



This is a repository copy of *Reducing the error rate of a superconducting logical qubit using analog readout information*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/219554/>

Version: Published Version

Article:

Ali, H. orcid.org/0000-0002-7454-0910, Marques, J. orcid.org/0000-0001-9939-051X, Crawford, O. et al. (7 more authors) (2024) Reducing the error rate of a superconducting logical qubit using analog readout information. *Physical Review Applied*, 22 (4). 044031. ISSN 2331-7043

<https://doi.org/10.1103/physrevapplied.22.044031>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Reducing the error rate of a superconducting logical qubit using analog readout information

Hany Ali^{1,†}, Jorge Marques¹, Ophelia Crawford,² Joonas Majaniemi², Marc Serra-Peralta³, David Byfield², Boris Varbanov,³ Barbara M. Terhal³, Leonardo DiCarlo¹ and Earl T. Campbell^{2,4,*}

¹QuTech and Kavli Institute of Nanoscience, Delft University of Technology, P.O. Box 5046, Delft 2600 GA, Netherlands

²Riverlane, Cambridge CB2 3BZ, United Kingdom

³QuTech, Delft University of Technology, P.O. Box 5046, Delft 2600 GA, Netherlands

⁴Department of Physics and Astronomy, University of Sheffield, Sheffield S3 7RH, United Kingdom

 (Received 20 March 2024; revised 24 June 2024; accepted 6 September 2024; published 11 October 2024)

Quantum error correction enables the preservation of logical qubits with a lower logical error rate than the physical error rate, with performance depending on the decoding method. Traditional decoding approaches rely on the binarization (“hardening”) of readout data, thereby ignoring valuable information embedded in the analog (“soft”) readout signal. We present experimental results showcasing the advantages of incorporating soft information into the decoding process of a distance-3 ($d = 3$) bit-flip surface code with flux-tunable transmons. We encode each of the 16 computational states that make up the logical state $|0_L\rangle$, and protect them against bit-flip errors by performing repeated Z -basis stabilizer measurements. To infer the logical fidelity for the $|0_L\rangle$ state, we average across the 16 computational states and employ two decoding strategies: minimum-weight perfect matching and a recurrent neural network. Our results show a reduction of up to 6.8% in the extracted logical error rate with the use of soft information. Decoding with soft information is widely applicable, independent of the physical qubit platform, and could allow for shorter readout durations, further minimizing logical error rates.

DOI: [10.1103/PhysRevApplied.22.044031](https://doi.org/10.1103/PhysRevApplied.22.044031)

I. INTRODUCTION

Small-scale quantum error correction (QEC) experiments have made significant progress over recent years, including fault-tolerant logical-qubit initialization and measurement [1,2], correction of both bit- and phase-flip errors in a distance-3 ($d = 3$) code [3–5], magic state distillation beyond break-even fidelity [6], suppression of logical errors by scaling a surface code from $d = 3$ to $d = 5$ [5], and demonstration of logical gates [7]. The performance of these logical-qubit experiments across various qubit platforms is dependent on the fidelity of physical quantum operations, the chosen QEC codes and circuits,

and the decoders used to process QEC readout data. Common decoding approaches with access to analog information often rely on digitized (binary) qubit readout data as input to the decoder. The process of converting a continuous measurement signal to binary outcomes inevitably leads to a loss of information that reduces decoder performance.

Pattison *et al.* [8] have proposed a method for incorporating this analog “soft” information in the decoding of QEC experiments, suggesting a potential 25% improvement in the threshold compared to decoding with “hard” (binary) information. The advantage of using soft information has also been demonstrated on simulated data with neural-network (NN) decoders [9,10]. Soft-information decoding has also been realized for a single physical qubit measured via an ancilla in a spin-qubit system [11] and for a superconducting-based QEC experiment with a simple error model assuming uniform qubit quality [12]. The incorporation of soft information with variable qubit fidelity can in theory provide further benefit when decoding experimental data. However, this can be challenging, as additional noise sources (e.g., leakage and other non-Markovian effects) add complexity; therefore,

*Contact author: earlrcampbell@gmail.com

†Present address: Quantware B.V., Elektonicaweg 10, Delft 2628 XG, Netherlands.

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

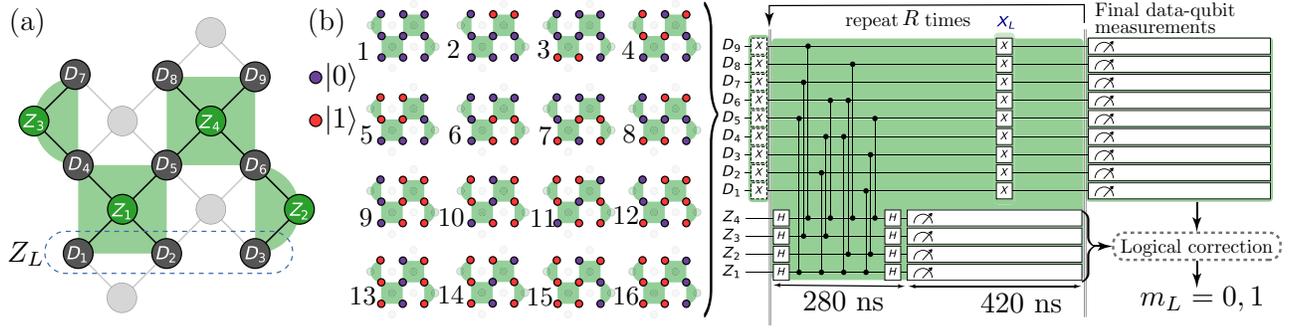


FIG. 1. The Surface-13 QEC experiment. (a) The device layout, with vertices indicating flux-tunable transmons and edges denoting nearest-neighbor coupling via fixed-frequency resonators. Nine data qubits in a 3×3 array (labeled D , dark gray) are subject to four Z -basis parity checks realized using ancilla qubits (green). Light gray vertices and edges are not used. (b) The quantum circuit for the experiment over R QEC rounds, each round taking 700 ns. During ancilla measurement, an X_L operation implemented transversally with π pulses on all data qubits is applied to average the logical error over the logical subspace. We show the 16 computational states over which we average. We employ various methods in postprocessing to decode the measurements and determine the value of the logical observable, m_L [13].

the advantage of decoding with soft information is not guaranteed.

In this paper, we demonstrate the use of soft information in the decoding of data obtained from a bit-flip $d = 3$ code in a 17-qubit device using flux-tunable transmons with fixed coupling. Unlike a typical $d = 3$ surface code, we repeatedly measure only Z -basis stabilizers, utilizing 13 out of the 17 qubits in the device [Fig. 1(a)]. This approach allows us to avoid problematic two-qubit gates between specific pairs of qubits that have strong interactions of the qubits with two-level-system (TLS) defects [14]. We refer to this experiment as Surface-13. We encode and stabilize each of the 16 computational basis states, shown in Fig. 1(b), that are eigenstates of the Z -basis stabilizers and the logical operator, Z_L , with eigenvalues $+1$. We approximate the performance of the full logical state by averaging across these states. The code protects the logical state from single bit-flip errors, similar to the $d = 3$ repetition code [15,16]. We employ two decoding strategies: a minimum-weight perfect-matching (MWPM) decoder and a recurrent NN decoder. For each strategy, we compare the performances of two variants: one with soft information and one without. With soft information, the extracted logical error rates are reduced by 6.8% and 5% for the MWPM and NN decoders, respectively.

II. EXPERIMENT CONFIGURATION

The experimental procedure begins by preparing the data-qubit register in one of the 16 physical computational states, as shown in Fig. 1(b). Next, repeated Z -basis stabilizer measurements are performed over a varying number of QEC rounds, R . Each round takes 700 ns, with 20 ns and 60 ns for single- and two-qubit gates, respectively, and 420 ns for readout. The logical state is flipped during

the ancilla measurement in each QEC round using the $X_L = X^{\otimes 9}$ transversal gate to symmetrize the effect of relaxation (T_1) errors, minimizing the dependence on the input state. A final measurement of all data qubits is used to determine the observed logical outcome m_L and compute final stabilizer measurements. The physical error rates of single-qubit gates, two-qubit gates, and readout are 0.1%, 1.6%, and 1.2%, respectively, averaging over the 13 qubits and 12 two-qubit gates used in the experiment. Further details about the device, calibration, and parity-check benchmarking are provided in Appendices A and B.

The decoder determines whether the outcome m_L needs to be corrected (flipped) based on the values of combinations of certain measurements (see below) and decoding success is declared if this corrected readout, m_L , matches the prepared state. We calculate F_L , the logical fidelity, for a fixed $R \in \{1, 2, 4, 8, 16\}$ and each input state as the fraction of successfully decoded runs. Finally, F_L is averaged over the 16 physical computational states, approximating the logical performance of the $|0_L\rangle$ state.

Qubit readout is performed by probing the state-dependent transmission of a dedicated dispersively coupled readout-resonator mode to infer the qubit state, $|j\rangle$ [17]. The readout pulse for each qubit has a rectangular envelope softened by a Gaussian filter of width $\sigma = 0.5$ ns (for additional details about readout calibration, see Appendix A). After amplification [18], the transmitted signal is down-converted to an intermediate frequency and the in-phase and in-quadrature (IQ) components integrated over 420 ns using optimal weight functions [19–21]. The resulting two numbers, I and Q , form the IQ signal $z = (I, Q)$ [Fig. 2(a)] comprising the soft information. The readout-pulse envelope and signal integration are performed using Zürich Instruments UHFQA analyzers sampling at 1.8 GSa/s.

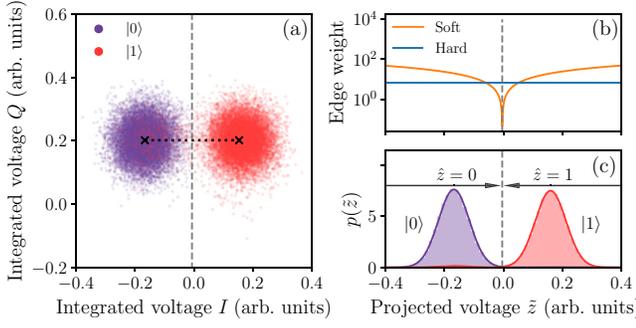


FIG. 2. (a) The measurement response of the $|0\rangle$ and $|1\rangle$ states in IQ space for data qubit D_6 , showing a projection line that connects the means of the two Gaussian peaks (black dotted line). (b) The edge weight as a function of the projected voltage \tilde{z} for soft and hard measurements (see Eq. (1)). Measurement errors are most likely in the region $\tilde{z} \approx 0$, where the edge weight is minimized. (c) The histogram and fitted probability density function (PDF) $P(\tilde{z} | j)$ for state preparations $j \in \{0, 1\}$.

III. CAPTURING SOFT INFORMATION

To transform an IQ signal $z \in \mathbb{R}^2$ to a binary-measurement outcome $\hat{z} \in \{0, 1\}$, we apply a hardening map, which we choose as the maximum-likelihood assignment. The hardened outcome is obtained by choosing $\hat{z} = 0$ if $P(0 | z) > P(1 | z)$ and $\hat{z} = 1$ otherwise, where $P(j | z)$ is the probability that the qubit was in state $|j\rangle$ just before the measurement, given the observed IQ value z . Assuming that the states $|0\rangle$ and $|1\rangle$ are equally likely, one has $P(j | z) \propto P(z | j)$. Therefore, $\hat{z} = 1$ if $P(z | 1) > P(z | 0)$ and $\hat{z} = 0$ otherwise. When we consider the $|2\rangle$ state, we assign the hardened-measurement outcome j to correspond to the largest $P(z | j)$.

The probability density functions (PDFs) denoted $P(z | j)$ are combinations of two and three Gaussians when we do and do not consider the $|2\rangle$ state, respectively. We find that this heuristic model works well for both two-state and three-state discrimination. To determine the fit parameters of the Gaussian model, we use 1.3×10^5 calibration shots per state preparation $j \in \{0, 1, 2\}$ for each of the 13 transmons. To reduce the dimension of the problem in the case in which we do not consider the $|2\rangle$ state, we project the IQ voltages to $\tilde{z} \in \mathbb{R}$ along the axis of symmetry [black dotted line in Fig. 2(a)]. We obtain the PDFs of the projected data $P(\tilde{z} | j)$ by decomposing $z = (\tilde{z}, z_\perp)$ to a parallel component \tilde{z} and a perpendicular component z_\perp , giving $P(\tilde{z} | j) = \int P(\tilde{z}, z_\perp | j) dz_\perp$. Assuming that the IQ responses of the computational basis measurements are symmetric along the axis joining the two centroids [marked by black crosses in Fig. 2(a)], the projection does not result in information loss. The hardened-measurement outcomes are then obtained by comparing $P(\tilde{z} | 0)$ and $P(\tilde{z} | 1)$. Further information on our classification methods is given in Appendix E 1.

