nature

https://doi.org/10.1038/s41586-024-08449-y

Accelerated Article Preview

Quantum error correction below the surface code threshold

Received: 24 August 2024

Accepted: 25 November 2024

Accelerated Article Preview

Cite this article as: Google Quantum AI and Collaborators. Quantum error correction below the surface code threshold. *Nature* https://doi.org/10.1038/s41586-024-08449-y (2024)

Google Quantum AI and Collaborators

Quantum error correction below the surface code threshold

Google Quantum AI and Collaborators
(Dated: November 27, 2024)

Quantum error correction [1–4] provides a path to reach practical quantum computing by combining multiple physical qubits into a logical qubit, where the logical error rate is suppressed exponentially as more qubits are added. However, this exponential suppression only occurs if the physical error rate is below a critical threshold. Here, we present two below-threshold surface code memories on our newest generation of superconducting processors, Willow: a distance-7 code, and a distance-5 code integrated with a real-time decoder. The logical error rate of our larger quantum memory is suppressed by a factor of $\Lambda = 2.14 \pm 0.02$ when increasing the code distance by two, culminating in a 101-qubit distance-7 code with 0.143% $\pm$ 0.003% error per cycle of error correction. This logical memory is also beyond break-even, exceeding its best physical qubit's lifetime by a factor of 2.4 ± 0.3. Our system maintains below-threshold performance when decoding in real time, achieving an average decoder latency of 63 µs at distance-5 up to a million cycles, with a cycle time of 1.1 µs. We also run repetition codes up to distance-29 and find that logical performance is limited by rare correlated error events occurring approximately once every hour, or 3×10^9 cycles. Our results present device performance that, if scaled, could realize the operational requirements of large scale fault-tolerant quantum algorithms.

I. INTRODUCTION

Quantum computing promises computational speedups in quantum chemistry [5], quantum simulation [6], cryptography [7], and optimization [8]. However, quantum information is fragile and quantum operations are error-prone. State-of-the-art many-qubit platforms have only recently demonstrated 99.9% fidelity entangling gates [9, 10], far short of the $< 10^{-10}$ error rates needed for many applications [11, 12]. Quantum error correction is postulated to realize high-fidelity logical qubits by distributing quantum information over many entangled physical qubits to protect against errors. If the physical operations are below a critical noise threshold, the logical error rate should be suppressed exponentially as we increase the number of physical qubits per logical qubit. This behavior is expressed in the approximate relation

$$\varepsilon_d \propto \left(\frac{p}{p_{\text{thr}}}\right)^{(d+1)/2} \tag{1}$$

for error-corrected surface code logical qubits [3, 4, 13]. Here, d is the code distance indicating $2d^2 - 1$ physical qubits used per logical qubit, p and ε_d are the physical and logical error rates respectively, and p_{thr} is the threshold error rate of the code. Thus, when $p \ll p_{\text{thr}}$, the error rate of the logical qubit is suppressed exponentially in the distance of the code, with the error suppression factor $\Lambda = \varepsilon_d/\varepsilon_{d+2} \approx p_{\text{thr}}/p$ representing the reduction in logical error rate when increasing the code distance by two. While many platforms have demonstrated different features of quantum error correction [14–20], no quantum processor has definitively shown below-threshold performance.

Although achieving below-threshold physical error rates is itself a formidable challenge, fault-tolerant quantum computing also imposes requirements beyond raw performance. These include features like stability over the hours-long timescales of quantum algorithms [21] and the active removal of correlated error sources like leakage [22]. Fault-tolerant quantum computing also imposes requirements on classical coprocessors – namely, the syndrome information produced by the quantum device must be decoded as fast as it is generated [23]. The fast operation times of superconducting qubits, ranging from tens to hundreds of nanoseconds, provide an advantage in speed but also a challenge for decoding errors both quickly and accurately.

In this work, we realize surface codes operating below threshold on two Willow processors. Using a 72-qubit processor, we implement a distance-5 surface code operating with an integrated real-time decoder. Subsequently, using a 105-qubit processor with similar performance, we realize a distance-7 surface code. These processors demonstrate $\Lambda > 2$ up to distance-5 and distance-7, respectively. Our distance-5 and distance-7 quantum memories are beyond break-even, with distance-7 preserving quantum information for more than twice as long as its best constituent physical qubit. To identify possible logical error floors, we also implement high-distance repetition codes on the 72-qubit processor, with error rates that are dominated by correlated error events occurring once an hour. These errors, whose origins are not yet understood, set a current error floor of 10^{-10} in the repetition code. Finally, we show that we can maintain below-threshold operation on the 72-qubit processor even when decoding in real time, meeting the strict timing requirements imposed by the processor's fast 1.1 µs cycle duration.

II. A SURFACE CODE MEMORY BELOW THRESHOLD

We begin with results from our 105-qubit Willow processor depicted in Fig. 1a. It features a square grid of

superconducting transmon qubits [25] with improved operational fidelities compared to our previously reported Sycamore processors [17, 26]. The qubits have a mean operating T_1 of 68 µs and $T_{2,\mathrm{CPMG}}$ of 89 µs, which we attribute to improved fabrication techniques, participation ratio engineering, and circuit parameter optimization [24]. Increasing coherence contributes to the fidelity of all of our operations which are displayed in Fig. 1b.

We also make several improvements to decoding, employing two types of offline high-accuracy decoders. One is a neural network decoder [27], and the other is a harmonized ensemble [28] of correlated minimum-weight perfect matching decoders [29] augmented with matching synthesis [30]. These run on different classical hardware, offering two potential paths towards real-time decoding with higher accuracy. To adapt to device noise, we fine-tune the neural network with processor data [27] and apply a reinforcement learning optimization to the matching graph weights [31].

We operate a distance-7 surface code memory comprising 49 data qubits, 48 measure qubits, and 4 additional leakage removal qubits, following the methods in Ref. [17]. Summarizing, we initiate surface code operation by preparing the data qubits in a product state corresponding to a logical eigenstate of either the X_L or Z_L basis of the $ZXXZ$ surface code [32]. We then repeat a variable number of cycles of error correction, during which measure qubits extract parity information from the data qubits to be sent to the decoder. Following each syndrome extraction, we run data qubit leakage removal (DQLR) [33] to ensure that leakage to higher states is short-lived. We measure the state of the logical qubit by measuring the individual data qubits and then check whether the decoder's corrected logical measurement outcome agrees with the initial logical state. It is worth noting that fault-tolerant computation does not require active correction of the code state; the decoder can simply reinterpret the logical measurement outcomes [13].

From surface code data, we can characterize the physical error rate of the processor using the bulk error detection probability [34]. This is the proportion of weight-4 stabilizer measurement comparisons that disagree with their ideal noiseless comparisons, thus detecting an error. The surface code detection probabilities are $p_{\mathrm{det}} = (7.7\%, 8.5\%, 8.7\%)$ for $d = (3, 5, 7)$. We attribute the increase in detection probability with code size to finite size effects [24] and parasitic couplings between qubits. We expect both effects to saturate at larger processor sizes [35].

We characterize our surface code logical performance by fitting the logical error per cycle ε_d up to 250 cycles, averaged over the X_L and Z_L bases. We average the performance of 9 different distance-3 subgrids and 4 different distance-5 subgrids to compare to the distance-7 code. Finally, we compute the error suppression factor Λ using linear regression of $\ln(\varepsilon_d)$ versus d. With our neural network decoder, we observe $\Lambda = 2.14 \pm 0.02$ and $\varepsilon_7 = (1.43 \pm 0.03) \times 10^{-3}$ (see Fig. 1c-d). With ensembled matching synthesis, we observe $\Lambda = 2.04 \pm 0.02$ and $\varepsilon_7 = (1.71 \pm 0.03) \times 10^{-3}$.

