Universal non-uniform concatenated code with low error diffusion based on pieceable fault-tolerant protocol

Chen Lin  
School of Cybersecurity, Chengdu University of Information Technology, Chengdu, Sichuan, 610225 and  
Advanced Cryptography and System Security Key Laboratory of Sichuan Province

Shibin Zhang  
School of Cybersecurity, Chengdu University of Information Technology, Chengdu, Sichuan, 610225

Xiaoyu Song and Marek. A. Perkowski  
Department of Electrical and Computer Engineering, Portland State University

Guowu Yang  
Big Data Research Center, School of Computer Science and Engineering,  
University of Electronic Science and Technology of China, Chengdu, 611731, P.R. China  
(Dated: April 12, 2022)

Quantum concatenation code is a convenient way to realize fault-tolerant universal computing in contrast with magic state distillation, but there are many non-fault-tolerant logical locations in the lowest encoding level of such a code, which thereby increases the possibility of error multiplication and limits its capability to realize a universal gate library. In this work, we first propose a general framework based on machine learning technologies for the decoder design of a pieceable fault-tolerant quantum circuit. Then following our design principles, we adopt the neural-network algorithm to give an optimized decoder for such a fault-tolerant circuit. To assess the effectiveness of our new decoder, we adopt it into the pieceable fault-tolerant logical controlled-NOT gates which act on the Steane-7 qubit encoded state and the Reed-Muller 15 qubit encoded state; then we simulate these circuits under depolarizing noise model and compare the gate error thresholds in contrast to the minimal-weight decoder. Finally, we provide a fault-tolerant universal gate library based on a 33-qubit non-uniform concatenated code. We introduce several level-1 pieceable fault-tolerant locations with optimized decoders to construct the fault-tolerant non-Clifford gate on this code, which effectively reduces the number of time steps for implementing this gate. Meanwhile, we analyze the pseudo-threshold of our universal scheme.

I. INTRODUCTION

In recent years, quantum error correction codes have been proven to be useful in realizing a reliable and scalable quantum computer [1–5]. Specifically, quantum information is stored in a so-called logical qubit through an encoding circuit, such method ensures that the information can be protected from environmental noise and decoherence effects by introducing periodic error corrections [6–8]. On the other hand, the logical elementary gates designed in a fault-tolerant manner can form any large-scale reliable circuit, so that the accuracy of the calculation result is guaranteed under the condition that the physical error is lower than a certain error threshold [9][12].

With the rapid development of noisy intermediate scalable quantum (NISQ) devices [13][16], designing a practical high accuracy universal logical gate library with low qubit overhead has been becoming increasingly important in the research field of quantum fault-tolerant computing. However, due to the limitation of the no-go theorem [17][18], existing schemes for implementing fault-tolerant universal computing usually require a large number of quantum state resources, such as preparation and distillation of the magic state $V|+\rangle$ for the non-Clifford gate $V$. Combined with some transversal Clifford operations, such a state can be taken to implement a fault-tolerant logical gate $V$ [19][21]. Meanwhile, some researchers have adopted the code concatenation method to achieve a universal gate library by two error correction codes that have complementary transversal logical gates. Due to the sacrifice of the fault tolerance of some of its low-level logical locations, this scheme still has the defect of low anti-noise ability [22][24].

In particular, by appropriately dividing a non-transversal logical circuit into several pieces and designing a self-adaptive decoder according to the pieceable fault-tolerant protocol (PFTP), fault-tolerant universal computing can be realized in a relatively fragile way [25][27]. For example, a logical controlled-controlled-Z (CCZ) gate that acts on Steane-7 logical qubits can be equivalently synthesized by a 4-pieceable fault-tolerant circuit, such a gate plays a crucial role in forming a universal gate library; Measurement-based quantum computing can be utilized to design pieceable fault-tolerant conversion circuits between two stabilizer codes who with complementary transversal quantum gates [27][28]. However, the
existing minimal weight decoder can only analyze the correctable error of the encoded circuit, but errors that occurred in the syndrome extraction circuit are easy to accumulate between different circuit pieces, thereby resulting in an undetectable logical error. Therefore, the PFTP is a less error-propagation-aware fault-tolerant circuit design method compared with the transversal protocol.

Through the noise simulation, we found that, for a circuit designed by PFTP, some machine learning algorithm can be utilized to analyze the potential relationship between the collection of all intermediate syndrome data and the event of error propagation \[29, 31\]. For instance, during the application of the code conversion circuits between Steane-7 qubit code and Reed-Muller-15 (RM-15) qubit code \[31\], the error that occurred in the syndrome extraction process of the former circuit piece usually affects the syndrome data of the latter circuit piece, thus resulting in error correction failure. With the noise model assuming that the error is independent of each other, we think that an additional discriminant model can be added to the decoder to identify this type of errors. So we propose an optimized concatenation universal fault-tolerant protocol. The stabilizer code is defined by a set of particular Pauli operators, also referred to as the stabilizer formalism developed by Gottesman \[7\].

**Definition 1 (n-qubit Pauli group)** A n-qubit Pauli group \(G_n\) consists of the tensor products of single-qubit Pauli operators that can be described by the following set:

\[
G_n = \{ \lambda \sigma_1 \otimes \sigma_2 \otimes \cdots \otimes \sigma_n : \sigma_i \in \{I, X, Y, Z\} \},
\]

where \(\lambda \in \{\pm 1, \pm \sqrt{-1}\}\).

Given an operator \(g \in G_n\), its support is defined as a subset of \([n] := \{1, 2, \ldots, n\}\), where we denote by the symbol \(\text{supp}(g)\) the set of all \(i \in [n]\) such that \(g\) acting on the \(i\)-th qubit is not identity. The weight of a Pauli operator \(wt(g)\) equals the size of its support. For example, \(wt(Z \otimes Z \otimes I) = wt(Z_0 Z_1) = 2\).

