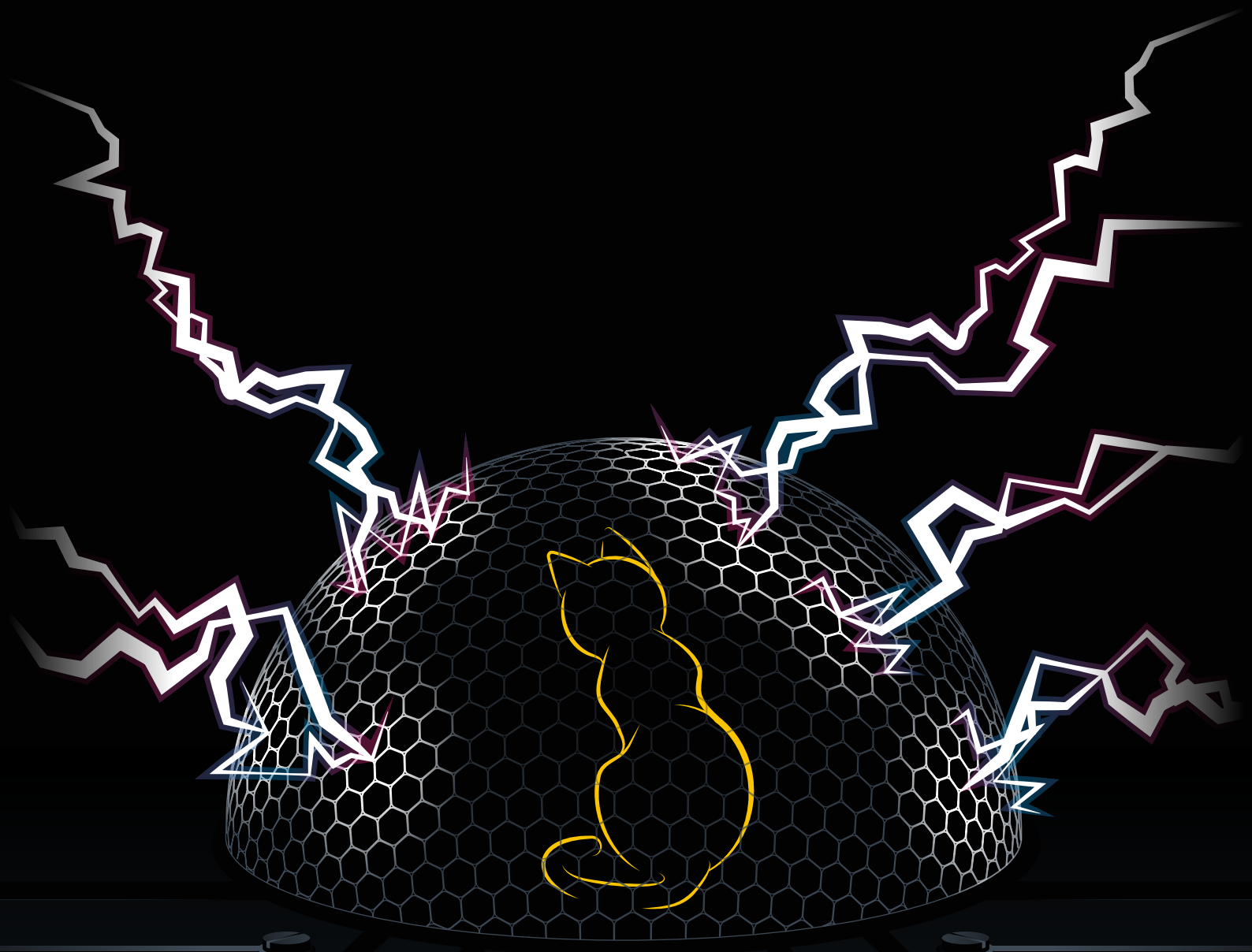
DEFINING THE LOGICAL QUBIT

FIVE CRITERIA TO BENCHMARK LOGICAL QUBIT CLAIMS



ALICE & BOB

WRITTEN BY

Wes Roberts, Charbel Eid

CURATED BY

Sacha Magee, Brian Siegelwax

ILLUSTRATED BY

Guillaume Gennet

To know more visit Alice & Bob's website

www.alice-bob.com

© June 2026. All images are all rights reserved and their use is prohibited.
The text is openly licensed via CC BY-SA 4.0, appropriate credit must be given to Alice & Bob.

Why do we need a standard definition of a "logical qubit"?

The quantum computing industry is doubling down on Fault-Tolerant Quantum Computing (FTQC) as the way to **build utility-scale systems**, and **logical qubits are the first step** along that path.

Despite significant progress in the development of logical qubits, **the term is used in a variety of ways across the industry** and research community, thus making claims difficult to benchmark. Each demonstration forces us to dig into the underlying technical details, which is no easy task. To help establish a common framework, this report defines what we believe constitutes a useful logical qubit.

Scientifically speaking, however, **a logical qubit is the result of quantum error correction** (QEC). These methods detect and correct errors on physical qubits before they can spread, using many physical qubits to create a smaller number of higher-quality "logical" qubits. QEC improves logical error rates, though errors are never entirely eliminated. Thanks to QEC, logical errors can be made so rare that useful applications with error-corrected, or so called "fault-tolerant" quantum computers, become possible.

IN THIS REPORT, WE AIM TO GIVE INVESTORS, ANALYSTS, AND ENTERPRISE DECISION-MAKERS A DEFINITION OF A USEFUL LOGICAL QUBIT. WE'LL PROVIDE FIVE ESSENTIAL CRITERIA THAT CAN BE USED TO "SCORE" ANY PROPOSED LOGICAL QUBIT, FOCUSING ON THE PROPERTIES THAT WILL BE REQUIRED FOR FTQC.