IV. MINIMUM-WEIGHT PERFECT-MATCHING DECODING

In QEC experiments, the detectors [22] are selected combinations of binary-measurement outcomes that have deterministic values in the absence of errors. A detector the value of which has flipped from the error-free value is a defect. A decoder takes observed defects in a particular experiment and, using a model of the possible errors and the defects they result in, calculates the logical correction. In Surface-13, assuming circuit-level Pauli noise and with detectors defined as described below, each error results in at most two defects. As a result, it is possible to represent potential errors as edges in a graph—the decoding graph—with the nodes on either end representing the defects caused by the error (with a virtual node added for errors that only lead to a single defect). A matching decoder can thus be used, which matches pairs of observed defects along minimum [23–25] or near-minimum-weight [26] paths within the graph and thereby approximately finds the most probable errors that cause the observed defects. From the most probable errors, one can deduce whether a logical correction is necessary.

Typically, QEC experiments are described assuming that ancilla qubits are reset following their measurement in every QEC round. However, this is not the case in our experiment. Nevertheless, without resetting qubits, suitable detectors can be chosen as $d_{i,r} = \hat{z}_{i,r} \oplus \hat{z}_{i,r-2}$, where $\hat{z}_{i,r} \in \{0, 1\}$ is the (hardened) measurement outcome of ancilla i in round r (for details, see Appendix C and Ref. [27]). We note that in our case, the error-free detector values are always 0. A key difference with the midcircuit-reset case is that ancilla-qubit errors (that change the qubit state) and measurement-classification errors (where the inferred hardened measurement does not match the true qubit state) have different defect signatures, apart from in the final round (see Appendix C). The structure of the decoding graph for the four-round experiment can be seen in Fig. 3(a).

To construct the decoding graph, one can define the probability of different error mechanisms and use a software tool such as STIM [28]. As we do not have direct knowledge of the noise, this graph may not accurately capture the true device noise. Therefore, we use a pairwise-correlation method [29–31] to construct the graph for the MWPM decoder, whereby the decoding-graph edge probabilities are inferred from the frequency of observed defects in the experimental data. In particular, this approach enables us to account for varying fidelities between different qubits. However, the pairwise-correlation method is susceptible to numerical instabilities that we stabilize using a “noise-floor graph” [31], as described in Appendix C 1. This is a crucial advance over previous experimental demonstrations of soft-information decoding [12], where a

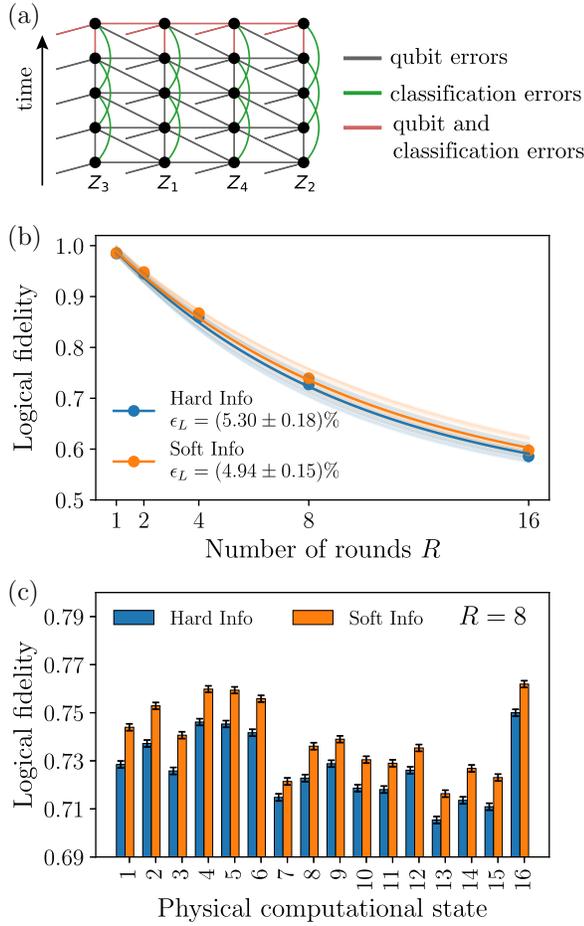


FIG. 3. (a) The decoding graph, showing different types of error mechanisms. The labels indicate the ancilla qubits associated with the detectors in each column. The soft MWPM decoder dynamically updates the weights of edges highlighted in green and red. (b) The logical fidelity of the MWPM decoder as a function of the number of rounds R , shown for each physical computational state that makes up $|0_L\rangle$ (transparent curves) and averaged across all states (opaque curves). (c) The logical fidelity at $R = 8$, shown for each initial computational state as indexed in Fig. 1(b).

graph derived from STIM with uniform qubit fidelities has been used.

To use soft information with a MWPM decoder, we follow Ref. [8]. The edge corresponding to a measurement-classification error is given a weight

$$w = -\log \left[\frac{P'(\hat{z} | 1 - \hat{z}')}{P'(\hat{z} | \hat{z}')} \right], \quad (1)$$

where \hat{z}' is the inferred state after measurement. This is found by taking $\hat{z}' = 1$ if $P'(\hat{z} | 1) > P'(\hat{z} | 0)$ and 0 otherwise. The PDFs $P'(\hat{z} | j')$ are obtained by keeping only the dominant Gaussian in the measurement PDFs—this is to avoid including ancilla-qubit errors during measurement in the classification-error edge. Therefore, to incorporate

soft information, we replace the weights calculated for the classification-error edges using the pairwise-correlation method with the weights from Eq. (1). This procedure is appropriate in all rounds but the final round of ancilla- and data-qubit measurements, where both qubit errors and classification errors have the same defect signature. In these cases, we instead: (i) calculate the mean classification error for each measurement by averaging the per-shot errors; (ii) calculate the mean classification error for each edge from those for each measurement; (iii) remove the mean classification error from the edge probability; and (iv) include the per-shot classification error calculated from the soft readout information. Further information is given in Appendix E 2.

V. NEURAL-NETWORK DECODING

Our second decoder—the NN decoder—can learn the noise model during training without making assumptions about it [9,10,32–34]. NNs have flexible inputs that can include leakage or soft information, as well as nonuniform qubit fidelities. This again contrasts our work with previous soft-information decoding experiments that have used a simpler noise model with uniform qubit fidelities [12]. Recent work [9,10] has shown that NNs can achieve similar performance to computationally expensive (tensor network) decoders when evaluated on experimental data for the $d = 3$ and $d = 5$ surface code. Those networks have been trained with simulated data, although the authors of Ref. [10] have done a fine-tuning of their models with approximately 2×10^4 experimental samples (while using 2×10^9 simulated samples for their main training). One may expect that the noise in the training and evaluation data should match to achieve the best performance.

We train a NN decoder on experimental data and study the performance improvement when employing various components of the readout information available from the experiment. We use two architectures for our NN, corresponding to the network from Varbanov *et al.* [9], and a variant of the network that includes encoding layers for handling different types of information [see Fig. 4(a)]. The inputs for our standard NN decoder are the observed defects. For our soft NN decoder, the inputs are the defect probabilities given the IQ values and the leakage flags, one for each ancilla-qubit measurement. A leakage flag l gives information about the qubit being in the computational space, i.e., $l = 1$ if $\hat{z} = 2$ and $l = 0$ otherwise, where \hat{z} is the hardened value of z using the three-state classifier. For the final round when all data qubits are measured, we do not provide the decoder with any soft information to ensure that we do not make the task for the decoder deceptively simple [10]; this is a drawback of running the decoder only on the logical $|0_L\rangle$ state and not on randomly chosen $|0_L\rangle$ or $|1_L\rangle$ (see the discussion in Appendix D 2).

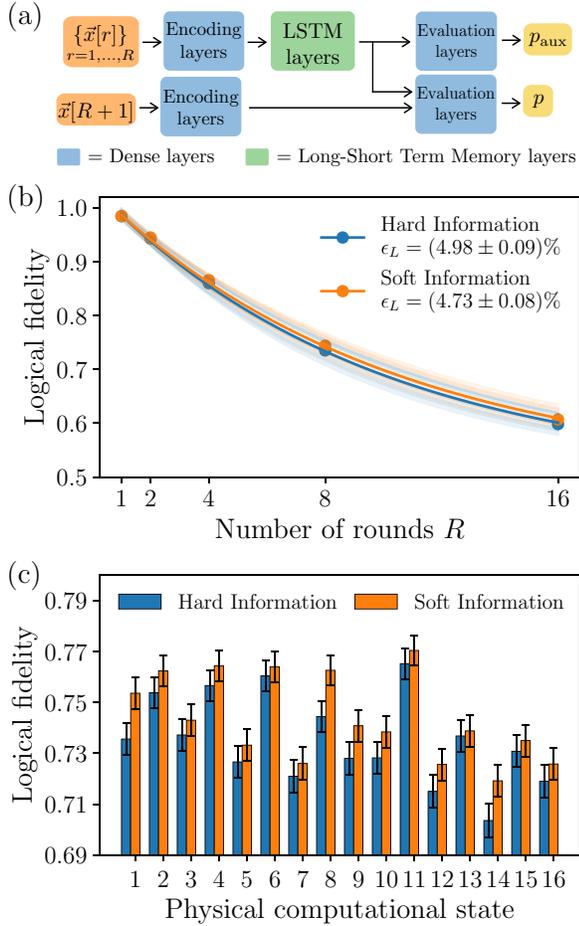


FIG. 4. (a) Our NN architecture, a variant of Ref. [9]. The input $\bar{x}[r]$ for the soft NN contains the defect probability and leakage flag data of round r , while for the (hard) NN it only contains the defect data. The vector $\bar{x}[R+1]$ contains the final-round defects. The output p is the estimated probability that a logical error has happened and p_{aux} is only used to help the training (see Appendix D 1). (b) The logical fidelity of the NN decoder as a function of the number of rounds R , shown for each physical computational state that makes up $|0_L\rangle$ (transparent curves) and averaged across all states (opaque curves). (c) The logical fidelity at $R=8$, shown for each initial computational state as indexed in Fig. 1(b). LSTM layer refers to a long short-term memory layer.

Due to the richer information of soft inputs, we can use larger networks than in the (hard-)NN case without encountering overfitting issues during training. The network performance when given different amounts of soft information is included in Appendix D 1, showing that the larger the amount, the better is the logical performance. We follow the same training as Ref. [9] but with some different hyperparameters; and we use the ensembling technique from machine learning to improve the network performance without a time cost but at a computing-resource cost [35].

VI. LOGICAL PERFORMANCE

The extracted logical fidelity as a function of R for the MWPM and NN decoders is shown in Figs. 3 and 4, respectively. The results are presented for the 16 physical computational states (transparent lines) and for their average (opaque lines), approximating the logical performance of the $|0_L\rangle$ state. To find a logical error rate, ϵ_L , the decay of logical fidelity is fitted using the model

$$\tilde{F}_L(R) = \frac{1}{2} [1 + (1 - 2\epsilon_L)^{R-R_0}], \quad (2)$$

where \tilde{F}_L indicates the fitted fidelity to the measured F_L and R_0 is a round offset parameter [13]. For both decoders, including soft readout information enhances the logical performance, resulting in the reduction of ϵ_L by 6.8% and 5.0% for MWPM and NN, respectively. We note that the error bars are different in the two cases due to the differing ways of splitting the data set of approximately 9×10^4 samples per round and initial state. With the MWPM decoder, we use half the data to perform the pairwise-correlation method to obtain the decoding graph and half for obtaining the logical error probability. We then swap the data halves and average the logical fidelities, thereby using every shot to obtain the overall logical fidelity. With the NN decoder, 95% of samples are used for training and validation and only 5% of samples are used to estimate logical fidelities, leading to larger error bars.