Furthermore, we simulate logical qubits of higher distances using a noise model based on the measured component error rates in Fig. 1b, additionally including leakage and stray interactions between qubits [17, 24]. These simulations are shown alongside the experiment in the inset of Fig. 1d, both decoded with ensembled matching synthesis. We observe reasonable agreement with experiment and decisive error suppression, affirming that the surface codes are operating below threshold.

Thus far, we have focused on the error suppression factor Λ, since below threshold performance guarantees that physical qubit lifetimes and operational fidelities can be surpassed with a sufficiently large logical qubit. In fact, our distance-7 logical qubit already has more than double the lifetime of its constituent physical qubits. While comparing physical and logical qubits is subtle owing to their different noise processes, we plot a direct comparison between logical error rate and physical qubit error rate averaged over X and Z basis initializations in Fig. 1c. To quantify qubit lifetime itself, we average uniformly over pure states using the metric proposed in Refs. [16, 24]. The distance-7 logical qubit lifetime is 291 ± 6 µs, exceeding the lifetimes of all the constituent physical qubits (median 85 ± 7 µs, best 119 ± 13 µs) by a factor of 2.4 ± 0.3. Our logical memory beyond break-even extends previous results using bosonic codes [16, 36, 37] to multi-qubit codes, and it is a critical step toward logical operation break-even.

III. LOGICAL ERROR SENSITIVITY

Equipped with below-threshold logical qubits, we can now probe the sensitivity of logical error to various error mechanisms in this new regime. We start by testing how logical error scales with physical error and code distance. As shown in Fig. 2a, we inject coherent errors with variable strength on both data and measure qubits, and extract two quantities from each injection experiment. First, we use detection probability as a proxy for the total physical error rate. Second, we infer the logical error per cycle by measuring logical error probability at 10 cycles, decoding with correlated matching [29].

In Fig. 2b, we plot logical error per cycle versus detection probability for the distance-3, -5, and -7 codes. We find that the three curves cross near a detection probability of 20%, roughly consistent with the crossover regime explored in Ref. [17]. The inset further shows that detection probability acts as a good proxy for $1/\Lambda$ [24, 34]. When fitting power laws below the crossing, we observe approximately 80% of the ideal value $(d{+}1)/2$ predicted by Eq. 1. We hypothesize that this deviation is caused by excess correlations in the device. Nevertheless, higher distance codes show faster reduction of logical error, realizing the characteristic threshold behavior *in situ* on a quantum processor.

To quantify the impact of correlated errors along with more typical gate errors, we form an error budget. Following the method outlined in Refs. [17, 38], we estimate the relative contribution of different component errors to $1/\Lambda$. We run simulations based on a detailed model of our 72-qubit processor. The model includes local noise sources due to gates and measurements, as well as two sources of correlated error: leakage, and stray interactions between neighboring qubits during our CZ gates which can induce correlated ZZ and swap-like errors [24]. Fig. 2c shows our estimated error budget for $1/\Lambda$ in the 72-qubit processor when decoding with correlated matching. Applying the same decoder to experimental data yields a $\Lambda = 1.97 \pm 0.02$. The error budget overpredicts Λ by 14% ("excess" in Fig. 2c), indicating that most but not all error effects in our processor have been captured. Leakage and stray interactions make up an estimated 17% of the budget, and while not a dominant contributor, we expect their importance to increase as error rates decrease. In addition, out-of-model long-range interactions or high-energy leakage might contribute to the error budget discrepancy. Overall, both local and correlated errors from CZ gates are the largest contributors to the error budget. Consequently, continuing to improve both coherence and calibration will be crucial to further reducing logical error.

One potential source of excess correlations that we actively mitigate is leakage to higher excited states of our transmon qubits. During logical qubit operation, we remove leakage from measure qubits using multi-level reset. For data qubits, DQLR swaps leakage excitations to measure qubits (or additional leakage removal qubits) [33]. To examine sensitivity to leakage, we measure logical error probability of distance-3 and distance-5 codes in our 72-qubit processor with and without DQLR, with the results shown in Fig. 2d. While activating DQLR does not strongly affect distance-3 performance, it substantially boosts distance-5 performance, resulting in a 35% increase in Λ. Comparatively, the detection probability decreases by only 12% [24], indicating that detection probability is only a good proxy for logical error suppression if errors are uncorrelated. Overall, we find that addressing leakage is crucial to operating surface codes with transmon qubits [15, 33, 39].

Finally, we test sensitivity to drift. Using our 72-qubit processor, we measure logical performance of one distance-5 and four distance-3 codes 16 times over 15 hours, with the results shown in Fig. 2e. Prior to the repeated runs, we employ a frequency optimization strategy which forecasts defect frequencies of two-level systems (TLSs). This helps to avoid qubits coupling to TLSs during the initial calibration as well as over the duration of the experiments. Additionally, between every four experimental runs, we recalibrate the processor to account for potential qubit frequency and readout signal drift. We observe an average $\Lambda = 2.18 \pm 0.07$ (standard deviation) and best $\Lambda = 2.31 \pm 0.02$ [24] when decoding with the neural network. While the performance of the worst distance-3 quadrant appears to fluctuate due to a transient TLS moving faster than our forecasts, this fluctuation is suppressed in the distance-5 code, suggesting that larger codes are less sensitive to component-level fluctuations. Additionally, the logical error rates of experiments right after drift recalibration are not appreciably lower than those just prior, indicating that our logical qubit is robust to the levels of qubit frequency and readout drift present. These results show that superconducting processors can remain stable over the hours-long timescales required for large scale fault-tolerant algorithms [21].

IV. A REPETITION CODE MEMORY IN THE ULTRA-LOW ERROR REGIME

Despite realizing below-threshold surface codes, orders of magnitude remain between present logical error rates and the requirements for practical quantum computation. In previous work running repetition codes, we found that high-energy impact events occurred approximately once every 10 seconds, causing large correlated error bursts which manifested a logical error floor around 10^{-6} [17]. Such errors would block our ability to run error-corrected algorithms in the future, motivating us to reassess repetition codes on our newer devices.

Using our 72-qubit processor, we run a distance-29 repetition code for 1000 cycles of error correction over 2×10^7 shots split evenly between bit- and phase-flip codes. In total, we execute 2×10^{10} cycles of error correction comprising 5.5 hours of processor execution time. Given the logical error probability p_L at 1000 cycles, we infer the logical error per cycle as $\varepsilon_d = \frac{1}{2}\left(1-(1-2p_L)^{1/1000}\right)$. To assess how the logical error per cycle scales with distance d, we follow Ref. [38] and subsample lower distance repetition codes from the distance-29 data.

Averaging over bit- and phase-flip repetition codes, we obtain an error suppression factor $\Lambda = 8.4 \pm 0.1$ when fitting logical error per cycle versus code distance between $d = 5$ and 11, as shown in Fig. 3a. Notably, the error per cycle on the 72-qubit processor is suppressed far below 10^{-6}, breaking past the error floor observed previously. We attribute the mitigation of high-energy impact failures to gap-engineered Josephson junctions [40]. However, at code distances $d \geq 15$, we observe a deviation from exponential error suppression at high distances culminating in an apparent logical error floor of 10^{-10}. Although we do not observe any errors at distance-29, this is likely due to randomly decoding correctly on the few most damaging error bursts. While this logical error per cycle might permit certain fault-tolerant applications [11], it is still many orders of magnitude higher than expected and precludes larger fault-tolerant circuits [12, 21].