These modality-agnostic properties will allow comparisons across the various FTQC approaches, and provide a practical sense of how challenging it is to build logical qubits.

While we will focus on memory – a logical qubit's ability to preserve information over time – it's important to note that FTQC will also require manipulating logical qubits to perform computation. We'll cover that in a later report.

Ultimately, the goal of FTQC research is for logical qubits to become qubits reliable enough that long quantum computations can be run on them. **These criteria are guidelines** for assessing whether an experimental logical qubit ticks the right boxes **to scale into a true building block of utility-scale FTQC.**

Why do we need logical qubits to begin with?

With classical computers, we can protect against errors simply by making copies of a bit. When we read all of these copies later, we check whether any bit disagrees with the majority of the other bits. If it does, we assume an error has occurred and correct it. All of the copies once again share the same value.

Unfortunately, we cannot protect quantum information in this way. Quantum mechanics prevents us from copying qubits and from directly reading the information on them. The core concept is the same, however, and **we have devised ways to store a single bit of quantum information within a system of physical qubits**. The errors that occur on the physical qubits used to store a logical qubit can be detected and corrected in a way that does not destroy the quantum bit of information stored in the logical qubit.

The logical qubit is created from physical qubits using a quantum error-correcting code. The process of mapping information into the code (initializing the logical qubit) is called the "encoding". The information encoded in a logical qubit could be read out by decoding the code into a physical qubit, and reading the physical qubit out. As this would expose the logical information to noise, and therefore destroy it, the logical qubit information is usually directly read out from the code itself. QEC is the full process of encoding, protecting, and reading out the logical information. **FTQC is the process of computing on the logical qubits** while the information is encoded and therefore protected.

CLASSICAL ERROR CORRECTION

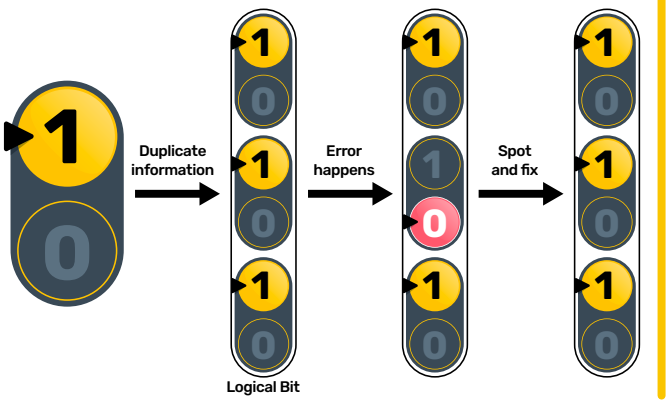
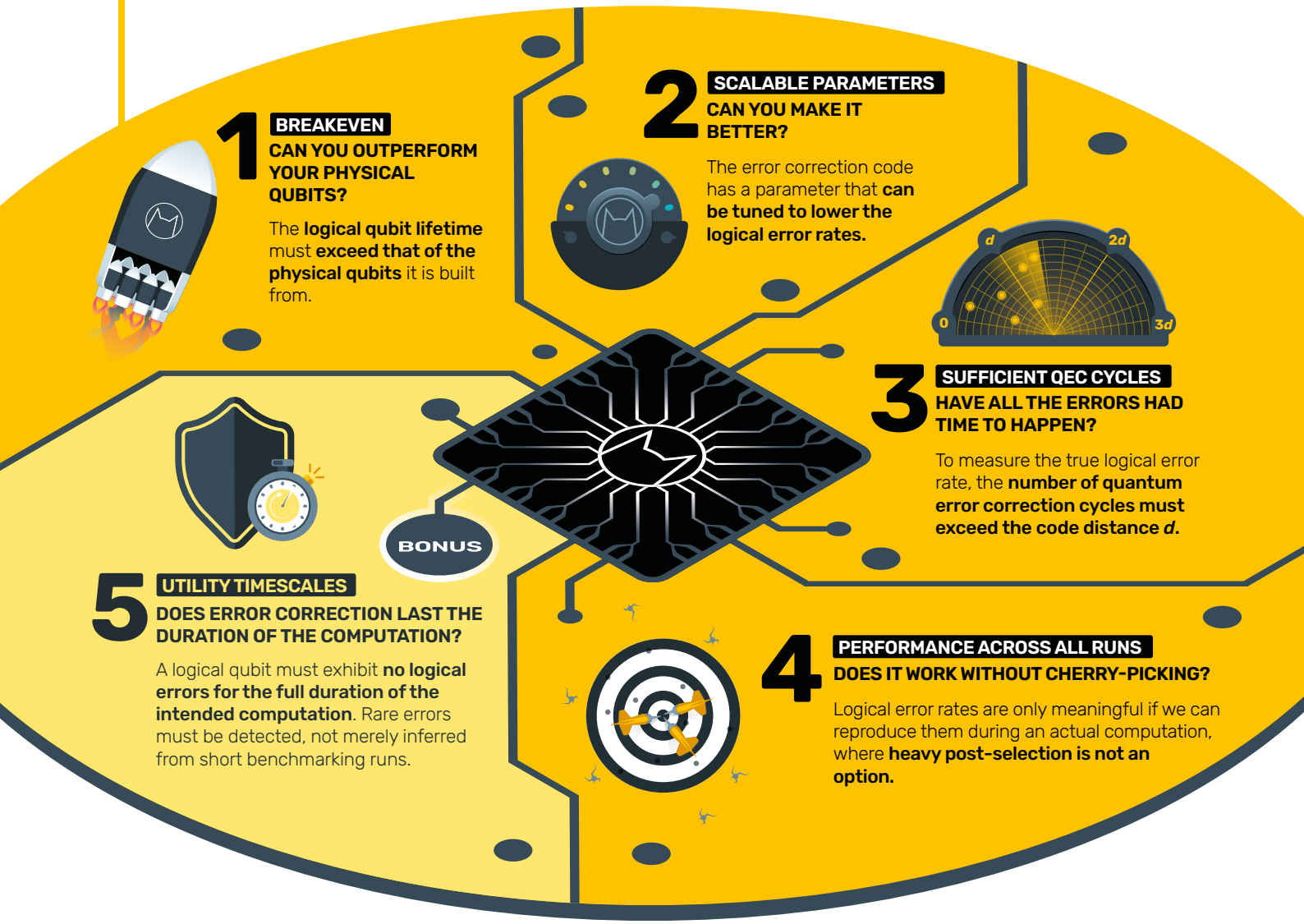