VII. SUMMARY AND CONCLUSIONS

We have experimentally demonstrated the benefits of using soft information in the decoding of a $d=3$ bit-flip surface code, utilizing 13 qubits. We have combined soft-information decoding with techniques that learn and account for variable qubit and gate fidelities, distinguishing our work from previous experiments [12]. With soft information, the NN decoder achieves the best extracted logical error rate of 4.73%. It is crucial to note that as we do not measure the X -basis stabilizers of the typical distance-3 surface code, the extracted ϵ_L is likely underestimated (compared with ϵ_L of 5.4% for the standard $d=3$ surface code by Ref. [3] without leakage postselection). However, the nature of the decoding problem will be the same and will benefit from the decoding optimizations explored in this paper.

Despite the modest improvement in this work, simulations [8,10] suggest further advantages of soft-information decoding: improvement in the error correction threshold and increased suppression of logical errors as the code distance increases. Implementation of a leakage-aware decoder [36] could also potentially enhance the logical performance with MWPM but this has yet to be explored in experiment. With the NN decoder, it is unclear if the defect probabilities and leakage flags are the optimal way to

present the information to the network and this could be the subject of further investigation. Finally, our experiments have utilized readout durations that have been optimized for measurement fidelity. Calibrating the measurement duration for optimal logical performance [8] instead could potentially lead to higher logical performance.

The source code of the NN decoder and the script to replicate the results are available in Ref. [37] and Ref. [38], respectively. The script to analyze the experimental data can be found in Ref. [38].

ACKNOWLEDGMENTS

This research is supported by the OpenSuperQPlus100 project (no. 101113946) of the European Union Flagship on Quantum Technology (HORIZON-CL4-2022-QUANTUM-01-SGA), the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the U.S. Army Research Office Grant No. W911NF-16-1-0071, and QuTech NWO funding 2020–2024—Part I “Fundamental Research” with Project no. 601.QT.001-1 and by the Allowance for Top Consortia for Knowledge and Innovation (TKIs) of the Dutch Ministry of Economic Affairs. We acknowledge the use of the DelftBlue supercomputer [39] for the training of the NNs. We thank G. Calusine and W. Oliver for providing the traveling-wave parametric amplifiers used in the readout amplification chain. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government.

H.A. and J.F.M. calibrated the device and performed the experiment and data analysis. D.B., J.M., and O.C. performed and optimized the MWPM decoding. M.S.P. performed and optimized the NN decoding. E.T.C. and B.M.T. supervised the theory work. L.D.C. supervised the experimental work. E.T.C., H.A., J.M., M.S.P. and O.C., wrote the manuscript, with contributions from B.M.V., B.M.T., D.B., and J.F.M. and feedback from all coauthors.

APPENDIX A: DEVICE OVERVIEW

Our 17-transmon device [Fig. 5(a)] consists of a two-dimensional (2D) array of nine data qubits and eight ancilla qubits, designed for the distance-3 rotated surface code. The qubit transition frequencies are organized into three frequency groups: high-frequency qubits (red), midfrequency qubits (blue and green), and low-frequency qubits (pink), as required for the pipelined QEC cycle proposed in Ref. [43]. Each transmon has a microwave drive line (orange) for single-qubit gates, a flux-control line (yellow) for two-qubit gates, and a dedicated pair of resonator modes (purple) distributed over three feed

lines (blue) for fast dispersive readout with Purcell filtering [17,42]. Nearest-neighbor transmons are coupled via dedicated coupling resonators (sky blue) [44]. Grounding air bridges (light gray) are fabricated across the device to interconnect the ground planes and to suppress unwanted modes of propagation. These air bridges are also added at the short-circuited end of each readout and Purcell resonator, allowing postfabrication frequency trimming [45]. After biasing all transmons to their flux sweet spot, the measured qubit frequencies clearly exhibit three distinct frequency groups, as depicted in Fig. 5(b). These values are obtained from standard qubit spectroscopy. The average relaxation (T_1) and dephasing ($T_{2,\text{echo}}$) times of the 13 qubits used in the experiment are 23 μs and 20 μs , respectively [Fig. 5(d)].

To counteract drift in optimal control parameters, we automate recalibration using dependency graphs [46]. The method, nicknamed graph-based tuneup (GBT) [47], is based on Ref. [48]. Single-qubit gates are autonomously calibrated with DRAG- (derivative removal by adiabatic gate) type pulses to avoid phase error and to suppress leakage [49,50] and benchmarked using single-qubit randomized-benchmarking protocols [51]. The average error of the calibrated single-qubit gates [Fig. 5(c)] across 13 qubits reaches 0.1%, with a leakage rate of 10^{-4} . All single-qubit gates have 20-ns duration.

Two-qubit controlled-Z (CZ) gates are realized using sudden net-zero flux pulses [52]. The QEC cycle in this experiment requires 12 CZ gates executed in four steps, each step performing three CZ gates in parallel. This introduces new constraints compared to tuning an individual CZ gates. For instance, parallel CZ gates must be temporally aligned to avoid overlapping with unwanted interaction zones on the way to, from, or at the intended avoided crossings. Moreover, simultaneous operations in time (vertical) and space (horizontal) may induce extra errors due to various crosstalk effects, such as residual ZZ coupling, microwave cross-driving, and flux crosstalk. To address these nontrivial errors, we introduce two main calibration strategies into a GBT procedure: vertical and horizontal calibrations (VCs and HCs). These tune simultaneous CZ gates in time and space as block units [53]. This approach absorbs some of the flux and residual-ZZ crosstalk errors. After calibration, GBT benchmarks the calibrated gates with two-qubit interleaved randomized-benchmarking protocols with leakage modification [40,41]. The individual benchmarking of the 12 CZ gates reveals an average error of 1.6% with a 0.24% leakage. All CZ gates have 60-ns duration.

Readout calibration is performed in three main steps, realized manually and not with a GBT procedure. In the first, readout spectroscopy is performed at fixed pulse duration (200–300 ns) to identify the optimal frequency maximizing the distance between the two complex transmission vectors $S_{21}^{(0)}$ and $S_{21}^{(1)}$ in the IQ plane. The second

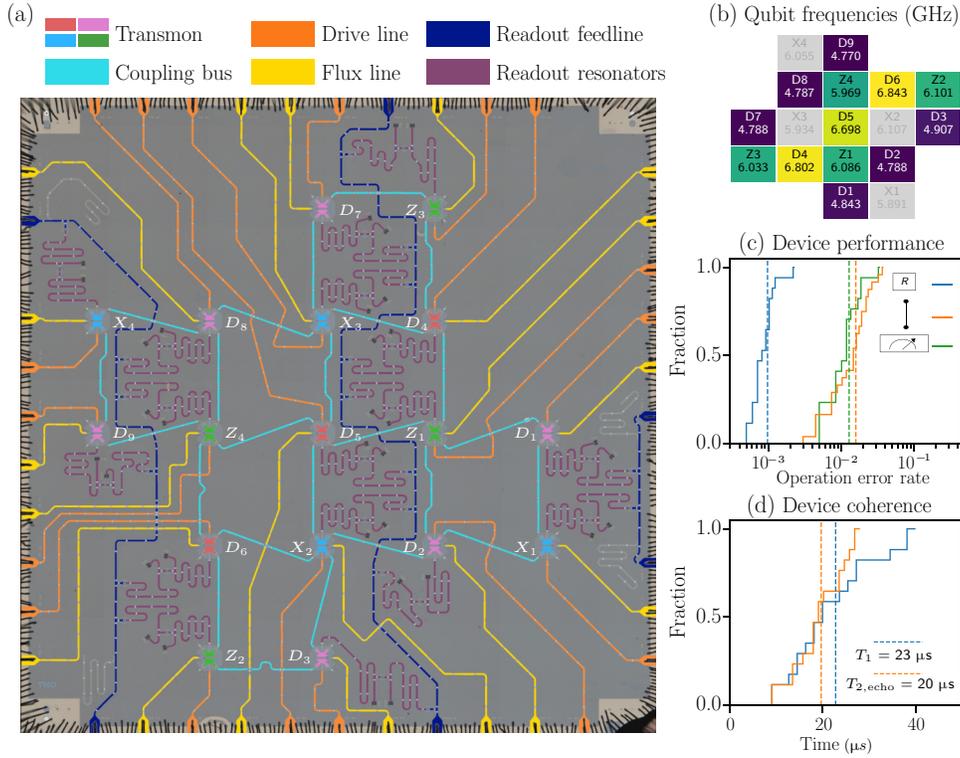


FIG. 5. The device characteristics. (a) An optical image of the 17-transmon device, with added false color to emphasize different circuit elements. The device is connected to a printed circuit board using aluminum wire bonds, visible at the edges of the image. (b) The measured qubit transition frequencies, with all transmons biased to their flux sweet spot. X -basis ancilla qubits (light gray) are not used in this experiment. (c) The cumulative distribution of error rates for single- and two-qubit gates, obtained by randomized-benchmarking protocols with modifications to quantify leakage [40,41], and average readout-assignment fidelities, extracted from single-shot readout histograms [42]. (d) The cumulative distribution of the measured qubit relaxation time T_1 and the echo dephasing time $T_{2,\text{echo}}$. The dashed lines in (c) and (d) indicate the average over the 13 qubits used and the 12 two-qubit gates.

step involves a 2D optimization over the pulse frequency and the amplitude. The goal is to determine readout-pulse parameters that minimize a weighted combination of readout-assignment error (ϵ_{RO}) and measurement quantum nondemolition (QND) probabilities (P_{QND} and $P_{\text{QND}\pi}$). These probabilities are obtained using the method of Ref. [54]. The final step verifies whether photons are fully depleted from the resonator within the target total readout time, 420 ns, using an ALLXY gate sequence between two measurements [55], where an ALLXY sequence is a gate consists of 21 sequences, each comprised of one pair of X and Y pulses. By comparing the ALLXY pattern obtained to the ideal staircase, we can determine whether the time dedicated for photon depletion is sufficient to not affect follow-up gate operations.

After calibrating optimal readout integration weights [55], we proceed to benchmark various readout metrics such as ϵ_{RO} and standard readout QND (F_{QND}) using the measurement butterfly technique [56]. The average ϵ_{RO} [Fig. 5(c)] is 1.2%, extracted from the single-shot histograms. We also perform simultaneous multiplexed readout of all 13 qubits, constructing

assignment-probability and cross-fidelity matrices [17]. The average multiplexed readout error rate is 1.6%, indicating that readout crosstalk is small. Moreover, the average F_{QND} for the four Z -basis ancillas is 95.3% considering a three-level transmon [56]. This also yields an average leakage rate due to ancilla measurement of 0.14%, predominantly from $|1\rangle$.

APPENDIX B: BENCHMARKING OF PARITY CHECKS

With the individual building blocks calibrated, we proceed to calibrate the four Z -basis stabilizer measurements as parallel block units using VC and HC strategies, as discussed in Appendix A. The average probabilities of correctly assigning the parity operator $\Pi_i Z_i$ are measured as a function of the input computational states of the data-qubit register. The measured probabilities [Fig. 6(a), solid blue bars] are compared with the ideal ones (black wire frame) to obtain average parity-assignment fidelities of 96.3%, 92.8%, 89.9%, and 92.3% for Z_3 , Z_2 , Z_1 ,

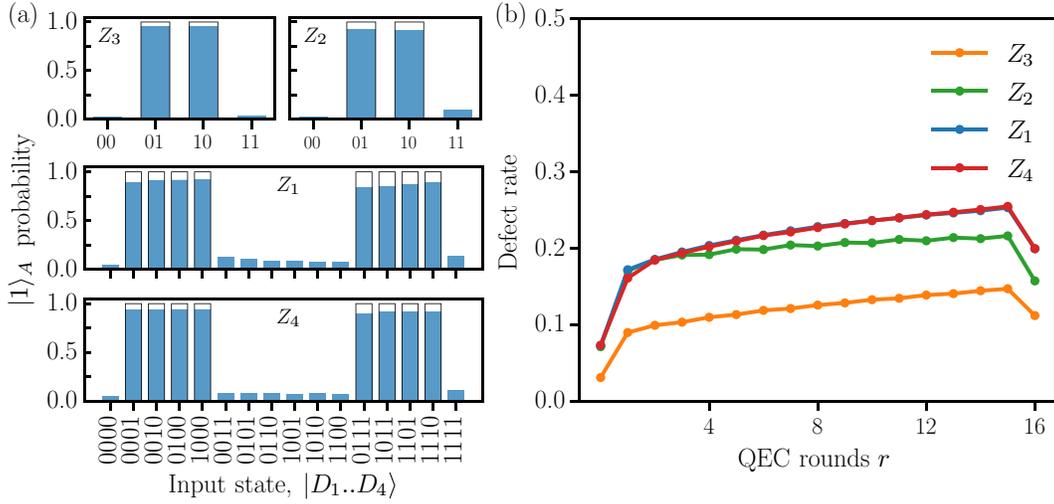


FIG. 6. Parity-check benchmarking. (a) The benchmarking of the assignment fidelity for four stabilizer measurements: $Z_{D_4}Z_{D_7}$, $Z_{D_6}Z_{D_3}$, $Z_{D_4}Z_{D_5}Z_{D_2}Z_{D_1}$, and $Z_{D_9}Z_{D_9}Z_{D_5}Z_{D_6}$. (b) The average defect rate as a function of the QEC rounds for each of the four Z -basis stabilizers across the 16 input states.

and Z_4 , respectively. These results are obtained after mitigating residual excitation effects by postselection on a premeasurement [56].

