

## Research Article

# Quantum Information Protection Scheme Based on Reinforcement Learning for Periodic Surface Codes

Yun-Jia Xue,<sup>1</sup> Hao-Wen Wang,<sup>1</sup> Yan-Bing Tian,<sup>1</sup> Yi-Nuo Wang,<sup>2</sup> Yu-Xuan Wang,<sup>3</sup> and Shu-Mei Wang<sup>2</sup> 

<sup>1</sup>School of Information and Control Engineering, Qingdao University of Technology, Qingdao, China

<sup>2</sup>School of Sciences, Qingdao University of Technology, Qingdao, China

<sup>3</sup>School of Electronic and Information Engineering, Soochow University, Suzhou, China

Correspondence should be addressed to Shu-Mei Wang; wangshumei@qut.edu.cn

Received 14 April 2022; Revised 1 June 2022; Accepted 11 June 2022; Published 18 July 2022

Academic Editor: Liuguo Yin

Copyright © 2022 Yun-Jia Xue et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Quantum information transfer is an information processing technology with high speed and high entanglement with the help of quantum mechanics principles. To solve the problem of quantum information getting easily lost during transmission, we choose topological quantum error correction codes as the best candidate codes to improve the fidelity of quantum information. The stability of topological error correction codes brings great convenience to error correction. The quantum error correction codes represented by surface codes have produced very good effects in the error correction mechanism. In order to solve the problem of strong spatial correlation and optimal decoding of surface codes, we introduced a reinforcement learning decoder that can effectively characterize the spatial correlation of error correction codes. At the same time, we use a double-layer convolutional neural network model in the confrontation network to find a better error correction chain, and the generation network can approach the best correction model, to ensure that the discriminant network corrects more nontrivial errors. To improve the efficiency of error correction, we introduced a double-Q algorithm and ResNet network to increase the error correction success rate and training speed of the surface code. Compared with the previous MWPM 0.005 decoder threshold, the success rate has slightly improved, which can reach up to 0.0068 decoder threshold. By using the residual neural network architecture, we saved one-third of the training time and increased the training accuracy to about 96.6%. Using a better training model, we have successfully increased the decoder threshold from 0.0068 to 0.0085, and the depolarized noise model being used does not require a priori basic noise, so that the error correction efficiency of the entire model has slightly improved. Finally, the fidelity of the quantum information has successfully improved from 0.2423 to 0.7423 by using the error correction protection schemes.

## 1. Introduction

In the recent years, noise generated by the operation of quantum computers will destroy the entanglement of quantum states, which is a problem that needs to be solved urgently. Quantum error correction [1, 2] is an effective means to protect the quantum information [3] from loss. To reduce the impact of quantum decoherence, error-correcting codes with topological properties are widely used. They can achieve good local stability, which provides feasibility for reversing or eliminating noise and errors in quantum systems.

The topological error correction codes are in the form of stabilizers [4]. Encoding the logical qubits in error-prone physical qubits facilitates us to find valid codes for error messages. Because the topology code information is stored in global degrees of freedom, a larger grid can provide a larger code distance. The threshold is an important representative of the performance of the physical qubits in line gate transmission. When the physical error rate is lower than a certain threshold, the quantum computers can suppress the logical error rate to a lower level by applying the quantum error correction schemes [5]. When the physical error rate of the logical qubit is better than that of the physical qubit, it is

called the pseudo-threshold. We encode the physical qubits with different code distances into logical qubits and obtain the decoding threshold by suppressing the logical error rates [6]. We need to infinitely approach the pseudo-threshold to the decoding threshold to achieve the best fault-tolerant effect. When the error rate is low enough, increasing the code distance will increase the probability of success in error correction. However, when the error rate is high, increasing the code distance will reduce the probability of successful error correction. Under the depolarization noise model, [7] the topological error correction code we want to choose is a surface code with generalized constraints [8, 9] (commutability and periodicity), which is convenient for further research on the properties of topological error correction codes. Since the surface code is not self-correcting, it will have defects when it is affected by the noise. Therefore, we must actively diagnose and correct errors, and the decoding algorithm plays an important role in it. The syndrome in the surface code can check a group of physical qubits in the form of parity checks to detect errors that have occurred. An algorithm that provides a set of recovery operations for correcting errors for a given syndrome is called a decoder and it provides suggestions for correcting any errors that may occur during the calculation [10]. Since the errors detected by the syndrome are not unique, the decoder needs to combine the error statistics detected by any syndrome. The decoder generates a correction chain for this information to eliminate the syndromes in the least likely situations to cause new errors.

To solve the problem of spatial correlation [11, 12] of the surface codes, we adopt a confrontation network structure of the reinforcement learning mechanism, which is an improvement of the neural network in the deep Q network (DQN) [13], which can help us better to find the best correction chain. The logic error rate is reduced by increasing the number of flipped bits of the correction chain [14], which solves the problem that there is a large gap between the pseudo-threshold and the decoder threshold and cannot be better fault-tolerant. Therefore, the confrontation network structure of the reinforcement learning mechanism can be used to solve the problem of low-pseudo-thresholds of the surface code [15].

The sections of this paper are organized in the following manner. First, Section 2 introduces the concept and properties of surface codes. Then, Section 3 explains the decoding steps of the dueling network and the double-Q algorithm. Next, Section 4 outlines the training process. The decoding performance results of different decoders and the analysis of threshold and error rates are shown in Section 5. Finally, conclusions are drawn as discussed in Section 6.

## 2. Background

The surface code is a planar variant of the Kitaev code, [16] which requires fewer quantum bits to achieve the same effect of the error correction strength. The model of the surface code is the quantum double model  $(D(Z_2))$ . It can be regarded as a quantum double model  $D(G)$  [17] that defines a general finite group  $G$  on a general two-dimensional lattice.

This paper chooses to use a surface code with a square lattice representation and chooses the group  $G$  as the abelian group  $Z_2$  [18]. The codespace can then be defined by the parity operator [19] acting on the four nearest quantum bits on the square lattice [20].