Figure 1: *The repetition code is used in classical computing to protect information in very noisy channels. This example shows a simple case where each bit is redundantly encoded three times. If one of the bits is corrupted and flips, it can be identified through a majority vote. The corrupted bit can then be flipped back, correcting the error.*

What makes a logical qubit useful?

A logical qubit is not just a collection of physical qubits and an encoding scheme; it is the full machinery involved in the error correction process. To form the basis of a fault-tolerant quantum computer and be application-relevant, **a logical qubit must meet the following requirements**.

The 5 Criteria to Define a Logical Qubit



1 • BREAKEVEN
CAN YOU OUTPERFORM YOUR PHYSICAL QUBITS?

The lifetimes of the logical qubit should be longer than the lifetimes of the physical qubits it is built from.

"Breakeven" is the point at which the logical qubit achieves the same error rate as its underlying physical qubits. "Below breakeven" results from physical qubits being too noisy, or from the error-correction process introducing too many additional errors, such that the logical qubit performs worse than its physical qubits in every respect. If a logical qubit experiences errors less frequently than its best physical qubit, it has achieved "beyond breakeven". To check this, both logical and physical error rates must be measured experimentally. ■

NOTABLY, GOOGLE QUANTUM AI DEMONSTRATED BEYOND-BREAKEVEN LOGICAL LIFETIMES USING A SURFACE CODE[1], SATISFYING CRITERION 1. THIS RESULT WAS AN IMPROVEMENT OVER ITS PREVIOUS WORK[2].

A beyond-breakeven logical qubit is better than every physical one

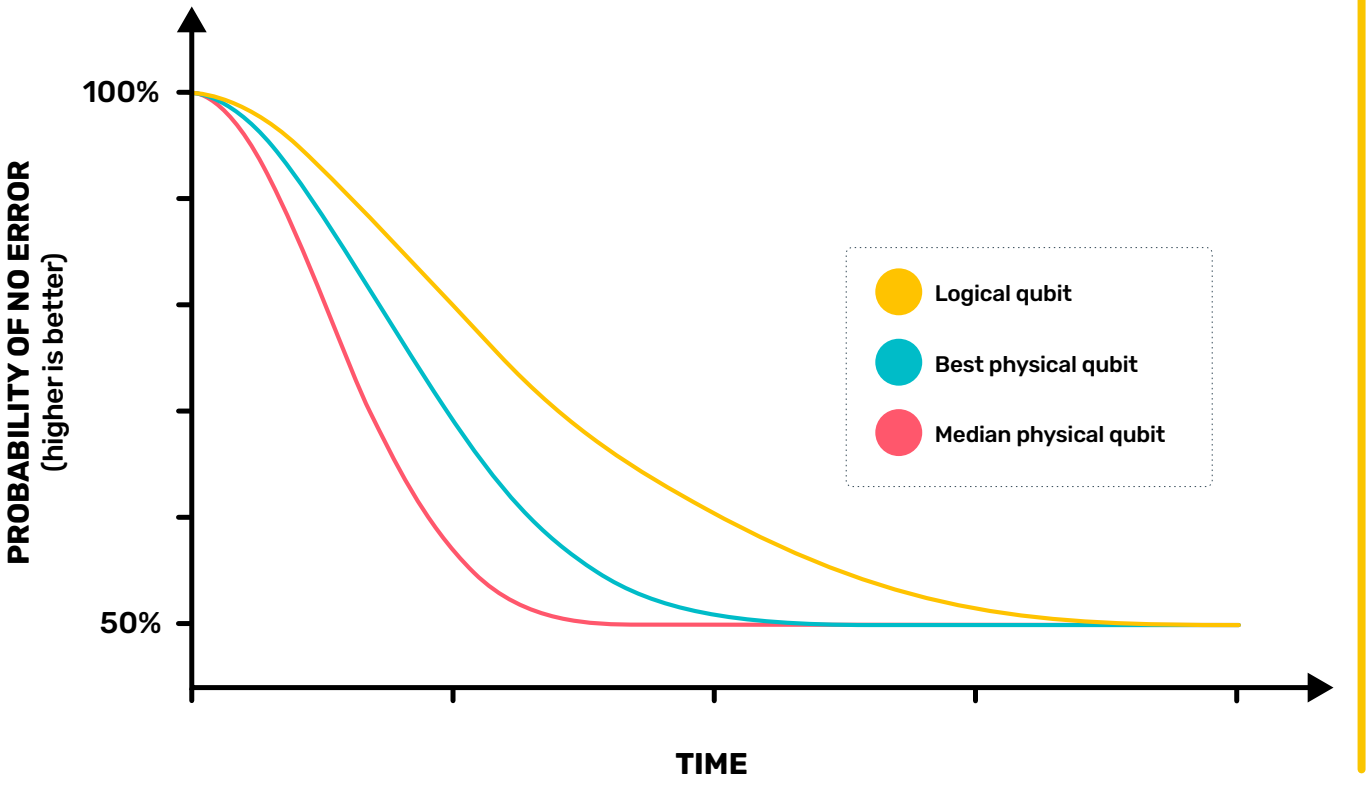
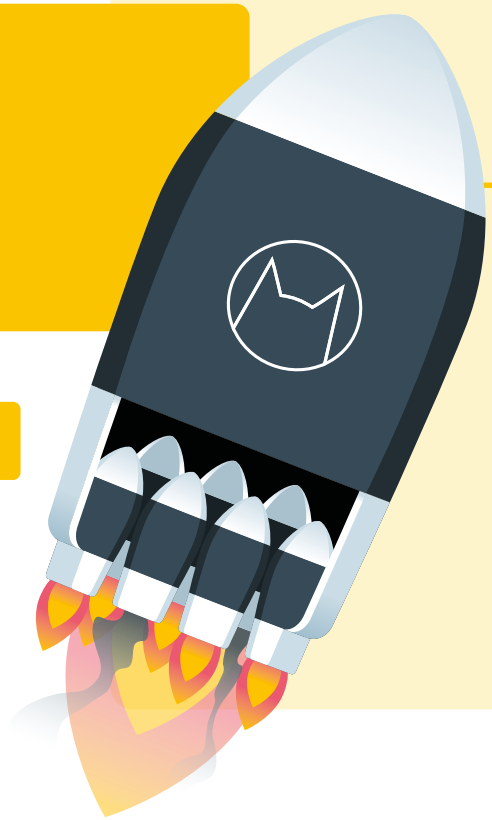


