

Global receptive field transformer decoder method on quantum surface code data and syndrome error correction

Ao-Qing Li(李熬庆), Ce-Wen Tian(田策文), Xiao-Xuan Xu(徐晓璇), Hong-Yang Ma(马鸿洋), and Jun-Qing Liang(梁俊卿)

Citation: Chin. Phys. B, 2025, 34 (**3**): 030306. DOI: 10.1088/1674-1056/adab63
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Quantum decoder design for subsystem surface code based on multi-head graph attention and edge weighting

Nai-Hua Ji(纪乃华), Hui-Qian Sun(孙汇倩), Bo Xiao(肖博), Ping-Li Song(宋平俐), and Hong-Yang Ma(马鸿洋)
Chin. Phys. B, 2025, 34 (**2**): 020309. DOI: 10.1088/1674-1056/ada1c7

Decoding topological XYZ^2 codes with reinforcement learning based on attention mechanisms

Qing-Hui Chen(陈庆辉), Yu-Xin Ji(姬宇欣), Ke-Han Wang(王柯涵), Hong-Yang Ma(马鸿洋), and Nai-Hua Ji(纪乃华)
Chin. Phys. B, 2024, 33 (**6**): 060314. DOI: 10.1088/1674-1056/ad342b

Recurrent neural network decoding of rotated surface codes based on distributed strategy

Fan Li(李帆), Ao-Qing Li(李熬庆), Qi-Di Gan(甘启迪), and Hong-Yang Ma(马鸿洋)
Chin. Phys. B, 2024, 33 (**4**): 040307. DOI: 10.1088/1674-1056/ad2bef

Performance of entanglement-assisted quantum codes with noisy ebits over asymmetric and memory channels

Ji-Hao Fan(樊继豪), Pei-Wen Xia(夏沛文), Di-Kang Dai(戴迪康), and Yi-Xiao Chen(陈一骁)
Chin. Phys. B, 2023, 32 (**12**): 120304. DOI: 10.1088/1674-1056/acf492

Approximate error correction scheme for three-dimensional surface codes based reinforcement learning

Ying-Jie Qu(曲英杰), Zhao Chen(陈钊), Wei-Jie Wang(王伟杰), and Hong-Yang Ma(马鸿洋)
Chin. Phys. B, 2023, 32 (**10**): 100307. DOI: 10.1088/1674-1056/acd8a9

Global receptive field transformer decoder method on quantum surface code data and syndrome error correction

Ao-Qing Li(李熬庆)[1], Ce-Wen Tian(田策文)[1], Xiao-Xuan Xu(徐晓璇)[1], Hong-Yang Ma(马鸿洋)[2], and Jun-Qing Liang(梁俊卿)[1,†]

[1] *School of Information and Control Engineering, Qingdao University of Technology, Qingdao* 266033, *China*
[2] *School of Sciences, Qingdao University of Technology, Qingdao* 266033, *China*

(Received 19 October 2024; revised manuscript received 25 December 2024; accepted manuscript online 17 January 2025)

Quantum computing has the potential to solve complex problems that are inefficiently handled by classical computation. However, the high sensitivity of qubits to environmental interference and the high error rates in current quantum devices exceed the error correction thresholds required for effective algorithm execution. Therefore, quantum error correction technology is crucial to achieving reliable quantum computing. In this work, we study a topological surface code with a two-dimensional lattice structure that protects quantum information by introducing redundancy across multiple qubits and using syndrome qubits to detect and correct errors. However, errors can occur not only in data qubits but also in syndrome qubits, and different types of errors may generate the same syndromes, complicating the decoding task and creating a need for more efficient decoding methods. To address this challenge, we used a transformer decoder based on an attention mechanism. By mapping the surface code lattice, the decoder performs a self-attention process on all input syndromes, thereby obtaining a global receptive field. The performance of the decoder was evaluated under a phenomenological error model. Numerical results demonstrate that the decoder achieved a decoding accuracy of 93.8%. Additionally, we obtained decoding thresholds of 5% and 6.05% at maximum code distances of 7 and 9, respectively. These results indicate that the decoder used demonstrates a certain capability in correcting noise errors in surface codes.

Keywords: quantum error correction, surface code, transformer decoder

PACS: 03.67.Pp, 03.67.–a

DOI: 10.1088/1674-1056/adab63 **CSTR:** 32038.14.CPB.adab63

1. Introduction

Quantum computers demonstrate significant advantages over classical computers in solving specific problems, such as integer factorization and quantum simulation,[1–3] by leveraging the properties of qubits, including superposition, entanglement and interference.[4,5] However, qubits are susceptible to environmental interference, leading to information loss. To address this, quantum error correction (QEC) techniques increase the number of physical qubits to reduce error rates.[6–8] Among these techniques, the two-dimensional surface code is widely used in fault-tolerant quantum computing due to its stability and high fault tolerance. These methods encode multiple physical qubits into a single logical qubit, with syndrome qubits used for error detection. During continuous measurements, changes in results are referred to as "syndrome signals", indicating the occurrence of errors.[9–12] However, the decoding process of QEC is computationally complex, necessitating the development of efficient and scalable decoders to support the practical application of large-scale quantum computing systems.