Next, we describe the stabilizer formalism. A stabilizer \(S\) is an abelian subgroup belonging to \(G_n\), which does not contain \(-I\). Because the operators in this group are Hermitian and mutually commuting, they can be diagonalized simultaneously. Therefore, the stabilizer code can be defined by the following statement:

**Definition 2 (n-qubit Stabilizer Code)** Given a stabilizer subgroup \(S < G_n\), a n-qubit stabilizer code \(C_n\) is the joint +1 eigenspace of operators in \(S\) belonging to the Hilbert space \((\mathbb{C}^2)^\otimes n\) and can be described by the following set:

\[
C_n = \{ |\psi\rangle : g|\psi\rangle = |\psi\rangle, \forall g \in S \}.
\]

Without loss of generality, we assume that the dimension of \(C_n\) is \(2^{n-m}\), where \(k = n - m\) is the number of logical qubits and \(d\) is the code distance. According to quantum coding theory, the stabilizer of \(C_n\) has \(m\) generators, denoted as \(S = \langle g_1, \ldots, g_m \rangle\), and \(C_n\) can correct any Pauli error of weight \(t \leq \left\lfloor \frac{d-1}{2} \right\rfloor\).

With the definition stated above, it is necessary to find corresponding encoded operations to realize a given computation task. The Pauli subgroup \(C(S)\), called the centralizer of \(S\), is defined to be a set of elements in \(G_n\) that commute with all elements in \(S\). We can first choose operators \(\tilde{Z}_1, \ldots, \tilde{Z}_k\) and \(\tilde{X}_1, \ldots, \tilde{X}_k\) in \(C(S)\) that are independent of the generators of \(S\) and satisfy the communication conditions \(\tilde{X}_i \tilde{Z}_k = (-1)^{\delta_{ij}} Z_1 X_k\). For convenience, we also refer to these operators as logical Pauli operators.
Therefore, it is not difficult to say that the encoded $|0⟩^\otimes k$ can be uniquely represented by the joint +1 eigenspace of the following set of $n$ operators: $\{g_1, \ldots, g_n, Z_1, \ldots, Z_k\}$. Similarly, we can give other basic encoded states by applying the corresponding logical Pauli X operators to $|0⟩^\otimes k$ such that $\sum_{i=1}^{k} X_i x_i |0⟩^\otimes k$, where $x_i \in \{0, 1\}$.

**TABLE I.** The generators of the stabilizer $S$ and logical Pauli operators of Steane’s $[[7;1;3]]$ code

| Stabilizer generators | Logical Pauli Operator |
|-----------------------|------------------------|
| $X_0X_2X_4X_6$       | $\tilde{X}_1 = X^\otimes 7$ |
| $X_1X_2X_3X_6$       | $\tilde{Z}_1 = Z^\otimes 7$ |
| $X_3X_4X_5X_6$       |                         |
| $Z_0Z_2Z_4Z_6$       |                         |
| $Z_1Z_2Z_5Z_6$       |                         |
| $Z_3Z_4Z_5Z_6$       |                         |

**TABLE II.** The generators of the stabilizer $S$ and logical Pauli operators of Reed-Muller $[[15;1;3]]$ code

| Stabilizer generators | Logical Pauli Operator |
|-----------------------|------------------------|
| $X_7X_8X_9X_{10}X_{11}X_{12}X_{13}X_{14}$ | $\tilde{X}_1 = X^\otimes 15$ |
| $X_1X_2X_3X_4X_5X_6X_7X_8X_{11}X_{12}X_{13}X_{14}$ | $\tilde{Z}_1 = Z^\otimes 15$ |
| $X_9X_10X_11X_12X_{13}X_{14}$ |                         |
| $Z_7Z_8Z_9Z_{10}Z_{11}Z_{12}Z_{13}Z_{14}$ |                         |
| $Z_1Z_2Z_3Z_4Z_5Z_6Z_7Z_{11}Z_{13}Z_{14}$ |                         |
| $Z_0Z_2Z_3Z_4Z_5Z_6Z_8Z_{12}Z_{13}Z_{14}$ |                         |
| $Z_9Z_{10}Z_{11}Z_{12}Z_{13}Z_{14}$ |                         |
| $Z_5Z_{10}Z_{11}Z_{12}Z_{13}Z_{14}$ |                         |
| $Z_9Z_{10}Z_{11}Z_{12}Z_{13}Z_{14}$ |                         |
| $Z_2Z_6Z_7Z_8Z_{11}Z_{12}Z_{13}Z_{14}$ |                         |

We show the stabilizer generators and logical Pauli operators of Steane’s 7-qubit code and RM-15 code in Table I and Table II. In addition, how to perform operations on an encoded state without losing the code’s protection is a highly important topic. We next give the definition of $t$-fault-tolerant; more details can be found in [7][8].

**Definition 3** ($t$-fault-tolerant) For a $[[n;k;d]]$ code $C_n$, let $t = \lceil \frac{n-k}{d-1} \rceil$, a quantum operation which is protected by $C_n$ is $t$-fault-tolerant, if the following two conditions are satisfied:

(i) For an input codeword with an error of weight $w_1$, if $w_2$ single-qubit faults occur during the operation with $w_1 + w_2 \leq t$, 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 $w \leq t$, 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 say that ideally decoding is equivalent to perform a round of noise-free error correction. Both conditions are required to ensure that correctable errors do not propagate through the entire operation and prevent errors from accumulating during multiple rounds of error correction.

