Hierarchical surface code for network quantum computing with modules of arbitrary size

Ying Li and Simon C. Benjamin*
Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, United Kingdom
(Received 28 December 2015; published 4 October 2016)

The network paradigm for quantum computing involves interconnecting many modules to form a scalable machine. Typically it is assumed that the links between modules are prone to noise while operations within modules have a significantly higher fidelity. To optimize fault tolerance in such architectures we introduce a hierarchical generalization of the surface code: a small "patch" of the code exists within each module and constitutes a single effective qubit of the logic-level surface code. Errors primarily occur in a two-dimensional subspace, i.e., patch perimeters extruded over time, and the resulting noise threshold for intermodule links can exceed ∼10% even in the absence of purification. Increasing the number of qubits within each module decreases the number of qubits necessary for encoding a logical qubit. But this advantage is relatively modest, and broadly speaking, a "fine-grained" network of small modules containing only about eight qubits is competitive in *total* qubit count versus a "course" network with modules containing many hundreds of qubits.

DOI: 10.1103/PhysRevA.94.042303

I. INTRODUCTION

There are two approaches to fabricating a large-scale universal quantum computer. One is to create a single "monolithic" architecture in which each qubit is directly and deterministically connected to its neighbors. An alternative is the network architecture [1–8], where a single quantum computer is formed from numerous interlinked small devices, *modules*, each having only a modest number of qubits and correspondingly little computational power. This approach may prove to be well suited to ion trap systems [8–11] or color centers in diamond [12], where optical activity can be directly harnessed to create a photonic link; modules comprised of superconducting qubits can also be networked either via microwave links [13] or by exploiting microwave-to-optical converters. It is likely that the size of a module, i.e., the number of physical qubits within it, may vary dramatically according to the technology: whereas a color center might have at most a dozen or so satellite nuclear spins, a superconducting module could easily be envisaged as a grid of hundreds of qubits. It is therefore interesting to ask what impact the module size has on performance characteristics such as the fault tolerance threshold and, thus, the total number of physical qubits needed per logical qubit.

An advantage of the network architecture is its manifest scalability. However, based on experimental results to date it is reasonable to assume that intermodule communications will only provide low-quality entanglement [12,14,15] compared with intramodule quantum gates [9,10,16,17]. Whatever approach one adopts to mitigate the noise on the links, there will inevitably be a resource cost versus an idealized monolithic architecture where all gates are of comparable fidelity to the intramodule operations. In other words, to implement the same quantum algorithm, more qubits are required in the network architecture to overcome network noise. A goal of this paper is to quantify this difference.

II. OUTLINE OF APPROACH

We investigate quantum computing with a network architecture involving modules containing from only two qubits to about a thousand qubits. In our study, we exploit two methods to negate errors: entanglement purification [1,4] and error correction via the surface code [18,19]. Entanglement purification is a low-level process that corrects errors on intermodule links and is carried out individually within each module with the help of classical communications. We use the term *broker unit* for the dedicated hardware (comprising one or more qubits) associated with entanglement purification.

For small modules with only a few qubits in total, each module only provides one qubit participating in the surface code, while the rest are involved in purification. This is equivalent to architectures that have been studied in earlier papers [20,21]. We include such small-module architectures in this paper for context; the challenge we tackle here is to efficiently exploit large modules with at least tens of qubits. Our solution retains the purification but additionally introduces a hierarchical variant of the surface code. A piece of surface code (or "patch") exists in every module, such that each module can be effectively regarded as a single qubit in a higher (logical-level) surface code. There are interesting consequences for the localization and correction of errors, given that such errors tend to occur at the boundaries between the modular patches. In essence the errors live in a two-dimensional space, one spatial and one temporal dimension, so that the relevant threshold is *equivalent* to that of a two-spatial-dimension system with perfect noise-free stabilizer measurement [19]. Thus the hierarchical surface code tolerates network errors up to 15%, and purification techniques need only bring the network noise within this limit.

One might expect that the performance of a network computer will approach that of a monolithic computer as the module size increases. To compare the resource requirements we study the number of qubits required for encoding a single

*simon.benjamin@materials.ox.ac.uk

logical qubit. We find that the total number of physical qubits required per logical qubit does indeed decrease with the size of modules. For a practical level of network noise and modules containing hundreds of qubits, the cost of encoding a logical qubit in the network architecture is still about nine times higher than the cost in the monolithic architecture. Meanwhile we find that for a "fine-grained" network comprised of small modules containing only about eight qubits, the overhead versus the monolithic system is a factor of about 15. It is perhaps surprising that the resource cost associated with adopting the flexible network paradigm varies so little over a wide range of module sizes, i.e., the network granularity does not strongly affect the total resource cost.

III. SYSTEM

We consider a quantum computer built with networked quantum modules as shown in Fig. 1. We focus on the case where each module contains an array of *client* qubits and a series of entanglement-purifying *broker units* on the perimeter of the client array; each broker unit contains several qubits as we presently discuss [see Fig. 1(a)]. Quantum information is stored in client qubits, and brokers are used to generate entanglement between neighboring modules. In each broker unit, there must be at least one qubit that is optically coupled with another module. Raw entanglement prepared with optical coupling is purified with the help of other qubits in broker units; the qubits forming the surface code "patch" therefore never "see" the raw entanglement, only the purified form. To provide a context for assessing the performance of the hierarchical surface code, we also consider the limit of small modules where each module may contain only one client qubit and one broker unit [see Fig. 1(b)], in which case the sole broker unit services links to all connected modules by rerouting the optical connection as required. These small modules do not use a hierarchical surface code, instead relying on a single surface code layer.

Intramodule controlled NOT (CNOT) gates are performed via interactions between qubits within the module. We assume that these CNOT gates are available for any pair of nearest-neighboring qubits in the same module. For client qubits in different modules, *distributed* CNOT gates are performed by consuming entanglement that has been generated between brokers. The circuit for the distributed CNOT gate [22] is shown in Fig. 6(a).

With the geometry of modules in Fig. 1, client qubits on the perimeters of neighboring modules are indirectly coupled through brokers. Therefore, ultimately a square lattice is formed by all client qubits in the network, in which (intramodule or distributed) CNOT gates are available for any

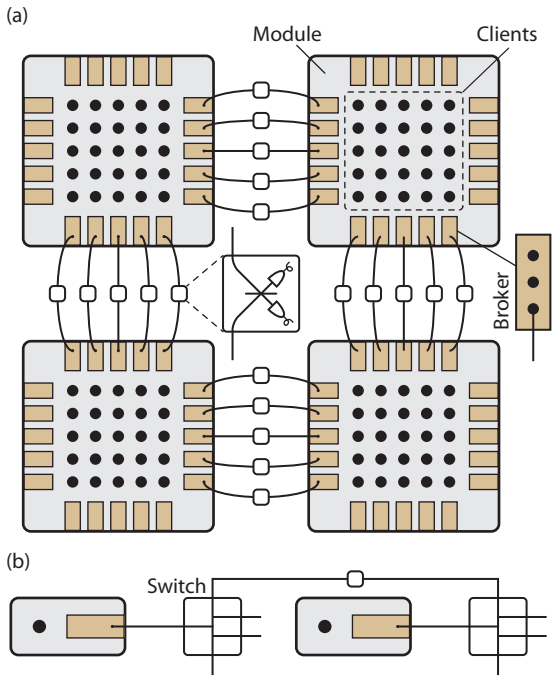

FIG. 1. A quantum computer built with optically networked quantum modules. Black circles represent qubits. (a) Each module contains a $D \times D$ client-qubit array as well as $4 \times D$ broker units on the perimeter, where D is the dimension of the module. The modules have $D = 5$. Broker units achieve entanglement with one another via, e.g., intermodule photonic couplings. Typically the raw entanglement will be generated by a joint measurement on photons emitted from optically active broker qubits. Inside a broker unit, there are additional ancilla qubits which are used to purify raw entanglement to a higher fidelity. (b) Each simple module contains only one client qubit and one broker unit. Each such module is coupled with four neighboring modules via a switch for rerouting the optical connection.