The surface code is chosen as a square lattice with the shape  $L \times L$  and the square contains  $L^2$  quantum bits. The quantum bits are placed at each lattice vertex, and the four points adjacent to the middle part are a lattice, and each logical quantum bit is encoded by these data quantum bits. After spinning the lattice by  $1/2$ , it is possible to form logical quantum bits by using a minimum number of physical quantum bits [21]. Some of the quantum bits are still located in the lattice before rotation, and the new quantum bits are defined as the auxiliary quantum bits. For the selected geometry, the surface code encodes logical quantum bits as  $L^2$  physical quantum bits and  $L^2 - 1$  ancilla qubits, and each stabilizer measurement [15] cycle can identify and correct up to  $\lfloor (L - 1)/2 \rfloor$  errors. We mark the edge as  $e_i$ , and we can represent any edge set as the formal sum  $s = \sum_i s_i e_i$ , where  $i$  exists on the edge of the qubit or lattice. When  $e_i$  is not in the edge set,  $s_i$  is 0, and when  $e_i$  is in the edge set,  $s_i$  is 1. We define a qubit on each edge, therefore each element of the calculation basis can be defined as follows:

$$|s\rangle := \otimes |s_i\rangle, \quad s \in S. \quad (1)$$

The form sum of any set of edges is  $s$  and all are in the edge abelian group  $S$ .

The data quantum bits are placed on the vertices of the lattice, as shown in Figure 1(a). The four internally adjacent vertices form two types of lattice operators,  $X$  and  $Z$ , while the boundary is formed by two vertices, as shown in Figure 1(b). In a lattice of surface codes with periodic boundaries, the four data quantum bits surrounding the  $Z$  auxiliary quantum bits form a  $Z$  stabilizer, and performing the four  $Z$  operations does not change the parity and therefore does not lead to a change in the corrector, and the  $X$ -stabilizer remains the same as shown in Figure 1(c). The blue circle corresponds to the stabilizer of the  $Z$  operator, and the yellow circle indicates the stabilizer of the  $X$  operator. The stabilizer performs a series of CNOT gates in the same manner as shown in Figure 1(d), to entangle each auxiliary quantum bit with its four neighboring data quantum bits. Depending on the representation chosen, the lattice stabilizer [22, 23] can be defined as follows:

$$S_p = \otimes_{i \in p} \sigma_i, \quad (2)$$

$$\sigma_i = \begin{cases} X_i & \text{for } p \in B_p \\ Z_i & \text{for } p \in W_p. \end{cases}$$

Since all the stabilizers commute with each other, the  $A_T$  and  $B_U$  form the stabilizer operator in the stabilizer code. All these operators are Hermitian with eigenvalues of  $\pm 1$  on different lattices of the matrix. The commutative Hamiltonian  $H_p$  is constructed under the topological stabilizer code.

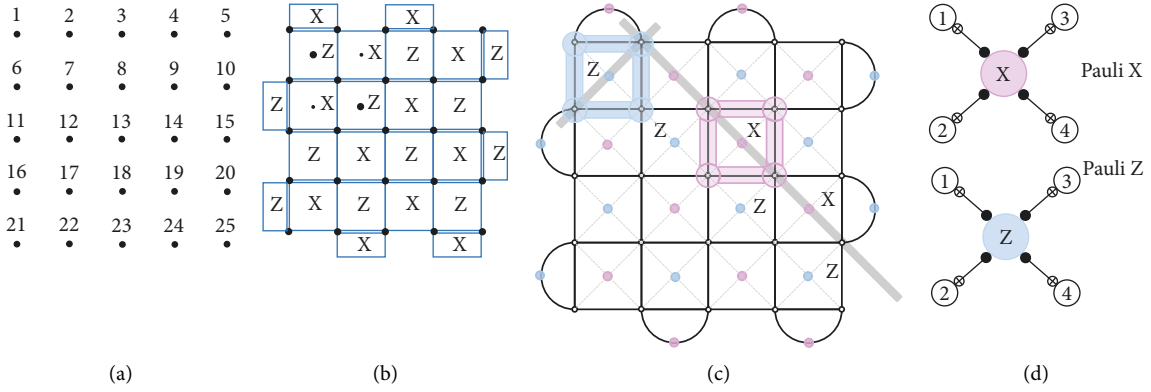


FIGURE 1: (a) The data qubit corresponds to the vertex of the lattice. (b) Qubits form two types of lattice operators: X and Z. (c) The four data quantum bits surrounding the X(Z) ancilla qubits form an X(Z) stabilizer. (d) Ancilla qubits the CNOT gate execution method.

$$H_P = -\sum_T X_p - \sum_U Z_p. \quad (3)$$

For the surface codes under the depolarization noise model generates equal probability of X, Y, and Z errors that lead to a bit-flip X or phase-flip Z of quantum bits, [24] so the ancillary qubits are placed in the middle of each lattice. Measuring the ancilla qubits only yields +1 or -1 eigenvalues, which can correspond to the parity of four (or two in the boundary lattice) adjacent data quantum bits of the internal lattice. The set of ancilla measurements is called a syndrome [25, 26]. The introduction of ancilla qubits ensures that no information loss occurs in the measurement of neighboring data quantum bits while performing the syndrome submeasurement. The intersection of the syndromes can help identify the most likely error operators. The task of the decoder is to find errors on the data qubits from the error correction subset. Due to the strong spatial correlation between the vertex operator and the lattice operator, it is difficult to use the statistical mapping method to perform an optimal correction chain. However, dueling networks [27] under reinforcement learning can solve this problem well.

### 3. Models and Algorithms

Reinforcement learning is a learning mechanism that seeks to solve the problem that the agent can make the corresponding action in a certain environmental state to get maximum reward [28]. The confrontation network is a network structure of the reinforcement learning, which is an improvement in the structure of the neural network in the deep Q network (DQN), which can make the system the fastest and the best close to the real maximum return.

**3.1. Finding the Optimal Error Correction Chain.** Our syndrome can be regarded as the next action that the agent will perform in the state space environment. The search for the optimal error correction chain means that a trivial ring correction chain or a nontrivial ring [29] correction chain is finally generated after a series of bit reversal or phase reversal operations (as action behaviors). The generation of a

nontrivial ring means that the system gets the smallest cumulative reward corresponding to the characteristic value of output -1, whereas the generation of the trivial ring opposite the system gets the largest reward corresponding to the characteristic value of output +1 as shown in Figure 2.