Next we introduce the PFTP. It would be convenient to intuitively imagine that a specific encoded circuit $C$ on $C_n$ can be decomposed into $r$ pieces:

$$C = C_rC_{r-1} \ldots C_1,$$

where parameter $r$ refers to the minimum number of circuit pieces that an encoding circuit can be divided without the loss of fault-tolerance. More specially, we can obtain a fault-tolerant variant of $C$ if each $C_i$ is carefully designed such that certain uncorrectable errors can be signaled by some adapted error correction process $E_i$. So a modified $t$-fault-tolerant variant of $C$, denoted as $\tilde{C}$, can be described as follows:

$$\tilde{C} = E_rE_{r-1} \ldots E_1C_1.$$

By performing adapted error correction after each $C_i$ on the encoded state, we obtain several fault-tolerant gadgets $E_iC_i (i = 1, \ldots, r)$.

To specifically explain these adjustment error correction processes, we would like to introduce some useful concepts. The first is contagious error; it is a type of Pauli error operators that can be described as:

$$E_r = \{ E \in G_n : \exists i \text{ s.t. } [E, C_i] = EC_i \neq 0 \}. $$

Only contagious error occurred in $C_i$ may propagate. For example, we assume that one of a circuit piece $C_m$ only contains a single physical control-Pauli $Z$ $(CZ)$ gate with control qubit $a$ and target qubit $b$, and the input state at the $a$-th qubit has a Pauli $X$ error; then after the application of $C_m$, the input error will become a 2-qubit error $X \otimes Z$, i.e., $CZ(X \otimes I)CZ^\dagger = X \otimes Z$. So the single Pauli $X$ is a contagious error related to $C_i$, and the $Z$-type Pauli errors $(I \otimes Z, Z \otimes I, Z \otimes Z)$ are non-contagious.

In each $E_i (i = 1, \ldots, r-1)$ we correct contagious errors immediately and left the non-contagious error until $E_r$. The syndrome information of $E_r$ will be recorded and sent to $E_r$ for the correction of non-contagious error. Here we also introduce the tool for correcting contagious error:

$$S_{EC} = \{g \in S : \forall i, [g, C_i] = 0 \}. $$

For a $r$-pieceable fault-tolerant ($r$-PFT for short) circuit $C$, the elements of the stabilizer group satisfying the above conditions are called constant stabilizers.

**III. MULTI-CLASSIFICATION MODEL FOR ERROR SYNDROME OF THE PFT LOGICAL CIRCUIT**

Next, we design a decoding scheme for the final error correction process of a PFT logical circuit. For convenience, we first introduce the definition of error syndrome. The syndrome is defined as a classical bit-string
for tracking error events of a fault-tolerant circuit, and it can be obtained by fault-tolerantly measuring stabilizer generators. Since the effect of most noise channels on the quantum state can be decomposed as a linear combination of Pauli operators [S], we only consider this kind of error throughout the paper. Assuming an error $E$ has occurred on a $[[n; k; d]]$ encoded state, we then fault-tolerantly measure the stabilizer generators and obtain a syndrome vector as follows:

$$s_i(E) = \begin{cases} 1 & \text{if } E_i \neq 0 \\ 0 & \text{if } E_i = 0 \end{cases}, \quad i = 1, \ldots, n - k. \quad (7)$$

Then the syndrome vector of $E$ can be defined as a bit-string $s(E) = (s_1(E), \ldots, s_{n-k}(E)) \in \mathbb{Z}_2^{n-k}$.

A typical decoding process can be described by matching the measured syndrome $s$ with its most likely recovery operator $R_s$ such that $R_s E \in S$. More specifically, the recovery operator can be written as:

$$R_s = \mathcal{L}(s) \mathcal{T}(s) \mathcal{G}(s), \quad (8)$$

where operator $\mathcal{G}(s) \in S$ and $\mathcal{T}(s)$ belongs to an Abelian group composed of Pauli operators called pure error [35]. This group has $n - k$ generators and satisfies $\{g_i, T_j\} = \delta_{ij}$, $i, j = 1, \ldots, n - k$. The part of the recovery operator $\mathcal{L}(s) \in N(S)/S$ belongs to the logical Pauli group acting on $k$ logical qubits.

A decoder based on the minimum weight scheme can quickly infer the pure error $\mathcal{T}(s)$ from the syndrome $s$. However, this kind of decoders are not scalable for large quantum code and cannot infer potential logical errors. Actually, an optimal decoder should be able to infer the most likely logical fault $\mathcal{L}(\mathcal{T})$ from the known syndrome, such that:

$$\mathcal{L}(\mathcal{T}) = \arg\max_{\mathcal{L}} \{ P(\mathcal{L}|\mathcal{T}) \}. \quad (9)$$

The PFTP described in Eq. (1) can make the syndrome data of different intermediate error corrections have a potential correlation. For example, the round-robin logical controlled-Z gate on $[[5; 1; 3]]$ code has been carefully designed to be fault-tolerant. The constant stabilizer syndromes of this gate can be further analyzed to infer some weight-2 $Z$-type Pauli errors [25]. This observation inspired us to design an extended decoding scheme to infer the possible error propagation event among several circuit pieces as much as possible. More specifically, for the final error correction process $\mathcal{E}_r$, we explain its decoding process as a pattern matching model so as to jointly analyze measurement information of $r$-pieceable error correction processes. Then this process would infer the most likely logical error if we effectively train the model functions. Therefore, we conclude that an optimal decoder of $\mathcal{E}_r$ should be able to infer the logical error information $\mathcal{L}(\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(r)})$ such that:

$$\mathcal{L}(\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(r)}) = \arg\max_{\mathcal{L}} \{ P(\mathcal{L}|\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(r)}) \}, \quad (10)$$

where $\mathcal{T}^{(i)}$ is the syndrome information of pure errors obtained from $\mathcal{E}_i$.