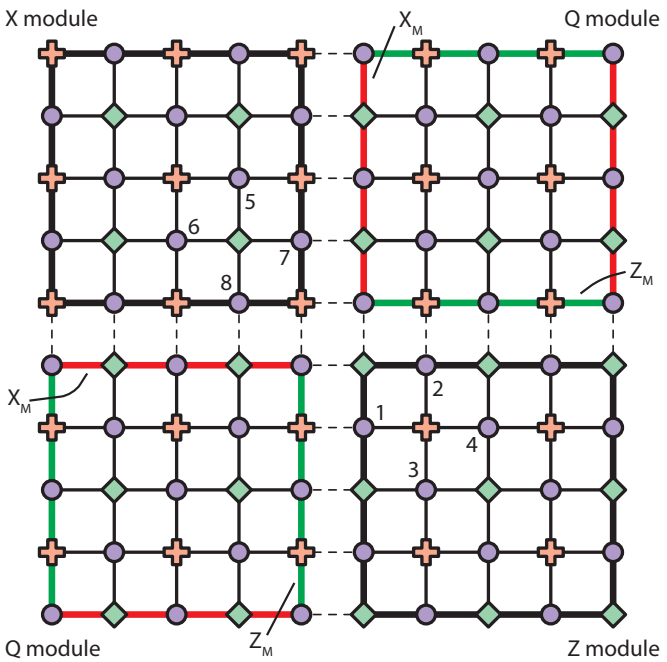

FIG. 2. Square lattice of qubits for implementing the surface code, showing the layout of the lowest level, physical qubits. Data qubits, X ancillary qubits, and Z ancillary qubits are represented by circles, crosses, and squares, respectively. We assume that a CNOT gate can be performed on any pair of qubits connected by an edge. Four modules are depicted, each with $D = 5$: solid edges correspond to intramodule gates; dashed edges, to gates implemented with intermodule entanglement, i.e., distributed CNOT gates. Errors arising from imperfectly purified remote entanglement will only directly affect qubits on the perimeter of a module; these are shown with thick edges. X_M and Z_M denote Pauli operators of module qubits.

pair of nearest-neighboring qubits. With such a lattice, we can implement the surface code across the entire module network.

Within the surface code lattice (Fig. 2), physical qubits are divided into three groups: data qubits (circles) and measurement-enabling qubits of two kinds—X ancillary qubits (crosses) and Z ancillary qubits (squares). The subspace for encoding information in the collective is defined by enforcing sets of stabilizers $XXXX$ and $ZZZZ$, which are products of Pauli operators on four data qubits surrounding X ancillaries and Z ancillaries, respectively [18,19]. Errors are detected by repeatedly measuring stabilizers [23] with circuits shown in Figs. 6(d) and 6(e).

IV. MODULAR SURFACE CODE AND THRESHOLDS

In our modular network, errors associated with the entanglement generated over network links are first reduced by entanglement purification. After the purification, there are still some residual errors on the intermodule entanglement because of the limited resources of each broker unit. These residual entanglement errors, together with errors arising from intramodule operations, are finally corrected by the surface code. Assuming that the entanglement is ideally in the form $(|00\rangle + |11\rangle)/\sqrt{2}$, we model the error-burdened entangled state as

$$\mathcal{E} = F[\mathbb{1}] + p_X[X] + p_Z[Z] + p_Z[Z], \tag{1}$$

where $F = 1 - p_X - p_Y - p_Z$ is the fidelity, the superoperator $[U]\rho = U\rho U^\dagger$, and X, Y, Z are Pauli operators on one of two entangled qubits. For intramodule operations, we assume that a qubit may be initialized in the incorrect state with probability ϵ_I; the measurement may report an incorrect outcome with probability ϵ_M; and each single-qubit gate and controlled-NOT (CNOT) gate may induce an error with probability ϵ_1 and ϵ_2, respectively. A noisy gate is modeled as a perfect gate followed by single-qubit depolarizing noise for single-qubit gates and two-qubit depolarizing noise for the CNOT gate [24]. See Appendix C for more details of the error model.

As one might expect, we find that if we consider modules containing a larger client array, then more residual entanglement errors can be corrected with the surface code. This would be true even if we were to simply regard all the "patches" of surface code as part of a single surface without giving any special status to the borders between patches. However, in doing so we would be failing to exploit our knowledge that errors are more common along the perimeters. To properly exploit the potential advantage of large modules, instead of continuously performing stabilizer measurements on the entire surface code lattice, we introduce an intermediate encoding based on the client array of each module.

Similar to physical qubits on the surface code lattice, modules are also divided into three groups: Q modules, X modules and Z modules (see Fig. 2). The client array of each Q module itself is a piece of complete surface code lattice, hence one informational qubit can be encoded in each Q module, and we refer to this as a *module qubit*. Moreover, the X modules and Z modules have roles similar to X ancillary qubits and Z ancillary qubits in the basic surface code: they perform X-stabilizer and Z-stabilizer measurements on neighboring

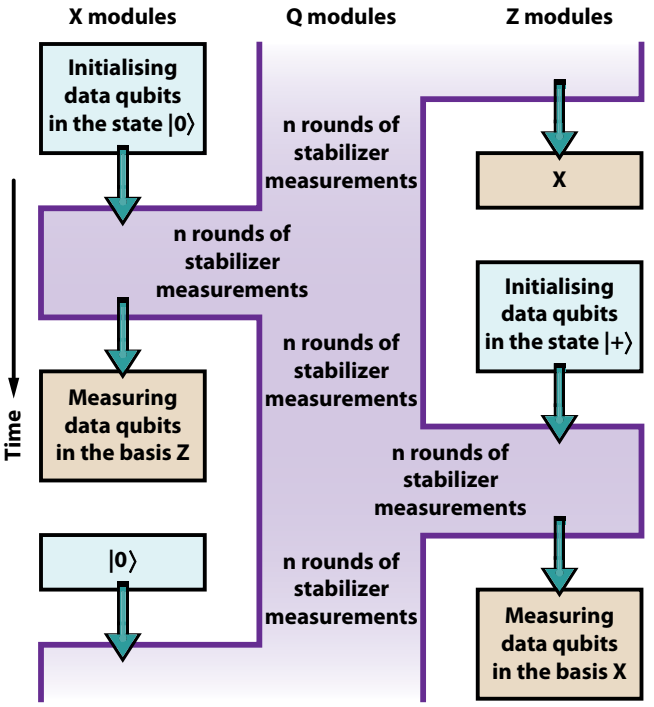

FIG. 3. Protocol for module-qubit stabilizer measurements. Each full round of module-qubit stabilizer measurements involves measuring both X-stabilizers and Z-stabilizers. To measure X-stabilizers at the module-qubit level, physical data qubits in X modules are initialized in state $|0\rangle$ and measured in the Z basis after n rounds of physical-qubit stabilizer measurements. Measuring Z-stabilizers at the module-qubit level is exactly analogous. When measurements of module-qubit X (Z) stabilizers are in progress, Z (X) modules are not involved in physical-qubit stabilizer measurements. Between each set of module-qubit stabilizer measurements, physical-qubit stabilizer measurements are performed on only Q modules for n rounds.

module qubits, respectively. Therefore, a network of modules forms a surface code on a higher level where each module qubit now constitutes a basic unit. Thus the approach is a *hierarchical* surface code: a logical qubit is realized through a small surface code, each qubit of which is an entire module; each module is itself a small surface code with an informational or measurement role as described above. One might worry that this nested approach increases the number of physical qubits required, but the important point is that the required surface code size at each of the two levels will be far smaller than that which would be required in an equivalent single, nonhierarchical code .

The protocol for performing stabilizer measurements at the module-qubit level is shown in Fig. 3. The stabilizer $XXXX$ of four module qubits equals the product of all physical-qubit stabilizers $XXXX$ within the X module. To measure the X-stabilizer of four module qubits, first, data qubits in the the corresponding X module are all initialized in the state $|0\rangle$; second, physical-qubit stabilizer measurements across the X module and measured Q modules (see Fig. 7) are performed for several rounds; and, finally, all data qubits in the X module are measured in the Z basis (see Fig. 3). The module-qubit Z-stabilizer measurement is analogous. Between each set of module-qubit stabilizer measurements, physical-qubit

stabilizer measurements are performed on only Q modules for several rounds. See Appendix B for an explanation of the protocol.