The defect rates (for the definition of a defect, see Sec. III), reflect the incidence of physical qubit errors (bit-flip and readout errors) detected throughout the rounds r , where $r \in \{1, 2, \dots, R\}$. For each of the four Z -basis stabilizers, the defect rate [Fig. 6(b)] is presented over 16 QEC rounds and averaged across the 16 physical computational input states. The sharp increase in the defect rate between rounds $r = 1$ and $r = 2$ is due to the low initialization error rates and the detection of errors occurring during the ancilla-qubit measurements in the first round. At the boundary round $r = 16$, the defects are obtained using the final data-qubit measurements, which, given the low readout error rates, lead to the observed decrease in the defect rate. Over rounds, defect rates gradually build up, until leveling at approximately 15%, 22%, 25%, and 25% for Z_3 , Z_2 , Z_1 , and Z_4 , respectively. The build-up may be due to leakage [3,5,31,56].

APPENDIX C: DECODING GRAPH

In this appendix, we describe in further detail the effect of not using midcircuit resets on the decoding graph. Here, we consider specifically Surface-13 but similar ideas apply to the full surface code and other stabilizer codes.

As discussed in Sec. IV, in order to detect errors in stabilizer codes, it is typical to define detectors, combinations of binary-measurement outcomes that have deterministic values in the absence of errors. We refer to detectors the value of which has flipped from the expected error-free value as

defects. A decoder takes observed defects in a particular experiment and, using a model of the possible errors and the defects they result in, predicts how the logical state has been affected by the errors. It is desirable for errors to result in a maximum of two defects, as this enables a matching decoder to be used, which can efficiently find the most probable [23–25] errors that cause the observed defects.

In Surface-13, the stabilizers are, in the bulk, weight-4 Z operators. On the boundaries, the stabilizers are of lower weight. In general, each data qubit is involved in two Z -type stabilizers or one on the boundary. The experiment proceeds by using ancilla qubits to measure the stabilizers for some number of rounds, R , where $s_{i,r}$ is the i th stabilizer outcome in the r th round. An X error on a data qubit in round r will change the value of the outcomes of the Z stabilizers involving that data qubit from round r onward. We define the detectors $d_{i,r}$ to be the difference between stabilizer measurements in adjacent rounds, so that

$$d_{i,r} = s_{i,r} \oplus s_{i,r-1}. \quad (C1)$$

We begin Surface-13 with initial data-qubit states that are eigenstates of the Z -type stabilizers with eigenvalues $+1$; therefore, we set $s_{i,0} = 0$. At the end of the experiment, the data qubits are measured in the Z basis; these measurements can be used to construct $s_{i,R+1}$ outcomes for the stabilizers.

With the definition in Eq. (C1), an X -data-qubit error before a QEC round results in a maximum of two defects, as desired. We note that, in general, there may be multiple

errors that result in the same defect signature. The probabilities of these errors are typically combined to give a single edge weight.

We now consider the effect of ancilla-qubit and measurement-classification errors. An ancilla-qubit error in round r on the qubit used to measure the i th stabilizer will change the stabilizer-measurement outcome in the r th round, thus resulting in defects $d_{i,r}$ and $d_{i,r+1}$. The effect of classification errors depends on how the stabilizer outcomes are obtained. If the ancilla qubits are reset after measurement, $s_{i,r} = \hat{z}_{i,r}$, where $\hat{z}_{i,r}$ is the hardened-measurement outcome on the i th ancilla at round r , and therefore a classification error in round r results in the same defects as an ancilla-qubit error in round r . However, if the ancilla qubits are not reset after measurement, as is the case in this paper, $s_{i,r} = \hat{z}_{i,r} \oplus \hat{z}_{i,r-1}$. Therefore, with respect to measurements, the detectors are defined as

$$d_{i,r} = \hat{z}_{i,r} \oplus \hat{z}_{i,r-2}, \quad (\text{C2})$$

for $2 \leq r \leq R$. In this case, a classification error in measurement $\hat{z}_{i,r}$ causes defects $d_{i,r}$ and $d_{i,r+2}$. In both cases, a data-qubit measurement-classification error looks like a data-qubit Pauli error between rounds R and $R+1$. We also define $d_{i,1} = \hat{z}_{i,1}$.

In this error model, we assume that two independent events are possible during measurement—an ancilla-qubit error and a classification error. In practice, these are not independent events, as both are affected by T_1 processes. However, should an error occur that causes the qubit to decay from the $|1\rangle$ to the $|0\rangle$ state during measurement, it can either be viewed as an ancilla-qubit error before measurement (if the inferred hardened-measurement outcome is 0) or an ancilla-qubit error after measurement (if the inferred hardened-measurement outcome is 1). Therefore, these coupled events can be viewed as a single ancilla-qubit error. We note, however, that such qubit errors are not symmetric and thus using a single edge weight is an approximation.

We note that we have not discussed midround qubit errors that result in so-called hook errors, as we have focused on explaining the differences between the decoding graphs with and without midcircuit reset. The hook errors in the two cases will be identical.

1. Noise-floor graph

As discussed in the main text, the pairwise-correlation method can be subject to numerical instabilities. We thus use a “noise-floor graph” that has specific values for each edge in the decoding graph. These instabilities can arise due to the finite data and the approximation that there are no error mechanisms resulting in more than two defects, such as leakage. The impact is more pronounced at the boundary of the code lattice where single defects occur [57].

The lower-bound error parameters used to construct the noise-floor decoding graph are given in Table I. The operation times are taken to be the same as the real device. We note, however, that the other parameters are not the same as those stated in Appendix A. This is because the parameters here set a lower bound on the error probabilities and should only be used when the pairwise-correlation method gives unfeasibly low values. While extensive exploration of the parameters has not been undertaken, the ones stated here have been found to give good performance and several other options have not resulted in significant changes to the results.

Each probability, p , is the probability of a depolarizing error after the specified operation. These are defined so that, for single-qubit gates, resets, and measurements, the probability of applying the Pauli error W is given by

$$p_W = \frac{p}{3}, \quad (\text{C3})$$

for $W \in \{X, Y, Z\}$. For two-qubit gates, the probability of applying the Pauli VW is given by

$$p_{VW} = \frac{p}{15}, \quad (\text{C4})$$

where $V, W \in \{I, X, Y, Z\}$, excluding $V = W = I$. Idle noise is incorporated by Pauli twirling the amplitude damping and dephasing channel to give, for an idling duration t [58],

$$p_X(t) = p_Y(t) = \frac{1}{4} (1 - e^{-t/T_1}), \quad (\text{C5})$$

$$p_Z(t) = \frac{1}{2} (1 - e^{-t/T_2}) - \frac{1}{4} (1 - e^{-t/T_1}). \quad (\text{C6})$$

We note that as $T_1 = T_2$ in our model, $p_X(t) = p_Y(t) = p_Z(t)$. The noise-floor graph is derived from the circuit containing the appropriate parameters using STIM [28].

TABLE I. The lower-bound noise parameters used in the experimental-graph derivation. These parameters are used to fix the minimum values of each edge.

Parameter	Value
Single-qubit gate-error probability	0.5×10^{-3}
Two-qubit gate-error probability	5×10^{-3}
Reset-error probability	0.0
Measurement qubit error probability	1×10^{-3}
Measurement classification error probability	1×10^{-3}
T_1	30 μ s
T_2	30 μ s
Single-qubit gate time	20 ns
Two-qubit gate time	60 ns
Measurement time	420 ns

APPENDIX D: DETAILS OF THE NEURAL-NETWORK DECODER

1. NN inputs, outputs, and decoding success

The inputs provided to the NN decoder consist of the defects, the defect probabilities, and the leakage flags [see Fig. 4(a)]. The only elements not described in the main text are the defect probabilities. These are obtained following Ref. [9] and using the two-state readout classifier from the main text and Appendix E 1 a. First, we express the probability of the measured qubit “having been in the state” $|j\rangle$ ($j \in \{0, 1\}$), given the IQ value z , as

$$P(j|z) = \frac{P(z|j)P(j)}{\sum_i P(z|i)P(i)}, \quad (\text{D1})$$

where $P(j)$ is the probability that the qubit was in state $|j\rangle$. We define the *incoming defects* in the bulk as $\tilde{d}_{i,r} = k_{i,r} \oplus k_{i,r-2}$, where $k_{i,r}$ is the state of the ancilla i before its measurement in round r .

Although we do not have access to the incoming defects, we can estimate the probability that the defect $\tilde{d}_{i,r}$ has been triggered given $z_{i,r}$ and $z_{i,r-2}$ (termed the *defect probability*) by

$$P(\tilde{d}_{i,r} = 1|z_{i,r}, z_{i,r-2}) = P(k_{i,r} = 0|z_{i,r})P(k_{i,r-2} = 1|z_{i,r-2}) + P(k_{i,r} = 1|z_{i,r})P(k_{i,r-2} = 0|z_{i,r-2}). \quad (\text{D2})$$

As we can incorrectly infer $k_{i,r}$ from $z_{i,r}$, the defect probabilities include assignment errors. Note that digitizing $P(\tilde{d}_{i,r} = 1|z_{i,r}, z_{i,r-2})$ leads to the “standard” defects defined in the main text [10]. The use of defect probabilities allows us to infer the defect reliability, e.g., $P(k_{i,r} = 0|z_{i,r}) \approx P(k_{i,r} = 1|z_{i,r})$ leads to $P(\tilde{d}_{i,r} = 1|z_{i,r}, z_{i,r-2}) \approx 1/2$, which is in-between 0 and 1 and thus uncertain.

We have used the incoming defects and not the “standard” defects for deriving the defect probabilities because $P(\hat{z}|z)$ is always 0 or 1, as \hat{z} is completely determined by z . In that case, we would not have any soft information and the defect probabilities would correspond to the defects. Remember that we do not use the defect probabilities of the final round, as explained in Appendix D 2, and that we assume that $P(0) = P(1) = 1/2$ when calculating $P(\tilde{d}_{i,r} = 1|z_{i,r}, z_{i,r-2})$.

For completeness, we study the performance of the NNs given four combinations of inputs: (a) defects, (b) defects and leakage flags, (c) defect probabilities, and (d) defect probabilities and leakage flags. The results in Fig. 7 show that the networks can process the richer inputs to improve their performance. The reason for not giving the network the soft-measurement outcomes z as input directly (but the defect probabilities instead) is that we have found that the NN decoder does not perform well on the soft measurements [9]; effectively, the NN has to additionally learn the defects, which is possible with larger NNs, as in Ref. [10].

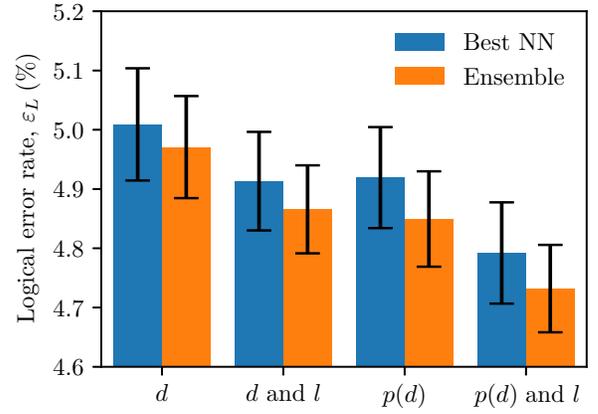


FIG. 7. The logical error rates of the NN decoders when given different inputs: the label d corresponds to defects, l to leakage flags, and the label $p(d)$ corresponds to defect probabilities given the soft information. “Best NN” (blue) and “Ensemble” (orange) correspond to the architectures with the lowest logical error rate for a single NN and for an ensemble of five NNs, respectively, as described in Appendix D 3.

