
Supplementary information

**Scaling and logic in the colour code on a
superconducting quantum processor**

In the format provided by the
authors and unedited

Supplemental Information for: Scaling and logic in the color code on a superconducting quantum processor

CONTENTS

A. Device and Experimental Setup	1
B. Superdense Syndrome Extraction Circuit	3
1. Circuit Intuition	3
2. Fault-tolerance of the Superdense Syndrome Extraction Circuit	4
3. Superdense Circuit Transformations	4
C. Distance scaling	5
1. Detailed Superdense Circuit	5
2. Color Code Location on Processor	5
3. Experimental Optimization and Calibration Details	5
4. Error syndrome detection probabilities	6
D. Decoding	8
1. Decoder details	8
a. Möbius Decoder (Chromobius)	8
b. Most Likely Error (MLE) Decoder	9
c. Neural Network Decoder	10
2. Decoding Priors	10
3. Comparison of Decoding Performance	11
E. Pauli Simulations near the Superdense Color Code Threshold	11
F. Logical Randomized Benchmarking	12
G. State injection	12
H. State teleportation	14
1. State Teleportation Circuit Details	14
2. Lower Bound on Average Fidelity of the Teleportation Channel	15
3. Bias of the Decoded $\langle \hat{Y}_L \rangle$ in Teleported State Tomography	17
I. Comparison of Qubit Resources Estimates for the Superdense Color Code and the Surface Code	17
References	19

Appendix A: Device and Experimental Setup

We implement our experiments on a 72-qubit device first introduced in Ref. 1. The device is installed at the bottom of a dilution refrigerator [2] and wired to custom room-temperature electronics for control. We use a single control line per flux-tunable transmon [3] qubit to realize single-qubit gates and to tune the transition frequency to implement two-qubit gates and multi-level

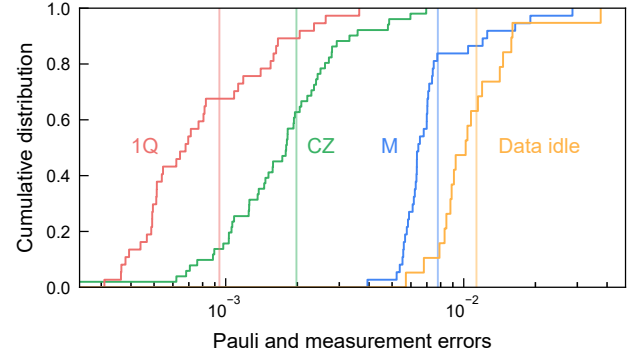
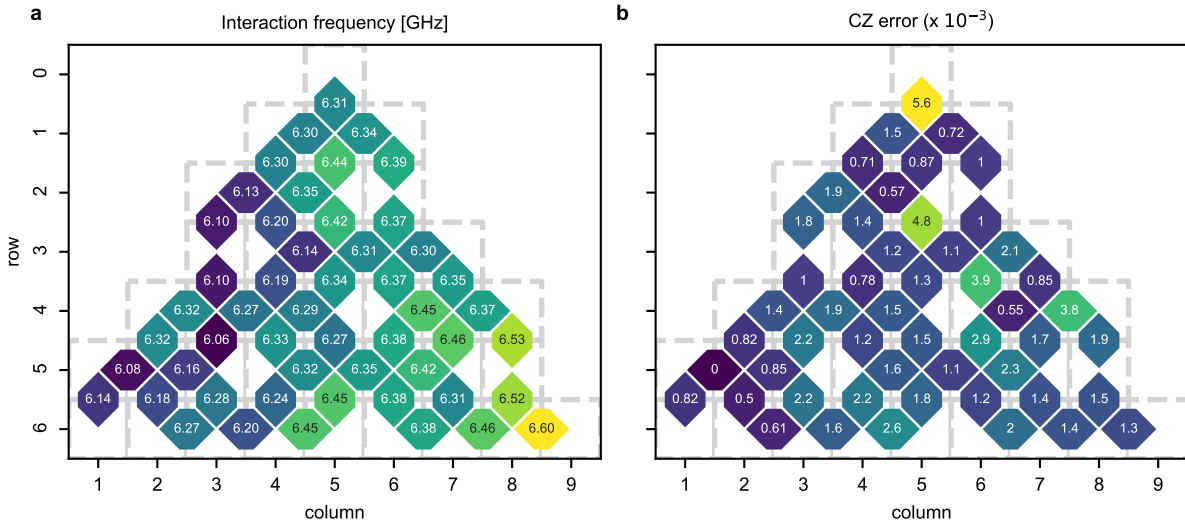
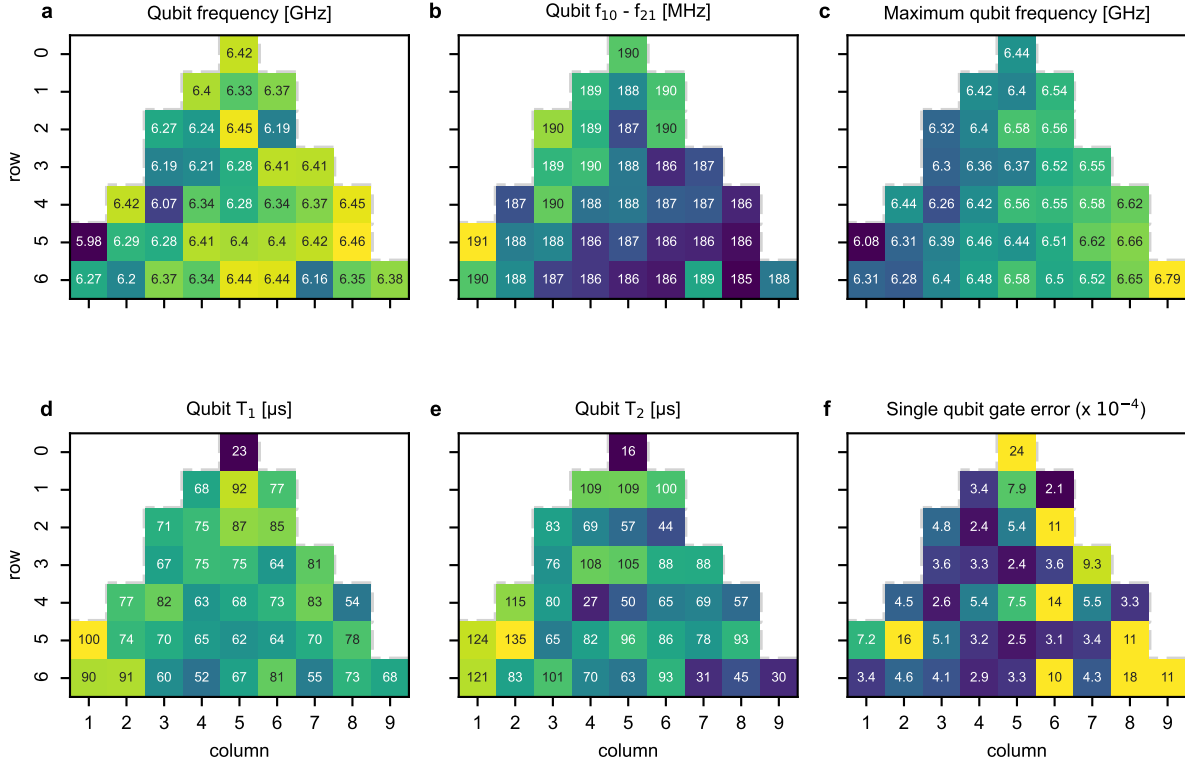


FIG. 1. Operation errors of the distance-five color code on the 72-qubit processor. Cumulative distributions of single-qubit gate Pauli errors (red), simultaneous CZ gate Pauli errors (green), measurement errors (blue), and data-qubit idle errors during measurement and reset of auxiliary qubits when applying a dynamical decoupling sequence (gold). The vertical faded lines indicate the mean value of the distributions.

reset [4]. We implement single-qubit gates using microwave DRAG pulses [5] for X rotations and virtual Z gates [6]. π -rotations and $\pi/2$ -rotations have durations of 35 ns and 18 ns, respectively. Controlled- Z (CZ) gates between neighboring qubits are realized by tuning the transition frequencies of the two qubits such that the $|11\rangle$ state is resonant with the $|20\rangle$ state, while also adjusting the frequency of a tunable coupler to control the coupling strength. These CZ gates have a duration of 37 ns.

Qubit states are read out dispersively by applying frequency-multiplexed microwave pulses. The readout protocol, including cavity depletion time, lasts for 600 ns. Following readout, we perform a 160-ns multi-level reset as described in Ref. 4. For state preservation experiments, we also include a data qubit leakage removal (DQLR) procedure lasting 53 ns.

We benchmark all operations of the distance-five color code and show the cumulative distribution of error rates in Fig. 1. We display device maps of single-qubit parameters in Fig. 2, two-qubit CZ gate parameters in Fig. 3, and readout parameters in Fig. 4. The coordinate system used for these maps matches the one depicting the code instances locations on the processor, see Appendix C 2 for details.



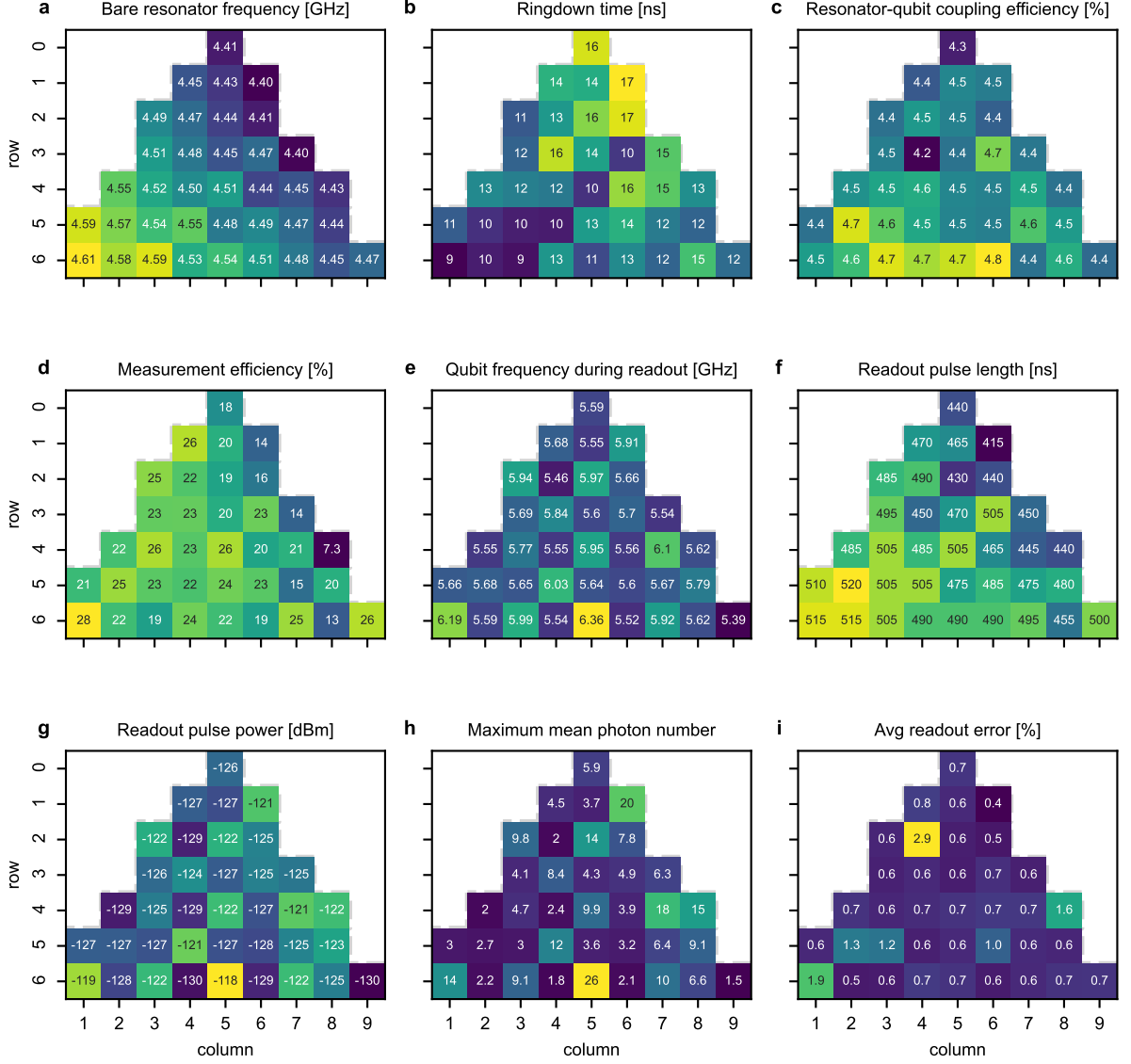


FIG. 4. **Readout parameters.** **a**, Bare frequency of each readout resonator. **b**, Ringdown time (equivalent to $1/\kappa$ of each readout resonator). **c**, The coupling efficiency (k_{qr}) between resonator and qubit. The qubit-resonator coupling strength is given by $g_{qr} = k_{qr}\sqrt{\omega_q\omega_r}/2$, where ω_q and ω_r are the qubit and resonator frequencies respectively. **d**, The measurement efficiency of the readout. This is the fraction of available information of the heterodyne measurement available after signal amplification and digitization. **e**, Qubit frequency during readout (in the absence of a readout drive). **f**, Length of the applied readout tone on each readout resonator. **g**, Applied power to each readout resonator. **h**, Simulated maximum mean photon number in each readout resonator, as calibrated using the AC-Stark shift. **i**, Average readout error, $(P(0|1) + P(1|0))/2$, when preparing random states on all qubits.

