nature physics

Article

https://doi.org/10.1038/s41567-023-02325-8

Time-Efficient Constant-Space-Overhead Fault-Tolerant Quantum Computation

Received: 19 September 2022

Accepted: 7 November 2023

Published online: 16 January 2024

Check for updates

Hayata Yamasaki [1,2,3] & Masato Koashi [4,5]

Scaling up quantum computers to attain substantial speedups over classical computing requires fault tolerance. Conventionally, protocols for fault-tolerant quantum computation demand excessive space overheads by using many physical qubits for each logical qubit. A more recent protocol using quantum analogues of low-density parity-check codes needs only a constant space overhead that does not grow with the number of logical qubits. However, the overhead in the processing time required to implement this protocol grows polynomially with the number of computational steps. To address these problems, here we introduce an alternative approach to constant-space-overhead fault-tolerant quantum computing using a concatenation of multiple small-size quantum codes rather than a single large-size quantum low-density parity-check code. We develop techniques for concatenating different quantum Hamming codes with growing size. As a result, we construct a low-overhead protocol to achieve constant space overhead and only quasi-polylogarithmic time overhead simultaneously. Our protocol is fault tolerant even if a decoder has a non-constant runtime, unlike the existing constant-space-overhead protocol. This code concatenation approach will make possible a large class of quantum speedups with feasibly bounded space overhead yet negligibly short time overhead.

Fault-tolerant quantum computation (FTQC) establishes a way to realize quantum computation, achieving useful computational acceleration compared with conventional classical computation, even in the presence of intrinsic noise[1,2]. To solve computational problems for an M-bit input, quantum computation may exploit a quantum circuit of polynomial size $O(\mathrm{poly}(M))$ in width and depth. If we run this original circuit directly on physical qubits, noise-induced errors may destroy the result of quantum computation. A fault-tolerant protocol reduces the effect of errors by simulating the original circuit on logical qubits of a quantum error-correcting code using an adequate number of physical qubits. Conventionally, using concatenated codes such as Steane's seven-qubit code[3,4] or quantum low-density parity-check (LDPC) codes such as the surface code[5–7], fault-tolerant protocols can arbitrarily suppress the error rate on logical qubits if that on physical qubits is below a certain threshold[5,8–16]. However, error suppression in the conventional protocols requires a growing ratio of the number of physical qubits per logical qubit, that is, a space overhead[17], which scales polylogarithmically in M and diverges to infinity. In practice, the number of physical qubits available for a quantum device is severely limited, and the space overhead has been a major obstacle to realizing quantum computation[18–20].

The fault-tolerant protocols also require extra runtime for implementing logical quantum gates in terms of the circuit depth, that is, a time overhead[17]. A class of conventional protocols can achieve a polylogarithmic time overhead with transversal implementation of Clifford gates and gate teleportation of non-Clifford gates[1,2,21–25]. The gate teleportation is assisted by auxiliary qubits that are to be prepared in fixed logical quantum states in a fault-tolerant way while executing

[1]Department of Physics, Graduate School of Science, University of Tokyo, Tokyo, Japan. [2]Institute for Quantum Optics and Quantum Information, Austrian Academy of Sciences, Vienna, Austria. [3]Atominstitut, Technische Universität Wien, Vienna, Austria. [4]Department of Applied Physics, Graduate School of Engineering, University of Tokyo, Tokyo, Japan. [5]Photon Science Center, Graduate School of Engineering, University of Tokyo, Tokyo, Japan. e-mail: hayata.yamasaki@gmail.com

the computation. In such conventional protocols, the gates are applicable to all logical qubits at a time, that is, parallelizable. Each logical gate is applied within a polylogarithmic time overhead, which can be considered to be negligibly small compared with a polynomial runtime of quantum computation.

The reduction of space and time overheads in FTQC is crucial for realizing wide-ranged applications of quantum information processing and hence has been of great interest from both practical and theoretical perspectives. In contrast to the conventional protocols, theoretical progress in refs. 17,26,27 has shown that the space overhead can indeed be made constant by using quantum expander codes[28–31], a family of quantum LDPC codes with a non-vanishing rate of logical qubits per physical qubit. However, unlike the conventional protocols, the protocol in refs. 17,26,27 has a drawback that the gates are not completely parallelizable, that is, are applicable only to an asymptotically vanishing fraction of the logical qubits at a time. As a result, sequential gate implementation is imposed, leading to a polynomially growing time overhead[17,27]. A key open problem in the field of FTQC, originally raised in ref. 17, is whether we can resolve this apparent trade-off between the space and time overheads in FTQC within the law of quantum mechanics. Simultaneously with the constant space overhead, it would be critical to achieve a strictly less time overhead than polynomials of arbitrarily small degree, so as not to ruin a large class of useful quantum accelerations including polynomial as well as exponential ones.

The establishment of a low-overhead protocol by resolving such a trade-off has been challenging as long as existing techniques have been used. While the existing constant-space-overhead protocol[17,26,27] implements gates by gate teleportation, error suppression requires a large code block, and its non-parallelizability arises from the fact that the state preparation required for gate teleportation has been hard for such a large code without relying on conventional concatenated codes after all[17]. Without such concatenated codes, non-fault-tolerant state preparation, for example, by using the protocols in refs. 32–34, would suffer from more errors as the code becomes larger, which is infeasible on large scales. However, because concatenated codes incur a growing space overhead, complete parallelism in the fault-tolerant state preparation has been impossible within constant space overhead[17]. Another gate implementation method for quantum LDPC codes may be to use code deformation[35], but it is unknown whether such an implementation can be faster than gate teleportation owing to the extra runtime of the code deformation. A more recent method based on lattice surgery requires many auxiliary qubits for complete parallelization over all the logical qubits, which ruins the constant space overhead[36]. Note that, for another family of quantum LDPC codes, that is, hyperbolic toric codes[37], parallel gate implementation may be possible[38,39]. However, it is unknown whether these codes can feasibly realize FTQC owing to the lack of an efficient decoder to decide how to recover from many errors within a feasible runtime[17,40]. Linear-distance quantum LDPC codes with non-vanishing rates have been developed more recently[41–43], but no time-efficient gate implementation method is known for these families of codes.

Even more problematically, to prove the existence of a threshold for fault tolerance, the analysis of the existing constant-space-overhead protocol assumes that classical data processing, which is used in the decoder and the gate teleportation, for example, can be performed instantaneously in zero time[17]. In practice, physical experiments towards realizing FTQC are indeed challenged by the fact that implementation of classical computation has non-zero runtime that grows on large scales[44,45]. However, with finite classical computational resources incurring such growing runtime, the time overhead and even the existence of a threshold of the existing constant-space-overhead protocol are still unknown.

To address these problems, in this Article, we develop an alternative fault-tolerant protocol that simulates a $O(\mathrm{poly}(M))$-size circuit within a constant space overhead $O(1)$ and only a quasi-polylogarithmic time overhead $\exp(O(\mathrm{polylog}(\log(M))))$. This time overhead is substantially smaller than polynomials for arbitrarily small degrees, that is, that shown for the existing constant-space-overhead protocol[17,26,27]. This advantage is important in realizing useful polynomial quantum speedups without polynomially large slowdown. Remarkably, our analysis of time overhead takes into account the waiting time for the non-zero-time classical computation during FTQC. The novelty of our protocol is to use a concatenated code with a non-vanishing rate constructed from a sequence of different quantum codes, rather than using a quantum LDPC code. In the following, we show a non-vanishing rate for our code, an efficient decoder, an implementation of universal quantum computation, the existence of a threshold, and the space and time overheads, followed by the conclusion.

Concatenated code at non-vanishing rate

The crucial technique here for achieving a constant space overhead is to construct a concatenated code $\mathcal{Q}^{(L)}$ with a non-vanishing rate from a sequence of L different quantum codes, where L denotes the total number of concatenations.

We introduce the code $\mathcal{Q}^{(L)}$ as follows (Fig. 1). With $N_r := 2^r - 1$, let $\mathcal{C}_r$ ($r = 2, 3, \ldots$) denote the family of $[N_r, N_r - r, 3]$ Hamming codes[46], with block length of N_r, dimension of $(N_r - r)$ bits and distance 3 (ref. 47). Quantum Hamming codes $\mathcal{Q}_r$ ($r = 3, 4, \ldots$) (ref. 48) are Calderbank–Shor–Steane (CSS) codes[2,4,49] of $\mathcal{C}_r$ over its dual code $\mathcal{C}_r^{\perp}$, which are in a family of $[[N_r, K_r, 3]]$ codes having $N_r = 2^r - 1$ physical qubits (that is, code size N_r), $K_r := N_r - 2r$ logical qubits and distance 3. We use $\mathcal{Q}_{r_l}$ with parameter $r_l := l + 2$ for the concatenation at each level $l \in \{1, \ldots, L\}$, which leads to the sequence $\mathcal{Q}_3, \mathcal{Q}_4, \ldots$ of quantum Hamming codes starting from Steane's seven-qubit code $\mathcal{Q}_3$ (refs. 3,4) at level 1, with rate converging to $K_{r_l}/N_{r_l} \to 1$ as $l \to \infty$.

