Concatenated pieceable fault-tolerant scheme for universal quantum computation

Chen Lin and GuoWu Yang*

Big Data Research Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, People's Republic of China

(Received 19 April 2020; accepted 30 October 2020; published 17 November 2020)

As existing approaches to get around the restriction of the no-go theorem generally exhibit high ancillary qubit overhead costs, we propose a scheme for universal fault-tolerant quantum computation by using the pieceable fault-tolerant protocol and code switching techniques. In additional, by utilizing the construction of the pieceable fault-tolerant circuit, we adopt a decoding strategy based on a deep-neural-network algorithm to improve the error threshold of the non-Clifford logical gate circuit. We describe our universal construction in detail with a two-level nonuniform concatenated 25-qubit code and perform numerical simulations to analyze the depolarizing noise threshold of a given universal gate set with this code. The resources required to implement this universal gate set are also estimated to further demonstrate the efficiency of our scheme. We compare these results with the existing universal concatenation methods and conclude that our method outperforms them in terms of the lower bound of the error threshold and qubit resource overhead.

DOI: 10.1103/PhysRevA.102.052415

I. INTRODUCTION

Quantum computers have generated great interest due to their enormous potential for accelerating classical algorithms and simulating physical phenomena that are intractable using classical computers [1–3]. It is hoped that a very large quantum computer will be realized that shows performance superior to that of classical computers for some problems of intrinsic interest. Actually, practical quantum computation may become available soon [4,5], and intermediate-scale quantum computers based on various quantum technologies, such as ion traps [6–8] or superconducting qubits [8,9], have already been established, providing the opportunity to implement some simple quantum algorithms [10,11] with the number of qubits ranging from 50 to a few hundred.

However, to protect the quantum information from environmental noise and carry out large-scale computational tasks with high fidelity, it is necessary to carefully choose an appropriate method to prevent multiplication and propagation of the errors introduced by noise. For instance, when considering the quantum circuits under a standard depolarization error model, an error may occur on one of the two qubits, which will propagate to another qubit after we apply a two-qubit coupling gate. Such a single error will be multiplied through the subsequent circuits until the first error-correction procedure and finally it may lead to a logical error after the application of active error correction.

In fact, most quantum algorithms are generally implemented in terms of quantum circuits that decompose a computational task into a sequence of elementary quantum gates acting on qubits. A common strategy is to directly execute a given set of quantum gates $\{U_i\}$ on encoded logical qubits in a fault-tolerant manner [12–14], rather than performing the risky physical operation on physical qubits. Meanwhile, as there always exists a set of gates such that any quantum circuit can be approximated with arbitrary precision by the product of several elements in this set, it is necessary to focus on the design of appropriate fault-tolerant protocols for the gates in this specific set. This set is called the universal quantum gate set and can be adopted to synthesize all other quantum gates. A universal gate set can be implemented in a fault-tolerant manner using different schemes. However, such fault-tolerant design usually requires considerable additional quantum resources due to the limitation of the no-go theorem [15,16], such as in the preparation of the magical state and its distillation procedure [17–19], or the combination of two different codes with complementary transversal gate sets into a single larger code with universal fault-tolerant gates but with more ancillary resources consumed [20,21].

Some recent improvements have motivated efforts to search for the optimal approach to reduce the resources of fault-tolerant universal computation. Chao and Reichardt [22] proposed a novel fault-tolerant scheme for extracting error syndrome information, which significantly reduces the auxiliary qubit resources required in the active error-correction process. By using the neural-network method, Chamberland and Ronagh [23] designed several efficient decoding algorithms and applied them to analyze several fault-tolerant error-correction protocols; they pointed out that the ability of noise resistance of fault-tolerant quantum circuits can be improved. Nikahd *et al.* [24] focused on optimizing the qubit overhead of concatenated code by combining two codes C_1 and C_2 in a nonuniform concatenated fashion.

In particular, Maslov and co-workers [25,26] have pointed out that, for a given circuit, the quantum gates of different Clifford hierarchy can be maximally gathered by using an efficient synthesis algorithm that greatly reduces the gate

*guowu@uestc.edu.cn

depth of the non-Clifford gate. Moreover, for Calderbank-Shor-Steane (CSS) codes [27,28], the Clifford group can be implemented fault tolerantly with little cost. Therefore, we can design an appropriate fault-tolerant structure for the different implementation steps of the non-Clifford circuit and the encoded information can be exchanged locally through code space conversion techniques [29–33]. More recent studies [26,34–36] have proposed several synthesis algorithms that can maximally parallelize the possible T gates of the circuit G belonging to the circuit library {CNOT, T} and make other moments of the application of G only contain controlled-NOT (CNOT) gates, where $T = \text{diag}(1, \exp(\frac{\pi}{4}i))$. The CNOT gates belonging to the Clifford group have a simple transversal fault-tolerant structure in CSS codes and Reed-Muller codes [37]. On the other hand, by fault-tolerantly exchanging encoded data between the two different codes C_1 and C_2, where each nontransversal gate in C_1 has a transversal implementation in C_2 and vice versa, a given universal gate set can be transversally implemented without magic state distillation. However, the code conversion procedure usually contains several rounds of ancillary preparation and error correction, which makes it easier to cause a logical error than other fault-tolerant quantum operations.

We also observed that some researchers try to use classical machine learning algorithms to improve the fault tolerance of encoded circuits [23,38,39]. These works have proposed several efficient algorithms to optimize the decoding procedure, and their numerical simulation has shown that these adjusted decoding procedures can effectively suppress error propagation.

These works inspired us to optimize the scheme of fault-tolerant universal computation based on code concatenation, that is, for each physical quantum gate in the logical non-Clifford circuit G on a given code C_1, different inner codes can be selected to further protect it so that any quantum gate in G is transversal in the current inner-layer code. Different inner-layer codes can be locally switched between each other by fault-tolerant code conversion techniques with neural-network-based decoding scheme; such processing will obtain a pieceable fault-tolerant [31] variant of G, and an appropriate inner code will facilitate the reduction of qubit overhead.

Our fault-tolerant non-Clifford circuit structure ensures that every single-qubit error in the inner code block will not spread to form a globally logical error when compared with the original concatenation strategy, which means that there is no need to sacrifice the overall concatenation code distance to realize fault-tolerant universal computation. Here we use the term “globally” to refer to the entire two-level concatenation code to distinguish it from the local logical qubit contained in this code.

In this work, we first combine the nonuniform code concatenation and code conversion methods to design a universal fault-tolerant quantum computation scheme. An example is given to demonstrate the fault tolerance of this scheme, and numerical simulation is also performed to further analyze the performance in terms of the pseudothreshold and asymptotic threshold. For the standard depolarizing error model, our simulated results show that a higher-threshold lower bound can be obtained by our scheme. We further analyze the upper bounds of the qubit resource based on our fault-tolerant scheme for simulating quantum circuits in the library {CNOT, H, T}, We compare our results with the 49- and 105-qubit codes proposed in Ref. [21] and show that our consumption of fault-tolerant Clifford gates is lower than their scheme.

The rest of the paper is organized as follows. In Sec. II we first introduce the basic concepts of quantum error correction and fault tolerance and then introduce the nonuniform concatenation strategy and code conversion techniques. In Sec. III we describe our fault-tolerant universal computation scheme. In Sec. IV we provide a specific implementation case to further explain our fault-tolerant scheme and analyze the performance and resource consumption for our 25-qubit code. In Sec. V we further consider the recursive simulation method and provide an upper-bound analysis of qubit resource consumption to implement the fault-tolerant circuit generated by CNOT and H gates. We summarize in Sec. VI.

II. STABILIZER CODE AND FAULT-TOLERANT COMPUTATION

We first describe the stabilizer formalism [14]. The quantum symbol $[[n; k; d]]$ usually refers to a stabilizer code space C_n that consists of some physical n-qubit states that are commonly $+1$ eigenstates of $n - k$ given stabilizer operators (generators). These operators typically belong to the n-qubit Pauli group P_n, are independent of each other, and generate a matrix product group called the stabilizer group S. The code distance d describes the error-correction ability of the stabilizer code and means that any Pauli error of weight $w \leqslant \lfloor \frac{d-1}{2} \rfloor$ can be detected and corrected by syndrome measurements and error-correction processes.

For a given computation task, how to perform operations on a logical state without losing the code’s protection against errors is a highly important topic. So the encoded information should be handled in a fault-tolerant way. We say that $t = \lfloor \frac{d-1}{2} \rfloor$, a quantum operation which is protected by a code with distance d, is t-fault tolerant if the following two conditions are satisfied [23].

(i) For an input codeword with error of weight w_1, if w_2 single-qubit faults occur during the operation with $w_1 + w_2 \leqslant t$, ideally decoding the output state gives the same codeword as ideally decoding the input state.

(ii) For w single-qubit faults during the implementation of a fault-tolerant operation with $w \leqslant t$, no matter how many errors are present in the input state, the output state differs from a codeword by an error with its weight no more than w.

Here we assert that ideally decoding is equivalent to performing a round of noise-free error correction. Actually, this noise-free process is mainly used to determine whether a noncorrectable error has occurred in the error-correction process during the application of a circuit. Both conditions are required to ensure that correctable errors do not propagate through the entire circuit and prevent the errors from accumulating between the different error-correction rounds.

Following these two principles, several paradigms for designing a fault-tolerant encoded operation have been proposed. For most quantum error-correction codes, an efficient fault-tolerant protocol for the operators belonging to the Clifford group usually exists, such as transversality. The Clifford group is a finite group of symmetries of the Pauli group [28] and can be generated by the CNOT, H, and P gates. For instance, encoded Clifford operators can be implemented transversally in the CSS code. However, universal quantum computing cannot be realized with only the Clifford operators, and non-Clifford fault-tolerant operations with low resource overhead are highly desirable for achieving large-scale universal quantum computation

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1}$$

By sacrificing the global code distance, the concatenated quantum error-correction scheme can provide a universal fault-tolerant encoded operation set for a given code. This scheme requires that the outer code and the inner code have complementary transversal encoded gates to form a globally fault-tolerant universal gate set. Meanwhile, the recovery operations (stabilizer measurement and error-correction operations) for the two codes must be globally transversal in the full concatenated code space. Recently, Chamberland *et al.* [21] rigorously analyzed the numerical values of the threshold of the concatenated 105-qubit code and the 49-qubit code in a given error model. The choice for the inner code of their concatenation scheme must be restricted to the 15-qubit Reed-Muller (RM-15) code because of its transversal construction for the T gate. However, this concatenation scheme for the Hadamard gate is less fault tolerant for single-qubit error because the encoded H gate for the RM-15 code is not 1-fault tolerant, making its error threshold much lower. On the other hand, the qubit overhead for the Clifford gate protected by these two concatenated codes is very large.

It should be noted that for a given encoded circuit, the code under this circuit will transfer to several intermediate codes when applying this circuit; if each intermediate code has a distance large enough to correct any correctable errors that may have arisen or if we carefully design an appropriate intermediate error-correction procedure to avoid possible error propagation, then the entire circuit can be pieceable fault tolerant [31]. Actually, it would be convenient to intuitively imagine that a certain encoded circuit $\mathcal{C}$ has the decomposition

$$\mathcal{C} = \mathcal{C}_r\mathcal{C}_{r-1}\cdots\mathcal{C}_1, \tag{2}$$

where r refers to the minimum number of circuit pieces into which $\mathcal{C}$ can be divided under the premise of satisfying fault tolerance.

We can obtain a fault-tolerant variant of circuit $\mathcal{C}$ if each $\mathcal{C}_i$ is carefully designed such that error propagation can be avoided. We now give the modified variant of $\mathcal{C}$, denoted by $\tilde{\mathcal{C}}$, which can be fault tolerant,

$$\tilde{\mathcal{C}} = \mathcal{E}_r\mathcal{C}_r\mathcal{E}_{r-1}\mathcal{C}_{r-1}\cdots\mathcal{E}_1\mathcal{C}_1, \tag{3}$$

where $\mathcal{E}_i$ is an adapted error-correction process. In fact, by performing the error correction after each $\mathcal{C}_i$ on the encoded data, we obtain several fault-tolerant gadgets $\mathcal{E}_i\mathcal{C}_i$ $(i = 1, \dots, r)$. This kind of design paradigm may broaden the choice of codes for the concatenation scheme and provide an opportunity for the reduction of the overall cost of concatenated fault-tolerant universal quantum computation.

