nature portfolio

Peer Review File

Quantum error correction below the surface code threshold

Corresponding Author: Dr Hartmut Neven

Version 0:

Reviewer comments:

Referee #1

(Remarks to the Author)
** Assessment of paper's main contribution **

The submitted manuscript "Quantum error correction below the surface code threshold" gives the first clear-cut demonstration "below threshold" repeated quantum error correction. The hallmark feature of quantum error correction is that as you increase the code distance (and therefore the number of physical qubits used) the logical error is exponentially suppressed, provided the qubits are "below threshold" meaning they have low enough physical error probabilities. To be more precise, if epsilon(d) is the logical error rate for a distance d code, Lambda(d) = epsilon(d) / epsilon(d+2) should be a constant larger than 1. The previous Google paper [Nature volume 614, 676–681 (2023)] came close to this landmark result as they measure epsilon(3) / epsilon(5) = 1.04, which indeed was larger than 1. However, this was a borderline result as numerical simulations suggest that Lambda(d) is not a constant. In particular, for d=3 there are significant finite size effects and these effects are especially pronounced near-threshold when Lambda is close to 1. As such the previous 2023 Google QEC paper could not claim to be clearly "below" threshold but only in a cross-over region near the threshold. Nevertheless, the 2023 Google paper was an impressive achievement that other groups has been unable to achieve even using post-selection tricks.

The current submitted manuscript is a more comprehensive demonstration, which goes beyond d=5 to a much larger d=7 and with higher quality qubits. The result is that Lambda(5) =2.12. Numerically, this is more clearly larger than 1 and is more convincing as it eliminates the finite size issues with d=3. Indeed, the supplementary material provides compelling numerical evidence that they understand their dominate noise sources and that Lambda(d) > 2 as d increases (at least until they hit a 10^-10 error floor discussed later).

Another paper to compare against is the recent Harvard-led experiments with neutral atoms tweezer arrays [Nature volume 626, 58–65 (2024)]. One of the many experiments in the Harvard paper included d=3,5,7 surface codes, similar to the submitted manuscript. Furthermore, they entangled two logical qubits and measured them out. They also observe the hallmark feature of Lambda >> 1 (exact value not reported). However, they only performed 1 single round of QEC due to limitations in their readout schemes, and many rounds will be needed for quantum computation. Indeed, only performing 1 round of QEC brings into question whether one has fully prepared and decoded the surface code, and this was discussed extensively in the reports of the Harvard paper. In contrast, the submitted manuscript collected 250 cycles of QEC data for their headline results, and for some secondary analysis goes up to a million cycles of QEC. Lastly, it is unclear whether the Harvard result would have yielded different conclusions if performed for a single logical qubit instead of two.

In the future, we might see experiments with bigger Lambda and at bigger distance. A natural question is what is "a substantial breakthrough" and what is "an incremental improvement". For context, Lambda=10 is a standard target at which error corrected quantum computing becomes practical. And we expect many technological breakthroughs to get to this goal. And so, a substantial breakthrough is one that meaningfully moves the needle from the current best Lambda significantly towards Lambda=10.

Overall, the main result of the submitted manuscript is substantial, and in particular a big enough leap ahead of the closest two recent papers (Google 2023 and Harvard's 2024 paper) to warrant publication in Nature.

I think there are some issues with the paper that need addressing before publication (see critical comment below), but broadly I think this is a fantastic achievement that has excited the community.

** Assessment of paper's secondary contributions **

I see the "below threshold" scaling as the cornerstone result that justifies publication in Nature, but the paper also includes

several complementary results of interest.

Extra result 1) They demonstrate “breakeven” in superconducting qubits, which had been achieved in bosonic codes already. Here breakeven means that a logical qubit has a longer coherence time than the composite physical qubits. Since a logical qubit also come with a longer gate duration, one can make a case that breakeven is a less crucial metric to track. Nevertheless, some readers will be interests that the breakeven landmark has been surpassed so it is worth presenting and the level of attention is appropriate.

Extra result 2) They demonstrate “real-time” decoding in a streaming fashion that has a high enough throughput to avoid the backlog problem with latencies kept constant all the way up to an impression 1 million QEC cycles. They use a software approach as opposed to hardware approaches that are usually associated with real-time decoding on superconducting qubits. However, they do have what look like high latency numbers (64 us for the decoding part, 10 us for the “Input” part, and some unmeasured contributions). This is a nice addition to the paper, though with the latencies on the longer side there is an interesting debate to be had on whether this is the right approach (since anything above about d microseconds is going to have a big impact on the time to execute each non-Clifford operation). The latency will get better as physical qubits are further improved. And on the other hand, the latency will get worse for bigger code distances. The paper is very light on details in this section and does not benchmark how their decoder would perform if operating in a lower error & bigger code regime. Without these details it is difficult to determine how these tradeoffs will play out to yield a practical real-time decoding based on this approach. Nevertheless, it is a good addition to the paper and outstanding engineering challenges are openly discussed in the manuscript.

Extra result 3) In addition to surface code experiments, they run very large repetition code experiments. These experiments can’t suppress both types of error, but with these they can reach much higher Lambda and much higher distances. This enables the authors to probe the hypothesis that we can exponentially suppress errors to any arbitrary level. We have solid mathematical grounds for believing this will happen, but the maths depends on assumptions including independent (uncorrelated errors). In previous (Google 2023) experiments, they saw that high energy events lead to large, long lasting correlated errors causing an error floor at 10^-6 logical error rate in repetition codes. Since the 2023 paper, the Google team showed (arXiv:2402.15644) that strongly gap engineered transmons seems to protect against these correlated sources, at least based on evidence at the physical qubit level. The new chips used in the submitted manuscript have this strong gap engineering and confirm that logical qubits using the repetition code can smash past 10^-6 logical error rate. However, they find a new error floor at 10^-10, due to as yet unknown physics. These are the only experiments really probing the fundamental assumptions in the scalability of quantum error correction, and a fascinating addition to the paper.

** Critical comments and requests for edits **

Comment 1
One peculiarity is that they only use the real-time decoder on a separate chip that only supports d<=5. Is there some reason the RT decoder was not used on the d=7 experiment? Are there any notable differences between these chips except qubit count?

Comment 2

Do both the chips used gap engineered? This is only mentioned in the context of the repetition code and 72-qubit chip.

Comment 3

Eq (1) is presented as “approximately” correct. And while the approximation is pretty good for the unrotated surface code, for the rotated surface code there are significant entropic effects discussed here
https://iopscience.iop.org/article/10.1088/1742-5468/ab25de
This means the scaling should look more like
Epsilon(d) = C (p / p_th)^{A (d+1)}
where “A” is close to 0.5 for the unrotated surface code, and for the rotated surface code I’ve seen (unpublished) numerics where A is fitted to be closer to 0.4 – 0.35 depending on the noise model. While Eq(1) is widely used in the literature, the community needs to stop it propagating without caveats because it is leading to inaccurate performance projections.

For most of the paper, the value of A is unimportant as “p” is a constant and you can just absorb 2A into p, so p/p_th -> (p/p_th)^{A2}.

However, when discussing error sensitivity, the authors inject coherent noise into the system. They fit a power law and find a deviation from the expected scaling in Eq (1), so that the exponent is only 80% of the predicted value.

80% of 0.5 is 0.4, so they seem to be saying their numerics are consistent with A=0.4. Rather than being unexpected this is exactly what I would expect from entropic effects, so I’d request that the authors re-run the numerics here and compare with an A value estimated from the rescaled noise model used in the supplementary materials.

Comment 4

The wording around Fig 2c is unclear in the main text and the Supp Mat. In particular, in the Supp Mat it says you compared

the data generated from Table (S4) and Eq (5) with “simulated” data. But Table (S4) is used to generate simulated data. I assume that in the sentence fragment
“…which directly calculating the ratio of logical error rates from simulated …”
the word “simulated” should be “experimental”. I also have to do some mental gymnastics to follow the main text and the Supp Mat, because the main text discusses 1/Lambda and the Supp Mat concludes with a discussion of Lambda ratios. Consistency in wording & math between the main text and Supp Mat would make it easier to follow.

Comment 5

When real-time decoding is discussed, more details would be appreciated. Let me give some examples.

The authors say “a specialized workstation” is used. Please provide some additional specs to enable fair future comparisons.

Naively, if I look at Fig 4, I see M=5 cycles per block, and 2 blocks active at any time. So if the latency is dominated by the Blossom stage, then I’d expect the decoding time per cycle to be 63 us / 10 = 6.3 us. However, the fact the decoder avoids the backlog problem shows the decoding time per cycle is more like 1.1 us. Reading the paper more deeply, I can see why the naïve counting is wrong but it would really help the reader a lot to have a fuller breakdown of timings.

For instance, it would help to M=10 QEC cycles per block into the main text, which I believe is the correct number and appear in the Sup Mat.

How many blocks are being processed in parallel ? Figure 4b suggestively hints that 2 blocks are processed in parallel (I count the yellow boxes at the Blossom layer). Then the Supp Mat suggests that “up-to” 128 blocks could be in the graph buffer (Fig 4b has 6 blocks in the buffer). How many blocks are actually processed simultaneously is unclear.

How much of the latency is due to fusion stages as opposed to the Blossom stages?

Reading Supp Mat D.1, the language is very different from the main text with new symbols introduced. It would help to connect the headline 63 microsedonc figure from the abstract to the numbers presented in the Supp Mat. I believe that T_software represents the 63 us number, so can you confirm this or not?

You say T_input is about 10 us. This is a sizeable contribution that should be included in the main text and so that you quote 73 microsecond latency in the abstract.

It is also unclear what contributes to T_input and what contributes to T_software. For example, where does the conversion from measurement data to detection events occur?

Comment 6

Fig (S6), the figure legend is using lower case \lambda rather \Lambda

Comment 7

Hypertext links from the citations (in the Supp Mat) would have saved me many minutes of scrolling.

Comment 8

In Supp Mat III.C., when you mention the “stability” experiment, can you more reference the relevant part of the paper, which I believe is figure 2e

Referee #2

(Remarks to the Author)
In this work, the Google team reports on two experimental realizations of a surface code memory, implemented on two distinct superconducting processors. On a 105-qubit chip (Q105), they realize a distance-7 code (along with distance-5 and distance-3 codes on subsets of qubits) while they realize a distance-5 code on the 72-qubit chip (Q72). The main achievements are
- On both chips, a surface code working below threshold, with a clear improvement by a factor $\simeq 2$ in the logical qubit lifetime as the code distance increases by 2. On Q105, is approximately constant when moving from distance-3 to 5 then 7
- The distance-7 code implements a logical memory with coherence clearly beyond break-even
- The distance-5 code on Q72 is decoded in real time, with performances only slightly below those of end-of-line decoders
- To estimate performances in the case where the physical error rates would be retained while scaling up the chip size, the authors implement a distance-29 repetition code on Q72, and demonstrate that one type of logical error (either bit or phase flip) can reach a probability as low as a few 10!"# per correction cycle.