When we examine the detection patterns for these high-distance logical failures, we observe two different failure modes [24]. The first failure mode manifests as one or two detectors suddenly increasing in detection probability by over a factor of 3, settling to their initial de-

tection probability tens or hundreds of cycles later [24]. These less damaging failures could be caused by transient TLSs appearing near the operation frequencies of a qubit, or by coupler excitations, but might be mitigated using methods similar to Refs. [39, 41]. The second and more catastrophic failure mode manifests as many detectors experiencing a larger spike in detection probability simultaneously; an example is shown in Fig. 3b. Notably, these anisotropic error bursts are spatially localized to neighborhoods of roughly 30 qubits (see inset). Over the course of our 2×10^{10} cycles of error correction, our processor experienced six of these large error bursts, which are responsible for the highest-distance failures. These bursts, such as the event shown in Fig. 3b, are different from previously observed high-energy impact events [17]. They occur approximately once an hour, rather than once every few seconds, and they decay with an exponential time constant around 400 µs, rather than tens of milliseconds. We do not yet understand the cause of these events, but mitigating them remains vital to building a fault-tolerant quantum computer. These results reaffirm that long repetition codes are a crucial tool for discovering new error mechanisms in quantum processors at the logical noise floor. However, surface codes are larger and sensitive to more errors than repetition codes, so these events may affect surface code performance differently.

Furthermore, while we have tested the scaling law in Eq. 1 at low distances, repetition codes allow us to scan to higher distances and lower logical errors. Following a similar coherent error injection method as in the surface code, we show the scaling of logical error versus physical error and code distance in Fig. 3c-d, observing good agreement with $O(p^{(d+1)/2})$ error suppression. For example, reducing detection probability by a factor of 2 manifests in a factor of 250 reduction in logical error at distance-15, consistent with the expected $O(p^8)$ scaling. This shows the dramatic error suppression that should eventually enable large scale fault-tolerant quantum computers, provided we can reach similar error suppression factors in surface codes.

V. REAL-TIME DECODING

In addition to a high-fidelity processor, fault-tolerant quantum computing also requires a classical coprocessor that can decode errors in real time. This is because some logical operations are non-deterministic; they depend on logical measurement outcomes that must be correctly interpreted on the fly. If the decoder cannot process measurements fast enough, an increasing backlog of syndrome information can cause an exponential blow-up in computation time [23]. Real-time decoding is particularly challenging for superconducting processors due to their speed. The throughput of transmitting, processing, and decoding the syndrome information in each cycle must keep pace with the fast error correcting cycle time of 1.1 µs. Using our 72-qubit processor as a platform, we demonstrate below-threshold performance alongside this vital module in the fault-tolerant quantum computing stack.

Our decoding system begins with our classical control electronics, where measurement signals are classified into bits then transmitted to a specialized workstation via low-latency Ethernet. Inside the workstation, measurements are converted into detections and then streamed to the real-time decoding software via a shared memory buffer. We employ the Sparse Blossom algorithm [42], which is optimized to quickly resolve local configurations of errors common in surface code decoding, using a parallelization strategy similar to Ref. [43]. The decoder operates on a constant-sized graph buffer that emulates the section of the error graph being decoded at any instant, but which does not grow with the total number of cycles used in the experiment. Different threads are responsible for different spacetime regions of the graph, processing their requisite syndrome information as it is streamed in [43–46]. These results are fused until a global minimum-weight perfect matching is found. The streaming decoding algorithm is illustrated in Fig. 4a-b. We also use a greedy edge reweighting strategy to increase accuracy by accounting for correlations induced by Y-type errors [29, 47].

In Fig. 4c, we report the decoder latency, which we define as the time between the decoding software receiving the final cycle of syndrome measurements and the time when the decoder returns its correction. For our distance-5 surface code, we test different problem sizes by increasing the number of error correction cycles up to 10^6. We observe that the average decoder latency remains roughly constant at a net average of 63 ± 17 µs independent of the length of the experiment (up to 1.1 seconds), indicating that the decoding problem is being processed in real time. This latency will eventually lower bound the reaction time of the logical processor when enacting non-Clifford gates. Other contributions to the reaction time include the data transmission time, which we estimate is less than 10 µs, and feedback, which we have not yet implemented. Additionally, our decoder latency scales with code size, underscoring the need for further optimization.

Importantly, we are able to maintain below-threshold performance even under the strict timing requirements imposed by real-time decoding. We run a dedicated experiment on our 72-qubit processor to compare real-time decoding to high-accuracy offline neural network decoding of the same data, with the results shown in Fig. 4d. Our real-time decoder achieves $\varepsilon_5 = 0.35\% \pm 0.01\%$ and $\Lambda = 2.0 \pm 0.1$ using a device-data-independent prior. Meanwhile, the neural network decoder achieves $\varepsilon_5 = 0.269\% \pm 0.008\%$ and $\Lambda = 2.18 \pm 0.09$ when later decoding offline. The modest reduction in accuracy when comparing the real-time decoder to an offline decoder is expected as the real-time decoder must operate significantly faster. It requires a throughput of less than 1.1 µs per cycle compared to the neural network's 24 µs per cycle [27]. However, we do expect that many of our

high-accuracy decoding methods can eventually be made real-time by introducing techniques like layered or windowed decoding [28, 44, 45].

VI. OUTLOOK

In this work, we have demonstrated surface code memory below threshold in our new Willow architecture. Each time the code distance increases by two, the logical error per cycle is reduced by more than half, culminating in a distance-7 logical lifetime more than double its best constituent physical qubit lifetime. This signature of exponential logical error suppression with code distance forms the foundation of running large scale quantum algorithms with error correction.

Our error-corrected processors also demonstrate other key advances towards fault-tolerant quantum computing. We achieve repeatable performance over several hours and run experiments up to 10^6 cycles without deteriorating performance, both of which are necessary for future large scale fault-tolerant algorithms. Furthermore, we have engineered a real-time decoding system with only a modest reduction in accuracy compared to our offline decoders.

Even so, many challenges remain ahead of us. Although we might in principle achieve low logical error rates by scaling up our current processors, it would be resource intensive in practice. Extrapolating the projections in Fig. 1d, achieving a 10^{-6} error rate would require a distance-27 logical qubit using 1457 physical qubits. Scaling up will also bring additional challenges in real-time decoding as the syndrome measurements per cycle increase quadratically with code distance. Our repetition code experiments also identify a noise floor at an error rate of 10^{-10} caused by correlated bursts of errors. Identifying and mitigating this error mechanism will be integral to running larger quantum algorithms.

However, quantum error correction also provides us exponential leverage in reducing logical errors with processor improvements. For example, reducing physical error rates by a factor of two would improve the distance-27 logical performance by four orders of magnitude, well into algorithmically-relevant error rates [11, 12]. We further expect these overheads will reduce with advances in error correction protocols [48–54] and decoding [55–57].

The purpose of quantum error correction is to enable large scale quantum algorithms. While this work focuses on building a robust memory, additional challenges will arise in logical computation [58, 59]. On the classical side, we must ensure that software elements including our calibration protocols, real-time decoders, and logical compilers can scale to the sizes and complexities needed to run multi-surface-code operations [60]. With below-threshold surface codes, we have demonstrated processor performance that can scale in principle, but which we must now scale in practice.

VII. ACKNOWLEDGEMENTS

We are grateful to A. Ashkenazi, S. Brin, S. Pichai, and R. Porat for their executive sponsorship of the Google Quantum AI team, and for their continued engagement and support.

VIII. AUTHOR CONTRIBUTIONS

The Google Quantum AI team conceived and designed the experiment. The theory and experimental teams at Google Quantum AI developed the data analysis, modeling and metrological tools that enabled the experiment, built the system, performed the calibrations, and collected the data. The Google DeepMind and Google Quantum AI teams jointly developed the machine learning decoder used. All authors wrote and revised the manuscript and the Supplementary Information.

IX. ETHICS DECLARATIONS

The authors declare no competing interests.

X. ADDITIONAL INFORMATION

Supplementary Information is available for this paper. Correspondence and requests for materials should be addressed to H. Neven (neven@google.com). Reprints and permissions information is available at `www.nature.com/reprints`.

XI. DATA AVAILABILITY

The data that support the findings of this study are available at `https://doi.org/10.5281/zenodo.13273331`.