We construct $\mathcal{Q}^{(L)}$ recursively by defining $\mathcal{Q}^{(l)}$ for $l = L, L-1, \ldots, 1$. Let $K^{(l)}$ and $N^{(l)}$ denote the numbers of logical and physical qubits of $\mathcal{Q}^{(l)}$, respectively, which turn out to be $K^{(l)} = \prod_{l'=1}^{l} K_{r_{l'}} = \exp(O(l^2))$ and $N^{(l)} = \prod_{l'=1}^{l} N_{r_{l'}} = \exp(O(l^2))$. We define a collection of $K^{(l)}$ logical qubits of $\mathcal{Q}^{(l)}$ as a level-l register, where a physical qubit is referred to as a level-0 register (or level-0 qubit). The recursive relation between $\mathcal{Q}^{(l)}$ and $\mathcal{Q}^{(l-1)}$ is presented in Fig. 1. In particular, we divide the $K^{(l)} = K^{(l-1)} \times K_{r_l}$ qubits in the level-l register for $\mathcal{Q}^{(l)}$ into $K^{(l-1)}$ blocks of K_{r_l} qubits. Then, for each $k^{(l-1)} \in \{1, \ldots, K^{(l-1)}\}$, picking the $k^{(l-1)}$th qubit from each of the N_{r_l} level-$(l-1)$ registers for $\mathcal{Q}^{(l-1)}$, we encode the K_{r_l} qubits in the $k^{(l-1)}$th block into the picked N_{r_l} qubits as the logical qubits of the quantum Hamming code $\mathcal{Q}_{r_l}$. We design this concatenated code so that, even if all the qubits in one of the level-$(l-1)$ registers suffer from correlated errors, we can still recover the encoded level-l register. In this way, we obtain a $[[N^{(L)}, K^{(L)}, 3^L]]$ concatenated code $\mathcal{Q}^{(L)}$. It should be noted that, unlike quantum LDPC codes, the ability of our code to suppress errors is not fully characterized by the code distance, owing to its concatenated structure.

We analytically prove that $\mathcal{Q}^{(L)}$ has an asymptotically non-vanishing rate $R(L) := K^{(L)}/N^{(L)} \overset{L\to\infty}{\to} \prod_{l=1}^{\infty} K_{r_l}/N_{r_l} \geqq 1/\eta_\infty > 0$ with a finite asymptotic overhead factor η_∞. The overhead factor, that is, the inverse of the rate, would diverge to infinity for the conventional concatenated and quantum LDPC codes, such as the surface code. By contrast, we numerically show that $\eta_\infty < 36$ in Fig. 2. Note that η_∞ is not optimized here. For example, the factor could be almost 1 by starting the concatenations from $\mathcal{Q}_{r_8}$ rather than $\mathcal{Q}_3$. Even at lower concatenation levels $L \leqq 4$, the advantage of $\mathcal{Q}^{(L)}$ over the $[[7^L, 1, 3^L]]$ concatenated seven-qubit code in the overhead factor can be orders of magnitude. Nevertheless, both codes can suppress the logical error rate doubly exponentially as $L \to \infty$, as shown later with our threshold analysis. See Supplementary Section B for details of our construction and analysis.

Note that the above construction of concatenated codes with non-vanishing rate is, in principle, applicable to other sequences of codes with rates approaching 1 sufficiently fast. For example, our

https://doi.org/10.1038/s41567-023-02325-8

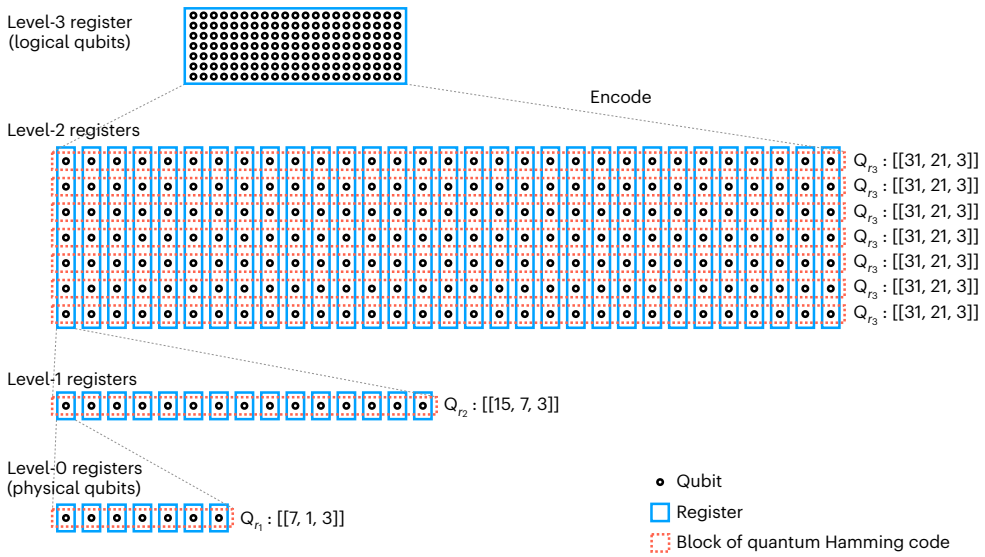

Fig. 1 | The construction of the concatenated code $\mathcal{Q}^{(L)}$. We concatenate a sequence of L different quantum Hamming codes $\mathcal{Q}_{r_1}, \mathcal{Q}_{r_2}, \ldots, \mathcal{Q}_{r_L}$ ($L = 3$ here). For each concatenation level $l \in \{L, L-1, \ldots, 1\}$, recursively, qubits (circles) in a level-l register (blue rectangle) are encoded into those of N_{r_l} level-$(l-1)$ registers using $K^{(l-1)}$ code blocks of $\mathcal{Q}_{r_l}$ (rectangles with dotted red border) in a grid pattern. The distance-3 code $\mathcal{Q}_{r_l}$ can correct one error per code block, and this pattern is designed so that we can recover the encoded level-l register even if all qubits in one of the level-$(l-1)$ registers suffer from correlated errors.

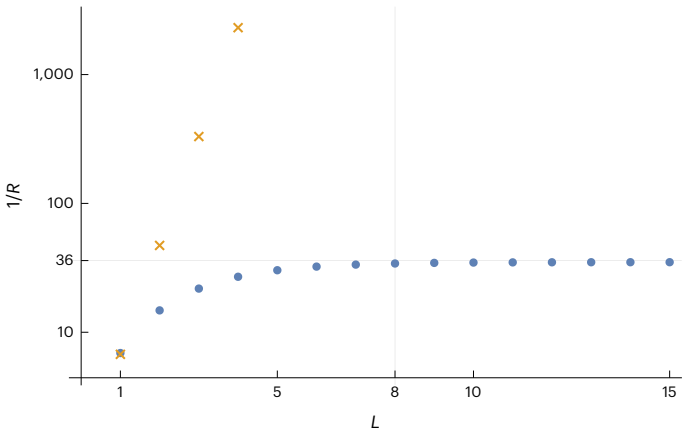

Fig. 2 | A comparison of the space overheads between our code $\mathcal{Q}^{(L)}$ and the concatenated seven-qubit code. The overhead factor, that is, the inverse of the rate $R(L)$, of the concatenated code $\mathcal{Q}^{(L)}$ developed here (blue circles) and that of the $[[7^L, 1, 3^L]]$ concatenated seven-qubit code (orange crosses) at concatenation level L. The overhead factor of $\mathcal{Q}^{(L)}$ converges to a constant $\eta_\infty < 36$ (horizontal line), saturated around $L = 8$ (vertical line), while that of the concatenated seven-qubit code diverges to infinity exponentially in L.

results hold for concatenation of quantum Hamming codes with parameter $r_l \approx \log(l)$, which would be beneficial for reducing time overhead compared with $r_l := l + 2$ while a constant factor in the space overhead may become larger (see Supplementary Section F for details). Even more generally, concatenation of a growing sequence of, for example, $[[2^{r_l} - 1, 2^{r_l} - 1 - 2tr_l, 2t + 1]]$ quantum Bose–Chaudhuri–Hocquenghem codes ($t \geqq 1$)[48], that is, CSS codes of classical Bose–Chaudhuri–Hocquenghem codes over their dual codes, can also provide a family of concatenated codes with non-vanishing rate using the same code parameter r_l as ours, which can have larger distance than quantum Hamming codes and thus have a potential advantage in faster error suppression as L increases. However, apart from constructing the code, our crucial contribution is the explicit construction of the overall fault-tolerant protocol for $\mathcal{Q}^{(L)}$ and the analysis of the threshold and the runtime, as shown in the following. The general protocol construction for other concatenated codes with a non-vanishing rate is left for future work, but our results provide a fundamental design principle for such protocols.

Efficient decoder

Remarkably, $\mathcal{Q}^{(L)}$ has an efficient decoder even though $\mathcal{Q}^{(L)}$ is not a quantum LDPC code. Efficient decoders are vital for the feasibility of FTQC yet challenging to construct in general. Indeed, no candidate among quantum LDPC codes for constant-space-overhead FTQC had such an efficient decoder until later research constructed a sufficiently efficient decoder for the quantum expander code[17,26,27].