The surface code is a typical scheme in topological QEC codes proposed by Kitaev, and has become the mainstream choice for large-scale QEC. In recent years, with the development of techniques such as belief propagation, union-find and matrix product states, researchers have developed various efficient surface code decoders.[13–16] The standard minimum-weight perfect matching (MWPM) decoder simplifies quantum maximum likelihood decoding into a minimum matching problem.[17–19] Machine learning (ML) techniques, especially advances in neural networks, offer new approaches to improving decoder performance, as they can capture error correlations and adapt to different physical error models.[20–24] However, ML-based decoders face challenges such as convolutional neural networks (CNNs) having limited receptive fields, making it difficult to correct long error chains, and multilayer perceptrons (MLPs) being constrained by fixed input–output sizes.[25,26] In contrast, the transformer model, with its attention mechanism, demonstrates higher accuracy in handling long-range dependences and complex error patterns.[27] Transformer-based decoders using generative models significantly reduce time complexity when correcting logical operators.[28] Leveraging the transformer's ability to process inputs of arbitrary length, researchers have proposed an efficient transfer learning method that can decode codes of varying distances based on existing data.[29] Google's recursive transformer decoder has shown the potential of machine

†Corresponding author. E-mail: junqingliang@qut.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

learning in quantum decoding by maintaining high accuracy over long distances and multiple cycles, even under realistic noise simulations.[30]

Traditional MWPM decoding methods assume that bit-flip and phase-flip errors are independent. However, in quantum devices these errors are often correlated.[31,32] This paper considers the correlation between bit-flip and phase-flip errors and employs a phenomenological error model. Additionally, errors may occur not only in the data qubits but also in the syndrome qubits.[33,34] Therefore, our work investigates a transformer decoder based on an attention mechanism, which combined with the stabilizer properties of the surface code can handle errors in both data and syndrome qubits. The model consists of an encoder and a decoder that work together to process syndrome data. By implementing a self-attention mechanism, the model is capable of processing comprehensive information from all inputs, allowing for global information reception. The decoder utilizes the self-attention mechanism to comprehensively handle syndrome information, ensuring the spatial correlation of the error correction code, thus reducing the logical error rate and improving decoding accuracy. This is crucial for fault tolerance in large-scale quantum computing.

2. QEC codes and transformer architecture

2.1. Quantum surface code

The surface code is constructed using a two-dimensional lattice of local interactions to create an extensible qubit structure, relying on local interactions between qubits.[35–37] For example, in a distance-d surface code, there are $N = d^2$ qubits arranged in a square lattice, representing a two-dimensional array of qubits, as shown in Fig. 1(a). In this lattice, each face (red or blue) represents an odd-parity check operator, referred to as a stabilizer. These stabilizers are represented as X-type and Z-type stabilizers, used to form local Pauli operator products for adjacent qubits. For each red (blue) face, $\hat{X}$ ($\hat{Z}$) operators are used. The formulae are defined as follows:

$$\bar{A}_f = \prod_{j\in\partial f} \hat{X}_j; \quad \bar{B}_f = \prod_{j\in\partial f} \hat{Z}_j. \tag{1}$$

We define the stabilizer $\bar{S}_f$ as $\bar{A}_f$ when f is a red face and as $\bar{B}_f$ when f is a blue face. These stabilizers are measured by measuring each of the $N-1$ qubits adjacent to the qubit to be measured, with a measurement result of $+1$ indicating parity of the qubits. Since each red or blue face shares an even number of qubits, all stabilizer operations can be measured simultaneously. Two types of stabilizer measurement circuits are shown in Fig. 1(b).

The encoding of the surface code is realized by applying $N-1$ binary parity checks on the data qubits.[38] The eigenvalue $+1$ measurement results form the logical state space of the surface code. The logical state encoding is expressed as

$$|0_{\rm L}\rangle = N_d \prod_{f\in\text{blue}} \frac{1}{2}\left(1+\bar{A}_f\right)|0\rangle^{\otimes n}, \tag{2}$$

$$|1_{\rm L}\rangle = N_d \prod_{f\in\text{blue}} \frac{1}{2}\left(1+\bar{A}_f\right)|1\rangle^{\otimes n}. \tag{3}$$