Since all codes considered in this paper encode a single logical qubit, we first rewrite the recovery operator of $\mathcal{E}_r$ as follows:

$$R_{s_r} = \mathcal{L}(s) \mathcal{T}(s_r) \mathcal{G}(s_r), \quad (11)$$

where $s_r$ is the syndrome vector obtained by the error correction process of $\mathcal{E}_r$ and $s = s_1 \times \cdots \times s_r$. Single logical qubit Pauli group can be expressed as $< i, \bar{X}_i, \bar{Z}_i >$, so the logical component can be decomposed as $\mathcal{L}(s) = \bar{X}_1^{g_2} \bar{Z}_1^{g_3}$ regardless of the global phase factor, where $a, b \in \mathbb{Z}_2$.

Therefore, we introduce two functions $g_X$ and $g_Z$ such that:

$$R_{s_r} = \bar{X}_1^{g_X(s)} \bar{Z}_1^{g_Z(s)} \mathcal{T}(s_r) \mathcal{G}(s_r). \quad (12)$$

According to Eq. (12), we model the decoding process as a labeled classification task of discrete data. Then we define the data set as $D \subseteq \{ s \} \times L$, and any element of $D$ can be represented with the form $(s, l)$, where $l$ is the label of class. Here we use the one-hot encoded label for $k$ classes, i.e., $l \in L = \{ l : l \in \{0,1\}^k, \mathcal{T}^l \mathcal{I} = \mathcal{I} \}$. From Eq. (12), the output of model functions need to give the predicted value $(g_X(s), g_Z(s))$ based on all intermediate syndromes. So we take different predicted values as classification labels to corresponding to logical recovery operators, and denote these labels as $\{ l_1, l_X, l_Y, l_Z \}$.

For a $[[n; k; d]]$ code, we note that the logical operator of this code belongs to the following subgroup $\mathcal{L} \subseteq < i, \bar{X}_1, \ldots, \bar{X}_k, \bar{Z}_1, \ldots, \bar{Z}_k >$. Similarly, we can construct a multi-classification model with the number of $2^k$ categories for a $r$-PFT logical circuit, and the recovery operator of $\mathcal{E}_r$ is given as follows:

$$R_{s_r} = \bar{X}_1^{g_X(s)} \cdots \bar{X}_k^{g_X(s)} \bar{Z}_1^{g_Z(s)} \cdots \bar{Z}_k^{g_Z(s)} \mathcal{T}(s_r) \mathcal{G}(s_r). \quad (13)$$

IV. THE DECODER CONSTRUCTION OF 2-PFT LOGICAL CNOT CIRCUIT

Now we analyze the logical CNOT gates acting on the Steane-7 logical qubit and RM-15 logical qubit. First, we show that each of it can be divided into a product sequence of two fault-tolerant sub-circuits. Secondly, following our design framework, we take $\mathcal{C}_4$ as an example and use the neural network algorithm to give the structure of its decoder.

Nikalaia et al. [36] has proposed a round-robin logical CNOT gate with $\mathcal{C}_7$ encoded state as the control and $\mathcal{C}_{15}$ as the target, but they didn’t give a specific solution to make this logical circuit fault-tolerant. Following their results, we divide this circuit into two pieces and insert a constant stabilizer error correction process between them to make the entire circuit fault-tolerant, as
FIG. 1. Pieceable fault-tolerant logical CNOT circuit with $C_7$-logical qubit as the control and $C_{15}$-logical qubit as the target, we use $\tilde{C}_A$ to represent this circuit. The intermediate contagious error correction process $E_1$ is applied after $C_1$, during which only the constant stabilizers are measured to correct the Pauli $X$ error for control code block and Pauli $Z$ error for target code block.

FIG. 2. Pieceable fault-tolerant logical CNOT circuit with $C_{15}$-logical qubit as the control and $C_7$-logical qubit as the target, we use $\tilde{C}_B$ to represent this circuit. The intermediate contagious error correction process $E_1$ is applied after $C_1$, during which only the constant stabilizers are measured to correct the Pauli $X$ error for control code block and Pauli $Z$ error for target code block.

shown in Fig. 1. For the control code block, the stabilizer $Z_0Z_2Z_4Z_6, Z_1Z_2Z_5Z_6$ and $Z_3Z_4Z_5Z_6$ are invariant under the conjugation of circuits $C_1$ and $C_2$, so these operators are constant stabilizers by definition. Similarly, for the target code block, those generators that only contain the Pauli-$X$ operator in Table II are constant stabilizers.

In contrast, we give a 2-PFT logical CNOT circuit with $C_{15}$ encoded state as control and $C_7$ encoded state as target, which is shown in Fig. 2. We have verified that the stabilizer $X_0X_2X_4X_6, X_1X_2X_5X_6$ and $X_3X_4X_5X_6$ of the target code block are invariant under the conjugation of circuits $C_1$ and $C_2$. For the control code block, those generators that only contain Pauli-$Z$ operator in Table II are constant stabilizers.

A. The construction of decoders

By keeping the constant stabilizer syndrome data in $E_1$ and passing it to $E_2$, some uncorrectable errors caused
by contagious error will have unique syndromes. Therefore, we can extend the error-syndrome lookup table to include more correspondences. This extended decoding strategy can ensure that $\mathcal{C}_A$ and $\mathcal{C}_B$ are at least 1-fault-tolerant circuits \cite{25}. A fault-tolerant logical circuit on a code can be modeled as a level-1 extend rectangle (1-exRec) \cite{10}, as shown in Fig. 3. Based on this circuit model, we can design a simulation scheme to calculate the pseudo-threshold of a 1-exRec. The pseudo-threshold can be defined as the intersection between the error rate line of the unprotected gate and its corresponding logical variant, the higher the pseudo-threshold of the logical circuit, the better its anti-noise ability. Actually, from our numerical experiments, we find that the above strategy cannot fully utilize the syndrome results of intermediate error corrections to infer the possible logical error.