The methods appear to be the same as those presented in a previous publication in Nature (Ref 17) except for improved

calibration and decoding techniques. Besides that, the main changes are (a) a larger (105Q) chip allowing them to increase code distance to 7 (b) improved physical qubit lifetimes roughly yielding a factor 2 improvement in control and readout operations (c) superconducting gap engineering limiting chip-scale qubit failures due to high-energy impacts. However, neither this work nor Ref 17 give details on circuit layout and fabrication techniques, so that accurate comparison is impossible.
This is undoubtfully a “landmark” paper: each achievement listed above is a world-first. Put together, they strongly support the claim that fault-tolerant quantum computing is now within reach (the authors are reasonably cautious about this claim, listing challenges ahead and highlighting that it supposes that performances can be retained and even improved at scale). This is one the most important results of the year (if not of the decade) in experimental quantum information. Therefore, I am very much inclined to recommend publication in Nature.

Nevertheless, the fact that some of the main results were obtained on only one of two different systems is in my view a serious weakness of the paper. In particular, it casts a doubt on the claim that decoding can be achieved in real-time with sufficient throughput and accuracy. Indeed, the spikes in decoder latency visible in Fig S11 (which I think should be mentioned in the main paper as they cannot be inferred from Fig 4) may be a sign that the distance-5 code is saturating the decoder throughput. If the authors are for some reason unable to acquire data on Q105 pertaining to real-time decoding, I recommend that they simulate real-time decoding on a distance-7 code by feeding the decoder offline with error-

syndromes acquired on Q105. In the same vein, implementing a repetition code on the Q105 chip and reporting on the dynamics of correlated bursts of errors would greatly strengthen the paper as it would clearly confirm (or not) that chip-scale qubit failures have been robustly suppressed.
Before recommending publication in Nature, I would like the authors to address this point by either (a) taking more data (or at least simulating real-time decoding as detailed above) or
(b) explaining why this data was not taken / whether it will be done in the near future / whether any preliminary result they may have reveals unforeseen challenges in extending some methods to the Q105 chip.

I list below further comments/questions that may help improve the paper.

1) In Fig 2b, the authors fit ϵ_d versus p_{phys} with power laws. The exponent of this power law is a fit parameter, found to be slightly below ideal value (d+1)/2. In Fig 4b, analyzing similar data from the repetition code, they adopt a different strategy by which the exponent value is fixed at (d+1)/2 and the prefactor is the only fit parameter. As a result, agreement between theoretical predictions and data degrades for short distance codes.
A) Can the authors comment on these two different strategies? Do they understand why the power law seems to reach its ideal value for long distance repetition codes but not for code distances below 11? Do they expect a similar trend for the surface code?
B) The authors should display the fitted exponent for the power law in Fig. 2b. It would also help to mention the ideal (d+1)/2 value in the caption (or represent it on the plot or refer to Eq. 1) as it took me a while to understand what the authors meant by “power law fits” (I initially assumed that the fitted lines had a curvature)
C) About Fig 2b, the authors comment that “When fitting power laws below the crossing, we observe approximately 80% of the ideal value (d+1)/2 predicted by Eq. 1.” and “We find that the three curves cross near a detection probability of 20%, roughly consistent with the crossover regime explored in Ref. [17].” Are these observations related?

2) Error injection is a powerful tool to analyze the device behavior. It looks like the authors could extend the method beyond what is presented in the paper.
A) They inject only coherent errors which are said to be a good proxy for all uncorrelated errors. Can they generalize the method and inject each type of error (as classified in Table S6) separately? Fitting the slope of 1/Λ against the probability of each injected error, it looks like they could estimate the w_i's and compare them to the simulated values reported in Table S6.
B) In the inset of Fig 2b, they should report the fitted slope of 1/Λ versus p_{phys} and compare it to the relevant values w_i reported in table S6. A comment on the fitted line not intersecting 0 would also be welcome (it is understandable from

supp. Mat. Eq. 5 but not from Eq.1 of the main text). A comment on the average
p_{phys} being a good proxy for single qubit error would also help (in the main text, the only justification is that 1/Λ scales linearly p_{phys}, whereas important results on the negligible impact of inhomogeneous error rates are given in the supplemental materials).
C) In Fig 3c (repetition code), the fitted slope should be given. Why does this fitted line intersect 0? Are correlated errors negligible for the repetition code?
D) Could the authors also estimate more precisely the DQLR efficiency by injecting leakage errors? (they give only a very crude estimate for the transition probabilities $P_{1\rightarrow}$) presented in the supplemental materials (bottom of p14)).

3) The authors mention that simulations overpredict Λ by 20%. In Fig 1d, the simulated lambda is 2.25 versus 2.14/2.04 for the measured value. Where does the 20% comes from? In Fig 2c, indicating the simulated and measured 1/Λ values would improve clarity.
The authors attribute this discrepancy to “excess correlations”: supporting this claim and characterizing correlated errors seems very important. In particular, do they remain local or “widespread” as the calibration of microwave drive crosstalks presented in the supplemental materials seem to indicate. Tying up with (1), is this correlated-error model compatible with observed logical error rates in the repetition code? Tying up with (2), can the impact of correlated errors be characterized by injecting these?

4) About drifts and re-calibration, Fig 2e shows that the memory performances are stable over time. From the text, we learn

that qubit frequencies are “optimized” before repeated runs (with details given in supplemental materials), and the processor is recalibrated every 4 runs. Does that mean that target frequencies are set once, and qubit biases adjusted every 4 runs to reach these frequencies?
From supp. Mat. Fig S10, we understand that calibration and adjustment of the decoder priors are crucial to reach optimal performances. The authors give a lot of details on the configuration of decoder priors, which is appreciable. However, it is hard to get an idea of the time it takes to fully calibrate the system (including qubit frequency calibration and decoder configuration) and how often this should be done to maintain performances. Commenting on this “duty-cycle” and how it is expected to scale with chip size would give an idea of the challenges lying ahead when operating large-scale processors.
5) About the real-time decoder, can the authors estimate how computing resources (classical) will scale with code size? Moreover, they mention that they herald failure of the decoder in the computationally most demanding cases, which has no impact on the logical error rate given the low probability of such events. Is this strategy viable for long distance codes or do they expect such failures to dominate before reaching the milestone of 10^-6 error per cycle?

Other minor comments and typos

- Below Eq.1, p and $\epsilon$$ are defined as error rates whereas they are error probabilities per cycle throughout the paper.
- Introduction: referring to Fig 1a when giving the number of physical qubit per logical qubit would help.
- In fig 1a, 102 qubits are represented (including leakage removal qubits) when the chip is said to embed 105 qubits. Can the authors comment on that?
- In the caption of Fig 1b, “Blue: Average identification error” seems at odd with the cumulative errors represented. Similarly, the weight-4 detection probabilities are said to be averaged. Please clarify.
- Caption of Fig 1c: each individual code and logical basis is said to be fitted separately, but only one value is reported per code (not one per basis).
- The authors mention the value of $T+,-./$ in the main text and give a few details on dynamical decoupling techniques in the Supp. Mat. Is this CPMG time measured in the same conditions as the DD sequences presented in S21? (comment on this in the main paper ?).
- In some figures (e.g. 2d, 3c) : fitted values are not given and cannot be inferred from other plots. The authors should give these values in captions or displayed on plots.
- “preparing the data qubits in a product state in either the XL or ZL basis”-> “in a product state corresponding to a logical eigenstate of either the XL or Zl basis”.
- Fig 2a: comment on dashed vertical lines (representing interactions with other qubits
?)
- The sentence “obtained by checking how long after each 10-cycle block’s data is received that the block is completed by the decoder” is hard to parse.
- L32 : “the logical error “ -> “the logical error rate”
- L133 and l160: “X and Z” -> “X_L and Z_L” ?
- Caption Fig 2 : “where the codes cross”

Referee #3

(Remarks to the Author)
The authors report on exciting result of surface code experiments using superconducting transmon qubits below the threshold for error correction.
These are novel results, which the community was actively pushing for, and a clean demonstration of the workings of quantum error correction.
They demonstrate the reduction of error with code distance on surface codes (bit and phase corrected simultaneously, up to d=7 with suppression factor ~2) and at larger distance on repetition codes (either bit or phase).
Various decoding methods are also described (see one point below though).

This could clearly mark a milestone for the field of quantum error correction, and quantum information processing in general.
The results are timely, nicely described, confirm expectations of theoretical scaling, and demonstrate the error correction below threshold. The text is clear, data consistent, well referenced, and conclusions seem reliable.
I do have various points below that I think the authors should address though:

1. *Improved processor*
While I am impressed by the results, very little is said about what enabled the authors can achieve those. The comparison to the experiment of 2023, which was at about the threshold rather than a factor 2 below it as now, is brushed in lines 91 and 92 under 'improved fabrication techniques, participation ratio engineering, and circuit parameter optimization'.
Nothing is said about fabrication techniques of this work, nor of the previous one.
Participation ratio cannot be judged without seeing any device picture, qubit design, or details of the processor. Did the shape of the transmons islands change, the junction designs, the packaging, the film thickness?
I'm not even sure what circuit parameter was optimized in order to improve the coherence of qubits (crosstalk? Purcell protection? quasiparticle mitigation strategies? or is that referring to the frequency optimization in-situ, suppl mat II?).
Clarifying those aspects would go a long way in enabling reproducibility of the results.

2. *Real time decoding*
In line 80, comparing to the cycle time of 1.1us, and later in section V, the authors make claims of real time decoding.
They report factually on a decoder than processes information from the stabilizer measurements with a latency of 63us on average, which is 63/1.1 > 50 cycles.

The decoding latency not increasing with the number of cycles is a formidable achievement, but it makes it 'bounded', or 'finite', possibly 'near realtime', but I'd argue not truely realtime.
In particular, as the authors point out on line 393, the 50+ cycle latency makes the processor react with that delay in implementing non-Clifford gates. In that delay, according to Fig.1, somewhere around 6% (12%) logical error is realized at distance 7 (5).
Certainly this wants to be avoided at algorithmic levels, though I'm fully aware that the problem is far from impossible to solve (once a particular decoder is decided upon, one could implement it on a faster ASIC or FPGA).
I believe the claim should be toned down.

3. *Detection probability*
Line 178, the detection probability is used as a proxy for the total physical error.
I believe this should be a very good proxy, yet nothing is said about it.
How good is it? The authors should be able to know how many error they inject and thus provide some hints on how to quantify this proxy (what is the fraction of injected error that is detected? Though I guess no correlated error was injected, and thus this aspect may remain unclear...).

4. *Suppression factor*
I really expected the fit on the inset of Fig.2b to be a direct proportionality, according to line 43 is should be 1/Lambda ~p/p_thr.
Here the authors have a significant offset. At p_det=0, 1/Lambda is about 0.15.
How does that work? Is it related to the correlated errors? (It seems to imply that even interpolated to zero (detector) error on the processor, some finite logical error will remain, which likely the undetected part, possibly correlated?)

(I also think lines 188-189 about the power law fitting belong to the main part of the figure, and should be therefore moved before the inset discussion.)

5. *Leakage reduction efficiency*
DQLR seems to help strongly distance-5 but not distance-3. I understand this fits with simulation in the supplementary material, but can one give an intuitive explanation of the reasoning in the main text? Is the key element leakage lifetime, or how it spreads across the lattice, ...?

6. *Noise floor at high-distance*
The authors claim to find two different failure modes (line 299). The main text would benefit from a description of the incidence of each: How many shots are we talking about? Which mode dominates? (no reference to the suppl mat discussing it exists...)
Besides, the separation into quartiles in Fig.3d is not really explained: it is neither clear what the quartile splitting is based on (errors during the given erroneous experimental run, averaged over all runs, averaged for a given failure mode?), nor what one learns from it (multiple but not all quartiles involved: is the relevance that many qubits are impacted, or rather that not all are?)