Here, $|0\rangle^{\otimes n}$ and $|1\rangle^{\otimes n}$ represent the states where all physical qubits are in the $|0\rangle$ and $|1\rangle$ states, respectively. Encoding logical qubits means mapping from a two-dimensional Hilbert space to a 2^n-dimensional space: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \to |\psi_{\rm L}\rangle = \alpha|0_{\rm L}\rangle + \beta|1_{\rm L}\rangle$. The logical operators $\hat{X}^{\rm L}$ and $\hat{Z}^{\rm L}$ are defined by the product of Pauli X and Z operators on qubits along the left and top boundaries of the lattice, respectively: $\hat{X}^{\rm L} = \prod_{j\in\text{left}} \hat{X}_j$ and $\hat{Z}^{\rm L} = \prod_{j\in\text{top}} \hat{Z}_j$. The encoded logical operators must satisfy the following relationships: $\hat{X}^{\rm L}|0_{\rm L}\rangle = |1_{\rm L}\rangle$, $\hat{X}^{\rm L}|1_{\rm L}\rangle = |0_{\rm L}\rangle$, $\hat{Z}^{\rm L}|0_{\rm L}\rangle = |0_{\rm L}\rangle$, $\hat{Z}^{\rm L}|1_{\rm L}\rangle = -|1_{\rm L}\rangle$.

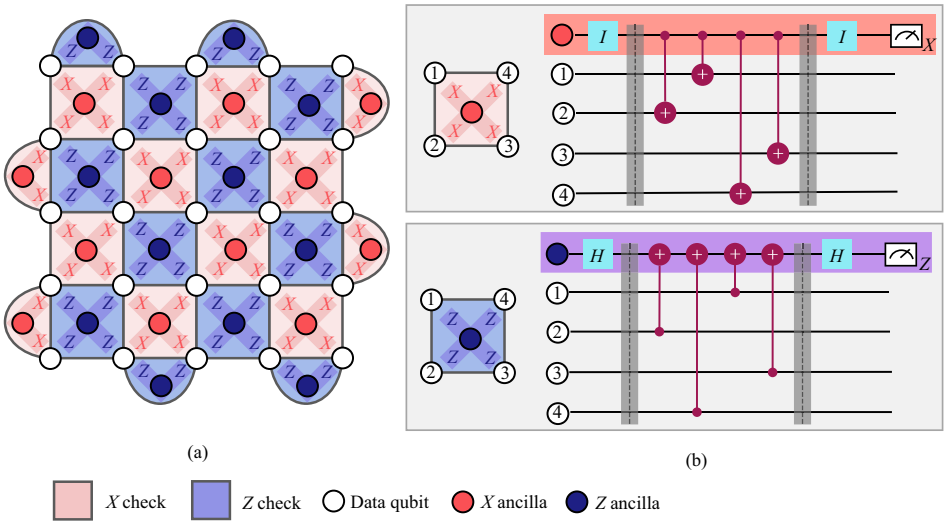


Fig. 1. (a) Here we consider a surface code with a code distance of $d = 5$, where data qubits are located at the lattice vertices and ancillary qubits are positioned at the centers of plaquette stabilizers. The red and blue squares represent stabilizers that check the X and Z parity of the qubits at the lattice vertices, respectively. (b) A single-cycle surface code quantum circuit that includes X-ancilla qubits and Z-ancilla qubits.

2.2. Transformer architecture

The core of the transformer is the multi-head self-attention mechanism, which effectively captures the relationships between elements in a sequence and is particularly suited for handling long-range dependences.[39,40] When addressing surface code decoding tasks, the transformer decoder can be analogized to natural language processing methods. Specifically, each stabilizer can be viewed as a word, while the aggregated data from each round of measurement correspond to a sentence. In this analogy, the decoding process is equivalent to a text classification problem. We generate embeddings for each stabilizer that are composed of the measurement and detection data from the current round, with positional encoding using sine embeddings to represent the stabilizer's spatial position within the code.

The multi-head self-attention module allows the model to jointly attend to information from different representational subspaces at different positions. By doing this, the transformer can more effectively integrate information from the sequence. By focusing attention on the same sequence as the input, we obtain the self-attention mechanism. The self-attention mechanism enables the model to attend to different parts of the input sequence when processing each word. The formula is defined as follows:

$$M_H(Q,K,V) = C_t(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O, \quad (4)$$

$$\text{where} \quad \text{head}_i = A_t(QW_i^Q, KW_i^K, VW_i^V). \quad (5)$$

Here, Q, K and V represent the query, key and value matrices, respectively. For each word, the transformer computes its importance score relative to other words. This approach enables the model to better capture contextual information and handle dependences between words that are far apart in the input sequence. Since transformers do not have a recurrent nature, they incorporate information about the positions of individual words in the sequence through positional encodings. These positional encodings are added to the input embeddings to preserve the order of the sequence. We can represent the attention score matrix as a normalized weighted accumulation function, defined as

$$A_{(i,j)} = \frac{\exp\left(\left(QW^Q\right)_{i,:} \cdot \left(KW^K\right)_{j,:}/\sqrt{d_k}\right)}{\sum_{k=1}^{n} \exp\left(\left(QW^Q\right)_{i,:} \cdot \left(KW^K\right)_{k,:}/\sqrt{d_k}\right)}, \quad (6)$$

where $(QW^Q)_{i,:}$ represents the vector at position i in the query matrix, $(QW^K)_{j,:}$ represents the vector at position j in the key matrix and $A_{(i,j)}$ is the attention score of position i with respect to position j.

