npj | quantum information

Article

Published in partnership with The University of New South Wales

https://doi.org/10.1038/s41534-026-01362-4

Unitary fault-tolerant encoding of Pauli states in surface codes

Check for updates

Luis Colmenarez[1,2] ✉, Remmy Zen[3,4], Jan Olle[3,5], Florian Marquardt[3,6] & Markus Müller[1,2]

In fault-tolerant quantum computation, the preparation of logical states is a ubiquitous subroutine, yet significant challenges persist even for the simplest states required. In the present work, we present a unitary, scalable, distance-preserving encoding scheme for preparing Pauli eigenstates in surface codes. Unlike previous unitary approaches whose fault-distance remains constant with increasing code distance, our scheme ensures that the protection offered by the code is preserved during state preparation. Building on strategies discovered by reinforcement learning for the surface-17 code, we generalize the construction to arbitrary code distances and both rotated and unrotated surface codes. The proposed encoding relies only on geometrically local gates, and is therefore fully compatible with planar 2D qubit connectivity, and it achieves circuit depth scaling as $O(d)$, consistent with fundamental entanglement-generation bounds. We design explicit stabilizer-expanding circuits with and without ancilla-mediated connectivity and analyze their error-propagation behavior. Numerical simulations under depolarizing noise show that our unitary encoding without ancillas outperforms standard stabilizer-measurement-based schemes, reducing logical error rates by up to an order of magnitude. These results make the scheme particularly relevant for platforms such as trapped ions and neutral atoms, where measurements are costly relative to gates and idling noise is considerably weaker than gate noise. Our work bridges the gap between measurement-based and unitary encodings of surface-code states and opens new directions for distance-preserving state preparation in fault-tolerant quantum computation.

Quantum error correction (QEC) promises to enable fault-tolerant (FT) quantum computation by correcting errors faster than they accumulate. QEC achieves this by introducing redundancy through the use of additional qubits, which allows for the detection and correction of errors. Consequently, many physical qubits are used to encode one or more logical qubits, which are the carriers of quantum information in FT quantum algorithms. Recent experiments across several platforms-including superconducting qubits[1–3], cat qubits[4], trapped ions[5,6], and neutral atoms[7–9]-have demonstrated the successful implementation of key QEC protocols, along with the expected suppression of errors in the low-noise regime[3,10]. Today, a leading contender for practical FT QEC are topological codes[11]. Among them, the surface code[11–14] is the most widely studied and experimentally pursued one, owing to its high threshold under circuit-level noise, its compatibility with a nearest neighbor layout in 2D, and its well-understood framework for scalable fault-tolerance. Several critical components for FT quantum computation are already well established within surface code architectures. Decoding can be performed efficiently using the minimum weight perfect matching (MWPM) algorithm[12,15], while maintaining a relatively high error threshold[14]. Logical CNOT gates can be implemented transversally, a strategy already demonstrated in neutral atom platforms[10], or via lattice surgery[16], a method especially relevant when qubit interactions are restricted to nearest neighbors, as is the case in superconducting qubit architectures[3,17,18]. Single-qubit logical gates can be executed through fold-transversal gate constructions[19–21], or by employing magic state distillation and injection[7,22,23]. More recently, magic state cultivation has emerged as a promising alternative[24,25].

One fundamental operation on logical qubits is state preparation and encoding[26]. This task involves preparing a desired logical state through the application of gates and measurements on bare physical qubits. As any protocol involving logical qubits begins with the preparation of one or more

[1]Institute for Theoretical Nanoelectronics (PGI-2), Forschungszentrum Jülich, Jülich, Germany. [2]Institute for Quantum Information, RWTH Aachen University, Aachen, Germany. [3]Max Planck Institute for the Science of Light, Erlangen, Germany. [4]School of Physics and Astronomy, Monash University, Clayton, VIC, Australia. [5]NVIDIA Corporation, Santa Clara, CA, USA. [6]Department of Physics, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany. ✉e-mail: colmenarez@physik.rwth-aachen.de

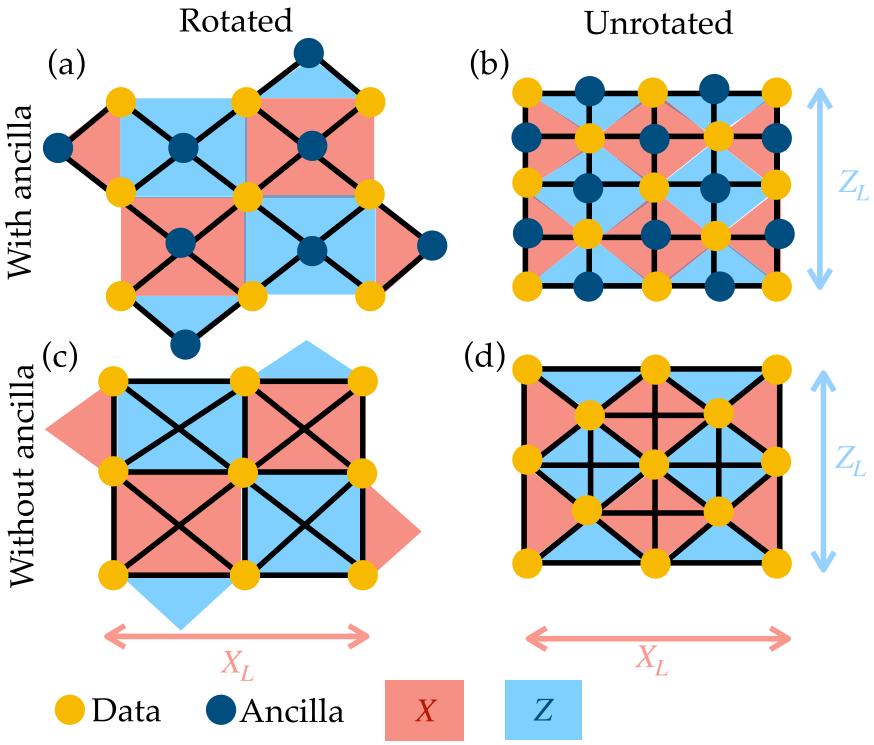

Fig. 1 | Surface codes and connectivity layout. **a**, **c** Rotated surface code of code distance $d = 3$. **b**, **d** Unrotated surface code of distance $d = 3$. Red and blue denote X and Z stabilizers, respectively. Logical operators Z_L and X_L are defined along the vertical and horizontal boundary, respectively. Yellow and dark blue dots represent data and ancilla qubits, respectively. The black lines represent the qubit connectivity.

logical states, this operation is ubiquitous across all QEC schemes and quantum algorithms. However, preparing logical states is particularly challenging due to the complex many-body entanglement structure inherent to these states, especially when device-specific constraints are taken into account. For example, in topological QEC codes, encoding logical states is equivalent to preparing topologically ordered states[27–29]. Consequently, any unitary circuit that prepares logical states in surface codes must respect limits on how quickly entanglement can be generated, as dictated by Lieb-Robinson-type bounds[30]. Specifically, when only geometrically local gates are allowed, the circuit depth for encoding scales as $\mathcal{O}(d)$, where d is both the code distance and the linear size of the lattice[31–33]. In contrast, if all-to-all connectivity is available, the depth can be reduced to $\mathcal{O}(\log d)$[27,28,34,35]. In contrast, measurement-based schemes can generate entanglement more rapidly, enabling logical state preparation in constant time[12,36,37].