Figure 2: *Over time the probability of measuring the correct result decreases. As in this illustrative example, the error correction is "beyond breakeven" when the logical qubit decays more slowly than the best-performing constituent physical qubit.*



HOW TO CHECK

Find the measured physical and logical lifetimes and compare them. Logical lifetimes should exceed the physical lifetimes.

GO BEYOND **The threshold theorem: good enough physical qubits**

Counterintuitively, an error-corrected logical qubit can perform worse than its physical qubits. Error correction only helps if the underlying hardware is good enough to support it.

Returning to the classical repetition code example *(Figure 1)*, the bit is stored redundantly three times. If one of those bits flips and is now in error, the other two bits still agree, the errant bit has been identified and can be corrected. The problem arises when two bits flip. The lone correct bit is assumed to be the one in error, and we flip it so that all three bits are now in error. Two errors are less likely than one, so repetition codes usually offer an improvement, but if hardware has high enough error rates, adding redundant bits can create more opportunities to fail. The encoded system can thus become less reliable than a single bit.

Quantum error correction follows the same principle. The mathematics is more complicated, but the idea is the same: below a certain error threshold, error correction helps; above it, it makes things worse.

2 • SCALABLE PARAMETERS
CAN YOU MAKE IT BETTER?

The error correction code is part of a parametrized code family, where a different choice of parameters would lead to lower error rates.

The error-correcting code defines the logical qubit. It requires tunable parameters - we speak here of a code family rather than a single code - **that can be tuned to achieve longer logical lifetimes.**

A common such parameter is the "code distance", which is proportional to the number of errors that can occur on the physical qubits before the logical qubit experiences an error. More precisely, for a code distance *d*, the maximum number of errors that can occur before the logical qubit experiences an error is (*d*-1)/2.

The code can be thought of as a blueprint for constructing a logical qubit, where parameter choices that improve performance often require blueprints that are more challenging to build.

Some qubits have their own specific tuning parameters. For example, parameters that change the internal encodings of the physical qubits themselves[3]. **The key point is that the code family provides a way to improve performance.**

