{array} \begin{array}{c} \bullet \\ \text{---} \end{array} = \begin{array}{c} \bullet \\ \text{---} \end{array} \begin{array}{c} \boxed{X} \\ \text{---} \end{array} \begin{array}{c} \oplus \\ \text{---} \end{array}, \quad \begin{array}{c} \text{---} \\ \bullet \end{array} \begin{array}{c} \text{---} \\ \bullet \end{array} = \begin{array}{c} \text{---} \\ \bullet \end{array} \begin{array}{c} \text{---} \\ \bullet \end{array} \begin{array}{c} \oplus \\ \text{---} \end{array} \begin{array}{c} \boxed{X} \\ \text{---} \end{array}, \\
 & \begin{array}{c} \boxed{Z} \\ \text{---} \end{array} \begin{array}{c} \bullet \\ \text{---} \end{array} = \begin{array}{c} \bullet \\ \text{---} \end{array} \begin{array}{c} \boxed{Z} \\ \text{---} \end{array} \begin{array}{c} \oplus \\ \text{---} \end{array}, \quad \begin{array}{c} \text{---} \\ \bullet \end{array} \begin{array}{c} \text{---} \\ \bullet \end{array} = \begin{array}{c} \text{---} \\ \bullet \end{array} \begin{array}{c} \text{---} \\ \bullet \end{array} \begin{array}{c} \oplus \\ \text{---} \end{array} \begin{array}{c} \boxed{Z} \\ \text{---} \end{array}, \quad (A2)
 \end{aligned}$$

the only non-trivial propagation arises from the S^Z stabilizers of the code overlapping with X_L . The even number of overlapping qubits results in an even number of CZ operations between the controlling bit-flip codes, which reciprocally cancel. Similarly, in the CCZ gate, the non-trivial propagation is due the intersection of S^X stabilizers and Z_L , which again results in the propagation of an even number of CZ s, which reciprocally cancel out.

In general, the $C^k X$ ($C^k Z$) gates controlled on the bit-flip codes only have non-trivial propagation of the S^Z (S^X) stabilizers of the code overlapping with X_L (Z_L). These lead to an even number of $C^{k-1} Z$ gates between the controlling

bit-flip codes, which mutually cancel out, thus preserving the stabilizer structure.

APPENDIX B: STEANE-TYPE MF GAUGE FIXING

As an alternative, we also propose a gauge fixing inspired by Steane-type MF QEC [38,43], which both corrects single-qubit bit flips and enforces the Shor gauge, using fewer qubits if fast resets are available.

This protocol is illustrated in Fig. 9. The main idea is to treat each row one at a time, extracting each ZZ gauge operator to detect bit flips (orange box). For each bit flip, we apply a XX gauge operator between the current row (green box) and the next one (purple box). This sets each of the Z gauges to their $+1$ eigenvalue, and moves single bit-flip errors to the last row. We can then perform a similar bit-flip correction on the last row to correct for a potentially remaining bit flip. As with Steane-type QEC, this scheme avoids the propagation of more than one error.

The scheme can be sped up, by extracting ZZ gauge operators of the top and bottom rows simultaneously, and then applying the XX gauge operator between said row and, respectively, the one below or above. By repeating this process iteratively, between rows progressively closer to the middle, errors are moved to the middle row. Finally, one round of extraction and correction on the middle row corrects a potentially remaining bit flip.

A comparison of the two different gauge fixing protocols presented is shown in Fig. 10, exhibiting very similar performance. We note that the performance of the teleportation circuit is independent of the initial state's gauge, while the Steane-type gauge fixing has a marginally lower error rate when the gauge is correct.