Another important factor to consider is the resilience of encoding schemes to errors occurring during gate execution and measurements. Although these errors are initially local, they can propagate rapidly as more entangling gates are applied[12]. This vulnerability is especially pronounced in previously proposed unitary encoding circuits for surface codes[27,28,33–35]. In such schemes, the *fault-distance*, defined as the minimum number of faults required to produce an undetectable logical error, remains constant regardless of the *code distanced*. This limitation arises from the uncontrolled spread of errors during circuit execution. To date, the only known *distance-preserving* scalable encoding scheme is the one in which qubits are initialized on a product state, e.g. $|0\rangle$, and stabilizers of the opposite type, e.g. $X-$ type, are measured at least once[10], under the assumption that the stabilizer measurement circuits themselves are FT. This method ensures that the fault-distance is equal to the code distance, preserving during state preparation the protection offered by the QEC code. Some encoding methods rely on state verification using flag qubits[38–40]. While this can make the encoding distance-preserving, such schemes are difficult to scale beyond small distances. It is important to note that when measurements and classical feed-forward are allowed the circuit depth to prepare a fully FT state can be constant[12,36,37,41]. In this work we focus on encoding schemes that do not rely on classical feed-forward, e.g. decoding, to achieve fault-tolerance or distance preservation.

In this work, we propose, for the first time, a unitary scalable encoding for the surface code that is distance-preserving, inspired by our recent results in ref. 42, where a reinforcement learning agent was used to find a FT encoding circuit for the surface-17 code[17,43,44]. Interestingly, the agent discovered a novel strategy that was previously unknown to human experts. This further illustrates how AI can provide new insights for solving complex quantum tasks. Here, we generalize the strategy that the reinforcement learning agent discovered to arbitrary code distances and to the preparation of Pauli eigenstates $|0\rangle_L$ and $|+\rangle_L$ for both rotated[44] and unrotated surface codes[12]. We assume a standard square-grid architecture of data and ancilla qubits, typical of superconducting qubit arrays[3,17], where each stabilizer is associated with an ancilla placed at its center. We design unitary encoding circuits where two-qubit gates are applied either between data and ancilla qubits (see Fig. 1a, b) or, in a variant of this scheme, between data qubits within the same plaquette (see Fig. 1c, d). The latter is feasible in architectures with non-static, i.e., dynamically reconfigurable qubit connectivity, such as neutral atoms[10] and trapped ions[45]. We have tested our circuits on both rotated and unrotated surface codes in the presence of gate errors and find very good performance when compared to logical state preparation based on stabilizer measurement[3,12,17,42]. Importantly, the proposed encoding scheme remains valid in any implementation of the surface code where hook errors can be oriented orthogonally to the desired logical operator and thus remain correctable[44,46,47]. Furthermore, for QEC codes in which certain classes of high-weight errors remain correctable, our encoding scheme can be adapted to engineer hook errors so that they fall within these correctable error classes. As expected for unitary encoding schemes constrained to local interactions, the circuit depth scales as $\mathcal{O}(d)$, and is therefore deeper than measurement-based encoding circuits, which can achieve constant depth $\mathcal{O}(1)$. However, because measurements are typically much slower than gates, our unitary encoding scheme may actually be *faster* than measurement-based encoding, thereby reducing the overall idling time. For example, in superconducting qubits[3,17], measurements can take up to five times longer than gates. For neutral atoms, the difference can be up to two orders of magnitude[48,49]. Thus, reducing measurements in favor of gate-based operations can significantly shorten the overall QEC protocol and reduce exposure to decoherence. Furthermore, our scheme is especially

relevant for quantum platforms with long coherence times, such as certain trapped ions[50,51] and neutral atom arrays[49], where idling errors are negligible, and measurements are preferably deferred to later stages of computation. Thus, implementing our encoding scheme offers a promising path to producing high-quality Pauli states in near-term quantum processors. Furthermore, our unitary encoding scheme is fully compatible with the paradigm of algorithmic fault-tolerance[52], making it a promising building block for FT quantum computing architectures that minimize measurement overhead.

The structure of the paper is as follows. In "Background", we introduce the basic concepts used in our work. In "Unitary fault-tolerant encoding", we describe in detail the unitary encoding we develop. In "Numerical simulations", we benchmark our unitary encoding scheme against the standard measurement-based encoding. Finally, in "Conclusions", we present our concluding remarks.

Background

Surface code

The surface code is a planar topological code usually defined on a square lattice[12,44]. It is characterized by a set of X- and Z-type stabilizer checks, also referred to as stars and plaquettes, respectively. For simplicity, we assume throughout that all stabilizer operators have eigenvalue $+1$, although this condition can be relaxed as long as the measured eigenvalue is known; see the discussion on measurement-based encoding in Sec. II B. The logical operators X_L and Z_L correspond to string operators that traverse the lattice in the horizontal and vertical directions, respectively (see Fig. 1). In this work, we focus on the preparation of logical states that are eigenstates of the corresponding logical operators, satisfying $Z_L|0\rangle_L = |0\rangle_L$ and $X_L|+\rangle_L = |+\rangle_L$. Importantly, any implementation of the surface code supports efficient decoding using MWPM[12,15].

The most widely used realization of the surface code is the $[[d^2, 1, d]]$ *rotated* surface code[3,17,18,44], where d denotes both the code distance and the linear size of the lattice. Its main advantage over other realizations is its efficient use of physical qubits, requiring fewer qubits to achieve a given code distance. The original formulation of the planar surface code, however, is the *unrotated* surface code[12], which has code parameters $[[d^2 + (d-1)^2, 1, d]]$. This version uses roughly twice as many physical qubits as the rotated code to achieve the same code distance. Nonetheless, recent studies have shown that the unrotated surface code can be more resilient to hook errors in syndrome measurement circuits[12,53] when implementing syndrome readout circuits via three-qubit gates[46,47]. These findings motivate a renewed interest in the unrotated surface code, especially in the context of studying stabilizer-related circuits. In Fig. 1, we illustrate both the $d = 3$ rotated and unrotated surface codes for comparison.