Parameters like code distance can be tuned to achieve lower error rates

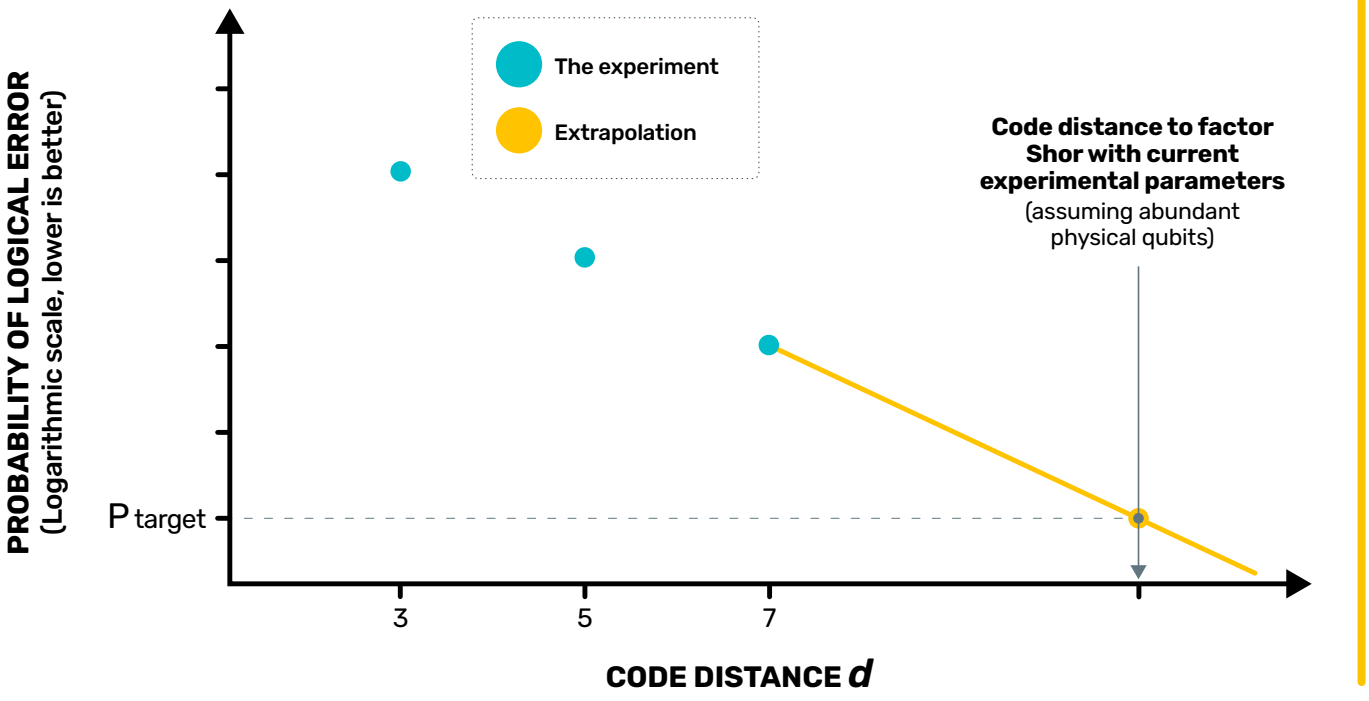


Figure 3: *One example of a scalable parameter is the code distance d, which in this illustration can be increased in order to reach lower logical error rates. For a given application (say Shor's algorithm) that requires a particular error probability P_{target}, the code distance can be increased in order to reach this target probability and run the algorithm. Since we are close to the threshold, the code distance required to reach the target logical error rate may end up being very large. This is why reducing the physical error rate matters, as the further below the threshold, the more tractable the required code distance becomes.*

HOW TO CHECK

This criterion is specific to the methods of error correction in use, which can be multiple physical qubits encoded as a logical qubit, physical-level stabilization, or a combination of both. For code-level error correction, check whether the code belongs to a parametrized family of related codes whose performance improves as resources are added. For physical-level error correction, look for physical parameters, such as the number of photons, or state squeezing, that can be adjusted to improve performance. In practice, physical-level parameters are usually combined with code-level parameters.

3 • SUFFICIENT QEC CYCLES
HAVE ALL THE ERRORS HAD TIME TO HAPPEN?

To measure the true logical error rate, the number of quantum error correction cycles has to exceed the code distance *d*.

The error correction process does not just detect and correct errors; it also introduces new ones. A logical error can only occur when many physical errors have happened at roughly the same time, on many of the physical qubits. Those error patterns typically spread in time over a certain number of rounds that can be up to the code distance *d* (or even a few times the code distance *d*). As a consequence, running too few error correction cycles allows only a fraction of the relevant logical errors to actually occur. In other words, **the error correction experiment was stopped "too early" for some logical errors to happen**, leading to an underestimated extrapolation of the logical error rate from the measured failures. The number of cycles needed to discern the true logical error rate is proportional to the code distance *d*. To reliably infer the logical error rate, **the experiment should last for at least a few multiples of d rounds of error correction**.

Measuring the true logical error rate requires running a sufficient number of QEC cycles

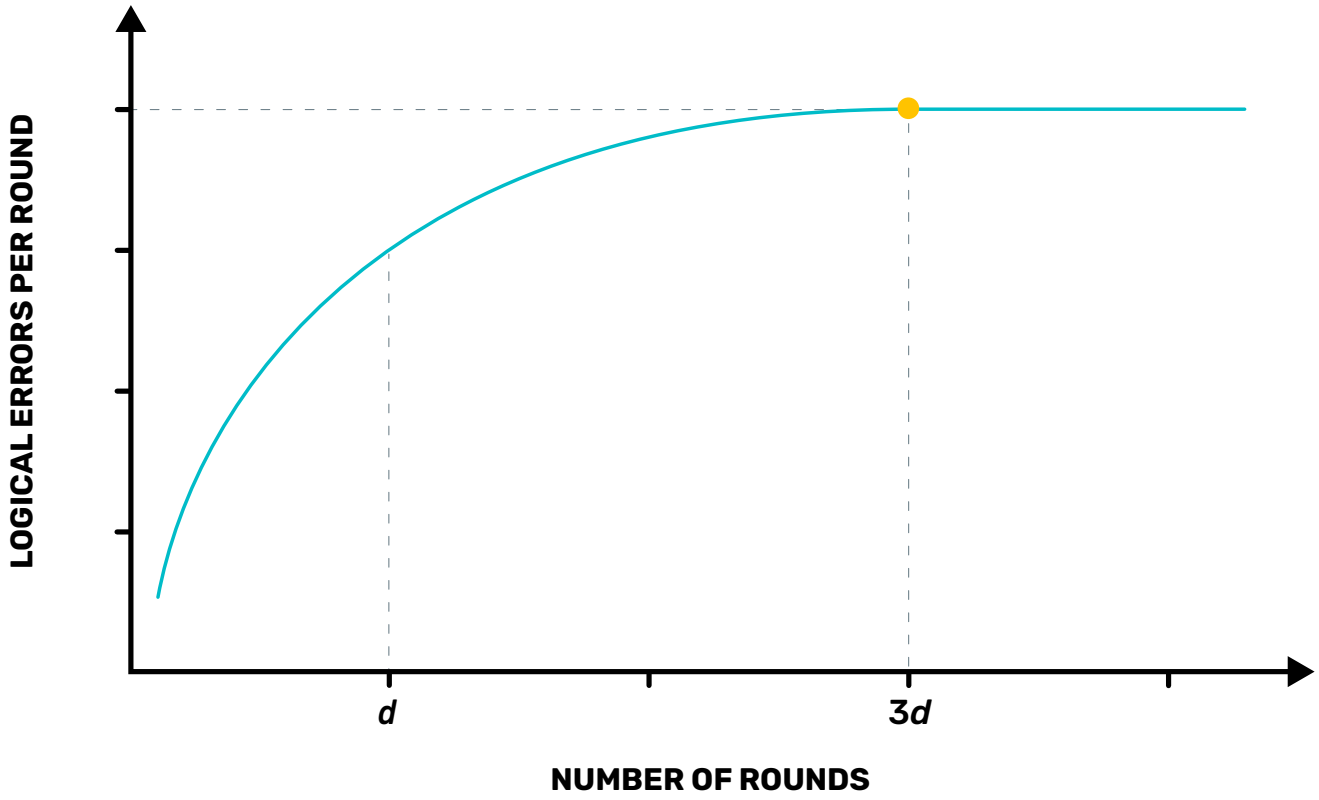
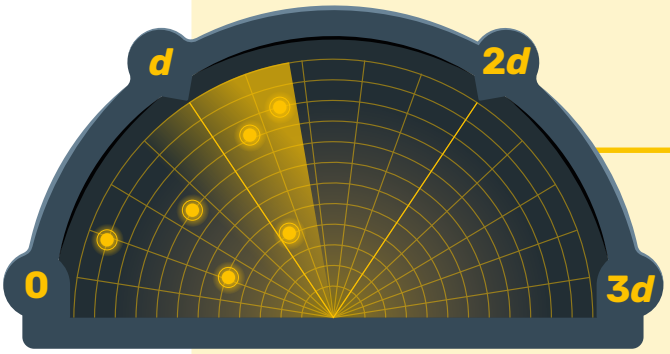