In order to perform universal quantum computing using the surface code, one must prepare (create and distill) resources called magic states [25]. It is therefore interesting to ask, Does the hierarchical surface code introduced here lead to difficulties in that process? In fact, it does not: Using stabilizer measurements, a logical magic state can be prepared in two stages: a module-qubit magic state can be prepared by initializing one data qubit in the magic states and then performing physical-qubit stabilizer measurements according to the protocol in Ref. [26]; with the same protocol, after performing module-qubit stabilizer measurements, the magic state is encoded into a logical qubit. When error rates are below error thresholds, the fidelity of logical magic states is suitable for purification [25], and one can perform such purification at the logical level using the protocols in the literature, as, for example, in Ref. [27].

Commensurate with our two-tier encoding, the error correction includes two tiers. In the first tier the outcomes of physical-qubit stabilizer measurements are used to correct errors at the physical-qubit level. As we describe below, this process can exploit the highly inhomogeneous nature of the physical errors, i.e., the high error density at the border of each module's "patch" of surface code. Once this step is complete, only sequences of errors (error chains) that span entire module qubits can survive. For Q modules, such an error is a bit or phase flip of that specific module qubit. For the X and Z modules, whose role is to provide stabilizer measurements on the surrounding four Q modules, the consequence of such an error chain is that the stabilizer outcome is incorrectly evaluated (i.e., it is the converse of the correct outcome). Both types of errors are handled in the second tier of the process, where one simply regards each Q module as a data qubit, and errors on these qubits are determined by analysis of the imperfect stabilizer measurements in the standard way (regardless of the fact that those measurements derive from entire X and Z modules). Please see Appendix D for more details.

To understand how the inhomogeneity in the distribution of errors is exploited, consider first the artificial case where intramodule operations are perfect (i.e., $\epsilon_{\mathrm{I}} = \epsilon_{\mathrm{M}} = \epsilon_1 = \epsilon_2 = 0$). Then all errors are due to imperfectly purified intermodule entanglement, and so errors occur strictly on perimeters of client arrays (thick lines in Fig. 2). During X-stabilizer measurements of module qubits, in a Q module bit errors and stabilizer measurement errors occur at the rate $p_{\mathrm{X}} + p_{\mathrm{Y}}$ on the two boundaries facing X modules (thick red lines), and in an X module phase errors and stabilizer measurement errors occur at the rate $p_{\mathrm{Z}} + p_{\mathrm{Y}}$ on the entire perimeter (thick black lines). We note that in the corner of X modules, error rates are approximately doubled. It is similar for Z-stabilizer measurements of module qubits. Here, p_{X}, p_{Y}, p_{Z} are error rates in the intermodule entanglement *after* any purification has taken place. Errors are restricted to the one-dimensional perimeter of modules, but the correction process involves n rounds and therefore the syndrome matching occurs in a two-dimensional space: one spatial and one temporal. This is in contrast to a standard surface code approach with homogeneous errors on gates and measurements, where we would need to match syndrome outcomes in a three-dimensional array. There is a very significant advantage in terms of the threshold: whereas the three-dimensional threshold is in the region of 3%, for the restricted two-dimensional case it is 10% [19]. Therefore, if entanglement error rates satisfy $p_{\mathrm{X}} + p_{\mathrm{Y}}, p_{\mathrm{Z}} + p_{\mathrm{Y}} < 10\%$, we are below threshold and thus the rates of module-qubit errors after the first step of error correction decrease with the dimension of the modules (i.e., the size of two-dimensional error-correction lattices). Moreover, if indeed the module-qubit error rates decrease with the module dimension, then the threshold for errors on the intermodule entanglement increases with the module dimension. In the limit of large modules, the threshold of the entanglement error rate should approach 15% for depolarizing errors, i.e., $p_{\mathrm{X}} = p_{\mathrm{Y}} = p_{\mathrm{Z}}$. If we now allow for a small but finite rate of errors for intramodule operations and measurements, the proceeding remarks all apply except that occasional errors will occur within the perimeter of the "patches," with the consequence that the tolerance of noise in the intermodule links will be somewhat reduced.

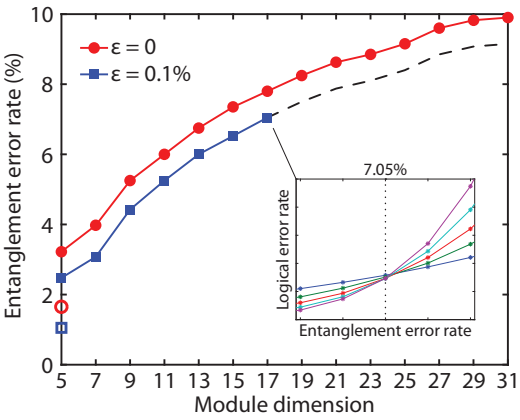

FIG. 4. Fault tolerance thresholds in terms of the rate of errors, $1 - F$, on the intermodule entanglement for modules with different dimensions D. Here F is the fidelity *subsequent* to any purification within broker units, so that we are seeing the corrective power of the hierarchical surface code alone. For context, thresholds for simple modules are shown by the open circle and square on the left vertical axis. We have assumed that entanglement errors are unpolarized, i.e., $p_{\mathrm{X}} = p_{\mathrm{Y}} = p_{\mathrm{Z}}$, and all intramodule operations have the same error rate, $\epsilon_{\mathrm{I}} = \epsilon_{\mathrm{M}} = \epsilon_1 = \epsilon_2 = \epsilon$. Please see Appendix C for details of the error model. These thresholds are obtained by counting errors on logical qubits encoded in $(2L-1) \times (2L-1)$ module arrays (see Appendix E for details). Inset: Logical error rates for $D = 17$ and $L = 3, 5, 7, 9, 11$; with an entanglement error rate lower than the threshold (dotted vertical line), the logical error rate decreases with L (left side of the threshold). The dashed line denotes inferred thresholds according to the difference (which is $\sim$0.75%) between two solid lines.

In Fig. 4 we show the results of a series of numerical simulations which verify this analysis. The figure shows the fault tolerance threshold for the rate of errors in the intermodule entanglement, assuming that (purified) entanglement errors are uniform over the X, Y, and Z channels. Note that this is not a favorable assumption: if the errors were not uniform, this would be an opportunity to enhance the threshold by

exploiting this knowledge in the decoder. The numerical results reveal that the threshold indeed increases with the module dimension, which coincides with expectations from the preceding analysis. For comparison we also find thresholds for simple modules, i.e., each module contains only one client qubit so we do not use the hierarchical approach.

The observed thresholds vary from 1.65% to 9.9%, depending on the size of modules. When we allow for errors induced by intramodule operations, the ability to correct errors on the intermodule operations is reduced as expected, i.e., the threshold rate of tolerable intermodule errors decreases with the error rate of intramodule operations. Taking all intramodule operations to have the same error rate, $\epsilon_{\mathrm{I}} = \epsilon_{\mathrm{M}} = \epsilon_1 = \epsilon_2 = 0.1\%$, we find that the threshold of entanglement error rate is reduced by 0.75%. As shown in Fig. 4, this reduction varies only very slightly with the module dimension.

V. QUBIT COSTS

In the preceding analysis, we considered the structure of the hierarchical surface code and its threshold in terms of the rate of errors on intermodule entanglement; the error rate was taken to be postpurification. Now in order to find the optimal structure for the network architecture, we must consider the power and cost of the brokering units and optimize the number of qubits assigned to that role. We can then find the overall resource cost of fault-tolerant computing given the specific error rate in the "raw" intermodule entanglement.

In a quantum computer based on the surface code, the unit of quantum computing is a logical qubit encoded in a $(2L-1)\times(2L-1)$ qubit (module) array, where the array distance L is the minimum number of data qubits (Q modules) for defining a logical Pauli operator. Given that operations are performed with an error rate lower than the system's threshold, the rate of logical errors decreases with the size of the logical qubit. The logical error rate per surface code cycle, i.e., a full round of stabilizer measurements, scales with the distance L as [28]

$$\epsilon_{\mathrm{L}} \simeq \epsilon_0 e^{-\kappa L}, \tag{2}$$

where parameters ϵ_0 and κ are determined by error rates of operations.