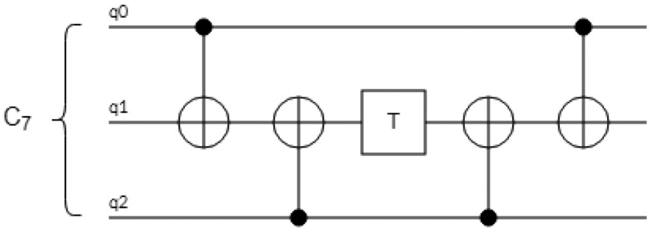

FIG. 1. Typical logical T gate for the CSS-7 code C_7, where qubits 0, 1, and 2 participate during the implementation of this logical circuit. The other four physical qubits that comprise the logical qubit are idle qubits, so they are not shown in this figure.

III. CONCATENATED PIECEABLE FAULT-TOLERANT PROTOCOL FOR THE REALIZATION OF NON-CLIFFORD GATES

Our method is based on the nonuniform code concatenation scheme [24] in which a logical qubit is encoded by code C_1 at the first level of the encoding and only the active qubits during the application of a nontransversal logical gate on C_1 will be further encoded by code C_2; then the qubit overhead of this scheme will be lower than that of the uniform scheme [20]. Generally, for the nonuniform concatenated code, we note that each encoding level can employ more than one code and this design can be used to efficiently construct a fault-tolerant non-Clifford logical gate.

For example, we note that a logical T gate in the CSS-7 code can be implemented by a CNOT gate and a single-qubit T gate, as shown in Fig. 1. We observe that the T gate is not transversal, which makes the single-qubit error in this circuit prone to spreading. Additional protection should be adopted to suppress this kind of error. For the concatenation scheme, a common strategy is that all of the individual qubits that comprise the CSS-7 logical qubit can be further encoded by the [[15; 1; 3]] quantum Reed-Muller code to obtain a 105-concatenated code. Because both the CNOT gate and the T gate are transversal on the inner code, the entire logical T gate can be globally fault tolerant.

However, for a specific logical gate, not all physical qubits interact during its application, i.e., the qubits under the outer code C_1 can be partitioned into the active qubit set Q_1 and the idle qubit set Q_2. A natural idea is to only encode the active qubits at the second encoding level, such as the 49- or 45-qubit code [24]. However, for the two-level concatenation scheme, given a universal gate set, there always exists an element that cannot be transversally applied on both the outer and inner codes (no-go theorem), which reduces the effective code distance of the concatenated code.

For example, the logical Hadamard gate in the RM-15 code is not 1-fault tolerant, as shown in Fig. 2. Logical H for the 105-qubit code is implemented fault tolerantly by applying each non-fault-tolerant logical H gate transversally. Assuming that a single-qubit error occurs in any two inner code blocks, the single-qubit error in these two inner code blocks may form a logical error through propagation. Because the code distance

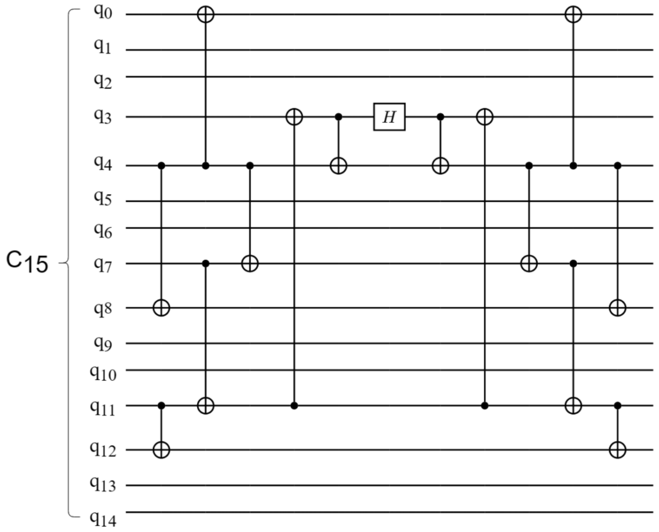

FIG. 2. Logical H gate for the [15; 1; 3] Reed-Muller code C_{15}; there is a total of nine time steps for its implementation.

of the first encoding level is 3, two such single-logical-qubit errors eventually may cause a global logical error in the 105-qubit code.

Inspired by the idea of code conversion, we observe that for a specific logical gate circuit, the inner code need not be restricted to only one code during each time step the circuit is applied. More specifically, the inner code should be further adjusted to make the protected component at least 1-fault tolerant. Therefore, the realization of this strategy depends on the fault-tolerant conversion technology for the intermediate code when we sequentially apply each component of a logical circuit. We also call this scheme concatenated pieceable fault tolerant (PFT). A similar idea is responsible for the success of the code deformation approach [31] that changes the error-correcting code such that a full cycle returning to the original code implements a logical gate.

IV. NONUNIFORM CONCATENATED 25-QUBIT CODE FOR UNIVERSAL FAULT-TOLERANT COMPUTATION

To further illustrate our fault-tolerant scheme, we first construct a 25-qubit two-level nonuniform concatenated code in order to prepare the T gate. Then we specifically demonstrate that this logical T gate construction in this code can be at least 1-fault tolerant in each time step of its application. Finally, with the addition of the transversal Clifford gates in this concatenated code, we give a fault-tolerant universal set known as Clifford + T.

A. Logical construction of non-Clifford gate T

In the construction of the 25-qubit code, the outer code C_1 and inner code C_2 are selected as the CSS-7 code; its stabilizer generators are listed in Table I. Since any gate in the Clifford group has transversal logical construction in this code [28], we can focus our attention on the non-Clifford gate T. Following the paradigm of the concatenated PFT scheme, the selection of the inner code is related to the construction of a logical T circuit, as shown in Fig. 3.

TABLE I. Generators of the stabilizer S and logical Pauli operators of the CSS-7 Steane code.

Stabilizer generators	Logical Pauli operator
$X_0X_2X_4X_6$	
$X_1X_2X_5X_6$	
$X_3X_4X_5X_6$	
$Z_0Z_2Z_4Z_6$	
$Z_1Z_2Z_5Z_6$	$\bar{X} = X_0X_1X_2$
$Z_3Z_4Z_5Z_6$	$\bar{Z} = Z_0Z_1Z_2$

It can be observed that in the first and fifth stages, we only need to guarantee that the inner code has a transversal logical CNOT gate, so we first choose the CSS-7 code as the inner-layer code. Meanwhile, with the nonuniform scheme, only the active qubits (qubits 0–2) are protected. In the third stage, we find that only the second active logical qubit should be transformed into some code that has a transversal logical T gate and such a code can be selected as the Reed-Muller code. Therefore, an additional fault-tolerant conversion procedure for the code under the second logical qubit should be added so that the second and fourth stages actually complete the transformation between the CSS-7 code and the RM-15 code. We note that such a code transformation is only restricted on the second active logical qubit and such a procedure can also be considered as a global conversion between the 25-qubit code and the 33-qubit code.

B. Pieceable fault tolerance of the logical T gate of the 25-qubit code

From Fig. 3 we noted that the logical component in the first, third, and fifth stages is transversal in the current inner code and the transversal structure is automatically fault tolerant. Therefore, we only need to guarantee that the code conversion procedure in the second and fourth stages is fault

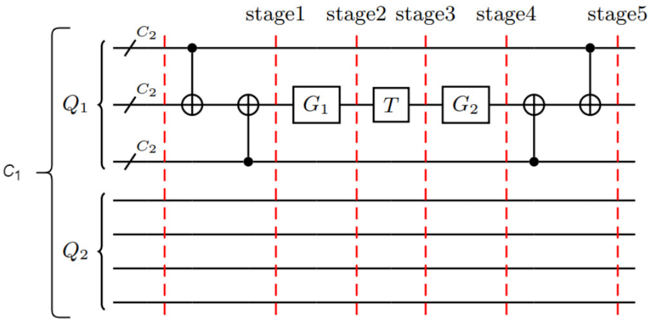

FIG. 3. Concatenated PFT T gate in the 25-qubit two-level concatenation code. The active qubit set Q_1 and idle qubit set Q_2 are marked in the figure. We partition the logical T circuit into five stages, and only the active qubits (0–2) are further protected by the inner code C_2. The circuit G_1 is used to fault-tolerantly convert CSS-7 code to RM-15 code, and G_2 is the opposite conversion. In stage 3, we only implement a transversal logical T gate on the second logical qubit, which is encoded by the RM-15 code. Therefore, the CNOT and T gates in this circuit are all at least 1-fault tolerant under the intermediate inner code, as required by our design guidelines.

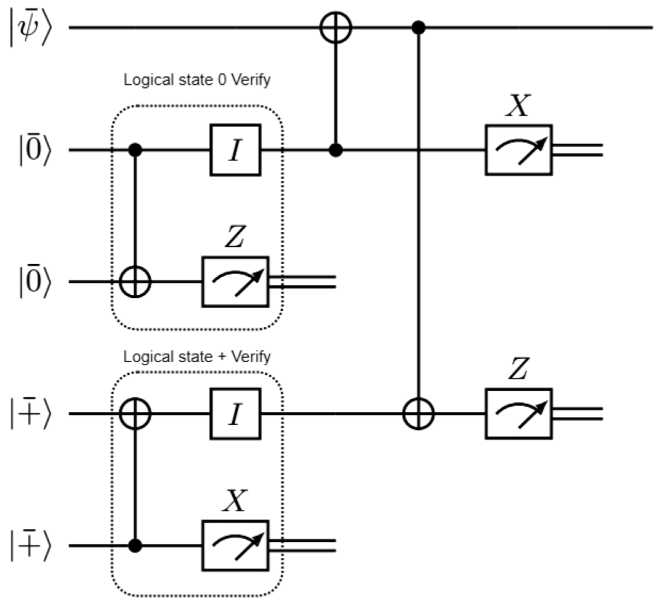

FIG. 4. Steane method for stabilizer measurement and error correction. We denote by $|\bar{\psi}\rangle$ the encoded data logical qubit. Note that the error-correction process includes two verification subcircuits for preparation of ancillary $|\bar{0}\rangle$ and $|\bar{+}\rangle$.

tolerant. In the construction of our concatenated code, for the H and CNOT gates, we use Steane's method for the fault-tolerant stabilizer measurement proposed in Ref. [13], with its details shown in Fig. 4. For G_1 and G_2, we use the Schrödinger cat state for the fault-tolerant stabilizer measurement.

For the construction of G_1 and G_2, we follow the method in Ref. [30], where a general method for fault-tolerant switching between different Reed-Muller codes was proposed. The circuit G_1 that convert CSS-7 to RM-15 is shown in Fig. 5 and the reversed transformation G_2 is shown in Fig. 13 in Appendix A.

From the description of Fig. 5, we conclude that the conversion process is fault tolerant. Moreover, because the other two logical qubits (qubits 0 and 2) do not interact with other inner code blocks or physical qubits, they can be regarded as a waiting gate during the conversion process. Therefore, we model this process as fault-tolerant idle 1-Rec [13]; this kind of model is detailed in Fig. 6.

We have guaranteed the fault tolerance of the logical T gate in the 25-qubit code, but, similar to other code conversion methods, the circuits G_1 and G_2 also need to perform long quantum computations (ancillary preparation and stabilizer measurement). In fact, to ensure that single fault does not spread, G_i $(i = 1, 2)$ is usually divided into several pieces such that each piece that contains an adapted error-correction process is fault tolerant. According to Eq. (2) and the description in Fig. 5, it can be seen that G_i is actually a pieceable fault-tolerant circuit. Furthermore, since each piece of G_i is at least 1-fault tolerant, with the consideration of possible error propagation, we claim that G_i is only a 1-fault-tolerant circuit.