Encoding of Pauli states

As a topological code defined on a d-dimensional lattice, a surface (or toric) code state can be prepared by a geometrically local unitary circuit of depth $\mathcal{O}(d)$[29,31,32]. If all-to-all connectivity is available, the circuit depth can be reduced to $\mathcal{O}(\log d)$[27,28,34,35]. Several constructions of unitary encoding circuits for surface codes exist. One general approach exploits the equivalence between graph states and stabilizer states[27], enabling a method in which each stabilizer is prepared as a GHZ state. This recipe applies to arbitrary stabilizer codes and does not rely on local connectivity[54]. Other constructions are based on renormalization group techniques, where the typical strategy is to grow a surface code patch from a smaller distance patch[31,35]. However, all of the aforementioned schemes are limited in their treatment of fault-tolerance and error proliferation. In particular, the encoding schemes proposed in Refs. 27,28,31,35 exhibit fault-distances that are lower than the code distance. As a result, the surface code states prepared by these circuits can introduce high-weight logical errors when used as components in logical quantum algorithms. To the best of our knowledge, our work is the first to demonstrate a distance-preserving unitary and deterministic encoding scheme for surface codes. This scheme assumes only local gate connectivity and achieves circuit depth $\mathcal{O}(d)$. Furthermore, it relies solely on the ability to orient hook errors orthogonal to the desired logical operators, making it applicable to any variant of the surface code. We design explicit encoding circuits that consider gate connectivity within each stabilizer plaquette, both with and without the use of ancilla qubits assigned to measure stabilizers.

It must be noted that our encoding scheme preserves the fault-distance only for the type of error that flips the corresponding codeword, for example, X errors when preparing the logical state $|0\rangle_L$. Therefore, a full round of QEC-namely, a fault-tolerant syndrome extraction using repeated stabilizer measurements (or Steane-type syndrome extraction[55]), followed by correction, is required to remove potential harmful errors of the complementary type. Further details are provided in Supplementary Information. This places our unitary encoding circuit on equal footing with the standard measurement-based encoding scheme, in which $|0\rangle_L$ ($|+\rangle_L$) is prepared by initializing all data qubits in the product state $|0\rangle^{\otimes n}$ ($|+\rangle^{\otimes n}$), followed by a single round of X- (Z-) stabilizer measurements. After this procedure, the Z (X) stabilizers are deterministically projected to the $+1$ eigenstate, allowing X (Z) errors to be corrected via a round of QEC; that is, by executing FT syndrome extractions and applying the appropriate correction. However, because the outcomes of the X (Z) stabilizer measurements are both random and potentially faulty, a single round of stabilizer measurements does not provide sufficient information to accurately determine the syndrome. This leads to a high chance of introducing a logical Z_L (X_L) error, even after an ideal QEC round. Fortunately, this is not a problem since we are preparing an eigenstate of precisely that logical operator, namely $Z_L|0\rangle_L = |0\rangle_L$ ($X_L|+\rangle_L = |+\rangle_L$). Therefore, we choose to compare our unitary encoding scheme directly against the measurement-based one, given their similar properties regarding fault tolerance.

In summary, achieving a distance-preserving encoding for both types of errors, X and Z, requires the execution of a full QEC round following the preparation of the logical state by any of the methods described in this work. Prior to the QEC round, the logical state is FT only with respect to a single type of error, either X or Z. Therefore, applying FT gadgets, e.g., transversal logical CNOTs, *before* this respective QEC round, may end up spreading high-weight errors to other logical qubits. A way to deal with such high-weight errors is to correct multiple logical qubits jointly[52,56]. However, in this work we consider the standard scenario of correcting each logical qubit independently.

Unitary fault-tolerant encoding

In this section, we generalize the unitary fault-tolerant (FT) scheme discovered for $d = 3$ rotated surface code by a reinforcement learning agent in the previous work[42] in the context of flag fault-tolerance. The fundamental building blocks are the stabilizer-expanding circuits shown in Fig. 2 for the X-type stabilizers. The corresponding Z-type stabilizers are obtained by reversing the orientation of the CNOT gates and appropriately initializing the data qubits; see Supplementary Information, Sec. A for details. In practice, these circuits operate by selecting a *pivot* qubit and transforming it into the desired stabilizer, following a construction analogous to the one used to prepare graph states [27,57,58], therefore, full connectivity among all data qubits within a stabilizer plaquette is required. However, we propose an equivalent circuit in which the plaquette qubits are connected only to a common ancilla qubit, specifically the one assigned to measure the corresponding stabilizer during subsequent QEC cycles. Importantly, the error propagation behavior in stabilizer-expanding circuits with or without ancillas is similar. In both cases, a single hook error can potentially compromise fault tolerance, as illustrated in Fig. 2. This allows us to use either circuit architecture interchangeably, depending on the specific qubit connectivity available within each plaquette. Importantly, the circuits shown in Fig. 2 do not involve any measurements; the ancilla qubit used in the construction does not perform a measurement on the newly prepared state. Instead, the ancilla serves solely as a bridge between data qubits, as discussed in ref. 42.

We now describe in detail the preparation of the logical state $|0\rangle_L$ in the rotated surface code. The encoding procedure for $|+\rangle_L$ is analogous and discussed in Supplementary Information. The explicit protocol for