In our network architecture, qubits used for entanglement generation and purification do not participate in forming logical qubits, i.e., these qubits assigned to the "broker units" are an additional cost due to the modular architecture. It is nontrivial to optimize the partitioning of qubits between broker units and the internal client arrays, in such a way as to minimize the total number of qubits needed to achieve a given logical error rate. The size of logical qubits (determined by parameters ϵ_0 and κ) depends on the intermodule entanglement error rate *after* purification, and this rate will improve as more qubits are dedicated to purification.

By numerically obtaining parameters ϵ_0 and κ, we can find the cost of physical qubits per logical qubit (logical-qubit size). We have considered error rates of intramodule operations, $\epsilon_{\mathrm{I}} = \epsilon_{\mathrm{M}} = \epsilon_1 = \epsilon_2 = 0.1\%$, and two possible values for the error rate in the *raw* entanglement, $1-F = 1.5\%$ and 15%. The logical-qubit size for achieving the logical error rate $\epsilon_{\mathrm{L}} = 10^{-12}$ is shown in Fig. 5. The raw entanglement error rate $1-F = 1.5\%$ is a tenth of the theoretical threshold in the large-module limit. For this low entanglement error rate, we expect that the purification is not necessary for large modules. However, the raw entanglement error rate $1-F = 15\%$ is

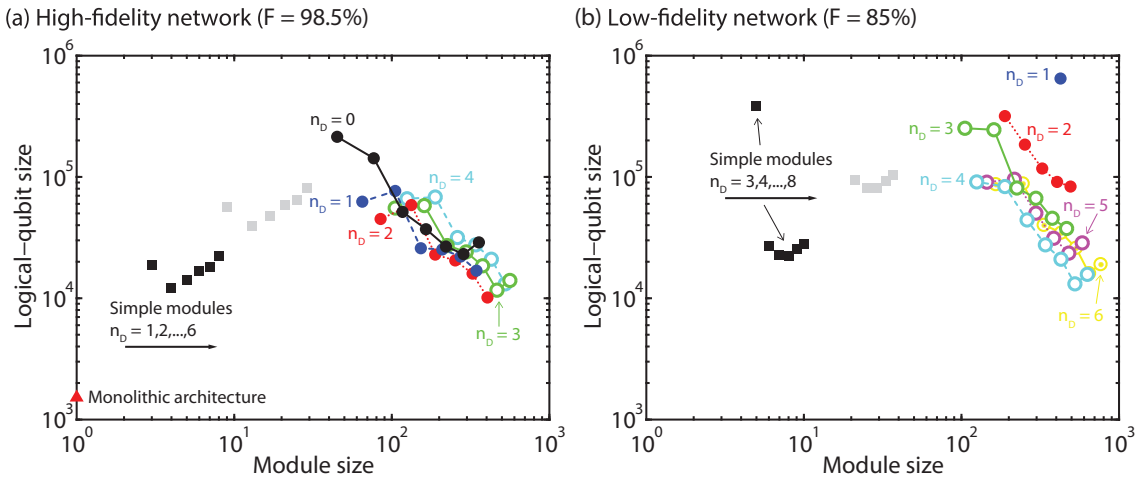

FIG. 5. The total count of physical qubits per logical qubit required to achieve the logical error rate $\epsilon_{\mathrm{L}} = 10^{-12}$. F is the fidelity of the "raw" entanglement between modules; we have assumed uniform raw entanglement errors on the form of Eq. (1), i.e., $p_{\mathrm{X}} = p_{\mathrm{Y}} = p_{\mathrm{Z}}$, and all intramodule operations have the same error rate, $\epsilon_{\mathrm{I}} = \epsilon_{\mathrm{M}} = \epsilon_1 = \epsilon_2 = 0.1\%$ (please see Appendix C for details of the error model). Circles represent qubit costs for modules with the dimension $D \geqslant 5$, so that a module qubit can be encoded in each Q module. For such modules, the module size (the total number of qubits in each module) equals $D^2 + 4D(n_{\mathrm{D}}+1)$, where each broker contains $n_{\mathrm{D}}+1$ qubits, and n_{D} is the number of entanglement purification tiers (see Appendix G). Black squares represent qubit costs for simple modules with only one broker. In each simple module, the qubit number is $n_{\mathrm{D}}+2$. Gray squares correspond to simple modules with four brokers, i.e., the module size equals $4n_{\mathrm{D}}+5$. The red triangle on the left vertical axis in (a) represents the qubit cost on the monolithic architecture [28], i.e., when noisy network links are not used and *all* gates are performed with the lower error rate of 0.1%. These qubit costs are obtained by numerically obtaining parameters ϵ_0 and κ in Eq. (2) (shown in Fig. 10; see Appendix F for details).

more practical for current technologies. For such a high error rate, we see that entanglement purification is always necessary.

In general, the qubit cost decreases with the size of modules (total number of qubits in each module). When the size of modules approaches a thousand qubits, the qubit cost is only about nine times higher than the cost given the (idealized) monolithic architecture. We note that a factor of 2 is due to the two-tier encoding considered in this paper. In this encoding, approximately half of the modules, i.e., all the X and Z modules, are ancillaries for stabilizer measurements. Without the overhead cost due to X and Z modules, and if physical-qubit stabilizer measurements are continuously performed across the whole module network, isolated two-dimensional error-correction lattices of entanglement errors merge into a single connected lattice. The error correction on the connected lattice, which is essentially three-dimensional, will be harder than the error correction on isolated lattices. However, this disadvantage may prove to be tolerable for very large modules, in which case the overhead cost due to X and Z modules may not be necessary. It would be interesting to perform an analysis of this case where modules are very large, exceeding the size required for storing logical qubits, in order to determine whether the hierarchical code introduced in this paper remains useful in that domain.

In our analysis and the data points shown in Fig. 5, we contrasted the performance of a network of substantial modules with that of a network of small modules, each containing one data qubit of a simple surface code. In terms of the total number of physical qubits needed to achieve a given low error rate at the logical level, our somewhat surprising conclusion is that simple modules are only marginally inferior to large modules containing nearly a thousand qubits. In this sense, our result is that *the granularity of a network does not strongly influence the resource costs*. However, as a caveat we must remark that in our study we have assumed that physical qubits have a long memory time, so that memory errors are negligible on the time scale required to perform entanglement purification (see Appendix G). If this is not the case, then a significant advantage for large modules could emerge because fewer purification tiers are necessary. An analysis of this scenario would open the way to a full audit of the time cost of network quantum computing, where the time needed for the multiple rounds of stabilization in the hierarchical picture is contrasted with the time needed for deep purification circuits.

VI. CONCLUSIONS

In this paper, we have introduced a variant of the surface code approach to fault-tolerant quantum information processing. Our variant is intended to support the network paradigm for quantum computing: the machine is divided into many modules which are connected by noisy interlinks, and each module contains a plurality of well-controlled (low-noise) qubits. Our approach is a two-tier hierarchical surface code, where the lower tier involves assigning a "patch" of surface code to each physical module. Errors then occur primarily at patch boundaries and this reduces the dimensionality of the syndrome matching task, thus boosting the threshold. We consider a general scenario where errors occurring in the intermodule links are first purified by broker units and then handled by the hierarchical code.

Both analytical reasoning and numerical results show that larger modules have advantages: the threshold is higher and the qubit cost is lower. However, the advantage in qubit cost is not significant enough to conclude that large modules are preferred platforms of quantum computing, taking into account the difficulty of building large modules. For small modules, we find that a size of approximately eight qubits per module is optimal for practical entanglement noise purification, and the total qubit cost per logical qubit is only 15 times higher than the cost of a monolithic architecture. Broadly our conclusion is that the granularity of a network-based quantum computer does not strongly affect the total resource costs, with the consequence that experimental efforts can target whatever module size is most convenient for the particular technology.

ACKNOWLEDGMENTS

This work was supported by the EPSRC platform grant Molecular Quantum Devices (EP/J015067/1) and the EPSRC National Quantum Technology Hub in Networked Quantum Information Processing (EP/M013243/1). The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work [29].

APPENDIX A: CIRCUITS

Circuits for performing the distributed CNOT gate, entanglement purifications, and stabilizer measurements are shown in Fig. 6.