In our protocol, for each level l, $\mathcal{Q}_{r_l}$ has an efficient decoder. In particular, the efficient decoder for the classical Hamming code $\mathcal{C}_{r_l}$ (ref. 46) can be used to correct one bit-flip error and one phase-flip error from syndromes of Z and X stabilizer generators, respectively. The decoder for $\mathcal{Q}^{(L)}$ runs efficiently by recursively using the decoder for $\mathcal{Q}_{r_l}$ as in the conventional hard-decision decoder for concatenated codes. With parallel classical computation, we can run this decoder in $O(\log(N_{r_l}))$ time at each level l, which will turn out to be polylogarithmic in the problem size M and hence small compared with the $O(\mathrm{poly}(M))$ depth of the original circuit. Remarkably, our threshold analysis shows that, even if the decoder has this non-zero runtime, our fault-tolerant protocol provably has a threshold.

Fault-tolerant protocol

Using this concatenated code $\mathcal{Q}^{(L)}$ with the non-vanishing rate, we construct a fault-tolerant protocol. To solve a family of computational problems with inputs represented by an M-bit string $(i_1, \ldots, i_M) \in \{0,1\}^M$, we use a $W(M)$-qubit $D(M)$-depth original circuit, where the width and the depth are polynomially bounded, that is, $W(M) = O(\mathrm{poly}(M))$ and $D(M) = O(\mathrm{poly}(M))$. To input $(i_1, \ldots, i_M)$ into the original circuit, we use an M-qubit initial state $\otimes_{m=1}^{M} |i_m\rangle = (\otimes_{m=1}^{M} X^{i_m}) |0\rangle^{\otimes M}$, and the rest of the original circuit is determined by M independently of the input. The original circuit is written in terms of a gate set of Clifford gates X, Y, Z, H, S, CNOT and CZ, and non-Clifford gates $R_y(\pm\pi/4)$ (see Methods for details of the notations on gates), starting from $(\otimes_{m=1}^{M} |i_m\rangle) \otimes |0\rangle^{\otimes(W(M)-M)}$ and ending with measurements of all qubits in Z basis. Our task is to simulate the original circuit by sampling from its output probability distribution within a given error ϵ in

https://doi.org/10.1038/s41567-023-02325-8

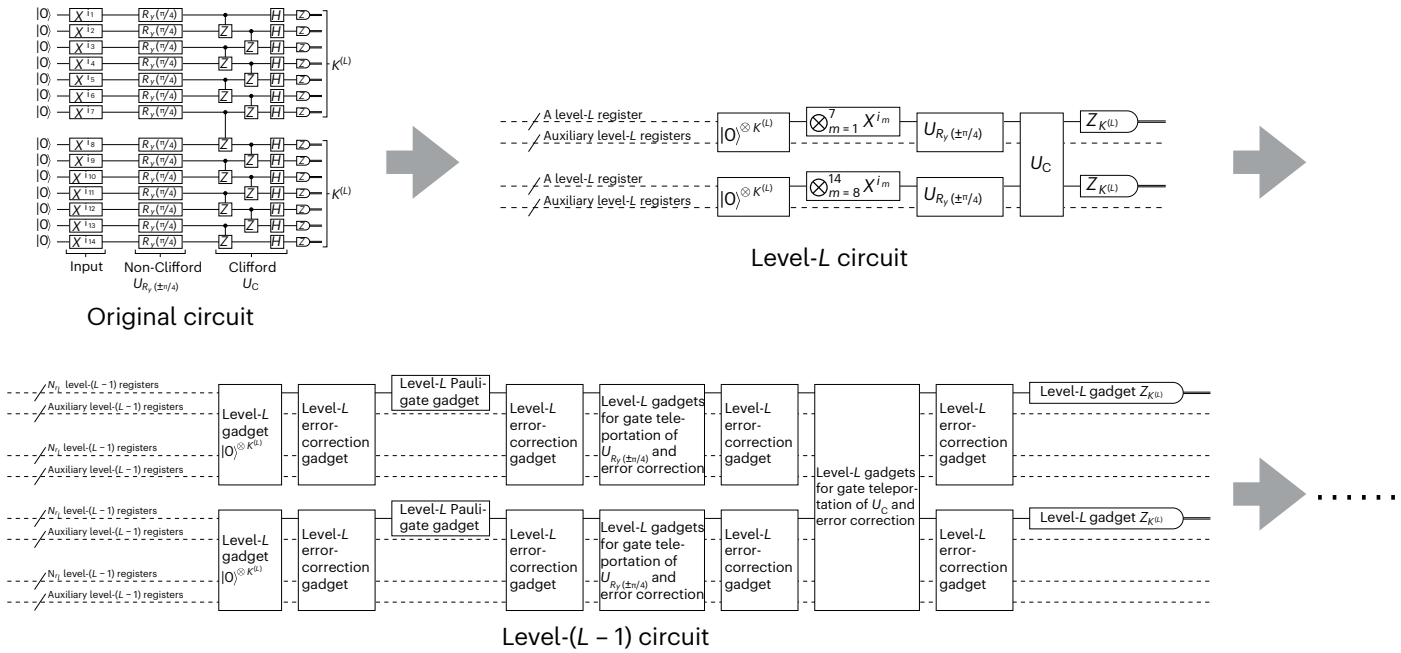

Fig. 3 | Compilation in our fault-tolerant protocol. We compile the original circuit (top left) into a level-L circuit composed of level-L elementary operations acting on level-L registers (top right). Two-register Clifford gates and one-register $R_y(\pm\pi/4)$ gates are implemented by a combination of elementary operations via gate teleportation, which are abbreviated as white boxes indicating U_C and $U_{R_y(\pm\pi/4)}$. For each $l = L, L-1, \ldots, 1$, we further compile the level-l circuit into the corresponding level-$(l-1)$ circuit by substituting each level-l elementary operation into the corresponding level-l gadget and inserting level-l error-correction gadgets in between (bottom). Unlike conventional fault-tolerant protocols with concatenated codes, the number of level-$(l-1)$ elementary operations in level-l gadgets may depend on l. We design our protocol in such a way that the level-0 circuit on physical qubits achieves the constant space overhead and the quasi-polylogarithmic time overhead compared with the original circuit.

the total variation distance. On the basis of the original circuit, our protocol recursively defines a level-l circuit ($l \in \{L, L-1, \ldots, 0\}$) composed of level-$l$ elementary operations acting on level-l registers, where a level-0 circuit is a circuit on physical qubits. The set of level-l elementary operations consists of a measurement operation, H-, CNOT-, CZ-, and Pauli-gate operations, initial-, Clifford- and magic-state preparation operations, and a wait operation (Methods). By combining these elementary operations, our protocol performs Clifford unitaries on two registers and non-Clifford $R_y(\pm\pi/4)$ gates on each register via gate teleportation.

Figure 3 illustrates the recursive construction of the circuits. We first compile the original circuit into a level-L circuit, where the required concatenation level L is determined depending on M and ϵ. For each $l = L, L-1, \ldots, 1$, we further compile the level-l circuit into the corresponding level-$(l-1)$ circuit using level-l gadgets, that is, level-$(l-1)$ circuits to implement level-l elementary operations. The construction here is different from the conventional protocol with concatenated codes[1] in that the circuits here are composed of elementary operations acting on registers rather than qubits, and that the procedure for converting a level-l circuit to a level-$(l-1)$ circuit depends on l. Unlike the conventional protocol, errors in the multiple qubits belonging to the same register may be highly correlated in our construction, and thus, we use the register as a unit in place of the qubit. The gadgets used in this compilation are designed to satisfy appropriate conditions for fault tolerance. To keep the overall space overhead constant, we design each level-l gadget to use only a constant number of additional auxiliary level-$(l-1)$ registers per encoded level-l register. See Methods for details of our construction.

Provable existence of a threshold

The novelty of our protocol is to use the concatenated code $\mathcal{Q}^{(L)}$ constructed from a sequence of different codes $\mathcal{Q}_{r_1}, \ldots, \mathcal{Q}_{r_L}$. However, the growth of code size in this sequence may make the existence of a threshold non-trivial. Conventional proofs of the threshold theorem for concatenated codes assume concatenation of the same code[1]. In contrast, a level-l register for $\mathcal{Q}^{(l)}$ is encoded into a growing number N_{r_l} of level-$(l-1)$ registers. We nevertheless prove that a threshold exists even if the number of level-$(l-1)$ elementary operations per level-l gadget grows polynomially $O(\mathrm{poly}(N_{r_l}))$, which is the case in our gadgets. Remarkably, our analysis deals with the setting where the runtime of classical computation in the decoder may also grow. The threshold analysis of the existing constant-space-overhead protocol has assumed that the decoder must run in zero time at arbitrarily large scales[17], and whether constant-space-overhead FTQC is possible with such growth of the non-zero runtime of classical computation has been unknown. After all, large-scale FTQC needs a large-size quantum LDPC code for error suppression, but if we wait for a growing runtime of the decoder for the large-size quantum LDPC code, physical qubits suffer from more errors during waiting. This situation violates the essential assumption for the existence of a threshold, that is, having a constant physical error rate between performing the error corrections. As a result, the protocols for the quantum LDPC codes may fail to correct a general class of errors on large scales (see Supplementary Section E for details). Note that the problem of requiring the non-growing runtime of classical computation persists even in the conventional protocols using quantum LDPC codes such as the two-dimensional (2D) surface code, where decoders must process the stream of syndrome data at the rate it is received, even if the code distance grows[50–54]. By contrast, our protocol based on the concatenated code $\mathcal{Q}^{(L)}$ establishes how we can perform constant-space-overhead FTQC even with finite computational resources.