TABLE II. Stabilizers of the intermediate codes during the fault-tolerant conversion procedure from the CSS-7 to the RM-15 code. More specifically, in the measurement of the stabilizers $Z_5Z_6Z_{13}Z_{14}$, $Z_4Z_6Z_{12}Z_{14}$, and $Z_2Z_6Z_{10}Z_{14}$, these Pauli operators are always the stabilizer of the intermediate code until we finally obtain the RM-15 logical state.

X-type stabilizers	Z-type stabilizers
$X_7X_8X_9X_{10}X_{11}X_{12}X_{13}X_{14}$	$Z_7Z_8Z_9Z_{10}Z_{11}Z_{12}Z_{13}Z_{14}$
$X_3X_4X_5X_6X_{11}X_{12}X_{13}X_{14}$	$Z_3Z_4Z_5Z_6Z_{11}Z_{12}Z_{13}Z_{14}$
$X_1X_2X_5X_6X_9X_{10}X_{13}X_{14}$	$Z_1Z_2Z_5Z_6Z_9Z_{10}Z_{13}Z_{14}$
$X_0X_2X_4X_6X_8X_{10}X_{12}X_{14}$	$Z_0Z_2Z_4Z_6Z_8Z_{10}Z_{12}Z_{14}$

However, the syndrome extraction needs to apply many two-qubit control gates and the error probability of the double-qubit gate is the same as that of the single-qubit gate for the standard depolarization noise model, which increases the possibility of the occurrence of weight-2 Pauli error and eventually makes the error threshold of G_i much lower compared to other fault-tolerant components in the T-gate circuit. Specifically, we use the Monte Carlo method to analyze the fraction of malignant locations [21] of G_i with the restriction that only one component has an error. Here we denote this fraction by f_{1,G_i}, and from numerical simulation we find that $\max\{f_{1,G_1}, f_{1,G_2}\} = 0.079 \pm 0.0014$. In summary, the low fault tolerance of G_i greatly reduces the error threshold of the T gate.

We observe that the syndrome information of each piece is related to the syndrome of the previous one. For example, as shown in Fig. 7, some weight-2 Pauli errors caused by a two-qubit coupling gate cannot be corrected in a single piece of G_i, but taking into consideration that the errors occurring after different quantum gates are independent and the physical error probability p is very small (usually 10^{-4}–10^{-3}), if a stabilizer of a single piece of G_i is triggered, then the possibility that the next circuit piece also has an error is considerably low.

Therefore, we have reason to believe that the occurrence of such error configurations discussed in Fig. 7 is more often than the case in which two errors occurred independently at different circuit pieces. In conclusion, by combining the syndrome data of different pieces, we can use a pattern recognition algorithm to infer these weight-2 errors. Next we introduce the deep-neural-network method to improve the performance of G_i.

C. Deep-neural-network model

Generally, a typical error-correction process can be described by matching the syndrome vector $\mathbf{s}$ with the most likely recovery operator $R_{\mathbf{s}}$ and applying this operator to the logical state. For a pieceable fault-tolerant circuit composed of r pieces, we define its final recovery operator as

$$R = \prod_{i=1}^{r} R_{\mathbf{s}_i} \quad (i = 1, \dots, r), \tag{4}$$

where $\mathbf{s}_i$ is the syndrome vector obtained by the error-correction process of circuit piece $\mathcal{C}_i$ and $R_{\mathbf{s}_i}$ is the corresponding recovery operator.

Next we will construct an adjusted decoding procedure based on the deep-learning multiclassification technique. More details can be found in [40,41].

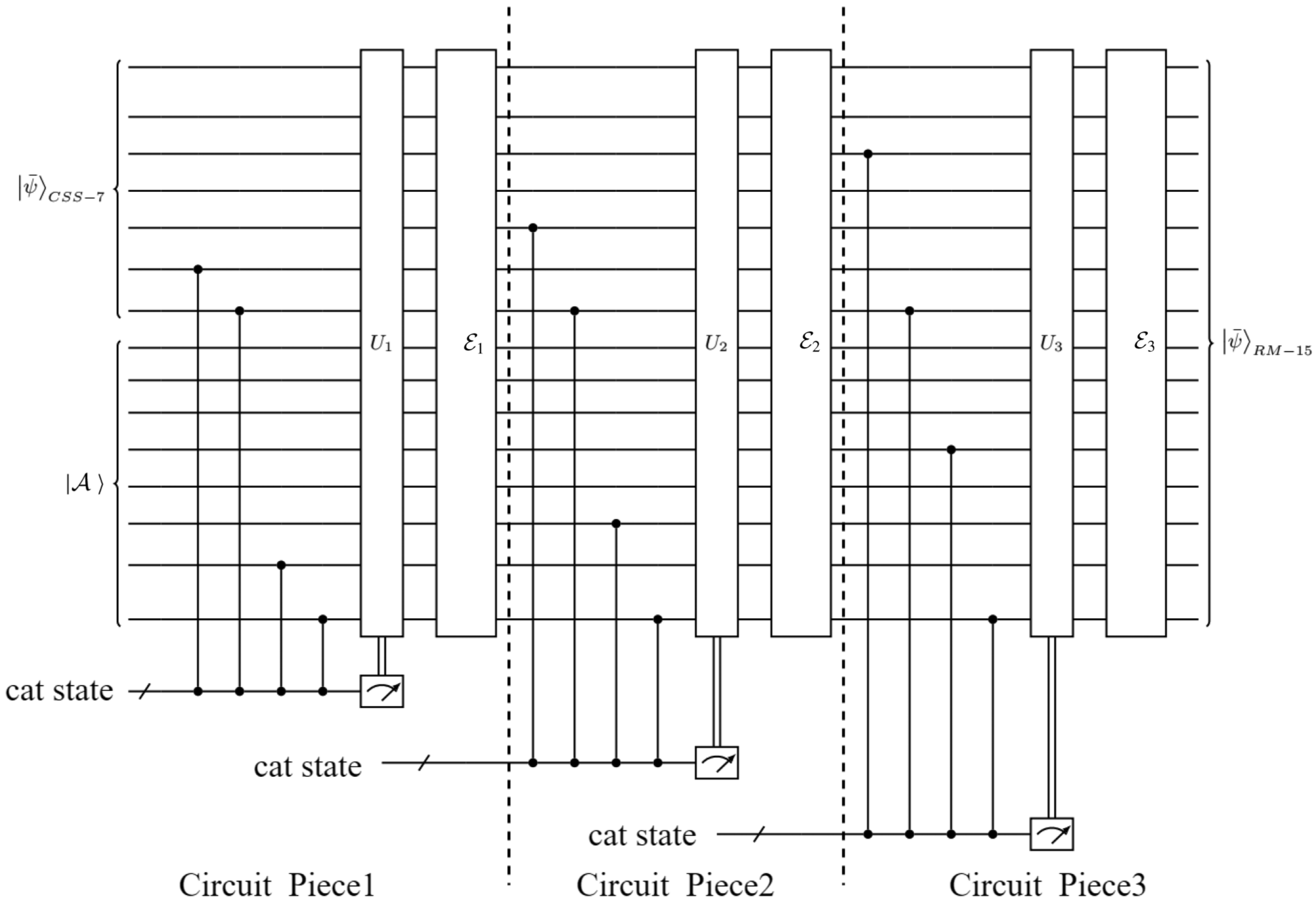

FIG. 5. Subcircuit G_1 that realizes fault-tolerant conversion from CSS-7 code to RM-15 code. We denote by $|\mathcal{A}\rangle$ the CSS-7 encoded $|\bar{+}\rangle$ prepared by the Steane method that interacts with the physical state $|0\rangle$ through the transversal CNOT gate. Then we divide the entire conversion process into three pieces: (1) Add the ancillary state $|\mathcal{A}\rangle = \frac{|\bar{0}\rangle|0\rangle+|\bar{1}\rangle|1\rangle}{\sqrt{2}}$ to the system, with the CSS-7 data encoding state $|\bar{\psi}\rangle_{CSS\text{-}7}$ constituting a new physical system $|\bar{\psi}\rangle_{CSS\text{-}7} \otimes |\mathcal{A}\,\rangle$; project this code state with stabilizer $Z_5Z_6Z_{13}Z_{14}$ and perform a round of error correction. Here the operator $U_1 = X_0X_2X_4X_6$. (2) Project the code state with stabilizer $Z_4Z_6Z_{12}Z_{14}$ and perform a round of error correction. The operator $U_2 = X_1X_2X_5X_6$. (3) Project the code state with stabilizer $Z_2Z_6Z_{10}Z_{14}$ and perform a round of error correction. The operator $U_3 = X_3X_4X_5X_6$. Table II shows the stabilizers used in $\mathcal{E}_1$ and $\mathcal{E}_2$. After step 3 we obtain the RM-15 encoded state $|\bar{\psi}\rangle_{RM\text{-}15}$.

1. Data set and labels

First we consider G_1. After its implementation in the initial CSS-7 encoded state, we collect all of the syndrome vectors generated by its leading error-correction circuit (LEC) and trailing error-correction circuit (TEC) as input data. The syndrome vectors in the LEC come from the fault-tolerant preparation of $|\bar{\psi}\rangle_{CSS\text{-}7}$ and the auxiliary state $|\mathcal{A}\rangle$ and the error correction of the initial input state $|\bar{\psi}\rangle_{CSS\text{-}7} \otimes |\mathcal{A}\rangle$. The syndrome vectors in the TEC come from $\mathcal{E}_i$ ($i = 1, 2, 3$). We then combine these syndrome data as $\mathbf{s} := \mathbf{s}_{LEC} \times \mathbf{s}_1 \times \mathbf{s}_2 \times \mathbf{s}_3$ and redefine the recovery operator of G_1 as

$$R_{G_1} = X_L^{g_X(\mathbf{s})} Z_L^{g_Z(\mathbf{s})} \prod_{i=1}^{3} R_{\mathbf{s}_i}, \tag{5}$$

where X_L and Z_L are logical Pauli operators and $g_X(\mathbf{s}), g_Z(\mathbf{s}) \in \mathcal{Z}_2$. So after applying $\prod_{i=1}^{3} R_{\mathbf{s}_i}$ according to the error-syndrome lookup table, the two functions are actually reused syndrome data to further predict the logical error in the final output state. Therefore, the next work is to find these two functions g_X and g_Z.

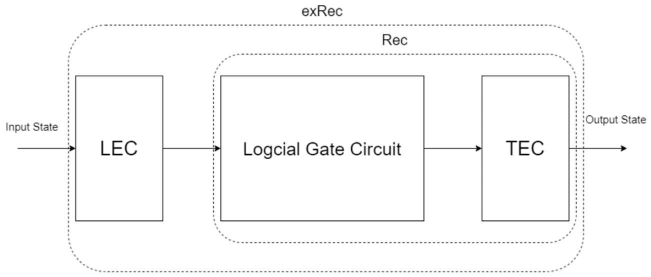

FIG. 6. For a given error-correction code, any physical quantum circuit has its corresponding logical variant in this code, also called rectangle (Rec). A 1-exRec is a level-one Rec along with its leading error-correction circuit (LEC) and trailing error-correction circuit (TEC).

We define the data set as $D \subseteq \{\mathbf{s}\} \times L$ and any element of D can be represented by the form $(\mathbf{s}, \boldsymbol{l})$, where $\boldsymbol{l}$ labels the class. Here we use the one-hot encoded label for k classes, i.e., $\boldsymbol{l} \in L = \{\boldsymbol{l} : \boldsymbol{l} \in \{0, 1\}^k, \mathbf{1}^T\boldsymbol{l} = 1\}$.