An action corresponding to the agent changes the current environment state  $s_0 \rightarrow s_1$ , accumulates rewards for this process  $r_1$ , and accumulates maximum return after a series of actions  $R = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots$ ; among them,  $\gamma$  is the discount rate, [30, 31] affected by noise and other factors, and the attenuation of the reward corresponds to the increase of Pauli operators such as X and Y during the correction process [32]. The specific operation process of the next dueling network is as follows: we define bit flip or phase flip as an action-value function.

$$Q_\pi(s, a) = E[U_t | S_t = s, A_t = a], \quad (4)$$

where  $Q_\pi(s, a)$  is the action-value function, which refers to the optimal correction chain that we expect to get after we know the error type and syndrome layout, and  $\pi$  is the strategy function, according to the current error state, the corrections give further correction instructions.  $U$  is the return,  $S$  is the state space set, and  $A$  is the action set. The optimal action value  $Q_*$  is defined as follows:

$$Q_*(s, a) = \max_{\pi} Q_\pi(s, a), \quad \forall s \in S, a \in A. \quad (5)$$

To eliminate the existence of different syndromes resulting in the return of  $\pi$  after the strategy selection (correction chain) is lower than the optimal value function. it eliminates the strategy function  $\pi$  to ensure that the optimal correction chain can be obtained. The state value function  $V_\pi(s)$  is the expectation of  $Q_\pi(s, A)$  about  $a$ :

$$V_\pi(s) = E_{A \sim \pi}[Q_\pi(s, A)]. \quad (6)$$

The definition of the optimal state value function  $V_*$  is

$$V_*(s) = \max_{\pi} V_\pi(s), \quad \forall s \in S. \quad (7)$$

The definition of the optimal advantage function is

$$D_*(s, a) \triangleq Q_*(s, a) - V_*(s). \quad (8)$$

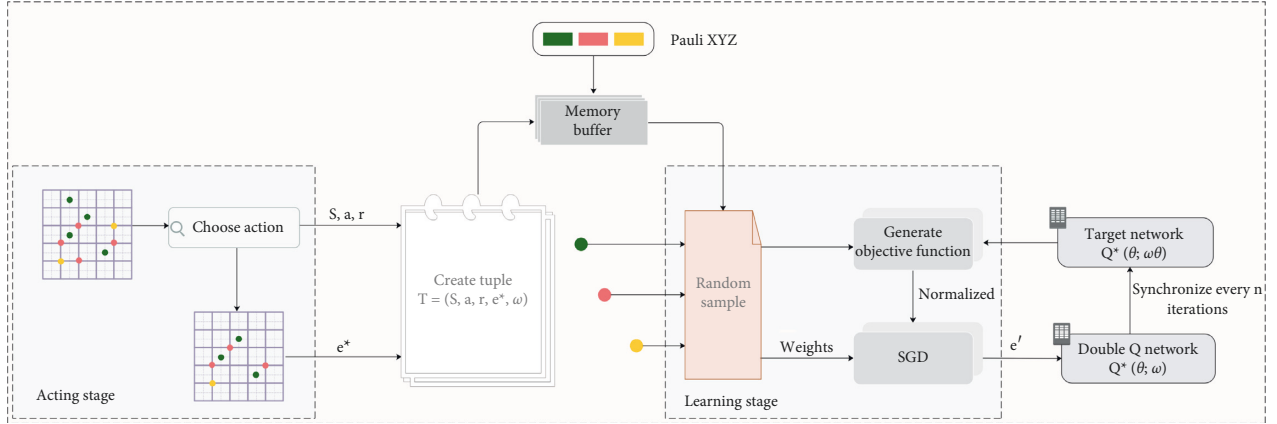


FIGURE 2: The confrontation network training inputs the syndrome to the convolutional layer, decodes the output feature vector, and then enters into the double-layer fully connected network to optimize the optimal action value, and the two combine to generate the optimal function value.

The duel network consists of two neural networks, one is denoted as  $D(s, a; \omega^D)$ , which is an approximation to the optimal advantage function  $D_*(s, a)$ . The other is denoted by the neural network  $V(s; \omega^V)$ , and it is an approximation of the optimal action-value function  $V_*(s)$ . Thus, we derive the optimal action-value function [19, 33].

$$Q(s, a; \omega) \triangleq V(s; \omega^D) + D(s, a; \omega^D) - \max_{a \in A} D(s, a; \omega^D). \quad (9)$$

Among them,  $Q(s, a; \omega)$  on the left side of the formula is the duel network we want, which is the optimal action-value function  $Q_*$ . Approximately, its parameters are recorded as  $\omega \triangleq (\omega^V; \omega^D)$ . The optimal action value is the optimal error we want to find in the correction chain as shown in Figure 2, the state space (syndrome) [34] is the input to the convolutional neural network and the feature vector is the output after the convolution operation, and the fully connected network is further used for multilayer connection to output multiple error chains [35] that we want to correct and finally output in the form of a feature vector [36, 37] to find the optimal error correction chain we need.

**3.2. Double-Q Learning Algorithm.** We use a duel network in the double-Q learning algorithm [38] to increase the number of flipped bits of the error correction chain and to reduce the error rate threshold [39]. The double-Q learning abbreviated as DDQN is the optimization of the Q learning algorithm; it can solve the bootstrap bias [40] that Q learning is difficult to solve, alleviate the overestimation problem caused by maximization, and better solve the problem of our optimization threshold which is too high. We use the fully connected network structure in the confrontation network to encode the syndrome of the surface code and use the DDQN algorithm to optimize the value of the action (bit-flip and phase-flip) and use the convolutional neural network decoder [41] to decode the output eigenvalues to gradually find the correction chain that we want to restore.