The NN gives two outputs, p and p_{aux} , which correspond to the estimated probabilities that a logical flip has occurred during the given sample. The output p has been calculated using all the information given to the NN, while p_{aux} does not use the final round data [see Fig. 4(a)]. We train the NN based on its accuracy in both p and p_{aux} because the latter helps the NN to not focus only on the final round but decode based on the full QEC data [9,59]. Note that the outputs correspond to physical probabilities (i.e., $p, p_{\text{aux}} \in [0, 1]$) due to using sigmoid activation functions in the last layer of the NN architecture.

To determine whether or not the NN has decoded the QEC data correctly, the output p is used as follows. If $p \geq 1/2$, we set the logical flip bit to $b = 1$; if $p < 1/2$, we set the logical flip bit to $b = 0$. On the basis of the final data-qubit measurements, we compute the (uncorrected) logical Z_L as a bit $z_{\text{out}} \in \{0, 1\}$. We take the logical input state z_{in} (in our experiments, $z_{\text{in}} = 0$ always, as we prepare $|0_L\rangle$ in all cases) and mark the run as successful when $z_{\text{in}} \oplus z_{\text{out}} \oplus b = 0$ and unsuccessful when $z_{\text{in}} \oplus z_{\text{out}} \oplus b = 1$.

2. Learning the final logical measurement

In machine learning, one needs to be careful about the information given to the network. For example, the NN could predict the logical bit-flip correction b without using the defect information gathered over multiple rounds. In particular, given that in our experiment we only start with state $|0_L\rangle$, the NN could just output the value of b such that $b \oplus z_{\text{out}} = 0$. It can do this by learning z_{out} and hence the network should get no explicit information about z_{out} . While it is true that we do not directly provide the data-qubit-measurement outcomes to the NN decoder, there still might be some partial information provided by the defect

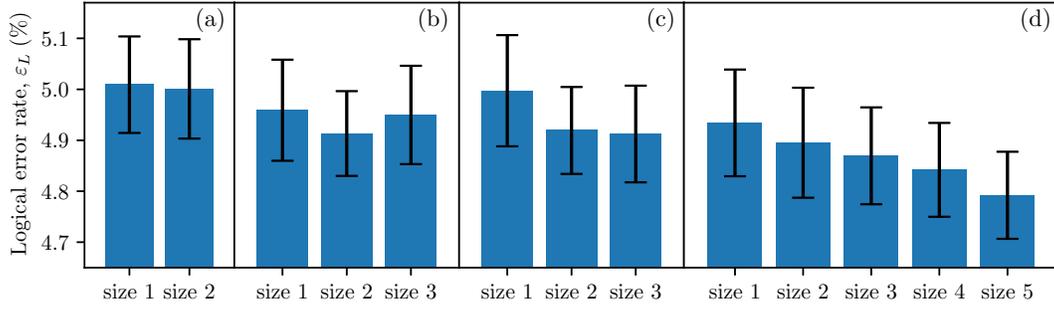


FIG. 8. The logical performance of the NNs given the four input combinations and different network sizes. The NN inputs $\vec{x}[r]$ in (a) are the defects, in (b) the defects and leakage flags, in (c) the defect probabilities, and in (d) the defect probabilities and the leakage flags. The NN sizes are summarized in Table II.

probabilities or the leakage flags, e.g., when a transmon in state $|2\rangle$ is more likely declared as a $|1\rangle$ than a $|0\rangle$ [for transmon Z_2 , see Fig. 10]. Note that this issue does not occur if one randomly trains and validates with either $|0_L\rangle$ or $|1_L\rangle$, since then the optimal $b = z_{\text{in}} \oplus z_{\text{out}}$ and z_{in} is a random bit unknown to the network.

We indeed observe an abnormally high performance for a single network when using soft information in the final round, with $\epsilon_L \sim 4.2\%$ when using defect probabilities and $\epsilon_L \sim 4.1\%$ if we include the leakage flags too. Such an increase suggests that the NN can partially infer z_{out} and thus “knows” how to set b . This phenomenon does not occur when giving only the leakage flags of data-qubit measurements, which instead leads to $\epsilon_L \sim 4.9\%$. Nevertheless, due to the reasons explained above, we have decided to not include the leakage flags in the final round, to be sure that we do not provide any information about z_{out} to the NN.

We note that Ref. [10] has also opted to just give the defects in the final round, because of the possibility that the NN decoder can infer the measured logical outcome from the final-round soft information.

3. Ensembling

Ensembling is a machine-learning technique to improve the network performance without costing more time, as networks can be trained and evaluated in parallel [35]. It consists of averaging the outputs, $\{p_i\}$, of a set of NNs, to obtain a single more accurate prediction, \tilde{p} . One can think that the improvement is due to “averaging out” the errors in the models [35]. In Ref. [10], the authors trained 20 networks with different random seeds and averaged their outputs with the geometric mean. In this work, the output “average,” \tilde{p} , is given by

$$\log\left(\frac{1-\tilde{p}}{\tilde{p}}\right) = \frac{1}{5} \sum_{i=1}^5 \log\left(\frac{1-p_i}{p_i}\right), \quad (\text{D3})$$

where the $\{p_i\}$ are the predictions of five individual NNs of the logical flip probability (see Appendix D 1). This expression follows the approach from the repeated qubit readout with soft information [60], which is optimal if the values are independently sampled from the same distribution. Once \tilde{p} is determined, we threshold it to set the flip bit b as described in Appendix D 1.

4. Network sizes, training hyperparameters, and data set

Due to the different amounts of information in each input, the NNs in Figs. 4 and 7 have different sizes to maximize their performance without encountering overfitting issues. In Fig. 8, the size of the network is increased given a set of inputs until overfitting degrades the performance or there is no further improvement. The specific sizes and hyperparameters of the NNs shown in the figures are summarized in Table II. These hyperparameters are the same as in Ref. [9] but with the following changes: (1) reducing the batch size to avoid overfitting, as the experimental data set is smaller, and (2) decreasing the learning rate for the large NNs. For comparison, the NN in Ref. [10] for a $d = 3$ surface code uses 5.4×10^6 free parameters, a learning rate of 3.5×10^{-4} , and a batch size of 256. The number of free parameters in a NN is related to its capacity to learn and generalize from data, the learning rate is related to the step size at which NN parameters are optimized, and the batch size is the number of training samples used in a single iteration of gradient descent.

The splitting of the experimental data set into the three sets of *training*, *validation*, and *testing* has been done as follows. For each initial state and each number of rounds, (1) randomly pick 5×10^3 samples from the given data and store them in the *testing* data set, (2) randomly select 90% of the remaining samples and store them in the *training* data set, and (3) store the rest in the *validation* data set. The reason for this choice is to ensure that the data sets are not accidentally biased toward an initial state or number of rounds. After the splitting, we have a training data set

TABLE II. The NN sizes and training hyperparameters used in this work. n_i refers to the number of layers in block $i \in \{\text{Enc, LSTM, Eval}\}$ and d_i the dimension of these layers. If n_{Enc} is not specified, the network does not have encoding layers. The blocks are shown in Fig 4(a). The number of free parameters depends on the given input combination but the changes are of the order of approximately 5000.

Label	n_{Enc}	d_{Enc}	n_{LSTM}	d_{LSTM}	n_{Eval}	d_{Eval}	Number of free parameters	Batch size	Learning rate	Dropout rate (%)
Size 1			2	90	2	90	$\sim 115\,000$	64	5×10^{-4}	20
Size 2	2	32	2	100	2	100	$\sim 160\,000$	64	2×10^{-4}	20
Size 3	2	64	2	120	2	120	$\sim 250\,000$	64	2×10^{-4}	22
Size 4	2	90	3	100	2	100	$\sim 285\,000$	64	2×10^{-4}	22
Size 5	2	100	3	100	2	100	$\sim 290\,000$	64	2×10^{-4}	20

consisting of 6.9×10^6 samples, a validation data set with 7.6×10^5 , and a testing data set with 4×10^5 . Note that the NN has been trained and tested on the same number of rounds because the experiment only goes until $R = 16$. However, in longer-memory experiments, the NN should be trained only up to a “low” number of rounds to avoid long periods of training.

The training for each single NN has been carried out on an NVIDIA Tesla V100S GPU and lasted around 10 h for the smallest size and 23 h for the largest one when using the training data set consisting of 6.9×10^6 samples with 7.6×10^5 samples for validation. The evaluation of the network performance has been done on an Intel Core(TM) i7-8650U CPU @ 1.90 GHz $\times 4$. We estimate that it takes approximately $127 \mu\text{s}$ per QEC cycle and approximately 1.77 ms for the final round when running a size-5 NN with soft information (see Fig. 9). The same NN without soft information takes approximately $124 \mu\text{s}$ per QEC cycle and approximately 1.57 ms for the final round. The size-2 NN used in the main text to decode data without soft information takes approximately $82 \mu\text{s}$ per QEC cycle and approximately 1.35 ms for the final round. For comparison, the NN in Ref. [10] for a $d = 3$ surface code takes approximately $20 \mu\text{s}$ to decode a QEC round, but it has been evaluated on a tensor processing unit (TPU).

APPENDIX E: SOFT-INFORMATION PROCESSING

The processing of the soft information [i.e., measurement edge weights in Eq. (1), the defect probabilities in Eq. (D2), and leakage flags], all use the PDFs $P(z | j)$, which can be found from experimental calibration. These PDFs are obtained by fitting a readout model to the readout calibration data, consisting of a set of IQ values for each prepared state $|j\rangle$. The performance of the soft decoders is limited by the accuracy of the readout models used; thus in this section we describe the models employed in this paper and their underlying assumptions about the qubit readout response.

1. Probability density function fits

Each of the 13 transmons in the device has a characteristic measurement response in IQ space, requiring a unique PDF to be fitted for each. In this section, we detail a heuristic model used to classify qubit states (Appendix E 1 a) and qutrit states (Appendix E 1 b), formulated as a linear combination of Gaussian distributions. We utilize this Gaussian-mixture model instead of a physics-derived measurement model such as the soft amplitude-damping model derived in Ref. [8] or the Bayesian approach taken in Ref. [61]. This is because we want to exclude the contribution of ancilla-qubit errors that occur during measurement from the classification-error probability. A Gaussian-mixture fit allows us to classify states according to noisy experimental

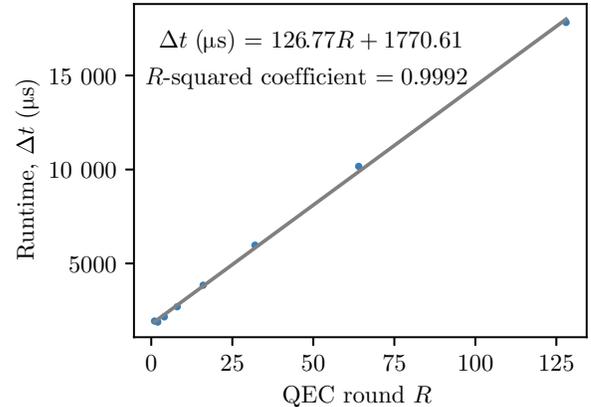


FIG. 9. The evaluation run time for the size-5 network with batch size = 1. The line corresponds to a linear regression where the R -squared coefficient shows that the fit is appropriate. The inputs for the NN in this calculation are created at random (not based on experimental data) because we are not interested in the logical performance and the number of operations the NN needs to perform only depends on the number of rounds. In particular, the number of operations in the LSTM and encoding layers grows linearly with R but for the evaluation layers it is constant. Therefore, we can associate the y intercept as the time required for the evaluation layers and the slope as the time spent on the encoding and LSTM layers. Each point is the average of 5×10^4 samples.

data, while excluding components representing $|1\rangle \rightarrow |0\rangle$ decay from the measurement-error edge weights. It is also easily extended to incorporate measurements with leakage to the $|2\rangle$ -state.

a. Two-state Gaussian-mixture model

For discrimination between $|0\rangle$ and $|1\rangle$, we can use projected coordinates \tilde{z} as defined in the main text, where the PDFs $P(\tilde{z} | j)$ for $j \in \{0, 1\}$ have the form

$$P(\tilde{z} | j) = (1 - r_j)f(\tilde{z}; \tilde{\mu}_0, \sigma) + r_j f(\tilde{z}; \tilde{\mu}_1, \sigma). \quad (\text{E1})$$

Here, $f(\tilde{z}; \tilde{\mu}_i, \sigma)$ is a one-dimensional (1D) Gaussian distribution with mean $\tilde{\mu}_i$ and standard deviation σ , and $r_j \in [0, 1]$ is an amplitude parameter that determines which normal distribution is dominant in the mixture. For $\{r_0 = 0, r_1 = 1\}$, the model represents a readout response with a single dominant component (i.e., no state-preparation errors), while $\{r_0 > 0, r_1 < 1\}$ represents a measurement response where, due to state-preparation errors, there are two distinct components to the measurement response.