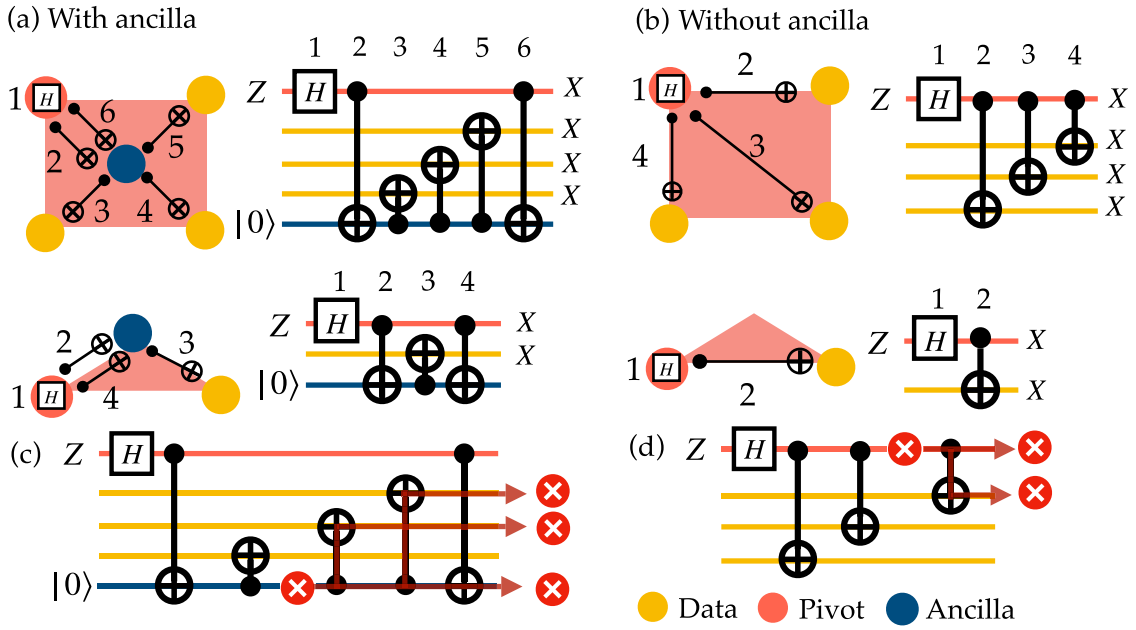

Fig. 2 | Circuits for expanding the *X* stabilizers. a, **b** Circuit for coherently initializing an *X* plaquette with and without ancilla in the middle, respectively. Each circuit assumes that the pivot qubit (positioned at the top of each diagram) is initialized in the state $|+\rangle$ (shown as a Hadamard gate in this depiction). The circuit then transforms the pivot qubit into an *X*-type stabilizer. This procedure is general and applies to stabilizers of any weight, although only weight-two and weight-four examples are shown. Notably, the construction requires at most one ancilla qubit-if any-independent of the stabilizer weight. Numbers indicate the order of the gates to avoid proliferation of dangerous hook errors. **c**, **d** Show the potential hook errors for the circuit realization with and without ancilla, respectively. In addition, several potentially dangerous hook *Z* errors are discussed in detail in Supplementary Information.

initializing $|0\rangle_L$ in the $d = 5$ rotated surface code using plaquette ancillas is shown in Fig. 3. First, all data and ancilla qubits are initialized in the state $|0\rangle^{\otimes n}$. Then, the *X*-type stabilizers are expanded row by row, with each stabilizer prepared using the circuits from Fig. 2. For the first row, the choice of pivot qubits is arbitrary. However, in subsequent rows, the pivot qubits must differ from those used in the previous row. A crucial constraint is that any CNOT gate applied to a qubit already involved in a previously prepared stabilizer must use that qubit only as a target, never as a control. The latter condition ensures that hook errors do not propagate into larger correlated errors, which could degrade the code's fault distance. As a result, stabilizers are prepared serially along each of the $d - 1$ rows (or columns), leading to a circuit depth proportional to *d*. In Fig. 4, we also show the encoding circuit under the assumption of full connectivity among data qubits within each plaquette. This version results in circuits with shorter depth due to the reduced number of gates required per stabilizer, i.e., three two-qubit gates instead of five for the four-qubit plaquettes. The same procedure applies to the unrotated surface code, as discussed in Supplementary Information.

Some remarks regarding the circuits shown in Figs. 3 and 4 are in order. First, the gate schedule depicted is not unique; further parallelization across different rows is feasible. Namely, some gates in row $d + 1$ can already be applied when some plaquettes in row *d* are still pending. Nonetheless, the overall circuit depth remains proportional to *d*. The schedule presented here is chosen to make the row-by-row structure of the operations explicit and to clearly illustrate the precise gate sequencing on each plaquette. Second, the encoding is *distance-preserving*, meaning the fault distance matches the code distance when preparing $|0\rangle_L$ ($|+\rangle_L$) and correcting *X* (*Z*) errors. However, when considering the complementary type of error-for example, *Z* errors during the preparation of $|0\rangle_L$-the effective fault distance is bounded. The same applies to the measurement encoding as well due to the presence of measurement errors[10]. Distance preserving and fault-tolerance of the encoding is further discussed in Supplementary Information. Such a scenario arises, for instance, when preparing logical Bell pairs using transversal CNOTs[10] or when initializing logical ancilla qubits for Steane or Knill-type error correction protocols[55,59,60].

Numerical simulations

To evaluate the performance of our encoding scheme, we model imperfect gates using depolarizing noise channels applied immediately after each gate operation. Specifically, two-qubit gates are followed by a two-qubit depolarizing channel,

$$\mathcal{N}_2(\rho) = (1-p)\rho + \frac{p}{15}\sum_P P\rho P, \tag{1}$$

where $P \in \{\sigma \otimes \sigma\} \backslash \{I \otimes I\}$ and $\sigma = \{I, X, Y, Z\}$ denotes the set of Pauli matrices. Preparation of the $|0\rangle$ and $|+\rangle$ single qubit states is performed via reset operations. Each preparation of the $|0\rangle$ state is followed by an error channel of the form:

$$\mathcal{N}_1^Z(\rho) = (1-p)\rho + pX\rho X. \tag{2}$$

Analogously, the preparation of the state $|+\rangle$ is followed by the error channel

$$\mathcal{N}_1^X(\rho) = (1-p)\rho + pZ\rho Z. \tag{3}$$

We benchmark our encoding scheme against the standard stabilizer measurement initialization scheme, which prepares the logical state $|0\rangle_L$ (or $|+\rangle_L$) by first initializing all physical qubits in the product state $|0\rangle^{\otimes n}$ (or $|+\rangle^{\otimes n}$), followed by a single round of *X*- (or *Z*-) stabilizer measurements. In the absence of noise, this procedure yields a state with all *Z*- (or *X*-) stabilizers equal to $+1$, while the conjugate stabilizers have a random value. Since error correction for $|0\rangle_L$ (or $|+\rangle_L$) relies only on the measurement of *Z*- (or *X*-) stabilizers, a single round of fault-tolerant stabilizer measurements suffices to prepare a fault-tolerant logical state. However, when logical operations are subsequently applied, a full quantum error correction (QEC) cycle comprising *d* rounds of stabilizer measurements is required to maintain fault tolerance. In this work, we focus solely on the quality of the prepared Pauli eigenstates $|0\rangle_L$ and $|+\rangle_L$, and therefore limit our analysis to a single round of noisy stabilizer