Google Quantum AI and Collaborators

Rajeev Acharya[1], Dmitry A. Abanin[1,2], Laleh Aghababaie-Beni[1], Igor Aleiner[1], Trond I. Andersen[1], Markus Ansmann[1], Frank Arute[1], Kunal Arya[1], Abraham Asfaw[1], Nikita Astrakhantsev[1], Juan Atalaya[1], Ryan Babbush[1], Dave Bacon[1], Brian Ballard[1], Joseph C. Bardin[1,3], Johannes Bausch[4], Andreas Bengtsson[1], Alexander Bilmes[1], Sam Blackwell[4], Sergio Boixo[1], Gina Bortoli[1], Alexandre Bourassa[1], Jenna Bovaird[1], Leon Brill[1], Michael Broughton[1], David A. Browne[1], Brett Buchea[1], Bob B. Buckley[1], David A. Buell[1], Tim Burger[1], Brian Burkett[1], Nicholas Bushnell[1], Anthony Cabrera[1], Juan Campero[1], Hung-Shen Chang[1], Yu Chen[1], Zijun Chen[1], Ben Chiaro[1], Desmond Chik[1], Charina Chou[1], Jahan Claes[1], Agnetta Y. Cleland[1], Josh Cogan[1], Roberto Collins[1], Paul Conner[1], William Courtney[1], Alexander L. Crook[1], Ben Curtin[1], Sayan Das[1], Alex Davies[4], Laura De Lorenzo[1], Dripto M. Debroy[1], Sean Demura[1], Michel Devoret[1,5], Agustin Di Paolo[1], Paul Donohoe[1], Ilya Drozdov[1,6], Andrew Dunsworth[1], Clint Earle[1], Thomas Edlich[4], Alec Eickbusch[1], Aviv Moshe Elbag[1], Mahmoud Elzouka[1], Catherine Erickson[1], Lara Faoro[1], Edward Farhi[1], Vinicius S. Ferreira[1], Leslie Flores Burgos[1], Ebrahim Forati[1], Austin G. Fowler[1], Brooks Foxen[1], Suhas Ganjam[1], Gonzalo Garcia[1], Robert Gasca[1], Élie Genois[1], William Giang[1], Craig Gidney[1], Dar Gilboa[1], Raja Gosula[1], Alejandro Grajales Dau[1], Dietrich Graumann[1], Alex Greene[1], Jonathan A. Gross[1], Steve Habegger[1], John Hall[1], Michael C. Hamilton[1,7], Monica Hansen[1], Matthew P. Harrigan[1], Sean D. Harrington[1], Francisco J. H. Heras[4], Stephen Heslin[1], Paula Heu[1], Oscar Higgott[1], Gordon Hill[1], Jeremy Hilton[1], George Holland[4], Sabrina Hong[1], Hsin-Yuan Huang[1], Ashley Huff[1], William J. Huggins[1], Lev B. Ioffe[1], Sergei V. Isakov[1], Justin Iveland[1], Evan Jeffrey[1], Zhang Jiang[1], Cody Jones[1], Stephen Jordan[1], Chaitali Joshi[1], Pavol Juhas[1], Dvir Kafri[1], Hui Kang[1], Amir H. Karamlou[1], Kostyantyn Kechedzhi[1], Julian Kelly[1], Trupti Khaire[1], Tanuj Khattar[1], Mostafa Khezri[1], Seon Kim[1], Paul V. Klimov[1], Andrey R. Klots[1], Bryce Kobrin[1], Pushmeet Kohli[4], Alexander N. Korotkov[1], Fedor Kostritsa[1], Robin Kothari[1], Borislav Kozlovskii[4], John Mark Kreikebaum[1], Vladislav D. Kurilovich[1], Nathan Lacroix[1,8], David Landhuis[1], Tiano Lange-Dei[1], Brandon W. Langley[1], Pavel Laptev[1], Kim-Ming Lau[1], Loïck Le Guevel[1], Justin Ledford[1], Joonho Lee[1,9], Kenny Lee[1], Yuri D. Lensky[1], Shannon Leon[1], Brian J. Lester[1], Wing Yan Li[1], Yin Li[4], Alexander T. Lill[1], Wayne Liu[1], William P. Livingston[1], Aditya Locharla[1], Erik Lucero[1], Daniel Lundahl[1], Aaron Lunt[1], Sid Madhuk[1], Fionn D. Malone[1], Ashley Maloney[1], Salvatore Mandrà[1], James Manyika[1], Leigh S. Martin[1], Orion Martin[1], Steven Martin[1], Cameron Maxfield[1], Jarrod R. McClean[1], Matt McEwen[1], Seneca Meeks[1], Anthony Megrant[1], Xiao Mi[1], Kevin C. Miao[1], Amanda Mieszala[1], Reza Molavi[1], Sebastian Molina[1], Shirin Montazeri[1], Alexis Morvan[1], Ramis Movassagh[1], Wojciech Mruczkiewicz[1], Ofer Naaman[1], Matthew Neeley[1], Charles Neill[1], Ani Nersisyan[1], Hartmut Neven[1], Michael Newman[1], Jiun How Ng[1], Anthony Nguyen[1], Murray Nguyen[1], Chia-Hung Ni[1], Murphy Yuezhen Niu[1,10], Thomas E. O'Brien[1], William D. Oliver[1,11,12,13], Alex Opremcak[1], Kristoffer Ottosson[1], Andre Petukhov[1], Alex Pizzuto[1], John Platt[1], Rebecca Potter[1], Orion Pritchard[1], Leonid P. Pryadko[1,14], Chris Quintana[1], Ganesh Ramachandran[1], Matthew J. Reagor[1], John Redding[1], David M. Rhodes[1], Gabrielle Roberts[1], Eliott Rosenberg[1], Emma Rosenfeld[1], Pedram Roushan[1], Nicholas C. Rubin[1], Negar Saei[1], Daniel Sank[1], Kannan Sankaragomathi[1], Kevin J. Satzinger[1], Henry F. Schurkus[1], Christopher Schuster[1], Andrew W. Senior[4], Michael J. Shearn[1], Aaron Shorter[1], Noah Shutty[1], Vladimir Shvarts[1], Shraddha Singh[1,15,16], Volodymyr Sivak[1], Jindra Skruzny[1], Spencer Small[1], Vadim Smelyanskiy[1], W. Clarke Smith[1], Rolando D. Somma[1], Sofia Springer[1], George Sterling[1], Doug Strain[1], Jordan Suchard[1], Aaron Szasz[1], Alex Sztein[1], Douglas Thor[1], Alfredo Torres[1], M. Mert Torunbalci[1], Abeer Vaishnav[1], Justin Vargas[1], Sergey Vdovichev[1], Guifre Vidal[1], Benjamin Villalonga[1], Catherine Vollgraff Heidweiller[1], Steven Waltman[1], Shannon X. Wang[1], Brayden Ware[1], Kate Weber[1], Travis Weidel[1], Theodore White[1], Kristi Wong[1], Bryan W. K. Woo[1], Cheng Xing[1], Z. Jamie Yao[1], Ping Yeh[1], Bicheng Ying[1], Juhwan Yoo[1], Noureldin Yosri[1], Grayson Young[1], Adam Zalcman[1], Yaxing Zhang[1], Ningfeng Zhu[1], Nicholas Zobrist[1]

[1] Google Research, Mountain View, CA, USA
[2] Department of Physics, Princeton University, Princeton, NJ, USA
[3] Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA
[4] Google DeepMind, London, England, UK
[5] Department of Physics, University of California, Santa Barbara, CA, USA
[6] Department of Physics, University of Connecticut, Storrs, CT, USA
[7] Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA
[8] Department of Physics, ETH Zurich, Zurich, Switzerland
[9] Department of Chemistry, Harvard University, Cambridge, MA, USA
[10] Department of Computer Science, University of California, Santa Barbara, CA, USA
[11] Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA, USA
[12] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA
[13] Department of Physics, Massachusetts Institute of Technology, Cambridge, MA, USA
[14] Department of Physics and Astronomy, University of California, Riverside, CA, USA
[15] Yale Quantum Institute, Yale University, New Haven, CT, USA
[16] Departments of Applied Physics and Physics, Yale University, New Haven, CT, USA

[1] Shor, P. W. Scheme for reducing decoherence in quantum computer memory. *Physical Review A* **52**, R2493 (1995).