In contrast, our neural-network decoding scheme can be used to analyze the relationship between syndrome and error, and design an error diffusion early warning mechanism to correct possible logical errors. We construct a syndrome data classification model for these 2-PFT circuits. We use the deep neural network algorithm to train these models and apply them to the final error correction process. We then propose the following decoding process:

(i) During the implementation of $\mathcal{E}_2$, we use the minimum weight decoding method to apply the pure error recovery operator $\mathcal{T}(s_2)$;

(ii) Take the syndrome data $s = s_1 \times s_2$ as model input to get the class value $g_X(s)$ and $g_Z(s)$. Then we apply the logical recovery operator $\mathcal{L}(s) = X_1^{g_X(s)} Z_1^{g_Z(s)}$.

![FIG. 3. For a given error correction code, any physical quantum circuit has its corresponding logical variant on this code, also called rectangle (Rec). A 1-exRec is a level-1 encoding Rec along with its leading error correction circuit (LEC) and trailing error correction circuit (TEC).](image)

1. Data set and labels

For a neural network decoder, the sparsity of training data is the main reason that limits its performance. Specifically, with the increasing of code distance and the decreasing of physical error rate, the effective training data will rarely occur in noise simulation, so that the neural network cannot learn some patterns from the syndrome data.

Chamberland et al. \cite{24} proposed the concept of a set of malignant locations, that is, a set of $n$ locations in a fault-tolerant 1-Rec is defined to be malignant if there exists an error configuration that acts on these $n$ locations such that the 1-Rec has a logical fault. Meanwhile, they pointed out that the Monte Carlo method can be utilized to estimate the proportion of $n$ malignant locations set in all combinations of locations in a 1-Rec, this rate is also denoted as $\hat{f}_n$.

Inspired by this, we propose that, for 1-Rec on a distance-3 code, the effective training data can be collected by the following way: First, we take the sampling method and perform circuit simulation to estimate the rate $\hat{f}_1$, where the number of simulations can be set large enough to obtain an estimated value with a sufficiently small variance; Then, after each simulation, we save the "valid" syndrome data and use the ideal Shor method to decide the logical error ($\tilde{X}$ or $\tilde{Z}$) that occurred on the data state. In addition, the syndrome data caused by multiple fault locations in 1-Rec may mislead the training of the neural network model. For example, for the 3-qubit repetition code, the syndrome data for errors $X_0X_1$ and $X_2$ are the same, but only the former will make the error correction fail and result in a logical error. In particular, when training the neural network decoders of circuits $\mathcal{C}_A$ and $\mathcal{C}_B$, we only use the valid data which is collected from the estimation of $\hat{f}_1$. After the training is completed, we will analyze the error correction capability of our decoders through depolarization noise simulation.

2. Model optimization strategy and objective function

The parameters of the model function need to be found by using a suitable machine learning algorithm. Next, we use the deep neural network algorithm to get the model function. More details can be referred to \cite{37, 38}.

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 $\omega$, then the problem of finding the optimal coefficients of the function $g_Xg_Z$ is transformed into finding the optimal parameter $\omega$. The objective function $\mathcal{L}(D, \omega)$ usually is taken as the goal of optimization:

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

The objective function is the quantification of the difference between the model output and the observation result, it can be described as follows:

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

where $I_{s}^{(j)}$ is the binary indicator if the $j$-th class label is the correct classification for measured syndrome $s$, $M$
is the number of layers for the network, and $\hat{P}_j$ is the model predicted probability that $s$ belongs to $j$-th class. Our network structure is shown in Fig. 4, the syndrome vector $s$ as the input data of the network is first sent to $M$ hidden layers, and the hidden layer function is shown as follows:

$$H_m = f(\omega^{(m-1)}H_{m-1} + b^{(m-1)}), \quad m = 1, \ldots, M. \quad (16)$$

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 $s$ belongs to the $j$-th category. We denote $\hat{P} = (\hat{P}_1, \ldots, \hat{P}_k)$, and adapt the following equation to obtain this

$$\hat{P}_j = \frac{e^{V_j}}{\sum_{j=1}^{k} e^{V_j}}, \quad j = 1, \ldots, k, \quad (17)$$

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

$$\hat{I}_s^{(i)} = \begin{cases} 1 & \text{for } i = \arg\max_j \hat{P}_j \\ 0 & \text{otherwise.} \end{cases}$$

In contrast, reducing the empirical error usually makes the model over-fit the training data and gives a more complicated model. 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 also called regularization. Commonly used regularization methods are $L1$ regularization and $L2$ regularization. Here we use $L2$ regularization, as shown in Eq. (15), which will limit the coefficient $\omega$ in the model function to reduce the complexity of the model. A hyper-parameter $\lambda$ is also introduced to balance loss function and regularized weight.

For the training details of $\tilde{C}_A$, we construct two binary classification models for the two types of data labeled by $g_X(s)$ and $g_Z(s)$. The dimension of input data is 47, and each model has 4 hidden layers and the number of hidden layer nodes are 256, 512, 1024 and 256. We set the batch size is 30, the learning rate is $10^{-4}$. We train our models on pytorch.