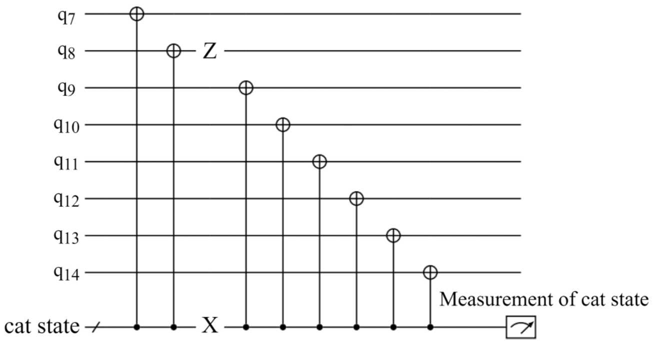

FIG. 7. For conversion circuit G_1, we first assume that there is no error in the input state. Then we consider the measurement of stabilizer $X_7X_8X_9X_{10}X_{11}X_{12}X_{13}X_{14}$ at $\mathcal{E}_1$. If a weight-2 error $X \otimes Z$ occurs after the second CNOT gate, with the premise that no error occurs in the rest of G_1, we would obtain a syndrome vector (00010000). Then we can take one of the following error-correction schemes: (1) We only refer to the lookup table of $\mathcal{E}_1$ and use the minimal weight decoding scheme to apply a recovery operator Z_0, which will cause an uncorrectable error Z_0Z_8, or (2) we still execute process (1), but after finish the application of G_1; with the syndrome (10000000) of $\mathcal{E}_2$ and the previous syndrome of $\mathcal{E}_1$, we can infer that a logical error $Z_0Z_7Z_8$ has occurred. The other seven physical qubits that belong to the logical qubit are idle, so they are not shown in this figure.

From Eq. (5), the output of our model functions needs to give the predicted value $(g_X(\mathbf{s}), g_Z(\mathbf{s}))$ based on the syndrome information obtained by the measurement. So we take different predicted values as classification labels corresponding to logical recovery operators and denote these labels by $\{\boldsymbol{l}_I, \boldsymbol{l}_X, \boldsymbol{l}_Y, \boldsymbol{l}_Z\}$.

The data we use to train the deep-network model are generated as follows.

(i) Fault-tolerantly prepare the CSS-7 code initial state $|\bar{\psi}\rangle_{CSS\text{-}7}$ and $|\mathcal{A}\rangle$, perform the leading error correction for the input state $|\bar{\psi}\rangle_{CSS\text{-}7} \otimes |\mathcal{A}\rangle$ before it is applied by G_i, and collect the error-correction syndrome $\mathbf{s}_{LEC}$, where its dimension is 14.

(ii) Apply G_1 to the input state and collect the syndrome vectors $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$ obtained during its application, where the dimension of $\mathbf{s}_1$ and $\mathbf{s}_2$ is 8 and that of $\mathbf{s}_3$ is 14.

(iii) Perform a round of noise-free projection after $\mathcal{E}_3$, which aims to make the uncorrectable error in the conversion circuit become a logical error, and then perform noise-free logical Z-based or X-based measurement to detect the logical error on the output state. Finally, we obtain the class label value.

We choose a particular physical error rate p and apply the depolarizing channel to G_1. All-zero data are excluded in the data set. Then we continue the simulation until $N = 2\times 10^7$ nonzero training data are gathered.

2. *Objective function*

In machine learning, the data set D can be seen as a set of points that are produced by the real function of the data model and our goal is to use neural networks to approximate this function. The neural network can be determined by the parameter $\boldsymbol{\omega}$ and then the problem of finding the optimal coefficients of the functions g_X and g_Z is transformed into finding the optimal parameter $\boldsymbol{\omega}$. The objective function $\mathcal{L}(D, \boldsymbol{\omega})$ usually is taken as the goal of optimization:

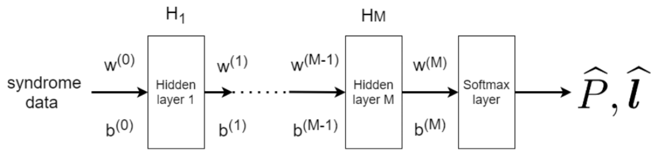

FIG. 8. Typical feedforward neural-network structure diagram.

$$\min_{\boldsymbol{\omega}} \mathcal{L}(D, \boldsymbol{\omega}). \tag{6}$$

The objective function is the quantification of the difference between the model output and the observation result. It can be described as

$$\mathcal{L}(D, \boldsymbol{\omega}) = -\sum_{\mathbf{s}} \sum_{j=1}^{k} \boldsymbol{l}_{\mathbf{s}}^{(j)} \ln \hat{P}_j(\mathbf{s}, \boldsymbol{\omega}) + \lambda \sum_{m=0}^{M} \|\boldsymbol{\omega}^{(m)}\|_2, \tag{7}$$

where $\boldsymbol{l}_{\mathbf{s}}^{(j)}$ is the binary indicator if the jth class label is the correct classification for the measured syndrome $\mathbf{s}$, M is the number of layers for the network, and $\hat{P}_j$ is the model predicted probability that $\mathbf{s}$ belong to the jth class. Our network structure is shown in Fig. 8. The syndrome vector $\mathbf{s}$ as the input data of the network is first sent to M hidden layers and the hidden layer function is

$$H_m = f(\omega^{(m-1)} H_{m-1} + b^{(m-1)}), \quad m = 1, \ldots, M. \tag{8}$$

Here we take the rectified linear unit as the active function f, i.e., $f(x) = \max\{0, x\}$. Then we pass the output of the last hidden layer to the softmax layer to calculate the probability that $\mathbf{s}$ belong to the jth category. We define $\hat{P} = (\hat{P}_1, \ldots, \hat{P}_k)$ and to obtain this adapt the equation

$$\hat{P}_j = \frac{e^{V_j}}{\sum_{j=1}^{k} e^{V_j}}, \quad j = 1, \ldots, k, \tag{9}$$

where $V = (V_1, \ldots, V_k) = \boldsymbol{\omega}^{(M)} H_M + b^{(M)}$. Finally, we predict the class label vector $\hat{\boldsymbol{l}} = (\hat{l}^{(1)}, \ldots, \hat{l}^{(k)})$ of $\mathbf{s}$ by

$$\hat{l}_{\mathbf{s}}^{(i)} = \begin{cases} 1 & \text{for } i = \operatorname*{argmax}_j \{\hat{P}_j\} \\ 0 & \text{otherwise.} \end{cases}$$

In contrast, reducing the empirical error usually makes the model overfit the training data and gives a more complicated model function. Such a case will greatly damage the generalization of the model and cause additional time consumption. Therefore, it is usually necessary to add an extra term to the objective function to reduce the structural error of the model; this term is called regularization. Commonly used regularization methods are $L1$ regularization and $L2$ regularization. Here we use $L2$ regularization in Eq. (7), which will limit the coefficient ω in the model function to reduce the complexity of the model. A hyperparameter λ is also introduced to balance the loss function and regularized weight.

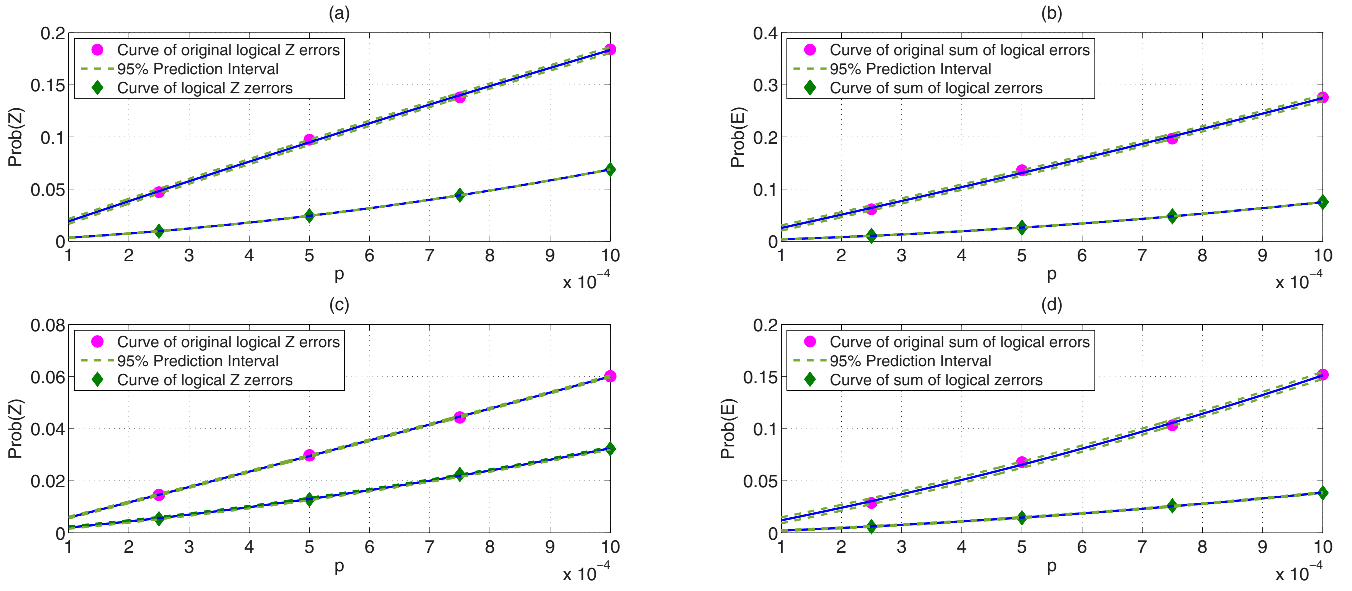

FIG. 9. (a) Simulation curve of the logical Pauli-Z error rate for G_1 and the corresponding curve for G_1 with adjusted decoder. (b) Simulation curve of the sum of logical Pauli errors rate for G_1 and the corresponding curve for G_1 with the adjusted decoder. (c) Simulation curve of the logical Pauli-Z error rate for G_2 and the corresponding curve for G_2 with the adjusted decoder. (d) Simulation curve of the sum of logical Pauli errors rate for G_2 and the corresponding curve for G_2 with the adjusted decoder.

For the conversion circuit G_1, we construct two binary classification models for the two types of data labeled $g_X(\mathbf{s})$ and $g_Z(\mathbf{s})$. We set the input layer with 44 dimensions, and the numbers of hidden layer nodes are 256, 512, 1024, and 256. We set the batch size as 10; the learning rate is 1×10^{-4}. We train our models on PYTORCH.

In order to test this neural-network-based decoder, we fix a sequence of physical error rates ranging from 10^{-4} to 10^{-3}. For each physical error rate p, we use our simulation scheme introduced in Sec. IV D to display the logical error rate of G_1 and G_2, as shown in Fig. 9. We compare the results with the decoder based on the lookup table. The simulation results show that our neural-network decoder can make the failure probability of the code conversion circuit lower.

D. Nonuniform concatenated 25-qubit code thresholds

We next analyze the performance of the 25-qubit logical H, CNOT, and T gates through simulation algorithms [42]. Our simulation experiments are executed by the platform called LIQUi$|\rangle$ [43].

For our numerical simulation scheme, we assess the performance of a fault-tolerant quantum operation in terms of a pseudothreshold and an asymptotic threshold. The pseudothreshold corresponds to the value ϵ_{th} such that when the physical error value satisfies $p < \epsilon_{th}$, the logical error rate of the fault-tolerant structure is lower than its corresponding physical operation value, that is, the physical operation can be effectively simulated by this fault-tolerant structure. In contrast, asymptotic thresholds correspond to the value ϵ_{asy} such that when the physical error value satisfies $p < \epsilon_{asy}$, the logical error rate can drop to any given accuracy if we keep increasing the concatenation level.

Our experiment has the following two basic assumptions. First, since the error rate in classical computers is usually very low (per operation), we assume that classical information can be ideally protected and do not consider errors in classical computers in our simulation experiments. Second, to ensure the full use of resources, we assume that in every single logical qubit, the auxiliary state that passes the verification can be reused before it is measured to be a classical bit.