[2] Gottesman, D. *Stabilizer codes and quantum error correction* (California Institute of Technology, 1997).

[3] Dennis, E., Kitaev, A. Y., Landahl, A. & Preskill, J. Topological quantum memory. *Journal of Mathematical Physics* **43**, 4452–4505 (2002).

[4] Kitaev, A. Y. Fault-tolerant quantum computation by anyons. *Annals of Physics* **303**, 2–30 (2003).

[5] Aspuru-Guzik, A., Dutoi, A. D., Love, P. J. & Head-Gordon, M. Simulated quantum computation of molecular energies. *Science* **309**, 1704–1707 (2005).

[6] Lloyd, S. Universal quantum simulators. *Science* **273**, 1073–1078 (1996).

[7] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review* **41**, 303–332 (1999).

[8] Farhi, E. *et al.* A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–475 (2001).

[9] McKay, D. C. *et al.* Benchmarking quantum processor performance at scale. *arXiv preprint arXiv:2311.05933* (2023).

[10] DeCross, M. *et al.* The computational power of random quantum circuits in arbitrary geometries. *arXiv preprint arXiv:2406.02501* (2024).

[11] Campbell, E. T. Early fault-tolerant simulations of the Hubbard model. *Quantum Science and Technology* **7**, 015007 (2021).

[12] Kivlichan, I. D. *et al.* Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via Trotterization. *Quantum* **4**, 296 (2020).

[13] Fowler, A. G., Mariantoni, M., Martinis, J. M. & Cleland, A. N. Surface codes: Towards practical large-scale quantum computation. *Physical Review A—Atomic, Molecular, and Optical Physics* **86**, 032324 (2012).

[14] Ryan-Anderson, C. *et al.* Realization of real-time fault-tolerant quantum error correction. *Physical Review X* **11**, 041058 (2021).

[15] Krinner, S. *et al.* Realizing repeated quantum error correction in a distance-three surface code. *Nature* **605**, 669–674 (2022).

[16] Sivak, V. V. *et al.* Real-time quantum error correction beyond break-even. *Nature* **616**, 50–55 (2023).

[17] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature* **614**, 676–681 (2023).

[18] Bluvstein, D. *et al.* Logical quantum processor based on reconfigurable atom arrays. *Nature* **626**, 58–65 (2024).

[19] Gupta, R. S. *et al.* Encoding a magic state with beyond break-even fidelity. *Nature* **625**, 259–263 (2024).

[20] Da Silva, M. *et al.* Demonstration of logical qubits and repeated error correction with better-than-physical error rates. *arXiv preprint arXiv:2404.02280* (2024).

[21] Gidney, C. & Ekerå, M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum* **5**, 433 (2021).

[22] Terhal, B. M. & Burkard, G. Fault-tolerant quantum computation for local non-Markovian noise. *Physical Review A—Atomic, Molecular, and Optical Physics* **71**, 012336 (2005).

[23] Terhal, B. M. Quantum error correction for quantum memories. *Reviews of Modern Physics* **87**, 307–346 (2015).

[24] Google Quantum AI and Collaborators. Supplementary Information for "Quantum error correction below the surface code threshold".

[25] Koch, J. *et al.* Charge-insensitive qubit design derived from the Cooper pair box. *Physical Review A—Atomic, Molecular, and Optical Physics* **76**, 042319 (2007).

[26] Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019).

[27] Bausch, J. *et al.* Learning high-accuracy error decoding for quantum processors. *Nature* **635**, 834–840 (2024).

[28] Shutty, N., Newman, M. & Villalonga, B. Efficient near-optimal decoding of the surface code through ensembling. *arXiv preprint arXiv:2401.12434* (2024).

[29] Fowler, A. G. Optimal complexity correction of correlated errors in the surface code. *arXiv preprint arXiv:1310.0863* (2013).

[30] Jones, C. Improved accuracy for decoding surface codes with matching synthesis. *arXiv preprint arXiv:2408.12135* (2024).

[31] Sivak, V., Newman, M. & Klimov, P. Optimization of decoder priors for accurate quantum error correction. *Physical Review Letters* **133**, 150603 (2024).

[32] Bonilla Ataides, J. P., Tuckett, D. K., Bartlett, S. D., Flammia, S. T. & Brown, B. J. The *XZZX* surface code. *Nature Communications* **12**, 2172 (2021).

[33] Miao, K. C. *et al.* Overcoming leakage in quantum error correction. *Nature Physics* **19**, 1780–1786 (2023).

[34] Hesner, I., Hetényi, B. & Wootton, J. R. Using detector likelihood for benchmarking quantum error correction. *arXiv preprint arXiv:2408.02082* (2024).

[35] Klimov, P. V. *et al.* Optimizing quantum gates towards the scale of logical qubits. *Nature Communications* **15**, 2442 (2024).

[36] Ofek, N. *et al.* Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature* **536**, 441–445 (2016).

[37] Ni, Z. *et al.* Beating the break-even point with a discrete-variable-encoded logical qubit. *Nature* **616**, 56–60 (2023).

[38] Chen, Z. *et al.* Exponential suppression of bit or phase flip errors with repetitive error correction. *Nature* **595**, 383–387 (2021).

[39] Varbanov, B. M. *et al.* Leakage detection for a transmon-based surface code. *npj Quantum Information* **6**, 102 (2020).

[40] McEwen, M. *et al.* Resisting high-energy impact events through gap engineering in superconducting qubit arrays. *arXiv preprint arXiv:2402.15644* (2024).

[41] Strikis, A., Benjamin, S. C. & Brown, B. J. Quantum computing is scalable on a planar array of qubits with fabrication defects. *Physical Review Applied* **19**, 064081 (2023).

[42] Higgott, O. & Gidney, C. Sparse Blossom: correcting a million errors per core second with minimum-weight matching. *arXiv preprint arXiv:2303.15933* (2023).

[43] Wu, Y. & Zhong, L. Fusion Blossom: Fast MWPM decoders for QEC. In *2023 IEEE International Conference*

on Quantum Computing and Engineering (QCE), vol. 1, 928–938 (IEEE, 2023).

[44] Skoric, L., Browne, D. E., Barnes, K. M., Gillespie, N. I. & Campbell, E. T. Parallel window decoding enables scalable fault tolerant quantum computation. *Nature Communications* **14**, 7040 (2023).

[45] Tan, X., Zhang, F., Chao, R., Shi, Y. & Chen, J. Scalable surface-code decoders with parallelization in time. *PRX Quantum* **4**, 040344 (2023).

[46] Bombín, H. *et al.* Modular decoding: parallelizable real-time decoding for quantum computers. *arXiv preprint arXiv:2303.04846* (2023).

[47] Paler, A. & Fowler, A. G. Pipelined correlated minimum weight perfect matching of the surface code. *Quantum* **7**, 1205 (2023).

[48] Litinski, D. & Nickerson, N. Active volume: An architecture for efficient fault-tolerant quantum computers with limited non-local connections. *arXiv preprint arXiv:2211.15465* (2022).

[49] Chamberland, C. & Campbell, E. T. Universal quantum computing with twist-free and temporally encoded lattice surgery. *PRX Quantum* **3**, 010331 (2022).

[50] Bravyi, S. *et al.* High-threshold and low-overhead fault-tolerant quantum memory. *Nature* **627**, 778–782 (2024).

[51] Xu, Q. *et al.* Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays. *Nature Physics* 1–7 (2024).