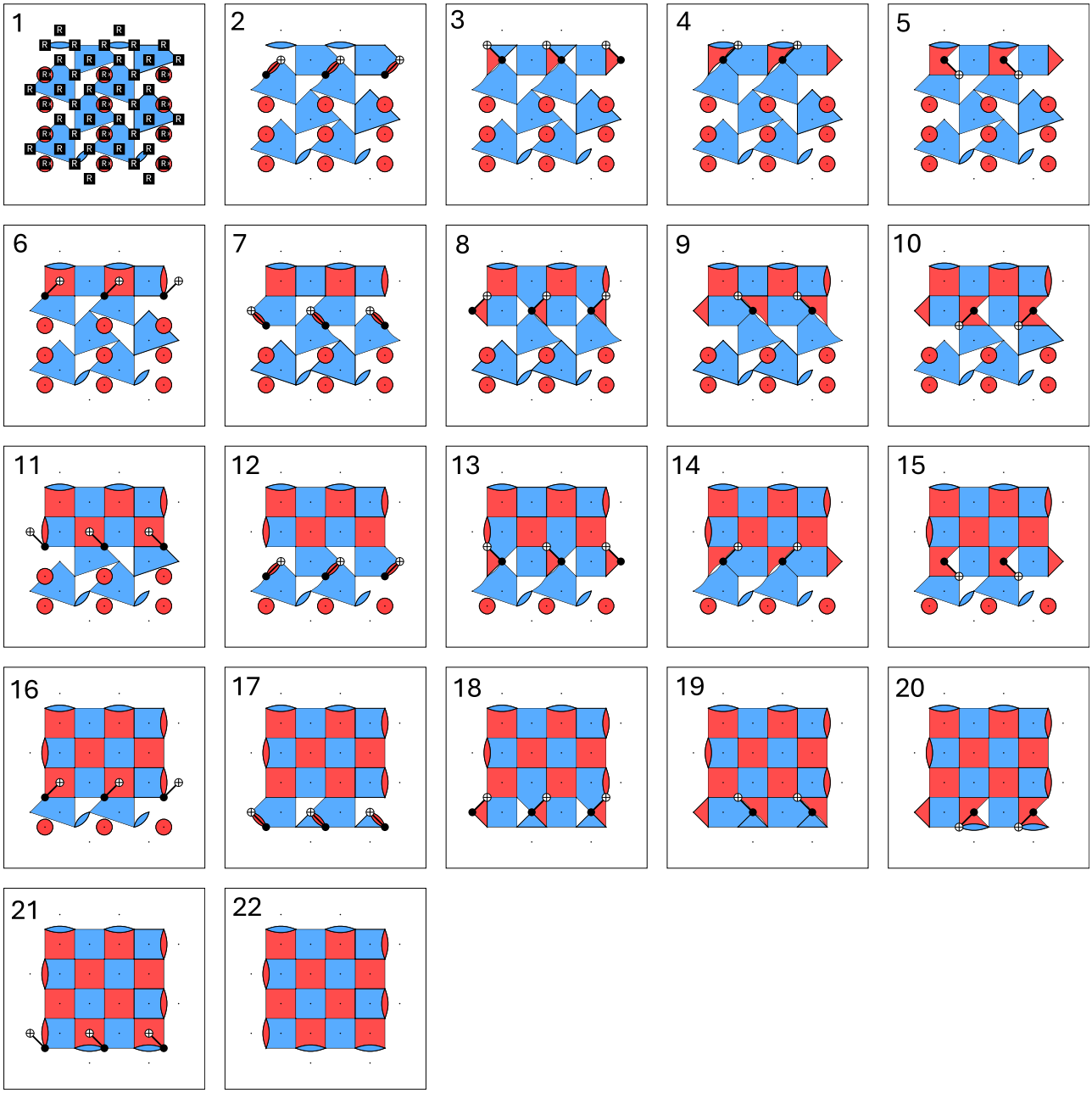

Fig. 3 | Unitary FT encoding for the $d = 5$ rotated surface code using ancillas as bridge qubits. In this layout, blue (red) denotes the Z (X) stabilizers and the logical operators X_L and Z_L lie along the horizontal and vertical directions, respectively. Step 1 initializes the data and ancilla qubits in the states $|0\rangle$ except the pivot qubits that are initialized in the state $|+\rangle$. The blue irregular regions at this stage denote possible groupings of $|0\rangle$ qubits that will form the standard Z stabilizers at the end of the encoding procedure. Steps 2 through 6 prepare the X-type stabilizers on the topmost row. Each plaquette within the row can be prepared in parallel. Steps 7 through 11 perform the preparation of X stabilizers on the second row. Steps 12 through 16 prepare the X stabilizers on the third row. Here, the location of the weight-two X stabilizer alternates between the left and right boundaries, such that the gate scheduling appears the same only every second row. Finally, steps 17 through 21 complete the preparation of the X stabilizers on the bottom row. Step 22 depicts the whole set of stabilizers after the last gate is applied. At the end of the circuit, only correctable weight-two X errors remain. However, certain hook Z errors may still be potentially harmful; these are discussed in detail in Supplementary Information.

measurements. Given that measuring a weight-four (or weight-two) stabilizer requires four (or two) two-qubit gates, the total number of two-qubit gates needed for a single round is $2d(d-1)$ for the rotated surface code; see Table 1.

We simulate three types of surface code encoding circuits: (i) measurement-based encoding (ME), (ii) unitary encoding with ancilla (UEA), and (iii) unitary encoding without ancilla (UE), for both the rotated and unrotated surface code. To evaluate the quality of the prepared logical states, each encoding circuit is followed by a perfect (noiseless) measurement of the data qubits and subsequent syndrome reconstruction and correction. This allows us to isolate the performance of the encoding circuit itself. First, we look at the gate count and circuit depth, see Table 1. Every encoding scheme shown has a different number of entangling gates per stabilizer: 4 for ME, 5 for UEA, and 3 for UE (in the case of 4-qubit plaquettes). The circuit depth is the same for both realizations of the surface codes and unitary encodings, however, the gate count of the rotated surface code is smaller due to the reduced number of stabilizers. Importantly, the gate count of the UE encoding is always smaller than the one of the ME encoding for $d > 1$. Second, in Fig. 5, we compare the ME and UEA schemes for the preparation of the logical state $|0\rangle_L$. For both rotated and unrotated