We construct an error model for the 25-code noise threshold calculation as a depolarization noise model, that is, we apply the following noise channel to each physical component in a 1-exRec:

$$\varepsilon(\rho) = \left(1 - \frac{3p}{4}\right)\rho + \frac{p}{4}(X\rho X + Z\rho Z + Y\rho Y). \qquad (10)$$

The primary noise components in our simulation are listed in Table III. We only consider the case in which the physical error rates of a two-qubit gate and a one-qubit gate are the same.

Furthermore, following the decoding scheme of Ref. [21], we design an adjusted decoding procedure for the H 1-exRec and CNOT 1-exRec.

(i) Implement error correction on the three inner code blocks (logical qubits 0–2) of the 25-qubit concatenated code and record the error position and type of each code block.

TABLE III. Types of physical noisy components present in the 1-exRec considered.

No.	Component type
1	basic state $|0\rangle$ preparation
2	basic state $|+\rangle$ preparation
3	X-basis measurement or Z-basis measurement
4	two-qubit quantum gate
5	single-qubit quantum gate (including waiting gate)

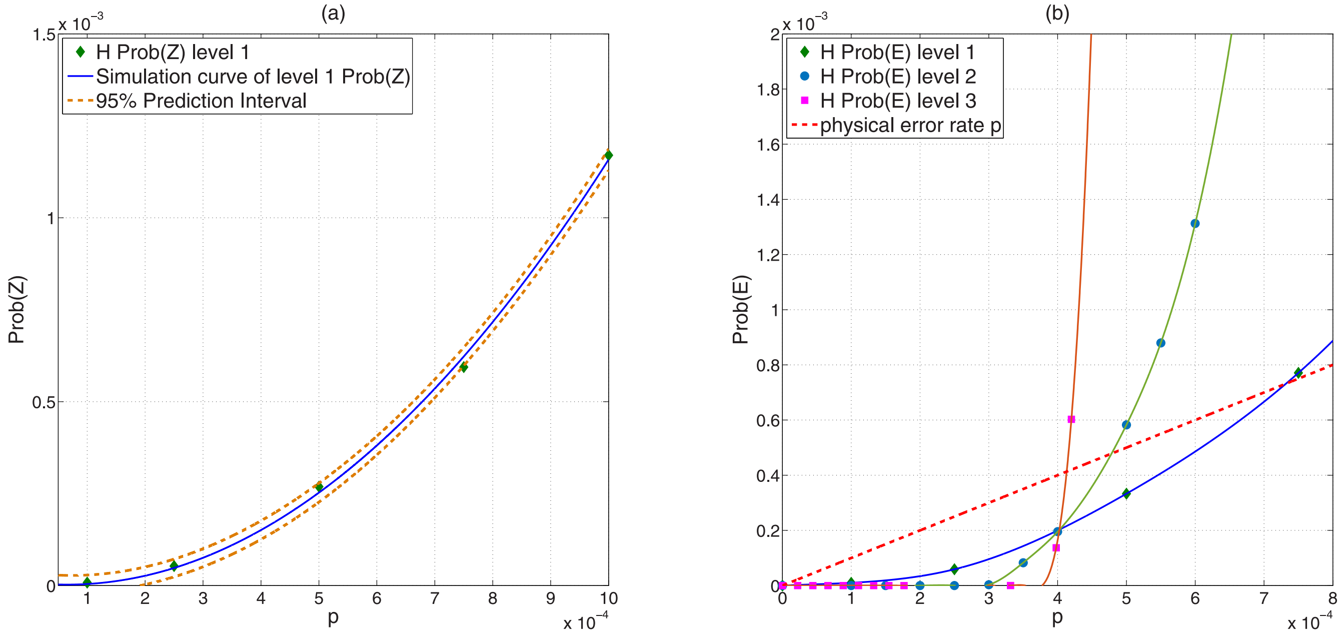

FIG. 10. (a) Rate of logical error Z for a 25-qubit H circuit under the standard depolarizing error model. The dashed lines represent the physical Z error rate of a single physical H gate. (b) Pseudothreshold for a 25-qubit H circuit under the standard depolarizing error model. The dashed line represents the physical error rate.

(ii) Perform error detection on the entire 25-qubit code to obtain a global syndrome vector.

(iii) Update the global syndrome vector based on the syndrome information from the three inner code blocks. For example, if the first two of the three inner code blocks detect a single-qubit Pauli-X-type error and independently perform the error correction, then during the global stabilizer measurement, a single logical X error is found in the third code block from the global syndrome vector. We point out that this is not the situation in which the third code block is error triggered, but rather two logical errors occur during the error-correction process of the first two code blocks. Therefore, the detected error is actually $\bar{X}_0\bar{X}_1$ rather than the complementary error $\bar{X}_2$ (where $\bar{X}_0\bar{X}_1\bar{X}_2$ is the logical X error on the 25-qubit code). Therefore, the global syndrome vector should be updated in this case.

(iv) If none of the inner code block stabilizer measurements are triggered, then perform a canonical global error correction.

With the decoding procedure and error model described above, we then design the following simulation scheme to calculate the logical failure rate for a 1-exRec.

(i) Given a 1-exRec, we fix a sequence of physical error rates, and for the physical error rate p, we fix an integer N, where N corresponds to the total number of iterations in which the depolarizing channel is applied to the 1-exRec and the ancillary block passes verification. Thus, we actually calculate the probability of the logical error rate conditioned on the acceptance of all ancillary states.

(ii) When a noisy logical 1-exRec fails, such as a CNOT, it applies the ideal CNOT gate followed by one of the 15 nontrivial two-qubit logical Pauli operators. Thus, for each possible logical error E, we prepare an appropriate initial encoded state and use the Monte Carlo method to estimate its conditional probability when the physical error rate is fixed. Here we set the iteration number $N = 10^6$ and define the logical error rate as $\mathcal{P}(E|\text{CNOT}, p) = n/N$, where n is the number of logical errors E occurring after N iterations.

For an effective fault-tolerant quantum operation, its logical error rate should be smaller than the error rate of the unprotected operation. It also has been proved that if the error propagation is limited and a good decoder is used, then the logical error rate should exhibit power-law scaling [44], implying that code concatenation techniques [21] can be adopted to exponentially reduce the logical error rate. Our simulation results have been collected and are presented in Table IV. Figures 10–12 show the simulation curves of the logical error.

E. Discussion

From our numerical simulation results, we get the pseudothreshold value of 4.08×10^{-4} for CNOT 1-exRec, 7.21×10^{-4} for H, and 3.93×10^{-4} for T. We think these are

TABLE IV. Pseudothreshold and asymptotic threshold results for the CNOT, H, and T gates of 105-, 49-, and 25-qubit codes. The results of the 105- and 49-qubit codes can be found in Ref. [21].

Code and gate	Pseudothreshold	Asymptotic threshold
105-qubit CNOT	2.11×10^{-3}	1.95×10^{-3}
105-qubit Hadamard	4.47×10^{-5}	1.28×10^{-3}
105-qubit T	4.89×10^{-4}	1.58×10^{-3}
49-qubit CNOT	1.21×10^{-3}	1.10×10^{-3}
49-qubit Hadamard	7.76×10^{-5}	9.69×10^{-4}
49-qubit T	4.18×10^{-4}	1.03×10^{-3}
25-qubit CNOT	4.08×10^{-4}	4.09×10^{-4}
25-qubit Hadamard	7.21×10^{-4}	4.05×10^{-4}
25-qubit T	3.93×10^{-4}	3.94×10^{-4}

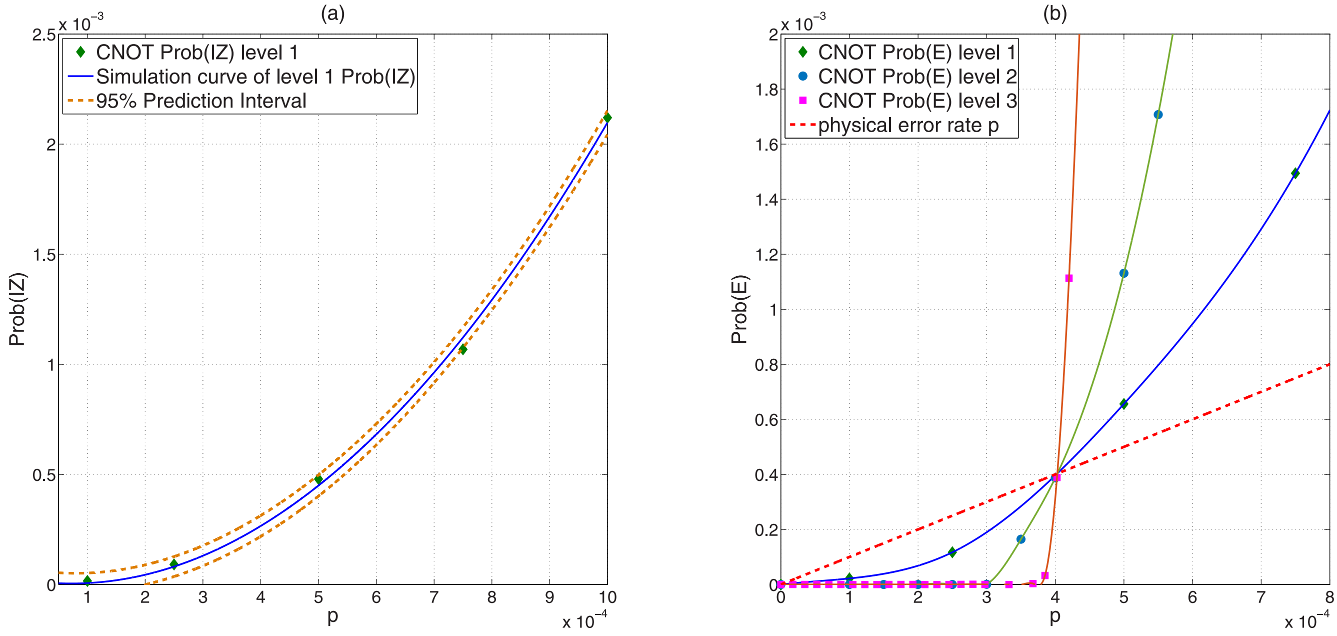

FIG. 11. (a) Rate of logical error IZ for a 25-qubit CNOT circuit under the standard depolarizing error model. The dashed lines represent the physical $I \otimes Z$ error rate of a single physical CNOT gate. (b) Pseudothreshold for a 25-qubit CNOT circuit under the standard depolarizing error model. The dashed line represents the physical error rate.

reasonable results. First, for the error correction based on the method shown in Fig. 4, its auxiliary state preparation only includes a single verification process, which may make some verified auxiliary states introduce new errors to the data state. For example, after verifying the encoded state $|\bar{0}\rangle$ with no Pauli-X error, we then immediately use it to detect possible Pauli-Z error in the data state. However, if there already exists a Z error in the auxiliary state and there is no Z error in the data state, then the measurement of the ancillary block would produce wrong syndrome information. Therefore, the upper bound of the pseudothreshold of our universal scheme is reduced. Second, for the 25-qubit code, the Clifford quantum gate circuits are transversal on the inner and outer codes, so a single-qubit error introduced by the ancillary state will not spread in encoded information, which effectively improves the pseudothreshold of the H 1-exRec.