Appendix B: Superdense Syndrome Extraction Circuit

1. Circuit Intuition

The superdense syndrome extraction circuit for the color code, introduced in Ref. 7, draws inspiration from the superdense coding protocol, where a Bell pair enables the encoding of two classical bits [8]. The superdense

circuit is constructed such that these two classical bits indicate whether a bit-flip error occurred (encoded as 0 for no error, 1 for an error) and/or a phase-flip error occurred (encoded as 0 for no error, 1 for an error) on the data qubits of a given color code tile.

The intuition behind this is that the two auxiliary qubits of the tile are prepared in a Bell state, i.e. a shared quantum resource. This Bell state encodes bit-flip errors in its parity, toggling between even and odd parity depending on whether a bit-flip error occurred on the neigh-

boring data qubits. Simultaneously, it encodes phase-flip errors in its relative phase, which flips between positive and negative depending on whether a phase-flip error occurred. This simultaneous encoding allows the circuit to identify both types of errors using a single round of measurements.

The details of this process can be understood by analyzing how the Bell state evolves throughout the syndrome extraction circuit. Initially, the auxiliary qubits are prepared in the Bell state $|B_{00}\rangle = |00\rangle + |11\rangle$, where the first index (0) represents even parity (symmetric state), and the second index (0) indicates a positive relative phase. Due to this symmetry, a bit flip on either auxiliary qubit transforms the state to $|B_{10}\rangle = |01\rangle + |10\rangle$, with odd parity (antisymmetric state). This property allows each auxiliary qubit to check for bit-flip errors on its three neighboring data qubits using CNOT gates, where the data qubits act as control. Each CNOT toggles the Bell state between $|B_{00}\rangle$ and $|B_{10}\rangle$ if the data qubit is in the $|1\rangle$ state. After all six CNOT gates (left gray box in Fig. 1b of the main text), the Bell state will remain $|B_{00}\rangle$ if the total number of flips is even, and flip to $|B_{10}\rangle$ otherwise.

Similarly, a phase flip on either auxiliary qubit changes the state to $|B_{01}\rangle = |00\rangle - |11\rangle$, where the second index (1) indicates a negative relative phase. This enables detection of phase flips on the data qubits by using the auxiliary qubits as control in CNOT gates (right gray box in Fig. 1b of the main text). If both bit and phase flips occur, the state becomes $|B_{11}\rangle = |01\rangle - |10\rangle$.

Finally, a Bell-basis measurement of the auxiliary qubits determines the resulting Bell state, producing outcomes 00, 01, 10 or 11 corresponding to $|B_{00}\rangle$, $|B_{01}\rangle$, $|B_{10}\rangle$ and $|B_{11}\rangle$, respectively. Here, the first bit is the outcome of the Z auxiliary qubit (detects bit-flip errors) and the second bit is the outcome of the X auxiliary qubit (detects phase-flip errors). This process reveals whether a bit-flip error, a phase-flip error, both, or none occurred on the data qubits. Remarkably, this circuit achieves this even though each auxiliary qubit interacts with only half the data qubits of the tile.

Note that the Z auxiliary qubit also detects bit-flip errors on the X auxiliary qubit and vice-versa. In that way, the two auxiliary qubits detect errors on each other in addition to the errors on the data qubits [7]. This feature is important for ensuring the fault-tolerance of the circuit, which is analyzed in greater detail in the following subsection.

2. Fault-tolerance of the Superdense Syndrome Extraction Circuit

It is generally not guaranteed that a syndrome extraction circuit for a given code preserves the full code distance. For instance, the conventional syndrome extraction circuit for the color code [9–12] requires flag qubits [13] to ensure fault-tolerance. In this Appendix

we provide evidence that the superdense syndrome extraction circuit [7] for the color code preserves the full distance of the color code.

We define the circuit distance d_{circ} to be the minimum number of independent error mechanisms in the syndrome extraction circuit that must occur to flip the logical operator without causing any syndrome element to flip. In general, $d_{\text{circ}} \leq d$, where d is the distance of the code (the minimum weight of a logical operator in the code). It is well known that the standard surface code syndrome extraction circuit preserves the code distance ($d_{\text{circ}} = d$), provided that the sequence of gates is chosen correctly to avoid hook errors [14]. However, the circuit distance d_{circ} of the superdense color code circuits from Ref. 7 has not previously been studied.

We verify the circuit distance of the superdense color code using two different methods. The first one is an exact method that consists in mapping the task of determining the code distance to a Maximum Satisfiability (MaxSAT) problem [15]. MaxSAT is an optimization problem where the goal is to find the maximum number of constraints that can be satisfied simultaneously in a given Boolean formula. This approach works for our problem because determining the code distance involves finding the smallest set of independent errors that lead to an undetectable logical error, which can be encoded as a set of Boolean constraints. To generate the set of constraints for our circuits, we use the stim software package to create a circuit-level noisy representation of the superdense syndrome extraction circuit, where each operation is susceptible to potential errors. We then employ `stim.Circuit.shortest_error_sat_problem()` to produce the corresponding set of constraints in the WDIMACS format. These constraints are solved exactly as a MaxSAT problem using the CASHWMaxSAT-CorePlus MaxSAT solver, which provides the exact circuit distance. Using this method, we compute the circuit distance for distances 3, 5, and 7, confirming that $d_{\text{circ}} = d$.

Solving the MaxSAT problem to determine the circuit distance is NP-hard [15], which makes it increasingly challenging to compute the exact distance for larger circuits. Therefore, we employ a heuristic method to search for undetectable logical errors in circuits with larger distances (up to $d = 15$). Specifically, we use the `Circuit.search_for_undetectable_logical_errors` function in stim (with the first two arguments set to 3). Note that this heuristic approach only provides an upper bound on the circuit distance. Nonetheless, the results also suggest that the superdense circuit preserves the full code distance for all distances tested.

3. Superdense Circuit Transformations

Here, we detail how our implementation of the superdense syndrome extraction circuit differs from the original one introduced in Ref. 7. The circuit shown in

Fig. 1b of the main text (reproduced in Fig. 5b for convenience) is identical to the one shown in Ref. 7, except for two modifications: we do not show the conditional Pauli flips on data qubits that are handled in pre-processing [7], and we invert the orientation of the final CNOT along with the corresponding measurement bases of the two auxiliary qubits, for reasons that will become evident below. This is possible because the two orientations are nominally equivalent sub-circuits for performing Bell measurements.

When a qubit is measured, the readout pulse can unintentionally dephase other qubits that are not being measured [16]. On our device, this effect is more pronounced for qubits located on the same readout line as the measured qubit. To execute this circuit on our processor with best performance, we therefore apply additional transformations to ensure that all qubits sharing a readout line are measured simultaneously. Specifically, we swap the roles of auxiliary and data qubits for specific pairs, as indicated by the red arrows in Fig. 1c of the main text, such that each readout line contains only auxiliary qubits or only data qubits but no mixture of both types.

This transformation is illustrated for a single tile in Fig. 5. The most straightforward approach involves moving the measurement and reset operations from the X auxiliary qubit to its lateral data qubit neighbor, D3, and introducing swap gates between these qubits before the measurement and after the reset in each cycle (see Fig. 5d). Effectively, this swaps the role of the X auxiliary qubit and qubit D3 (we therefore relabel the qubits accordingly in Fig. 5d).

Since our device does not support direct implementation of swap gates, each swap must be decomposed into three two-qubit gates (Fig. 5e), which adds six layers of two-qubit gates to the error correction cycle. To circumvent this overhead, we implement a logically equivalent circuit (Fig. 5h) that increases the cycle by only two layers of two-qubit gates by reusing operations already present in the cycle. This optimized circuit is derived from the one shown in Fig. 5b by commuting CNOT gates between the X auxiliary qubit and data qubit D3 (Fig. 5f) outwards and applying circuit identities, as highlighted in the green and blue boxes in Fig. 5g-h. The final circuit contains a Z gate applied to D3 conditioned on the measurement outcome of the X auxiliary qubit being 1. Similarly to other conditional Pauli operations in this circuit, this gate is not applied physically but instead commuted through the circuit in pre-processing by our error correction software [17].

Applying this circuit transformation to every tile of the code results in a deformed code layout (Fig. 5c) where each readout line (dotted gray line) exclusively contains either data qubits or auxiliary qubits. To achieve this using the described transformations—where the X auxiliary qubit is swapped with its lateral data qubit neighbor—neighboring tiles must have a mirrored layout for the X and Z auxiliary qubits. At the code boundaries, where the X auxiliary qubit has no lateral neighbor, the swap

is performed with the upper data qubit neighbor (D1) instead, as shown in Fig. 1c of the main text. Note that these circuit transformations are applicable to arbitrary distances of the superdense color code.

Finally, before executing the circuit on our hardware, the CNOT gates are decomposed into a combination of Hadamard and CZ gates which are natively supported by our processor. The compiling software optimizes this decomposition by canceling consecutive Hadamard gates on the same qubit to minimize the number of gates applied to each qubit, see also Appendix C. The total error correction cycle duration after decomposition and optimization is approximately 1.3 μ s.

Appendix C: Distance scaling

In this Appendix, we provide additional details about the state preservation experiments performed on distance-three and distance-five logical qubits.

1. Detailed Superdense Circuit

We outline the steps of a Z-basis state preservation experiment on the distance-five logical qubit, after performing all circuit transformations described in Appendix B 3. First, the data qubits are initialized in a randomly selected product state of Z-basis eigenstates (i.e. a random bitstring) and all auxiliary qubits are prepared in the $|+\rangle$ state (step 0 in Fig. 6). Then, we execute n cycles of error correction (steps 1 to 19). During the auxiliary qubit measurement and reset (steps 19 and 1), we apply an XY4 dynamical decoupling sequence [18] on all data qubits. After the auxiliary qubit resets, we also apply data-qubit leakage removal (DQLR) [19]. Note that in the first cycle, resetting the auxiliary qubits to the $|+\rangle$ state (frame 1 and 2) is skipped as it is already performed simultaneously to the data qubit initialization (step 0). In the last cycle, the auxiliary qubits are read out (step 19) simultaneously to all data qubits (step 20).

2. Color Code Location on Processor

We use 37 qubits of a 72-qubit quantum processor to encode the distance-five logical qubit, and three 13-qubit subsets to encode the distance-three logical qubits. The placement of the distance-three qubits on the processor is chosen to cover most of the qubits used in the encoding of the distance-five qubit, as visualized in Fig. 7.