3. Transformer-based framework for QEC

3.1. Error generation process

The purpose of the surface code is to protect the integrity of quantum states by detecting errors affecting logical qubits. By performing parity checks on the stabilizers, syndromes can be generated to determine the presence of errors. During the measurement process, qubits may experience bit-flip (X errors), phase-flip (Z errors) or a combination of both (Y errors), which affects the parity of neighboring stabilizers, resulting in a stabilizer measurement outcome of -1, as shown in Fig. 2(a). The probability of syndrome S_i occurring can be expressed using the following formula:

$$P(S_i = -1) = 1 - \prod_{\eta \in N(i)} (1 - p_\eta), \quad (7)$$

where S_i represents the measurement result of stabilizer i and -1 indicates that an error has been detected. p_η represents the error probability of the ηth qubit associated with stabilizer i. $N(i)$ represents the set of qubits on which stabilizer i acts. By observing changes in the parity of neighboring stabilizers, errors can be detected and located. For example, when a data qubit undergoes an X error, the parity of the two adjacent Z syndrome qubits changes; similarly, when a data qubit undergoes a Z error, the parity of adjacent X syndrome qubits changes. Therefore, when a data qubit experiences an error, the corresponding syndrome measurement results around it will change accordingly.[41,42]

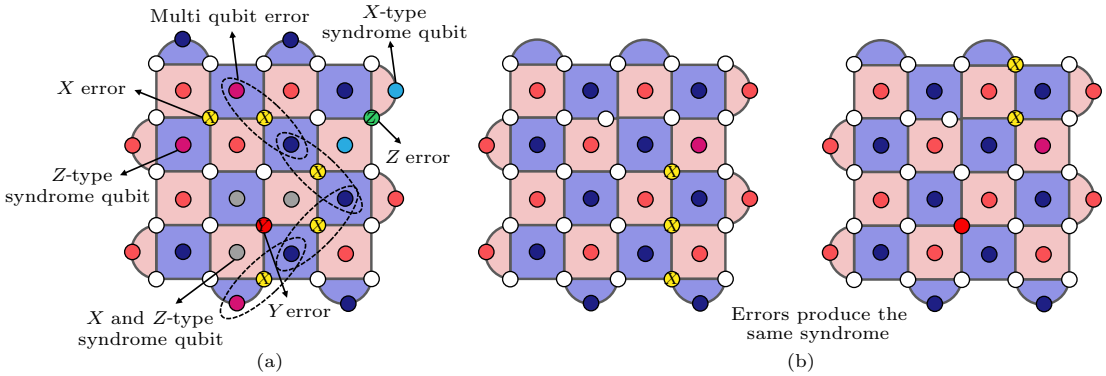


Fig. 2. (a) The surface code consists of data qubits and two types of syndrome qubits. The red X-type syndrome qubits are used to detect Z errors, while the blue Z-type syndrome qubits are used to detect X errors. When an error occurs in a data qubit, the adjacent syndromes may flip depending on the parity of the data qubit. If multiple errors occur, syndrome states with an even number of flips remain unchanged, whereas syndrome states with an odd number of flips will change. (b) In the surface code, different combinations of errors may result in the same syndrome pattern.

When multiple data qubits experience errors simultaneously, the syndrome patterns become more complex, increasing the difficulty of decoding. This is because the same measurement qubit may be shared by multiple data qubits, and different errors may produce the same syndrome, as shown in Fig. 2(b). We use conditional probability to describe the propagation of different error types and their impact on the syndrome, with the formula defined as follows:

$$P(S_j = -1 \mid E_i) = \sum_{k=1}^{n} \binom{n}{k} p_\eta^k (1-p_\eta)^{n-k} f_{i,j}(\eta), \tag{8}$$

where E_i is the ith type of error. Therefore, the mapping between syndromes and errors is not one-to-one. Additionally, performing dual Pauli operator operations on data qubits is equivalent to performing an identity operation. Therefore, when multiple qubit errors occur, it is necessary to consider odd- and even-parity flips, further increasing the complexity of the syndrome patterns.

3.2. Error correction cycle process

In QEC of surface codes, it is essential to consider not only errors in data qubits but also errors in syndrome qubits. During measurement cycles, measurement errors can cause changes in the sign of measurement results. To detect and correct these errors, the same qubit is typically measured repeatedly over multiple cycles.[43,44] During each measurement round, the measurement result M may contain errors. Let the true value of the tth measurement be $M_i^{(t)}$ and the erroneous measurement result be $\tilde{M}_i^{(t)}$.