The DDQN algorithm will randomly take out a state syndrome from our system space each time, as represented

by a quadruple  $(s_i, a_i, r_i, s_{i+1})$ , in the decoding process we need to forward the syndrome to get

$$\hat{q}_i = Q(s_i, a_i; \omega_{now+}). \quad (10)$$

Among them,  $\omega_{now+}$  is the DQN parameter, select action (corresponding bit flip operation)

$$a^* = \underset{a \in A}{\operatorname{argmax}} Q(s_{i+1}, a; \omega_{now+}). \quad (11)$$

After performing a flip, the new syndrome is

$$\widehat{q}_{i+1} = Q(s_{i+1}, a^*; \omega_{now-}). \quad (12)$$

Then, do backpropagation to DQN to get the gradient  $\nabla_{\omega} Q(s_i, a_i; \omega_{now+})$ . Update and optimize the number of flips for gradient descent

$$\omega_{new+} \leftarrow \omega_{now+} - \alpha \cdot \delta_i \cdot \nabla_{\omega} Q(s_i, a_i; \omega_{now+}). \quad (13)$$

In DDQN, we set the rate of return  $\gamma \in (0, 1)$  and update the parameters of the CNN network by weighted average to get the best syndrome

$$\omega_{new-} \leftarrow \gamma \cdot \omega_{now+} + (1 - \gamma) \cdot \omega_{now-}. \quad (14)$$

The abovementioned algorithm flow can be summarized as shown in Figure 3 and Algorithm 1.

## 4. Training

The syndrome (agent) can be chosen to add to the lattice of defects in any direction (left, right, up, or down), which corresponds to flip one of the physical qubits on the lattice containing the defect. Figure 3 is divided into the action phase and the learning phase. The training starts from the action phase, [42] in which the state of the syndrome is sent to the agent (a two-dimensional matrix is formed). The syndrome uses a double-Q network to propose the defect (error position) state  $S$  and action  $a$ . The agent uses a greedy strategy, which means that it will suggest the operation with the highest Q value with a probability of  $(1 - \epsilon)$ . Otherwise,

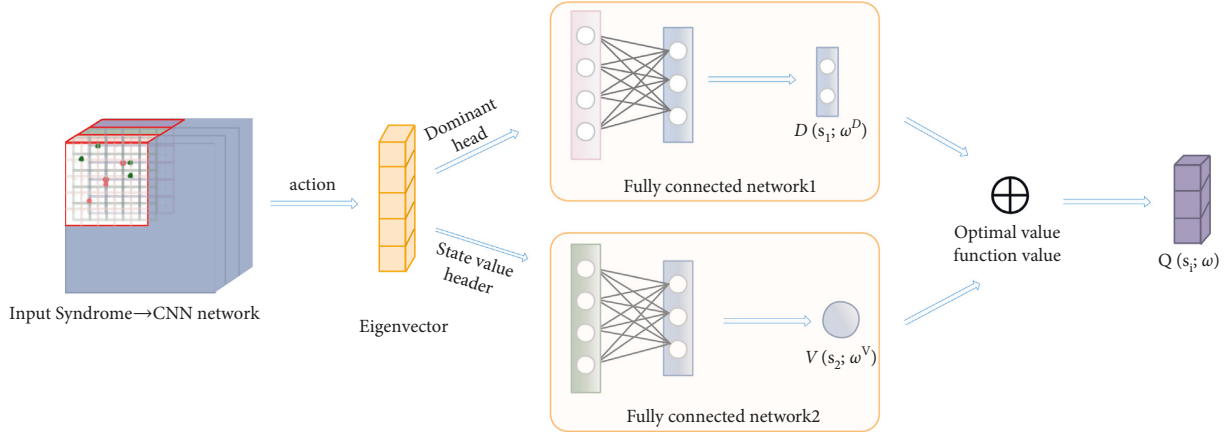


FIGURE 3: From the initial action selection stage, through the buffer pool, samples are randomly selected for reinforcement learning, and SGD is used to minimize the mean square error of the true value and the predicted value. Then, we use the double-Q algorithm to perform multiple iterations on the syndrome, and finally, we get the error correction chain that is best close to what we want.

random measures are recommended. The action is to cause the reward  $r$  and the new observation value  $Q'$  derived from the obtained syndrome at the defect  $e$  corresponding to  $S$ . The network training of this reinforcement learning utilizes the DDQN algorithm and reimplments the technology based on experience to store the new correction chain obtained by the flipping action after the syndrome row is completed in the form of a binary array in the buffer. When we update the parameter  $\omega$  of the confrontation network, we randomly sample a small batch of samples from the buffer to randomly obtain our experience samples. In order to correct the correlation of the training data with the gradient descent and to minimize their correlation, the training accuracy of our neural network can be made more accurate. To improve the training speed of the model, and to ensure the efficiency of the model training under a larger dataset, the ResNet network [43] layer is introduced as the underlying architecture, which is known as the “shortcut,” and it can ensure that a large number of stacks can be stacked without reducing the learning efficiency convolutional layer by adding the shortcut output to the output of the stacked-layer through the residual block and by using the ResNet7, 14, and 21 network layers for data training. In order to ensure the integrity of the data, we also need to periodically synchronize the parameters of the fully connected network in the confrontation network with the parameters of the CNN network.

The training starts with the input of the syndrome and by randomly selecting a reversal action with a probability of  $\epsilon$ . It records the action  $a$  and the defect (error center). The network uses a greedy algorithm to pregenerate the probability of the value action with the highest return  $(1 - \epsilon)$ . This action will generate a reward  $r$ , a new state environment  $s_1$ , a new defect center  $e^\Lambda$ , and a new syndrome. The whole process of DDQN algorithm training is to input the quadruple  $(r, s, e^*; \omega)$  in binary form into the buffer for temporary storage and then perform gradient descent to minimize the correlation of the data. Learn  $\{T_i = (r, s, e^*; \omega)\}_{i=1}^N$ , where  $N$  is the size of a given batch. Here is the optimal value of the network training target