Figure 4: *As this illustrative plot shows, when we run an experiment with too few error correction rounds, we risk measuring an artificially low logical error rate. As we increase the number of rounds in the experiment, our measured value approaches the true logical error rate. To ensure we make a reliable measurement, we should run the experiment for at least d rounds, with d being the code distance.*



HOW TO CHECK

Look for the number of quantum error correction (QEC) cycles that were run to test the logical qubit. If this is not clearly stated, you can calculate it by dividing the total runtime by the QEC cycle time.

The code distance will be standard for the error correction code being used. Compare it to the number of QEC cycles, which should ideally be much higher than the code distance.

4 • PERFORMANCE ACROSS ALL RUNS
DOES IT WORK WITHOUT CHERRY-PICKING?

Claimed logical error rates are only meaningful if we can reproduce them during a utility-scale computation. Error detection experiments that rely heavily on post-selection should not qualify.

The method used to test the logical qubit and obtain the logical error rate must mirror what actually happens while QEC is running during a utility-scale computation. **If anything is discarded like logical qubits or runs, the logical error rate would be underestimated** and not representative of a real scenario.

Errors can happen at three distinct points in the QEC process:

- **Between QEC rounds**, either naturally or induced by operations
- **During physical measurements** and error correction operations, due to faulty QEC hardware
- **While decoding** the measurement results, attributing them to one error when they were induced by another

If the full QEC process is not run, then we ignore many of the ways in which errors can occur, and our measurements do not accurately reflect how the logical qubit would perform under normal operating conditions. **We cannot correct the simple errors and discard the difficult ones**. We cannot just discard qubits entirely because they are in error. FTQC requires quantum error correction, not quantum error discarding. ■

Post-selection can lead to underestimation of utility-relevant logical error rates

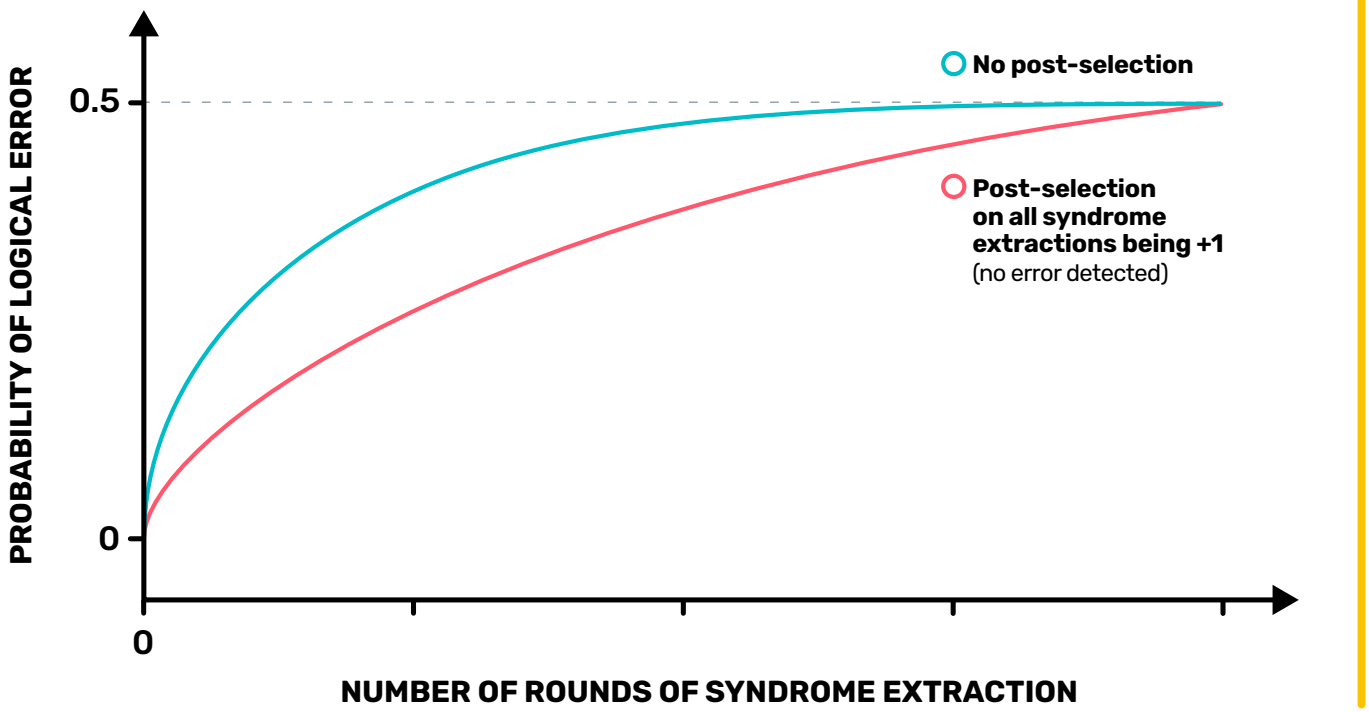


Figure 5: *The probability of a logical error increases as we perform more rounds of syndrome extraction, because this process can itself introduce errors. This illustration shows that by using post selection, where we discard runs in which an error is detected, we will measure a lower probability of logical error than without post-selection. However, this would not be representative of the expected logical error rate in many utility applications where post-selection cannot be utilized.*