The Gaussian-mixture model allows us to discard state-preparation errors from the $P(\tilde{z} | 0)$ by fitting the parameters $\tilde{\mu}_0, \tilde{\mu}_1$ and σ for r_0 from Eq. (E1) and then setting $r_0 = 0$. This assumption holds on the condition that no $|0\rangle \rightarrow |1\rangle$ processes are present over the course of the measurement time—if significant amplitude damping occurs over the course of the measurement, the PDF found using the Gaussian method is inaccurate. When comparing experimental results for logical fidelity with and without setting $r_0 = 0$ for the ground-state distribution $P(\tilde{z} | 0)$, we find no statistically significant difference. We assume that the absence of a fidelity improvement is due to the rarity of $|0\rangle$ state-preparation errors, which are mitigated by heralded initialization. As mentioned in the main text, while we include both Gaussians in the PDF in order to classify the measurement, we use only the main peak in calculating the soft edge weights. This removes the component of qubit error that occurs during measurement from the edge associated with measurement-classification error.

b. Three-state classifier

The 1D projected model is unable to characterize leakage to $|2\rangle$, which has its own characteristic response in the 2D IQ space, shown in Fig. 10. To model this three-state regime and discriminate leakage, we fit a mixture of 2D Gaussians to normalized histograms of the calibration data, giving PDFs $P(z | j)$ for $z \in \mathbb{R}^2, j \in \{0, 1, 2\}$ as follows:

$$P(z | j) = A_j f(z; \vec{\mu}_0, \sigma) + B_j f(z; \vec{\mu}_1, \sigma) + C_j f(z; \vec{\mu}_2, \sigma), \quad (\text{E2})$$

where $f(z; \vec{\mu}_j, \sigma)$ is the PDF of a 2D Gaussian distribution with mean $\vec{\mu}_j$ and covariance matrix $\sigma^2 I$, and parameters A_j, B_j , and C_j are to be fitted for each state $|j\rangle$.

In Fig. 10, we observe that the $|2\rangle$ state has a measurement response that is off the $\vec{\mu}_0 - \vec{\mu}_1$ axis, forming a distinct constellation in the IQ space below the other two measurement responses $|0\rangle$ and $|1\rangle$. The ground-state response is centered around $\vec{\mu}_0$, while the $|1\rangle$ response is distributed between a dominant peak around $\vec{\mu}_1$ and a small number of data points closer to $\vec{\mu}_0$, indicating $|1\rangle \rightarrow |0\rangle$ decay. The response of the $|2\rangle$ state can be seen to decay to both $|0\rangle$ and $|1\rangle$ states, most notably for transmon Z_1 , where the effect can be clearly seen.

Given this simple model, the three-state classifier used to set the leakage flags as input for the NN decoder works as follows. Maximum-likelihood classification means that given z , we should pick $j = 0, 1, 2$, which maximizes $P(j | z)$ in Eq. (D1). The denominator in Eq. (D1) can be dropped, as it solely depends on z . For the numerator, we need to know $P(j)$, which we assume to be independent of j (which is not completely warranted, as $j = 2$ is much less likely) and hence $P(j | z) \propto P(z | j)$ with $P(z | j)$ in Eq. (E2). These arguments identically apply to the two-state classifier discussed in the main text.

2. Combining soft information with the pairwise-correlation method in the final round

As discussed in the main text, we obtain the decoding-graph edge weights from experimental data. These weights will include some averaged probability of a classification error that we wish to remove and replace with a soft-information-based weight on a per-shot basis. In the bulk of the experiment, this is straightforward—the weight of the edge corresponding to a classification error can simply be replaced with that calculated using the soft information, following Eq. (1), as classification and qubit errors have different defect signatures.

However, in the final round, both ancilla- and data-qubit errors result in the same defects as classification errors. Therefore, we expect the total (averaged across shots) edge probability to be given by

$$p = q(1 - c) + c(1 - q), \quad (\text{E3})$$

where c is the averaged probability of any classification error and q is the probability of qubit errors. The validity of this assumption depends on the degree to which we have made correct assumptions about the possible error channels, including that there is no correlated noise. In our case, p is obtained by the pairwise-correlation method but it could be obtained by other means. We note that, in general, errors on multiple qubits may result in the same defects and thus contribute to the same edge probability. In our Surface-13 experiment, this only occurs for certain final-round data-qubit measurements—the pairs D1 and D2, and D8 and D9.

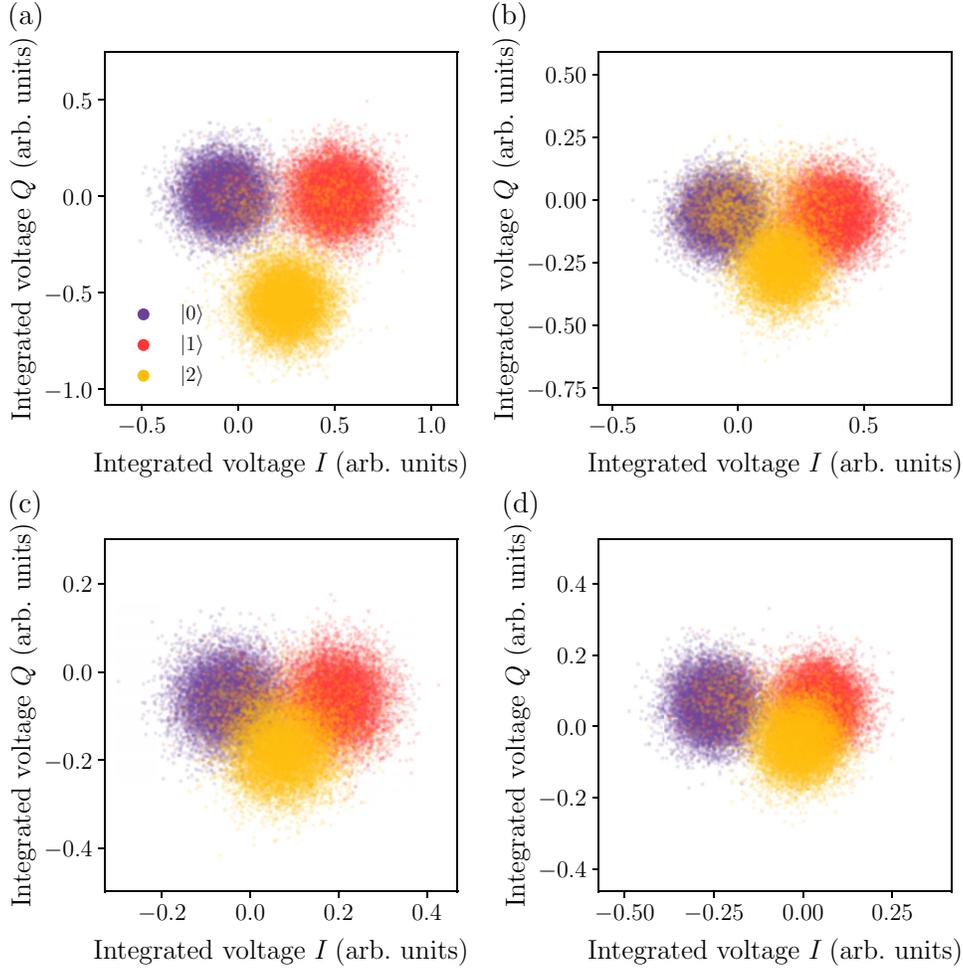


FIG. 10. Calibration shots for experimental IQ voltages, shown for each ancilla prepared in $|0\rangle$, $|1\rangle$, and $|2\rangle$: (a) transmon Z_3 ; (b) transmon Z_1 ; (c) transmon Z_4 ; (d) transmon Z_2 . The cluster for $|2\rangle$ is off axis when compared to the clusters for $|0\rangle$ and $|1\rangle$ and its separation from the two other clusters varies from transmon to transmon. State $|0\rangle$ has the cleanest response, as it does not decay to any other state, while $|1\rangle$ and $|2\rangle$ decay partially to lower-energy states over the course of measurement.

We wish to retain the edge contributions due to the qubit errors, q , and replace only the averaged classification contribution, c , with the per-shot value. We thus have several steps to calculate the soft-information-based weight for final-round edges:

- (i) For each measurement k , calculate the mean classification error, c^k , by averaging the per-shot errors.
- (ii) For each edge, calculate the total edge-classification error, c , from the individual measurement-classification errors, c^k , through

$$c = \frac{1}{2} \left[1 - \prod_k (1 - 2c^k) \right], \quad (E4)$$

where the product is over all measurements the classification errors of which result in the edge defects.

- (iii) For each edge, remove the mean classification error from the edge probability by rearranging Eq. (E3), to find

$$q = \frac{p - c}{1 - 2c}. \quad (E5)$$

- (iv) For each edge, include the per-shot classification error calculated from the softreadout information by combining it appropriately with q .

We now explain the above steps in more detail.

a. Step (i): Calculating the mean classification error for each measurement

For each experiment shot s , we have a soft-measurement outcome for each measurement k , given by z_s^k , and an

associated inferred state after measurement, \hat{z}_s^k . As discussed in the main text, in our case, this is found by taking $\hat{z}_s^k = 1$ if $P'(z_s^k | 1) > P'(z_s^k | 0)$ and 0 otherwise. The PDFs $P'(z^k | j^k)$ are obtained by keeping only the dominant Gaussian in the measurement-classification PDFs. We can calculate the estimated averaged probability of a classification error in measurement k , c^k , via

$$c^k = \frac{1}{N} \sum_{s=1}^N c_s^k \quad (\text{E6})$$

$$= \frac{1}{N} \sum_{s=1}^N \frac{P(1 - \hat{z}_s^k)P'(z_s^k | 1 - \hat{z}_s^k)}{P(\hat{z}_s^k)P'(z_s^k | \hat{z}_s^k) + P(1 - \hat{z}_s^k)P'(z_s^k | 1 - \hat{z}_s^k)}, \quad (\text{E7})$$

where $P(j)$ is the overall probability, across all shots, that the qubit is in state $|j\rangle$; e.g., if $\hat{z}_s^k = 0$, then $P(\hat{z}_s^k) = P(0)$. In this work, we take $P(0) = P(1) = 1/2$. Therefore, we have

$$c^k = \frac{1}{N} \sum_{s=1}^N \frac{P'(z_s^k | 1 - \hat{z}_s^k)}{P'(z_s^k | \hat{z}_s^k) + P'(z_s^k | 1 - \hat{z}_s^k)}. \quad (\text{E8})$$

b. Step (ii): Calculating the mean classification error for each edge

Using the estimated values for c^k obtained with Eq. (E8), we use Eq. (E4) to obtain the mean classification error, c , for each edge. In the case in which an edge is only due to a single measurement-classification error, we simply have $c = c^k$, for the relevant measurement k . The only other case of relevance for Surface-13 is where two classification errors contribute to an edge; in this case, $c = c^{k_1}(1 - c^{k_2}) + (1 - c^{k_1})c^{k_2}$ for the relevant measurements k_1 and k_2 .

c. Step (iii): Removing the mean classification error from each edge probability

We now calculate q for each edge using Eq. (E5).

d. Step (iv): Including the per-shot classification error

We now wish to combine the per-shot soft information with q in order to obtain the full edge weight. In doing so, we mostly follow Ref. [8], with the difference that we merge edges that result in the same defects into a single edge. In everything that follows, we are considering a single experiment shot and thus, to reduce notational clutter, we drop the index s from above.