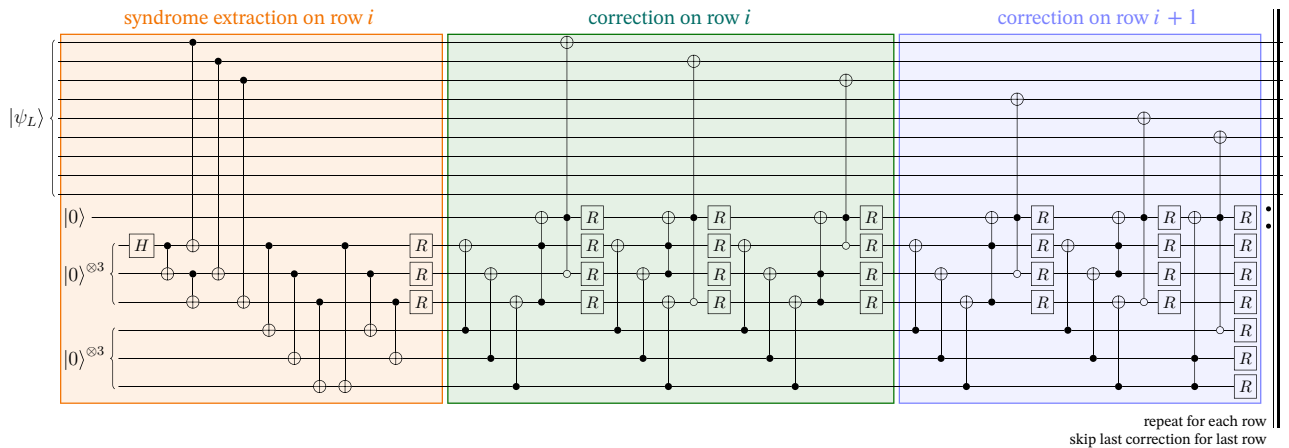


FIG. 9. Gauge fixing by adapting Steane-type error correction. This circuit maps the X errors of a single row into a three-qubit cat state, decodes it by a redundant extraction of the parities [39] (orange box), and, if a bit flip occurred, applies a XX gauge operator between the current row (green box) and the next one (purple box). This circuit is repeated for all rows except for the last one, where remaining bit-flip errors are extracted and then corrected by applying only the first part of the corrections.

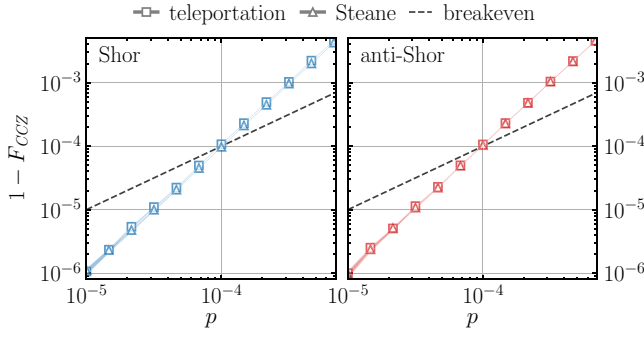


FIG. 10. Logical error rate of the CCZ_L gate, employing the teleportation or the Steane-type gauge fixing protocols. This is computed using Eq. (6), for states encoded in either the Shor (left) or anti-Shor gauge (right).

APPENDIX C: PRACTICAL CONSIDERATIONS

In this section, we compare the implementation of the MF protocols and more conventional QEC protocols on a potential neutral-atom platform, from the point of view of logical clock time. In particular, we assume a two-dimensional array of atoms, with parallel shuttling capabilities. Since the concatenation procedure increases the footprint of the logical register, it is important to account for potentially long shuttling times. For measurements, we consider the situation of either reconfigurable arrays with a separate readout zone [23], or a static array with local imaging of ancillae, enabled for example by shelving of the data qubits to different atomic levels. For simplicity, we only consider shuttling and measurement times, since these are typically much more important than the execution time of individual gates [23,93].

With regards to shuttling, its speed is limited by the amount of heating it induces. Two main shuttling trajectories have been designed to address this problem, namely a constant-jerk (CJ) movement [93] or an acceleration profile based on shortcut-to-adiabaticity (STA) techniques [94]. If an atom undergoes K movements of distance D_i each, the time taken by each movement is

$$\Delta t_i = (\alpha K D_i^2)^f, \quad (C1)$$

where $f = 1/4$ ($1/6$) for CJ (STA) shuttling, and α is a constant that depends on the experimental setting and the trajectory. More details can be found at Refs. [93,94]. On the other hand, with regards to mid-circuit measurements and feed-forward, current state-of-the-art experiments across various atomic species and qubit encodings have attained times ranging between 1 ms to 20 ms [23,29–32,88,95].

For the MF QEC protocol, we assume the quasi-parallel extraction of individual stabilizers, since we can devise an alternative MF QEC protocol using four ancilla qubits instead of three, cf. Fig. 11. The advantage of this protocol

is that it avoids measuring the stabilizer S_3 (cf. Sec. IV B), which requires longer-range interactions, by measuring S_1 and S_2 twice. Compared to Fig. 1(e), it exhibits a slightly lower pseudothreshold, despite having more redundancy. The total time required for a subcircuit C^N at N levels of concatenation then can be broken down as

$$T_C^{(N)} = T_{\text{circ}}^{(N)} + 5T_C^{(N-1)}, \quad (C2)$$

where $T_{\text{circ}}^{(N)}$ is the time required for shuttling in between logical gates, whereas $5T_C^{(N-1)}$ accounts for the error-correcting subcircuits C^{N-1} .