HOW TO CHECK

Look at the experimental process that was used to measure the logical lifetimes. See if there were continuous QEC cycles throughout the experiment and that all rounds were kept. Watch for mentions of "post-selection" or "discarded rounds." Also, check for a graph that shows a "success probability" that typically decreases exponentially as the experiment progresses.

GO BEYOND **Syndromes and decoders**

It can be useful to understand how measurement results are interpreted, and turned into decisions about what should be corrected.

To diagnose an error, we first measure "syndromes". Syndrome qubits are adjacent to the data qubits that store the quantum information. We measure 0 or 1, with 1 indicating that an error has been detected. However, we don't yet know the error's location.

For that, we need all the syndrome measurements. By piecing together all the detection events, we identify regions in space and time where something must have gone wrong. We then use a decoder to infer the most probable cause of each specific error.

Unfortunately, some syndrome measurement outcomes can look the same even though the errors are different, and some can even look as though no errors occurred; therefore, the decoder is making a best guess. Like the other parts of the logical qubit architecture, however, it would "only" need the decoder to guess correctly often enough to achieve the targeted logical error rate.

The logical qubit's lifetime inevitably sets an upper limit on how long a fault-tolerant computation can run. Should an error reach the logical level, the computation fails, since no further layer of error correction exists to "catch" it, and the computation must be started again from scratch. This leads us to our final criterion, though it is better understood as a "bonus": necessary in principle, yet something of an end in itself.

5 • UTILITY TIMESCALES

DOES ERROR CORRECTION LAST THE DURATION OF THE COMPUTATION?

A logical qubit must exhibit no logical errors for the full duration of the computation intended to be run on the final machine. Under continuous operation, it must experimentally show that low-frequency errors have actually been detected, rather than merely inferred from short benchmarking runs.

Quantum error correction assumes that noise is local in space and time. Fault-tolerant computation on logical qubits requires that this assumption remains true throughout the computation.

Unfortunately, **there are rare physical error processes that cause non-local errors which are not correctable**. If the frequency of such errors is low compared to the duration of the experiment, then most runs of the experiment will not experience them. If the frequency is high, however, then such errors are likely to occur as soon as we try to run practical large-scale computation.

This is a consequence of quantum error processes not being fully known. Physical phenomena such as the passage of certain high-energy cosmic rays can cause multi-qubit errors hourly, or even more frequently. **A qubit experiment that is run for a shorter time than the target computation may not be subjected to errors that could still affect the computation**.

Error correction must account for all error types, including low-frequency ones

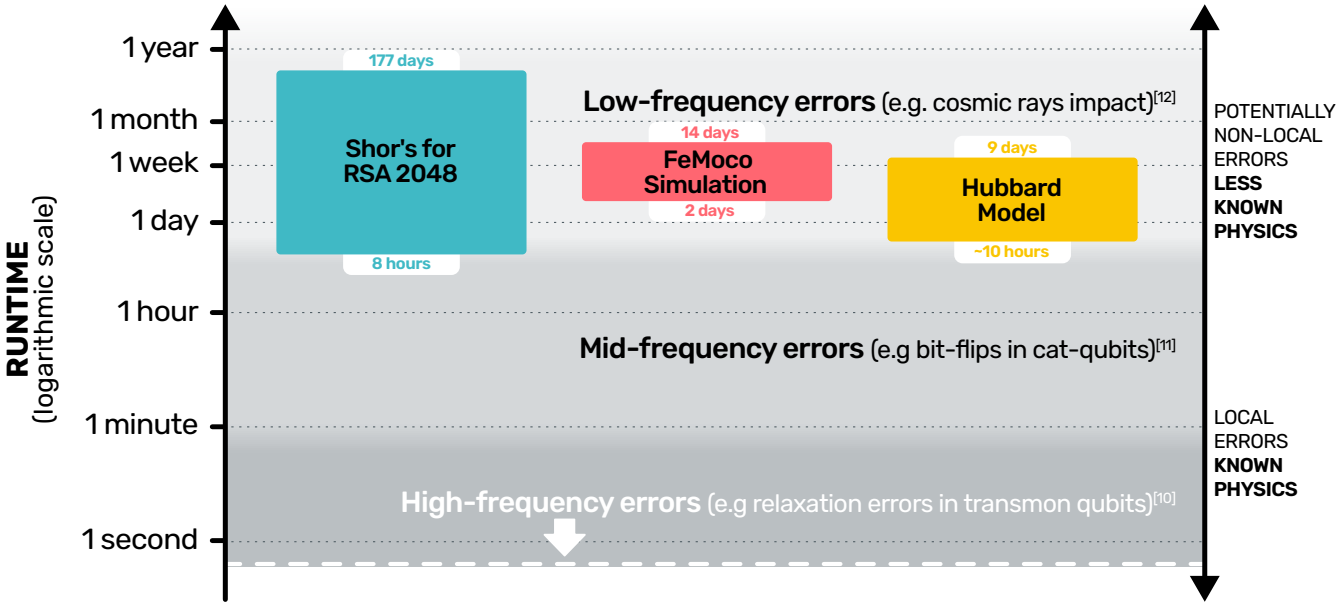


Figure 6: *The colored bars show estimated runtimes for large-scale quantum algorithms - starting at a few hours: Shor's algorithm for RSA-2048 [4,5], FeMoco simulation [6,7], and the Hubbard Model [8,9]. At these longer timescales, the underlying physics is less well understood and qubits may be subjected to very low-frequency, rare errors. Some of these errors are thought to be potentially non-local (they can spread across distant qubits). To rightly account for such categories of errors, an experiment should approach, as closely as possible, the expected runtime of the target algorithm.*

HOW TO CHECK

Look for how long the experiment was run: are the results per round or do they cover multiple rounds? Depending on the eventual goals of the research team, the utility timescales should be anywhere from an hour to a week. If not stated, utility timescales should generally be implied.