We begin by considering the full decoding problem that we wish to solve. We have a set of possible errors \mathbb{E} and we consider a single error, e_i , to consist of all events that contribute to the same edge. This includes both

classification errors and other errors. We further have a set of (labeled) soft-measurement outcomes, \mathbb{Z} , which is the union of all sets \mathbb{Z}_i , where \mathbb{Z}_i is the set of measurements the incorrect classification of which leads to the same defect combination as e_i . We wish to find the combination of errors \mathbb{D} that explains the observed defects and maximizes

$$P(\mathbb{D} | \mathbb{Z}) = \frac{P(\mathbb{D} \cap \mathbb{Z})}{P(\mathbb{Z})} \propto P(\mathbb{D} \cap \mathbb{Z}), \quad (\text{E9})$$

where $\mathbb{D} \cap \mathbb{Z}$ is the event that the combination of errors \mathbb{D} occurs and the soft-measurements outcomes \mathbb{Z} are obtained. We can ignore the denominator $P(\mathbb{Z})$, as it is a constant rescaling of all probabilities $P(\mathbb{D} | \mathbb{Z})$ and thus does not need to be considered in order to find the most likely error.

Assuming independence of events, we split $P(\mathbb{D} \cap \mathbb{Z})$ into individual terms for each edge, so that

$$P(\mathbb{D} \cap \mathbb{Z}) = \prod_{e_i \in \mathbb{D}} P(e_i \cap \mathbb{Z}_i) \prod_{e_i \notin \mathbb{D}} P(\bar{e}_i \cap \mathbb{Z}_i). \quad (\text{E10})$$

Rearranging, we find

$$P(\mathbb{D} \cap \mathbb{Z}) = \prod_{e_i \in \mathbb{E}} P(\bar{e}_i \cap \mathbb{Z}_i) \prod_{e_i \in \mathbb{D}} \frac{P(e_i \cap \mathbb{Z}_i)}{P(\bar{e}_i \cap \mathbb{Z}_i)} \quad (\text{E11})$$

$$\propto \prod_{e_i \in \mathbb{D}} \frac{P(e_i \cap \mathbb{Z}_i)}{P(\bar{e}_i \cap \mathbb{Z}_i)}, \quad (\text{E12})$$

where, again, we can drop the term that is common to all error combinations. Maximizing $P(\mathbb{D} \cap \mathbb{Z})$ is equivalent to minimizing

$$-\log[P(\mathbb{D} \cap \mathbb{Z})] = - \sum_{e_i \in \mathbb{D}} \log \left[\frac{P(e_i \cap \mathbb{Z}_i)}{P(\bar{e}_i \cap \mathbb{Z}_i)} \right] \equiv \sum_{e_i \in \mathbb{D}} w_i, \quad (\text{E13})$$

where we have defined

$$w_i = - \log \left[\frac{P(e_i \cap \mathbb{Z}_i)}{P(\bar{e}_i \cap \mathbb{Z}_i)} \right]. \quad (\text{E14})$$

Let us now consider a particular error, e_i , and its n_i associated soft measurements z_i^k for $k = 1, \dots, n_i$. We recall that in the Surface-13 case, n_i , which is the number of classification errors that contribute to edge i , is a maximum of two, and we use this below. In order to calculate w_i , we split e_i into two: e_i^c , which consists of classification errors only, and e_i^g , which consists of all other errors. There are now two ways in which e_i can occur: (i) e_i^g occurs and e_i^c does not occur (i.e., there are an even number of

classification errors) or (ii) e_i^q does not occur and e_i^c does occur (i.e., there are an odd number of classification errors). Therefore,

$$\begin{aligned} P(e_i \cap \mathbb{Z}_i) &= P(e_i^q)P(\bar{e}_i^c \cap \mathbb{Z}_i) + P(\bar{e}_i^q)P(e_i^c \cap \mathbb{Z}_i) = qP(\bar{e}_i^c \cap \mathbb{Z}_i) + (1-q)P(e_i^c \cap \mathbb{Z}_i), \\ P(\bar{e}_i \cap \mathbb{Z}_i) &= P(\bar{e}_i^q)P(e_i^c \cap \mathbb{Z}_i) + P(e_i^q)P(\bar{e}_i^c \cap \mathbb{Z}_i) = qP(e_i^c \cap \mathbb{Z}_i) + (1-q)P(\bar{e}_i^c \cap \mathbb{Z}_i), \end{aligned}$$

where we have defined $P(e_i^q) = q$.

The probability of obtaining the observed soft-measurement outcomes and having an *odd* number of classification errors is

$$P(e_i^c \cap \mathbb{Z}_i) = \begin{cases} P'(z_i^1 | 1 - \hat{z}_i^1), & n_i = 1, \\ P'(z_i^1 | 1 - \hat{z}_i^1)P'(z_i^2 | \hat{z}_i^2) + P'(z_i^1 | \hat{z}_i^1)P'(z_i^2 | 1 - \hat{z}_i^2), & n_i = 2, \end{cases} \quad (\text{E15})$$

and the probability of obtaining the observed soft-measurement outcomes and having an *even* number of classification errors is

$$P(\bar{e}_i^c \cap \mathbb{Z}_i) = \begin{cases} P'(z_i^1 | \hat{z}_i^1), & n_i = 1, \\ P'(z_i^1 | \hat{z}_i^1)P'(z_i^2 | \hat{z}_i^2) + P'(z_i^1 | 1 - \hat{z}_i^1)P'(z_i^2 | 1 - \hat{z}_i^2), & n_i = 2. \end{cases} \quad (\text{E16})$$

These expressions can easily be extended to larger values of n_i but we omit the general expressions here for brevity.

From these, we calculate the edge weight using Eq. (E14) and find

$$w_i = \begin{cases} -\log\left(\frac{L_i^1 + L_i^q}{1 + L_i^1 L_i^q}\right), & n_i = 1, \\ -\log\left(\frac{L_i^1 + L_i^2 + L_i^q + L_i^1 L_i^2 L_i^q}{1 + L_i^1 L_i^2 + L_i^1 L_i^q + L_i^2 L_i^q}\right), & n_i = 2, \end{cases} \quad (\text{E17})$$

where

$$L_i^q = \frac{q}{1-q}, \quad (\text{E18})$$

$$L_i^k = \frac{P'(z_i^k | 1 - \hat{z}_i^k)}{P'(z_i^k | \hat{z}_i^k)}. \quad (\text{E19})$$

APPENDIX F: CALCULATION OF LOGICAL ERROR RATE

To extract the logical error rate ϵ_L from experimental data, we calculate the logical fidelity $F_L(R)$ for each round R of the experiment and fit the data to a decay curve of the form given in Eq. (2). The error in the logical fidelity is given by $\sigma_{F_L}^2 = F_L(1 - F_L)/N$, where N is the number of samples for the given F_L [10], which we propagate through the fitting process to get estimates of uncertainty in ϵ_L and the offset R_0 .

APPENDIX G: ADDITIONAL LOGICAL-ERROR-RATE FIGURES

We show additional plots of the logical fidelity of the soft and hard MWPM decoders for each round of the experiment in Fig. 12. To illustrate the improvement that soft information gives to logical fidelity, in Fig. 11 we show the absolute difference in logical fidelity $F_L(R)$ between the soft and hard MWPM decoders for each round

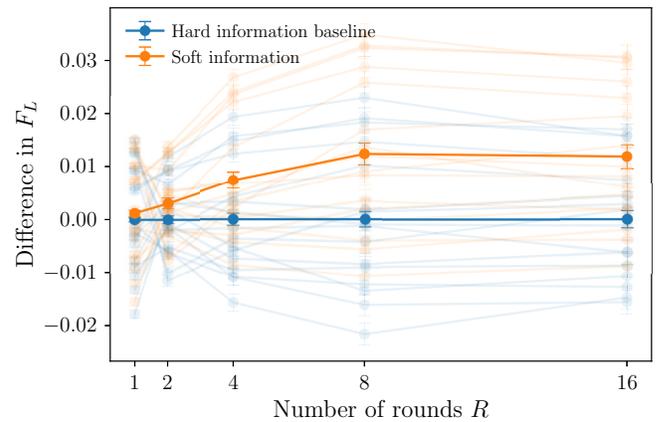


FIG. 11. Absolute difference in logical fidelity F_L as a function of rounds, shown for each individual logical state preparation $|0_L\rangle$ (transparent) and on average across 16 states (opaque) for soft versus hard MWPM.

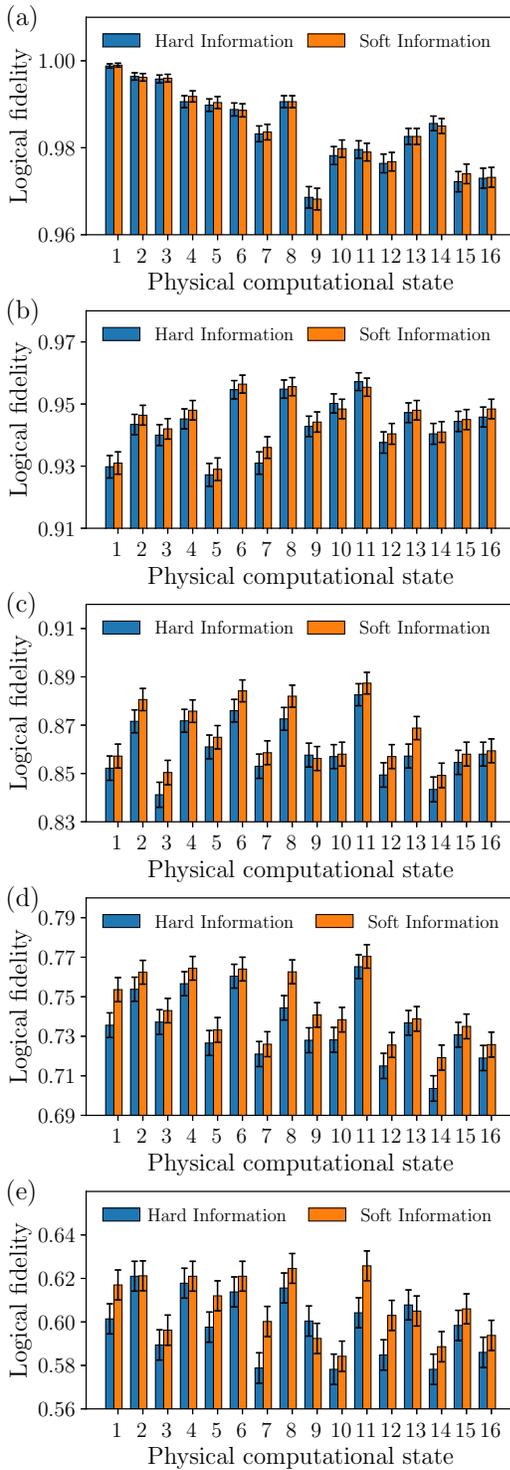


FIG. 12. The logical fidelity using a soft versus a hard MWPM decoder for each round of the experiment: (a) $R = 1$; (b) $R = 2$; (c) $R = 4$; (d) $R = 8$; (e) $R = 16$.

of the experiment. The average performance is shown in solid lines and the fidelity for each individual state preparation $|0_L\rangle$ is shown in the transparent lines.