In this scheme, each ancilla moves at most $K = 6$ times: thrice horizontally and twice diagonally during the syndrome extraction [cf. Fig. 11(b)], and at most once more horizontally for the application of the feedback Toffoli operation—the unencoding gadget can be realized with nearest-neighbor gates only. Each horizontal movement, at the N th concatenation level, covers a distance $D_i^{(N)} = \delta x d_{N-1}$, where δx is the spacing between data qubits and $d_N = 3^N$; diagonal movements are longer by a factor $\sqrt{2}$. We note that the shuttling schedule presented is compatible with the constraints imposed by the acousto-optical-deflectors (AODs) used to steer atoms in neutral-atom hardware [96,97]. Then, by using Eq. (C1), we have

$$T_{\text{circ}}^{(N)} = \sum_i \Delta t_i = \kappa (K \alpha 3^{2(N-1)} \delta x^2)^f, \quad (C3)$$

where κ depends on the parallelization and relative size of the moves. There are a total of 10 steps of parallel shuttling, covering syndrome extraction—which contributes eight steps as per cf. Fig. 11(c)—and the feedback operation. Six steps only include shorter horizontal moves, while in the remaining four the longest movement is diagonal. Then, $\kappa = 6 + 4 \cdot 2^f$. Inserting this result in Eq. (C2) we obtain

$$T_C^{(N)} = 5T_C^{(N-1)} + \kappa (K \alpha 3^{2(N-1)} \delta x^2)^f. \quad (C4)$$

Unraveling this recursive relation yields

$$\begin{aligned} T_C^{(N)} &= \kappa (K \alpha \delta x^2)^f \sum_{n=0}^{N-1} 5^n 3^{2(N-1-n)f} \\ &= \kappa (K \alpha \delta x^2)^f \frac{5^N - 3^{2Nf}}{5 - 3^{2f}}. \end{aligned} \quad (C5)$$

For comparing with equivalent feed-forward implementations, we use a QEC code that is equivalent by the number of correctable errors t_N rather than the distance, cf. Eq. (8). Thus, we consider a logical qubit encoded in a $d_N^{\text{eff}} \times d_N^{\text{eff}}$ register, using some CSS code such as the rotated surface code, allowing for local stabilizer extractions without shuttling. We assume that by correctly decoding the

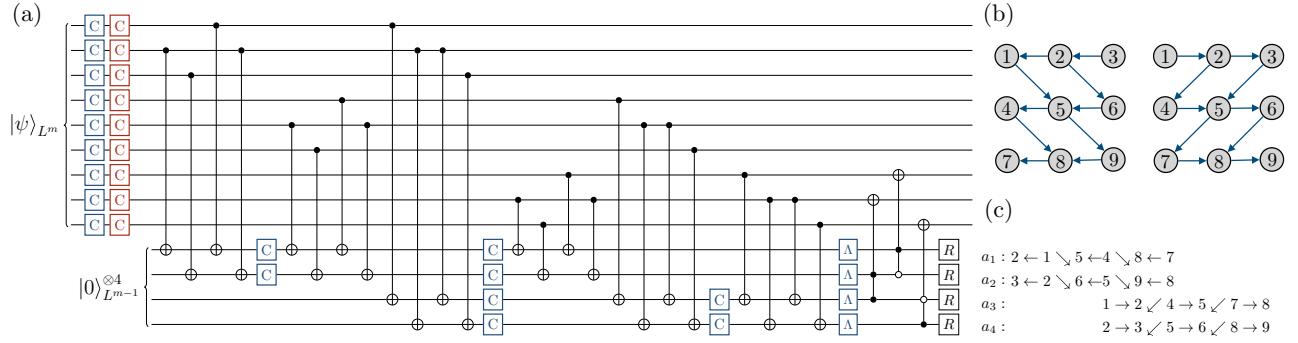


FIG. 11. (a) Alternative measurement-free and fault-tolerant QEC implementation for the Bacon-Shor code, using four ancillae. The colored operations follow the same convention of Fig. 6, indicating subcircuit error correction and unencoding routines. (b) Shuttling pattern for the first two (left) and last two (right) ancillae. (c) Shuttling schedule for the syndrome extraction, showing which data qubit each ancilla a_i interacts with at a certain moment in time, and the direction of the shuttling in between. The order with which ancillae interact with data qubits is important to preserve fault-tolerance [39]. The schedule proposed is chosen to maximize the number of parallel operations, while also avoiding the propagation of uncorrectable errors from the ancillae to the data qubits.