Referee #4

(Remarks to the Author)
The manuscript "Quantum error correction below the surface code threshold" presents a compelling demonstration of below threshold operation of a fault tolerant logical qubit. While there have now been numerous experimental demonstrations of quantum error correction, and several recent demonstrations of "beyond break even" performance, this is the first experimental realization of fault tolerance that, if scaled up, could implement quantum computation at scale. As such, this is a significant milestone in quantum computing.

The authors present strong evidence that their two systems implementing surface codes of two sizes are operating below the code threshold. Their data shows that the logical error rate is exponentially suppressed by increasing the code distance according to the theoretically expected relationship. By artificially introducing additional errors they also show a traditional "threshold crossing" plot, with the limiting case of no additional errors clearly being well into the below threshold regime. Overall this evidence is clearly described and presented and in my view presents sound evidence to support the claim of operating below threshold.

The most notable caveat to the data is that the detection probabilities (which the authors use as a proxy for the physical error rate) increase between code sizes between d=3 to d=7. If the physical error rate continues to increase with system size this could prevent larger scale error correction. The authors state that they expect the effects to saturate. Given the importance of this point to the extrapolation of their result to larger scales a little more discussion around this point may be warranted. (Note that this is not a caveat to the central claim of being below threshold, only about the viability of future scaling)

In addition to this main result the authors provide additional experimental data on the limits of error suppression using a repetition code. They show an improved noise floor in comparison to previous demonstrations, and provide a nice analysis of the behavior of the limiting errors (although they are not yet understood). The noise floor is now low enough that, if it holds for the surface code, some fault tolerant computation in the "useful" regime may be possible even without resolving these mystery error sources.

Finally the authors present data on real-time decoding, which was run in addition to the slightly higher performing offline

decoding algorithms used in their main results. They show that decoder latency does not increase with time of operation, demonstrating that the throughput is sufficient to meet the data output rate. To my knowledge this is the first time such a demonstration has been shown, which in itself is an important result in quantum error correction.

Overall this is a well-written, clearly presented manuscript. The data and its analysis appears sound. The results are of high significance and I recommend it for publication.

Below are some minor suggested edits:

- I would suggest elaborating on the justification of why the apparent increase in physical error rate with code size is expected to saturate, and not prohibit further scaling.

- The term "detection probability" is used to refer to the probability that each syndrome bit is flipped. The name could easily be misunderstood, and is not clearly defined in the main text.

- The following sentence gave the impression that the d=5 code was beyond break-even but the d=7 code was not. My understanding is that both codes are beyond break even.
"Our distance-5 quantum memories are beyond break-even, with distance-7 preserving quantum information for more than twice as long as its best constituent physical qubit."

- The caption of figure 4 refers to End-of-shot and sub-shot latencies. It was not clear to me what was meant by these terms. The main text only mentions and defines "decoding latency".

Version 1:

Reviewer comments:

Referee #1

(Remarks to the Author)
I am responding to the third manuscript draft dated 31st October and after having seen the second round of correspondence with referee 2 and 3.

Much of the requested additional information and clarifications can now be found somewhere in the main text or supplementary material. In particular, the real-time decoder clarification and additional simulations are welcome, and all of my questions around Fig 2b have been clarified by the addition of Table S8 and Table S9 in the supplementary material.

I recommend publication though with one extra minor suggestion

When extracting this information will still be a difficult process for some readers, which can be eased by making direct connections between claims in the main text and the supp material. The authors have already implemented several changes in the decoding appendices that have helped. But the meaning of Fig 2b could be improved further, in particular the claims that “When fitting power laws below the crossing, we observe approximately 80% of the ideal value (d+1)/2 predicted by Eq. 1.” and also the claim that Eq (1) indeed approximately holds true.

My current understanding is that the 80% claim comes from Table S8, and then taking the ratio of k and (d+1)/2, for each d, and then averaging to get approx 80%. Additionally, there is an assumption here that p and p_det are interchangeable in Eq (1), upto a constant. The following discussion assumes this.

Then my suggestion is simply to add the ratio of k and (d+1)/2 into Table T8 as an extra column, and add to the caption add a remark such as “The values in the final column average to approximately 80%, leading the conclusion that the injected noise achieves 80% of the ideal value (d+1)/2 predicted by Eq 1. ”. A similar comparison should be made for Table S9, to justify the claim that simulated data is close to the ideal (d+1)/2 scaling, since my calculations indicate that there is a numerically meaningful (over 4%) deviation from the ideal.

Indeed, the point I was making in my initial report that led to confusion was that one should compare the k values from S8 and S9 directly, rather than comparing S8 with Eq 1. However, this is not needed if (as proposed above) the Supp Mat makes it clear the Table S9 data indeed deviates from Eq (1).

Referee #2

(Remarks to the Author)
The authors have minimally addressed my concerns and questions, and when they did respond, their replies often did not lead to any changes in the manuscript. Therefore, I still do not recommend the publication.

To sum up my concerns, each claim in the abstract taken separately would merit publication in a high-quality journal. However, the collection of claims, as presented, is misleading:

a) Real-time decoding is used on one of two systems only. Why not include the responses to my questions on real-time decoding (and those of referee 1) in the paper or its supplemental materials? Why not test the real-time decoder on actual data from the d=7 code (offline) and not on simulated data?

b) Error-burst suppression is also investigated in a single system only. The authors simply reply that this is “the subject of ongoing investigation”. This is a central claim of the paper and while I personally do not doubt the results or the seriousness of the authors’ work, as a referee I cannot accept what looks like cherry-picking of data over multiple systems.

c) It is said in the abstract that the repetition code “probes the limits of error-correction performances” implying that scaling the surface code to d=29 would result in a similar error-rate. However, the surface code seems to behave differently from the repetition code (unknown error sources revealed by the power laws of Fig 2b and the offset of the 1/Λ vs p_det curve in the inset).
The authors replied to one of my questions that “We do see some evidence of correlated error in the repetition code”, but if they refer to the plateau at the 10e-10 level, this looks like a completely different mechanism from the hypothesized correlated errors on the surface code. As for their point that correlated errors appear in the surface code due to a larger number of “two-qubit interactions”, this mechanism is included in their error-budget and does not explain the discrepancy with the observed value of 1/Λ.

Including the comments on real-time decoding in the paper and putting a few dB of attenuation on the claim on ultra-low error regimes (for instance mentioning that the repetition code probes some particular type of correlated errors appearing as random bursts, and that this is done on a single system) are minimal changes for me to accept the paper.

A few final comments that I would very much like the authors to take into account:
- All fitted values should be reported somewhere (or said to be irrelevant), otherwise fitted lines are merely guides to the eye. The offset in the 1/Λ vs p_det curve in 2b looks very relevant to me. The Pauli simulation included in the response to referee 1 could indeed give an idea of the fitted exponents, but it does not appear anywhere in the paper.
- The error budget in Fig 2c should indicate the actual value of 1/Λ found in the experiment. In the current version of this figure, it looks like the error-budget is fully understood (even though details are given in the text).

Note by the editor:

Once the authors revised their manuscript one further time addressing the points above, Reviewer #2 recommended publication.

Referee #3

(Remarks to the Author)
The authors have addressed many of my comments in their rebuttal letter in a satisfactory way. I wish they had taken the opportunity to feed these insights into the main article, where they would be more visible to the readers (especially when multiple reviewers picked on it).
In particular about the decoding system. I now follow the author's definition of real-time as "not increasing backlog", which is used in some literature but definitely not all. Eg Fig1 of https://iopscience.iop.org/article/10.1088/2399-1984/aceba6 feeds back in the same error correction cycle. Or https://arxiv.org/pdf/2303.04846 has "short reaction time" as one practical requirement of real-time decoding.
In the context of latency limiting logical clock rate (the processor would need to wait for the decoder to catch up to apply a non-clifford gate I assume), the reply to reviewer 2 that the latency scales with distance is a practical problem. This wasn't clear in my first read, and still wouldn't be without reading the rebuttal letter. I would find some comments about latency, its implications and scaling justified in the article.

Note by the editor:

Once the authors revised their manuscript one further time addressing the points above, Reviewer #3 recommended publication.

We thank the referees for both their positive impressions of this work, as well as the many suggestions which have improved the paper presentation. Generally, because we are already at the word limit for a Nature article, several of their suggestions have been addressed in the supplementary information.

Referee #1

Assessment of paper's main contribution

The submitted manuscript "Quantum error correction below the surface code threshold" gives the first clear-cut demonstration "below threshold" repeated quantum error correction. The hallmark feature of quantum error correction is that as you increase the code distance (and therefore the number of physical qubits used) the logical error is exponentially suppressed, provided the qubits are "below threshold" meaning they have low enough physical error probabilities. To be more precise, if epsilon(d) is the logical error rate for a distance d code, Lambda(d) = epsilon(d) / epsilon(d+2) should be a constant larger than 1. The previous Google paper [Nature volume 614, 676–681 (2023)] came close to this landmark result as they measure epsilon(3) / epsilon(5) = 1.04, which indeed was larger than 1. However, this was a borderline result as numerical simulations suggest that Lambda(d) is not a constant. In particular, for d=3 there are significant finite size effects and these effects are especially pronounced near-threshold when Lambda is close to 1. As such the previous 2023 Google QEC paper could not claim to be clearly "below" threshold but only in a cross-over region near the threshold. Nevertheless, the 2023 Google paper was an impressive achievement that other groups has been unable to achieve even using post-selection tricks.

The current submitted manuscript is a more comprehensive demonstration, which goes beyond d=5 to a much larger d=7 and with higher quality qubits. The result is that Lambda(5) =2.12. Numerically, this is more clearly larger than 1 and is more convincing as it eliminates the finite size issues with d=3. Indeed, the supplementary material provides compelling numerical evidence that they understand their dominate noise sources and that Lambda(d) > 2 as d increases (at least until they hit a 10^-10 error floor discussed later).

Another paper to compare against is the recent Harvard-led experiments with neutral atoms tweezer arrays [Nature volume 626, 58–65 (2024)]. One of the many experiments in the Harvard paper included d=3,5,7 surface codes, similar to the submitted manuscript. Furthermore, they entangled two logical qubits and measured them out. They also observe the hallmark feature of Lambda >> 1 (exact value not reported). However, they only performed 1 single round of QEC due to limitations in their readout schemes, and many rounds will be needed for quantum computation. Indeed, only performing 1 round of QEC brings into question whether one has fully prepared and decoded the surface code, and this was discussed extensively in the reports of the Harvard paper. In contrast, the submitted manuscript collected 250 cycles of QEC data for their headline results, and for some secondary analysis goes up to a million cycles of QEC. Lastly, it is unclear whether the Harvard result would have yielded different conclusions if performed for a single logical qubit instead of two.

In the future, we might see experiments with bigger Lambda and at bigger distance. A natural question is what is "a substantial breakthrough" and what is "an incremental improvement". For context, Lambda=10 is a standard target at which error corrected quantum computing becomes practical. And we expect many technological breakthroughs to get to this goal. And so, a substantial breakthrough is one that meaningfully moves the needle from the current best Lambda significantly towards Lambda=10.

Overall, the main result of the submitted manuscript is substantial, and in particular a big enough leap ahead of the closest two recent papers (Google 2023 and Harvard's 2024 paper) to warrant publication in Nature.