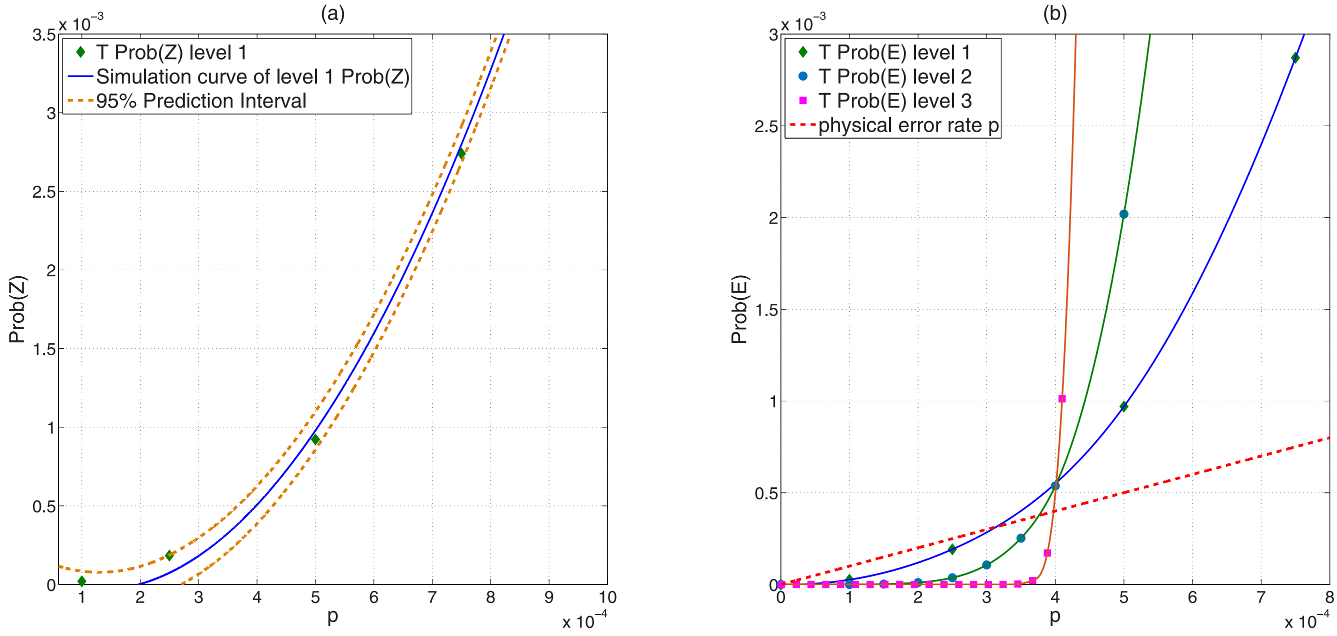

FIG. 12. (a) Rate of logical error Z for a 25-qubit T circuit under the standard depolarizing error model. The dashed lines represent the physical Z error rate of a single physical T gate. (b) Pseudothreshold for a 25-qubit T circuit under the standard depolarizing error model. The dashed line represents the physical error rate.

Adding more verification processes can effectively reduce the possibility of errors introduced by the auxiliary state, but it also significantly increases the scale of the auxiliary preparation circuit and reduces the magnitude of the decrease of the logical error rate as the physical error rate decreases.

Our decoding strategy effectively reduces the logical error rate of the T gate, which also implies that the neural-network-based decoding algorithm can effectively improve the antinoise ability of the pieceable fault-tolerant circuit. An efficient but higher-resource-consumption code conversion circuit [31] with our neural-network decoding scheme might be taken as an alternative to be applied in our logical T gate.

V. ESTIMATION OF PHYSICAL QUBIT RESOURCES FOR UNIVERSAL QUANTUM COMPUTING

In the numerical simulation of the H, CNOT, and T gates, we want to estimate the qubit resource consumption when our logical gate error rate is less than a specified accuracy. In addition, we prove that our scheme to implement the fault-tolerant T gate consumes fewer qubit resources than the 49-qubit concatenation code. Here we use the following measure, also called raw qubit overhead [21], which is given to measure how many physical qubits are needed in the logical circuit with a logical error rate less than a given accuracy. Recalling the settings in our simulation experiments, we first assume that only when the auxiliary block passes verification can it be accepted for error detection. Therefore, the ancillary state preparation is a majority selection process that is performed asynchronously and it can be repeated until a passed ancillary block is obtained. This makes the number of qubits used to prepare an accepted block obey the binomial distribution. So the qubit overhead of a k-Rec can be estimated by calculating the expected value of the auxiliary state resources. Second, following the method in Ref. [21], we also assume that the auxiliary state that passes the verification can be reused before it is measured to be a classical bit.

We argue that the above two basic assumptions are also adopted for the resource analysis of the 105- and 49-qubit codes, so we can fairly compare resources under the same calculation method. Finally, in the kth encoding level, the TEC process of a k-Rec is also the LEC process of the next one. These overlapping error-correction circuits should be noted so that repeated resource estimation can be avoided.

For Steane's method, there are two types of auxiliary state preparation processes, as shown in Fig. 4, so the auxiliary block acceptance probability should be introduced to facilitate the subsequent calculation of resource consumption. We consider the level-k n-qubit concatenation code and define $P^{(k)}_{\overline{|0\rangle}}$ as the probability that the level-k auxiliary encoded state $\overline{|0\rangle}$ passes verification, while $P^{(k)}_{\overline{|+\rangle}}$ corresponds to the encoded state $\overline{|+\rangle}$ and $N^{(k)}_L$ corresponds to the number of physical qubits comprising an level-k encoded state. These two probabilities can be estimated by the Monte Carlo method. Therefore, the mathematical expectation of the auxiliary state qubit resource consumption used in the level-k error correction of a code block is

$$N^{(k)}_{\overline{|0\rangle}} = \sum_{m=1}^{\infty} 2mN^{(k-1)}_L P^{(k)}_{\overline{|0\rangle}} \big(1 - P^{(k)}_{\overline{|0\rangle}}\big)^m = \frac{2N^{(k-1)}_L}{P^{(k)}_{\overline{|0\rangle}}}, \tag{11}$$

$$N^{(k)}_{\overline{|+\rangle}} = \sum_{m=1}^{\infty} 2mN^{(k-1)}_L P^{(k)}_{\overline{|+\rangle}} \big(1 - P^{(k)}_{\overline{|+\rangle}}\big)^m = \frac{2N^{(k-1)}_L}{P^{(k)}_{\overline{|+\rangle}}}. \tag{12}$$

So we can get the total number of auxiliary quantum states required to implement a level-k fault-tolerant logical qubit as

$$\begin{aligned} N^{(k)}_L &= N^{(k)}_{\mathcal{E}} + n^k & (13)\\ &= N^{(k)}_{\overline{|0\rangle}} + N^{(k)}_{\overline{|+\rangle}} + n^k, & (14)\end{aligned}$$

where we have set $N^{(0)}_L = 1$.

According to the above recursive calculation method, we can obtain the level-k quantum state resource estimation of the transversal H gate and CNOT gate on the n-qubit concatenation quantum code:

$$N^{(k)}_H = N^{(k)}_{\mathcal{E}} + n^k, \tag{15}$$

$$N^{(k)}_{\text{CNOT}} = 2N^{(k)}_{\mathcal{E}} + 2n^k. \tag{16}$$

We then provide an estimation of the raw qubit overhead for the T gate. We first propose a method for computing the consumption of ancillary qubits of a pieceable fault-tolerant logical gate $\tilde{\mathcal{C}}$. According to Eq. (2), we should note that for this kind of circuit, the auxiliary qubit consumption values of $\mathcal{E}_i$ $(i = 1, 2, \ldots, r)$ may be different, such as the controlled-Z gate in the five-qubit code [31]. The auxiliary states used in each of the $\mathcal{E}_i$ are restored to classical bits. So the ancillary resources required by $\tilde{\mathcal{C}}$ are actually the sum of auxiliary resources consumed by each circuit piece. Consequently, the raw qubit overhead needed for error correction for the entire $\tilde{\mathcal{C}}$ at level-k encoding hierarchy can be defined as

$$N^{(k)}_{\tilde{\mathcal{C}}} = \sum_{i=1}^{r} N^{(k)}_{\mathcal{E}_i} + n^k. \tag{17}$$

Then, for the 25-qubit pieceable fault-tolerant T gate, we first calculate the level-1 ancillary resource of the error-correction procedure in each of the five stages.

The first stage is a sequence of two CNOT 1-Rec in the inner CSS-7 code. We use the Steane method for error detection and similarly define $P^{(1)}_{\overline{|0\rangle}_{\text{inner1}}}$ as the probability that the level-1 auxiliary inner-layer encoded state $\overline{|0\rangle}$ passes verification, etc. So we give the ancillary qubit overhead as

$$N^{(1)}_{\text{stage1}} = 4 \times 2n_{\text{inner1}} N^{(0)}_L \left(\frac{1}{P^{(1)}_{\overline{|0\rangle}_{\text{inner1}}}} + \frac{1}{P^{(1)}_{\overline{|+\rangle}_{\text{inner1}}}} \right). \tag{18}$$

Here we use n_{inner1} to denote the number of qubits needed for the first inner code; for our T gate, we set $n_{\text{inner1}} = 7$.

The second stage mainly includes the conversion process G_1 in the 1th logical qubit. Because the 0th and 2th logical qubits are not applied in any quantum operation until the fifth stage, we delay the error-correction process of these idle inner code blocks until the final stage. The ancillary consumption

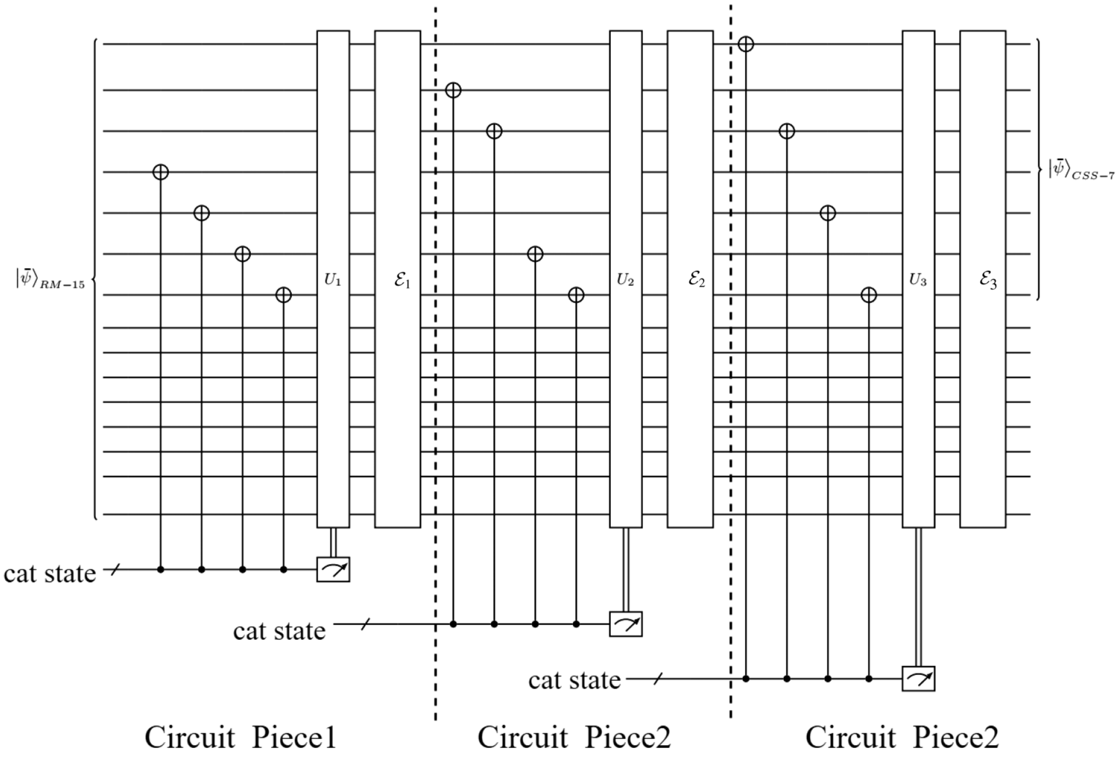

FIG. 13. Subcircuit G_2 that realizes fault-tolerant conversion from RM-15 code to CSS-7 code. We can divided this circuit into three pieces: (1) Project the initial code state with stabilizer $X_3X_4X_5X_6$ and perform a round of error correction. The operator $U_1 = Z_2Z_6Z_{10}Z_{14}$. (2) Project the code state with stabilizer $X_1X_2X_5X_6$ and perform a round of error correction. The operator $U_2 = Z_4Z_6Z_{12}Z_{14}$. (3) Project the code state with stabilizer $X_0X_2X_4X_6$ and perform a round of error correction. The operator $U_3 = Z_5Z_6Z_{13}Z_{14}$. Table II shows the stabilizers used in $\mathcal{E}_1$ and $\mathcal{E}_2$. After $\mathcal{E}_3$ we obtain the output state $|\bar{\psi}\rangle_{CSS\text{-}7} \otimes |\mathcal{A}\rangle$.