syndrome information, we can correct up to $t_N = \lfloor (d_N^{\text{eff}} - 1)/2 \rfloor$ faults on the data qubits. In this setting, a QEC cycle is then dominated by the measurement and feed-forward time T_{FF} . In particular, for an extraction of a single type of syndromes, the measurements of the stabilizers is generally performed d_N^{eff} times to be robust against circuit-level noise. Within this measurement time we also include the shuttling time to a separate readout zone, if needed, which can be non-negligible [23].

The breakeven measurement and feed-forward time T_{FF}^* for which the MF protocol with shuttling would match the FF implementation, for the same tolerance t_N , is then

$$T_{FF}^*(N) = 2\kappa (K\alpha\delta x^2)^f \frac{5^N - 3^{2Nf}}{5 - 3^{2f}} \frac{1}{2^{N+1} - 1}, \quad (\text{C6})$$

where the factor 2 arises from the fact that the MF protocol measures X -type and Z -type stabilizers one after the other, whether we assume that they are measured in parallel in FF schemes. With an estimate of $\delta x = 4 \mu\text{m}$, and the experimentally demonstrated values [93,94]

$$\begin{aligned} \alpha_{\text{STA}} &\approx 1.4 \times 10^{-11} \text{ ms}^6/\mu\text{m}^2, \\ \alpha_{\text{CJ}} &\approx 1.3 \times 10^{-7} \text{ ms}^4/\mu\text{m}^2, \end{aligned} \quad (\text{C7})$$

we obtain

$$T_{FF}^* \simeq 0.23 \text{ ms}, 0.64 \text{ ms and } 1.6 \text{ ms} \quad (\text{C8})$$

for $N = 1, 2$ and 3 with STA shuttling, and roughly twice as much with CJ shuttling. Recalling that current state-of-the-art experiments have attained a T_{FF} that ranges between 1 ms to 20 ms [23,29–32,88,95], then the MF approach not only avoids the need for mid-circuit measurements but can shorten the QEC cycle time.

For the purposes of clarifying the impact of the experimental design on the performance of the protocol, we note that the QEC cycle time of the MF protocol is estimated to scale as

$$T_C^{(N)} \propto \alpha^f \propto \frac{m^{\frac{1}{2}(1+f)} w_{\text{tr}}^{1-f}}{U_{\text{tr}}^{\frac{1}{2}(1-f)}}, \quad (\text{C9})$$

where m is the mass of the used atom, w_{tr} and U_{tr} are the width and the depth of the trap, and recall that $f = 1/4$ ($1/6$) for CJ (STA) shuttling. This implies that by increasing the trapping depth and adopting lighter atomic species as ancilla qubits, a significant reduction in time could be achieved.

With regards to the comparison between MF and FF approaches, we note that, on the one hand, the FF implementation could be improved in principle by not applying d rounds of syndrome extraction for every type of stabilizer, as recent research has shown that for a logical computation fewer rounds of measurements can be performed and fault-tolerance can still be preserved, at the price of a more complicated decoding [98]. It remains to be seen if such approaches can be adapted to the measurement-free setting as well. On the other hand, for a FF implementation of a logical computation it is important to take into account shuttling times to a separate readout zone, if required. An increase in the number of logical qubits or in code distance necessarily increases the separation from the readout zone, which in turn lengthens the measurement time. Considering that practically-relevant quantum algorithms will require hundreds to thousands logical qubits [99], future experiments will need to tackle such increasing overhead. Moreover, the complexity of classical decoding increases with the distance of the QEC code, in turn providing additional overhead and latency.