The measurement error can be represented as

$$E_i^{(t)} = M_i^{(t)} \oplus \tilde{M}_i^{(t)}, \tag{9}$$

where $\oplus$ denotes the XOR operation. If $E_i^{(t)} = 1$, it indicates that a measurement error occurred in the tth round. By performing a majority vote on these measurement results, we can identify and correct measurement errors, with the final measurement result being given by

$$\hat{M}_i = \arg\max_{M} \prod_{t=1}^{T} P(\tilde{M}_i^{(t)} | M) P(M), \tag{10}$$

where T is the number of repetitions of the measurement. $P(\tilde{M}_i^{(t)}|M)$ represents the conditional probability of the tth measurement result given the true measurement result M. $P(M)$ is the prior probability. By analyzing the results of multiple consecutive measurements, single measurement errors can be effectively identified. The probability of a single measurement error is relatively low, and the probability of multiple consecutive measurement errors is even lower. Therefore, repeated measurements effectively suppress the impact of measurement errors, resulting in more reliable measurement outcomes, as shown in Fig. 3.

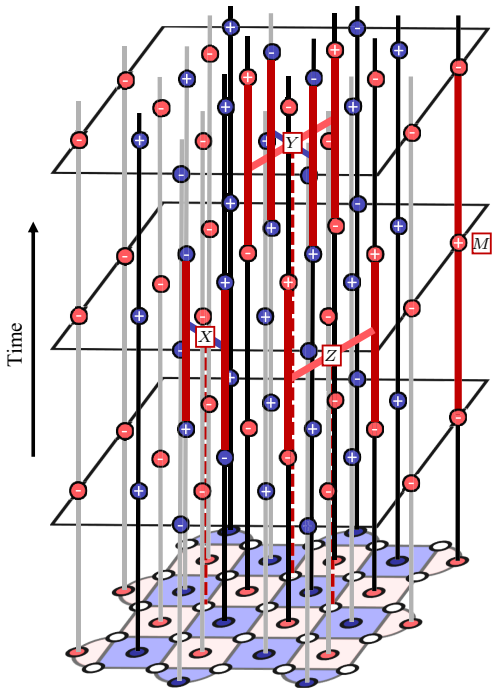


Fig. 3. The multi-round measurement process of the surface code error correction cycle. Time progresses upwards from the bottom of the array in the figure, with each horizontal plane representing a measurement step. The vertical thick red lines connect time steps where measurement outcomes have changed. Spatial correlations indicate X error, Z error or $Y = XZ$ errors. In practice, errors can also occur in the syndrome extraction circuit and the syndrome qubits, necessitating multiple rounds of measurements for decoding. On the right side, temporal correlations represent measurement errors (M), which occur sequentially over time, and measurement errors on the syndrome qubits can also lead to syndrome flips.

When a data qubit at position (x,y) experiences an X or Z error, the measurement results of the corresponding syndrome qubits will change, generating an error syndrome. Let the stabilizers before and after the measurement be $S_{x,y}^t$ and $S_{x,y}^{t+1}$, respectively. The error syndrome at the tth round is represented as

$$\sigma_{x,y}^{(t)} = \left(S_{x,y}^{(t)} \oplus S_{x,y}^{(t+1)}\right) \oplus \sum_{k=1}^{K} \alpha_k E_{x_k,y_k}^{(t)}. \tag{11}$$

If $\sigma_{x,y}^{(t)} = 1$, it indicates that an error has occurred at that position. α_k is used to represent the influence of different errors on the current error syndrome. $E_{x_k,y_k}^{(t)}$ represents the error occurring at position (x_k, y_k) during the tth round. The task of the decoder is to infer the errors on data qubits based on the observed error syndromes $\left\{\sigma_{x,y}^{(t)}\right\}$ and to determine the correction operators C. The correction operators can be represented as a set of operations $C = \prod_{(x,y)} \hat{E}_{x,y}$, where $\hat{E}_{x,y}$ is the correction operation calculated by the decoder. Since multiple data qubit configurations can produce the same error syndrome, the generation of error syndromes becomes an irreversible problem, making it difficult to precisely identify the actual error configuration. This issue represents a major challenge in the implementation of the decoder.