In our setting, we assume that the circuit on physical qubits undergoes a conventional local stochastic error model, where adversarial and correlated errors may occur at faulty locations of operations in the level-0 circuit at a physical error rate $p_0 > 0$, but the probability of s locations simultaneously having the errors is bounded by p_0^s

(refs. 1,17). We call each level-l elementary operation in a level-l circuit a level-l location. Then, our analysis proves that faults at level-L locations in the level-L circuit of our protocol also occur according to the local stochastic error model. Moreover, we prove the following proposition on error suppression, which scales doubly exponentially in L in the same way as the conventional concatenated codes[1] (see Methods for details):

Proposition 1: error suppression with concatenation of quantum Hamming codes. Under the local stochastic error model, we have an explicit construction of a fault-tolerant protocol using the concatenated code $\mathcal{Q}^{(L)}$ with a non-zero threshold constant $p_{\text{th}} > 0$ such that, if the physical error rate is below the threshold $p_0 < p_{\text{th}}$, then our protocol using the concatenated quantum Hamming code $\mathcal{Q}^{(L)}$ can suppress the logical error rate as $p_L = \exp(-O(2^L))$.

A practical threshold is achievable with minor modifications to our protocol. For example, the same threshold as the surface code is achievable by encoding each level-0 qubit of $\mathcal{Q}^{(L)}$ for our protocol into the logical qubit of a constant-size surface code at a logical error rate below the threshold constant p_{th} here, which can take the advantage of the surface code in tolerating biased noise on physical qubits[55–58]. Note that, even if the noise on physical qubits is biased, the logical error is not necessarily biased, and thus we do not have to modify our protocol for biased noise in this case. Compared with conventional protocols, the concatenation of the surface code and our code $\mathcal{Q}^{(L)}$ has merits due to the constant space overhead, potential speedup of decoders on large scales and provable existence of a threshold even with non-zero-time decoders.

Space and time overheads

The significance of our protocol is to simultaneously achieve the constant space overhead and the quasi-polylogarithmic time overhead compared with the original circuit, as shown in the following proposition (see Methods for details):

Proposition 2: overhead achieved by concatenation of quantum Hamming codes. Under the local stochastic error model, we have an explicit construction of a fault-tolerant protocol using the concatenated code $\mathcal{Q}^{(L)}$ with a concatenation level $L = \Theta(\log(\log(M/\epsilon)))$ to simulate any $W(M)$-qubit $D(M)$-depth original circuit within error $\epsilon > 0$ in total variational distance using at most $W(M) \times O(1)$ physical qubits and $D(M) \times \exp(O(\log^2(\log(M/\epsilon))))$ runtime in terms of the depth of the fault-tolerant circuit, where $W(M) = O(\text{poly}(M))$ and $D(M) = O(\text{poly}(M))$.

These overheads include those for preparing the auxiliary states required for the gate teleportation and the error correction. Unlike the previous analysis of the existing constant-space-overhead protocol[17,26,27], the runtime also includes wait operations to wait for non-zero-time classical computations such as ones for the decoder and the gate teleportation.

As long as one uses the existing techniques for the constant-space-overhead protocol[17,26,27], it remains challenging to achieve these space and time overheads simultaneously. Indeed, the existing protocol[17,26,27] relies on conventional concatenated codes in preparing the auxiliary states for gate teleportation and hence cannot achieve the parallel gate implementation on all logical qubits within constant space overhead. Then, sequential gate implementation incurs a polynomially large time overhead in implementing parallel gates of the original circuit. By contrast, our protocol is designed to attain complete parallelizability in the gate teleportation to apply the gates to all logical qubits of $\mathcal{Q}^{(L)}$ at a time. All the auxiliary states required for the gate teleportation can be prepared in parallel within the constant space overhead owing to the non-vanishing rate of $\mathcal{Q}^{(L)}$. Consequently, our protocol is advantageous in terms of the time overhead compared with the existing constant-space-overhead protocol[17,26,27].

Tolerance for architectural overheads

Remarkably, our analysis also shows that a threshold would exist even with any polynomially growing architectural time overhead $O(\text{poly}(N^{(l)}))$ in the code size at each concatenation level l, which may be imposed by restrictions such as nearest-neighbour interactions on 2D geometry, limited classical computational resources for the decoder and insufficient parallelization in preparing auxiliary states used for gate teleportation and error correction. The suppression of the logical error rate is much faster than the growth of the code size, and thus, the concatenated codes can tolerate the time overhead at higher concatenation levels by just waiting by performing identity gates, as in our protocol. This unique property of the concatenated code contrasts with the fact that quantum LDPC codes have to be decoded at least once in a constant time to avoid the accumulation of errors. Thus, our protocol is expected to be implementable on various architectures with minor adaptation.

For example, one can rewrite fault-tolerant protocols with concatenated codes into those respecting the 2D (or even one-dimensional) geometry by well-established procedures in refs. 9,59–61, and our analysis shows that a threshold exists even with a polynomial time overhead in such rewriting. Note that, when the original circuit includes two-qubit gates on arbitrary pairs of qubits, any protocol on such 2D architectures to simulate the original circuit unavoidably incurs a polynomially long time overhead (see Supplementary Section A for details). Moreover, the constant space overhead would not be achievable on a single fully 2D chip[62]. Hence, it is essential to investigate architectures with multiple 2D layers, such as that in ref. 63, or with full connectivity, such as photonics. We leave the investigation of practical architectures to implement our protocol for future research.

We also leave exact evaluation of the threshold p_{th} for future research. This will require numerical simulation taking the architectural overhead into account, as the analytical bound is not usually tight. In such numerical simulation, comparison with architectural proposals for the constant-space-overhead protocol with quantum LDPC codes[63] may also be an interesting direction. However, importantly, the above merit of our protocol in tolerating architectural constraints such as non-constant-time decoders is not known to hold for the existing constant-space-overhead protocol with the quantum LDPC codes. We expect that such numerical simulation will also be able to clarify the effects of the architectural constraints that appear in practice.

Conclusion and outlook

We have constructed a protocol for FTQC achieving constant space overhead and quasi-polylogarithmic time overhead simultaneously. A crucial technical development is to use a concatenated code constructed from a growing sequence of quantum Hamming codes. Our technique leads to a non-vanishing rate, the existence of an efficient decoder, the space-saving and fast protocol for simulating universal quantum computation and the provable existence of a threshold for doubly exponential error suppression as we increase the concatenation level. Progressing beyond previous studies of the existing constant-space-overhead protocol based on quantum LDPC codes[17,26,27], we take into account non-zero runtime of classical computation in proving these results. Our results are fundamental for realizing FTQC feasibly within constant space overhead and yet short time overhead with parallelization. Remarkably, this achievement is made possible with the technique of code concatenation, which opens a promising route for low-overhead FTQC.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41567-023-02325-8.