[52] Gidney, C., Newman, M., Brooks, P. & Jones, C. Yoked surface codes. *arXiv preprint arXiv:2312.04522* (2023).

[53] Gidney, C. Inplace access to the surface code Y basis. *Quantum* **8**, 1310 (2024).

[54] Cain, M. *et al.* Correlated decoding of logical algorithms with transversal gates. *arXiv preprint arXiv:2403.03272* (2024).

[55] Smith, S. C., Brown, B. J. & Bartlett, S. D. Local predecoder to reduce the bandwidth and latency of quantum error correction. *Physical Review Applied* **19**, 034050 (2023).

[56] Barber, B. *et al.* A real-time, scalable, fast and highly resource efficient decoder for a quantum computer. *arXiv preprint arXiv:2309.05558* (2023).

[57] Liyanage, N., Wu, Y., Tagare, S. & Zhong, L. FPGA-based distributed union-find decoder for surface codes. *arXiv preprint arXiv:2406.08491* (2024).

[58] Gidney, C. Stability experiments: the overlooked dual of memory experiments. *Quantum* **6**, 786 (2022).

[59] Lin, S. F., Peterson, E. C., Sankar, K. & Sivarajah, P. Spatially parallel decoding for multi-qubit lattice surgery. *arXiv preprint arXiv:2403.01353* (2024).

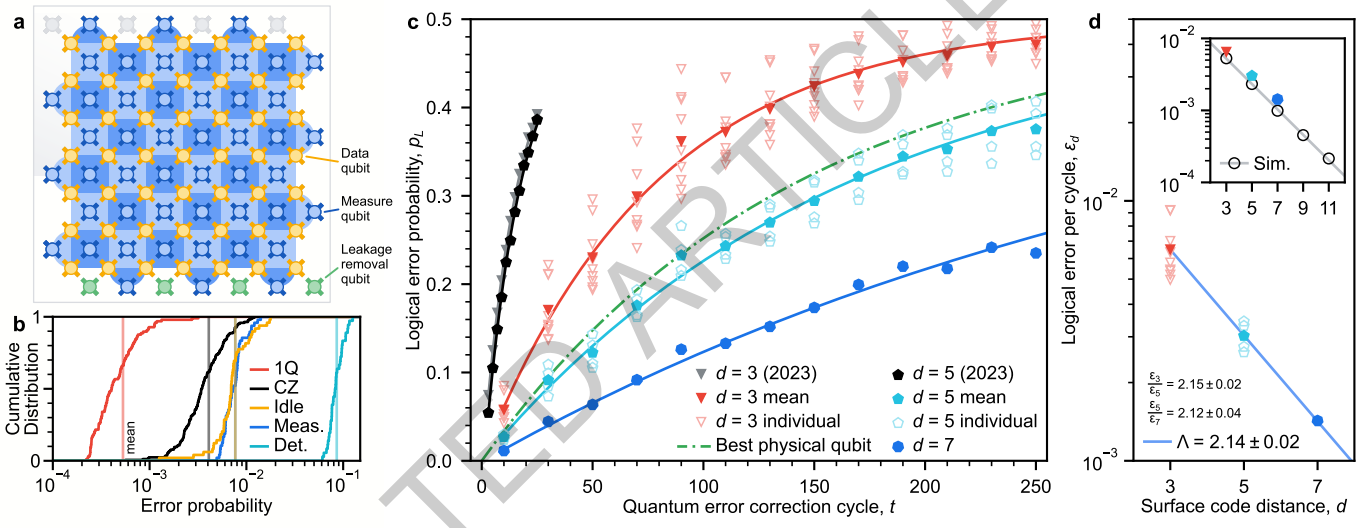
[60] Bombín, H. *et al.* Logical blocks for fault-tolerant topological quantum computation. *PRX Quantum* **4**, 020303 (2023).

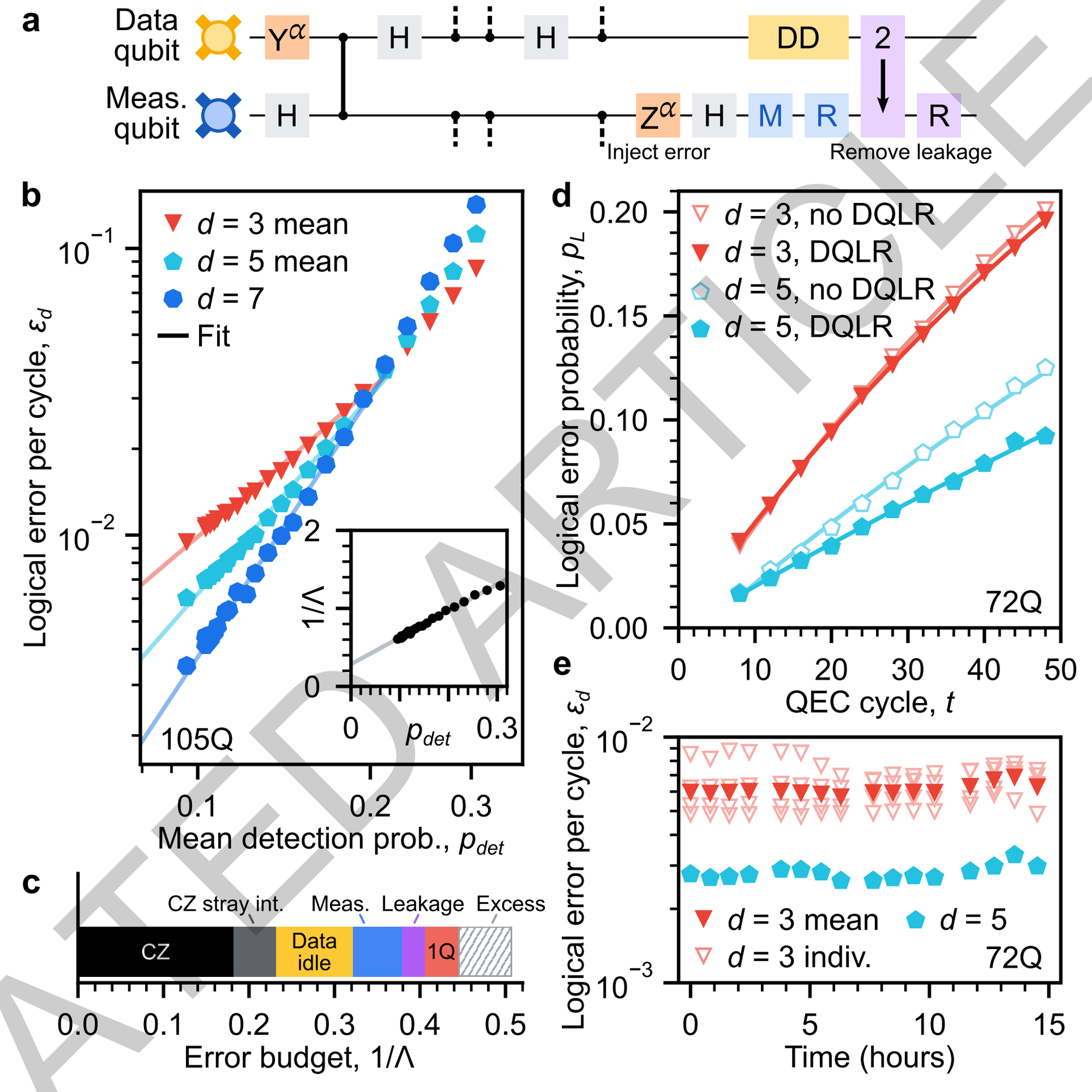
FIG. 1. **Surface code performance. a,** Schematic of a distance-7 ($d = 7$) surface code on a 105-qubit processor. Each measure qubit (blue) is associated with a stabilizer (blue colored tile). Data qubits (gold) form a $d \times d$ array. We remove leakage from each data qubit via a neighboring qubit below it, using additional leakage removal qubits at the boundary (green). **b,** Cumulative distributions of error probabilities measured on the 105-qubit processor. Red: Pauli errors for single-qubit gates. Black: Pauli errors for CZ gates. Blue: identification error for measurement. Gold: Pauli errors for data qubit idle during measurement and reset. Teal: weight-4 detection probabilities (distance-7, averaged over 250 cycles). **c,** Logical error probability, p_L, for a range of memory experiment durations. Each datapoint represents 10^5 repetitions decoded with the neural network and is averaged over logical basis (X_L and Z_L). Black and gray: data from Ref. [17] for comparison. Curves: exponential fits after averaging p_L over code and basis. To compute ε_d values, we fit each individual code and basis separately and report their average [24]. **d,** Logical error per cycle, ε_d, reducing with surface code distance, d. Uncertainty on each point is less than 7×10^{-5}. Symbols match panel c. Means for $d = 3$ and $d = 5$ are computed from the separate ε_d fits for each code and basis. Line: fit to Eq. 1, determining Λ. Inset: simulations up to $d = 11$ alongside experimental points, both decoded with ensembled matching synthesis for comparison. Line: fit to simulation, $\Lambda_{\text{sim}} = 2.25 \pm 0.02$.