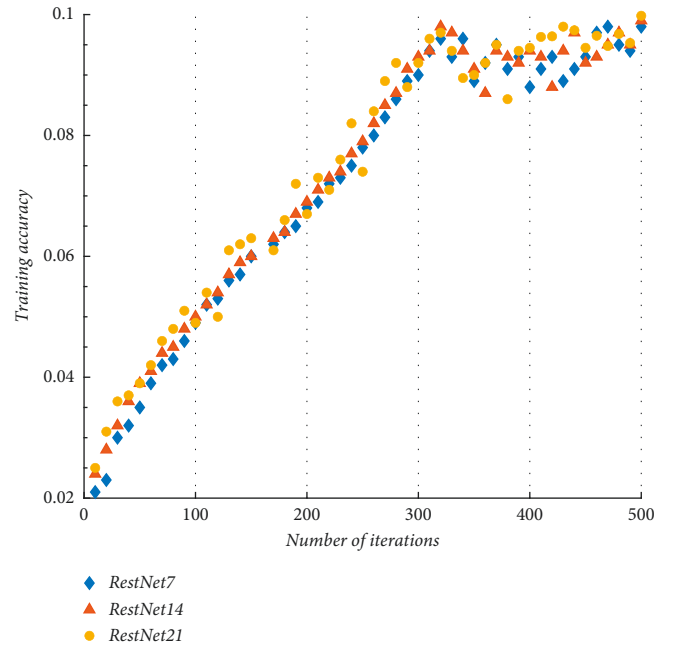


FIGURE 4: The 7 layers (marked by blue circles), 14 layers (marked by red triangles), and 21 layers (marked by yellow circles) of the ResNet network are used as the medium training models. After increasing the number of training syndromes, the correction accuracy can be improved to over 95%. The effective error rate function is generated according to the number of iterations.

$$\hat{y}_i = r_i + \gamma \max_{a \in A} \bar{Q}(r, s, e^*; \omega), \quad (15)$$

where  $\gamma$  is the reward factor, in which the target network is changed by the parameter  $\omega$  to predict the future cumulative reward (different corrected trivial chain and nontrivial chain). Afterwards, the gradient descent method is used to minimize the difference between the sample value and the double-Q network value, according to  $-\nabla \sum_{i \in N} (\hat{y}_i - \bar{Q}(r, s, e^*; \omega))^2$ , normalization improves the training parameters of the network. Then, train a new

- (1) **While** the original syndrome defect still exists **do**
- (2) The syndrome is temporarily stored in the buffer pool
- (3) Randomly select the samples from the buffer pool:  $\text{Data} = \{x_1, x_2, x_3 \dots\}$
- (4) Calculate  $Q(s, a; \omega)$  using the dual-Q network for all perspectives  $s \in \text{Data}$
- (5) Choose which defect  $\mathcal{S}$  to move with the action  $\mathcal{A}$  using the experience reapplication technology
- (6) Use the neural network in the duel network to find the optimal weight:  $\omega \rightarrow \omega_{\text{weight}}$  to equation (14).
- (7) The feature vector into the fully connected network to get the target network:  $Q^*(\theta; \omega^\theta)$
- (8) SGD gets the optimal dual-Q network after normalization:  $Q^*(\theta; \omega)$
- (9) Get the new syndrome and store it in the quadruple:  $T = (S, a, e^*; \omega)$
- (10) **for each** transition tuple  $T_i$  in the sample **do**
- (11) Construct targets  $y_i$  using the target network  $\theta_T$  and reward  $\gamma_i$  to equation (16).
- (12) **end for**
- (13) Update dual-Q network parameters  $\theta$
- (14) Every  $n$  iterations, synchronize the target network with the network setting of  $\theta_T = \theta$
- (15) **end while**

ALGORITHM 1: Training the reinforcement learning agent decoder.

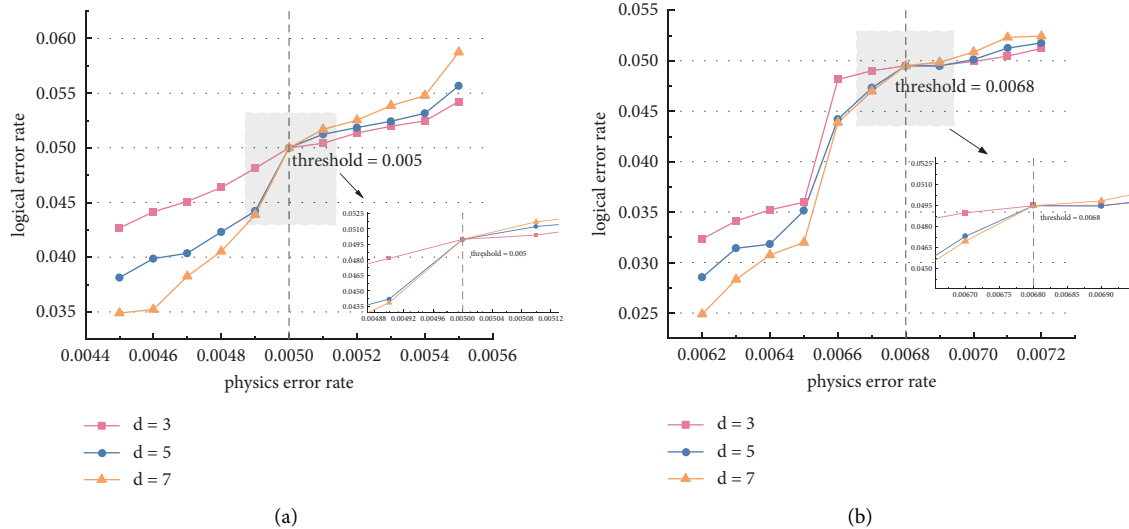


FIGURE 5: We used two decoders at  $d=3$  (marked with pink squares),  $d=5$  (marked with blue circles), and  $d=7$  for the ResNet network (marked with yellow triangles) as training models. (a) The variation of the logical error rate of the MWPM decoder for different  $d$  values, and the zoomed-in plot shows the variation of the threshold at around 0.005. The logical error rate increases gradually with the increase of the physical error rate. (b) The variation of the CNN decoder logical error rates for different  $d$  values with a threshold value of 0.0068.

sequence and synchronize the weight of the target network with the dual-Q network  $\omega_T$  at a specified rate. Finally, use the ResNet network architecture in the CNN network and perform multiple iterations and predictions [7] and simultaneously synchronize the parameters of the regular fully connected network and the CNN network. The training process is shown in Figure 4, in order to achieve our ideal accuracy of more than 95%, because the network model of ResNet21 can achieve the accuracy we want under lower parameters, but for the 7-layer and 14-layer models, we have to increase the training parameters, and finally, after ensuring that the iteration depth reaches 500 times, the accuracy of the three can reach more than 95%, which ensures that the generation of the corrected trivial chain with a code distance of 5 can be accurately predicted under the noise intensity of 0.042.