3. Experimental Optimization and Calibration Details

We employ similar techniques as described in Refs. 1 and 20 to minimize coherent phase errors in the cir-

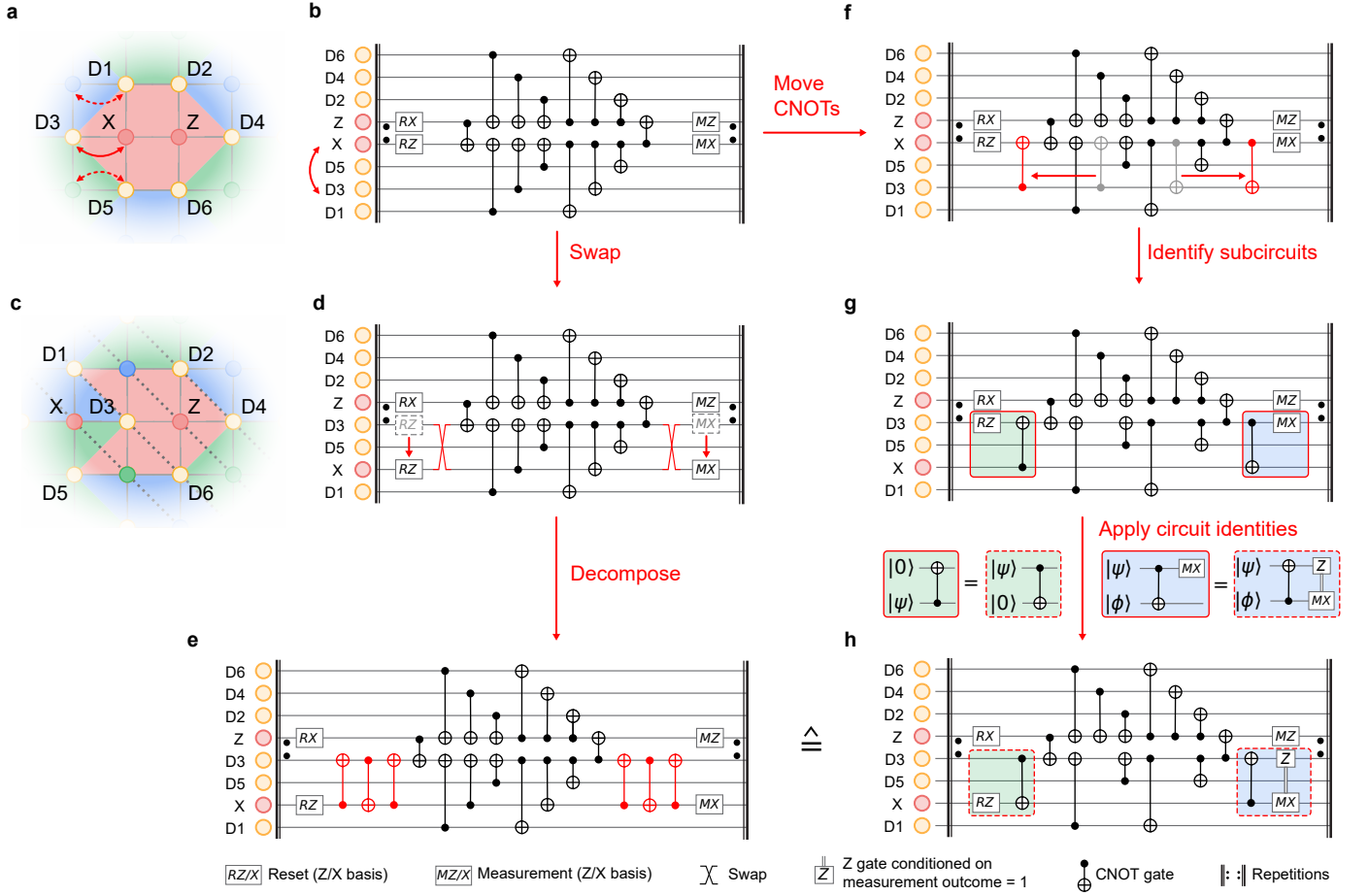


FIG. 5. **Transformation of the superdense syndrome extraction circuit.** Schematic representation of a hexagonal tile in the bulk of a color code (a) and its corresponding syndrome extraction circuit (b). The solid red arrow indicates the pair of qubits of this tile that will be swapped in our implementation. The dashed red arrows indicate similar swaps performed on neighboring tiles but whose circuit transformations are not shown in b. Schematic representation of a deformed tile (c) and its corresponding syndrome extraction circuit (d) with swap gates before the measurement and after the reset. **e** Syndrome extraction circuit for a naive swap gate decomposition. The CNOT highlighted in red correspond to the swap gates depicted in d. **f-h** Circuit transformations leading to a logically equivalent circuit as in e but with less physical operations (see text for details).

cuit [21], reduce dephasing during idle time on data qubits using dynamical decoupling, and ensure short-lived leakage errors with multi-level reset [4] on auxiliary qubits and a data-qubit leakage removal (DQLR) [19] protocol at the end of each cycle.

To mitigate effects from calibration drift, we interleave experiments with different codes, initial bases, and number of cycles following the approach described in Ref. 20. Each experiment is repeated 5000 times for 10 different initialization bitstrings: five are randomly chosen and the remaining five are their binary complements, thereby ensuring equal representation of both eigenstates for odd-distance codes. This results in a total of 50000 experimental runs per experiment.

4. Error syndrome detection probabilities

In each error correction cycle, the measurement of each auxiliary qubit indicates the parity of its associated stabilizer: 0 for even parity and 1 for odd parity. After initialization, the stabilizer parity (and thus the auxiliary qubit measurement outcome) should remain constant, except for intentional flips introduced by the circuit. Errors, such as bit-flips or phase-flips, alter the parity of neighboring stabilizers, leading to changes in the auxiliary qubit measurement outcomes between consecutive cycles. Comparing auxiliary qubit measurements across cycles therefore reveals information about errors occurring on the device. These comparisons for individual auxiliary qubits between two consecutive cycles are referred to as syndrome elements (also sometimes called detectors [20]) and take a value of 1 if an error is detected by

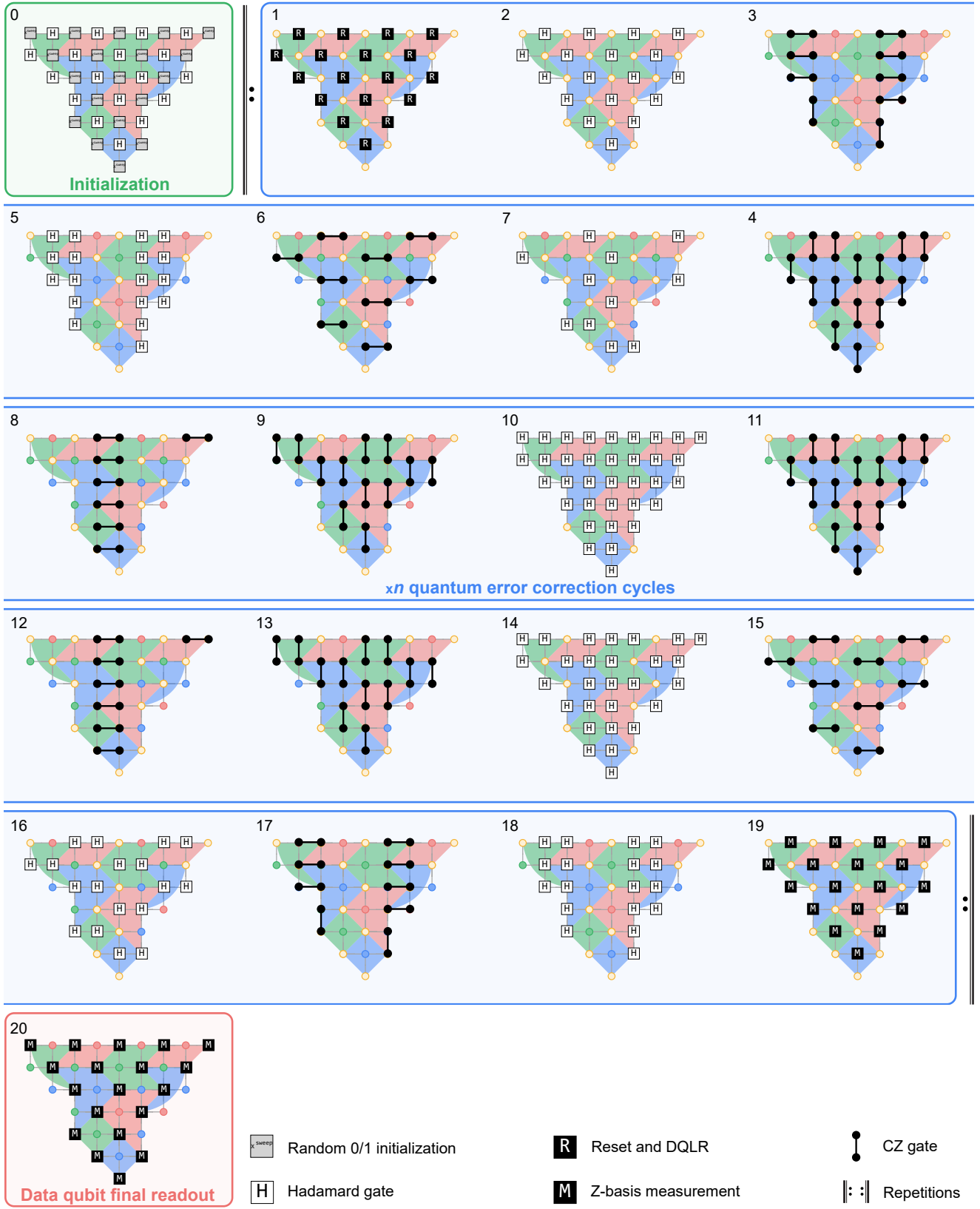


FIG. 6. **Z-basis state preservation superdense color code circuit for the distance-five logical qubit.** The initialization, quantum error correction cycle and final data qubit readout are highlighted in green, blue, and red, respectively. See text for details.

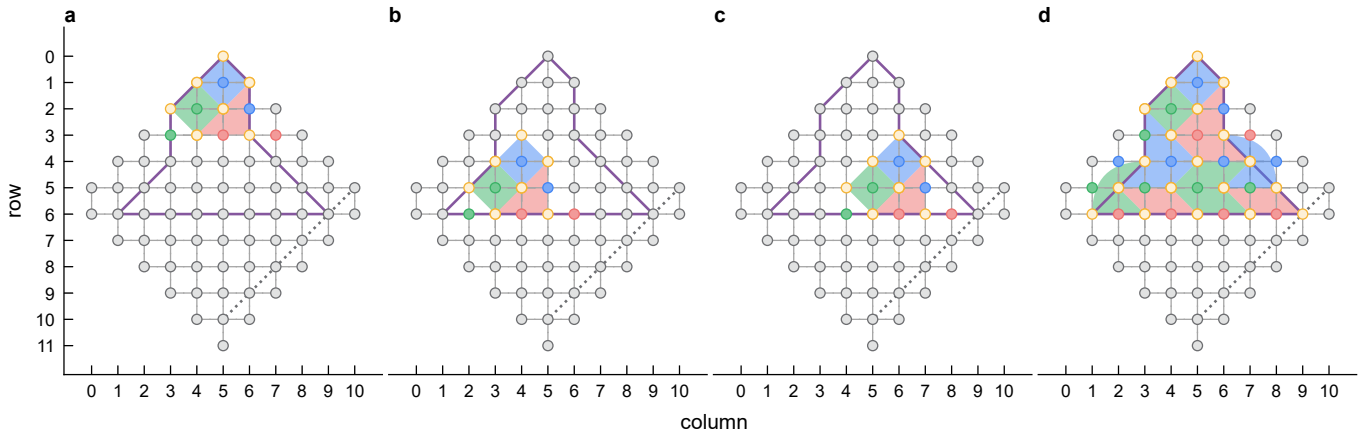


FIG. 7. **Qubit subsets used to implement the three distance-three (a-c) and the single distance-five (d) color codes for the distance-scaling experiment.** Data qubits (auxiliary qubits) for each subset are highlighted in gold (red, green, and blue), while inactive qubits of the 72-qubit processor are shown in gray. The purple outline marks the distance-five data qubit subset as a guide to the eye. Readout lines run diagonally from top right to bottom left, as indicated by a dotted gray line.

the syndrome extraction circuit, and 0 otherwise.

Analyzing how often each syndrome element detects an error provides insights into the frequency of errors across different parts of the device and stages of the quantum circuit. We calculate the error detection probability p_d for each syndrome element by averaging its value across all experimental runs. For all codes and bases, syndrome elements in the bulk of the experiment ($5 < n < 25$) show relatively stable detection probabilities between 0.1 and 0.2, indicating that errors are rare and do not increase over time (see Fig. 8). In the distance-five state preservation experiments, weight-6 stabilizers (red lines) exhibit higher error detection probabilities than boundary weight-4 stabilizers (golden lines). This is expected since weight-4 stabilizers involve fewer gates and are thus less prone to errors. Error detection probabilities are consistent across all experiments, except for certain X-type syndrome elements in the lowest distance-three code and the top-right part of the distance-five code, which have higher detection probabilities. Identifying the exact physical mechanism behind these additional phase errors requires further investigation. However, we hypothesize that they may result from spurious frequency shifts in data qubits induced by flux pulses used to reset auxiliary qubits, which are not fully corrected by our in-situ phase calibration [21]. Near the time boundaries of the circuit ($n < 5$ and $n > 25$), higher error detection probabilities are observed in the X basis for both distance-three and distance-five codes. We suspect this issue may be related to the one described above, and therefore warrants further investigation.