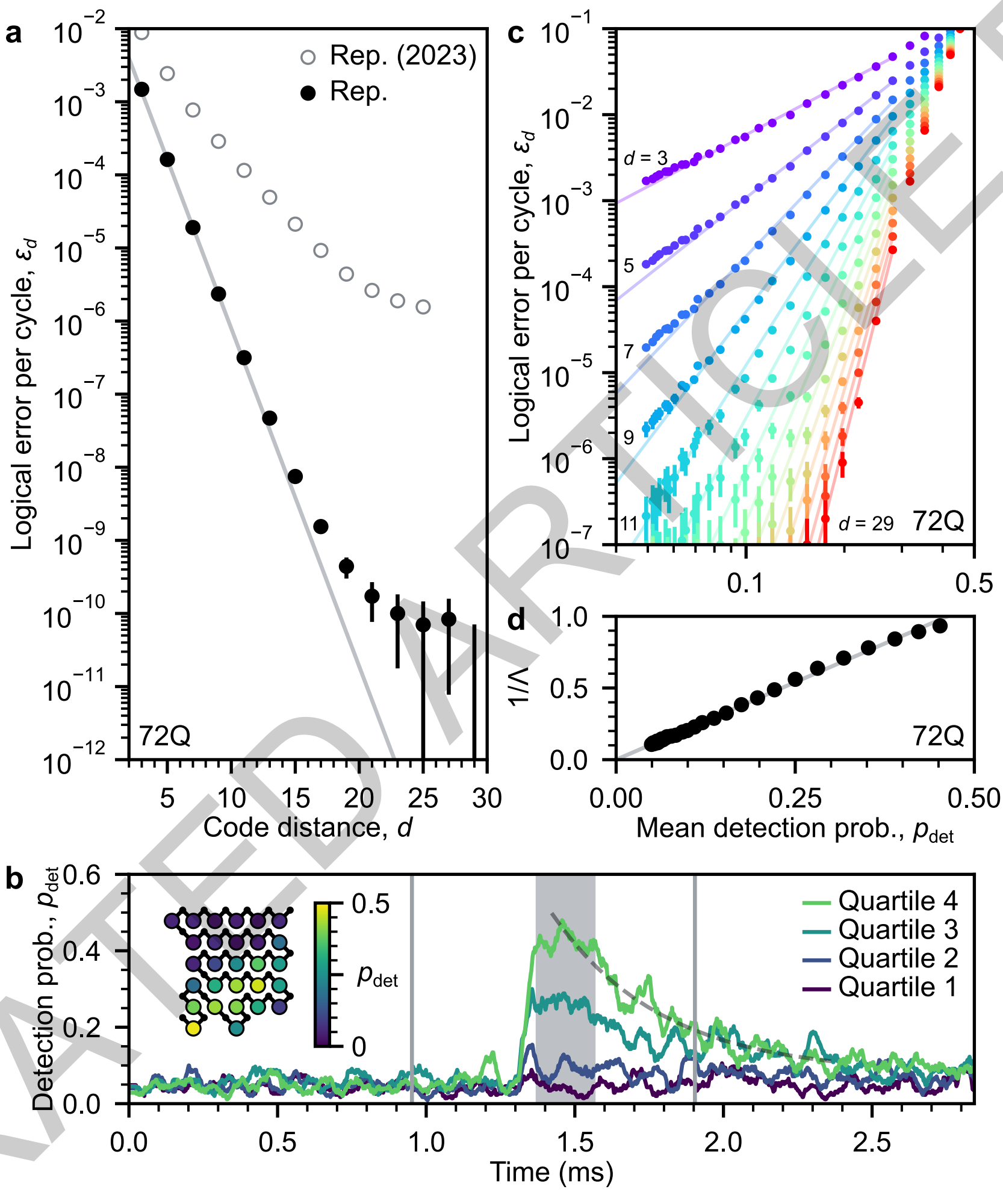
FIG. 2. **Error sensitivity in the surface code. a,** One cycle of the surface code circuit, focusing on one data qubit and one measure qubit. Black bar: CZ, H: Hadamard, M: measure, R: reset, DD: dynamical decoupling. Orange: Injected coherent errors. Purple: Data qubit leakage removal (DQLR) [33]. **b,** Error injection in the surface code. Distance-3 averages over 9 subset codes, and distance-5 averages over 4 subset codes, as in Fig. 1. Logical performance is plotted against the mean weight-4 detection probability averaging over all codes, where increasing the error injection angle α increases detection probability. Each experiment is 10 cycles with 2×10^4 total repetitions. Lines: power law fits for data points at or below where the codes cross. Inset: Inverse error suppression factor, $1/\Lambda$, versus detection probability. Line: fit to points where $1/\Lambda < 1$, $3.4p_{\text{det}} + 0.29$. **c,** Estimated error budget for the surface code based on component errors and simulations. CZ: CZ error, excluding leakage and stray interactions. CZ stray int.: CZ error from unwanted interactions. Data idle: Data qubit idle error during measurement and reset. Meas.: Measurement and reset error. Leakage: Leakage during CZs and due to heating. 1Q: Single-qubit gate error. Excess: unmodeled error, difference between experimental and simulated $1/\Lambda$ (correlated matching). **d,** Comparison of logical performance with and without data qubit leakage removal each cycle. Distance-3 points (red triangles) are averaged over four quadrants. Each experiment is 10^5 repetitions. Curves: exponential fits. **e,** Repeating experiments to assess performance stability, comparing distance-3 and distance-5. Each point represents a sweep of logical performance versus experiment duration, up to 250 cycles.