I think there are some issues with the paper that need addressing before publication (see critical comment below), but broadly I think this is a fantastic achievement that has excited the community.

We thank the referee for their contextualization of the current state of the art in quantum error correction experiments, and the nice summary of the many different results in this paper.

Naturally, we agree that these results represent a significant leap in experimental quantum error correction.

Assessment of paper's secondary contributions

I see the "below threshold" scaling as the cornerstone result that justifies publication in Nature, but the paper also includes several complementary results of interest.

Extra result 1) They demonstrate "breakeven" in superconducting qubits, which had been achieved in bosonic codes already. Here breakeven means that a logical qubit has a longer coherence time than the composite physical qubits. Since a logical qubit also come with a longer gate duration, one can make a case that breakeven is a less crucial metric to track. Nevertheless, some readers will be interests that the breakeven landmark has been surpassed so it is worth presenting and the level of attention is appropriate.

Extra result 2) They demonstrate "real-time" decoding in a streaming fashion that has a high enough throughput to avoid the backlog problem with latencies kept constant all the way up to an impression 1 million QEC cycles. They use a software approach as opposed to hardware approaches that are usually associated with real-time decoding on superconducting qubits. However, they do have what look like high latency numbers (64 us for the decoding part, 10 us for the "Input" part, and some unmeasured contributions). This is a nice addition to the paper, though with the latencies on the longer side there is an interesting debate to be had on whether this is the right approach (since anything above about d microseconds is going to have a big impact on the time to execute each non-Clifford operation). The latency will get better as physical qubits are further improved. And on the other hand, the latency will get worse for bigger code distances. The paper is very light on details in this section and does not benchmark how their decoder would perform if operating in a lower error & bigger code regime. Without these details it is difficult to determine how these tradeoffs will play out to yield a practical real-time decoding based on this approach. Nevertheless, it is a good addition to the paper and outstanding engineering challenges are openly discussed in the manuscript.

Our real-time decoding system remains a work in progress. In particular, its performance has not yet been fully optimized orcharacterized. We expect upcoming changes to further reduce latency and improve throughput and accuracy. With that said, at these code sizes, the system already meets the throughput requirement and its accuracy is only modestly worse than the best offline decoder we have. Arguably, these requirements are more important than latency as they determine whether fault-tolerant operation is at all possible.

We agree with the referee that the question of whether decoding in software is ultimately the right approach remains open. To the best of our knowledge, no hardware implementation of real-time decoding today implements the correlations that we use to realize $\Lambda > 2$ in real time. We are confident that in future work we can further improve decoding speed without sacrificing accuracy. As software is easier to customize than hardware, we believe that a flexible real-time software decoding system will be an asset as we explore different approaches. Even so, we may consider seeking some form of hardware acceleration, or even partially or completely porting our software components to hardware.

Extra result 3) In addition to surface code experiments, they run very large repetition code experiments. These experiments can't suppress both types of error, but with these they can reach much higher Lambda and much higher distances. This enables the authors to probe the hypothesis that we can exponentially suppress errors to any arbitrary level. We have solid mathematical grounds for believing this will happen, but the maths depends on assumptions including independent (uncorrelated errors). In previous (Google 2023) experiments, they saw that high energy events lead to large, long lasting correlated errors causing an error floor at 10^-6 logical error rate in repetition codes. Since the 2023 paper, the Google team showed (arXiv:2402.15644) that strongly gap engineered transmons seems to protect against these correlated sources, at least based on evidence at the physical qubit level. The new chips used in the submitted manuscript have this strong gap engineering and confirm that logical qubits using the repetition code can smash past 10^-6 logical error rate. However, they find a new error floor at 10^-10, due to as yet unknown physics. These are the only

experiments really probing the fundamental assumptions in the scalability of quantum error correction, and a fascinating addition to the paper.

Critical comments and requests for edits

Comment 1
One peculiarity is that they only use the real-time decoder on a separate chip that only supports d<=5. Is there some reason the RT decoder was not used on the d=7 experiment? Are there any notable differences between these chips except qubit count?

An excellent question. We had already attached the real-time decoding infrastructure to the d=5 chip and collected data when the d=7 chip became available. It would take time and effort to reattach that infrastructure to the d=7 chip, so we didn't. In the future, we intend to run real-time decoding on d=7 (and naturally, much higher distance) codes. There are no meaningful differences between the chips that prevent this from happening; all the components of our real-time decoding stack are compatible with the d=7 chips.

Comment 2

Do both the chips used gap engineered? This is only mentioned in the context of the repetition code and 72-qubit chip.

Yes, the qubits on both chips are gap engineered.

Comment 3

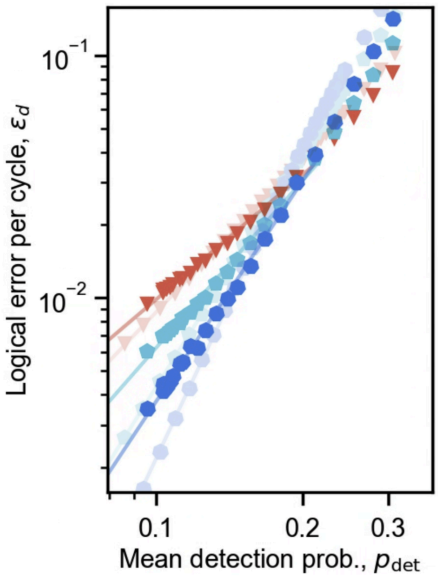
Eq (1) is presented as "approximately" correct. And while the approximation is pretty good for the unrotated surface code, for the rotated surface code there are significant entropic effects discussed here
https://iopscience.iop.org/article/10.1088/1742-5468/ab25de
This means the scaling should look more like
Epsilon(d) = C (p / p_th)^{A (d+1)}
where "A" is close to 0.5 for the unrotated surface code, and for the rotated surface code I've seen (unpublished) numerics where A is fitted to be closer to 0.4 – 0.35 depending on the noise model. While Eq(1) is widely used in the literature, the community needs to stop it propagating without caveats because it is leading to inaccurate performance projections.

For most of the paper, the value of A is unimportant as "p" is a constant and you can just absorb 2A into p, so p/p_th -> (p/p_th)^{A2}.

However, when discussing error sensitivity, the authors inject coherent noise into the system. They fit a power law and find a deviation from the expected scaling in Eq (1), so that the exponent is only 80% of the predicted value.

80% of 0.5 is 0.4, so they seem to be saying their numerics are consistent with A=0.4. Rather than being unexpected this is exactly what I would expect from entropic effects, so I'd request that the authors re-run the numerics here and compare with an A value estimated from the rescaled noise model used in the supplementary materials.

We're a bit confused by this. In the arxiv version of the quoted article, Fig. 6 mentions "We observe slow convergence of α to $1/2^{3/2}(1/2)$ for the square(rotated)-lattice models", where α is the aforementioned exponent. We certainly agree that Eq (1) is oversimplified, but we also believe that the smaller exponents we see are due to correlated error as the device noise begins to dominate the independent coherent errors we inject. For example, we hypothesize that the upward inflection of points away from the fits in the low error regime is evidence of these correlated errors. Below, we plot the paper data (opaque) against a Pauli simulation (semi-transparent). The Pauli simulation is of standard depolarizing circuit noise, and does not include any correlated noise effects. Indeed, we observe that the fitted simulated exponents are approximately (d+1)/2.



Comment 4

The wording around Fig 2c is unclear in the main text and the Supp Mat.

In particular, in the Supp Mat it says you compared the data generated from Table (S4) and Eq (5) with “simulated” data. But Table (S4) is used to generate simulated data.

The referee correctly interprets that the component errors in Table S4 (reproduced in Table S6) are used in two distinct ways:

1. To arithmetically estimate Λ as an additive budget of error sources, using a linear combination of the sensitivities w_i and component errors, as described by Eq (5).
2. For generating simulated data, e.g. to model noise channels in the simulations described in Section IV.A.1 of the supplementary information. In this case, we directly simulate distance-3 and distance-5 surface code circuits to obtain a simulated logical error rate for each code distance. We then take the ratio of these two logical error rates to arrive at a simulated value of Λ.

We compare these two different ways of predicting Λ, which both use the component errors of Table S4.. To avoid confusion, we have added designations for these different values of Λ as “linear” for (1) and “Pauli+” for (2), which we also compare to the experimentally observed (“expt.”) value of Λ. We have added the above explanations to Section IV.A.3 of the supplementary information and further refined the surrounding language to improve clarity.

I assume that in the sentence fragment “…which directly calculating the ratio of logical error rates from simulated …” the word “simulated” should be “experimental”.

We believe the referee is misinterpreting the “direct calculation” referenced in this sentence, and we have clarified the language to reduce confusion (see explanation above).

I also have to do some mental gymnastics to follow the main text and the Supp Mat, because the main text discusses 1/Lambda and the Supp Mat concludes with a discussion of Lambda ratios. Consistency in wording & math between the main text and Supp Mat would make it easier to follow.

We note that both the main text and the supplementary information discuss values of 1/Λ as well as values of Λ. In particular, we find it more intuitive to use 1/Λ when constructing error budgets (e.g. Fig 2c of the main text, and Section.III.A.3 of the supplementary information), and Λ when referencing overall surface code performance (e.g. main text abstract, and periodically throughout the main text, as well as the supplementary information).

> Comment 5
>
> When real-time decoding is discussed, more details would be appreciated. Let me give some examples.
>
> The authors say “a specialized workstation” is used. Please provide some additional specs to enable fair future comparisons.

We use a 64-core AMD Ryzen Threadripper PRO 5995WX with 62 GB of RAM and a real-time Linux operating system. We have added these details to the supplementary information.

> Naively, if I look at Fig 4, I see M=5 cycles per block, and 2 blocks active at any time. So if the latency is dominated by the Blossom stage, then I’d expect the decoding time per cycle to be 63 us / 10 = 6.3 us. However, the fact the decoder avoids the backlog problem shows the decoding time per cycle is more like 1.1 us. Reading the paper more deeply, I can see why the naïve counting is wrong but it would really help the reader a lot to have a fuller breakdown of timings.

To clarify, the figure does not depict the actual hyperparameters (e.g. block size, number of parallel workers, etc.) used by the real-time decoder. It is meant as a visual aide to describe the algorithm. Indeed, because we are decoding each block in parallel over multiple cores, the decoder throughput is faster than 1.1 μs per cycle.

> For instance, it would help to M=10 QEC cycles per block into the main text, which I believe is the correct number and appear in the Sup Mat.

The block size does appear in the figure caption, but also changes depending on the distance of the code. For example, when running higher-distance repetition codes in real-time, we use larger block sizes.

> How many blocks are being processed in parallel ? Figure 4b suggestively hints that 2 blocks are processed in parallel (I count the yellow boxes at the Blossom layer). Then the Supp Mat suggests that “up-to” 128 blocks could be in the graph buffer (Fig 4b has 6 blocks in the buffer). How many blocks are actually processed simultaneously is unclear.

Indeed, the 128 blocks refers to how many blocks we keep in memory. However, we only use 32 of the 64 available cores when decoding. This is because the decoder can already exceed the necessary throughput using a limited number of cores.

> How much of the latency is due to fusion stages as opposed to the Blossom stages?

This is a great question, and budgeting the costs of different contributions to latency is something we intend to follow up on in the future. We haven’t tested this carefully, but we have seen evidence that Blossom is (unsurprisingly) the most costly subroutine.