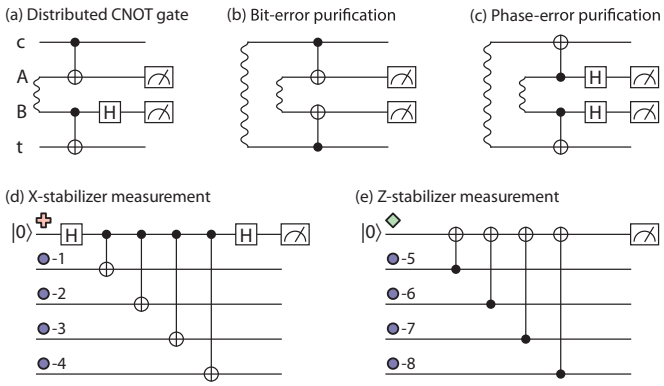

FIG. 6. Circuits for the distributed CNOT gate, entanglement purifications, and stabilizer measurements. (a) This circuit is equivalent to a CNOT gate on qubit-c (control) and qubit-t (target), up to Pauli gates Z_c and X_t, depending on the measurement outcomes. (b, c) If the ideal entangled state is in the form $(|00\rangle + |11\rangle)/\sqrt{2}$, the output entanglement is discarded if two measurement outcomes are different. In the bit-error purification, the bit-error rate is reduced from q_B for the input entanglement to $\sim q_B^2$ for the postselected output entanglement, but the phase-error rate is increased from q_P to $\sim 2q_P$. It is similar for the phase-error purification. (d, e) Each full round of stabilizer measurements involves both X-stabilizer measurements and Z-stabilizer measurements. Labels on data qubits (see Fig. 2) indicate the sequence of CNOT gates (CNOT gates with the same orientation are performed in parallel).

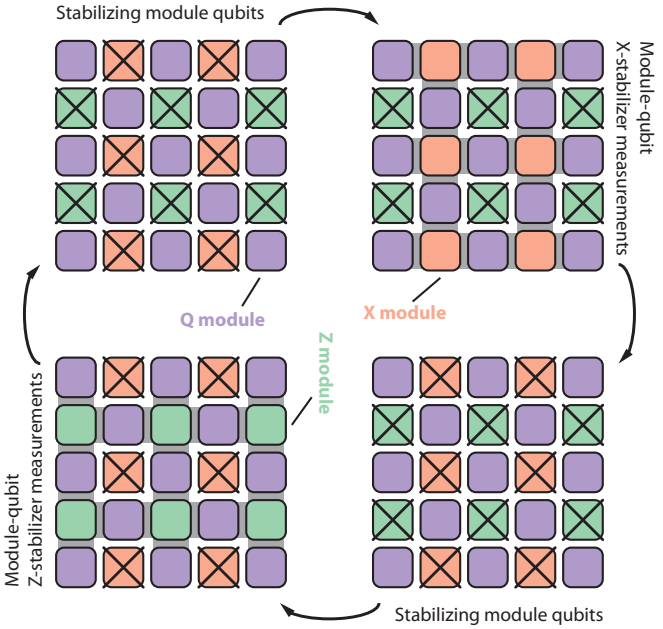

FIG. 7. Cycle of module-qubit stabilizer measurements. When each module qubit is individually stabilized, only Q modules are involved, and intermodule entanglement is not required. When module-qubit stabilizers are measured, either X modules or Z modules are used to read stabilizers of module qubits (as described in the text). This process needs intermodule entanglement for implementing distributed CNOT gates.

APPENDIX B: MODULE-QUBIT STABILIZER MEASUREMENTS

The layout of module-qubit stabilizer measurements is shown in Fig. 7.

The cycle of module-qubit stabilizer measurements shown in the figure allows us to measure module-qubit stabilizers and track physical-qubit errors at the same time. Taking module-qubit X-stabilizers as an example, the important observation is that the product of all physical-qubit X-stabilizers in the X module equals the product of X logical Pauli operators of four surrounding Q modules. Therefore, one can measure the X-stabilizer of four Q modules by measuring these physical-qubit X-stabilizers in an X module. Because data qubits in X modules are initialized in state $|0\rangle$, values of all Z-stabilizers of physical qubits should be $+1$ in the absence of errors (assuming initially that all physical-qubit stabilizers in Q modules are $+1$). Hence, bit-flip errors on data qubits can be detected. In Q modules, because X-stabilizers of data qubits are isolated from X modules, their values should also be $+1$ in the absence of errors, thus phase-flip errors on data qubits in Q modules can be detected.

The story is slightly more complex for X modules: because values of X-stabilizers are stochastic due to the initial state of surrounding data qubits, phase-flip errors on data qubits in X modules cannot be detected for the first few rounds of physical-qubit stabilizer measurements. However, these errors do not affect Q modules (module qubits). The only way that these errors can affect module qubits is that they cooperate with measurement errors of physical-qubit X-stabilizers and flip the value of a physical-qubit X-stabilizer throughout n rounds of physical-qubit stabilizer measurements (see Fig. 3). Such a chain of errors flips the outcome of the module-qubit stabilizer measurement, which is an error on the module-qubit level. When n is large, the probability of such an error chain is suppressed. Finally, data qubits in X modules are measured in the Z basis, which allows us to correct bit-flip errors for the last few rounds of physical-qubit stabilizer measurements and recover values of physical-qubit Z-stabilizers on the boundary of Q modules.

APPENDIX C: ERROR MODEL

Here we describe the error model which we have used in simulating the performance of the modular computer. Within each module, there are finite rates of error for initialization, measurements, single-qubit operations, and two-qubit operations. For initialization operations the state of a qubit is initialized in the correct state $|0\rangle$ with probability $\epsilon_{\rm I}$ and the incorrect state $|1\rangle$ with probability $1-\epsilon_{\rm I}$. For measurement operations, if the state of the qubit is $|0\rangle$ ($|1\rangle$), the measurement outcome is correct [i.e., 0 (1)] with probability $\epsilon_{\rm M}$ and incorrect [i.e. 0 (1)] with probability $1-\epsilon_{\rm M}$.

For a single-qubit gate described by the unitary operator U_1, the operation actually performed on the qubit is $\mathcal{E}_1\mathcal{U}_1$, where the superoperator $\mathcal{U}_1\rho = U_1\rho U_1^\dagger$. The superoperator $\mathcal{E}_1$ describes errors on the qubit. If errors are depolarized,

$$\mathcal{E}_1 = (1-\epsilon_1)[\mathbb{1}] + \frac{\epsilon_1}{3}([X]+[Y]+[Z]). \tag{C1}$$

The total probability of errors is ϵ_1, and three types of Pauli errors occur with the same probability.

Similarly, for a two-qubit gate described by the unitary operator U_2, the operation actually performed on two qubits (qubits A and B) is $\mathcal{E}_2\mathcal{U}_2$, where the superoperator $\mathcal{U}_2\rho = U_2\rho U_2^\dagger$. The superoperator $\mathcal{E}_2$ describes errors on two qubits. If errors are depolarized,

$$\begin{aligned}\mathcal{E}_2 = (1-\epsilon_2)[\mathbb{1}] + \frac{\epsilon_2}{15}([X_{\rm A}]+[Y_{\rm A}]+[Z_{\rm A}] \\ +[X_{\rm B}]+[Y_{\rm B}]+[Z_{\rm B}]+[X_{\rm A}X_{\rm B}]+[Y_{\rm A}X_{\rm B}]+[Z_{\rm A}X_{\rm B}] \\ +[X_{\rm A}Y_{\rm B}]+[Y_{\rm A}Y_{\rm B}]+[Z_{\rm A}Y_{\rm B}]+[X_{\rm A}Z_{\rm B}] \\ +[Y_{\rm A}Z_{\rm B}]+[Z_{\rm A}Z_{\rm B}]).\end{aligned} \tag{C2}$$

The total probability of errors is ϵ_2, and 15 types of Pauli errors occur with the same probability.

For all cases where we have considered finite errors in this paper, in particular, for the data in Figs. 4 and 5, we have set all internal error sources to be simultaneously present and equal to 0.1%, that is,

$$\epsilon_{\rm I} = \epsilon_{\rm M} = \epsilon_1 = \epsilon_2 = 0.1\%.$$

This level of fidelity has been reached or exceeded for all such operations in a certain type of ion trap [11].