V. CODE CONSTRUCTION AND THRESHOLD ANALYSIS

A. Nonuniform Concatenated 33-qubit Code for Universal Fault-tolerant Computation

The final error correction process $\mathcal{E}_r$ of a $r$-PFT circuit $\mathcal{C}$ can be used to correct errors that marked by any fault-tolerant sub-circuit $\mathcal{E}_r\mathcal{C}_l(i = 1, \ldots, r - 1)$ or correctable errors occurred in $\mathcal{C}_r$. Specifically, let

$$\mathcal{C}_{Rec} = \mathcal{C}_r\mathcal{C}_{r-1}\ldots\mathcal{C}_1. \quad (18)$$

Then, $\mathcal{C}_{Rec}$ actually implement the effect of the corresponding physical circuit on the logical qubits, and $\mathcal{E}_r$ can be taken as its error correction process. So we can model $\mathcal{C} = \mathcal{E}_r\mathcal{C}_{Rec}$ as a 1-Rec. On the other hand, for the logical qubit applied by a non-transversal logical circuit, some of its physical qubits that are interacted during the implementation of this circuit can be re-encoded so as to fulfill the requirement of fault-tolerance.

Based on the above two points, we now give the construction of a 33-qubit non-uniform concatenated code. First, as shown in Fig. 5 for a Steane-7 logical qubit, we re-encode the 0, 5 and 6-th physical qubits. The 5-th qubit is replaced with an RM-15 logical qubit, because the qubit will be applied with a T gate later. The 0 and 6-th qubits are replaced with Steane-7 logical qubits. Therefore, the controlled-NOT gates that act on two different logical qubits can be implemented under PFTP and equipped with our neural network-based decoder. Secondly, we give a description about how to implement the universal gate library $\{H, CNOT, T\}$ on this code. Since the level-2 code of the 33-qubit code is the Steane-7 qubit code, it can transversally realize any Clifford operators. Especially, from Fig. 5 have we pointed out that all of the level-1 logical components that make up the logical T gate can be fault-tolerant, so the 33-qubit code can implement universal computing in a fault-tolerant manner.

B. Numerical simulation and threshold analysis

We analyze the threshold performance of the 33-qubit code through the stabilizer circuit simulation algorithms [39]. Our simulation experiments are executed on the platform called LIQUID [40]. We first show the pseudo-threshold results of $\tilde{C}_A$ and $\tilde{C}_B$ equipped with our neural-network decoding procedure. Subsequently, we collect the depolarization error rates for all level-1 logical elements that 33-qubit code used to realize the universal gate library. Based on this, we give the universal computing pseudo-threshold value. At the same time, we make a comparison with the existing results in terms of qubit resource consumption, T gate execution time, and the lower bound of the threshold for realizing universal quantum computing.
For convenience, we first introduce the concept of malignant error event [23], which can be defined as follows: Let $|\psi\rangle$ be the input encoded state. As can be seen from Fig. 6, the LEC circuit is first applied to the input state, and then we ideally project the output state of the LEC circuit into the original code space. Let $|\psi_{LEC}\rangle$ be the state after this projection, and $|\psi_{TEC}\rangle$ be the final output state after the TEC and a round of ideal projection. Then we define the malignant error event as

$$|\psi_{TEC}\rangle = E \cdot \text{cnot}(|\psi_{LEC}\rangle),$$

where $E$ is a single or two-logical qubit error. For example, suppose the input state is $|\emptyset\rangle$, and we get $|01\rangle$ after implementation of CNOT 1-exRec, then a malignant error event $E$ occurs, where $E$ can be one of the following logical errors: $X, Y, Z, XZ, ZY$.

The logical error rate of a 1-exRec can be defined as

$$\sum_E \mathcal{P}(E|\epsilon)$$

where $\epsilon$ represents the physical error rate for all components in a quantum circuit, $E$ represents all the possible malignant error events that a noisy-CNOT gate will occur, i.e., with a probability of $\frac{1}{2^{16}}$, a two-qubit Pauli error drawn uniformly and independently from $\{I, X, Y, Z\} \otimes \{I, X, Y, Z\} \backslash \{I \otimes I\}$, and $\mathcal{P}(E|\epsilon)$ is the probability of logical error $E$ with a given physical error rate. Therefore, the pseudo-threshold of a 1-exRec is the physical error probability that satisfies $\epsilon = \sum_E \mathcal{P}(E|\epsilon)$.

For our simulation scheme, we construct depolarization noise model for threshold calculation, that is, apply the following noise channel to each noise physical component in a 1-exRec:

$$\epsilon(\rho) = (1 - \frac{3\epsilon}{4})\rho + \frac{\epsilon}{4}(X\rho X + Z\rho Z + Y\rho Y).$$

The basic noise components in our simulation are listed in Table IV. Here we make the assumption that the physical error rate of a two-qubit gate and one-qubit gate being the same. With the basic assumption described above, we now give the calculate method of logical failure rate for $\tilde{C}_A$ or $\tilde{C}_B$ as follows:

\begin{table}[h]
\centering
\caption{Types of physical noisy locations present in the simulation experiment of a 1-exRec.}
\begin{tabular}{|c|c|}
\hline
No. & component type \\
\hline
1 & Basic state $|\emptyset\rangle$ preparation \\
2 & Basic state $|+\rangle$ preparation \\
3 & Measurement of Pauli operator \\
4 & Two-qubit quantum gate \\
5 & Single-qubit non-identity quantum gate \\
\hline
\end{tabular}
\end{table}

(i) First, we choose a sequence of physical error rates ranging from $10^{-4}$ to $10^{-3}$. For each fixed physical error rate $\epsilon$, we simulate the $\tilde{C}_A$ or $\tilde{C}_B$ under a depolarizing noise channel. The simulation is valid if all the ancillary blocks in a 1-exRec pass the verification.