> Reading Supp Mat D.1, the language is very different from the main text with new symbols introduced. It would help to connect the headline 63 microsedonc figure from the abstract to the numbers presented in the Supp Mat. I believe that T_software represents the 63 us number, so can you confirm this or not?

This is correct, 63 μs corresponds to the T_{software}. We clarified this in D.1.

> You say T_input is about 10 us. This is a sizeable contribution that should be included in the main text and so that you quote 73 microsecond latency in the abstract.

We have estimated this time but have not measured it directly. Consequently, we have renamed the 63 us average latency in the main text to “average decoder latency”, as it does not include the extra several microseconds required for the electronics to deliver the syndrome to our software stack. Of course, the overall response time of a real-time control system will eventually include this time alongside the time required to feed back into the device, which is future work.

> It is also unclear what contributes to T_input and what contributes to T_software. For example, where does the conversion from measurement data to detection events occur?

The conversion from measurement data to detection events is included in $T_{software}$. Indeed, one can think of T_{input} as the time it takes to get the data from the device into the decoding software stack (which includes measurement-to-detection-event conversion).

> Comment 6
>
> Fig (S6), the figure legend is using lower case \lambda rather \Lambda

Fixed.

> Comment 7
>
> Hypertext links from the citations (in the Supp Mat) would have saved me many minutes of scrolling.

Apologies - fixed.

> Comment 8
>
> In Supp Mat III.C., when you mention the “stability” experiment, can you more reference the relevant part of the paper, which I believe is figure 2e

Done.

Referee #2

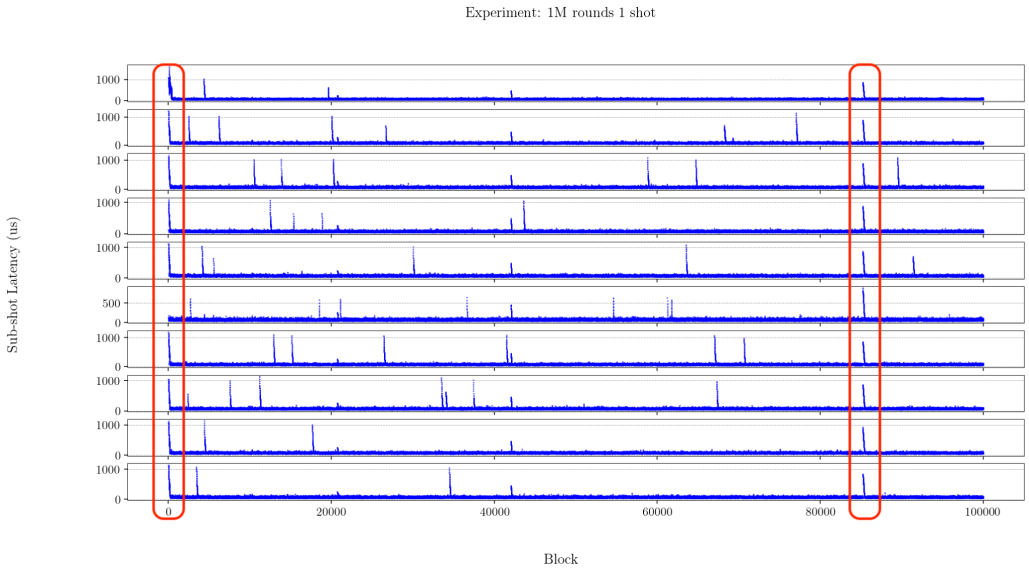
> In this work, the Google team reports on two experimental realizations of a surface code memory, implemented on two distinct superconducting processors. On a 105-qubit chip (Q105), they realize a distance-7 code (along with distance-5 and distance-3 codes on subsets of qubits) while they realize a distance-5 code on the 72-qubit chip (Q72). The main achievements are
> - On both chips, a surface code working below threshold, with a clear improvement by a factor $\Lambda \simeq 2$ in the logical qubit lifetime as the code distance increases by 2. On Q105, Λ is approximately constant when moving from distance-3 to 5 then 7
> - The distance-7 code implements a logical memory with coherence clearly beyond break-even
> - The distance-5 code on Q72 is decoded in real time, with performances only slightly below those of end-of-line decoders
> - To estimate performances in the case where the physical error rates would be retained while scaling up the chip size, the authors implement a distance-29 repetition code on Q72, and demonstrate that one type of logical error (either bit or phase flip) can reach a probability as low as a few 10^{-10} per correction cycle.
>
> The methods appear to be the same as those presented in a previous publication in Nature (Ref 17) except for improved calibration and decoding techniques. Besides that, the main changes are (a) a larger (105Q) chip allowing them to increase code distance to 7 (b) improved physical qubit lifetimes roughly yielding a factor 2 improvement in control and readout operations (c) superconducting gap engineering limiting chip-scale qubit failures due to high-energy impacts. However, neither this work nor Ref 17 give details on circuit layout and fabrication techniques, so that accurate comparison is impossible.
> This is undoubtfully a “landmark” paper: each achievement listed above is a world-first. Put together, they strongly support the claim that fault-tolerant quantum computing is now within reach (the authors are reasonably cautious about this claim, listing challenges ahead and highlighting that it supposes that performances can be retained and even improved at scale). This is one the most important results of the year (if not of the decade) in experimental quantum information. Therefore, I am very much inclined to recommend publication in Nature.

We thank the referee for their positive assessment of the paper’s overall significance, as well as their recognition of its different technical achievements.

> Nevertheless, the fact that some of the main results were obtained on only one of two different systems is in my view a serious weakness of the paper. In particular, it casts a doubt on the claim that decoding can be achieved in real-time with sufficient throughput and accuracy. Indeed, the spikes in decoder latency visible in Fig S11 (which I think should be mentioned in the main paper as they cannot be inferred from Fig 4) may be a sign that the distance-5 code is saturating the decoder throughput. If the authors are for some reason unable to acquire data on Q105 pertaining to real-time decoding, I recommend that they simulate real-time decoding on a distance-7 code by feeding the decoder offline with error-
> syndromes acquired on Q105. In the same vein, implementing a repetition code on the Q105 chip and reporting on the dynamics of correlated bursts of errors would greatly strengthen the paper as it would clearly confirm (or not) that chip-scale qubit failures have been robustly suppressed.

The spikes in latency are concerning, and something we are investigating. We have added to the main text to point out that spikes in latency in excess of 1 ms can occur, with reference to the supplementary information. However, we do not believe that these are caused by the distance-5 code saturating throughput.

We suspect that performance spikes during benchmarking fall into two categories. First are memory-related spikes, which may be caused by allocations from the C++ standard library. When these allocations are filled, the library needs to reallocate memory, typically doubling the size. Second are cold start spikes, which happen at the beginning of the process, possibly due to initialization activities (see below).



Concerning the error bursts, this is the subject of ongoing investigation, as there are T_1/T_2 experiments that can more directly probe the effects of these correlated error events. As the referee might imagine, we are very interested in characterizing and resolving these bursts. However, we have not yet run these tests across multiple chips.

> Before recommending publication in Nature, I would like the authors to address this point by either (a) taking more data (or at least simulating real-time decoding as detailed above) or (b) explaining why this data was not taken / whether it will be done in the near future / whether any preliminary result they may have reveals unforeseen challenges in extending some methods to the Q105 chip.

The reason for not performing real-time decoding on 105Q is fairly mundane: we had already attached the real-time decoding infrastructure and collected data on the d=5 chip when the d=7 chip became available. We do not anticipate any unforeseen challenges when extending to the 105Q chip. In particular, when running the real-time decoder, we only used 32 out of 64 cores, as this was more than sufficient to meet the throughput requirements for real-time decoding. On simulated real-time decoding with similar error rates and cycle times at distance-7, we observe a decoder latency of ~114 μs after a 10^6 cycle experiment. This is roughly consistent with processing twice as many measurements at distance-5. In the future, we will certainly test our full real-time decoding stack (both in simulation and experiment) at distance-7 and higher.

> I list below further comments/questions that may help improve the paper.
>
> 1) In Fig 2b, the authors fit ε_d versus p_{det} with power laws. The exponent of this power law is a fit parameter, found to be slightly below ideal value (d+1)/2. In Fig 4b, analyzing similar data from the repetition code, they adopt a different strategy by which the exponent value is fixed at (d+1)/2 and the prefactor is the only fit parameter. As a result, agreement between theoretical predictions and data degrades for short distance codes.
> A) Can the authors comment on these two different strategies? Do they understand why the power law seems to reach its ideal value for long distance repetition codes but not for code distances below 11? Do they expect a similar trend for the surface code?

We hypothesize that as the independent injected error diminishes with respect to the background device noise, correlated errors in the device are causing a deviation from the ideal power law. This hypothesis could explain the slight upward inflection of the points in the low

error regime of Fig. 3a. At higher distance repetition codes, we do not fit these low-physical-error points, and so do not see as substantial a deviation. In addition, one might expect less correlated error in the repetition code than the surface code as the former requires fewer calibrated two-qubit interactions.

> B) The authors should display the fitted exponent for the power law in Fig. 2b. It would also help to mention the ideal (d+1)/2 value in the caption (or represent it on the plot or refer to Eq. 1) as it took me a while to understand what the authors meant by “power law fits” (I initially assumed that the fitted lines had a curvature)

We felt that this makes the plot a bit busy and can be inferred from the text, but have included a Pauli simulation of the data in the response to Referee 1. This contrasts the observed exponents against ideal power law fits.

> C) About Fig 2b, the authors comment that “When fitting power laws below the crossing, we observe approximately 80% of the ideal value (d+1)/2 predicted by Eq. 1.” and “We find that the three curves cross near a detection probability of 20%, roughly consistent with the crossover regime explored in Ref. [17].” Are these observations related?

These are not directly related. The error model at the threshold crossing is dominated by the injected error. As we mention above, we believe that the deviation from the ideal power law is caused primarily by background device error.

> 2) Error injection is a powerful tool to analyze the device behavior. It looks like the authors could extend the method beyond what is presented in the paper.
> A) They inject only coherent errors which are said to be a good proxy for all uncorrelated errors. Can they generalize the method and inject each type of error (as classified in Table S6) separately? Fitting the slope of 1/Λ against the probability of each injected error, it looks like they could estimate the w_i's and compare them to the simulated values reported in Table S6.

This could be an interesting study, but could be somewhat expensive depending on how thorough we wanted to be when emulating different noise sources like leakage and crosstalk. We might consider this in the future.

> B) In the inset of Fig 2b, they should report the fitted slope of 1/Λ versus p_{det} and compare it to the relevant values w_i reported in table S6. A comment on the fitted line not intersecting 0 would also be welcome (it is understandable from supp. Mat. Eq. 5 but not from Eq.1 of the main text). A comment on the average p_{det} being a good proxy for single qubit error would also help (in the main text, the only justification is that 1/Λ scales linearly p_{det}, whereas important results on the negligible impact of inhomogeneous error rates are given in the supplemental materials).

We have added the slope to the caption. As we are injecting error between surface code cycles and preceding measurements, this would most closely correspond to the data qubit idle and measurement error components. However, it is worth noting that the errors we inject are quite different from the noise used in simulation. For example, we are injecting coherent Y-type errors between rounds, so we don’t necessarily expect a direct connection between the sensitivity of the device to the injected error and the sensitivity of the simulated device to different error sources. Regarding p_{det} acting as a good proxy for physical error, we have added references to the supplementary information as well as a relevant recent publication (arXiv:2408.02082).