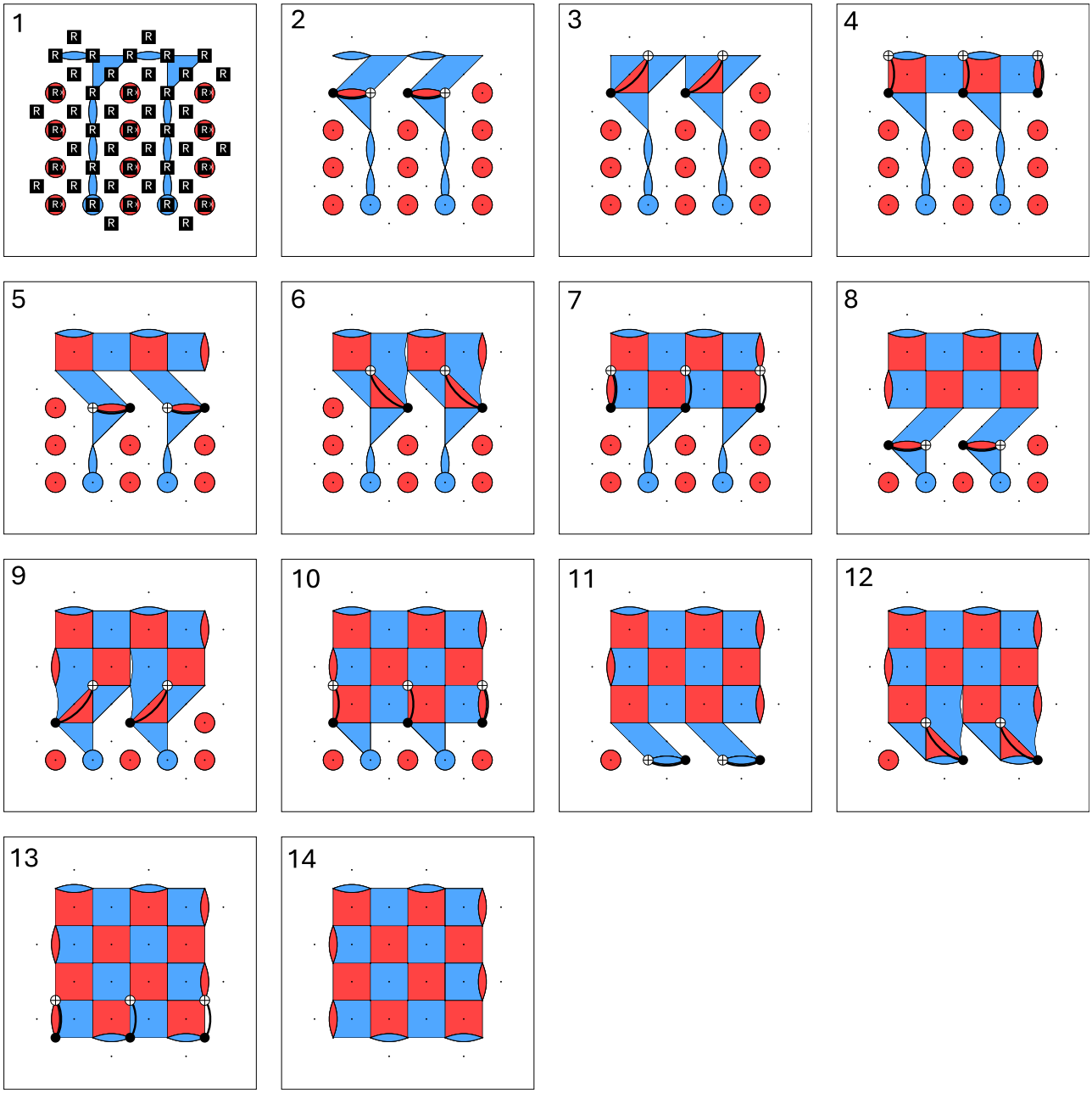

Fig. 4 | Unitary FT encoding circuit for the $d = 5$ rotated surface code with full intra-plaquette connectivity. This encoding circuit assumes full connectivity among data qubits within each plaquette, eliminating the need for ancilla-mediated interactions. As a result, the total circuit depth is reduced to 13 time steps, compared to the 21 steps required in the ancilla-based version shown in Fig. 3. This reduction is due to the simplification of the stabilizer-expanding circuits, which require two fewer gates per stabilizer, as illustrated in Fig. 2.

Table 1 | Two-qubit gate count and circuit depth of the unitary encoding with ancilla (UEA) and without (UE) for the rotated (Rot.) and unrotated (UnRot.) surface code for odd code distance *d*

	Gate count	Circuit depth	Measurements
Rot. UEA	$(5t+3)(d-1)$	$5(d-1)$	No
Rot. UE	$(3t+1)(d-1)$	$3(d-1)$	No
Rot. ME	$2d(d-1)$	4	Yes
UnRot. UEA	$(5d-2)(d-1)$	$5(d-1)$	No
UnRot. UE	$(3d-2)(d-1)$	$3(d-1)$	No
UnRot. ME	$2(2d-1)(d-1)$	4	Yes

Here we use $t = \lfloor (d-1)/2 \rfloor$. Gate count and circuit depth of measurement encoding (ME) are also shown.

surface code, the ME scheme consistently yields a lower logical error rate than the UEA scheme, which can be attributed to its smaller number of gates. For example, for $d = 5$ the ME and UEA encoding protocols for the rotated surface code need 40 and 52 gates, respectively. We also compare the ME and UE schemes. Here, the UE circuit outperforms ME for both the rotated and unrotated surface code, as expected given its lower gate count. For example, for $d = 5$ the unrotated surface the ME and UE encoding schemes need 72 and 44 gates, respectively. In some instances, we observe a reduction in logical error rate by up to an order of magnitude. Similar trends are observed for the preparation of the $|+\rangle_L$ state, as discussed in Supplementary Information. Since error propagation in each type of circuit is similar, specifically, hook errors are always kept under control, the differences in performance can be attributed mostly to the number of error locations in each circuit. Hence, avoiding the use of

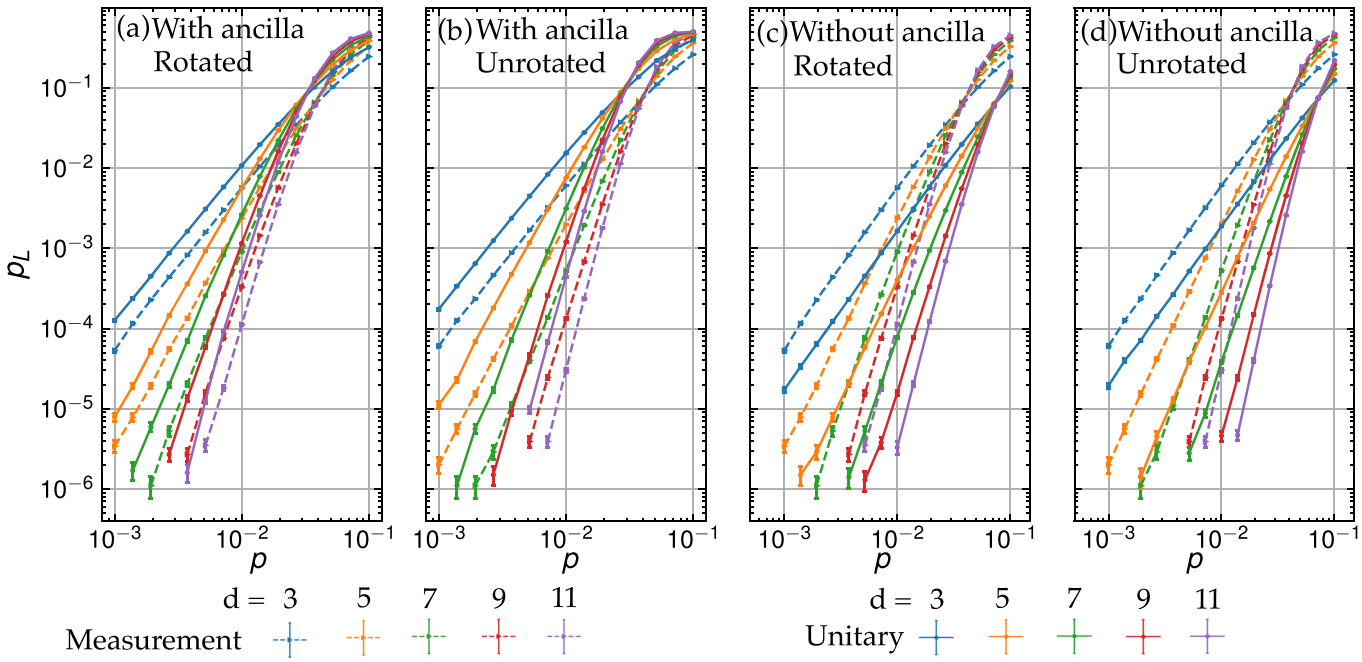

Fig. 5 | Performance of unitary and measurement encodings under circuit level noise. Logical error rate pL for preparing $|0\rangle_L$ for the unitary encoding schemes with **a**, **b**, and without **c**, **d**, and ancilla for the rotated and unrotated surface code for $d = 3, 5, 7, 9, 11$. p is the physical error rate of gates and resets. Dashed lines are the measurement encoding (ME), while solid lines show different unitary encodings. For both types of codes, the encodings without ancillas perform better than the respective measurement encoding.

ancilla qubits considerably reduces the number of error locations, since ancilla qubits are themselves noisy and can introduce additional errors into the data qubits.