- [1] B. M. Terhal, Quantum error correction for quantum memories, *Rev. Mod. Phys.* **87**, 307 (2015).
- [2] E. T. Campbell, B. M. Terhal, and C. Vuillot, Roads towards fault-tolerant universal quantum computation, *Nature* **549**, 172 (2017).
- [3] Google Quantum AI, Exponential suppression of bit or phase errors with cyclic error correction, *Nature* **595**, 383 (2021).
- [4] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, G. J. Norris, C. K. Andersen, M. Müller, A. Blais, C. Eichler, and A. Wallraff, Realizing repeated quantum error correction in a distance-three surface code, *Nature* **605**, 669 (2022).
- [5] Y. Zhao *et al.*, Realization of an error-correcting surface code with superconducting qubits, *Phys. Rev. Lett.* **129**, 030501 (2022).
- [6] Google Quantum AI, Suppressing quantum errors by scaling a surface code logical qubit, *Nature* **614**, 676 (2023).
- [7] R. S. Gupta, N. Sundaresan, T. Alexander, C. J. Wood, S. T. Merkel, M. B. Healy, M. Hillenbrand, T. Jochym-O'Connor, J. R. Wootton, T. J. Yoder, A. W. Cross, M. Takita, and B. J. Brown, Encoding a magic state with beyond break-even fidelity, *Nature* **625**, 259 (2024).
- [8] L. Caune *et al.*, Demonstrating real-time and low-latency quantum error correction with superconducting qubits, *arXiv:2410.05202*.
- [9] Google Quantum AI and Collaborators, Quantum error correction below the surface code threshold, *Nature* **638**, 920 (2024).
- [10] A. Eickbusch *et al.*, Demonstration of dynamic surface codes, *Nat. Phys.* (2025).
- [11] N. Lacroix *et al.*, Scaling and logic in the colour code on a superconducting quantum processor, *Nature* **645**, 614 (2025).
- [12] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, M. Cetina, and C. Monroe, Fault-tolerant control of an error-corrected qubit, *Nature* **598**, 281 (2021).
- [13] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, T. M. Gatterman, S. K. Halit, K. Gilmore, J. A. Gerber, B. Neyenhuis, D. Hayes, and R. P. Stutz, Realization of real-time fault-tolerant quantum error correction, *Phys. Rev. X* **11**, 041058 (2021).
- [14] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. G. Poschinger, Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer, *Phys. Rev. X* **12**, 011032 (2022).
- [15] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, R. Blatt, P. Schindler, M. Müller, and T. Monz, Demonstration of fault-tolerant universal quantum gate operations, *Nature* **605**, 675 (2022).
- [16] Y. Wang, S. Simsek, T. M. Gatterman, J. A. Gerber, K. Gilmore, D. Gresh, N. Hewitt, C. V. Horst, M. Matheny, T. Mengle, B. Neyenhuis, and B. Criger, Fault-tolerant one-bit addition with the smallest interesting color code, *Sci. Adv.* **10**, eado9024 (2024).
- [17] L. Postler, F. Butt, I. Pogorelov, C. D. Marciniak, S. Heußen, R. Blatt, P. Schindler, M. Rispler, M. Müller, and T. Monz, Demonstration of fault-tolerant Steane quantum error correction, *PRX Quantum* **5**, 030326 (2024).
- [18] I. Pogorelov, F. Butt, L. Postler, C. D. Marciniak, P. Schindler, M. Müller, and T. Monz, Experimental fault-tolerant code switching, *Nat. Phys.* **21**, 298 (2025).
- [19] C. Ryan-Anderson *et al.*, High-fidelity teleportation of a logical qubit using transversal gates and lattice surgery, *Science* **385**, 1327 (2024).
- [20] M. P. da Silva *et al.*, Demonstration of logical qubits and repeated error correction with better-than-physical error rates, *arXiv:2404.02280*.
- [21] N. Berthussen, J. Dreiling, C. Foltz, J. P. Gaebler, T. M. Gatterman, D. Gresh, N. Hewitt, M. Mills, S. A. Moses, B. Neyenhuis, P. Siegfried, and D. Hayes, Experiments with the four-dimensional surface code on a quantum charge-coupled device quantum computer, *Phys. Rev. A* **110**, 062413 (2024).
- [22] M. H. Abobeih, Y. Wang, J. Randall, S. J. H. Loenen, C. E. Bradley, M. Markham, D. J. Twitchen, B. M. Terhal, and T. H. Taminiau, Fault-tolerant operation of a logical qubit in a diamond quantum processor, *Nature* **606**, 884 (2022).
- [23] D. Bluvstein *et al.*, Logical quantum processor based on reconfigurable atom arrays, *Nature* **626**, 58 (2024).
- [24] B. W. Reichardt, *et al.*, Logical computation demonstrated with a neutral atom quantum processor, *arXiv:2411.11822*.
- [25] M. J. Bedalov *et al.*, Fault-tolerant operation and materials science with neutral atom logical qubits, *arXiv:2412.07670*.
- [26] P. S. Rodriguez *et al.*, Experimental demonstration of logical magic state distillation, *Nature* **645**, 620 (2025).
- [27] D. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, California Institute of Technology (1997).
- [28] T. M. Graham, L. Phuttitarn, R. Chinnarasu, Y. Song, C. Poole, K. Jooya, J. Scott, A. Scott, P. Eichler, and M. Saffman, Midcircuit measurements on a single-species neutral alkali atom quantum processor, *Phys. Rev. X* **13**, 041051 (2023).
- [29] K. Singh, C. E. Bradley, S. Anand, V. Ramesh, R. White, and H. Bernien, Mid-circuit correction of correlated phase errors using an array of spectator qubits, *Science* **380**, 1265 (2023).
- [30] W. Huie, L. Li, N. Chen, X. Hu, Z. Jia, W. K. C. Sun, and J. P. Covey, Repetitive readout and real-time control of nuclear spin qubits in ^{171}Yb atoms, *PRX Quantum* **4**, 030337 (2023).
- [31] J. W. Lis, A. Senoo, W. F. McGrew, F. Rönchen, A. Jenkins, and A. M. Kaufman, Midcircuit operations using the *omg* architecture in neutral atom arrays, *Phys. Rev. X* **13**, 041035 (2023).
- [32] M. A. Norcia *et al.*, Midcircuit qubit measurement and rearrangement in a ^{171}Yb atomic array, *Phys. Rev. X* **13**, 041034 (2023).
- [33] 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).
- [34] C.-K. Li, M. Nakahara, Y.-T. Poon, N.-S. Sze, and H. Tomita, Recovery in quantum error correction for general