> C) In Fig 3c (repetition code), the fitted slope should be given. Why does this fitted line intersect 0? Are correlated errors negligible for the repetition code?

We have added the slope. It is plausible that the repetition code suffers from less correlated error, as there are fewer gate layers to optimize, and the repetition code is sensitive to fewer error mechanisms. However, we do believe that some correlated errors occur in the repetition code as well.

> D) Could the authors also estimate more precisely the DQLR efficiency by injecting leakage errors? (they give only a very crude estimate for the transition probabilities $P_{i\to j}$ presented in the supplemental materials (bottom of p14)).

This study was previously undertaken in the paper that first introduced the DQLR method ("Overcoming leakage in quantum error correction." Nature Physics 19.12 (2023)).

> 3) The authors mention that simulations overpredict Λ by 20%. In Fig 1d, the simulated lambda is 2.25 versus 2.14/2.04 for the measured value. Where does the 20% comes from? In Fig 2c, indicating the simulated and measured 1/Λ values would improve clarity.
> The authors attribute this discrepancy to “excess correlations”: supporting this claim and characterizing correlated errors seems very important. In particular, do they remain local or “widespread” as the calibration of microwave drive crosstalks presented in the supplemental materials seem to indicate. Tying up with (1), is this correlated-error model compatible with observed logical error rates in the repetition code? Tying up with (2), can the impact of correlated errors be characterized by injecting these?

We apologize as this section of the paper was quite unclear, and contained an error. The simulated Λ of 2.25 was achieved using a correlated minimum-weight matching decoder, not the more accurate ensembled matching or neural-network decoders mentioned in the main text. When we decode the experimental data using this less accurate decoder, we achieve a Λ of 1.97. This corresponds to a 14% overestimate of Λ in the error budget. We have updated the text to clarify this point.

Understanding the source of the discrepancy is an ongoing research area. We note that microwave crosstalk as described in the supplementary information is a form of classical signal crosstalk, and from the perspective of each qubit, the crosstalk is not correlated with other qubits. However, over the course of the experiment, removing frequency collisions between spatially nearby qubits empirically seems to reduce the discrepancy between simulation and experiment, suggesting that small parasitic couplings leading to unintended swap interactions could be a source of correlated noise.

We do see some evidence of correlated error in the repetition code. In particular, as we fit Λ across higher distances (and correspondingly at lower error rates), we do observe that the value decreases. While we could try to inject correlated error, a more direct approach to characterize its impact might be to determine different correlations present in the data available. This is something we plan to look closer into.

> 4) About drifts and re-calibration, Fig 2e shows that the memory performances are stable over time. From the text, we learn that qubit frequencies are “optimized” before repeated runs (with details given in supplemental materials), and the processor is recalibrated every 4 runs. Does that mean that target frequencies are set once, and qubit biases adjusted every 4 runs to reach these frequencies?
> From supp. Mat. Fig S10, we understand that calibration and adjustment of the decoder priors are crucial to reach optimal performances. The authors give a lot of details on the configuration of decoder priors, which is appreciable. However, it is hard to get an idea of the time it takes to fully calibrate the system (including qubit frequency calibration and decoder configuration) and how often this should be done to maintain performances. Commenting on this “duty-cycle” and how it

> is expected to scale with chip size would give an idea of the challenges lying ahead when operating large-scale processors.

It takes ~30 minutes to optimize a d7 grid and <1 minute to re-optimize ~10 gates after performance failures (e.g. after new TLS / qubit collisions). Both optimization and healing can be made constant-time with parallelization. However, the reviewer is correct about the recalibration procedure: prior to our repeated runs, we choose a set of frequencies, and then every four runs we adjust the biases to maintain these frequencies, inferring the change in bias required using Ramsey measurements. Regarding the decoder calibration, we finetune the decoder for the nth dataset by training on the (n-1)th dataset. This training is slow, but is also not something we've optimized. For matching-based decoders, we can use correlation methods, which in the supplementary information we show are fairly accurate (compared with a longer reinforcement learning based approach), and can be performed relatively quickly.

> 5) About the real-time decoder, can the authors estimate how computing resources (classical) will scale with code size? Moreover, they mention that they herald failure of the decoder in the computationally most demanding cases, which has no impact on the logical error rate given the low probability of such events. Is this strategy viable for long distance codes or do they expect such failures to dominate before reaching the milestone of 10^-6 error per cycle?

The memory resources scale as $O(rd^2)$ where d is the code distance and r is the number of cycles kept in memory, which can be far smaller than the total experiment length. The average time complexity of the decoder is $O(d^2)$ per cycle on problem instances that arise in practice. This is significantly better than the theoretical worst-case complexity of the MWPM algorithm (see also arXiv:2303.15933).

The probability of heralded failures increases with the code distance. However, it also decreases exponentially with the number of cycles per block, because these failures result from error chains that span more than two blocks. Increasing the block size enables us to prevent heralded errors from having a significant impact on the logical error rate, and so we do not expect such failures to dominate.

> Other minor comments and typos
> - Below Eq.1, p and ε_d are defined as error rates whereas they are error probabilities per cycle throughout the paper.
> - Introduction: referring to Fig 1a when giving the number of physical qubit per logical qubit would help.
> - In fig 1a, 102 qubits are represented (including leakage removal qubits) when the chip is said to embed 105 qubits. Can the authors comment on that?
> - In the caption of Fig 1b, "Blue: Average identification error" seems at odd with the cumulative errors represented. Similarly, the weight-4 detection probabilities are said to be averaged. Please clarify.
> - Caption of Fig 1c: each individual code and logical basis is said to be fitted separately, but only one value is reported per code (not one per basis).
> - The authors mention the value of T_{2CPMG} in the main text and give a few details on dynamical decoupling techniques in the Supp. Mat. Is this CPMG time measured in the same conditions as the DD sequences presented in S21? (comment on this in the main paper ?).
> - In some figures (e.g. 2d, 3c) : fitted values are not given and cannot be inferred from other plots. The authors should give these values in captions or displayed on plots. - "preparing the data qubits in a product state in either the XL or ZL basis"-> "in a product state corresponding to a logical eigenstate of either the XL or Zl basis". - Fig 2a: comment on dashed vertical lines (representing interactions with other qubits ?)
> - The sentence "obtained by checking how long after each 10-cycle block's data is received that the block is completed by the decoder" is hard to parse.
> - L32 : "the logical error " -> "the logical error rate"
> - L133 and l160: "X and Z" -> "X_L and Z_L" ?
> - Caption Fig 2 : "where the codes cross"

We thank the referee for their attention to detail on these. We have taken many of their suggestions. Regarding Fig. 1a, note that there are three gray qubits at the top of the figure. Regarding dynamical decoupling, the value of $T_{2,CPMG}$ is a measure of the qubit memory time after echoing away low frequency noise, but without any consideration for the surface code context of measuring and resetting measure qubits alongside the data qubits being echoed (as described in the supplementary information).

Referee #3

The authors report on exciting result of surface code experiments using superconducting transmon qubits below the threshold for error correction.
These are novel results, which the community was actively pushing for, and a clean demonstration of the workings of quantum error correction.
They demonstrate the reduction of error with code distance on surface codes (bit and phase corrected simultaneously, up to d=7 with suppression factor ~2) and at larger distance on repetition codes (either bit or phase).
Various decoding methods are also described (see one point below though).

This could clearly mark a milestone for the field of quantum error correction, and quantum information processing in general. The results are timely, nicely described, confirm expectations of theoretical scaling, and demonstrate the error correction below threshold. The text is clear, data consistent, well referenced, and conclusions seem reliable.

We are grateful to the referee for their favorable evaluation of our paper content and its presentation.

I do have various points below that I think the authors should address though:

1. Improved processor
While I am impressed by the results, very little is said about what enabled the authors can achieve those. The comparison to the experiment of 2023, which was at about the threshold rather than a factor 2 below it as now, is brushed in lines 91 and 92 under 'improved fabrication techniques, participation ratio engineering, and circuit parameter optimization'.
Nothing is said about fabrication techniques of this work, nor of the previous one.
Participation ratio cannot be judged without seeing any device picture, qubit design, or details of the processor. Did the shape of the transmons islands change, the junction designs, the packaging, the film thickness?
I'm not even sure what circuit parameter was optimized in order to improve the coherence of qubits (crosstalk? Purcell protection? quasiparticle mitigation strategies? or is that referring to the frequency optimization in-situ, suppl mat II?).
Clarifying those aspects would go a long way in enabling reproducibility of the results.

While we cannot share all of the device improvements that contributed to this experiment, a few notable changes include increasing the characteristic qubit capacitor feature size from 30 μm to 150 μm, an increase in the qubit/resonator coupling to reduce MIST during readout and minimize idling error, and numerous cleanliness improvements owed to moving from a shared research cleanroom environment to a dedicated fabrication facility. We have added these, along with a few additional details, to the supplementary information.

2. Real time decoding
In line 80, comparing to the cycle time of 1.1us, and later in section V, the authors make claims of real time decoding.
They report factually on a decoder than processes information from the stabilizer measurements with a latency of 63us on average, which is 63/1.1 > 50 cycles.
The decoding latency not increasing with the number of cycles is a formidable achievement, but it makes it 'bounded', or 'finite', possibly 'near realtime', but I'd argue not truely realtime.
In particular, as the authors point out on line 393, the 50+ cycle latency makes the processor react with that delay in implementing non-Clifford gates. In that delay, according to Fig.1, somewhere around 6% (12%) logical error is realized at distance 7 (5).
Certainly this wants to be avoided at algorithmic levels, though I'm fully aware that the problem is far from impossible to solve (once a particular decoder is decided upon, one could implement it on a faster ASIC or FPGA).
I believe the claim should be toned down.

We believe that our use of the term "real-time" to refer to our decoding system is appropriate. The common definition of real-time decoding is one in which the response time of the decoder is independent of the length of the experiment. Consequently, a real-time decoding system is one that processes the syndrome information at a rate at least as high as the rate at which the

information is coming in, i.e. one that meets the throughput requirement. This is the key necessity, as it overcomes the backlog problem.

If the above delay is constant but large, the system is real-time but its logical clock rate is slow. This is indeed something to optimize, as it will increase both the time footprint and (in requiring lower error rates) the space footprint of a quantum computation.

That said, we agree with the referee that optimally the system's latency would be much lower. Reducing this latency is the long-term objective of our research and development work on the system. Some of that work may involve the use of custom hardware as anticipated by the referee.

> **3. Detection probability**
> Line 178, the detection probability is used as a proxy for the total physical error.
> I believe this should be a very good proxy, yet nothing is said about it.
> How good is it? The authors should be able to know how many error they inject and thus provide some hints on how to quantify this proxy (what is the fraction of injected error that is detected? Though I guess no correlated error was injected, and thus this aspect may remain unclear...).

To clarify, we mean that 1/Λ is roughly linear in the observed detection probability. We note the use of detection probability as a proxy for physical error has also been described in the recent work arXiv:2408.02082.