Conclusions

We have presented a unitary encoding scheme for the surface code that is distance-preserving and requires only local connectivity. The circuit depth is $\mathcal{O}(d)$, as expected from fundamental bounds on entanglement generation in topological codes. Our encoding scheme does not rely on measurements and classical feedback, making it particularly suitable for trapped-ion and neutral-atom platforms, where idling noise can be negligible, and measurements are often considerably more noisy and slower than gates. The measurement-free encoding without ancillas outperforms the standard stabilizer-based encoding due to the reduced number of entangling gates per stabilizer and the absence of errors propagating from the ancilla qubits to the data qubits. In contrast, the unitary encoding with ancilla qubits performs worse. However, in a more realistic scenario that includes idling errors, replacing measurements with gates can reduce the overall protocol duration and the accumulation of idling errors, since measurements are usually considerably slower than gates.

Our work bridges the gap between unitary and non-unitary encodings of surface-code Pauli states. However, some important questions remain open. For instance, is it possible to extend the present encoding scheme to all-to-all connectivity and thereby further reduce the circuit depth? Since our encoding strategy relies on avoiding hook errors, it is not clear how to extend it in a distance-preserving manner in the presence of long-range gates. Nonetheless, this does not exclude the possibility that a different strategy for unitary encoding could be designed for all-to-all connectivity and achieve fault tolerance.

Another open question is whether there exists a unitary encoding that simultaneously mitigates both types of hook errors. Such an encoding would be highly desirable for Steane-type error correction and for quantum algorithms supporting transversal logical gates.

Data availability

The Stim circuits for the simulations can be found in the Zenodo repository https://doi.org/10.5281/zenodo.18014961.

Received: 4 February 2026; Accepted: 13 August 2026;
Published online: 01 October 2026

References

1. Besedin, I. et al. Lattice surgery realized on two distance-three repetition codes with superconducting qubits. *Nat. Phys.*1–6 https://www.nature.com/articles/s41567-025-03090-6. (2026).
2. Huang, W. et al. Logical multi-qubit entanglement with dual-rail superconducting qubits. *Nat. Phys.* **22**, 591–597 (2026).
3. Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature* **638**, 920–926 (2025).
4. Putterman, H. et al. Hardware-efficient quantum error correction via concatenated bosonic qubits. *Nature* **638**, 927–934 (2025).
5. Pogorelov, I. et al. Experimental fault-tolerant code switching. *Nat. Phys.* **21**, 298–303 (2025).
6. Yamamoto, K. et al. Quantum error-corrected computation of molecular energies. PRX Quantum **7**, 020319 (2026).
7. Sales Rodriguez, P. et al. Experimental demonstration of logical magic state distillation. *Nature* **645**, 620–625 (2025).
8. Muniz, J. A. et al. Repeated ancilla reuse for logical computation on a neutral atom quantum computer. *Phys. Rev. X.* **15**, 041040 (2025).
9. Bluvstein, D. et al. A fault-tolerant neutral-atom architecture for universal quantum computation. *Nature* **649**, 39–46 (2026).
10. Bluvstein, D. et al. Logical quantum processor based on reconfigurable atom arrays. *Nature* **626**, 58–65 (2024).
11. Kitaev, A. Y. Fault-tolerant quantum computation by anyons. *Ann. Phys.* **303**, 2–30 (2003).
12. Dennis, E., Kitaev, A., Landahl, A. & Preskill, J. Topological quantum memory. *J. Math. Phys.* **43**, 4452–4505 (2002).