3.3. Transformer decoder model

The transformer is a multi-layer architecture constructed by stacking multiple transformer blocks. Each block includes a position-wise feedforward network, layer normalization and residual connections.[45–48] The input to a transformer model is typically in the form of a tensor with the shape $\mathbb{R}^B \times \mathbb{R}^N$, where B represents the batch size and N represents the sequence length. These data are first processed through an embedding layer, which converts one-hot encoded representations into dense embeddings of dimension d_{model}. After the conversion, the resulting tensor has a shape of $\mathbb{R}^B \times \mathbb{R}^N \times \mathbb{R}^{d_{\text{model}}}$. Subsequently, the embedding tensor is combined with positional encodings to preserve the sequential order of the data before being passed to the multi-head self-attention module. Finally, the output is passed to the position-wise feedforward network, with residual connections and layer normalization to facilitate signal propagation and learning. The entire process can be represented by the following formula:

$$X' = \text{LayerNorm}\left(\sum_{i=1}^{h} w_i \phi_1\left(F_{S_i}(X')\right) + \phi_2(X')\right) + \lambda X', \quad (12)$$

where F_S can be the multi-head self-attention mechanism or a position-wise feedforward network. $F_{S_i}(X')$ represents the result of the ith head in the multi-head attention mechanism. w_i is the weight of each attention head, satisfying $\sum_{i=1}^{h} w_i = 1$. Figure 4 shows the structure of the transformer.

To convert input data into a format that can be processed by a neural network, we map the surface code onto a $(d+1)\times(d+1)$ square lattice and add $2(d+1)$ virtual nodes to ensure that the syndrome qubits are located at intersections, as shown in Fig. 5. The virtual nodes remain set to zero during the decoding process. In the traditional surface code setup, there are usually d measurement rounds, but we add an additional layer, making the total number of measurement rounds $d+1$. Ultimately, the features of each syndrome from every round are stored in a $(d+1)$-dimensional lattice, with each node containing a feature vector of length six. The X and Z syndrome qubits are represented through separate channels, ensuring that the position and measurement results of each syndrome qubit are accurately utilized.

In the syndrome input encoding, each grid cell contains a feature vector of length six, denoted as

$$V_i^{(t)} = \sigma\Bigg(\left(\left[V_{X_i}^{(t)}, V_{Z_i}^{(t)}, D_{X_i}^{(t)}, D_{Z_i}^{(t)}, T_1^{(t)}, T_2^{(t)}\right] * W^{(t)}\right) + \sum_{j=1}^{k} \alpha_j V_j^{(t-1)} \cdot \beta_j\Bigg). \quad (13)$$

Here, $W^{(t)}$ represents the weight matrix of the convolution kernel and α_j and β_j are weight coefficients, representing the influence of historical temporal features on the current features. The first two channels represent the positions of X and Z syndrome qubits, while the third and fourth channels represent the syndrome states at different time points. The fifth and sixth channels define temporal boundaries, set to one in the first round and zero afterward, helping the network recognize the start and end of measurement sequences. Finally, the data qubits are measured on a specific basis to correct errors.

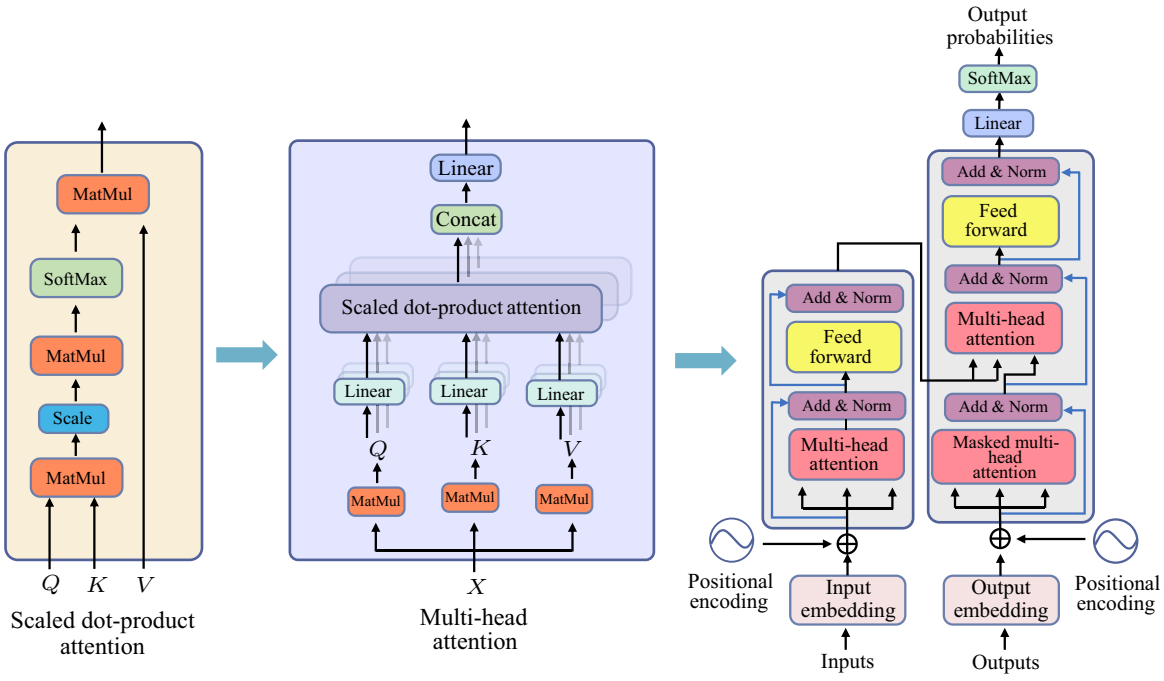


Fig. 4. The architecture of the transformer model, which includes an encoder and a decoder. The encoder consists of N identical layers, each containing two sub-layers: a multi-head self-attention mechanism and a fully connected feed-forward network. The decoder also consists of N identical layers, but with an additional sub-layer for masked multi-head attention to prevent positions from attending to subsequent positions.