## 5. Simulation Analysis

We use the DDQN algorithm in the confrontation network to generate our prediction model. The efficiency and accuracy achieved by different network layers are also different. The dataset is shown in Table 1. We readded the MWPM comparison again. It can be seen that the number of training steps and parameters have been reduced by 2 data level and 1 data level, respectively, in Table 1.

As can be seen in Figure 5(a), the logic error rate under the MWPM decoder with a smaller code distance shows a slow rise, where the pseudo-threshold of  $d=7$  is closest to the decoder threshold of 0.0050 obtained in the gray region as shown in the small figure, at which the physical quantity quantum bits can be well tolerated under noise interferences, but the overhead is too large and the threshold is too low



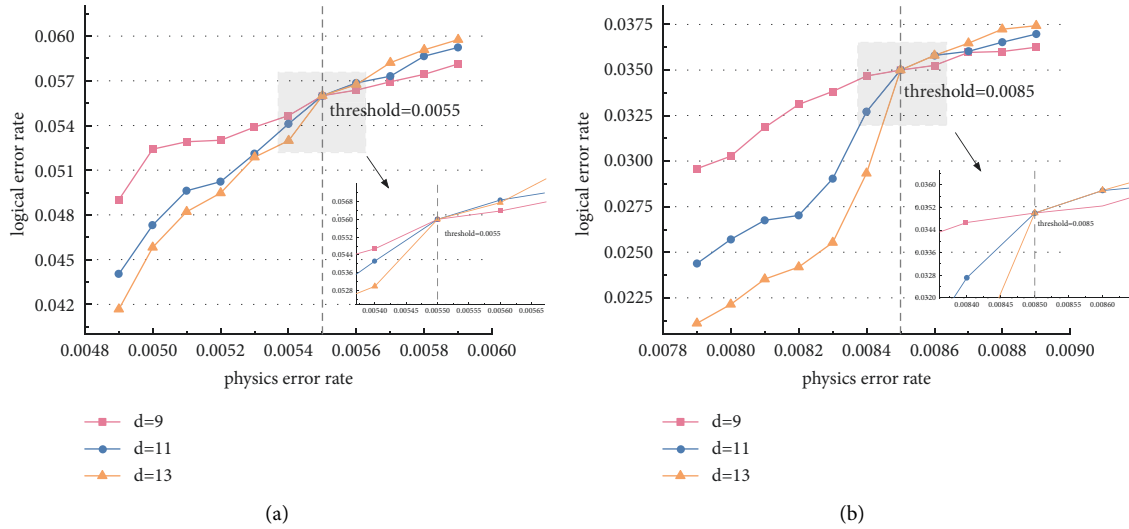


FIGURE 6: We used two decoders at  $d = 9$  (marked with pink squares),  $d = 11$  (marked with blue circles), and  $d = 13$  for the ResNet network (marked with yellow triangles) as training models. (a) The variation of the logical error rates of the MWPM decoder for different  $d$  values, and the zoomed-in plot shows the variation of the threshold at around 0.0055. The logical error rate increases gradually with the increase of the physical error rate. (b) The variation of the CNN decoder logical error rate for different  $d$  values with a threshold value of 0.0085.

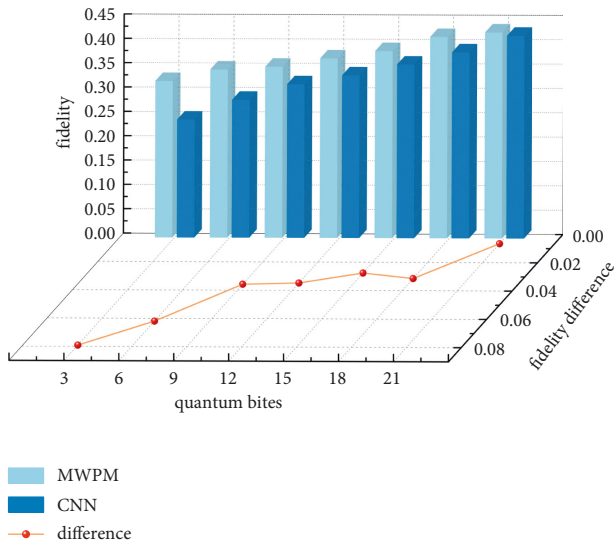


FIGURE 7: Using the MWPM decoder with a threshold of 0.005 (indicated by light blue bars) and the CNN decoder (indicated by dark blue bars) as training models, the fidelity at quantum bits of 3, 6, 9, 12, 15, 18, and 21 and the fidelity difference curves are determined.

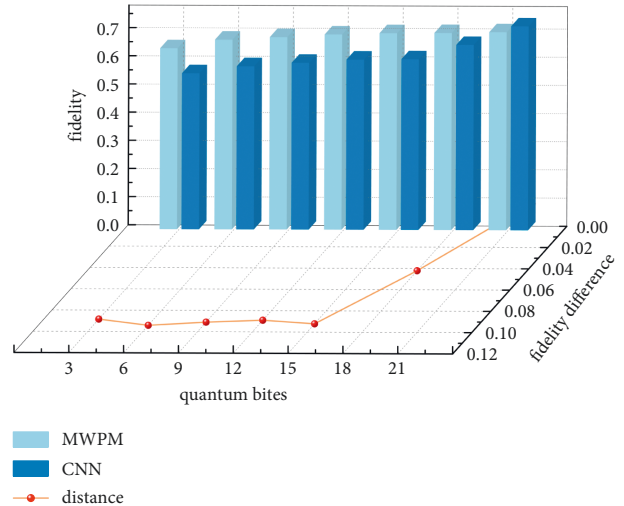


FIGURE 8: Using the MWPM decoder with a threshold of 0.0085 (indicated by light blue bars) and the CNN decoder (indicated by dark blue bars) as training models, the fidelity at quantum bits of 3, 6, 9, 12, 15, 18, and 21 and the fidelity difference curves are determined.

leading to huge cost/expenditures. After using CNN decoder, the pseudo-threshold of the same code distance  $d = 7$  as shown in Figure 5(b) is closer to the decoder threshold of 0.0055 in the gray area, although the pseudo-threshold is closer to the decoder threshold in dimension, but the threshold strength is still not high enough.