- noise without measurement, *Quantum Inf. Comput.* **12**, 149 (2012).
- [35] D. Crow, R. Joynt, and M. Saffman, Improved error thresholds for measurement-free error correction, *Phys. Rev. Lett.* **117**, 130503 (2016).
- [36] H. E. Ercan, J. Ghosh, D. Crow, V. N. Premakumar, R. Joynt, M. Friesen, and S. N. Coppersmith, Measurement-free implementations of small-scale surface codes for quantum-dot qubits, *Phys. Rev. A* **97**, 012318 (2018).
- [37] M. A. Perlin, V. N. Premakumar, J. Wang, M. Saffman, and R. Joynt, Fault-tolerant measurement-free quantum error correction with multiqubit gates, *Phys. Rev. A* **108**, 062426 (2023).
- [38] S. Heußen, D. F. Locher, and M. Müller, Measurement-free fault-tolerant quantum error correction in near-term devices, *PRX Quantum* **5**, 010333 (2024).
- [39] S. Veroni, M. Müller, and G. Giudice, Optimized measurement-free and fault-tolerant quantum error correction for neutral atoms, *Phys. Rev. Res.* **6**, 043253 (2024).
- [40] K. Singh, S. Anand, A. Pocklington, J. T. Kemp, and H. Bernien, Dual-element, two-dimensional atom array with continuous-mode operation, *Phys. Rev. X* **12**, 011040 (2022).
- [41] M. A. Norcia *et al.*, Iterative assembly of ^{171}Yb atom arrays with cavity-enhanced optical lattices, *PRX Quantum* **5**, 030316 (2024).
- [42] F. Gyger, M. Ammenwerth, R. Tao, H. Timme, S. Snigirev, I. Bloch, and J. Zeiher, Continuous operation of large-scale atom arrays in optical lattices, *Phys. Rev. Res.* **6**, 033104 (2024).
- [43] A. M. Steane, Active stabilization, quantum computation, and quantum state synthesis, *Phys. Rev. Lett.* **78**, 2252 (1997).
- [44] B. Eastin and E. Knill, Restrictions on transversal encoded quantum gate sets, *Phys. Rev. Lett.* **102**, 110502 (2009).
- [45] S. Bravyi and R. König, Classification of topologically protected gates for local stabilizer codes, *Phys. Rev. Lett.* **110**, 170503 (2013).
- [46] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, *Phys. Rev. A* **71**, 022316 (2005).
- [47] E. T. Campbell, H. Anwar, and D. E. Browne, Magic-state distillation in all prime dimensions using quantum Reed-Muller codes, *Phys. Rev. X* **2**, 041021 (2012).
- [48] E. T. Campbell and M. Howard, Unified framework for magic state distillation and multiqubit gate synthesis with reduced resource cost, *Phys. Rev. A* **95**, 022316 (2017).
- [49] J. O’Gorman and E. T. Campbell, Quantum computation with realistic magic-state factories, *Phys. Rev. A* **95**, 032338 (2017).
- [50] D. Litinski, Magic state distillation: Not as costly as you think, *Quantum* **3**, 205 (2019).
- [51] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, Fault-tolerant conversion between the Steane and Reed-Muller quantum codes, *Phys. Rev. Lett.* **113**, 080501 (2014).
- [52] M. E. Beverland, A. Kubica, and K. M. Svore, Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes, *PRX Quantum* **2**, 020341 (2021).