Conclusion

A logical qubit is a nuanced concept, but it can be defined rigorously. We should hold it to the same high standards that utility-scale applications will ultimately require.

Core to the logical qubit's identity is the use of a quantum error correcting scheme to encode a unit of quantum information. However, **the use of an error-correcting code is not by itself sufficient to determine its potential** for use in real FTQC applications.

The criteria proposed in this report are intended to provide a consistent framework for evaluating logical qubit claims across different platforms, as they currently stand.

As we move into the era of early FTQC, we will see a greater need both for clarity in the way quantum researchers and vendors communicate their results, and for awareness around the importance of those results.

WE NEED TO ADOPT A STANDARDIZED LANGUAGE FOR CONVEYING PROGRESS IN FTQC, STARTING WITH HOW WE APPLY THE TERM "LOGICAL QUBIT".

Glossary

PHYSICAL LIFETIME
For a specific type of error, the average time it takes (in seconds) for this error to occur on a physical qubit. During multiple lifetimes the same error can happen multiple times reversing its own effect.

PHYSICAL ERROR RATE
For a specific type of error, the average number of times an error occurs per second on a physical qubit. Inverse of the physical lifetime.

LOGICAL LIFETIME
For a specific type of error, the average time it takes (in seconds) for this error to occur on a logical qubit.

LOGICAL ERROR RATE
For a specific type of error, the average number of times an error occurs per second on a logical qubit. Inverse of the logical lifetime.

ERROR DETECTION MEASUREMENTS
Measurements applied to the physical qubits to check for errors. They do not measure the qubits directly as to not destroy the state. Instead they are "joint measurements" applied in ways to extract the correlations between multiple qubits.

ERROR CORRECTION CYCLE
One round of syndrome extraction, in which an error detection measurement is performed. A fault-tolerant computation will typically involve many such cycles.

ERROR PROBABILITY PER CYCLE
For a specific type of error and for the given duration of an error correction cycle, the effective probability that an error occurs during the given time. (Takes into account events where an error happens twice canceling itself)

CODE DISTANCE
A measure of the error-correcting capability of a quantum error-correcting code. It is proportional to how many physical errors can be tolerated before a logical error happens. For the 5-qubit repetition code for instance, the code distance is simply 5.

Bibliography

[1] *Google Quantum AI and Collaborators, "Quantum error correction below the surface code threshold," Nature, vol. 638, pp. 920–926, 2025. doi: 10.1038/s41586-024-08449-y.*

[2] *Google Quantum AI, "Suppressing quantum errors by scaling a surface code logical qubit," Nature, vol. 614, pp. 676–681, 2023. doi: 10.1038/s41586-022-05434-1.*

[3] *A. L. Grimsmo, J. Combes, and B. Q. Baragiola, "Quantum computing with rotation-symmetric bosonic codes," Phys. Rev. X, vol. 10, p. 011058, 2020. doi: 10.1103/PhysRevX.10.011058.*

[4] *C. Gidney and M. Ekerå, "How to factor 2048-bit RSA integers in 8 hours using 20 million noisy qubits," Quantum, vol. 5, p. 433, Apr. 2021. doi: 10.22331/q-2021-04-15-433.*

[5] *É. Gouzien and N. Sangouard, "Factoring 2048-bit RSA integers in 177 days with 13 436 qubits and a multimode memory," Phys. Rev. Lett., vol. 127, no. 14, p. 140503, Sept. 2021. doi: 10.1103/PhysRevLett.127.140503.*

[6] *J. Lee et al., "Even more efficient quantum computations of chemistry through tensor hypercontraction," PRX Quantum, vol. 2, no. 3, p. 030305, July 2021. doi: 10.1103/PRXQuantum.2.030305.*

[7] *V. von Burg et al., "Quantum computing enhanced computational catalysis," Phys. Rev. Research, vol. 3, p. 033055, 2021. doi: 10.1103/PhysRevResearch.3.033055.*

[8] *N. Yoshioka et al., "Hunting for quantum-classical crossover in condensed matter problems," npj Quantum Inf., vol. 10, p. 45, 2024. doi: 10.1038/s41534-024-00839-4.*

[9] *R. Toshio et al., "Practical Quantum Advantage on Partially Fault-Tolerant Quantum Computer," Phys. Rev. X, vol. 15, no. 2, p. 021057, May 2025, doi: 10.1103/PhysRevX.15.021057.*

[10] *M. Tuokkola et al., "Methods to achieve near-millisecond energy relaxation and dephasing times for a superconducting transmon qubit," Nat. Commun., vol. 16, p. 5421, 2025. doi: 10.1038/s41467-025-61126-0.*

[11] *U. Réglade et al., "Quantum control of a cat qubit with bit-flip times exceeding ten seconds," Nature, vol. 629, pp. 778–783, 2024. doi: 10.1038/s41586-024-07294-3.*

[12] *J. M. Martinis, "Saving superconducting quantum processors from decay and correlated errors generated by gamma and cosmic rays," npj Quantum Inf., vol. 7, p. 90, 2021. doi: 10.1038/s41534-021-00431-0.*

ALICE & BOB

contact@alice-bob.com