In order to further improve the decoder threshold, we expand the code distance as shown in Figure 6, and it is obvious that the logic error rate rises fast under the expanded code distance, and the pseudo-threshold of  $d = 9$  as shown in Figure 6(a) has far exceeded the pseudo-threshold

of  $d = 3$  and is closer to the decoder threshold and is increased to 0.0055; in addition, the decoder threshold as shown in Figure 6(b) has also increased to 0.0085. It is close to the perfect threshold effect, and the pseudo-threshold of  $d = 13$  has achieved the idealized approach, which largely solves the problem of the low threshold of the surface code and the errors leading to the poor error correction ability. It makes a reliable guarantee for the integrity of the quantum information transmission and improves the quantum information fidelity to the level of 0.7243.

After the decoder threshold has been significantly raised, this paper verifies the integrity of the quantum information

TABLE 1: The accuracy of the prediction model. Calculate the code distance 5 and use independent noise for training.

	Trainable data	Steps	Accuracy
MWPM	$1.42 \times 10^5$	$4.9 \times 10^4$	74.544%
ResNet 7	$1.32 \times 10^3$	$3.2 \times 10^3$	83.176%
ResNet 14	$2.76 \times 10^3$	$2.8 \times 10^3$	89.977%
ResNet 21	$3.76 \times 10^3$	$1.9 \times 10^3$	96.259%

transmission at the line guys, as shown in Figures 7 and 8, and the effect of quantum information fidelity produced under different decoders. As can be seen in Figure 7, the fidelity of the quantum bit information transmission under the MWPM decoder is somewhat lower than that of the CNN decoder, and in the case of the MWPM decoder, the threshold bit is 0.0055, but the fidelity only reaches 0.435, which fails to guarantee information transmission well. We placed the boosted decoder threshold into the new quantum bit transmission as shown in Figure 8 and it can be seen that the quantum information fidelity has been boosted to a high-performance fidelity of 0.754 under the CNN decoder, thus upgrading the quantum information to a new stage in the information transmission process against noise interference and achieving a breakthrough of raising the fidelity information to double.

## 6. Conclusion

In this paper, the surface code is selected as the quantum error correction code under the ideal error, and it has produced a good effect in the quantum error correction. To avoid the problem of excessively high prediction values, we have introduced a double-Q algorithm, which improves the decoder threshold to about 0.0055 compared to the MWPM's 0.005 error correction. Using the 14 and 21 layers of the ResNet network shortens the training time by 30%, while at the same time increases the training accuracy to about 96.6%. Using the dual-layer convolutional neural network model in the dual network, we succeeded in increasing the decoder threshold from 0.0068 to 0.0085, which has greatly improved the error correction efficiency of the whole model. However, from the simulation results in this paper, we can see that the factors affecting the transmission of quantum information are not only external noise interference and code spacing but also other external factors such as the possibility of stable measurement errors which can affect the quantum bits and can lead to errors. Not only the optimization of the decoder algorithm needs to get closer to the optimal threshold but also the development of precision instrumentation for quantum computers is essential. Therefore, we need to further investigate the quantum error correction mechanism to obtain better quantum information transmission results.

## Data Availability

Data are available on request.

## Conflicts of Interest

The authors declare no potential conflicts of interest.

## Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant No. 61772295), Natural Science Foundation of Shandong Province, China (Grant Nos. ZR2021MF049 and ZR2019YQ01), and Project of Shandong Provincial Natural Science Foundation Joint Fund Application (ZR202108020011).

## References

- [1] X. Xu, S. C. Benjamin, and X. Yuan, "Variational circuit compiler for quantum error correction," *Physical Review Applied*, vol. 15, no. 3, Article ID 034068, 2021.
- [2] A. G. Fowler, "Analytic asymptotic performance of topological codes," *Physical Review A*, vol. 87, no. 4, Article ID 040301, 2013.
- [3] A. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2–30, 2003.
- [4] W. Z. Zhang, Y. H. Ma, J. F. Chen, and C. P. Sun, "Quantum illumination assistant with error-correcting codes," *New Journal of Physics*, vol. 22, no. 1, Article ID 013011, 2020.
- [5] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, "Optimizing quantum error correction codes with reinforcement learning," *Quantum*, vol. 3, p. 215, 2019.
- [6] S. Varona and M. A. Martin-Delgado, "Determination of the semion code threshold using neural decoders," *Physical Review A*, vol. 102, no. 3, Article ID 032411, 2020.
- [7] D. Fitzek, M. Eliasson, A. F. Kockum, and M. Granath, "Deep Q-learning decoder for depolarizing noise on the toric code," *Physical Review Research*, vol. 2, no. 2, Article ID 023230, 2020.
- [8] H. Bombin, *An Introduction to Topological Quantum Codes*, <https://arxiv.org/abs/1311.0277>, 2013.
- [9] A. Fowler, *Optimal Compl Exity Correction of Correlated Errors in the Surface Code*, <https://arxiv.org/abs/1310.0863>, 2013.
- [10] R. Sweke, M. S. Kesselring, E. P. L. van Nieuwenburg, and J. Eisert, "Reinforcement learning decoders for fault-tolerant quantum computation," *Machine Learning: Science and Technology*, vol. 2, no. 2, Article ID 025005, 2020.
- [11] H. Bombin and M. A. Martin-Delgado, "Topological computation without braiding," *Physical Review Letters*, vol. 98, no. 16, Article ID 160502, 2007.
- [12] M. Sheth, S. Z. Jafarzadeh, and V. Gheorghiu, "Neural ensemble decoding for topological quantum error-correcting codes," *Physical Review A*, vol. 101, no. 3, Article ID 032338, 2020.
- [13] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, "Surface code quantum computing by lattice surgery," *New Journal of Physics*, vol. 14, no. 12, Article ID 123011, 2012.
- [14] M. A. Levin and X. G. Wen, "String-net condensation: a physical mechanism for topological phases," *Physical Review B*, vol. 71, no. 4, Article ID 045110, 2005.
- [15] C. K. Andersen, A. Remm, S. Lazar et al., "Repeated quantum error detection in a surface code," *Nature Physics*, vol. 16, no. 8, pp. 875–880, 2020.