- [1] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, N. C. Brown, *et al.*, Realization of real-time fault-tolerant quantum error correction, *Phys. Rev. X* **11**, 041058 (2021).
- [2] M. Abobeih, Y. Wang, J. Randall, S. Loenen, C. Bradley, M. Markham, D. Twitchen, B. Terhal, and T. Taminiau, Fault-tolerant operation of a logical qubit in a diamond quantum processor, *Nature* **606**, 884 (2022).
- [3] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, *et al.*, Realizing repeated quantum error correction in a distance-three surface code, *Nature* **605**, 669 (2022).
- [4] Y. Zhao, *et al.*, Realization of an error-correcting surface code with superconducting qubits, *Phys. Rev. Lett.* **129**, 030501 (2022).
- [5] Google Quantum AI, *et al.*, Suppressing quantum errors by scaling a surface code logical qubit, *Nature* **614**, 676 (2023).
- [6] 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).
- [7] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, *et al.*, Logical quantum processor based on reconfigurable atom arrays, *Nature* **626**, 58 (2024).
- [8] C. A. Pattison, M. E. Beverland, M. P. da Silva, and N. Delfosse, Improved quantum error correction using soft information, [ArXiv:2107.13589](https://arxiv.org/abs/2107.13589).
- [9] B. M. Varbanov, M. Serra-Peralta, D. Byfield, and B. M. Terhal, Neural network decoder for near-term surface-code experiments, [ArXiv:2307.03280](https://arxiv.org/abs/2307.03280) (2023).
- [10] 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 to decode the surface code with a recurrent, transformer-based neural network, [ArXiv:2310.05900](https://arxiv.org/abs/2310.05900).
- [11] X. Xue, B. D'Anjou, T. F. Watson, D. R. Ward, D. E. Savage, M. G. Lagally, M. Friesen, S. N. Coppersmith, M. A. Eriksson, W. A. Coish, and L. M. Vandersypen, Repetitive quantum nondemolition measurement and soft decoding of a silicon spin qubit, *Phys. Rev. X* **10**, 021006 (2020).
- [12] 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, *Nat. Commun.* **14**, 2852 (2023).
- [13] T. E. O'Brien, B. Tarasinski, and L. DiCarlo, Density-matrix simulation of small surface codes under current and projected experimental noise, *npj Quantum Inf.* **3**, 39 (2017).
- [14] C. Müller, J. H. Cole, and J. Lisenfeld, Towards understanding two-level-systems in amorphous solids: Insights from quantum circuits, *Rep. Progr. Phys.* **82**, 124501 (2019).
- [15] J. M. Martinis, Qubit metrology for building a fault-tolerant quantum computer, *npj Quantum Inf.* **1**, 15005 (2015).

- [16] J. Kelly, *et al.*, State preservation by repetitive error detection in a superconducting quantum circuit, *Nature* **519**, 66 (2015).
- [17] J. Heinsoo, C. K. Andersen, A. Remm, S. Krinner, T. Walter, Y. Salathé, S. Gasparinetti, J.-C. Besse, A. Potočnik, A. Wallraff, and C. Eichler, Rapid high-fidelity multiplexed readout of superconducting qubits, *Phys. Rev. Appl.* **10**, 034040 (2018).
- [18] C. C. Bultink, B. Tarasinski, N. Haandbaek, S. Poletto, N. Haider, D. J. Michalak, A. Bruno, and L. DiCarlo, General method for extracting the quantum efficiency of dispersive qubit readout in circuit QED, *Appl. Phys. Lett.* **112**, 092601 (2018).
- [19] J. Gambetta, W. A. Braff, A. Wallraff, S. M. Girvin, and R. J. Schoelkopf, Protocols for optimal readout of qubits using a continuous quantum nondemolition measurement, *Phys. Rev. A* **76**, 012325 (2007).
- [20] C. A. Ryan, B. R. Johnson, J. M. Gambetta, J. M. Chow, M. P. da Silva, O. E. Dial, and T. A. Ohki, Tomography via correlation of noisy measurement records, *Phys. Rev. A* **91**, 022118 (2015).
- [21] E. Magesan, J. M. Gambetta, A. D. Córcoles, and J. M. Chow, Machine learning for discriminating quantum measurement trajectories and improving readout, *Phys. Rev. Lett.* **114**, 200501 (2015).
- [22] O. Higgott and C. Gidney, Sparse blossom: Correcting a million errors per core second with minimum-weight matching, *ArXiv:2303.15933*.
- [23] A. G. Fowler, Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $O(1)$ parallel time, *ArXiv:1307.1740*.
- [24] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *J. Math. Phys.* **43**, 4452 (2002).
- [25] O. Higgott, PyMatching: A PYTHON package for decoding quantum codes with minimum-weight perfect matching, *ACM Trans. Quantum Comput.* **3**, 1 (2024).
- [26] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, *Quantum* **5**, 595 (2021).
- [27] B. M. Varbanov, F. Battistel, B. M. Tarasinski, V. P. Ostroukh, T. E. O'Brien, L. DiCarlo, and B. M. Terhal, Leakage detection for a transmon-based surface code, *npj Quantum Inf.* **6**, 102 (2020).
- [28] C. Gidney, Stim: a fast stabilizer circuit simulator, *Quantum* **5**, 497 (2021).
- [29] S. T. Spitz, B. Tarasinski, C. W. Beenakker, and T. E. O'Brien, Adaptive weight estimator for quantum error correction in a time-dependent environment, *Adv. Quantum Technol.* **1**, 1800012 (2018).
- [30] E. H. Chen, T. J. Yoder, Y. Kim, N. Sundaresan, S. Srinivasan, M. Li, A. D. Córcoles, A. W. Cross, and M. Takita, Calibrated decoders for experimental quantum error correction, *Phys. Rev. Lett.* **128**, 110504 (2022).
- [31] Google Quantum AI, Exponential suppression of bit or phase errors with cyclic error correction, *Nature* **595**, 383 (2021).
- [32] Y. Ueno, M. Kondo, M. Tanaka, Y. Suzuki, and Y. Tabuchi, NEO-QEC: Neural network enhanced online superconducting decoder for surface codes, *ArXiv:2208.05758*.
- [33] P. Baireuther, M. D. Caio, B. Criger, C. W. Beenakker, and T. E. O'Brien, Neural network decoder for topological color codes with circuit level noise, *New J. Phys.* **21**, 013003 (2019).
- [34] M. Lange, P. Havström, B. Srivastava, V. Bergentall, K. Hammar, O. Heuts, E. van Nieuwenburg, and M. Granath, Data-driven decoding of quantum error correcting codes using graph neural networks, *arXiv preprint arXiv:2307.01241*.
- [35] U. Naftaly, N. Intrator, and D. Horn, Optimal ensemble averaging of neural networks, *Network: Comput. Neural Syst.* **8**, 283 (1997).
- [36] M. Suchara, A. W. Cross, and J. M. Gambetta, Leakage suppression in the toric code, *Quant. Inf. Comp.* **15**, 997 (2015).
- [37] B. M. Varbanov and M. Serra-Peralta, Github Repository Quantum REcurrent Neural Network Decoder (QRENND), <https://github.com/BorisVarbanov/qrennd> (2023).
- [38] M. Serra-Peralta, Github Repository Surface-13 NN decoder, https://github.com/MarcSerraPeralta/surface-13_nn (2024).
- [39] Delft High Performance Computing Centre (DHPC), Delft-Blue documentation, <https://doc.dhpc.tudelft.nl/delftblue/> (2023), (accessed: 2023-05-15).
- [40] 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, *Phys. Rev. Lett.* **109**, 080505 (2012).
- [41] C. J. Wood and J. M. Gambetta, Quantification and characterization of leakage errors, *Phys. Rev. A* **97**, 032306 (2018).
- [42] T. Walter, P. Kurpiers, S. Gasparinetti, P. Magnard, A. Potočnik, Y. Salathé, M. Pechal, M. Mondal, M. Oppliger, C. Eichler, and A. Wallraff, Rapid high-fidelity single-shot dispersive readout of superconducting qubits, *Phys. Rev. Appl.* **7**, 054020 (2017).
- [43] R. Versluis, S. Poletto, N. Khammassi, B. Tarasinski, N. Haider, D. J. Michalak, A. Bruno, K. Bertels, and L. DiCarlo, Scalable quantum circuit and control for a superconducting surface code, *Phys. Rev. Appl.* **8**, 034021 (2017).
- [44] J. Majer, J. M. Chow, J. M. Gambetta, B. R. Johnson, J. A. Schreier, L. Frunzio, D. I. Schuster, A. A. Houck, A. Wallraff, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Coupling superconducting qubits via a cavity bus, *Nature* **449**, 443 (2007).
- [45] S. Vallés-Sanclemente, S. L. M. van der Meer, M. Finkel, N. Muthusubramanian, M. Beekman, H. Ali, J. F. Marques, C. Zachariadis, H. M. Veen, T. Stavenga, N. Haider, and L. DiCarlo, Post-fabrication frequency trimming of coplanar-waveguide resonators in circuit QED quantum processors, *Appl. Phys. Lett.* **123**, 034004 (2023).
- [46] J. Kelly, *et al.*, Scalable *in situ* qubit calibration during repetitive error detection, *Phys. Rev. A* **94**, 032321 (2016).
- [47] M. A. Rol, L. Ciorciaro, and P. Eendebak, AutoDepGraph, <https://github.com/AdriaanRol/AutoDepGraph> (2017).

- [48] J. Kelly, P. O'Malley, M. Neeley, H. Neven, and J. M. Martinis, Physical qubit calibration on a directed acyclic graph, [ArXiv:1803.03226](https://arxiv.org/abs/1803.03226).
- [49] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm, Simple pulses for elimination of leakage in weakly nonlinear qubits, *Phys. Rev. Lett.* **103**, 110501 (2009).
- [50] J. M. Chow, L. DiCarlo, J. M. Gambetta, F. Motzoi, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, Optimized driving of superconducting artificial atoms for improved single-qubit gates, *Phys. Rev. A* **82**, 040305 (2010).
- [51] E. Magesan, J. M. Gambetta, and J. Emerson, Scalable and robust randomized benchmarking of quantum processes, *Phys. Rev. Lett.* **106**, 180504 (2011).
- [52] V. Negîrmeac, H. Ali, N. Muthusubramanian, F. Battistel, R. Sagastizabal, M. S. Moreira, J. F. Marques, W. J. Vlothuizen, M. Beekman, C. Zachariadis, N. Haider, A. Bruno, and L. DiCarlo, High-fidelity controlled-Z gate with maximal intermediate leakage operating at the speed limit in a superconducting quantum processor, *Phys. Rev. Lett.* **126**, 220502 (2021).
- [53] H. Ali, J. Marques, O. Fedorets, M. Finkel, C. Zachariadis, M. Moreira, W. Vlothuizen, M. Beekman, N. Haider, A. Bruno, and L. DiCarlo, in *APS March Meeting Abstracts*, APS Meeting Abstracts (2022), Vol. 2022, p. Q35.010, <https://ui.adsabs.harvard.edu/abs/2022APS..MARQ35010A>.
- [54] E. H. Chen, T. J. Yoder, Y. Kim, N. Sundaresan, S. Sriniwasan, M. Li, A. D. Córcoles, A. W. Cross, and M. Takita, Calibrated decoders for experimental quantum error correction, *Phys. Rev. Lett.* **128**, 110504 (2022).
- [55] C. C. Bultink, M. A. Rol, T. E. O'Brien, X. Fu, B. C. S. Dikken, C. Dickel, R. F. L. Vermeulen, J. C. de Sterke, A. Bruno, R. N. Schouten, and L. DiCarlo, Active resonator reset in the nonlinear dispersive regime of circuit QED, *Phys. Rev. Appl.* **6**, 034008 (2016).
- [56] J. F. Marques, H. Ali, B. M. Varbanov, M. Finkel, H. M. Veen, S. L. M. van der Meer, S. Valles-Sanclemente, N. Muthusubramanian, M. Beekman, N. Haider, B. M. Terhal, and L. DiCarlo, All-microwave leakage reduction units for quantum error correction with superconducting transmon qubits, *Phys. Rev. Lett.* **130**, 250602 (2023).
- [57] B. M. Varbanov, Ph.D. thesis, Applied Physics, Delft University of University, <https://repository.tudelft.nl/islandora/object/uuid3A140e7b0d-5b24-4e1f-8aa8-fe6edcfd735d?collection=research> (2024).
- [58] P. K. Sarvepalli, A. Klappenecker, and M. Rötteler, Asymmetric quantum codes: Constructions, bounds and performance, *Proc. R. Soc. A* **465**, 1645 (2009).
- [59] P. Baireuther, T. E. O'Brien, B. Tarasinski, and C. W. J. Beenakker, Machine-learning-assisted correction of correlated qubit errors in a topological code, *Quantum* **2**, 48 (2018).
- [60] B. D'Anjou, Generalized figure of merit for qubit readout, *Phys. Rev. A* **103**, 042404 (2021).
- [61] F. Cosco and N. Lo Gullo, Enhancing qubit readout with Bayesian learning, *Phys. Rev. A* **108**, L060402 (2023).