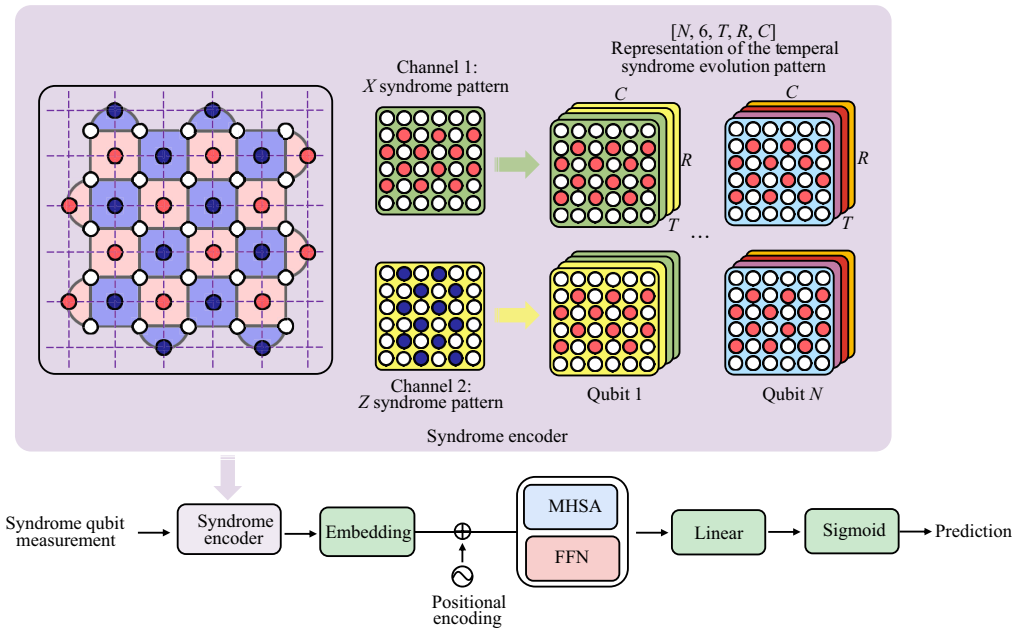


Fig. 5. The architecture of the transformer decoder model. First, the input syndromes are encoded into a cubic grid of dimension $d+1$, which is then fed into the transformer encoder that includes self-attention and the feed-forward network layers. Next, the transformer decoder processes the d-dimensional positional encoding of the data qubits to generate the predictions for physical errors.

In our transformer encoding architecture, each stabilizer generates an embedding that includes syndrome measurement and detection data for each round. Measurement data represent the current round's stabilizer measurement result, while detection data are the XOR between the current and previous round's stabilizer measurements. Qubit features are mapped to a high-dimensional space through an embedding layer, combined with three-dimensional sinusoidal positional encodings to preserve position information. Additionally, a binary flag is introduced to indicate whether it is the final round of decoding. These data are flattened into a long sequence and fed into the transformer model. The transformer encoder consists of a multi-head self-attention layer and a feed-forward network layer. The self-attention mechanism is given by

$$A_t(Q,K,V) = S_m\left(\frac{\phi(QW_Q)(\phi(KW_K))^{\mathrm{T}} + \mu I}{\sqrt{d_k}}\right) VW_V, \quad (14)$$

where W_Q, W_K and W_V are the query, key and value matrices, respectively, obtained by linear transformations of the input embeddings. The multi-head attention mechanism is represented by

$$\begin{aligned} &A_t(Q,K,V) \\ &= C_t\left(\prod_{i=1}^{h}(1+\phi_2(W_2\cdot\phi_1(W_1\cdot \mathrm{head}_i + b_1)+b_2))\right) W_O, \end{aligned} \quad (15)$$

where $\mathrm{head}_i = A_t(Q_i,K_i,V_i)$, W_O is the output transformation matrix and h is the number of attention heads. Using multi-head self-attention, the model can capture relationships between different syndrome features, enhancing the ability to perceive multi-qubit errors. The feed-forward network layer applies two linear transformations with activation functions to capture non-linear features and restore them to the original dimension, thereby improving the model's representation of the surface code state.