Appendix D: Decoding

For each experimental run, the collection of all syndrome elements form an error syndrome that can be used by a decoder to infer whether the logical operator has been flipped. In the color code, each data qubit is included in three X-type and three Z-type stabilizers. Therefore, a single bit-flip or phase-flip error on a data-qubit in the bulk of the code will typically flip three syndrome elements (versus two for the surface code), creating a so-called hyper-edge in the syndrome graph. This prevents the direct use of the widely-used minimum-weight-perfect-matching decoders [22], which rely on each error triggering a pair of syndrome elements.

In this work, we explore three decoders for the color code: a Möbius decoder (Chromobius) [7], a most-likely-error (MLE) decoder, and a neural-network decoder (AlphaQubit) [23]. We provide a short description of each of these decoders in the following Subsection and highlight their advantages and limitations.

1. Decoder details

a. Möbius Decoder (Chromobius)

Chromobius [7] decomposes the color-code decoding problem in three main steps. First, it decomposes the color-code decoding graph into three complementary matchable graphs without hyper-edges. These subgraphs are connected at their spacial boundaries in a Möbius topology [24]. Compared to other decoders that use similar decomposition methods [25], connecting the boundaries enables more accurate weighting of the relative costs of matching syndrome elements to a boundary (typically

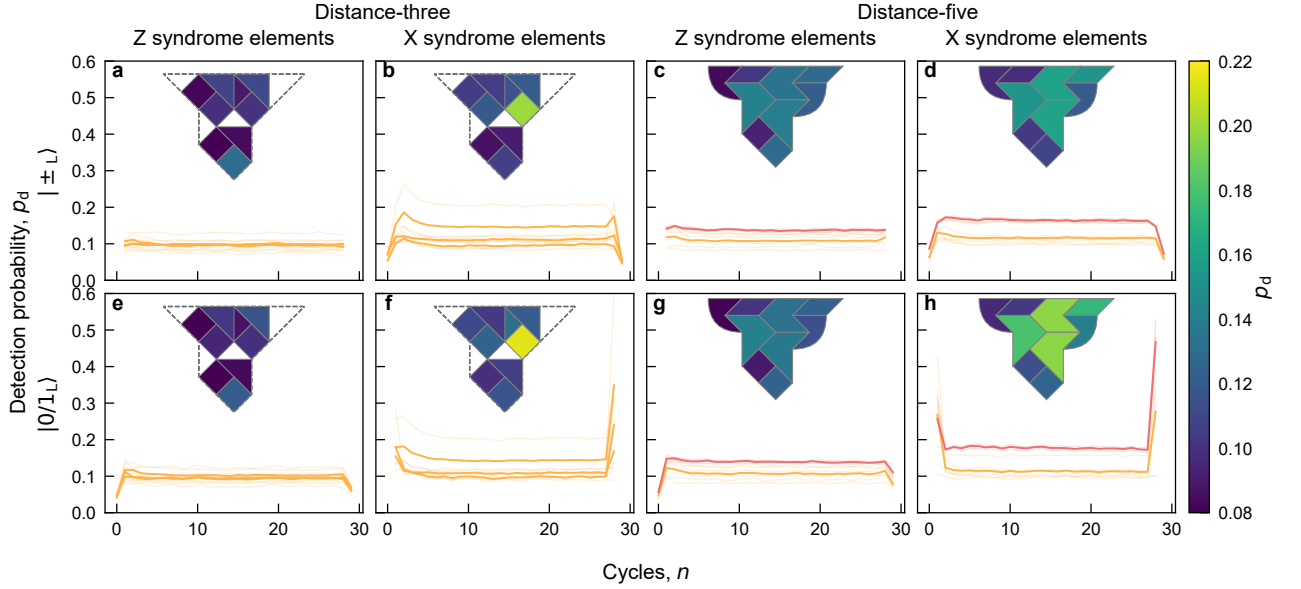


FIG. 8. **Average error detection probabilities for all syndrome elements in the distance-scaling experiment.** **a-d** Average weight-four (gold) and weight-6 (red) error detection probability, p_d , as a function of the error correction cycle, n , in the $|\pm_L\rangle$ (X-basis) state preservation experiments. Panels **a** and **b** show the Z syndrome and the X syndrome detection probabilities of the distance-three logical qubits, respectively. Faded lines are the detection probabilities of individual syndrome elements while the solid lines correspond to their averages for each logical qubit. Insets display heatmaps of the detection probabilities averaged over all cycles for each stabilizer of the code. The orientation of the code layouts in the insets matches the one of Fig. 1 in the main text. Panel **c** and **d** display the Z syndrome and X syndrome detection probabilities for the distance-five logical qubit. **e-h** is analogous to **a-d** but for the $|0/1_L\rangle$ (Z-basis) state preservation experiments.

underestimated when graphs are disconnected) versus matching them to other syndrome elements in the bulk of the complementary subgraphs [7, 24]. The resulting decoding graph is then processed by a minimum-weight perfect matching algorithm to find a perfect matching solution. Finally, the solution is lifted back to the original color code decoding graph by merging the matched syndrome elements using a dynamic programming algorithm [7].

Chromobius decodes data much faster than many other color code decoders, including the two other decoders used in this work, because the graph decomposition, the matching subroutine, and the solution lifting steps can all be performed efficiently. This makes Chromobius a promising candidate for real-time decoding in the near future.

Nonetheless, Chromobius has two notable limitations. First, it does not currently support flag-based decoding nor the ability to exploit correlations between the Z-basis and X-basis matching graphs. Second, it does not support certain logical operations that modify the decoding graph, such as those transforming X stabilizers into Y stabilizers. Consequently, Chromobius does not provide optimal performance and cannot be used to decode logical randomized benchmarking circuits.

In this work, we use Chromobius for the state injection experiment, and compare its performance to other decoders for the state preservation experiments (see Ap-

pendix D 3 for details).

b. Most Likely Error (MLE) Decoder

Inspired by Refs. 12 and 26, we develop a most-likely-error (MLE) decoder that identifies the most likely chain of Pauli errors given the observed error syndrome. The most-likely error is found by mapping the decoding problem onto a mixed-integer program [27, 28, Supp. 1A] which is solved with the High Performance Optimization Software (HiGHS) package [29]. This method is more accurate than matching-based approaches and naturally accounts for correlations between X and Z syndrome elements. Additionally, the Pauli error chains returned by the decoder can be used to iteratively refine the decoder's prior knowledge of frequent errors, as detailed in Appendix D 2.

The MLE decoder also has limitations. First, it is considerably slower than the other decoders. Specifically, it is currently about 880 times slower than the Möbius decoder and 260 times slower than the neural-network decoder when applied to the experimental data of the distance-five color code. While none of the decoders used in this work were fully optimized for speed, such optimizations will be necessary in the future to enable real-time decoding.

Second, while being more accurate than matching-

based decoders, it is not an optimal decoder such as a maximum likelihood decoder [30] or its tensor-network approximation [20]. These optimal decoders consider the probability of all possible error sets yielding the observed error syndrome, dividing them into two groups: those that flip the logical operator and those that do not. They then select the group with the greater total likelihood. However, such optimal decoders require much larger computational resources than our MLE decoder.

We compare the performance of the MLE decoder to other decoders for the state preservation experiments and use it in the simulations described in Appendices E and I.

c. Neural Network Decoder

AlphaQubit is a recurrent attention-based neural network first described in Ref. 23, and used in Ref. 1 to decode surface code experiments. Here, we adapt the network to decode color code experiments. Because of the lower translational symmetries of the color code compared to the surface code, the architecture of the model does not contain convolutions. In all experiments decoded in this work besides the state-teleportation experiments, we add up to two attention/recurrent layers to compensate for the loss of computational power. The resulting networks are significantly smaller: 4.1 million parameters (for the two layer variant) and 2.1 million (for the single layer variant), compared to the 5.4 million used in Ref. 1. We use a recurrence update as described in Ref. 23, rather than a gated-recurrence [31] update as in Ref. 1.

For each experiment, AlphaQubit is trained to predict the logical observable, based on the error syndrome calculated from the stabilizer measurements. The network processes one cycle of syndrome elements at a time, and (for the randomized benchmarking experiments) receives information about logical gates applied in between cycles, updating an internal state representation which consists of a vector for each stabilizer. At the end of the experiment the state vectors are combined and further processed to make the prediction of the logical observable.

For state preservation experiments, separate models are trained for the X and Z bases for each code instance. By contrast, for logical randomized benchmarking and state-teleportation experiments, a single model is used for all logical circuits in each respective experiment. For state teleportation, a single instance of AlphaQubit processes the syndrome elements from both logical qubits in each cycle, along with additional syndrome elements corresponding to the stabilizers at the interface of the logical qubits during the merge cycles.

The network is trained in two stages. First, it is pre-trained on synthetic data generated with Stim [17] using a device-agnostic noise model (SI1000 with a range of noise levels). After pretraining, each network is fine-tuned by further training on a limited amount of experimental data (or device-specific simulated data e.g. for

the error budget). We do not fine-tune the network on a synthetic dataset generated with an error model derived from experimental data (as was previously done in Ref. 1). Particular care is taken to never train and test on the same data, by either using:

- an even/odd experimental run partitioning (randomized benchmarking),
- a causal split where we only fine-tune on data taken in a separate run prior to the test dataset (state preservation experiments),
- a fully independent synthetic training dataset (error budget), or
- a separate part of the experimental dataset for testing purposes (state teleportation).

After training, AlphaQubit can decode at an intermediate speed which is about 3 times slower than Chromobius, but 260 times faster than the MLE decoder at distance $d = 5$. Due to its high accuracy, AlphaQubit is the decoder we used in most of the main text of this article (besides the magic state injection experiments, where we make use of Chromobius and post-selection).

2. Decoding Priors

Both Chromobius and the MLE decoder rely on prior knowledge of the probabilities of individual errors occurring in the circuit. To provide this information, we explore three different approaches.

The simplest method calculates error probabilities of each operation in the circuit based on the SI1000 error model [32] parametrized with a single error parameter p . This model reflects the typical hierarchy of error rates in superconducting qubits (for instance, all two qubit gates have an error p while single-qubit gates have an error $p/10$) but remains device-agnostic.

The remaining two approaches make use of experimental data to refine a Pauli error model where each circuit operation has an associated a depolarizing error parameter and/or a bit-flip error parameter. The first experimentally-informed approach derives error probabilities for each circuit operation from experimental data decoded with Chromobius, following the reinforcement-learning methodology outlined in Ref. 33. The error probabilities are inferred from the 9-cycles dataset for each experiment separately, and then extrapolated using time-invariance to all other numbers of cycles [33].

Finally, we explore an edge-reweighting strategy inspired by Ref. 34, which iteratively adjusts the probabilities of individual errors based on their frequency as reported by the MLE decoder. Initially, all errors are assumed to be equally likely. In the first iteration, a calibration dataset is decoded using the MLE decoder, which provides the most likely error chain for each experimental run in the dataset. The probabilities of individual errors

are then updated by counting their occurrences in the list of most likely error chains across the dataset. This process is repeated until the decoded performance stabilizes, typically after 3 to 6 iterations.

3. Comparison of Decoding Performance

We use the state preservation experiments on the distance-three and distance-five codes to compare five decoding strategies:

1. Chromobius decoder with an SI1000 prior (green symbols in Fig. 9)
2. Chromobius decoder with a learned prior from experimental data (blue symbols)
3. MLE decoder with an SI1000 prior (red symbols)
4. MLE decoder with a prior obtained via iterative edge reweighting based on experimental data (purple symbols)
5. Neural-network decoder (AlphaQubit), pretrained on SI1000 simulated data and fine-tuned on experimental data (yellow symbols)

For each decoding strategy, we fit the decoded data and extract the logical error per cycle for each code instance and basis independently. Note that, for clarity, we show the logical error as a function of cycle number and the logical error per cycle averaged over the X and Z bases for each code instance in Fig. 2 of the main text. We average all distance-three and all distance-five experiments, and compute the error suppression factor $\Lambda_{3/5}$. We use the standard error reported by each fit and standard error propagation techniques to obtain error bars for the error suppression factor.

Three of the five strategies achieve a lower average logical error per cycle for the distance-five code compared to the distance-three code, as shown in Fig. 9. The best decoding strategy provides an error suppression factor of $\Lambda_{3/5} = 1.56(4)$, demonstrating operation of the color code below its critical noise threshold. As expected, both the Chromobius and MLE decoders perform better with experimentally-informed priors than with device-agnostic (SI1000) priors. Notably, the MLE decoder achieves performance comparable to the neural-network decoder, highlighting that a simple Pauli error model with depolarizing parameters for each circuit operations captures well the errors of the device. However, the neural-network decoder is significantly faster than the MLE decoder.