(ii) When a noisy logical CNOT gate fails, it applies an 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 number of simulation $N = 10^6$ and denote the logical error rate as $\mathcal{P}(E|cnot, \epsilon) = \frac{m}{N}$, where $m$ is the number of logical errors $E$ occurring after $N$ iterations.

A polynomial fitting algorithm is used to obtain two types of 1-exRec logical error curves, and it can be found that the best fitting effect can be achieved by using quadratic function fitting. Actually, for an effective fault-tolerant quantum operation, its logical error rate should be smaller than the error rate of unprotected one, there also has been proved that if error propagation through the entire fault-tolerant circuit is limited, and a good decoder is used, the logical error rate should exhibit the power-law scaling [24], which means code concatenation techniques can be adapted to exponentially reduce the logical error rate.

Following the above simulation scheme, we show the performance of the decoder we designed through Fig[7].

\begin{table}[h]
\centering
\caption{Pseudo-threshold results for minimal weight decoding strategy and neural-network based decoding strategy.}
\begin{tabular}{|c|c|c|c|}
\hline
logical circuit & minimal weight & neural-network & \\
\hline
$\tilde{C}_A$ & $9.36 \times 10^{-7}$ & $1.06 \times 10^{-4}$ & \\
$\tilde{C}_B$ & $1.99 \times 10^{-6}$ & $1.98 \times 10^{-4}$ & \\
\hline
\end{tabular}
\end{table}
FIG. 7. (a) The rate of logical error $XI$ for $\tilde{C}_A$. (b) The rate of logical error $IZ$ for $\tilde{C}_A$. (c) The rate of the sum of logical errors for $\tilde{C}_A$.

Fig. 8. (a) The rate of logical error $XI$ for $\tilde{C}_B$. (b) The rate of logical error $IZ$ for $\tilde{C}_B$. (c) The rate of the sum of logical errors for $\tilde{C}_B$.

It can be observed that the fitting curve of the logical error rate of $\tilde{C}_A$ and $\tilde{C}_B$ based on two decoding strategies show the characteristics of a quadratic curve. We think this is a reasonable numerical simulation result. Because the code distance of Steane and RM code is three, decoders for these codes are effective if they can correct most of the single-qubit errors, such that the logical error can only be caused by two or more single-qubit errors that occur independently. In contrast, our decoding strategy can more effectively reduce the logical error rate of the pieceable fault-tolerant circuit, so it leads to a higher threshold result, as shown in Table III.

TABLE V. The pseudo threshold required for realizing fault-tolerant universal computing of three non-uniform concatenation codes and the estimation of the circuit depth required to realize a fault-tolerant T gate.

| Code\gate | pseudo-threshold | Time steps of T gate |
|-----------|------------------|---------------------|
| 49-qubit CNOT | $1.21 \times 10^{-3}$ | 40 |
| 49-qubit Hadamard | $7.76 \times 10^{-5}$ | |
| 49-qubit T | $4.18 \times 10^{-4}$ | |
| 25-qubit CNOT | $4.08 \times 10^{-4}$ | 91 |
| 25-qubit Hadamard | $7.21 \times 10^{-4}$ | |
| 25-qubit T | $3.93 \times 10^{-4}$ | |
| 33-qubit CNOT | $3.25 \times 10^{-4}$ | 59 |
| 33-qubit Hadamard | $4.17 \times 10^{-4}$ | |
| 33-qubit T | $1.12 \times 10^{-4}$ | |

We use the Steane method to make the error correction fault-tolerant throughout this paper. We assume that a
circuit $C$ can be divided into several time steps so that at each time step, each qubit (excluding those added at later time steps or deleted at earlier time steps) involves exactly one location, that is, each qubit in the first position is preparation and the last position of each qubit is a measurement. For most of the fault-tolerant protocols, the processing speed of syndrome information in classical computers should keep up with quantum computers. Besides, in the process of applying a 1-Rec on the logical qubit, it is reasonably to think that the auxiliary state for error correction can be prepared asynchronously, so that the auxiliary state can be introduced immediately whenever it is needed for syndrome extraction. For example, we estimate the number of unit time steps required for the error correction process on the Steane 7-qubit code.

As shown in Fig. 12, this circuit can be divided into six time steps (including the recovery operation for the data state). Therefore, a transversal fault-tolerant gate on this code needs to consume seven time steps. Similarly, we can also estimate the time step resources required for the error correction process that uses the Shor method. In Table V, we compare the total number of time steps required to achieve a fault-tolerant T gate by three non-uniform concatenated codes. It can be seen that, without the use of inner level code conversion, our circuit structure effectively reduces the number of time steps for implementing a T gate compared with 25-qubit code.
VI. CONCLUSIONS

In summary, the main contribution of this work is as follows: first, we propose a general design framework based on machine learning technology for the decoder of the pieceable fault-tolerant circuit. Based on this, we propose an improved concatenation universal fault-tolerant computing scheme; In particular, we design a non-uniform concatenation 33-qubit code and provide the construction of a universal quantum gate library \( \{ H, CNOT, T \} \) on its corresponding logical qubit. Compared with the existing concatenation code, we have reduced the number of non-fault-tolerant locations in the level-1 encoding layer of the logical Hadamard gate. Meanwhile, we introduced the pieceable fault-tolerant locations equipped with a neural network decoder in the level-1 encoding layer of the logical T gate, such replacement makes it consumes fewer time steps than 25-qubit concatenation code. Secondly, to verify the improvement of our scheme, we compare the pseudo-threshold with the existing concatenation code under the depolarized noise model. The simulation results showed that the scheme we designed has a higher lower bound than the 49-qubit code.