Entanglement between modules is generated by jointly detecting photons emitted from two remote optically active broker qubits. The expression for the error-burdened "raw" entangled state between broker qubits is given as Eq. (1), which we reiterate here for completeness:

$$\mathcal{E} = F[\mathbb{1}] + p_{\rm X}[X] + p_{\rm Z}[Z] + p_{\rm Z}[Z].$$

TABLE I. The number N of raw entanglement pairs required to obtain one pair of n_D-tier purified entanglement with success probability P_S. The failure to prepare a purified entangled state results in missing the corresponding stabilizer measurement for one round, i.e., the stabilizer measurement is not successful in that round, which could be compensated by enlarging the logical qubit. When $P_S \sim 99.9\%$, we expect that missing a small portion of stabilizer measurements increases the resource cost only slightly. We have assumed that raw entanglement has fidelity F, entanglement errors are unpolarized, i.e., $p_X = p_Y = p_Z$, and all intramodule operations have the same error rate, $\epsilon_I = \epsilon_M = \epsilon_1 = \epsilon_2 = 0.1\%$.

$P_S = 99\%$		$P_S = 99.9\%$	
n_D	N	n_D	N
	$F = 98.5\%$		
0	1	0	1
1	4	1	4
2	8	2	12
3	16	3	20
4	32	4	40
5	64	5	80
6	132	6	158
7	268	7	320
8	544	8	646
	$F = 85\%$		
0	1	0	1
1	6	1	10
2	18	2	26
3	32	3	44
4	66	4	92
5	114	5	150
6	218	6	286
7	420	7	542
8	848	8	1064

The process of entanglement generation is probabilistic due to the intrinsic nondeterministic nature of linear-optical Bell measurements and photon loss; the state specified above is the result of a heralded success, while failures are abandoned and the generation procedure repeats. It is worth noting that during the entanglement generation process, optically active broker qubits are not entangled with any client qubits. Thus, loss leads to detected failures and the consequent repeat of the process does not directly harm the quantum information in client qubits. We assume that memory errors are negligible, i.e., the environmental decoherence time of qubits is much longer than the time required to generate entanglement. We report the time cost in terms of entanglement events in Table I.

APPENDIX D: ERROR CORRECTION

For simple modules, the error correction is directly performed on a conventional surface code error correction lattice. Even in this simple case there is scope for optimizing the weights in the decoding algorithm, and we have done so broadly as described in prior papers [30].

For large modules, we employ our hierarchical surface code, for which error correction has two steps. First, physical-qubit errors are corrected on the error correction lattice shown in Fig. 8(a), which is a three-dimensional lattice formed by cubes of size $\sim(D+1)/2 \times (D+1)/2 \times n$. In our numerical simulations, we have included error correlations (the same as simple modules) and the inhomogeneity of the error distribution in the first-step error correction lattice. The weight of an edge on the error correction lattice is given by

$$w = \ln \frac{1-p}{p}, \tag{D1}$$

where p is the rate of errors corresponding to the edge. For an edge on the boundary of a module [see Fig. 8(a)], the error rate p is much higher than for edges within modules, i.e., the weight w is lower. By taking weights according to the probability of errors, Edmonds' minimum weight matching algorithm takes account of the fact that edges connecting modules are more likely to be noisy, so that the performance of the error correction is improved compared with the case of taking homogeneous weights.

After the first step, there are only module-qubit errors left, which are further corrected on a conventional surface code error correction lattice representing the array of module qubits. In our numerical simulations, we have neglected correlations between module-qubit errors, i.e., the error correction lattice is a simple cubic lattice.

Using Edmonds' algorithm in the quantum context [31] to identify errors requires centralising stabilizer measurement outcomes in one classical information processor: in this sense it is a global calculation over the entire quantum computer. However, it is worth noting that a decoder for the standard surface code that requires only local information processing has been developed [32] and in principle there is no reason that this could not be adapted to the present hierarchical surface code.

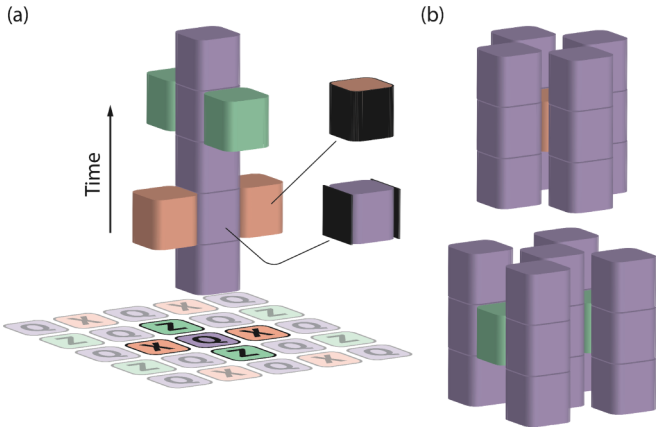

FIG. 8. (a) Error correction lattice of physical-qubit errors. When intramodule operations are ideal, measurement errors of module-qubit X-stabilizers are all due to errors on the surface of X-module (red) cubes, and module-qubit phase errors are all due to errors on the surface of Q-module (purple) cubes connected with Z-module (green) cubes. It is similar for measurement errors of module-qubit Z-stabilizers and module-qubit bit errors. (b) Triple-size lattices. On the triple-size lattice with an X-module cube at the center, errors near the boundary of the X-module cube are sufficiently considered. On the triple-size lattice with a Q-module cube at the center, errors near the boundary of the Q-module cube are sufficiently considered.

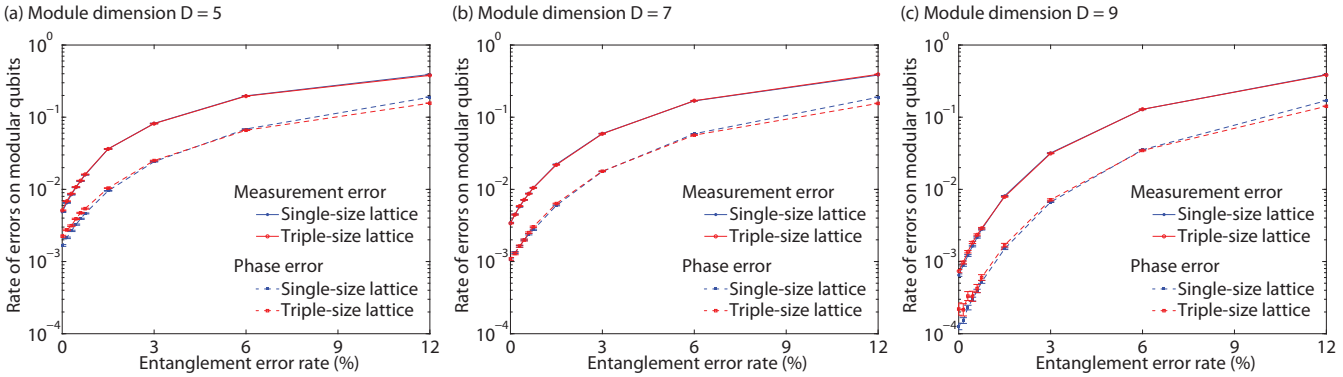

FIG. 9. Module-qubit error rates. The rate of measurement errors of module-qubit X-stabilizers P_{M} and the rate of module-qubit phase errors P_{P} are obtained with single-size lattices (individual cubes) and triple-size lattices [see Fig. 8(b)]. We have assumed that entanglement errors are unpolarized, i.e., $p_{\mathrm{X}} = p_{\mathrm{Y}} = p_{\mathrm{Z}}$, and all intramodule operations have the same error rate, $\epsilon_{\mathrm{I}} = \epsilon_{\mathrm{M}} = \epsilon_{1} = \epsilon_{2} = 0.1\%$.

APPENDIX E: SIMULATION OF THRESHOLDS

Logical-qubit error rates are obtained using the Monte Carlo method by simulating errors occurring in a logical qubit encoded in a $(2L-1)\times(2L-1)$ module array during L rounds of module-qubit stabilizer measurements. We have used Edmonds's minimum weight matching algorithm in the surface code error correction. In our simulations, we have set $n = (D+1)/2$ (see Fig. 3).