Note that the three decoding strategies relying on experimental data (strategies 2, 4, and 5) use different train/test data partitioning schemes. Consequently, the test dataset for each of these strategies may contain different experimental runs.

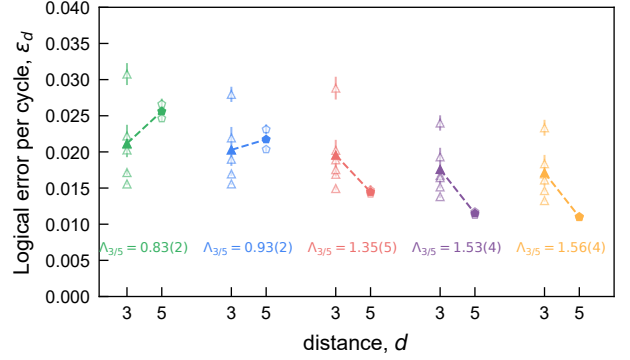


FIG. 9. **Comparison of decoding strategies for the distance scaling experiment.** Logical error per cycle of the state preservation experiments as a function of code distance for five decoding strategies: Chromobius decoder with SI1000 prior (green), Chromobius decoder with reinforcement-learning prior (blue), MLE decoder with SI1000 prior (red), MLE decoder with iterative edge-reweighting prior (purple), neural network decoder fine-tuned on experimental data (yellow). Symbols with contours only correspond to the logical error rates for individual experiments (i.e. one code instance in a given basis) and the filled symbols correspond to the average over all experiments. Error bars indicate the standard error from the fit. The dashed line serves as a guide to the eye. The error suppression factor $\Lambda_{3/5}$ is indicated for each decoding strategy with the corresponding color.

Appendix E: Pauli Simulations near the Superdense Color Code Threshold

In this article, we observe clear error suppression ($\Lambda_{3/5} = 1.56(4)$) when scaling the distance of the color code from three to five. However, there is a crossover region near the code threshold ($\Lambda \approx 1$) where the error suppression factor Λ can depend on the code distance, as reported in Ref. 20. Therefore, to relate our experimental results to the color code threshold, we perform simulations extrapolating to larger code distances.

We simulate the superdense color code circuit from Ref. 7 for distances $d = 3$ to 9 using an SI1000 error model [32] with varying noise strength p , and decode the data with the MLE decoder. We observe that the threshold at which the logical error per cycle for the higher distances intersect is around $p_{\text{th}} \approx 0.005$ (see Fig. 10). Near this threshold, $\Lambda_{3/5}$ is about 1.1 but decreases to about 1.0 at higher distances $\Lambda_{d/(d+2)}$. The simulated logical error per cycle for $p = 0.005$ is $\varepsilon_3 = 0.039(1)$ and $\varepsilon_5 = 0.036(1)$.

We compare these simulated values to the experimental observations. Experimentally, we observe $\Lambda_{3/5} = 1.53(4)$, $\bar{\varepsilon}_3 = 0.0176(3)$ and $\bar{\varepsilon}_5 = 0.0115(2)$, using the same MLE decoder. These are significantly lower than the simulated values at threshold, indicating that our experiment is indeed below the color code noise threshold.

Note that the SI1000 error model is a simplified repre-

sensation of the experimental system; it assumes homogeneity and does not account for error mechanisms such as leakage and stray interactions. Therefore, it would be valuable to experimentally measure the logical error per cycle at larger code distances in the future.

Appendix F: Logical Randomized Benchmarking

Demonstrating the implementation of logical Clifford gates is a critical step for fault-tolerant quantum computing, but benchmarking their error rates at the logical level presents significant challenges. Previous experimental efforts have primarily relied on partial [35] or full [36] process tomography. However, this method is sensitive to logical SPAM (State Preparation and Measurement) errors, which can be comparable to or even exceed the gate error itself, especially for the surface code in which Y-basis readout is not fault-tolerant [36]. Additionally, it may not accurately capture the performance of gates when applied within a logical circuit. Similar to how randomized benchmarking [37] provides robust error estimates at the physical qubit level, logical randomized benchmarking [38] offers a more reliable method for assessing the cost of logical Clifford gates within logical circuits.

We focus on the implementation of LRB on a single distance-three qubit. The 24 single-qubit logical Clifford gates can be decomposed into a sequence of Hadamard gates H (π -rotation along the $x + z$ axis of the Bloch sphere), phase gates S ($\pi/2$ -rotation along the z axis of the Bloch sphere) and Pauli X, Y , and Z gates. Specifically, they are obtained from a matrix product $A \cdot B$ where $A \in \{I, H, S, HS, SH, HSH\}$ and $B = \{I, X, Y, Z\}$.

In our implementation, we first generate logical circuits comprising a sequence of m randomly chosen Clifford gates, followed by a Clifford recovery gate [37] that returns the logical qubit to its initial state ($|0_L\rangle$ in our case). These gates are interleaved with error correction cycles.

Applying a logical gate between error correction cycles transforms the stabilizers, changing how syndrome elements are constructed, as illustrated in Fig. 3b-d of the main text. For instance, a transversal H gate turns an X stabilizer into a Z stabilizer (and vice-versa). Consequently, the X auxiliary qubit measurement in cycle n must be compared to the Z measurement in cycle $n + 1$ to form a valid syndrome element. Similarly, an S -gate transforms X stabilizers into Y stabilizers (leaving Z stabilizers unaffected), such that a phase flip in cycle n flips both X and Z auxiliary measurements in cycle $n + 1$. We track stabilizer transformations and determine which measurements to compare in each cycle according to these rules.

The circuits are then compiled into individual gates compatible with our hardware. During this compilation, single-qubit gates used for transversal logical gates may be merged with single-qubit operations from the neigh-

boring error correction cycles. This approach is reasonable because it could be employed for more complex logical circuits. The observed logical gate errors reflect not only by the additional physical operations performed for logical gates but also by the frequent basis changes on the logical qubit, including time spent in the Y -basis, which is vulnerable to both bit-flip and phase-flip errors.

Using the idling experiment as reference [37], we extract the additional error introduced by logical Clifford gates per cycle

$$\varepsilon_C = \frac{1}{2} \left(1 - \frac{p_C}{p_I} \right) \quad (\text{F1})$$

where p_C and p_I are the depolarizing parameters of the decaying Clifford experiment and of the reference idling experiment, respectively. p_C and p_I are obtained from fitting the average values shown in Fig. 3e of the main text to the function $Ap^m + 1/2$ with fitting parameters A and $p \in \{p_C, p_I\}$.

Note that LRB is a benchmarking tool, and the logical circuits it generates are not intended for practical applications, as these circuits comprising of Clifford gates only can be efficiently simulated on a classical computer [39].

Appendix G: State injection

To inject an arbitrary state $|\psi_L\rangle$ into a distance-three logical qubit, we first prepare the desired state on a physical qubit and then use a code expansion protocol [40, 41] to grow the color code from a degenerate distance $d = 1$ code to a $d = 3$ code. In this protocol, the six data qubits that are added to the code are pairwise initialized in Bell states, as indicated by the purple ellipses in Fig. 11a. The standard circuit to prepare a Bell state from two neighboring qubits initially in the zero state involves applying a Hadamard gate to the first qubit, followed by a CNOT gate with the first qubit acting as the control. However, since these pairs of data qubits are not directly connected on our device, we create the Bell state between them using an intermediate auxiliary qubit, as indicated by the dashed black contour in Fig. 11a. This process requires three two-qubit gates if the auxiliary qubit starts in the $|0\rangle$ state, or four if it begins in the $|+\rangle$ state, see Fig. 11b and Fig. 11c for a visual derivation of the circuit, respectively. While the Bell pairs are prepared, the injection qubit is prepared in a desired arbitrary state with single-qubit gates parametrized by the polar and azimuthal angles introduced in the main text.

After this initialization step, we perform a standard error correction cycle on the distance-three qubit, followed by a logical measurement in the X, Y , or Z basis for logical state tomography. We repeat the circuit 20 000 times for each basis and compute the expectation value of the corresponding logical operator $\langle O_L \rangle = 1 - 2\langle m_O \rangle$, where $\langle m_O \rangle$ is the logical outcome averaged over all repetitions.

We show the expectation values of $\hat{X}_L$, $\hat{Y}_L$, and $\hat{Z}_L$ for states along x - y plane of the Bloch sphere in Fig. 12a.

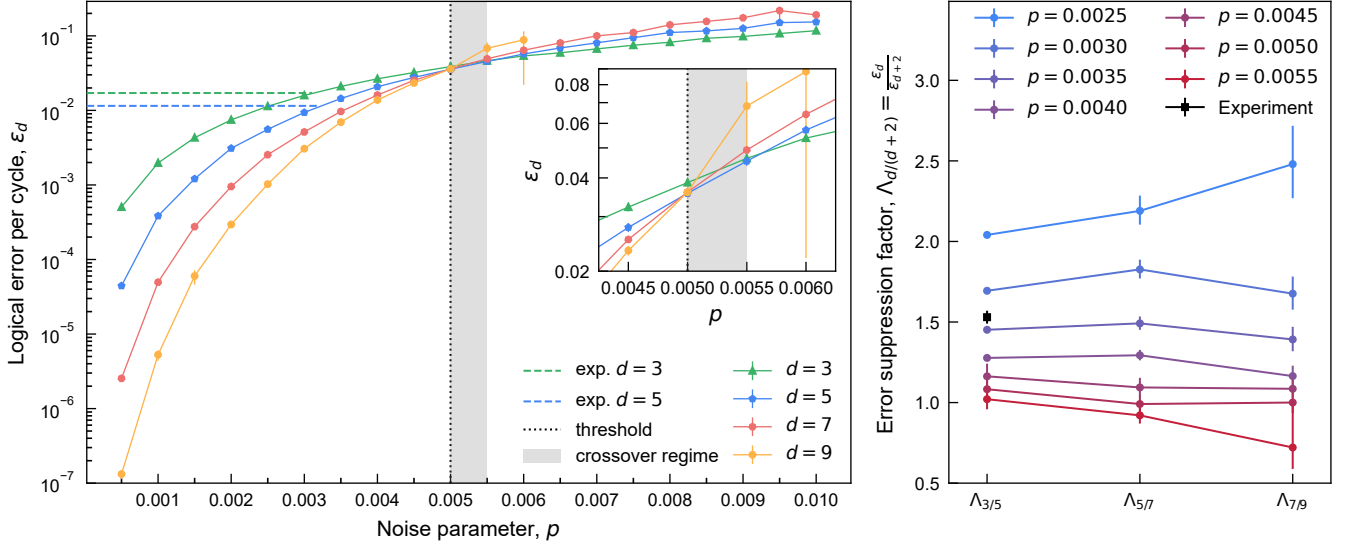


FIG. 10. **Simulation of the superdense color code using an SI1000 error model and decoded with the MLE decoders.** **a**, Logical error per cycle versus the noise parameter p of the SI1000 error model. The different curve correspond to different code distances d , and the gray region represents an approximate crossover region. We define the threshold (dotted black line) as the left bound of this crossover region. For smaller p than the $p_{\text{th}} \approx 0.005$, we find that higher distances outperform lower distances with a confidence interval of 90 % (which is represented by the error bars). The horizontal dashed lines correspond to the experimental value for $\bar{\varepsilon}_3$ (green) and ε_5 (blue). **b**, Simulated $\Lambda_{d,d+2}$ for different values of d . Each curve is associated with a different p noise strength, and the black point corresponds to the experimentally measured $\Lambda_{3/5}$.

To prepare these states on the injection qubit, we fix the polar angle $\theta = \pi/2$ and sweep the azimuthal angle ϕ from 0 to 2π . Note that we add a phase offset of $0.044\pi \approx 7.9^\circ$ to compensate for a coherent phase rotation of the logical qubit determined using logical state tomography in a prior calibration measurement. We reconstruct the approximate density matrix

$$\tilde{\rho}_L = \frac{1}{2} (I + \langle X_L \rangle \sigma_x + \langle Y_L \rangle \sigma_y + \langle Z_L \rangle \sigma_z) \quad (\text{G1})$$

where I is the identity matrix and $\sigma_{x,y,z}$ are the Pauli matrices. We then calculate the state fidelity

$$\mathcal{F}(\tilde{\rho}_L, \rho_L) = \left(\text{Tr} \sqrt{\sqrt{\tilde{\rho}_L} \rho_L \sqrt{\tilde{\rho}_L}} \right)^2 \quad (\text{G2})$$

to the target density matrix $\rho_L = |\psi_L\rangle\langle\psi_L|$, where Tr denotes the trace operation.