Besides, with the consideration of the noise in the current experimental platform, we will use the statistical learning methods to effectively analyze its impact on the output of the circuit in our future work.

ACKNOWLEDGMENTS

This work is supported by the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province under Grant No. SKLACSS-202109, the National Natural Science Foundation of China under Grant No. 62076042 and the National Natural Science Foundation of China under Grant No. 62172075.

[1] A. M. Steane, Error correcting codes in quantum theory, Physical Review Letters 77, 793 (1996).
[2] A. M. Steane, Active stabilization, quantum computation, and quantum state synthesis, Physical Review Letters 78, 2252 (1997).
[3] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Physical Review A 86, 032324 (2012).
[4] E. T. Campbell, B. M. Terhal, and C. Vuillot, Roads towards fault-tolerant universal quantum computation, Nature 549, 172 (2017).
[5] J. Preskill, Quantum computing in the nisq era and beyond, Quantum 2, 79 (2018).
[6] D. E. Gottesman, Stabilizer Codes and Quantum Error Correction, Ph.D. thesis, California Institute of Technology (1997).
[7] D. Gottesman, An introduction to quantum error correction and fault-tolerant quantum computation, in Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics, Vol. 68 (2010) pp. 13–58.
[8] M. A. Nielsen and I. L. Chuang, Quantum computation and quantum information (2000).

---

[10] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, Quantum Information & Computation 6, 97 (2006).
[11] D. Aharonov and M. Ben-Or, Fault-tolerant quantum computation with constant error rate, SIAM Journal on Computing (2008).
[12] P. Reinhold, S. Rosenblum, W.-L. Ma, L. Frunzio, L. Jiang, and R. J. Schoelkopf, Error-corrected gates on an encoded qubit, Nature Physics 16, 822 (2020).
[13] H. Häffner, C. F. Roos, and R. Blatt, Quantum computing with trapped ions, Physics reports 469, 155 (2008).
[14] 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).
[15] 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 536, 63 (2016).
[16] M. H. Devoret and R. J. Schoelkopf, Superconducting circuits for quantum information: an outlook, Science 339, 1169 (2013).
[17] B. Eastin and E. Knill, Restrictions on transversal encoded quantum gate sets, Physical review letters 102, 110502 (2009).
[18] B. Zeng, A. Cross, and I. L. Chuang, Transversality versus universality for additive quantum codes, IEEE Transactions on Information Theory 57, 6272 (2011).
[19] S. Bravyi and A. Kitaev, Universal quantum computation with ideal clifford gates and noisy ancillas, Physical Review A 71, 022316 (2005).
[20] J. O’Gorman and E. T. Campbell, Quantum computation with realistic magic-state factories, Physical Review A 95, 032338 (2017).
[21] A. G. Fowler, S. J. Devitt, and C. Jones, Surface code implementation of block code state distillation, Scientific reports 3, 1939 (2013).
[22] T. Jochym-O’Connor and R. Laflamme, Using concatenated quantum codes for universal fault-tolerant quantum gates, Physical Review Letters 112, 010505 (2014).

---

[23] C. Chamberland, T. Jochym-O’Connor, and R. Laflamme, Thresholds for universal concatenated quantum codes, Physical Review Letters 117, 010501 (2016).
[24] C. Chamberland, T. Jochym-O’Connor, and R. Laflamme, Overhead analysis of universal concatenated quantum codes, Physical Review A 95, 022313 (2017).
[25] T. J. Yoder, R. Takagi, and I. L. Chuang, Universal fault-tolerant gates on concatenated stabilizer codes, Physical Review X 6, 031039 (2016).
[26] D. Gottesman, Efficient fault tolerance, Nature 540, 44 (2016).
[27] K. R. Colladay and E. J. Mueller, Rewiring stabilizer codes, New Journal of Physics 20, 083030 (2018).
[28] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, Fault-tolerant conversion between the steane and reed-muller quantum codes, Physical review letters 113, 080501 (2014).
[29] G. Torlai and R. G. Melko, Neural decoder for topological codes, Physical review letters 119, 060501 (2017).
[30] C. Chamberland and P. Ronagh, Deep neural decoders for near term fault-tolerant experiments, Quantum Science and Technology 3, 044002 (2018).
[31] R. Sweke, M. S. Kesselring, E. P. van Nieuwenburg, and J. Eisert, Reinforcement learning decoders for fault-tolerant quantum computation, arXiv preprint arXiv:1810.07207 (2018).
[32] P. Baireuther, T. E. O’Brien, B. Tarassik, and C. W. Beenakker, Machine-learning-assisted correction of correlated qubit errors in a topological code, Quantum 2, 48 (2018).
[33] P. Baireuther, M. Caio, B. Criger, C. W. Beenakker, and T. E. O’Brien, Neural network decoder for topological color codes with circuit level noise, New Journal of Physics 21, 013003 (2019).
[34] C. Lin and G. Yang, Concatenated pieceable fault-tolerant scheme for universal quantum computation, Physical Review A 102, 052415 (2020).
[35] D. Poulin, Optimal and efficient decoding of concatenated quantum block codes, Physical Review A 74, 052333 (2006).
[36] E. Nahid, M. S. Zaman, and M. Sedghi, Low-overhead code concatenation approaches for universal quantum computation, arXiv preprint arXiv:1707.00981 (2017).
[37] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics) (Springer-Verlag New York, Inc., 2006).
[38] Y. Lecun, Y. Bengio, and G. Hinton, Deep learning, Nature 521, 436 (2015).
[39] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Physical Review A 70, 052328 (2004).
[40] D. Wecker and K. M. Svore, Liquid >> A software design architecture and domain-specific language for quantum computing, arXiv preprint arXiv:1402.4467 (2014).