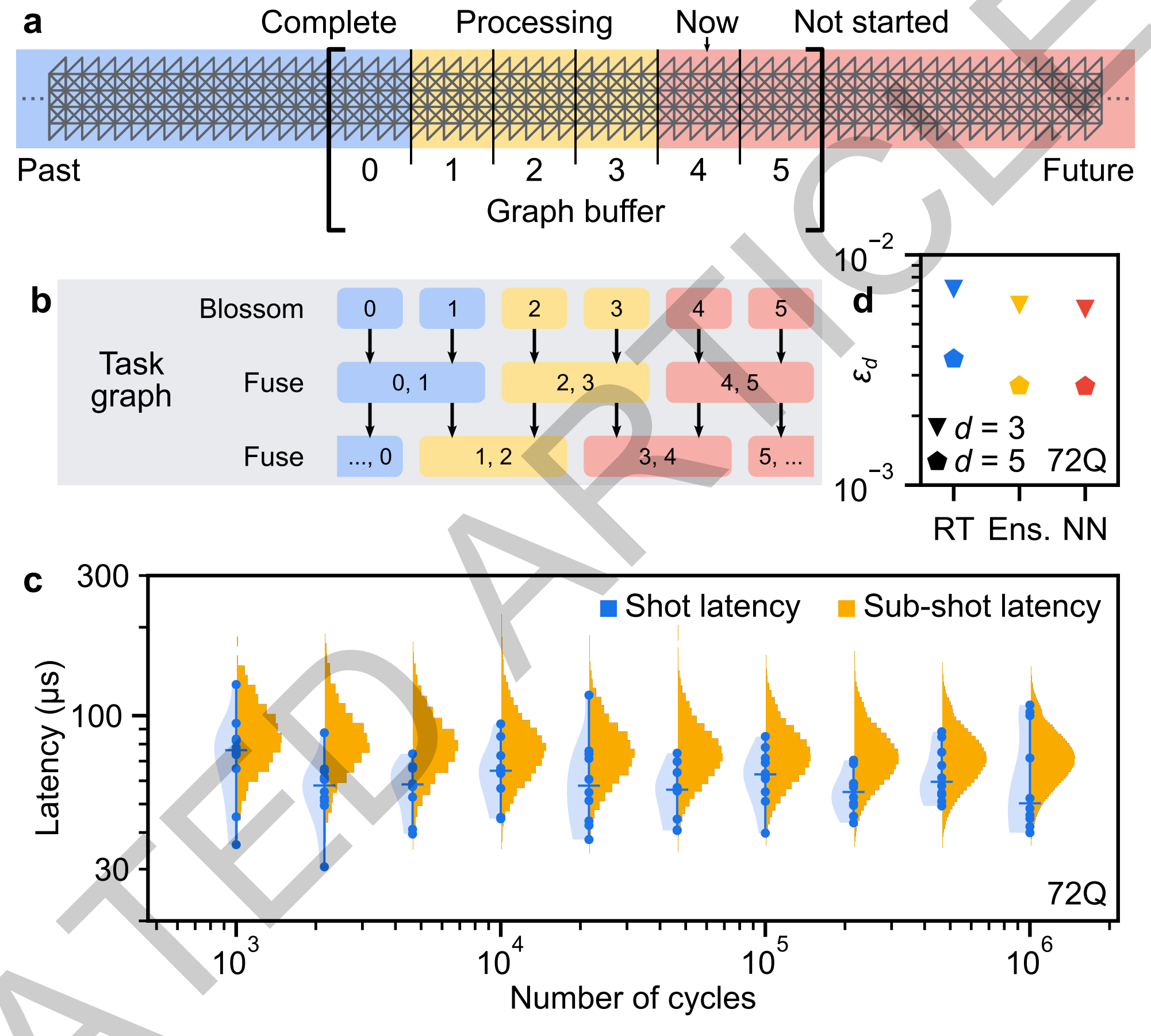
FIG. 3. **High-distance error scaling in repetition codes. a,** Logical error per cycle, ε_d, versus code distance, d, when decoding with minimum-weight perfect matching. Repetition code points are from $d = 29$, 10^3-cycle experiments, 10^7 repetitions for each basis X and Z. We subsample smaller codes from the same $d = 29$ dataset, averaging over subsamples. Line: fit of error suppression factor Λ. We include data from Ref. [17] for comparison. **b,** Example event causing elevated detection probabilities which decay exponentially with time constant $369 \pm 6\,\mu\text{s}$ (gray dashed line). Three consecutive experimental shots are plotted, delimited by vertical gray lines. The 28 measure qubits are divided into four quartiles based on average detection probability in the gray-shaded window. Each trace represents the detection probability averaged over one quartile and a time window of 10 cycles. Roughly half the measure qubits experience an appreciable rise in detection probability. Inset: Average detection probability for each measure qubit (colored circle) within the gray-shaded window. **c,** Logical error scaling with injected error. We inject a range of coherent errors on all qubits and plot against observed mean detection probability p_{det}. Each experiment is 10 cycles, and we average over 10^6 repetitions. Smaller code distances are again subsampled from $d = 29$. Lines: power law fits $\varepsilon_d = A_d p_{\text{det}}^{(d+1)/2}$ (one fit parameter, A_d), restricted to $\varepsilon_d > 10^{-7}$ and $p_{\text{det}} < 0.3$. **d,** $1/\Lambda$ scaling with injected error. Typical relative fit uncertainty is 2%. Line: fit, $2.2p_{\text{det}}$.

FIG. 4. **Real-time decoding. a,** Schematic of the streaming decoding algorithm. Decoding problems are subdivided into blocks, with different threads responsible for different blocks. **b,** Task graph for processing blocks. Detections are allowed to match to block boundaries, which will then be processed downstream during a fuse step. If a configuration of detection events cannot be resolved by a future fuse step, the decoder heralds failure. We use 10-cycle blocks to ensure that the heralded failure rate is negligible compared to the logical failure rate. **c,** Decoder latency versus experiment duration. Each blue point corresponds to a latency measurement for a full shot (10 shots per duration, horizontal bar: median, blue shading: violin plot). Yellow histograms represent fine-grained latency measurements of the time between receiving data and completing decoding for each 10-cycle block within a shot. The values from these fine-grained measurements, which we refer to as sub-shot latencies, tend to be slightly larger than those from full shot latency measurements as the decoder may need to wait to fuse with detection events in future cycles. Infrequently, we see brief sub-shot latency spikes above 1 ms [24]. **d,** Accuracy comparison for the surface code with three decoders. We include the real-time decoder (RT), ensembled matching synthesis (Ens.), and the neural network decoder (NN). Uncertainty on each point is less than 4×10^{-4} [24].


a
Data qubit
Measure qubit
Leakage removal qubit
b
Cumulative Distribution
mean
1Q
CZ
Idle
Meas.
Det.
Error probability
c
Logical error probability, p_L
Quantum error correction cycle, t
d = 3 (2023)
d = 3 mean
d = 3 individual
d = 5 (2023)
d = 5 mean
d = 5 individual
Best physical qubit
d = 7
d
Logical error per cycle, ε_d
Surface code distance, d
Sim.
$\frac{\varepsilon_3}{\varepsilon_5}$ = 2.15 ± 0.02
$\frac{\varepsilon_5}{\varepsilon_7}$ = 2.12 ± 0.04
Λ = 2.14 ± 0.02


a
Data qubit
Meas. qubit
Y^{α}
H
H
H
DD
2
H
Z^{α}
H
M
R
R
Inject error
Remove leakage
b
d = 3 mean
d = 5 mean
d = 7
Fit
Logical error per cycle, ε_d
10^{-1}
10^{-2}
105Q
1/Λ
2
0
0
p_{det}
0.3
0.1
0.2
0.3
Mean detection prob., p_{det}
c
CZ stray int.
Meas.
Leakage
Excess
CZ
Data idle
1Q
0.0
0.1
0.2
0.3
0.4
0.5
Error budget, 1/Λ
d
d = 3, no DQLR
d = 3, DQLR
d = 5, no DQLR
d = 5, DQLR
Logical error probability, p_L
0.20
0.15
0.10
0.05
0.00
72Q
0
10
20
30
40
50
QEC cycle, t
e
Logical error per cycle, ε_d
10^{-2}
10^{-3}
d = 3 mean
d = 5
d = 3 indiv.
72Q
0
5
10
15
Time (hours)




a
Complete
Processing
Now
Not started
Past
0
1
2
3
4
5
Future
Graph buffer
b
Task graph
Blossom
Fuse
Fuse
0
1
2
3
4
5
0, 1
2, 3
4, 5
..., 0
1, 2
3, 4
5, ...
d
10^{-2}
10^{-3}
ε_d
d = 3
d = 5
72Q
RT
Ens.
NN
c
300
100
30
Latency (μs)
Shot latency
Sub-shot latency
72Q
10^3
10^4
10^5
10^6
Number of cycles