When intramodule operations are ideal, i.e., $\epsilon = 0$ (see Fig. 4), errors (due to entanglement errors) only occur on boundaries of connected cubes [see Fig. 8(a)]. These boundary surfaces are separated, hence errors on each boundary surface can be individually corrected and simulated. By simulating errors on the surface of an X-module cube [black surface of the red cube in Fig. 8(a)], we can find the rate P_{M} of measurement errors of module-qubit X-stabilizers (measurement errors after physical-qubit error correction). By simulating errors on the surface of a Q-module cube connected with a Z-module cube [black surface of the purple cube in Fig. 8(a)], we can find the rate P_{P} of module-qubit phase errors during one round of module-qubit stabilizer measurements. It is similar

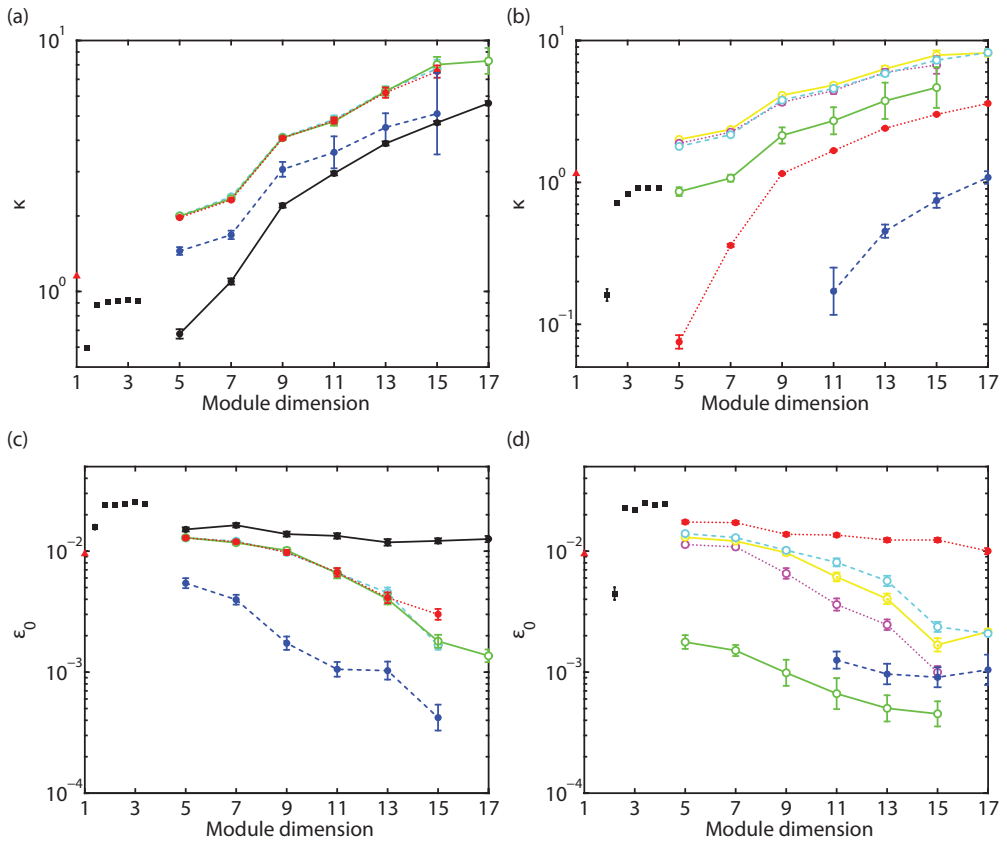

FIG. 10. Parameters ϵ_0 and κ for entanglement error rates of 1.5% (a, c) and 15% (b, d). Symbols are consistent with Fig. 5. For simple modules, the module dimension is always $D = 1$, and the number of purification tiers increases form left to right.

for measurement errors of module-qubit Z-stabilizers and module-qubit bit errors. With error rates P_M and P_P, we can simulate module-qubit errors on the conventional surface code error correction lattice to find the rate of logical-qubit errors.

When intramodule operations are not ideal, e.g., $\epsilon = 0.1\%$ (see Fig. 4), errors are not restricted to boundary surfaces. In this case, the first step in error correction (correcting physical-qubit errors) must be performed on the entire lattice [see Fig. 8(a)]. However, for large modules ($D \geqslant 5$), the entire lattice is too large to be directly simulated. Therefore, we individually simulate errors in each cube to approximately calculate the rate of module-qubit errors. We note that, when intramodule operations are not ideal, module-qubit errors also occur when module qubits are stabilized (corresponding to Q-module cubes without contact with X-module or Z-module cubes), which must be taken into account. Because we are interested in the case where intramodule operations have error rate $\epsilon = 0.1\%$, which is much lower than the 1% error-rate threshold for intramodule operations [30], the probabilities of error chains (after physical-qubit error correction) decrease rapidly with their lengths. Therefore, by simulating cubes individually, only the effects of errors (induced by intramodule operations) near boundary surfaces are approximately considered. In Fig. 9, we compare module-qubit error rates obtained with individual cubes and error rates obtained on triple-size lattices [see Fig. 8(b)]. Each triple-size lattice has dimension $\sim 3n$, where the boundary effect has been sufficiently considered. Even in the case of the smallest cube ($D = 5$), neglecting the boundary effect only slightly changes the error rates of module qubits.

To obtain each threshold point in Fig. 4, we need to generate a logical error rate versus entanglement error rate plot similar to the inset. In each such plot, we calculated 100 data points (25 data points are shown in the inset), and each data point corresponds to approximately 1 million simulations. To obtain all the data in this paper, we used 800 high-performance cores running for 2 weeks. In the inset in Fig. 4, the discrimination between entanglement error rates is 0.15%, which bounds the uncertainty of thresholds.

The computer code used to generate the results in this paper has been made openly available online at https://figshare.com/articles/network_FTQC/3473687

APPENDIX F: SIMULATION OF QUBIT COSTS

The qubit cost is calculated with parameters ϵ_0 and κ. For a given logical error rate ϵ_L, one can find the minimum L satisfying $\epsilon_L \geqslant \epsilon_0 e^{-\kappa L}$. This minimum L determines the size of the logical qubit. The total number of qubits in each logical qubit is $(2L-1)^2 \times S$, where S is the number of qubits in each module (module size).

Parameters ϵ_0 and κ are shown in Fig. 10. For simple modules, parameters ϵ_0 and κ are obtained by directly fitting logical qubit error rates for $L = 3, 5, 7, 9, 11$, with the function

$$\epsilon_L = \epsilon_0 e^{-\kappa L}. \tag{F1}$$

For large modules with $D \geqslant 5$, because module-qubit error rates are very low, it is hard to directly find logical-qubit error rates in simulations. Therefore, first, we obtain the module-qubit error rates (P_M, P_P), and then we find the logical-qubit error rates for module-qubit error rates (rP_M, rP_P). Here, the ratio r is chosen so that (rP_M, rP_P) are large enough for simulating logical-qubit error rates (not far below the threshold). Then we fit logical-qubit error rates for $L = 3, 5, 7, 9, 11$ with the function

$$\epsilon_L = e^{(\alpha \ln r + \beta)L + \gamma}. \tag{F2}$$

With fitting parameters α, β, and γ, we can obtain parameters $\kappa = \beta$ and $\epsilon_0 = e^{\gamma}$.

Standard deviations of parameters ϵ_0 and κ are also shown in Fig. 10. Small variations in these fitting parameters might change our estimations of the qubit cost in Fig. 5, but it is unlikely that our conclusion would be changed. For the low-fidelity network [Fig. 5(b)], the minimum cost of simple modules will change from 22 472 qubits to 20 808 qubits as we vary ϵ_0 and κ by an amount equal to their deviations, while the minimum cost of complex modules, which is 13 125 qubits, will not be changed (note that the cost is not a continuous function of ϵ_0 and κ).