- [53] F. Butt, S. Heußen, M. Risppler, and M. Müller, Fault-tolerant code-switching protocols for near-term quantum processors, *PRX Quantum* **5**, 020345 (2024).
- [54] T. J. Yoder, Universal fault-tolerant quantum computation with Bacon-Shor codes, [arXiv:1705.01686](https://arxiv.org/abs/1705.01686).
- [55] T. J. Yoder, R. Takagi, and I. L. Chuang, Universal fault-tolerant gates on concatenated stabilizer codes, *Phys. Rev. X* **6**, 031039 (2016).
- [56] A. Paetznick and B. W. Reichardt, Universal fault-tolerant quantum computation with only transversal gates and error correction, *Phys. Rev. Lett.* **111**, 090505 (2013).
- [57] T. Jochym-O’Connor and R. Laflamme, Using concatenated quantum codes for universal fault-tolerant quantum gates, *Phys. Rev. Lett.* **112**, 010505 (2014).
- [58] C. Chamberland, T. Jochym-O’Connor, and R. Laflamme, Overhead analysis of universal concatenated quantum codes, *Phys. Rev. A* **95**, 022313 (2017).
- [59] F. Butt, D. F. Locher, K. Brechtelsbauer, H. P. Büchler, and M. Müller, Measurement-free, scalable, and fault-tolerant universal quantum computing, *Sci. Adv.* **11**, eadv2590 (2025).
- [60] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, *Phys. Rev. A* **54**, 1098 (1996).
- [61] A. M. Steane, Error correcting codes in quantum theory, *Phys. Rev. Lett.* **77**, 793 (1996).
- [62] D. Bacon, Operator quantum error-correcting subsystems for self-correcting quantum memories, *Phys. Rev. A* **73**, 012340 (2006).
- [63] P. Aliferis and A. W. Cross, Subsystem fault tolerance with the Bacon-Shor code, *Phys. Rev. Lett.* **98**, 220502 (2007).
- [64] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, *Phys. Rev. A* **52**, R2493 (1995).
- [65] M. Li, D. Miller, M. Newman, Y. Wu, and K. R. Brown, 2D compass codes, *Phys. Rev. X* **9**, 021041 (2019).
- [66] M. Vasmer and A. Kubica, Morphing quantum codes, *PRX Quantum* **3**, 030319 (2022).
- [67] SparseStates.jl, GitHub repository, <https://github.com/giacomogiudice/SparseStates.jl>
- [68] S. Jaques and T. Häner, Leveraging state sparsity for more efficient quantum simulations, *ACM Trans. Quantum Comput.* **3**, 1 (2022).
- [69] L. H. Pedersen, N. M. Møller, and K. Mølmer, Fidelity of quantum operations, *Phys. Lett. A* **367**, 47 (2007).
- [70] To reduce the computational cost, for numerical simulations of CCZ_L we exploit the permutational invariance and the local Z_L symmetry, and adjust the weights in Eq. (6) accordingly.
- [71] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).
- [72] V. V. Shende and I. L. Markov, On the cnot-cost of Toffoli gates, *Quantum Inf. Comput.* **9**, 461 (2008).
- [73] E. Knill, R. Laflamme, and W. H. Zurek, Resilient quantum computation, *Science* **279**, 342 (1998).
- [74] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, *SIAM J. Comput.* **38**, 1207 (2008).