We obtain an average infidelity of $1 - \mathcal{F}(\tilde{\rho}_L, \rho_L) = 0.012(7)$ over the sweep when post-selecting on experimental runs in which all syndrome elements are 0 (keeping 74.4 % of the data), see Fig. 12b. We observe the largest discrepancy between decoded and post-selected values for $\hat{Y}_L$, which we attribute to both its sensitivity to both bit-flip and phase-flip errors, and the limited ability of the Möbius decoder to capture correlations between X and Z syndromes.

As described in the main text, we also prepare specific magic states of interest for realizing T -gates. Specifically, we examine $|A_L\rangle = \frac{1}{\sqrt{2}}(|0_L\rangle + e^{i\pi/4}|1_L\rangle)$,

$|H_L\rangle = \cos \frac{\pi}{8} |0_L\rangle + \sin \frac{\pi}{8} |1_L\rangle$, and $|T_L\rangle = \cos \frac{\beta}{2} |0_L\rangle + e^{i\pi/4} \sin \frac{\beta}{2} |1_L\rangle$ (with $\beta = \arccos 1/\sqrt{3}$). For these states, we perform an additional numerical optimization to obtain the maximum likelihood estimate of the experimentally reconstructed density matrix which ensures physicality constraints (all positive eigenvalues and a trace equal to 1).

We compare our results to other experimentally prepared magic states in prior work with superconducting qubits [36, 42–44] and trapped ions [35, 45, 46], see Fig. 13 for an overview of magic state infidelity as a function of the rejected data fraction r . For our implementation (indicated by green markers in Fig. 13 and also shown in Fig. 4e of the main text), we investigate four post-selection methods:

1. no post-selection ($r = 0$);
2. partial post-selection rejecting runs in which the syndrome $\sigma_R = 1$ ($r \approx 0.08$);
3. partial post-selection rejecting runs in which the syndrome $\sigma_R = 1$ or $\sigma_B = 1$ ($r \approx 0.13$);
4. full post-selection where all experimental runs with any non-trivial syndrome are rejected ($r \approx 0.25$).

Here, σ_R and σ_B denote the syndromes constructed from the final data qubit readout of the red and blue tiles in Fig. 11a, respectively. Among the seven syndromes in the circuit, these two are expected to be the most effective ones for detecting errors on the injected state.

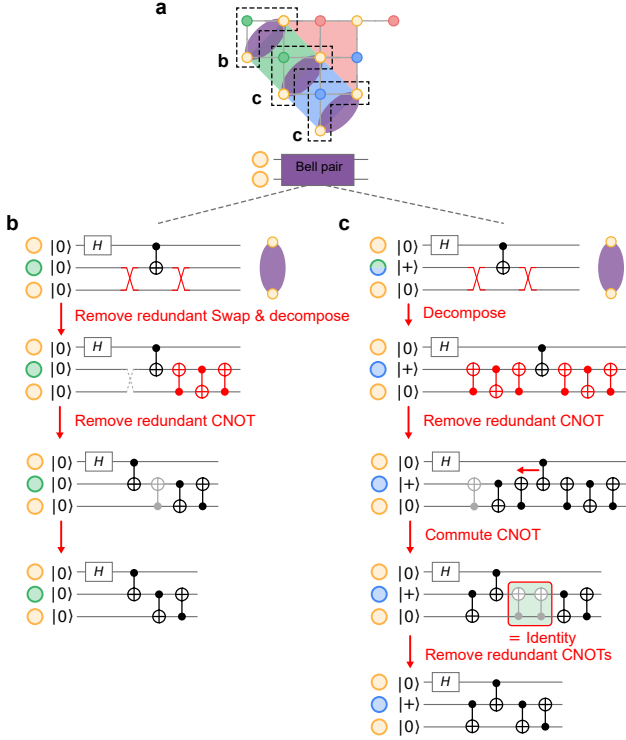


FIG. 11. **Bell state initialization circuit for magic state injection.** **a** Schematic of a distance-three color code, with data qubits shown as gold dots and auxiliary qubits as red, blue, and green dots. Gray lines represent qubit connectivity. Purple ellipses indicate pairs of data qubits initialized in Bell states for the magic state injection protocol (see main text for details). The dashed black contour highlights qubit subsets used to generate Bell states with the circuits shown in **b** and **c**. **b** Circuit transformations to generate a Bell state between two data qubits using an intermediate auxiliary qubit initialized in the $|0\rangle$ state. Red arrows indicate the sequence of circuit transformations. Starting with a circuit using swap gates (highlighted in red), the first swap gate is removed as both qubits are in the $|0\rangle$ state. The remaining swap gate is decomposed into three CNOT gates (second circuit), with the first CNOT removed (grayed out in third circuit) since its control qubit is in $|0\rangle$. The resulting circuit uses 3 two-qubit gates (fourth circuit). **c** Similar approach as in **b** but for a set of qubits in which the auxiliary qubit is initialized in $|+\rangle$. The first swap gate cannot be removed trivially. Swap gates are decomposed into CNOT gates (second circuit), the first CNOT is removed since its control qubit is in $|0\rangle$, and commuting CNOT gates are reordered (third circuit). Two consecutive CNOT gates cancel out (fourth circuit), resulting in a final circuit with 4 two-qubit gates (fifth circuit).

This is because σ_R includes the readout of the injection qubit and is therefore sensitive to errors on that qubit throughout most of the injection protocol while σ_B is sensitive to errors on the injected qubit during a limited part of the circuit. The remaining five syndromes are not directly sensitive to errors on the injection qubit at any stage. Conditioning on trivial outcomes of σ_R and σ_B thus increases the likelihood of detecting errors on the

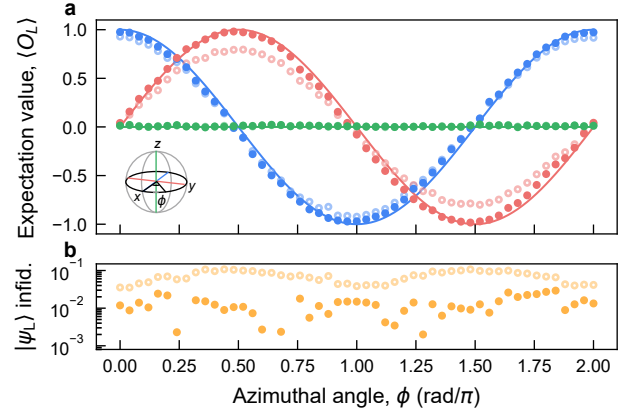


FIG. 12. **Arbitrary state injection along the x - y plane of the logical Bloch sphere.** **a** Decoded (semi-transparent circles) and post-selected (solid dots) expectation value of the logical Pauli operators $\hat{X}_L$ (blue), $\hat{Y}_L$ (red), $\hat{Z}_L$ (green) when sweeping the azimuthal angle ϕ . The solid lines correspond to ideal expectation values. **d** Decoded (semi-transparent circles) and post-selected (solid dots) infidelities for the prepared logical state $|\psi_L\rangle$.

injected state while minimizing the rejected data fraction.

For convenience, we use the Möbius decoder along with a device-agnostic SI1000 noise-model prior for all state-injection experiments. When full post-selection is applied, the achieved infidelity is independent of the decoder, as no syndrome information remains for the decoder to process. For partial or no post-selection, more elaborate decoders or priors could offer modest improvements in performance, though they ultimately would not reach the infidelity levels achieved with full post-selection.

We observe that our implementation can achieve higher fidelities and lower data rejection fractions than other implementations with superconducting qubits, and is on the Pareto front with some trapped ion implementations.

In our realization, the infidelity of $|A_L\rangle$ appears lower than that of the other two magic states under full post-selection, but we note that the difference is not statistically significant within the 95% bootstrapped confidence intervals (indicated by the error-bars).

Appendix H: State teleportation

1. State Teleportation Circuit Details

We provide a detailed circuit for teleporting $|1_L\rangle$ from logical qubit L1 to logical qubit L2 using lattice surgery. The circuit is divided into three parts: initialization, M_{XX} merge (X basis parity measurement of the two logical qubits), and split, highlighted in green, blue, and red in Fig. 14, respectively.

During initialization, L1 is prepared in $|\Psi_L\rangle = |1_L\rangle$

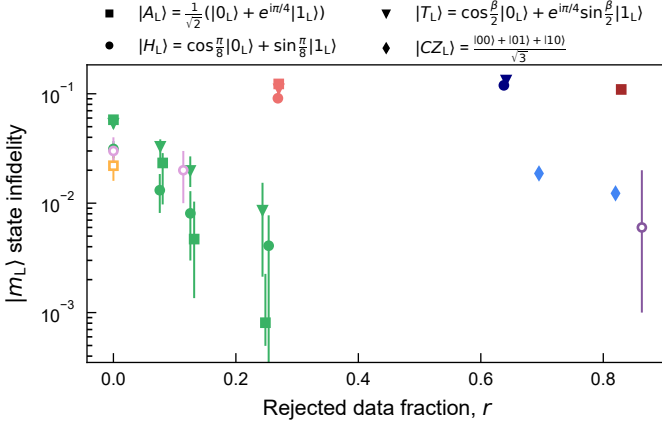


FIG. 13. **Comparison of magic state injection experiments.** Infidelity of four experimentally realized magic states (see legend) plotted against the rejected data fraction, r , with $r = 0$ corresponding to no post-selection. Open symbols represent implementations with trapped ions, while filled symbols correspond to superconducting qubits. Each color represents states realized in a different study. Each color represents states realized in a different study: purple [35], brown [36], red [42], light-blue [43], dark-blue [44], pink [45], yellow [46], and green (this work). Some works (including ours) report results with various post-selection strategies, leading to several symbols for the same magic state to appear multiple times in the same color. All data points have error bars, but some are too small to see. The error bars reported from our work correspond to a 95 % confidence interval calculated using bootstrapping.

and L2 in $|0_L\rangle$, using one cycle of error correction applied separately to each logical qubit. The merge operation, consisting of three cycles, introduces stabilizers in the X basis at the interface between the two logical qubits (hashed blue tiles) and expands four stabilizers (one X-type and one Z-type for each of the two boundary tiles) from weight-4 to weight-6 by incorporating two additional data qubits. In the first merge cycle, these interface data qubits are prepared in a Bell state (grayed operation in the dashed red box, panels 10-11) and are measured in the Bell basis during the third merge cycle (grayed operation in the dashed red box, panel 19). The M_{XX} parity outcome is the product of the two interface X stabilizers and the additional data qubit read out in the X basis.

The split operation consists of a single error correction cycle, after which the data qubits of L1 are measured in the Z basis, while those of L2 are measured in the X, Y, or Z basis for logical tomography. The Pauli frame update $\hat{Z}_L^{m_1} \hat{X}_L^{m_2}$ (where $m_1, m_2 \in \{0, 1\}$) is applied to the logical operators of L2 in post-processing (i.e. offline), based on the parity outcome m_1 and the measurement result of L1 m_2 , as outlined in the circuit diagram in Fig. 5a in the main text.

Our primary goal in this lattice surgery experiment is

to demonstrate functionality, and given circuit complexity, we choose not to apply the circuit transformation introduced in Appendix B 3. Nevertheless, we believe that such transformations can in principle still be applied in the context of lattice surgery. In future work, the impact of measurement-induced dephasing may be reduced through a combination of circuit-level techniques (such as these transformations) as well as hardware design choices, including wider frequency spacing of readout resonators, individual Purcell filters [47], or pulse shaping.

2. Lower Bound on Average Fidelity of the Teleportation Channel

In the main text, we report the expectation values of logical Pauli operators and the corresponding state fidelity for the teleportation of the Pauli eigenstates $|0_L\rangle$, $|1_L\rangle$, $|+_L\rangle$, and $|-_L\rangle$. These values can be used to provide a lower bound on the *average fidelity* $\bar{F}$ [48] of the teleportation channel $\mathcal{T}$ with respect to the identity channel $\mathcal{I}$.