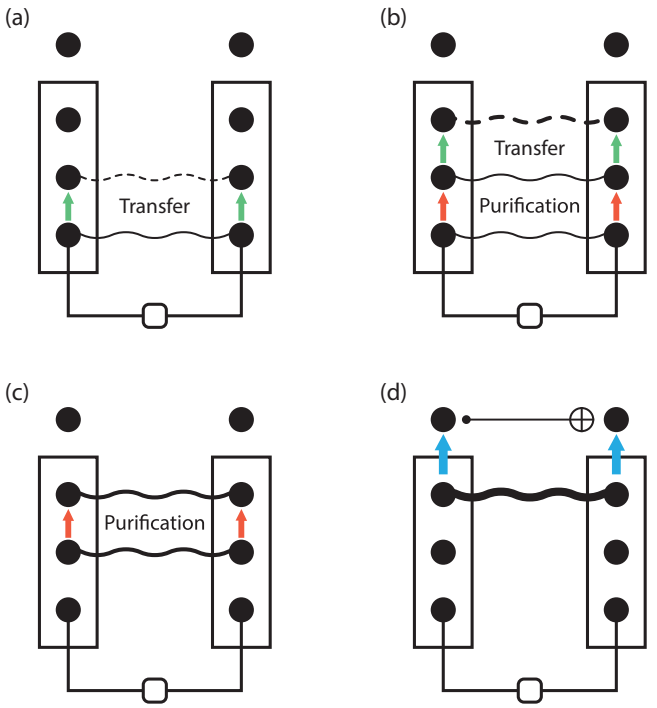

FIG. 11. Protocol of entanglement purification. (a) First, the raw entanglement is generated with optically coupled qubits (qubits at the bottom), and this raw entanglement is transferred to the upper pair of qubits via swap gates. Each swap gate is realized with three CNOT gates in our simulations. (b) The second raw engagement is generated with optically coupled qubits and is used to purify the raw entanglement of the upper pair of qubits. The first-tier purified entanglement is then transferred upward. (c) The second first-tier purified entanglement is prepared and used to purify the previous first-tier purified entanglement. With more qubits in each broker, by transferring purified entanglement upward, this purification process continues until reaching the top of the broker. (d) The finally purified entanglement is used to perform the distributed CNOT gates on client qubits.

APPENDIX G: PURIFICATION AND TIME COST

In our simulations, we have considered the purification protocol proposed in Ref. [4], in which phase errors and bit errors are corrected alternatively, i.e., in each tier of the purification either only phase errors or only bit errors are corrected. Circuits for bit-error purification and phase-error purification are shown in Figs. 6(b) and 6(c). The overall purification protocol is shown in Fig. 11. Note that in our simulations, for distributed CNOT gates involving an X (Z) ancillary qubit, phase (bit) errors are corrected in the first tier.

If the time cost of generating raw intermodule entanglement is much higher than the time cost of intramodule operations, the time cost of one round of module-qubit stabilizer measurements is $2n \times N \times \tau$, where half of the $4n$ rounds of physical-qubit stabilizer measurements (see Fig. 3) need intermodule entanglement, N is the number of raw entanglement pairs for preparing one pair of purified entanglement (see Table I), and τ is the time cost of preparing one pair of raw entanglement. We have taken $n = (D + 1)/2$ in our simulations. For simple modules, if each module has only one broker, the time cost is amplified by a factor of 2. The overall time cost of the computing, i.e., the total rounds of module-qubit stabilizer measurements, will of course also depend on the both the selected decoding algorithm and the high-level algorithm.

[1] W. Dür and H.-J. Briegel, Entanglement Purification for Quantum Computation, Phys. Rev. Lett. **90**, 067901 (2003).

[2] S. C. Benjamin, D. E. Browne, J. Fitzsimons, and J. J. L. Morton, Brokered graph-state quantum computation, New J. Phys. **8**, 141 (2006).

[3] E. T. Campbell, Distributed quantum-information processing with minimal local resources, Phys. Rev. A **76**, 040302(R) (2007).

[4] L. Jiang, J. M. Taylor, A. S. Sørensen, and M. D. Lukin, Distributed quantum computation based on small quantum registers, Phys. Rev. A **76**, 062323 (2007).

[5] S. C. Benjamin, B. W. Lovett, and J. M. Smith, Prospects for measurement-based quantum computing with solid state spins, Laser Photon. Rev. **3**, 556 (2009).

[6] Y. Li and S. C. Benjamin, High threshold distributed quantum computing with three-qubit nodes, New J. Phys. **14**, 093008 (2012).

[7] K. Fujii, T. Yamamoto, M. Koashi, and N. Imoto, A distributed architecture for scalable quantum computation with realistically noisy devices, arXiv:1202.6588.

[8] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects, Phys. Rev. A **89**, 022317 (2014).

[9] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, High-Fidelity Preparation, Gates, Memory and Readout of a Trapped-Ion Quantum Bit, Phys. Rev. Lett. **113**, 220501 (2014).

[10] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, High-Fidelity Quantum Logic Gates Using Trapped-Ion Hyperfine Qubits, Phys. Rev. Lett. **117**, 060504 (2016).

[11] J. P. Gaebler, T. R. Tan, Y. Lin, Y. Wan, R. Bowler, A. C. Keith, S. Glancy, K. Coakley, E. Knill, D. Leibfried, and D. J. Wineland, High-Fidelity Universal Gate Set for $^{9}Be^{+}$ Ion Qubits, Phys. Rev. Lett. **117**, 060505 (2016).

[12] H. Bernien, B. Hensen, W. Pfaff, G. Koolstra, M. S. Blok, L. Robledo, T. H. Taminiau, M. Markham, D. J. Twitchen, L. Childress, and R. Hanson, Heralded entanglement between solid-state qubits separated by three metres, Nature **497**, 86 (2013).

[13] N. Roch, M. E. Schwartz, F. Motzoi, C. Macklin, R. Vijay, A. W. Eddins, A. N. Korotkov, K. B. Whaley, M. Sarovar, and I. Siddiqi, Observation of Measurement-Induced Entanglement and Quantum Trajectories of Remote Superconducting Qubits, Phys. Rev. Lett. **112**, 170501 (2014).

[14] D. L. Moehring, P. Maunz, S. Olmschenk, K. C. Younge, D. N. Matsukevich, L.-M. Duan, and C. Monroe, Entanglement of single-atom quantum bits at a distance, Nature **449**, 68 (2007).

[15] D. Hucul, I. V. Inlek, G. Vittorini, C. Crocker, S. Debnath, S. M. Clark and C. Monroe, Modular entanglement of atomic qubits using photons and phonons, Nat. Phys. **11**, 37 (2015).

[16] K. R. Brown, A. C. Wilson, Y. Colombe, C. Ospelkaus, A. M. Meier, E. Knill, D. Leibfried, and D. J. Wineland, Single-qubit-gate error below 10^{-4} in a trapped ion, Phys. Rev. A **84**, 030303(R) (2011).

[17] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, Superconducting quantum circuits at the surface code threshold for fault tolerance, Nature **508**, 500 (2014).

[18] A. Kitaev, Fault-tolerant quantum computation by anyons, Ann. Phys. **303**, 2 (2003).

[19] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, J. Math. Phys. **43**, 4452 (2002).

[20] N. H. Nickerson, Y. Li, and S. C. Benjamin, Topological quantum computing with a very noisy network and error rates approaching one percent, Nat. Commun. **4**, 1756 (2013).

[21] N. H. Nickerson, J. F. Fitzsimons, and S. C. Benjamin, Freely Scalable Quantum Technologies Using Cells of 5 to 50 Qubits with Very Lossy and Noisy Photonic Links, Phys. Rev. X **4**, 041041 (2014).

[22] J. Eisert, K. Jacobs, P. Papadopoulos, and M. B. Plenio, Optimal local implementation of nonlocal quantum gates, Phys. Rev. A **62**, 052317 (2000).

[23] A. G. Fowler, A. M. Stephens, and P. Groszkowski, High-threshold universal quantum computation on the surface code, Phys. Rev. A **80**, 052312 (2009).

[24] R. Raussendorf and J. Harrington, Fault-Tolerant Quantum Computation with High Threshold in Two Dimensions, Phys. Rev. Lett. **98**, 190504 (2007).

[25] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[26] Y. Li, A magic state's fidelity can be superior to the operations that created it, New J. Phys. **17**, 023037 (2015).
[27] J. O'Gorman and E. T. Campbell, Quantum computation with realistic magic state factories, arXiv:1605.07197.
[28] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).
[29] http://dx.doi.org/10.5281/zenodo.22558
[30] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Surface code quantum computing with error rates over 1%, Phys. Rev. A **83**, 020302(R) (2011).
[31] V. Kolmogorov and V. Blossom, A new implementation of a minimum cost perfect matching algorithm, Math. Program. Comput. **1**, 43 (2009).
[32] M. Herold, E. T. Campbell, J. Eisert, and M. J. Kastoryano, Cellular-automaton decoders for topological quantum memories, npj Quantum Info. **1**, 15010 (2015).