> **4. Suppression factor**
> I really expected the fit on the inset of Fig.2b to be a direct proportionality, according to line 43 is should be 1/Lambda ~p/p_thr.
> Here the authors have a significant offset. At p_det=0, 1/Lambda is about 0.15.
> How does that work? Is it related to the correlated errors? (It seems to imply that even interpolated to zero (detector) error on the processor, some finite logical error will remain, which likely the undetected part, possibly correlated?)
>
> (I also think lines 188-189 about the power law fitting belong to the main part of the figure, and should be therefore moved before the inset discussion.)

This is a reasonable hypothesis. In particular, in the presence of correlated error, we don’t necessarily expect the scaling we observe in the inset to continue. The error we inject is entirely independent. Were we able to hypothetically ‘scale down’ the device noise, we might find that correlated errors present in the device would become the dominant contribution. We see some evidence of this in the suboptimal power law exponents.

> **5. Leakage reduction efficiency**
> DQLR seems to help strongly distance-5 but not distance-3. I understand this fits with simulation in the supplementary material, but can one give an intuitive explanation of the reasoning in the main text? Is the key element leakage lifetime, or how it spreads across the lattice, ...?

Qualitatively, a single long-lived leakage event can cause failure by spreading errors across the lattice. We hypothesize that at larger distances, this high-weight correlated failure mechanism becomes more pronounced relative to the higher degree error suppression.

> **6. Noise floor at high-distance**
> The authors claim to find two different failure modes (line 299). The main text would benefit from a description of the incidence of each: How many shots are we talking about? Which mode dominates? (no reference to the suppl mat discussing it exists...)

> Besides, the separation into quartiles in Fig.3d is not really explained: it is neither clear what the quartile splitting is based on (errors during the given erroneous experimental run, averaged over all runs, averaged for a given failure mode?), nor what one learns from it (multiple but not all quartiles involved: is the relevance that many qubits are impacted, or rather that not all are?)

As we are already over the word limit, we have elected to insert a reference to the supplementary information where these different failure modes are discussed. Regarding Fig. 3d, the quartiles are split based on the average detection probability of different detectors, windowed over 10 rounds, for the three shots delimited by the gray lines. Indeed, one important message is that the detection event spikes are somewhat localized - not all of the qubits are involved. We've added a line in the figure caption to reflect this.

Referee #4

> The manuscript "Quantum error correction below the surface code threshold" presents a compelling demonstration of below threshold operation of a fault tolerant logical qubit. While there have now been numerous experimental demonstrations of quantum error correction, and several recent demonstrations of "beyond break even" performance, this is the first experimental realization of fault tolerance that, if scaled up, could implement quantum computation at scale. As such, this is a significant milestone in quantum computing.
>
> The authors present strong evidence that their two systems implementing surface codes of two sizes are operating below the code threshold. Their data shows that the logical error rate is exponentially suppressed by increasing the code distance according to the theoretically expected relationship. By artificially introducing additional errors they also show a traditional "threshold crossing" plot, with the limiting case of no additional errors clearly being well into the below threshold regime. Overall this evidence is clearly described and presented and in my view presents sound evidence to support the claim of operating below threshold.
>
> The most notable caveat to the data is that the detection probabilities (which the authors use as a proxy for the physical error rate) increase between code sizes between d=3 to d=7. If the physical error rate continues to increase with system size this could prevent larger scale error correction. The authors state that they expect the effects to saturate. Given the importance of this point to the extrapolation of their result to larger scales a little more discussion around this point may be warranted. (Note that this is not a caveat to the central claim of being below threshold, only about the viability of future scaling)
>
> In addition to this main result the authors provide additional experimental data on the limits of error suppression using a repetition code. They show an improved noise floor in comparison to previous demonstrations, and provide a nice analysis of the behavior of the limiting errors (although they are not yet understood). The noise floor is now low enough that, if it holds for the surface code, some fault tolerant computation in the "useful" regime may be possible even without resolving these mystery error sources.
>
> Finally the authors present data on real-time decoding, which was run in addition to the slightly higher performing offline decoding algorithms used in their main results. They show that decoder latency does not increase with time of operation, demonstrating that the throughput is sufficient to meet the data output rate. To my knowledge this is the first time such a demonstration has been shown, which in itself is an important result in quantum error correction.
>
> Overall this is a well-written, clearly presented manuscript. The data and its analysis appears sound. The results are of high significance and I recommend it for publication.

We thank the referee for the endorsement of our paper and for the suggestions below.

> Below are some minor suggested edits:
>
> - I would suggest elaborating on the justification of why the apparent increase in physical error rate with code size is expected to saturate, and not prohibit further scaling.

As we are already over the word limit, we have not elected to expand on the main text. But Fig. S21 shows an increase in detection event fraction as the boundary becomes a smaller fraction of the overall lattice, even in a simple Pauli noise model. Furthermore, the reference shows that in the presence of local stray interactions, an increase in the effective noise quickly saturates at moderate code distances.

> - The term "detection probability" is used to refer to the probability that each syndrome bit is flipped. The name could easily be misunderstood, and is not clearly defined in the main text.

The situation is slightly more subtle, as our Pauli frame can sometimes change e.g. due to the dynamical decoupling sequence. So a comparison of two bits can be the same, but indicate an error relative to the Pauli frame change. We have rewritten the line to try and clarify.

- The following sentence gave the impression that the d=5 code was beyond break-even but the d=7 code was not. My understanding is that both codes are beyond break even.
"Our distance-5 quantum memories are beyond break-even, with distance-7 preserving quantum information for more than twice as long as its best constituent physical qubit."

The referee's interpretation is correct, and we have rewritten the sentence for clarity.

- The caption of figure 4 refers to End-of-shot and sub-shot latencies. It was not clear to me what was meant by these terms. The main text only mentions and defines "decoding latency".

We have updated the caption for clarity.

Note: We received the second round of feedback from Referees #2 and #3 first and prepared a response and revision. Referee #1 also saw that response and revision while preparing their response. We thank all the referees for their timely feedback, which has improved the manuscript substantially.

Referee #1

> I am responding to the third manuscript draft dated 31st October and after having seen the second round of correspondence with referee 2 and 3.
>
> Much of the requested additional information and clarifications can now be found somewhere in the main text or supplementary material. In particular, the real-time decoder clarification and additional simulations are welcome, and all of my questions around Fig 2b have been clarified by the addition of Table S8 and Table S9 in the supplementary material.
>
> I recommend publication though with one extra minor suggestion

We thank the referee for recommending publication.

> When extracting this information will still be a difficult process for some readers, which can be eased by making direct connections between claims in the main text and the supp material. The authors have already implemented several changes in the decoding appendices that have helped. But the meaning of Fig 2b could be improved further, in particular the claims that “When fitting power laws below the crossing, we observe approximately 80% of the ideal value (d+1)/2 predicted by Eq. 1.” and also the claim that Eq (1) indeed approximately holds true.
>
> My current understanding is that the 80% claim comes from Table S8, and then taking the ratio of k and (d+1)/2, for each d, and then averaging to get approx 80%. Additionally, there is an assumption here that p and p_det are interchangeable in Eq (1), upto a constant. The following discussion assumes this.
>
> Then my suggestion is simply to add the ratio of k and (d+1)/2 into Table T8 as an extra column, and add to the caption add a remark such as “The values in the final column average to approximately 80%, leading the conclusion that the injected noise achieves 80% of the ideal value (d+1)/2 predicted by Eq 1. ”. A similar comparison should be made for Table S9, to justify the claim that simulated data is close to the ideal (d+1)/2 scaling, since my calculations indicate that there is a numerically meaningful (over 4%) deviation from the ideal.
>
> Indeed, the point I was making in my initial report that led to confusion was that one should compare the k values from S8 and S9 directly, rather than comparing S8 with Eq 1. However, this is not needed if (as proposed above) the Supp Mat makes it clear the Table S9 data indeed deviates from Eq (1).

We have implemented the suggested change in the supplementary tables S8 and S9 including their captions. We appreciate that the referee raised this point in the first place, as the addition of supplementary figure S25 helps contextualize the experimental results on surface code error injection.

Referee #2

> The authors have minimally addressed my concerns and questions, and when they did respond, their replies often did not lead to any changes in the manuscript. Therefore, I still do not recommend the publication.

We appreciate the referee's prompt response and attention to detail, and we believe our further revisions based on their feedback are improving the manuscript substantially. We are striving to respond comprehensively here and hope the referee will recommend publication with the revisions. We do note that we are constrained on manuscript length, which limits the scope of changes we could make in the main manuscript, and that this correspondence is intended to be made available to the public alongside the manuscript.

> To sum up my concerns, each claim in the abstract taken separately would merit publication in a high-quality journal. However, the collection of claims, as presented, is misleading:

We agree with the referee on the strength and impact of the results we present. We disagree with the characterization that the presentation of the collection of results is misleading. Let us try to assuage the referee's concerns with some additional context on our experimental progression before addressing the details.

To start with a brief summary, we had completed much of the work on the 72Q device when the 105Q device became available. We decided to then use the 105Q device, which was only available temporarily, to test for exponential scaling up to distance-7.

Elaborating, we began with the primary goal of maximizing surface code performance in our improved devices, as measured by $\Lambda_{3/5}$ and ε_5, arriving at the landmark $\Lambda_{3/5} > 2$. Meanwhile, we pursued a series of real-time decoding integration tests on the same system and revisited the repetition code, each a substantial undertaking. We decided to write a paper based on these results.

Separately, our team was testing device scaling and began making 105-qubit devices of similar design with an intent to iterate on relatively short time scales. Within the span of one of these iterations, we were able to conduct our distance-7 experiments as presented in the text, but not additional follow-up work such as real-time decoding or investigations of large-scale correlated errors. Given our team's finite resources, we chose to prioritize continuing to iterate on devices towards advancing our technology, rather than repeating all of the experiments on the distance-7 device. We considered sticking with the 72Q data alone, but we decided including the distance-7 data added an important message for the future of QEC, where the exponential behavior is consistent across three data points (which isn't clear with two).

The addition of surface code data on a second larger device does not detract from the merit of the data and claims from the first device. The larger device allowed us to reproduce two of the central claims of the paper: $\Lambda > 2$ (below threshold) and exceeding the lifetime of the constituent physical qubits. We view the fact that we demonstrate $\Lambda > 2$ on two processors to be a strength of the work and evidence of the field maturing technologically. We hope this added context will

help the referee understand our presentation of results from the two devices, and that the lack of additional comparisons between the two systems was due to a resourcing constraint, nothing more.

> a) Real-time decoding is used on one of two systems only.

This is true, and hopefully the additional context and analysis alleviate the referee's concerns. We agree it would be nice to include real-time results at distance-7, but we do not have those results and don't feel it should delay the publication of this paper. Ultimately, we will have to scale far beyond distance-7, and so we do not feel that the lack of a distance-7 real-time decoding demonstration significantly detracts from the paper, particularly given the simulated distance-7 real-time decoding results. We believe the presentation is clear on this point and have made revisions based on the feedback to improve it further (see below).

> Why not include the responses to my questions on real-time decoding (and those of referee 1) in the paper or its supplemental materials?

We appreciate the feedback and are adding more of this content. We are making an effort to select the best comments to add to the main text and supplement, deferring to the editor on length and welcoming further additions. In addition to the previous revisions, from our previous responses to referees 1 and 2, we have:

1. Added the comment about the estimated 10 μs data transfer time to the main text.
2. Added the limited core usage to the supplement.
3. Added a plot showing the streamed distance-7 data latency that the referee has asked for in the supplement.