References

1. Gottesman, D. An introduction to quantum error correction and fault-tolerant quantum computation. In *Proc. Symposia in Applied Mathematics* vol. 68 (ed. Lomonaco Jr., S. J.) 13–58 (American Mathematical Society, 2010).
2. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge Univ. Press, Cambridge, 2010).
3. Steane, A. M. Error correcting codes in quantum theory. *Phys. Rev. Lett.* **77**, 793–797 (1996).
4. Steane, A. Multiple-particle interference and quantum error correction. *Proc. R. Soc. Lond. A* **452**, 2551–2577 (1996).
5. Kitaev, A. Y. Quantum computations: algorithms and error correction. *Russ. Math. Surv.* **52**, 1191–1249 (1997).
6. Kitaev, A. Fault-tolerant quantum computation by anyons. *Ann. Phys.* **303**, 2–30 (2003).
7. Bravyi, S. B. & Kitaev, A. Y. Quantum codes on a lattice with boundary. Preprint at https://arxiv.org/abs/quant-ph/9811052 (1998).
8. Aharonov, D. & Ben-Or, M. Fault-tolerant quantum computation with constant error. In *Proc. 29th Annual ACM Symposium on Theory of Computing* 176–188 (Association for Computing Machinery, 1997).
9. Aharonov, D. & Ben-Or, M. Fault-tolerant quantum computation with constant error rate. *SIAM J. Comput.* **38**, 1207–1282 (2008).
10. Shor, P. W. Fault-tolerant quantum computation. In *Proc. 37th Annual Symposium on Foundations of Computer Science* (ed. Sipple, R. S.) 56 (IEEE Computer Society, 1996).
11. Knill, E., Laflamme, R. & Zurek, W. H. Resilient quantum computation: error models and thresholds. *Proc. R. Soc. Lond. A* **454**, 365–384 (1998).
12. Aliferis, P., Gottesman, D. & Preskill, J. Quantum accuracy threshold for concatenated distance-3 codes. *Quantum Inf. Comput.* **6**, 97–165 (2006).
13. Reichardt, B. W. Fault-tolerance threshold for a distance-three quantum code. In *Proc. 33rd International Conference on Automata, Languages and Programming* (eds Bugliesi, M., et al.) 50–61 (Springer-Verlag, 2006).
14. Terhal, B. M. & Burkard, G. Fault-tolerant quantum computation for local non-markovian noise. *Phys. Rev. A* **71**, 012336 (2005).
15. Dennis, E., Kitaev, A., Landahl, A. & Preskill, J. Topological quantum memory. *J. Math. Phys.* **43**, 4452–4505 (2002).
16. Fowler, A. G. Proof of finite surface code threshold for matching. *Phys. Rev. Lett.* **109**, 180502 (2012).
17. Gottesman, D. Fault-tolerant quantum computation with constant overhead. *Quantum Inf. Comput.* **14**, 1338–1372 (2014).
18. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
19. Sanders, Y. R. et al. Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum* **1**, 020312 (2020).
20. Gidney, C. & Ekerå, M. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum* **5**, 433 (2021).
21. Gottesman, D. Theory of fault-tolerant quantum computation. *Phys. Rev. A* **57**, 127–137 (1998).
22. Knill, E. Quantum computing with realistically noisy devices. *Nature* **434**, 39–44 (2005).
23. Knill, E. Scalable quantum computing in the presence of large detected-error rates. *Phys. Rev. A* **71**, 042322 (2005).
24. Gottesman, D. & Chuang, I. L. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature* **402**, 390–393 (1999).
25. Zhou, X., Leung, D. W. & Chuang, I. L. Methodology for quantum logic gate construction. *Phys. Rev. A* **62**, 052316 (2000).
26. Kovalev, A. A. & Pryadko, L. P. Fault tolerance of quantum low-density parity check codes with sublinear distance scaling. *Phys. Rev. A* **87**, 020304 (2013).
27. Fawzi, O., Grospellier, A. & Leverrier, A. Constant overhead quantum fault tolerance with quantum expander codes. *Commun. ACM* **64**, 106–114 (2020).
28. Tillich, J. & Zémor, G. Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Trans. Inf. Theory* **60**, 1193–1202 (2014).
29. Kovalev, A. A. & Pryadko, L. P. Improved quantum hypergraph-product ldpc codes. In *Proc. 2012 IEEE International Symposium on Information Theory Proceedings* (eds) 348–352 (IEEE, 2012).
30. Fawzi, O., Grospellier, A. & Leverrier, A. Efficient decoding of random errors for quantum expander codes. In *Proc. 50th Annual ACM SIGACT Symposium on Theory of Computing* (eds Diakonikolas, I. et al.) 521–534 (Association for Computing Machinery, 2018).
31. Leverrier, A., Tillich, J.-P. & Zemor, G. Quantum expander codes. In *Proc. 2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (ed. O'Conner, L.) 810–824 (IEEE Computer Society, 2015).
32. Lai, C.-Y., Zheng, Y.-C. & Brun, T. A. Fault-tolerant preparation of stabilizer states for quantum Calderbank–Shor–Steane codes by classical error-correcting codes. *Phys. Rev. A* **95**, 032339 (2017).
33. Zheng, Y.-C., Lai, C.-Y. & Brun, T. A. Efficient preparation of large-block-code ancilla states for fault-tolerant quantum computation. *Phys. Rev. A* **97**, 032331 (2018).
34. Zheng, Y.-C., Lai, C.-Y., Brun, T. A. & Kwek, L.-C. Constant depth fault-tolerant clifford circuits for multi-qubit large block codes. *Quantum Sci. Technol.* **5**, 045007 (2020).
35. Krishna, A. & Poulin, D. Fault-tolerant gates on hypergraph product codes. *Phys. Rev. X* **11**, 011023 (2021).
36. Cohen, L. Z., Kim, I. H., Bartlett, S. D. & Brown, B. J. Low-overhead fault-tolerant quantum computing using long-range connectivity. *Sci. Adv.* **8**, eabn1717 (2022).
37. Freedman, M. H., Meyer, D. A. & Luo, F. Z2-systolic freedom and quantum codes. In *Mathematics of Quantum Computation* (eds Brylinski,R. K., & Chen, G.) (Chapman & Hall/CRC, 2002).
38. Lavasani, A., Zhu, G. & Barkeshli, M. Universal logical gates with constant overhead: instantaneous Dehn twists for hyperbolic quantum codes. *Quantum* **3**, 180 (2019).
39. Breuckmann, N. P. & Burton, S. Fold-transversal clifford gates for quantum codes. Preprint at https://arxiv.org/abs/2202.06647 (2022).
40. Hastings, M. B. Decoding in hyperbolic spaces: quantum LDPC codes with linear rate and efficient error correction. *Quantum Inf. Comput.* **14**, 1187–1202 (2014).
41. Panteleev, P. & Kalachev, G. Asymptotically good quantum and locally testable classical ldpc codes. In *Proc. 54th Annual ACM SIGACT Symposium on Theory of Computin* (eds Leonardi, S. & Gupta, A.) 375–388 (Association for Computing Machinery, 2022).
42. Leverrier, A. & Zemor, G. Quantum Tanner codes. In *Proc. 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science* (ed. O'Conner, L.) 872–883 (IEEE Computer Society, 2022).
43. Dinur, I., Hsieh, M.-H., Lin, T.-C. & Vidick, T. Good quantum ldpc codes with linear time decoders. In *Proc. 55th Annual ACM SIGACT Symposium on Theory of Computing* (eds Saha, B. & Servedio, R. A.) 905–918 (Association for Computing Machinery, 2023).
44. Ueno, Y., Kondo, M., Tanaka, M., Suzuki, Y. & Tabuchi, Y. Qecool: on-line quantum error correction with a superconducting decoder for surface code. In *Proc. 2021 58th ACM/IEEE Design Automation Conference* 451–456 (IEEE, 2021).

45. Bourassa, J. E. et al. Blueprint for a scalable photonic fault-tolerant quantum computer. *Quantum* **5**, 392 (2021).
46. Hamming, R. W. Error detecting and error correcting codes. *Bell Syst. Tech. J.* **29**, 147–160 (1950).
47. Lint, J. H. V. *Introduction to Coding Theory*, 3rd edn (Springer-Verlag, 1998).
48. Steane, A. M. Simple quantum error-correcting codes. *Phys. Rev. A* **54**, 4741–4751 (1996).
49. Calderbank, A. R. & Shor, P. W. Good quantum error-correcting codes exist. *Phys. Rev. A* **54**, 1098–1105 (1996).
50. Holmes, A. et al. Nisq+: boosting quantum computing power by approximating quantum error correction. In *Proc. ACM/IEEE 47th Annual International Symposium on Computer Architecture* (ed. O'Conner, L.) 556–569 (IEEE, 2020).
51. Chamberland, C., Goncalves, L., Sivarajah, P., Peterson, E. & Grimberg, S. Techniques for combining fast local decoders with global decoders under circuit-level noise. *Quantum Sci. Technol.* **8**, 045011 (2022).
52. Skoric, L., Browne, D. E., Barnes, K. M., Gillespie, N. I. & Campbell, E. T. Parallel window decoding enables scalable fault tolerant quantum computation. *Nat. Commun.* **14**, 7040 (2022).
53. Tan, X., Zhang, F., Chao, R., Shi, Y. & Chen, J. Scalable surface code decoders with parallelization in time. Preprint at https://arxiv.org/abs/2209.09219 (2022).
54. Bombín, H. et al. Modular decoding: parallelizable real-time decoding for quantum computers. Preprint at https://arxiv.org/abs/2303.04846 (2023).
55. Tuckett, D. K., Bartlett, S. D. & Flammia, S. T. Ultrahigh error threshold for surface codes with biased noise. *Phys. Rev. Lett.* **120**, 050505 (2018).
56. Tuckett, D. K. et al. Tailoring surface codes for highly biased noise. *Phys. Rev. X* **9**, 041031 (2019).
57. Tuckett, D. K., Bartlett, S. D., Flammia, S. T. & Brown, B. J. Fault-tolerant thresholds for the surface code in excess of 5% under biased noise. *Phys. Rev. Lett.* **124**, 130501 (2020).
58. Ataides, J. P. B., Tuckett, D. K., Bartlett, S. D., Flammia, S. T. & Brown, B. J. The XZZX surface code. *Nat. Commun.* **12**, 2172 (2021).
59. Gottesman, D. Fault-tolerant quantum computation with local gates. *J. Mod. Opt.* **47**, 333–345 (2000).
60. Svore, K. M., Terhal, B. M. & DiVincenzo, D. P. Local fault-tolerant quantum computation. *Phys. Rev. A* **72**, 022317 (2005).
61. Svore, K. M., DiVincenzo, D. P. & Terhal, B. M. Noise threshold for a fault-tolerant two-dimensional lattice architecture. *Quantum Inf. Comput.* **7**, 297–318 (2007).
62. Baspin, N., Fawzi, O. & Shayeghi, A. A lower bound on the overhead of quantum error correction in low dimensions. In *Proc. 13th Innovations in Theoretical Computer Science Conference*, Vol. 215 (ed. Braverman, M.) 68:1–68:20 (Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022).
63. Tremblay, M. A., Delfosse, N. & Beverland, M. E. Constant-overhead quantum error correction with thin planar connectivity. *Phys. Rev. Lett.* **129**, 050504 (2022).