of G_1 is due to two processes: the fault-tolerant preparation of ancillary state $|\mathcal{A}\rangle$ and projection of three weight-4 stabilizers. We use Shor's method in $\mathcal{E}_1$ and $\mathcal{E}_2$, and after the fault-tolerant measurement of the last stabilizer $Z_2Z_6Z_{10}Z_{14}$, we can use Steane's method in $\mathcal{E}_3$. We denote the auxiliary resource consumption of the above two processes by N_{slice1} and N_{slice2}, so the entire ancillary qubit overhead is given by

$$N^{(1)}_{\text{stage2}} = N^{(1)}_{\text{slice1}} + N^{(1)}_{\text{slice2}}, \tag{19}$$

where

$$\begin{aligned} N^{(1)}_{\text{slice1}} = &\left[2n_{\text{inner1}}N^{(0)}_L\left(\frac{1}{P^{(1)}_{\overline{|0\rangle}_{\text{inner1}}}} + \frac{1}{P^{(1)}_{\overline{|+\rangle}_{\text{inner1}}}}\right)\right.\\ &\left. + n_{\text{inner1}}N^{(0)}_L\right]\frac{1}{1-\epsilon^{(1)}_{\overline{|+\rangle}_{\text{inner1}}}} + N^{(0)}_L\\ &+ 8\frac{8N^{(0)}_L + 7N^{(0)}_L}{P^{(1)}_{\text{weight-8 cat state}}}, \end{aligned}\tag{20}$$

$$\begin{aligned} N^{(1)}_{\text{slice2}} = &\,3\left(\frac{4N^{(0)}_L + 3N^{(0)}_L}{P^{(1)}_{weight\text{-}4\ cat\ state}}\right) + 2\left(8\frac{8N^{(0)}_L + 7N^{(0)}_L}{P^{(1)}_{weight\text{-}8\ cat\ state}}\right)\\ &+ 2n_{\text{inner2}}N^{(0)}_L\left(\frac{1}{P^{(1)}_{\overline{|0\rangle}_{\text{inner2}}}} + \frac{1}{P^{(1)}_{\overline{|+\rangle}_{\text{inner2}}}}\right). \end{aligned}\tag{21}$$

Here we use $P^{(1)}_{\text{weight-t cat state}}$ to denote the probability that a level-1 weight-t cat state passes verification, $\epsilon^{(1)}_{\overline{|+\rangle}_{\text{inner1}}}$ to denote the probability that a level-1 CSS-7 $\overline{|+\rangle}$ state is unsuccessfully prepared, and n_{inner2} to denote the number of qubits needed for the second inner-layer code; for our example, we set $n_{\text{inner2}} = 15$.

The third stage only includes a transversal T gate on the 1th logical qubit with Steane's error-correction method, so we have

$$N^{(1)}_{\text{stage3}} = 2n_{\text{inner2}}N^{(0)}_L\left(\frac{1}{P^{(1)}_{\overline{|0\rangle}_{\text{inner2}}}} + \frac{1}{P^{(1)}_{\overline{|+\rangle}_{\text{inner2}}}}\right). \tag{22}$$

The fourth stage includes the conversion process G_2 in the 1th logical qubit and the waiting process of the 0th and 2th logical qubits. Similar to the second stage, we only consider the auxiliary quantum state resources consumed by G_2. In the process of converting the RM-15 code to the CSS-7 code, we only need to perform three fault-tolerant stabilizer measurements to obtain the state $|\bar{\psi}\rangle_{CSS\text{-}7} \otimes |\mathcal{A}\rangle$. After the implementation of G_2, we can use Steane's method to correct the errors in $|\bar{\psi}\rangle_{CSS\text{-}7}$, as shown in Fig. 13. For G_2, we include the ancillary consumption of $\mathcal{E}_3$ in the fifth stage. So we have

$$N^{(1)}_{\text{stage4}} = 3\frac{4N^{(0)}_L + 3N^{(0)}_L}{P^{(1)}_{\text{weight-4 cat state}}} + 2\frac{8N^{(0)}_L + 7N^{(0)}_L}{P^{(1)}_{\text{weight-8 cat state}}}. \tag{23}$$

The fifth stage is also a sequence of two CNOT 1-Rec on the inner CSS-7 code. We put the process $\mathcal{E}_3$ of G_2 in this stage, which makes the number of error-correction processes of the inner code 4, so the ancillary consumption of this stage is the same as the first stage:

$$N^{(1)}_{\mathrm{stage5}} = 4\times 2n_{\mathrm{inner1}}N^{(0)}_L\left(\frac{1}{P^{(1)}_{\overline{|0\rangle}_{\mathrm{inner1}}}}+\frac{1}{P^{(1)}_{\overline{|+\rangle}_{\mathrm{inner1}}}}\right). \tag{24}$$

Finally, we obtain the auxiliary state resource consumption of all the inner logical gate circuits of the pieceable fault-tolerant T gate. In addition, we still have to apply global error correction of the 25-qubit code after the application of T gates. In this process, we use the Steane method. From Eq. (13) we can obtain the auxiliary state resource consumption of the level-1 25-qubit T as follows:

$$N^{(1)}_T = \sum_{j=1}^{5} N^{(1)}_{\mathrm{stage_j}} + N^{(1)}_{\mathcal{E}} + n. \tag{25}$$

Then the level-k raw qubit overhead estimation of the T gate on the 25-qubit concatenation quantum code is

$$N^{(k)}_T = \sum_{j=1}^{5} N^{(k)}_{\mathrm{stage_j}} + N^{(k)}_{\mathcal{E}} + n^k. \tag{26}$$

To compare the qubit consumption of the 25- and 49-qubit T gates, we first estimate the resource upper bound of the 25-qubit T gate and give the lower bound of resources of the 49-qubit T gate. Then we illustrate that our construction costs fewer resources than the 49-qubit code.

We denote by $P^{(k)}_{\mathcal{A}}$ the probability that a level-k ancillary block is accepted and by $N^{(k)}_{\mathcal{A}}$ the number of $(k-1)$-exRec in the preparation and verification circuits of this ancillary block. For a level-k ancilla to be rejected, the previous preparation and verification circuits must contain at least one fault $(k-1)$-exRec, but not all fault components in the preparation circuit will cause the auxiliary state to be rejected. Therefore, we can give the upper and lower bounds of $P^{(k)}_{\mathcal{A}}$ as

$$1-\epsilon^{(k)} \geqslant P^{(k)}_{\mathcal{A}} \geqslant 1-N^{(k)}_{\mathcal{A}}\epsilon^{(k)}, \tag{27}$$

where $\epsilon^{(k)}$ is the upper bound of the failure rate for all level-$(k-1)$ components.

Following Eq. (27), we get the upper bound of the qubit overhead of the level-k 25-qubit T gate as $1057N^{(k-1)}_{25,L}+25^k$ and the lower bound of the corresponding 49-qubit T gate is $1112N^{(k-1)}_{49,L}+49^k$; here we use the notation $N^{(k-1)}_{25,L}$ to denote the qubit overhead of a level-k logical qubit with the 25-qubit code as its top encoding layer, similar to $N^{(k-1)}_{49,L}$. So we claim that our qubit resource overhead of the T gate is lower than that in the 49-qubit code. The details of our proof are left to Appendix B.

Because the physical error rate of a single quantum operation on a general quantum physical computing experimental platform is usually 10^{-3}–10^{-4} and the error rate of a classical computer is generally less than 10^{-15}, we calculated the qubit resources consumed by the logical circuit under this constraint and show the results in Table V.

TABLE V. Overhead estimation for the 105-, 49-, and 25-qubit codes at concatenation level 3; the estimation of 105- and 49-qubit codes can be found in Ref. [21]. The first column indicates the code and corresponding logical gate for which the overhead is computed. The second column indicates the largest physical error rate that should be achieved such that the logical error rate is below $p = 10^{-15}$. The third column gives the qubit overhead for the given logical error rate.

Code and gate	Physical error rate	Qubit overhead
105-qubit CNOT	1.39×10^{-3}	6.01×10^{9}
105-qubit Hadamard	1.60×10^{-5}	3.00×10^{9}
49-qubit CNOT	3.92×10^{-4}	1.94×10^{8}
49-qubit Hadamard	8.47×10^{-5}	7.33×10^{7}
25-qubit CNOT	2.78×10^{-4}	2.54×10^{6}
25-qubit Hadamard	3.46×10^{-4}	1.27×10^{6}

VI. CONCLUSION

In this paper, the code conversion technique was adopted to optimize the resource overhead of a universal concatenated scheme. In addition, a neural-network-based decoder algorithm was proposed to improve the performance of a logical circuit with pieceable fault-tolerant protocol.

Following the simulation scheme of calculating the failure rate of a logical circuit by the Monte Carlo method in Ref. [21], we designed the same experimental process and calculated the lower bound of a pseudothreshold for a logical universal gate set in a 25-qubit nonuniform concatenated code and found it to be 3.93×10^{-4}. By utilizing the CSS code structure of the 7- and 15-qubit codes, the estimation of the qubit overhead of multilevel concatenation can be simplified as a recursive computation process. Different from the existing concatenation approaches, such as 49- and 105-qubit codes, which exhibit high overhead costs, we obtained a raw qubit overhead for our level-3 non-Clifford T gate of 1.34×10^{9} when the logical error rate was below 10^{-15}; the qubit overhead for the Clifford gate under a given logical accuracy was also greatly reduced, as shown in Table V.

As realistic error models become increasingly more relevant, the development of environment-specific fault-tolerant logical gates will become increasingly important. How to optimize the decoder to effectively suppress the propagation of errors under a complex error model is indeed a challenge. In addition, optimization of ancillary resources and measurement times must be achieved to realize large-scale quantum computation. A natural direction of future work is to study a better decoding method and combine with a synthesis algorithm to obtain more practical fault-tolerant logical non-Clifford circuits.

ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China under Grant No. 2018YFA0306703 and the National Natural Science Foundation of China under Grant No. 61572109.

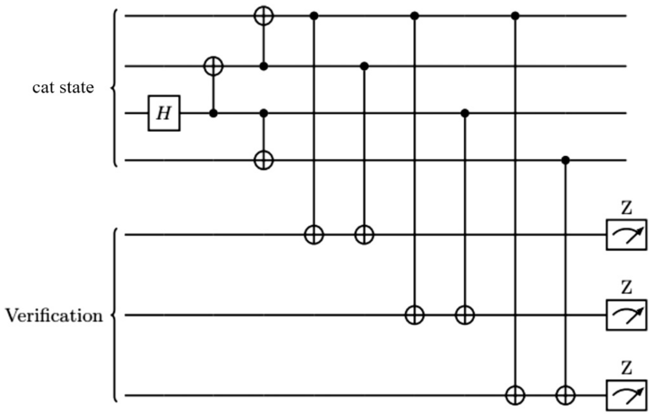

FIG. 14. Preparing weight-4 cat state $\frac{|0000\rangle+|1111\rangle}{\sqrt{2}}$. Note that all seven physical qubits are initialized as $|0\rangle$.

APPENDIX A: CONVERSION CIRCUITS AND ENCODING CIRCUITS

We describe the conversion circuit G_2 in Fig. 13. The auxiliary state used in stabilizer projection is a weight-4 cat state, as shown in Fig. 14. The encoding circuits of the 25-qubit code used in our simulation experiment are shown in Figs. 15 and 16.