13. Fowler, A. G., Mariantoni, M., Martinis, J. M. & Cleland, A. N. Surface codes: towards practical large-scale quantum computation. *Phys. Rev. A* **86**, 032324 (2012).
14. Fowler, A. G., Stephens, A. M. & Groszkowski, P. High-threshold universal quantum computation on the surface code. *Phys. Rev. A* **80**, 052312 (2009).
15. Higgott, O. PyMatching: a Python package for decoding quantum codes with minimum-weight perfect matching. Preprint at http://arxiv.org/abs/2105.13082 (2021).
16. Horsman, D., Fowler, A. G., Devitt, S. & Meter, R. V. Surface code quantum computing by lattice surgery. *N. J. Phys.* **14**, 123011 (2012).
17. Krinner, S. et al. Realizing repeated quantum error correction in a distance-three surface code. *Nature* **605**, 669–674 (2022).
18. Zhao, Y. et al. Realization of an error-correcting surface code with superconducting qubits. *Phys. Rev. Lett.* **129**, 030501 (2022).
19. Breuckmann, N. P. & Burton, S. Fold-transversal Clifford gates for quantum codes. *Quantum* **8**, 1372 (2024).
20. Moussa, J. E. Transversal Clifford gates on folded surface codes. *Phys. Rev. A* **94**, 042316 (2016).
21. Kubica, A., Yoshida, B. & Pastawski, F. Unfolding the color code. *N. J. Phys.* **17**, 083026 (2015).
22. Postler, L. et al. Demonstration of fault-tolerant universal quantum gate operations. *Nature* **605**, 675–680 (2022).
23. Bravyi, S. & Kitaev, A. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A* **71**, 022316 (2005).
24. Gidney, C., Shutty, N. & Jones, C. Magic state cultivation: growing T states as cheap as CNOT gates. Preprint at http://arxiv.org/abs/2409.17595 (2024).
25. Sahay, K. et al. Fold-transversal surface code cultivation. Preprint at http://arxiv.org/abs/2509.05212 (2025).
26. Gottesman, D. Stabilizer codes and quantum error correction. Preprint at http://arxiv.org/abs/quant-ph/9705052 (1997).
27. Liao, P. & Feder, D. L. Graph-state representation of the toric code. *Phys. Rev. A* **104**, 012432 (2021).
28. Aguado, M. & Vidal, G. Entanglement renormalization and topological order. *Phys. Rev. Lett.* **100**, 070404 (2008).
29. Aharonov, D. & Touati, Y. Quantum circuit depth lower bounds for homological codes. Preprint at http://arxiv.org/abs/1810.03912 (2018).
30. Bravyi, S., Hastings, M. B. & Verstraete, F. Lieb-Robinson bounds and the generation of correlations and topological quantum order. *Phys. Rev. Lett.* **97**, 050401 (2006).
31. Higgott, O. et al. Optimal local unitary encoding circuits for the surface code. *Quantum* **5**, 517 (2021).
32. Chen, P., Yan, B. & Cui, S. X. Quantum circuits for toric code and X-cube fracton model. *Quantum* **8**, 1276 (2024).
33. Claes, J. Lower-depth local encoding circuits for the surface code. Preprint at http://arxiv.org/abs/2509.09779 (2025).
34. Vidal, G. Class of quantum many-body states that can be efficiently simulated. *Phys. Rev. Lett.* **101**, 110501 (2008).
35. Tsai, P.-K. & Puri, S. A unitary encoder for surface codes. http://arxiv.org/abs/2506.04084 (2025).
36. Raussendorf, R., Bravyi, S. & Harrington, J. Long-range quantum entanglement in noisy cluster states. *Phys. Rev. A* **71**, 062313 (2005).
37. Bergamaschi, T. & Liu, Y. On fault tolerant single-shot logical state preparation and robust long-range entanglement. In *Proc. 16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, vol. 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, (ed. Meka, R.) 16:1–16:9 https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2025.16 (Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025).
38. Chao, R. & Reichardt, B. W. Flag fault-tolerant error correction for any stabilizer code. *PRX Quantum* **1**, 010302 (2020).
39. Chamberland, C. & Beverland, M. E. Flag fault-tolerant error correction with arbitrary distance codes. *Quantum* **2**, 53 (2018).
40. Chao, R. & Reichardt, B. W. Quantum error correction with only two extra qubits. *Phys. Rev. Lett.* **121**, 050502 (2018).
41. Cobos, J., Locher, D. F., Bermudez, A., Müller, M. & Rico, E. Noise-aware variational eigensolvers: a dissipative route for lattice gauge theories. *PRX Quantum* **5**, 030340 (2024).
42. Zen, R. et al. Quantum circuit discovery for fault-tolerant logical state preparation with reinforcement learning. *Phys. Rev. X* **15**, 041012 (2025).
43. Google Quantum AI. Exponential suppression of bit or phase errors with cyclic error correction. *Nature* **595**, 383–387 (2021).
44. Tomita, Y. & Svore, K. M. Low-distance surface codes under realistic quantum noise. *Phys. Rev. A* **90**, 062320 (2014).
45. Moses, S. et al. A race-track trapped-ion quantum processor. *Phys. Rev. X* **13**, 041052 (2023).
46. Old, J., Tasler, S., Hartmann, M. J. & Müller, M. Fault-tolerant stabilizer measurements in surface codes with three-qubit gates. *Phys. Rev. Lett.* **135**, 240601 (2025).
47. Jandura, S. & Pupillo, G. Surface code stabilizer measurements for Rydberg atoms. *Phys. Rev. Research.* **8**, L022059 (2026).
48. Radnaev, A. et al. Universal neutral-atom quantum computer with individual optical addressing and nondestructive readout. *PRX Quantum* **6**, 030334 (2025).
49. Henriet, L. et al. Quantum computing with neutral atoms. *Quantum* **4**, 327 (2020).
50. Blinov, B. B., Leibfried, D., Monroe, C. & Wineland, D. J. Quantum computing with trapped ion hyperfine qubits. *Quantum Inf. Process.* **3**, 45–59 (2004).
51. Ringbauer, M. et al. A universal qudit quantum processor with trapped ions. *Nat. Phys.* **18**, 1053–1057 (2022).
52. Zhou, H. et al. Low-overhead transversal fault tolerance for universal quantum computation. *Nature.* **646**, 303–308 (2025)
53. Manes, A. G. & Claes, J. Distance-preserving stabilizer measurements in hypergraph product codes. *Quantum* **9**, 1618 (2025).
54. Amaro, D. Stabgraph. https://github.com/davamaro/stabgraph (2019).
55. Postler, L. et al. Demonstration of fault-tolerant Steane quantum error correction. *PRX Quantum* **5**, 030326 (2024).
56. Cain, M. et al. Correlated decoding of logical algorithms with transversal gates. *Phys. Rev. Lett.* **133**, 240602 (2024).
57. Lang, N. & Büchler, H. P. Minimal instances for toric code ground states. *Phys. Rev. A* **86**, 022336 (2012).
58. Amaro, D., Müller, M. & Pal, A. K. Scalable characterization of localizable entanglement in noisy topological quantum codes. *N. J. Phys.* **22**, 053038 (2020).
59. Knill, E. Scalable quantum computing in the presence of large detected-error rates. *Phys. Rev. A* **71**, 042322 (2005).
60. Steane, A. M. Active stabilization, quantum computation, and quantum state synthesis. *Phys. Rev. Lett.* **78**, 2252–2255 (1997).
61. Harris, C. R. et al. Array programming with NumPy. *Nature* **585**, 357–362 (2020).
62. Gidney, C. Stim: a fast stabilizer circuit simulator. *Quantum* **5**, 497 (2021).

Acknowledgements

We thank Josias Old for fruitful discussions. L.C. and M.M. gratefully acknowledge funding by the U.S. ARO Grant No. W911NF-21-1-0007. M.M. furthermore acknowledges funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101114305 ("MILLENION-SGA1" EU Project), and the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the Research Program Quantum Systems, research project 13N17317 ("SQale"), and MUNIQC-ATOMS (Grant No. 13N16070). This research is also part of the Munich Quantum Valley (K-4 and K-8), which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus. M.M. acknowledges funding from the ERC Starting Grant

QNets through Grant No. 804247, and from the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101046968 (BRISQ). L.C. and M.M. also acknowledge support for the research that was sponsored by IARPA and the Army Research Office, under the Entangled Logical Qubits program, and was accomplished under Cooperative Agreement Number W911NF-23-2-0216. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of IARPA, the Army Research Office, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. M.M. acknowledges support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1 390534769. The authors gratefully acknowledge the computing time provided to them at the NHR Center NHR4CES at RWTH Aachen University (Project No. p0020074). This is funded by the Federal Ministry of Education and Research and the state governments participating on the basis of the resolutions of the GWK for national high-performance computing at universities. We used the Python libraries NumPy[61] and Stim[62].

Author contributions

R.Z. and J.O. performed the numerical experiments that led to the main idea of the work. L.C. analyzed and generalized the main ideas with input from R.Z., J.O., and M.M., and perform numerical simulations. R.Z. prepared the figures of the paper, and L.C. wrote the paper with input from all co-authors. M.M. and F.M. supervised the work.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-026-01362-4.

Correspondence and requests for materials should be addressed to Luis Colmenarez.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2026