Methods

We first summarize our notation and then present the construction of our fault-tolerant protocol, the derivation of the existence of a threshold and the analysis of the space and time overheads. See Supplementary Sections C–F for further details.

Notation

For a qubit $\mathbb{C}^2$, the Z basis is denoted by $\{|0\rangle, |1\rangle\}$ and the X basis by $\{|\pm\rangle := (1/\sqrt{2})(|0\rangle \pm |1\rangle)\}$. Matrix elements are represented in terms of the Z basis. By convention of ref. 2, we use the following notation:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{1}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2}$$

$$Y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix} \propto XZ, \tag{3}$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{4}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{i} \end{pmatrix}, \tag{5}$$

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{6}$$

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \tag{7}$$

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}. \tag{8}$$

The identity operator is denoted by

$$\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{9}$$

In the same way as referring to a running time $\exp(O(\log^c(M)))$ for fixed $c > 0$ as a quasi-polynomial time in M, we call

$$\exp\left(O(\log^c(\log(M)))\right) \tag{10}$$

a quasi-polylogarithmic time. A quasi-polylogarithmic time with $c > 1$ may be larger than a polylogarithmic time $\exp(O(\log(\log(M)))) = O(\mathrm{polylog}(M))$. However, a quasi-polylogarithmic time for any $c > 0$ is much smaller than a polynomial time, that is, $\exp(O(\log(M))) = O(\mathrm{poly}(M)) = O(M^\alpha)$, even for an arbitrarily small degree $\alpha > 0$ of the polynomial.

Construction of fault-tolerant protocol

In our fault-tolerant protocol, we use level-l elementary operations to write a level-l circuit for each $l \in \{L, L-1, \ldots, 0\}$. The set of level-$l$ elementary operations consists of a level-l measurement operation, level-l H-, CNOT-, CZ-, and Pauli-gate operations, level-l initial-, Clifford- and magic-state preparation operations, and a level-l wait operation. The measurement operation implements measurements in the Z basis of all qubits in a level-l register. The H-, CNOT-, CZ- and Pauli-gate operations implement H, CNOT and CZ gates on all qubits in level-l registers and the tensor product of any combination of Pauli gates on the qubits in a level-l register. The initial-state preparation operation prepares a level-l register in $|0\rangle^{\otimes K^{(l)}}$. To assist implementing any given two-register Clifford unitary U_{C} on the $2K^{(l)}$ qubits in two level-l registers, the Clifford-state preparation operation prepares four level-l registers B_1, B_2, B_3 and B_4 in

$$\left(\mathbb{1}^{B_1B_2} \otimes U_{\mathrm{C}}^{B_3B_4}\right) |\Phi^{(l)}\rangle^{B_1B_2B_3B_4}, \tag{11}$$

where $|\Phi^{(l)}\rangle^{B_1B_2B_3B_4} = |\Phi\rangle^{B_1B_3} \otimes |\Phi\rangle^{B_2B_4}$, and $|\Phi\rangle^{B_jB_{j'}} \propto \sum_{m=0}^{2^{K^{(l)}}-1} |m\rangle^{B_j} \otimes |m\rangle^{B_{j'}}$ for $(j,j') \in \{(1,3),(2,4)\}$ is a maximally entangled state between B_j and $B_{j'}$. To assist implementing any given unitary $U_{R_y(\pm\pi/4)}$ in the form of a tensor product of $R_y(\pi/4)$, $R_y(-\pi/4)$ and $\mathbb{1}$ on the $K^{(l)}$ qubits in a level-l register, the magic-state preparation operation prepares two level-l registers in

$$(R_y(\pi/4)|0\rangle)^{\otimes K^{(l)}} \otimes (R_y(\pi/4)|0\rangle)^{\otimes K^{(l)}}. \tag{12}$$

A wait operation is a Pauli-gate operation of $\mathbb{1}^{\otimes K^{(l)}}$.

Using these operations in combination, we implement U_{C} and $U_{R_y(\pm\pi/4)}$ for a level-l circuit by gate teleportation. Note that H-, CNOT- and CZ-gate operations in our protocol perform the gates on all qubits in the registers simultaneously, and we use U_{C} for implementing all the other Clifford gates, for example, a single-qubit H gate in one of the two registers (while acting as the identity gate on the other register), and a CNOT gate on a specific pair of qubits in the two registers. Indeed, U_{C} is not limited to a one- or two-qubit Clifford gate but may represent an arbitrary sequence of Clifford gates acting on the $2K^{(l)}$ qubits in two level-l registers as a single Clifford unitary. Similarly, $U_{R_y(\pm\pi/4)}$ is not necessarily a single-qubit non-Clifford gate but can apply non-Clifford gates on any number of qubits in a level-l register. In particular, a level-l two-register Clifford gate U_{C} in our protocol is implemented by means of gate teleportation[22–24], assisted by the auxiliary state $(\mathbb{1}^{B_1B_2} \otimes U_{\mathrm{C}}^{B_3B_4})|\Phi^{(l)}\rangle^{B_1B_2B_3B_4}$ prepared by the level-l Clifford-state preparation operation, along with other level-l gate and measurement operations. The correction of byproducts in the gate teleportation is performed by level-l Pauli-gate operations. Regarding level-l $U_{R_y(\pm\pi/4)}$, the gate teleportation for $U_{R_y(\pm\pi/4)}$ is assisted by an auxiliary magic state $(R_y(\pi/4)|0\rangle)^{\otimes K^{(l)}}$ prepared by the level-l magic-state preparation operation, and also an auxiliary state $|0\rangle^{\otimes K^{(l)}}$ prepared by the level-l initial-state preparation operation, along with other level-l gate and measurement operations. To apply $R_y(\pm\pi/4)$ in $U_{R_y(\pm\pi/4)}$ to a desired subset of qubits in a register, we prepare the required auxiliary state in the tensor product of $R_y(\pi/4)|0\rangle$ and $|0\rangle$ by applying single-qubit SWAP gates between $(R_y(\pi/4)|0\rangle)^{\otimes K^{(l)}}$ and $|0\rangle^{\otimes K^{(l)}}$ using the level-l two-register Clifford gate. Then, assisted by this auxiliary state, we perform the gate teleportation. The correction of byproducts, which are single-qubit Clifford gates on a level-l register, is performed by the level-l two-register Clifford gate acting trivially on another auxiliary level-l register.

To simulate a level-l circuit at each level $l \in \{L, \ldots, 1\}$, we construct a level-l gadget corresponding to each level-l elementary operation, that is, a level-$(l-1)$ circuit for simulating the elementary operation on encoded level-l registers. Apart from these level-l gadgets, we use a level-l error-correction gadget, a level-$(l-1)$ circuit for correcting errors on one of the N_{r_l} level-$(l-1)$ registers for an encoded level-l register. For the existence of a threshold, each level-l gadget must be fault tolerant. That is, roughly speaking, even if one of the level-$(l-1)$ locations in the gadget has a fault, the resulting error should be

correctable using the decoder of $\mathcal{Q}_{r_l}$ at the end of the gadget (see Supplementary Section D for the precise definition). This definition of fault-tolerant gadgets in our protocol is a suitable modification of the conventional definition for the concatenated codes[1] so that we can prove the existence of a threshold by applying the conventional argument in ref. 1 to our protocol. Using the fault-tolerant level-l gadgets, we convert a level-l circuit into the corresponding level-$(l-1)$ circuit by replacing each level-l elementary operation with the corresponding level-l gadget, followed by inserting the level-l error correction gadgets between all pairs of adjacent level-l elementary operations. Repeating this conversion recursively for $l \in \{L, \ldots, 1\}$ yields a level-0 circuit, which leads to a fault-tolerant circuit on physical qubits to simulate the original circuit.

In the following, we sketch our construction of level-l gadgets used for the fault-tolerant protocol. See Supplementary Section D for further details. Note that gate implementations for some classes of CSS codes with multiple logical qubits have also been discussed in refs. 17,64, but the main contribution of our work is to present the gadgets explicitly for our code $\mathcal{Q}^{(l)}$ so that we can prove the existence of a threshold and bound the space and time overheads rigorously for our fault-tolerant protocol.

We implement the level-l measurement gadget by performing level-$(l-1)$ measurement operations for all the level-$(l-1)$ registers and then calculating bit values of the outcome by a decoder, using the logical Z operator for each of the $K^{(l)}$ logical qubits in the encoded level-l register. We let $Z_{K^{(l)}}$ label this Z-basis measurement with the $K^{(l)}$-bit outcome. The fault tolerance follows from transversality.