APPENDIX B: BOUND OF QUBIT RESOURCES OF The T GATE FOR 25- AND 49-QUBIT CODEs

According to Eq. (27), we first estimate the upper bound of qubit resources for every stage of the level-k 25-qubit T gate as follows:

$$N_{\text{stage1}}^{(k)} \leqslant \frac{112N_{25,L}^{(k-1)}}{1-59\epsilon^{(k-1)}},$$

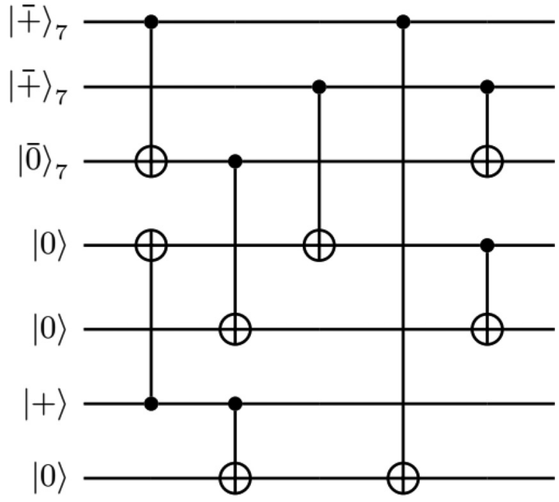

FIG. 15. Encoding $|\bar{0}\rangle$ circuits for the nonuniform 25-qubit concatenation code, where only qubits 0, 1, and 2 are replaced with the CSS-7 logical qubit.

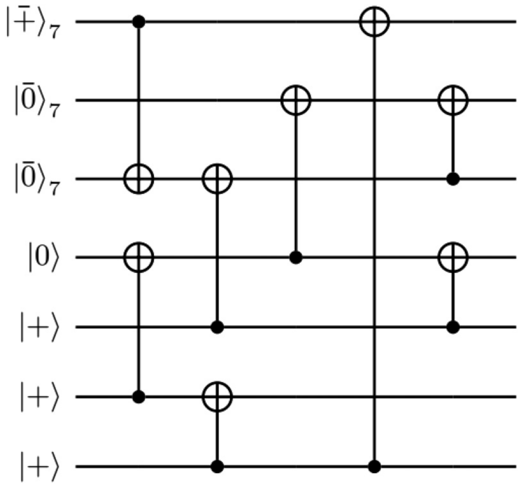

FIG. 16. Encoding $|\bar{+}\rangle$ circuits for the nonuniform 25-qubit concatenation code, where only qubits 0, 1, and 2 are replaced with the CSS-7 logical qubit.

$$\begin{aligned} N_{\text{stage2}}^{(k)} \leqslant & \left[\frac{\frac{28}{1-59\epsilon^{(k-1)}}+7}{1-\epsilon_{|\bar{+}\rangle_{\text{inner1}}}^{(k-1)}} + 1 + \frac{120}{1-56\epsilon^{(k-1)}} \right. \\ & + \frac{21}{1-19\epsilon^{(k-1)}} + \frac{240}{1-56\epsilon^{(k-1)}} \\ & \left. + \frac{60}{1-72\epsilon^{(k-1)}} \right] N_{25,L}^{(k-1)}, \\ N_{\text{stage3}}^{(k)} \leqslant & \frac{60N_{25,L}^{(k-1)}}{1-72\epsilon^{(k-1)}}, \\ N_{\text{stage4}}^{(k)} \leqslant & \frac{21}{1-19\epsilon^{(k-1)}} + \frac{240}{1-56\epsilon^{(k-1)}}, \\ N_{\text{stage5}}^{(k)} \leqslant & \frac{112N_{25,L}^{(k-1)}}{1-59\epsilon^{(k-1)}}. \end{aligned}$$

In fact, the pseudothreshold of several concatenation codes is usually on the order of 10^{-4}. Therefore, when we only consider the validity of the encoded logical circuit, the $\epsilon^{(k-1)}$ in the above resource upper bound estimation can be strictly smaller than 10^{-3}, so we can get

$$N_{25,T}^{(k)} \leqslant 1057N_{25,L}^{(k-1)} + 25^k.$$

For the 49-qubit level-k T gate [21], we use the same method to estimate its lower bound of qubit resources. Actually, with Eq. (27), we give the computation process as

$$N_{49,T}^{(k)} = N_{\text{inner }\mathcal{E}}^{(k)} + N_{\mathcal{E}}^{(k)} + 49^k.$$

Because the 49-qubit concatenation code only adapts RM-15 as its inner layer code, all its inner error correction can use Steane's method. Following the ancillary verification process, we have

$$N^{(k)}_{\text{inner }\mathcal{E}} = 6\left[\frac{2n_{\text{inner}}N^{(k-1)}_{49,L}\left(\frac{1}{P^{(k)}_{\overline{|0\rangle}^1_{\text{inner}}}}+\frac{1}{P^{(k)}_{\overline{|0\rangle}^2_{\text{inner}}}}\right)}{P^{(k)}_{\overline{|0\rangle}^3_{\text{inner}}}} + \frac{2n_{\text{inner}}N^{(k)}_{49,L}\left(\frac{1}{P^{(k)}_{\overline{|+\rangle}^1_{\text{inner}}}}+\frac{1}{P^{(k)}_{\overline{|+\rangle}^2_{\text{inner}}}}\right)}{P^{(k)}_{\overline{|+\rangle}^3_{\text{inner}}}}\right] \geqslant \left(\frac{360}{P^{(k)}_{\overline{|0\rangle}^3_{\text{inner}}}}+\frac{360}{P^{(k)}_{\overline{|+\rangle}^3_{\text{inner}}}}\right)N^{(k-1)}_{49,L} \geqslant 720N^{(k-1)}_{49,L}$$

and

$$N^{(k)}_{\mathcal{E}} = 2\times 49\left[\frac{\frac{1}{P^{(k)}_{\overline{|0\rangle}^1_{\text{global}}}}+\frac{1}{P^{(k)}_{\overline{|0\rangle}^2_{\text{global}}}}}{P^{(k)}_{\overline{|0\rangle}^3_{\text{global}}}} + \frac{\frac{1}{P^{(k)}_{\overline{|+\rangle}^1_{\text{global}}}}+\frac{1}{P^{(k)}_{\overline{|+\rangle}^2_{\text{global}}}}}{P^{(k)}_{\overline{|+\rangle}^3_{\text{global}}}}\right]N^{(k-1)}_{49,L} \geqslant 392N^{(k-1)}_{49,L}.$$

So the lower bound of the corresponding 49-qubit T gate is $1112N^{(k-1)}_{49,L} + 49^k$.

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

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[4] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[5] C. Neill, P. Roushan, K. Kechedzhi, S. Boixo, S. V. Isakov, V. Smelyanskiy, A. Megrant, B. Chiaro, A. Dunsworth, K. Arya *et al.*, A blueprint for demonstrating quantum supremacy with superconducting qubits, Science **360**, 195 (2018).

[6] H. Häffner, C. F. Roos, and R. Blatt, Quantum computing with trapped ions, Phys. Rep. **469**, 155 (2008).

[7] 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).

[8] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Demonstration of a small programmable quantum computer with atomic qubits, Nature (London) **536**, 63 (2016).

[9] M. H. Devoret and R. J. Schoelkopf, Superconducting circuits for quantum information: An outlook, Science **339**, 1169 (2013).

[10] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM J. Comput. **46**, 1920 (2017).

[11] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Characterizing quantum supremacy in near-term devices, Nat. Phys. **14**, 595 (2018).

[12] P. W. Shor, in *Proceedings of the 37th Conference on Foundations of Computer Science, Burlington, 1996* (IEEE, Piscataway, 1996), pp. 56–65.

[13] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Inf. Comput. **6**, 97 (2006).

[14] D. Gottesman, Quantum information science and its contributions to mathematics, in *Proceedings of Symposia in Applied Mathematics* (AMS, Providence, 2010), Vol. 68, pp. 13–58.

[15] B. Eastin and E. Knill, Restrictions on Transversal Encoded Quantum Gate Sets, Phys. Rev. Lett. **102**, 110502 (2009).

[16] B. Zeng, A. Cross, and I. L. Chuang, Transversality versus universality for additive quantum codes, IEEE Trans. Inf. Theory **57**, 6272 (2011).

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

[18] A. G. Fowler, S. J. Devitt, and C. Jones, Surface code implementation of block code state distillation, Sci. Rep. **3**, 1939 (2013).

[19] J. O'Gorman and E. T. Campbell, Quantum computation with realistic magic-state factories, Phys. Rev. A **95**, 032338 (2017).

[20] T. Jochym-O'Connor and R. Laflamme, Using Concatenated Quantum Codes for Universal Fault-Tolerant Quantum Gates, Phys. Rev. Lett. **112**, 010505 (2014).

[21] C. Chamberland, T. Jochym-O'Connor, and R. Laflamme, Overhead analysis of universal concatenated quantum codes, Phys. Rev. A **95**, 022313 (2017).

[22] R. Chao and B. W. Reichardt, Fault-tolerant quantum computation with few qubits, npj Quantum Inf. **4**, 42 (2018).

[23] C. Chamberland and P. Ronagh, Deep neural decoders for near term fault-tolerant experiments, Quantum Sci. Technol. **3**, 044002 (2018).

[24] E. Nikahd, M. Sedighi, and M. Saheb Zamani, Nonuniform code concatenation for universal fault-tolerant quantum computing, Phys. Rev. A **96**, 032337 (2017).

[25] V. Kliuchnikov, D. Maslov, and M. Mosca, Fast and efficient exact synthesis of single-qubit unitaries generated by clifford and T gates, Quantum Inf. Comput. **13**, 607 (2013).

[26] M. Amy, D. Maslov, and M. Mosca, Polynomial-time T-depth optimization of Clifford+T circuits via matroid partitioning, IEEE Trans. Comput.-Aid. Des. **33**, 1476 (2014).

[27] A. M. Steane, Enlargement of Calderbank-Shor-Steane quantum codes, IEEE Trans. Inf. Theory **45**, 2492 (1999).

[28] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).

[29] H. Bombin, Clifford gates by code deformation, New J. Phys. **13**, 043005 (2011).

[30] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, Fault-Tolerant Conversion Between the Steane and Reed-Muller Quantum Codes, Phys. Rev. Lett. **113**, 080501 (2014).

[31] T. J. Yoder, R. Takagi, and I. L. Chuang, Universal Fault-Tolerant Gates on Concatenated Stabilizer Codes, Phys. Rev. X **6**, 031039 (2016).

[32] K. R. Colladay and E. J. Mueller, Rewiring stabilizer codes, New J. Phys. **20**, 083030 (2018).

[33] D.-X. Quan, L.-L. Zhu, C.-X. Pei, and B. C. Sanders, Fault-tolerant conversion between adjacent Reed-Muller quantum codes based on gauge fixing, J. Phys. A: Math. Theor. **51**, 115305 (2018).

[34] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits, IEEE Trans. Comput.-Aid. Des. **32**, 818 (2013).

[35] D. Maslov, Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization, Phys. Rev. A **93**, 022311 (2016).

[36] M. Amy and M. Mosca, T-count optimization and Reed-Muller codes, IEEE Trans. Inf. Theory **65**, 4771 (2019).

[37] A. M. Steane, Quantum Reed-Muller codes, IEEE Trans. Inf. Theory **45**, 1701 (1999).

[38] R. Sweke, M. Kesselring, E. van Nieuwenburg, and J. Eisert, Reinforcement learning decoders for fault-tolerant quantum computation, Mach. Learn.: Sci. Technol. (2020).

[39] G. Torlai and R. G. Melko, Neural Decoder for Topological Codes, Phys. Rev. Lett. **119**, 030501 (2017).

[40] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer, New York, 2006).

[41] Y. Lecun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[42] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Phys. Rev. A **70**, 052328 (2004).

[43] D. Wecker and K. M. Svore, LIQUi$|\rangle$: A software design architecture and domain-specific language for quantum computing, arXiv:1402.4467.

[44] 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).