First, we note that [48]

$$\bar{F}(\mathcal{T}) = \frac{dF_e(\mathcal{T}) + 1}{d + 1} \quad (\text{H1})$$

where $d = 2$ is the dimension of the single logical qubit Hilbert space and

$$F_e(\mathcal{T}) = \langle \phi | (\mathcal{I} \otimes \mathcal{T}) (|\phi\rangle\langle\phi|) | \phi \rangle \quad (\text{H2})$$

is the *entanglement fidelity* [49] of the teleportation channel. Here, $|\phi\rangle = (|0_L 0_L\rangle + |1_L 1_L\rangle)/\sqrt{2}$ is a Bell state of two logical qubits. Suppose $\mathcal{T}$ has a Kraus representation $\mathcal{T}(\rho) = \sum_k K_k \rho K_k^\dagger$. By expanding each Kraus operator in the logical Pauli basis, we can express $\mathcal{T}$ as

$$\mathcal{T}(\rho) = \sum_k K_k \rho K_k^\dagger \quad (\text{H3})$$

$$= \sum_k \left(\sum_i a_{ki} P_i \right) \rho \left(\sum_j a_{kj} P_j \right)^\dagger \quad (\text{H4})$$

$$= \sum_{ij} \left(\sum_k a_{ki} \overline{a_{kj}} \right) P_i \rho P_j \quad (\text{H5})$$

where $i, j \in \{0, 1, 2, 3\}$, $P_0 = I_L$, $P_1 = X_L$, $P_2 = Y_L$, $P_3 = Z_L$, and $k \in \{1, \dots, r\}$ where r is the channel's Kraus rank. The matrix $\chi_{ij} = \sum_k a_{ki} \overline{a_{kj}}$ is called the channel's *process matrix* [50]. The action of $\mathcal{T}$ on a density matrix ρ can then be written in terms of the process matrix as

$$\mathcal{T}(\rho) = \sum_{ij} \chi_{ij} P_i \rho P_j. \quad (\text{H6})$$

Note that χ is positive semidefinite and has unit trace.

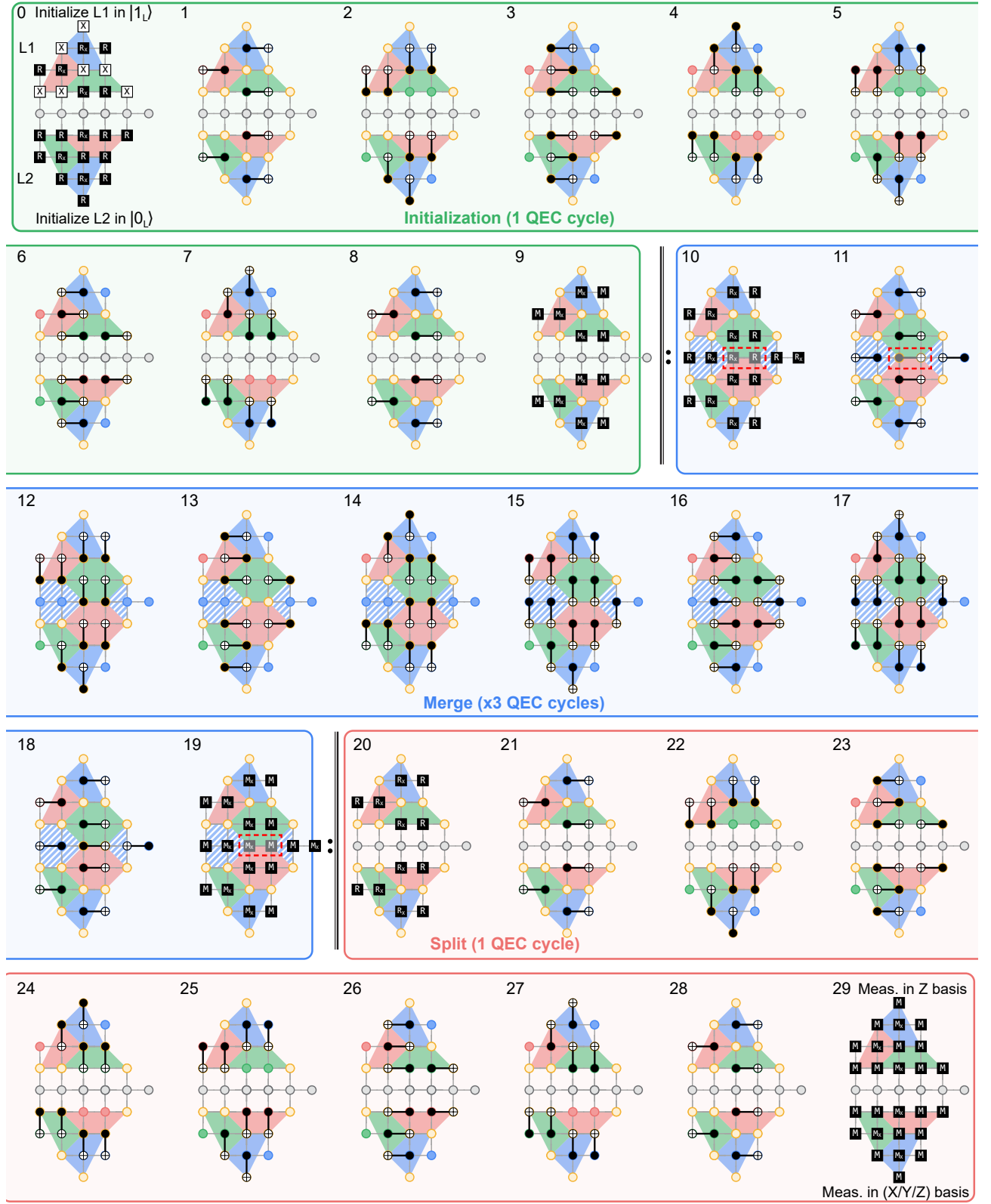


FIG. 14. Detailed quantum circuit used to teleport $|1_L\rangle$ from logical qubit L1 to logical qubit L2. Active data qubits (auxiliary qubits) are denoted with golden dots (red, green and blue dots), and inactive qubits are shown in gray. The red, green, and blue tiles indicate X and Z stabilizers with support on the data qubits located at the vertices of the tile, while the hatched blue tiles indicate X stabilizers only. The green, blue, and red boxes highlight the three parts of the protocol: initialization, merge, and split. Double black lines indicate repeated cycles of the merge operation, with dashed red outlines marking operations performed only in the first or last cycle (see text for details). The black boxes labeled R and M (Rx and Mx) correspond to reset and measurement operations in the Z basis (X basis), respectively.

Moreover,

$$F_e(\mathcal{T}) = \sum_{ij} \chi_{ij} \langle \phi | (I \otimes P_i) | \phi \rangle \langle \phi | (I \otimes P_j) | \phi \rangle \quad (\text{H7})$$

$$= \frac{1}{4} \sum_{ij} \chi_{ij} \text{tr}(P_i) \text{tr}(P_j) \quad (\text{H8})$$

$$= \chi_{00} \quad (\text{H9})$$

where the first equality results from the substitution of Eq. (H6) into Eq. (H2) and the second from the definition of $|\phi\rangle$. Therefore, we can read entanglement fidelity directly from a single element of the process matrix χ .

For any two (logical) Pauli operators P and Q , the quantity $\frac{1}{2} \text{tr}(Q\mathcal{T}(P))$, which we term *Pauli correlation*, is experimentally accessible by taking linear combinations of preparations of the eigenstates of P and measuring in the basis of Q . To see this, suppose that we perform the following experiment N^0 times: prepare $|0_L\rangle$, apply $\mathcal{T}$ and measure the observable Z_L , obtaining the +1 eigenvalue N_{agree}^0 times and the -1 eigenvalue N_{disagree}^0 times with $N_{\text{agree}}^0 + N_{\text{disagree}}^0 = N^0$. Then the Born rule implies that

$$\text{tr}(Z_L \mathcal{T}(|0_L\rangle\langle 0_L|)) = \frac{\mathbb{E}[N_{\text{agree}}^0] - \mathbb{E}[N_{\text{disagree}}^0]}{N^0} \quad (\text{H10})$$

where $\mathbb{E}$ denotes the expectation value. Similarly, if we perform a similar experiment N^1 times, preparing $|1_L\rangle$ instead of $|0_L\rangle$ and obtain the +1 eigenvalue N_{disagree}^1 times and the -1 eigenvalue N_{agree}^1 times with $N_{\text{disagree}}^1 + N_{\text{agree}}^1 = N^1$, then

$$\text{tr}(Z_L \mathcal{T}(|1_L\rangle\langle 1_L|)) = \frac{\mathbb{E}[N_{\text{disagree}}^1] - \mathbb{E}[N_{\text{agree}}^1]}{N^1}. \quad (\text{H11})$$

Subtracting Eq. (H11) from Eq. (H10) and using linearity of the trace and the channel $\mathcal{T}$, we find

$$\text{tr}(Z_L \mathcal{T}(Z_L)) = \frac{\mathbb{E}[N_{\text{agree}}^0] - \mathbb{E}[N_{\text{disagree}}^0]}{N^0} + \quad (\text{H12})$$

$$\frac{\mathbb{E}[N_{\text{agree}}^1] - \mathbb{E}[N_{\text{disagree}}^1]}{N^1}. \quad (\text{H13})$$

We can also express Pauli correlations as linear combinations of the elements of the process matrix. In particular,

$$\frac{1}{2} \text{tr}(X_L \mathcal{T}(X_L)) = \frac{1}{2} \sum_{ij} \chi_{ij} \text{tr}(X P_i X P_j) \quad (\text{H14})$$

$$= \chi_{00} + \chi_{11} - \chi_{22} - \chi_{33} \quad (\text{H15})$$

and

$$\frac{1}{2} \text{tr}(Z_L \mathcal{T}(Z_L)) = \chi_{00} - \chi_{11} - \chi_{22} + \chi_{33}. \quad (\text{H16})$$

Therefore,

$$\frac{1}{2} \left(\frac{1}{2} \text{tr}(X_L \mathcal{T}(X_L)) + \frac{1}{2} \text{tr}(Z_L \mathcal{T}(Z_L)) \right) = \chi_{00} - \chi_{22} \quad (\text{H17})$$

$$\leq \chi_{00} \quad (\text{H18})$$

$$= F_e(\mathcal{T}) \quad (\text{H19})$$

where we used $0 \leq \chi_{22}$ implied by the positive semidefiniteness of the process matrix. Note that χ_{22} tends to zero as $\mathcal{T}$ tends to the identity, so the bound above is asymptotically tight when $\mathcal{T} \rightarrow \mathcal{I}$.

Because Eq. (H1) is an affine transformation of the entanglement fidelity, it implies a lower bound on the average fidelity of the channel.

3. Bias of the Decoded $\langle \hat{Y}_L \rangle$ in Teleported State Tomography

In the lattice surgery data (see Fig. 5f-i of the main text), we find that initializing the logical qubit 1 (L1) in the X-basis (to either $|+_L\rangle$ or $|-_L\rangle$) and measuring in the Y-basis results in a non-zero (decoded) expectation value. This is unexpected because the teleported logical state does not commute with the logical measurement basis. Consequently the logical outcome in an ideal experiment should be fully random, such that $\langle \hat{Y}_L \rangle \approx 0$. However, the neural-network decoder detects a bias in the logical measurement, leading to $\langle \hat{Y}_L \rangle \approx 0.2$.

The exact source of the bias remains to be investigated, although we hypothesize that it is caused by a systematic Z-rotation of all the data qubits in the initialization cycle (before the merge operation) that was uncompensated in these two circuits specifically. In particular, in our matched basis experiments (i.e. experiments with a deterministic outcome such as measuring $|+_L\rangle$ in the X basis), we calibrate and apply virtual phases to compensate for the Z-rotations. However, no such phases were calibrated or applied in the non-deterministic mismatched basis circuits (e.g. measuring $|+_L\rangle$ in the Y basis), which may explain the observed bias.

We expect to be able to correct this phase in future work by performing an appropriate virtual-Z rotation on each data qubit.

Appendix I: Comparison of Qubit Resources Estimates for the Superdense Color Code and the Surface Code

Estimating the number of physical qubits required to reach algorithmically relevant error rates ($< 10^{-6}$) on a single logical qubit is critical for evaluating the resources required for large-scale quantum computation. The superdense color code requires $n_{cc} = \frac{3}{2}d^2 - \frac{1}{2}$, to encode a distance d logical qubit where n_{cc} is the number of qubits in the circuit (including auxiliary qubits). This

improves on the rotated surface code circuit, which requires $n_{\text{sc}} = 2d^2 - 1$ qubits – about 25 % more at a fixed distance.

However, the codes have different noise thresholds and their error suppression factor depend on the physical error rates of the device. Therefore, to assess the qubit overhead needed to reach algorithmically relevant error rates for each code, we compare them under realistic noise conditions of near-term processors. Specifically, we simulate both codes using an SI1000 error model [32] with noise parameters $p = 10^{-3}$ (about two times smaller than the error rates of our current device) and $p = 5 \cdot 10^{-4}$, for increasing code distances. We simulate d cycles of error correction for a distance- d code. We decode the resulting data with the MLE decoder to determine the logical error per cycle for each code and distance.