- [16] J. R. Wootton and D. Loss, “High threshold error correction for the surface code,” *Physical Review Letters*, vol. 109, no. 16, Article ID 160503, 2012.
- [17] Z. A. Jia, Y. H. Zhang, Y. C. Wu, L. Kong, G. C. Guo, and G. P. Guo, “Efficient machine-learning representations of a surface code with boundaries, defects, domain walls, and twists,” *Physical Review A*, vol. 99, no. 1, Article ID 012307, 2019.
- [18] C. W. von Keyserlingk, F. J. Burnell, and S. H. Simon, “Three-dimensional topological lattice models with surface anyons,” *Physical Review B*, vol. 87, no. 4, Article ID 045107, 2013.
- [19] S. Bravyi, M. Suchara, and A. Vargo, “Efficient algorithms for maximum likelihood decoding in the surface code,” *Physical Review A*, vol. 90, no. 3, Article ID 032326, 2014.
- [20] H. Bombin, R. S. Andrist, M. Ohzeki, H. G. Katzgraber, and M. A. Martin-Delgado, “Strong resilience of topological codes to depolarization,” *Physical Review X*, vol. 2, no. 2, Article ID 021004, 2012.
- [21] D. Ristè, S. Poletto, M. Z. Huang et al., “Detecting bit-flip errors in a logical qubit using stabilizer measurements,” *Nature Communications*, vol. 6, no. 1, p. 6983, 2015.
- [22] N. P. Breuckmann and X. Ni, “Scalable neural network decoders for higher dimensional quantum codes,” *Quantum*, vol. 2, p. 68, 2018.
- [23] N. Ofek, A. Petrenko, R. Heeres et al., “Extending the lifetime of a quantum bit with error correction in superconducting circuits,” *Nature*, vol. 536, no. 7617, pp. 441–445, 2016.
- [24] S. Krastanov and L. Jiang, “Deep neural network probabilistic decoder for stabilizer codes,” *Scientific Reports*, vol. 7, no. 1, p. 11003, 2017.
- [25] G. Torlai and R. G. Melko, “Neural decoder for topological codes,” *Physical Review Letters*, vol. 119, no. 3, Article ID 030501, 2017.
- [26] A. G. Fowler, A. M. Stephens, and P. Groszkowski, “High-threshold universal quantum computation on the surface code,” *Physical Review A*, vol. 80, no. 5, Article ID 052312, 2009.
- [27] A. M. Steane, “Simple quantum error-correcting codes,” *Physical Review A*, vol. 54, no. 6, pp. 4741–4751, 1996.
- [28] A. N. Claudell, E. Waks, and J. M. Taylor, “Serialized quantum error correction protocol for high-bandwidth quantum repeaters,” *New Journal of Physics*, vol. 18, no. 9, Article ID 093008, 2016.
- [29] D. Nigg, M. Müller, E. A. Martinez et al., “Quantum computations on a topologically encoded qubit,” *Science*, vol. 345, no. 6194, pp. 302–305, 2014.
- [30] B. D. Clader, C. J. Trout, J. P. Barnes, K. Schultz, G. Quiroz, and P. Titum, “Impact of correlations and heavy tails on quantum error correction,” *Physical Review A*, vol. 103, no. 5, Article ID 052428, 2021.
- [31] E. Knill, “Quantum computing with realistically noisy devices,” *Nature*, vol. 434, no. 7029, pp. 39–44, 2005.
- [32] K. Fukui, A. Tomita, A. Okamoto, and K. Fujii, “High-threshold fault-tolerant quantum computation with analog quantum error correction,” *Physical Review X*, vol. 8, no. 2, Article ID 021054, 2018.
- [33] H. L. Huang, Y. Du, M. Gong et al., “Experimental quantum generative adversarial networks for image generation,” *Physical Review Applied*, vol. 16, no. 2, Article ID 024051, 2021.
- [34] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: towards practical large-scale quantum computation,” *Physical Review A*, vol. 86, no. 3, Article ID 032324, 2012.
- [35] A. Hutter, J. R. Wootton, and D. Loss, “Efficient Markov chain Monte Carlo algorithm for the surface code,” *Physical Review A*, vol. 89, no. 2, Article ID 022326, 2014.
- [36] H. W. Wang, Y. J. Xue, Y. L. Ma, N. Hua, and H. Y. Ma, “Determination of quantum toric error correction code threshold using convolutional neural network decoders,” *Chinese Physics B*, vol. 31, no. 1, Article ID 010303, 2021.
- [37] S. Varsamopoulos, B. Criger, and K. Bertels, “Decoding small surface codes with feedforward neural networks,” *Quantum Science and Technology*, vol. 3, no. 1, Article ID 015004, 2017.
- [38] P. Sarvepalli and R. Raussendorf, “Efficient decoding of topological color codes,” *Physical Review A*, vol. 85, no. 2, Article ID 022317, 2012.
- [39] N. Delfosse and N. H. Nickerson, “Almost-linear time decoding algorithm for topological codes,” *Quantum*, vol. 5, p. 595, 2021.
- [40] D. D. Yan, X. K. Fan, Z. Y. Chen, and H. Y. Ma, “Low-loss belief propagation decoder with Tanner graph in quantum error-correction codes,” *Chinese Physics B*, vol. 31, no. 1, Article ID 010304, 2022.
- [41] Y. Ueno, M. Kondo, M. Tanaka, Y. Suzuki, and Y. Tabuchi, “Qecool: on-line quantum error correction with a superconducting decoder for surface code,” in *Proceedings of the IEEE/ACM Design Automation Conference (DAC)*, pp. 451–456, San Francisco, CA, USA, 2021.
- [42] N. H. Nickerson, Y. Li, and S. C. Benjamin, “Topological quantum computing with a very noisy network and local error rates approaching one percent,” *Nature Communications*, vol. 4, no. 1, p. 1756, 2013.
- [43] M. Bukov, A. G. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, “Reinforcement learning in different phases of quantum control,” *Physical Review X*, vol. 8, no. 3, Article ID 031086, 2018.