> Why not test the real-time decoder on actual data from the d=7 code (offline) and not on simulated data?

We find the latency measurements most persuasive when running on very long experiments, while the experimental data we have at distance-7 only goes up to 250 cycles. In our previous response, we checked the latency on simulated distance-7 runs up to 10^6 cycles. We appreciate the referee's interest here and have now also tested distance-7 experimental data by stitching together 250-cycle experiments into longer runs. We have added latencies of both the distance-7 simulated and experimental real-time decoded data to the supplement. We note that the simulated latency is slightly shorter than reported in our previous response as we refined our error model to more closely match the device. Furthermore, the experimental latency is slightly longer than the simulated latency (108 μs versus 95 μs) likely because decoding the experimental data is somewhat harder.

> b) Error-burst suppression is also investigated in a single system only. The authors simply reply that this is "the subject of ongoing investigation". This is a central claim of the paper and while I personally do not doubt the results or the seriousness of the authors' work, as a referee I cannot accept what looks like cherry-picking of data over multiple systems.

We now understand that by “error-burst suppression,” the referee is referring to overcoming the previously-observed bursts manifesting in a 10^{-6} repetition code error floor (Google Quantum AI, *Nature* **614**, 676-681 (2023); McEwen et al., *Nat. Phys.* **18**, 107-111 (2022)). In our previous response on this topic, we were speaking to the newly-discovered, rarer error bursts. We apologize for misunderstanding the referee’s comment (specifically “implementing a repetition code on the Q105 chip and reporting on the dynamics of correlated bursts of errors”).

Addressing the previously-observed problem with gap engineering is studied in depth in a different paper (Ref. [40] https://arxiv.org/abs/2402.15644, among others in the literature). It is not a novel claim of this work. The fact that our repetition code moved substantially beyond the previous 10^{-6} floor was a confirmation of the previous results employed in the error correction context. We have not yet measured repetition codes on a 105-qubit processor, but we see no cause for concern, given the separate compelling study demonstrating the effectiveness of gap engineering and the shared design and fabrication processes for our processors.

We hope the additional context in our response will clarify the situation for the referee. We note that it was a substantial practical challenge to take 2×10^{10} cycles of state-of-the-art repetition code data, and as discussed previously, our timeline did not permit a repeat on the 105Q processor. We would also point out that we discovered a novel problem in the 10^{-10} floor and highlighted it in the manuscript.

At the referee’s suggestion, we have added another clarification to the main text that the observation of error below 10^{-6} was on the 72Q device, in addition to the existing statements in the text and figure.

> c) It is said in the abstract that the repetition code “probes the limits of error-correction performances” implying that scaling the surface code to d=29 would result in a similar error-rate. However, the surface code seems to behave differently from the repetition code (unknown error sources revealed by the power laws of Fig 2b and the offset of the 1/Λ vs p_det curve in the inset).

We thank the referee for pointing this out; we did not intend that implication. We have modified the language “To probe the limits of our error-correction performance” in the abstract to clarify.

> The authors replied to one of my questions that “We do see some evidence of correlated error in the repetition code”, but if they refer to the plateau at the 10e-10 level, this looks like a completely different mechanism from the hypothesized correlated errors on the surface code.

We apologize for the confusion here, as there are multiple sorts of correlated errors under discussion which manifest in different ways. We’d first like to re-address the referee’s original question, *“is this correlated-error model [which we claim is responsible for the discrepancy between simulation and experimental surface code lambda] compatible with observed logical error rates in the repetition code?”* We do not have a concrete physical picture of the correlations which may be causing discrepancies in our models; if we did, we would include them in our simulations. Thus, in general, we do not know if the surface code discrepancies are consistent with repetition code data.

In the response below, we discuss one particular hypothesis for excess correlations in the surface code. However, note that the surface code has more layers of CZs, giving the opportunity for more unexpected interactions, and the surface code is also sensitive to more error mechanisms than the repetition code. As noted by the referee, scaling a repetition code should not imply that scaling a surface code will behave similarly, and we have taken steps in the main text to remove this implication.

Finally, to clarify our remark in the first referee response, we were primarily referring to the deviation of logical error from the exponential fit line which becomes visible at distance-15 (10^{-8} logical error), not the plateau at 10^{-10} .

> As for their point that correlated errors appear in the surface code due to a larger number of “two-qubit interactions”, this mechanism is included in their error-budget and does not explain the discrepancy with the observed value of 1/Λ.

It is true that we try to account for stray CZ interactions in the 1/Λ error modeling, but those models do not necessarily catch all of the unwanted interactions happening on the device. For example, our models of stray CZ interactions only account for interactions between neighboring CZ pairs, but not potential interactions between non-neighboring CZ pairs. We believe such interactions are present but characterizing them is an ongoing topic of research which was motivated by this work.

We have updated the text to highlight longer range interactions as a potential source of the unmodeled discrepancy. Note also that such interactions would be difficult to controllably engineer, precluding our ability to study them with injection.

> Including the comments on real-time decoding in the paper and putting a few dB of attenuation on the claim on ultra-low error regimes (for instance mentioning that the repetition code probes some particular type of correlated errors appearing as random bursts, and that this is done on a single system) are minimal changes for me to accept the paper.

We thank the referee for their concrete requests here, and hope that our changes satisfy their criteria.

> A few final comments that I would very much like the authors to take into account:
> - All fitted values should be reported somewhere (or said to be irrelevant), otherwise fitted lines are merely guides to the eye. The offset in the 1/Λ vs p_det curve in 2b looks very relevant to me.

In addition to the reported logical metrics, we have added the fit parameters for the 1/Λ fits (Figs. 2b, 3b) to the figure captions in the main text, and we have added several tables of fit parameters to the supplement.

> The Pauli simulation included in the response to referee 1 could indeed give an idea of the fitted exponents, but it does not appear anywhere in the paper.

We have added a version of this plot to the supplement.

- The error budget in Fig 2c should indicate the actual value of 1/Λ found in the experiment. In the current version of this figure, it looks like the error-budget is fully understood (even though details are given in the text).

We have added an “Excess” bar to Fig. 2c to illustrate the discrepancy.

Referee #3

> The authors have addressed many of my comments in their rebuttal letter in a satisfactory way. I wish they had taken the opportunity to feed these insights into the main article, where they would be more visible to the readers (especially when multiple reviewers picked on it).

We appreciate the referee's prompt reply. We are unfortunately already over the word limit in the main article. Nevertheless, we are making more additions in this round of revisions based on the feedback, and we defer to the editor on length and additions.

> In particular about the decoding system. I now follow the author's definition of real-time as "not increasing backlog", which is used in some literature but definitely not all. Eg Fig1 of https://iopscience.iop.org/article/10.1088/2399-1984/aceba6 feeds back in the same error correction cycle. Or https://arxiv.org/pdf/2303.04846 has "short reaction time" as one practical requirement of real-time decoding.

We thank the referee for considering our definition and for looking at specific examples in the literature. Considering those references, we believe our results are consistent with their understanding of real-time.

The first is Battistel et al., *Nano Futures* **7** (2023) 032003. The referee notes that the schematic in Fig. 1c depicts feedback in the same cycle. Such rapid feedback would certainly be nice; then logical operations would not be slowed down by decoding. Reading the text, we do not believe the authors intend to make this a requirement. Here are some supporting passages that we feel are consistent with our treatment.

> **Abstract:** *"The decoder must be accurate, fast enough to keep pace with the QEC cycle (e.g. on a microsecond timescale for superconducting qubits) and with hard real-time system integration to support logical operations. As such, real-time decoding is essential to realize fault-tolerant quantum computing and to achieve quantum advantage."*
>
> **Figure 1 caption:** *"The time from the last considered measurement to an impending logical non-Clifford gate is the available time for real-time decoding."* This clarifies they do not require same-cycle feedback and reinforces the idea that more latency can slow down how rapidly one could execute non-Clifford gates.
>
> **Section 2:** *"For superconducting and photonic qubits, a high operational rate implies that the decoding throughput and latency are paramount [21, 42]. Throughput (defined as the time it takes the decoder coprocessor to perform the decoding once it receives the measurement outcomes) is even more important than latency (defined as the time it takes to send the measurement outcomes to the decoder coprocessor) [41, 43]. In recent superconducting-qubit experiments, QEC rounds were performed every ~1 μs [21–24], leading to an estimate that a utility-scale quantum computer would generate a few tens of Mbps of syndrome data per logical qubit, for a total of many Tbps [44]. For each logical qubit, the decoder needs to process that data at the acquisition rate or close to it to avoid an exponential slowdown due to an ever-growing data backlog."*
>
> **Figure 3:** Measurement outcome rates are listed as "~Tbps" (consistent with superconducting qubits, see above). Feedback to correction is listed as "~1 μs - 1 ms."

The second is Bombín et al., https://arxiv.org/abs/2303.04846 (2023). As the referee notes, they do identify "short reaction time" as desirable, and we agree, faster is better. We think our experiments demonstrate a good result in that regard and are actively working on improving the latency further, but there is no hard requirement listed on how fast it must be. We highlight some relevant passages below.

> **Background and motivation:**

> *“However, decoded outcomes need to become available throughout the computation. The decoding process must quickly provide partial results on the logical outcomes in terms of available physical outcomes. In general, the entirety of the quantum circuit is not even defined as subsequent quantum gates may be chosen depending on logical outcomes obtained. This means that, in practice, decoding can not be modeled as a monolithic offline process but rather online as a streaming transformation of physical outcomes into logical outcomes.*
>
> *“The time taken from the moment relevant physical measurement outcomes become available to the time the decoded logical outcomes can be used to control further logic is called the reaction time. The reaction time should be kept short, as it can potentially limit the rate of logic gates (such as T-gates), as illustrated by the example of arbitrary angle rotations (see Fig. 1). Contributions to the reaction time come from the classical processing time required for (partial) decoding as well as communication latency.*
>
> *“If the decoder can not keep up with the outcome data rate (also called throughput), the quantum computation may be forced to idle (i.e., perform identity gates) until a conditional follow-up operation can be determined, and in the process generate more syndrome data to be processed. This approach leads to a decoding back-log [2, 3] growing exponentially with the depth of the logical adaptivity and the reaction-time growing exponentially as the quantum circuit progresses. Thus, the decoders processing throughput must fully cover the outcome data throughput to keep the reaction-time constant (and hopefully small).”*
>
> **Motivating example:** *“To motivate the need for low-latency decoders (i.e., low reaction time), we consider the problem of gate synthesis in surface code fault-tolerance. This refers to approximating arbitrary single-qubit gates as products of T gates (π/8-rotations in the Z basis) and Clifford operations. This decomposition is representative and of widespread practical relevance. As we will see, a natural implementation of this sub-routine involves adaptive sequences of operations wherein each operation is conditioned on a logical measurement outcome from the immediately preceding one. This motivates the need for a decoding approach which can provide logical measurement outcomes as soon as possible.”*

In the context of latency limiting logical clock rate (the processor would need to wait for the decoder to catch up to apply a non-clifford gate I assume), the reply to reviewer 2 that the latency scales with distance is a practical problem. This wasn't clear in my first read, and still wouldn't be without reading the rebuttal letter. I would find some comments about latency, its implications and scaling justified in the article.

We agree with the referee that this should be mentioned explicitly. We have added it to the discussion on decoder latency in the main text. Overall, we hope the additions in the latency paragraph clarify the situation and reinforce our alignment with the referee’s references.