We implement the H-, CNOT- and CZ-gate gadgets by applying level-$(l-1)$ H-, CNOT- and CZ-gate operations, respectively, to all level-$(l-1)$ registers transversally. We implement the Pauli-gate gadget by level-$(l-1)$ Pauli-gate operations to apply the tensor product of Pauli gates representing the logical Pauli operators to the level-$(l-1)$ registers transversally. The wait gadget is a special case of the Pauli-gate gadget to apply the identity gate. The fault tolerance follows from transversality.

The level-l initial-state preparation gadget is implemented by transforming states $|0\rangle^{\otimes K^{(l-1)}}$ prepared by the level-$(l-1)$ initial-state preparation operations into logical $|0\rangle^{\otimes K^{(l)}}$ by a level-$(l-1)$ stabilizer circuit in a non-fault-tolerant way[65–67], followed by verification with post-selection. For the verification, we prepare another logical $|0\rangle^{\otimes K^{(l)}}$, and using this auxiliary $|0\rangle^{\otimes K^{(l)}}$, we measure the logical Z operators and the Z stabilizer generators of $\mathcal{Q}_{r_l}$. If no logical X error is detected from this measurement on the logical $|0\rangle^{\otimes K^{(l)}}$ prepared in this first run, that is, in the case of success in the verification, then the gadget outputs this state. Otherwise, the gadget discards the prepared state and repeats the same level-$(l-1)$ stabilizer circuit to output the logical $|0\rangle^{\otimes K^{(l)}}$ prepared in this second run without verification. This repetition makes the gadget fault tolerant while at most doubling the depth of the gadget.

Assisted by the registers in logical states $|0\rangle^{\otimes K^{(l)}}$ prepared by the level-l initial-state preparation gadgets, the level-l error correction gadget is implemented here in a fault-tolerant way by Knill's error correction[22,23] based on quantum teleportation[68]. The fault tolerance follows from transversality. Unlike the quantum LDPC codes using an auxiliary physical qubit per extracting each syndrome bit, the weight of stabilizer generators does not matter for the feasibility of error correction with the concatenated codes. In particular, using the above technique for the concatenated codes, we can prepare encoded codewords $|0\rangle^{\otimes K^{(l)}}$ in a fault-tolerant way, and using this fault-tolerant state preparation to perform Knill's error correction, we can obtain all the syndrome bits simultaneously from the measurement outcomes for quantum teleportation, without using the auxiliary physical qubit per syndrome.

Note that we could also use Steane's error correction here[1], but Knill's error correction may have merits in our protocol since Knill's error correction can be implemented in the same way as the gate teleportation used for implementing the level-l two-register Clifford gates. An additional advantage of Knill's error correction over Steane's error correction is its ability to correct leakage errors[23], while the error model in our analysis does not explicitly consider the leakage errors for simplicity.

The level-l Clifford-state preparation gadget is implemented by non-fault-tolerant state preparation followed by verification, similar to the level-l initial-state preparation gadget. In particular, we first transform logical states $|0\rangle^{\otimes 4K^{(l)}}$ prepared by the level-l initial-state preparation gadgets into logical $(\mathbb{1} \otimes U_C)\,|\Phi^{(l)}\rangle$ by a level-$(l-1)$ stabilizer circuit in a non-fault-tolerant way[65]. Then, we perform verification with post-selection. In the verification, we let the state be in the code space of $\mathcal{Q}_{r_l}$ using the level-l error-correction gadgets. Then, as $(\mathbb{1} \otimes U_C)\,|\Phi^{(l)}\rangle$ is a stabilizer state, we measure the logical stabilizer operators for $(\mathbb{1} \otimes U_C)\,|\Phi^{(l)}\rangle$ (that is, multi-qubit Pauli operators) using the controlled Pauli gates implemented by the gate teleportation. To make the gadget fault tolerant, we design the gadget in such a way that an error on the auxiliary registers used as the control qubits for these controlled Pauli gates should not propagate to $(\mathbb{1} \otimes U_C)\,|\Phi^{(l)}\rangle$ conditioned on the post-selection, using the technique of flag qubits[69]. Since we concatenate the distance-3 quantum Hamming codes, the verification can be made fault tolerant by adding one flag qubit per extraction of logical stabilizer operators as in ref. 69. Note that flag qubit techniques in refs. 70,71 may also be used for potential generalization to concatenating higher-distance codes. If no error is detected from measuring the logical stabilizer operators and the flag qubits, that is, in the case of success in the verification, then the gadget outputs the logical $(\mathbb{1} \otimes U_C)\,|\Phi^{(l)}\rangle$ prepared in this first run. Otherwise, the gadget discards the prepared state and repeats the same level-$(l-1)$ stabilizer circuit to output the logical $(\mathbb{1} \otimes U_C)\,|\Phi^{(l)}\rangle$ prepared in this second run without verification. In the same way as the initial-state preparation gadget, the repetition at most doubles the depth of the gadget.

The level-l magic-state preparation gadget is also implemented by non-fault-tolerant state preparation followed by verification. First, we prepare states $(R_y(\pi/4)\,|0\rangle)^{\otimes 2K^{(l)}}$ and $|0\rangle^{\otimes 2(N^{(l)}-K^{(l)})}$ by the level-$(l-1)$ magic- and initial-state preparation operations, respectively, and transform them into logical $(R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}} \otimes (R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}}$ by a level-$(l-1)$ stabilizer circuit for encoding, that is, for transforming the magic states into the same logical states in a non-fault-tolerant way[65]. Then, we perform the verification with post-selection by ensuring the state in the code space of $\mathcal{Q}_{r_l}$ and measuring the logical stabilizer operators. This magic-state preparation does not use magic state distillation[72,73] but instead uses verification to reduce errors. In particular, since $(R_y(\pi/4)\,|0\rangle)$ is stabilized by H, that is, $H(R_y(\pi/4)\,|0\rangle) = R_y(\pi/4)\,|0\rangle$, we implement controlled H gates for measuring the logical stabilizer operators for $(R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}} \otimes (R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}}$, using techniques similar to state-of-the-art low-overhead magic-state preparation protocols in refs. 74–76. To make the gadget fault tolerant, similar to the level-l Clifford-state preparation gadget, the level-l magic-state preparation gadget is also designed in such a way that an error on the auxiliary registers used as the control qubits for the controlled H gates should not propagate to $(R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}} \otimes (R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}}$ conditioned on the post-selection, using the technique of flag qubits[69]. If no error is detected from measuring the logical stabilizer operators and the flag qubits, that is, in the case of success in the verification, then the gadget outputs the logical $(R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}} \otimes (R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}}$ prepared in this first run. Otherwise, the gadget discards the prepared state and repeats the same level-$(l-1)$ circuit to output the logical $(R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}} \otimes (R_y(\pi/4)\,|0\rangle)^{\otimes K^{(l)}}$ prepared in this second run without verification. In the same way as the initial- and Clifford-state preparation gadgets, the repetition at most doubles the depth of the gadget.

With a synthesis of stabilizer circuits[77,78], we show that all the level-l gadgets here have at most $O(\mathrm{poly}(N_{r_l}))$ depths, including the wait operations to wait for classical computation. Consequently, each level-l gadget has at most $O(\mathrm{poly}(N_{r_l}))$ locations, even if we take into account wait operations to wait for non-zero-time classical computation such as ones in the decoder and the gate teleportation.

Analysis of threshold existence and improvement

We sketch the proof of the existence of a threshold in our fault-tolerant protocol and discuss how to achieve a practical threshold with minor protocol modifications. See Supplementary Section E for further details.

As in the conventional proof for the concatenated code, the proof of the existence of a threshold in our protocol is given by bounding a logical error rate at a higher concatenation level by that at a lower level, based on counting the number of locations in extended rectangles (ExRecs)[1] (see also the figure illustrating ExRecs in Supplementary Section E). Given a level-l circuit for $l \in \{1, \ldots, L\}$, a level-l ExRec refers to a part of the corresponding level-$(l-1)$ circuit that includes a level-l gadget at each level-l location and its adjacent level-l error-correction gadgets[1]. For the distance-3 code such as the code used here, a level-l ExRec is said to be good if the ExRec contains at most one faulty level-$(l-1)$ location, and bad otherwise. Intuitively, a good ExRec can implement the logical operation correctly, but a bad ExRec may not. Thus, to bound the logical error rate, it suffices to evaluate the probability of having a bad ExRec using the number of locations therein.