As expected, the logical error per cycle decreases with increasing code distance (and thus the number of qubits used), see Fig. 15. The results indicate that the surface code and the superdense color code will require a comparable number of qubits to achieve a given logical error per cycle. At large distances, the simulations suggest that the color code may require slightly fewer qubits than the surface code to reach logical error rates on the order of $\sim 10^{-8}$, particularly at $p = 5 \cdot 10^{-4}$.

This finding is promising, as it suggests that the two codes require similar resources at the physical error rates expected in near-term processors. This allows the choice of error-correcting code to be guided by other considerations, such as the ability to perform logical gates efficiently – a distinct advantage demonstrated in this work for the color code. On a longer horizon where physical error rates get even smaller, the color code is expected to eventually outperform the surface code in terms of qubit overhead.

We observe that the logical error per round for the superdense color code does not follow a linear trend on the square-root scale. Namely, the error suppression factor does not remain constant but increases with code distance. We attribute this effect to spacial boundary effects of triangular color code at small d , which become less relevant at larger distances.

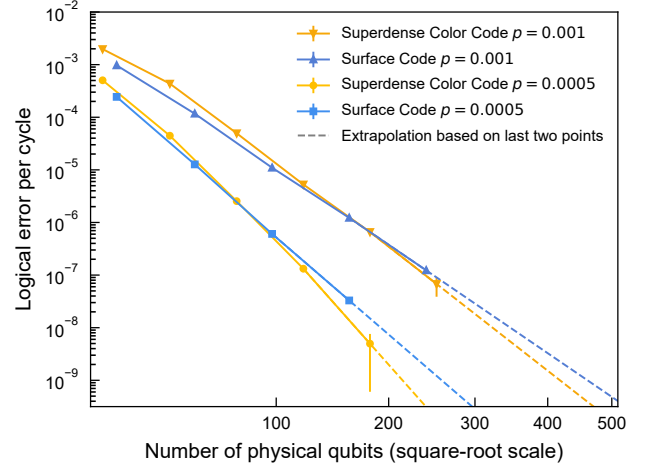


FIG. 15. **Numerical simulations of the superdense color code and surface code.** Simulated logical error per cycle for a qubit encoded in the superdense color code as in Ref. 7 (yellow) and the surface code (blue) for increasing code distances. The error per cycle is plotted as a function of the number of physical qubits required to encode the corresponding logical qubit. Simulations are based on samples taken from an SI1000 error model [32] with noise parameter $p = 10^{-3}$ and $p = 5 \cdot 10^{-4}$ decoded using a most-likely-error decoder implemented as a mixed integer program (Appendix D). The error bars are calculated from a 90% confidence interval of a binomial test. Dashed lines correspond to linear extrapolations based on the logical error per cycle of the highest-two code distances.

-
- [1] Google Quantum AI, Quantum error correction below the surface code threshold, *Nature* **638**, 920–926 (2025).
- [2] S. Krinner, S. Storz, P. Kurpiers, P. Magnard, J. Heinsoo, R. Keller, J. Lütolf, C. Eichler, and A. Wallraff, Engineering cryogenic setups for 100-qubit scale superconducting circuit systems, *EPJ Quantum Technology* **6**, 1 (2019).
- [3] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Charge-insensitive qubit design derived from the Cooper pair box, *Physical Review A—Atomic, Molecular, and Optical Physics* **76**, 042319 (2007).
- [4] M. McEwen, D. Kafri, Z. Chen, J. Atalaya, K. Satzinger, C. Quintana, P. V. Klimov, D. Sank, C. Gidney, A. Fowler, *et al.*, Removing leakage-induced correlated errors in superconducting quantum error correction, *Nature communications* **12**, 1761 (2021).
- [5] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm, Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits, *Physical Review Letters* **103**, 110501 (2009).
- [6] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, Efficient Z gates for quantum computing, *Physical Review A* **96**, 022330 (2017).
- [7] C. Gidney and C. Jones, New circuits and an open source decoder for the color code, *ArXiv e-prints* 10.48550/arXiv.2312.08813 (2023), 2312.08813.
- [8] C. H. Bennett and S. J. Wiesner, Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states, *Physical Review Letters* **69**, 2881 (1992).
- [9] M. Gutiérrez, M. Müller, and A. Bermúdez, Transversality and lattice surgery: Exploring realistic routes toward coupled logical qubits with trapped-ion quantum processors, *Physical Review A* **99**, 022330 (2019).
- [10] A. Bermudez, X. Xu, M. Gutiérrez, S. C. Benjamin, and M. Müller, Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources, *Physical Review A* **100**, 062307 (2019).
- [11] C. Chamberland, A. Kubica, T. J. Yoder, and G. Zhu, Triangular color codes on trivalent graphs with flag qubits, *New Journal of Physics* **22**, 023019 (2020).
- [12] Y. Takada and K. Fujii, Improving Threshold for Fault-Tolerant Color-Code Quantum Computing by Flagged Weight Optimization, *PRX Quantum* **5**, 030352 (2024).
- [13] R. Chao and B. W. Reichardt, Quantum Error Correction with Only Two Extra Qubits, *Physical Review Letters* **121**, 050502 (2018).
- [14] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, *Physical Review A—Atomic, Molecular, and Optical Physics* **86**, 032324 (2012).
- [15] M. W. Krentel, The complexity of optimization problems, *Journal of Computer and System Sciences* **36**, 490 (1988).
- [16] J. Heinsoo, C. K. Andersen, A. Remm, S. Krinner, T. Walter, Y. Salathé, S. Gasparinetti, J.-C. Besse, A. Potočník, A. Wallraff, and C. Eichler, Rapid High-fidelity Multiplexed Readout of Superconducting Qubits, *Physical Review Applied* **10**, 034040 (2018).
- [17] C. Gidney, Stim: a fast stabilizer circuit simulator, *Quantum* **5**, 497 (2021).
- [18] A. A. Maudsley, Modified Carr-Purcell-Meiboom-Gill sequence for NMR fourier imaging applications, *Journal of Magnetic Resonance* (1969) **69**, 488 (1986).
- [19] K. C. Miao, M. McEwen, J. Atalaya, D. Kafri, L. P. Pryadko, A. Bengtsson, A. Opremcak, K. J. Satzinger, Z. Chen, P. V. Klimov, *et al.*, Overcoming leakage in quantum error correction, *Nature Physics* **19**, 1780 (2023).
- [20] Google Quantum AI, Suppressing quantum errors by scaling a surface code logical qubit, *Nature* **614**, 676 (2023).
- [21] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, E. Lucero, M. Neeley, C. Neill, P. J. J. O’Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner, and J. M. Martinis, Scalable in situ qubit calibration during repetitive error detection, *Physical Review A* **94**, 032321 (2016).
- [22] O. Higgott and C. Gidney, Sparse Blossom: correcting a million errors per core second with minimum-weight matching, *arXiv preprint arXiv:2303.15933* (2023).
- [23] J. Bausch, A. W. Senior, F. J. H. Heras, T. Edlich, A. Davies, M. Newman, C. Jones, K. Satzinger, M. Y. Niu, S. Blackwell, G. Holland, D. Kafri, J. Atalaya, C. Gidney, D. Hassabis, S. Boixo, H. Neven, and P. Kohli, Learning high-accuracy error decoding for quantum processors, *Nature* **635**, 834 (2024).
- [24] K. Sahay and B. J. Brown, Decoder for the triangular color code by matching on a möbius strip, *PRX Quantum* **3**, 010310 (2022).
- [25] A. Kubica and N. Delfosse, Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders, *Quantum* **7**, 929 (2023), 1905.07393.
- [26] A. J. Landahl, J. T. Anderson, and P. R. Rice, Fault-tolerant quantum computing with color codes, *ArXiv e-prints* 10.48550/arXiv.1108.5738 (2011), 1108.5738.
- [27] A. Aggoun, C. Maravelias, and A. Vazacopoulos, Mixed Integer Programming/Constraint Programming Hybrid Methods, in *Encyclopedia of Optimization* (Springer, Boston, MA, Boston, MA, USA, 2008) pp. 2270–2276.
- [28] M. Cain, C. Zhao, H. Zhou, N. Meister, J. P. B. Ataiades, A. Jaffe, D. Bluvstein, and M. D. Lukin, Correlated decoding of logical algorithms with transversal gates, *ArXiv e-prints* 10.48550/arXiv.2403.03272 (2024), 2403.03272.
- [29] Q. Huangfu and J. A. J. Hall, Parallelizing the dual revised simplex method, *Mathematical Programming Computation* **10**, 119 (2018).
- [30] N. Sundaresan, T. J. Yoder, Y. Kim, M. Li, E. H. Chen, G. Harper, T. Thorbeck, A. W. Cross, A. D. Córcoles, and M. Takita, Demonstrating multi-round subsystem quantum error correction using matching and maximum likelihood decoders, *Nature Communications* **14**, 1 (2023).
- [31] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, *ArXiv e-prints* 10.48550/arXiv.1406.1078 (2014), 1406.1078.
- [32] C. Gidney, M. Newman, A. Fowler, and M. Broughton, A Fault-Tolerant Honeycomb Memory, *Quantum* **5**, 605 (2021), 2108.10457v2.
- [33] V. Sivak, M. Newman, and P. Klimov, Optimization of

- decoder priors for accurate quantum error correction, arXiv preprint arXiv:2406.02700 (2024).
- [34] H. Wang, P. Liu, Y. Liu, J. Gu, J. Baker, F. T. Chong, and S. Han, DGR: Tackling Drifted and Correlated Noise in Quantum Error Correction via Decoding Graph Re-weighting, ArXiv e-prints 10.48550/arXiv.2311.16214 (2023), 2311.16214.
 - [35] 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), 2111.12654.
 - [36] J. F. Marques, B. M. Varbanov, M. S. Moreira, H. Ali, N. Muthusubramanian, C. Zachariadis, F. Battistel, M. Beekman, N. Haider, W. Vlothuizen, A. Bruno, B. M. Terhal, and L. DiCarlo, Logical-qubit operations in an error-detecting surface code, *Nature Physics* **18**, 80 (2022).
 - [37] E. Magesan, J. M. Gambetta, B. R. Johnson, C. A. Ryan, J. M. Chow, S. T. Merkel, M. P. da Silva, G. A. Keefe, M. B. Rothwell, T. A. Ohki, M. B. Ketchen, and M. Steffen, Efficient Measurement of Quantum Gate Error by Interleaved Randomized Benchmarking, *Physical Review Letters* **109**, 080505 (2012).
 - [38] J. Combes, C. Granade, C. Ferrie, and S. T. Flammia, Logical Randomized Benchmarking, ArXiv e-prints 10.48550/arXiv.1702.03688 (2017), 1702.03688.
 - [39] D. Gottesman, The Heisenberg Representation of Quantum Computers, arXiv 10.48550/arXiv.quant-ph/9807006 (1998), quant-ph/9807006.
 - [40] J. Zhang, Y.-C. Wu, and G.-P. Guo, Facilitating practical fault-tolerant quantum computing based on color codes, *Physical Review Research* **6**, 033086 (2024).
 - [41] C. Jones, P. Brooks, and J. Harrington, Gauge color codes in two dimensions, *Physical Review A* **93**, 052332 (2016).
 - [42] Y. Ye, T. He, H.-L. Huang, Z. Wei, Y. Zhang, Y. Zhao, D. Wu, Q. Zhu, H. Guan, S. Cao, F. Chen, T.-H. Chung, H. Deng, D. Fan, M. Gong, C. Guo, S. Guo, L. Han, N. Li, S. Li, Y. Li, F. Liang, J. Lin, H. Qian, H. Rong, H. Su, S. Wang, Y. Wu, Y. Xu, C. Ying, J. Yu, C. Zha, K. Zhang, Y.-H. Huo, C.-Y. Lu, C.-Z. Peng, X. Zhu, and J.-W. Pan, Logical Magic State Preparation with Fidelity beyond the Distillation Threshold on a Superconducting Quantum Processor, *Physical Review Letters* **131**, 210603 (2023).
 - [43] 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).
 - [44] Y. Kim, M. Sevier, and M. Usman, Magic State Injection on IBM Quantum Processors Above the Distillation Threshold, ArXiv e-prints 10.48550/arXiv.2412.01446 (2024), 2412.01446.
 - [45] 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).
 - [46] 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, *Physical Review X* **11**, 041058 (2021).
 - [47] 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).
 - [48] M. A. Nielsen, A simple formula for the average gate fidelity of a quantum dynamical operation, *Physics Letters A* **303**, 249 (2002).
 - [49] B. Schumacher, Sending entanglement through noisy quantum channels, *Physical Review A* **54**, 2614 (1996).
 - [50] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. (Cambridge University Press, Cambridge, UK, 2010).