- [75] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, *Quant. Inf. Comput.* **6**, 97 (2006).
- [76] D. Poulin, Optimal and efficient decoding of concatenated quantum block codes, *Phys. Rev. A* **74**, 052333 (2006).
- [77] F. Battistel, C. Chamberland, K. Johar, R. W. J. Overwater, F. Sebastiano, L. Skoric, Y. Ueno, and M. Usman, Real-time decoding for fault-tolerant quantum computing: Progress, challenges and outlook, *Nano Futures* **7**, 032003 (2023).
- [78] L. Skoric, D. E. Browne, K. M. Barnes, N. I. Gillespie, and E. T. Campbell, Parallel window decoding enables scalable fault tolerant quantum computation, *Nat. Commun.* **14**, 7040 (2023).
- [79] N. Liyanage, Y. Wu, S. Tagare, and L. Zhong, FPGA-based distributed union-find decoder for surface codes, *IEEE Trans. Quantum Eng.* **5**, 3103318 (2024).
- [80] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, *Quantum* **5**, 595 (2021).
- [81] C. Gidney, STIM: A fast stabilizer circuit simulator, *Quantum* **5**, 497 (2021).
- [82] S. Veroni, A. Paler, and G. Giudice, Data for “universal quantum computation via scalable measurement-free error correction” [data set] (2025), <https://zenodo.org/records/15707012>
- [83] S. A. Moses *et al.*, A race-track trapped-ion quantum processor, *Phys. Rev. X* **13**, 041052 (2023).
- [84] C. M. Löschner, J. M. Toba, A. C. Hughes, S. A. King, M. A. Weber, R. Srinivas, R. Matt, R. Nourshargh, D. T. C. Allcock, C. J. Ballance, C. Matthiesen, M. Malinowski, and T. P. Harty, Scalable, high-fidelity all-electronic control of trapped-ion qubits, *PRX Quantum* **6**, 040313 (2025).
- [85] J.-S. Chen, E. Nielsen, M. Ebert, V. Inlek, K. Wright, V. Chaplin, A. Maksymov, E. Pérez, A. Poudel, P. Maunz, and J. Gamble, Benchmarking a trapped-ion quantum computer with 30 qubits, *Quantum* **8**, 1516 (2024).
- [86] S. Ma, A. P. Burgers, G. Liu, J. Wilson, B. Zhang, and J. D. Thompson, Universal gate operations on nuclear spin qubits in an optical tweezer array of ^{171}Yb atoms, *Phys. Rev. X* **12**, 021028 (2022).
- [87] S. J. Evered, D. Bluvstein, M. Kalinowski, S. Ebadi, T. Manovitz, H. Zhou, S. H. Li, A. A. Geim, T. T. Wang, N. Maskara, H. Levine, G. Semeghini, M. Greiner, V. Vuletić, and M. D. Lukin, High-fidelity parallel entangling gates on a neutral-atom quantum computer, *Nature* **622**, 268 (2023).
- [88] R. Finkelstein, R. B.-S. Tsai, X. Sun, P. Scholl, S. Direkci, T. Gefen, J. Choi, A. L. Shaw, and M. Endres, Universal quantum operations and ancilla-based read-out for tweezer clocks, *Nature* **634**, 321 (2024).
- [89] R. B.-S. Tsai, X. Sun, A. L. Shaw, R. Finkelstein, and M. Endres, Benchmarking and fidelity response theory of high-fidelity Rydberg entangling gates, *PRX Quantum* **6**, 010331 (2025).
- [90] J. A. Muniz *et al.*, High-fidelity universal gates in the ^{171}Yb ground-state nuclear-spin qubit, *PRX Quantum* **6**, 020334 (2025).
- [91] J. Preskill, Beyond NISQ: The Megaquop Machine, *ACM Trans. Quantum Comput.* **6**, 3723153 (2025).
- [92] C. Vuillot, L. Lao, B. Criger, C. García Almudéver, K. Bertels, and B. M. Terhal, Code deformation and lattice surgery are gauge fixing, *New J. Phys.* **21**, 033028 (2019).
- [93] 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 (2022).
- [94] S. Hwang, H. Hwang, K. Kim, A. Byun, K. Kim, S. Jeong, M. P. Soegianto, and J. Ahn, Fast and reliable atom transport by optical tweezers, *Optica Quantum* **3**, 64 (2025).
- [95] A. G. Radnaev *et al.*, Universal neutral-atom quantum computer with individual optical addressing and nondestructive readout, *PRX Quantum* **6**, 030334 (2025).
- [96] L. Schmid, D. F. Locher, M. Rispler, S. Blatt, J. Zeiher, M. Müller, and R. Wille, Computational capabilities and compiler development for neutral atom quantum processors—connecting tool developers and hardware experts, *Quantum Sci. Technol.* **9**, 033001 (2024).
- [97] N. Constantinides, A. Fahimniya, D. Devulapalli, D. Bluvstein, M. J. Gullans, J. V. Porto, A. M. Childs, and A. V. Gorshkov, Optimal routing protocols for reconfigurable atom arrays, *arXiv:2411.05061*.
- [98] H. Zhou, *et al.*, Low-overhead transversal fault tolerance for universal quantum computation, *Nature* **646**, 303 (2025).
- [99] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, *Quantum Algorithms: A Survey of Applications and End-to-end Complexities* (Cambridge University Press, Cambridge, 2025).