In particular, let $A(l)$ be the maximum number of pairs of level-$(l-1)$ locations in a level-l ExRec, where we take the maximum over all the possible level-l ExRecs. Since all the level-l gadgets used in our protocol have $O(\mathrm{poly}(N_{r_l}))$ level-$(l-1)$ locations, we have $A(l) = O(\mathrm{poly}(N_{r_l})) = \exp(O(l))$. For simplicity of presentation, let $\alpha > 0$ denote a constant factor such that

$$A(l) \leqq 2^{\alpha l} \tag{13}$$

for all $l \geqq 1$. Crucially, our definition of a gadget being fault tolerant is made analogous to the conventional definition in ref. 1, so that the same argument as in ref. 1 is applicable to our protocol. When a level-$(l-1)$ circuit simulates a level-l circuit, this argument leads to the fact that, if the level-$(l-1)$ circuit undergoes the local statistic error model, the level-l circuit also does. Then, let p_0 be the physical error rate of level-0 locations, and p_l denote the logical error rate of level-l locations at each level l. The conventional argument for the threshold theorem proves that the logical error rates are upper bounded by the probability of having two errors in an ExRec, that is, $p_l \leqq A(l)p_{l-1}^2$ for each l (ref. 1), which leads to $p_l \leqq 2^{\alpha l}p_{l-1}^2$. Using this bound recursively, we can prove that the logical error rate p_L is bounded by $p_L \leqq (2^{2\alpha}p_0)^{2^L}/2^{2\alpha}$, as shown in Supplementary Section E. This shows the existence of a threshold $p_{\mathrm{th}} \geqq 2^{-2\alpha} > 0$ such that the logical error rate $p_L \leqq (p_0/p_{\mathrm{th}})^{2^L}p_{\mathrm{th}}$ can be suppressed doubly exponentially in L if the physical error rate satisfies $p_0 < p_{\mathrm{th}}$. Note that the same argument as ours for the threshold existence holds even in cases where the exponent αl of $2^{\alpha l}$ in equation (13) is replaced with $\mathrm{poly}(l)$. For example, even if the gadgets had $O(\mathrm{poly}(N^{(l)}))$ depths due to architectural overhead or insufficient parallelization, a threshold would still exist.

Remarkably, a practical threshold is also achievable with minor modifications to our protocol. Any quantum code with one logical qubit can be concatenated with $\mathcal{Q}^{(L)}$ by using the logical qubit of the code in place of each level-0 qubit of $\mathcal{Q}^{(L)}$, as long as the code can implement required operations for our fault-tolerant protocol at level 0, namely preparation of a qubit in $|0\rangle$, a single-qubit measurement in the Z basis and the H, S, CNOT, CZ, Pauli and $R_y(\pm\pi/4)$ gates. For example, we can concatenate the surface code and $\mathcal{Q}^{(L)}$ and replace physical operations for our fault-tolerant protocol at concatenation level 0 with the corresponding logical operations on the surface code. Indeed, the surface code has well-established procedures for implementing logical operations for universal quantum computation[79,80]. Thus, we can use the logical qubit of a constant-size surface code in place of each physical qubit of $\mathcal{Q}^{(L)}$ in our protocol. With this modification, we can achieve the same threshold as that of the surface code, and at the same time attain the constant overhead asymptotically. See also Supplementary Section E regarding further options for protocol modifications for a better threshold.

We also remark that the above lower bound $2^{-2\alpha}$ of the non-zero threshold value p_{th}, derived here for the rigorous proof of its existence is not necessarily close to p_{th}. Thus, it would be misleading to use $2^{-2\alpha}$ as an estimate of p_{th}. To estimate p_{th}, one needs to perform a more precise numerical simulation, which is essential for finding out which part of the protocol is a bottleneck to be modified for further improvement. In addition to the threshold value itself, the achievable logical error rate at a finite concatenation level may also be of interest. After all, what matters to FTQC in practice is the overall balance of the protocol, depending on the specific settings of the error model and the architectural constraint. We leave the numerical simulation of the protocols based on concatenating quantum Hamming codes for future work, but our theoretical contribution is fundamental for research towards such a practical direction.

Analysis of space and time overhead

We sketch the analysis of space and time overheads of our fault-tolerant protocol. See Supplementary Section F for further details.

To achieve the constant space overhead, our protocol uses the code $\mathcal{Q}^{(L)}$ with a non-vanishing rate of logical qubits per physical qubit. However, it is still non-trivial to achieve the constant space overhead since the protocol may additionally use auxiliary level-$(l-1)$ registers in level-l gadgets for implementability. Crucially, we design each level-l gadget to use only a constant number of auxiliary level-$(l-1)$ registers per encoded level-l register, so as to keep the overall space overhead constant

$$O(1) \quad \text{as } M \to \infty, \tag{14}$$

including physical qubits used for the auxiliary registers.

To save time overhead, it is essential to realize gates acting on all the level-L registers in parallel. At the same time, it is also crucial to keep the code size for sufficient error suppression as small as possible. After all, a smaller code size leads to a faster preparation of auxiliary states for gate teleportation and thus a smaller time overhead in implementing each gate acting on the level-L registers. As our threshold analysis shows, the suppression of the logical error rate $p_L = \exp(-O(2^L))$ in our protocol is exponentially faster than the growth of the code size $N^{(L)} = \exp(O(L^2))$ of $\mathcal{Q}^{(L)}$. By choosing $L = \Theta(\log\log(M/\epsilon))$, we can reduce the overall error in simulating the original circuit to ϵ. With this choice, the size of each code block $\mathcal{Q}^{(L)}$ becomes only quasi-polylogarithmic $N^{(L)} = \exp(O(L^2)) = \exp(O(\log^2(\log(M/\epsilon))))$. On the other hand, each gadget in our protocol is designed to be implementable within at most polynomial time in the code size. Therefore, this code size leads to the quasi-polylogarithmically small time overhead

$$\exp\left(O(\log^2(\log(M/\epsilon)))\right) \tag{15}$$

in implementing the gates and thus in simulating the original circuit. This time overhead includes that for preparing auxiliary states for gate teleportation and error correction, and also that for waiting for the non-zero-time classical computation during the protocol, such as the ones required for the decoder and the gate teleportation.

Data availability

The data used in this study are available from the corresponding author upon reasonable request.

Code availability

The codes used in this study are available from the corresponding author upon reasonable request.

References

64. Steane, A. M. & Ibinson, B. Fault-tolerant logical gate networks for calderbank-shor-steane codes. *Phys. Rev. A* **72**, 052335 (2005).
65. Cleve, R. & Gottesman, D. Efficient computations of encodings for quantum error correction. *Phys. Rev. A* **56**, 76–82 (1997).
66. Paetznick, A. & Reichardt, B. W. Fault-tolerant ancilla preparation and noise threshold lower boudds for the 23-qubit golay code. *Quantum Inf. Comput.* **12**, 1034–1080 (2012).
67. Steane, A. M. Fast fault-tolerant filtering of quantum codewords. Preprint at https://arxiv.org/abs/quant-ph/0202036 (2002).
68. Bennett, C. H. et al. Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels. *Phys. Rev. Lett.* **70**, 1895–1899 (1993).
69. Chao, R. & Reichardt, B. W. Quantum error correction with only two extra qubits. *Phys. Rev. Lett.* **121**, 050502 (2018).
70. Chamberland, C. & Beverland, M. E. Flag fault-tolerant error correction with arbitrary distance codes. *Quantum* **2**, 53 (2018).
71. Chao, R. & Reichardt, B. W. Flag fault-tolerant error correction for any stabilizer code. *PRX Quantum* **1**, 010302 (2020).
72. Bravyi, S. & Kitaev, A. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A* **71**, 022316 (2005).
73. Knill, E. Fault-tolerant postselected quantum computation: schemes. Preprint at https://arxiv.org/abs/quant-ph/0402171 (2004).
74. Yamasaki, H., Fukui, K., Takeuchi, Y., Tani, S. & Koashi, M. Polylog-overhead highly fault-tolerant measurement-based quantum computation: all-Gaussian implementation with Gottesman–Kitaev–Preskill code. Preprint at https://arxiv.org/abs/2006.05416 (2020).
75. Goto, H. Minimizing resource overheads for fault-tolerant preparation of encoded states of the steane code. *Sci. Rep.* **6**, 19578 (2016).
76. Chamberland, C. & Cross, A. W. Fault-tolerant magic state preparation with flag qubits. *Quantum* **3**, 143 (2019).
77. Aaronson, S. & Gottesman, D. Improved simulation of stabilizer circuits. *Phys. Rev. A* **70**, 052328 (2004).
78. Patel, K. N., Markov, I. L. & Hayes, J. P. Optimal synthesis of linear reversible circuits. *Quantum Inf. Comput.* **8**, 282–294 (2008).
79. Horsman, C., Fowler, A. G., Devitt, S. & Meter, R. V. Surface code quantum computing by lattice surgery. *N. J. Phys.* **14**, 123011 (2012).
80. Litinski, D. A game of surface codes: large-scale quantum computing with lattice surgery. *Quantum* **3**, 128 (2019).

Acknowledgements

H.Y. acknowledges K. Fujii and Y. Suzuki for comments in the meeting of JST PRESTO. This work was supported by JSPS Overseas Research Fellowships, JST PRESTO grant no. JPMJPR201A and JST (Moonshot R&D, grant no. JPMJMS2061).

Author contributions

H.Y. and M.K. contributed to the conception of the work, the analysis and interpretation in the work, and the preparation and revision of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41567-023-02325-8.

Correspondence and requests for materials should be addressed to Hayata Yamasaki.

Peer review information *Nature Physics* thanks Nicolas Delfosse and the other, anonymous, reviewers for their contribution to the peer review of this work.