The transformer decoder predicts physical errors based on the syndrome measurement results from each round and removes the corresponding syndromes to obtain the global parity of the predicted errors. The process of removing syndromes involves flipping the syndromes associated with the errors again, resulting in the final output of the predicted global parity of the surface code syndromes.

3.4. Training process

To improve the predictive capability of the model, we trained the transformer model on the QEC dataset to accurately identify syndrome patterns under different error rates.[49–51] The model considers both syndrome measurement errors and data qubit errors, with syndrome error probability p and data qubit errors modeled as depolarizing errors, where Pauli X, Y and Z errors occur with equal probability p. The formula is defined as

$$\mathcal{E}(\rho) = (1-p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z), \quad (16)$$

and the actual observed result s' for a single syndrome qubit during measurement follows a binary symmetric channel model, defined as

$$P(s'|s) = (1-p)\delta_{s',s} + p\delta_{s',s\oplus 1}, \quad (17)$$

where $\delta_{a,b}$ is the Kronecker delta, which is equal to one when $a=b$ and zero otherwise. Here, $s\oplus 1$ indicates flipping the binary variable s. For each qubit in the system, we can represent the evolution of the qubit state using the following formula:

$$\rho(t) = \sum_{i=0}^{3} p_i(t) E_i \rho(0) E_i^{\dagger}, \quad (18)$$

where $\rho(0)$ represents the quantum state density matrix at time t and $p_i(t)$ represents the probability of each noise operator E_i at time t, satisfying $\sum_{i=0}^{3} p_i(t) = 1$. Considering the interactions between multiple qubits, we use the multi-qubit noise propagation equation to describe the evolution of the noise

$$\rho_{\text{total}}(t) = \text{Tr}_E \left[U(t) \left(\rho_S(0) \otimes \rho_E(0) \right) U^{\dagger}(t) \right], \tag{19}$$

where $\rho_{\text{total}}(t)$ is the total density matrix and $U(t) = \mathrm{e}^{-\mathrm{i}Ht}$ represents the joint evolution operator between the system and the environment. H is the interaction Hamiltonian between the system and the environment. We trained the transformer decoder for decoding surface codes with code distances $d = 3$, 5, 7, 9, with six encoder and decoder layers, embedding dimension of 64, four attention heads and a feed-forward network hidden dimension of 256. The training dataset contained 10^5 samples, with a base learning rate of 1%. Each epoch processed a batch of the dataset, updating weights through forward and backward propagation. To prevent overfitting, dropout was used, and the Adam optimizer was applied to adaptively adjust the learning rate. We also used a binary cross-entropy loss function to predict the parity of syndrome decoding. The loss function of the model can be expressed using the following formula:

$$\begin{aligned} L(\theta) = & -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right] \\ & + \lambda \left(\frac{1}{2} \sum_{l=1}^{L} \| W^{(l)} \|_2^2 + \alpha \| W^{(l)} \|_1 \right), \end{aligned} \tag{20}$$

where N is the total number of samples and y_i and $\hat{y}_i$ are the true labels and model predictions, respectively.

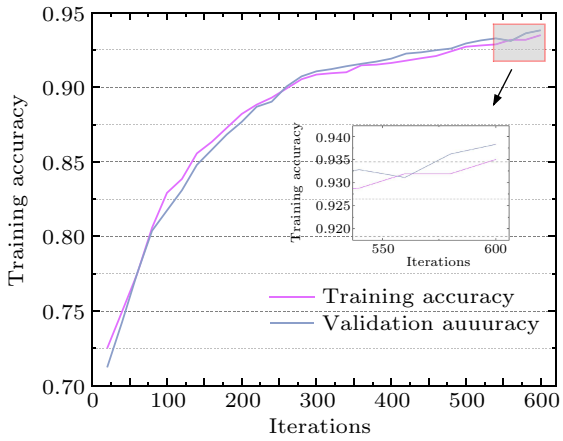


Fig. 6. The number of training iterations for the decoder is a function of training error and accuracy. In the graph, the horizontal axis represents the number of training iterations, while the vertical axis represents training accuracy. The purple curve denotes the accuracy of the training set and the blue curve represents the accuracy of the validation set. To provide a clearer visualization of the data trends, scaling has been applied to the graph.

During the training process, we first trained with small code distance datasets and stopped when the threshold was approached. Then, we increased the code distance and continued training until approaching the threshold again. Finally, when the code distance was further increased, the prediction results deviated from the threshold, indicating that the model had achieved high precision for datasets close to the threshold. The prediction accuracy was almost consistent with that on the training set, as shown in Fig. 6.

4. Main results

4.1. Error correction success rate

Our transformer decoder consists of six functional modules. To enhance the model's performance in deep neural networks, we used the ReLU activation function in the pre-performance network layer. This activation function not only enhances the neural network's ability to handle complex data patterns but also effectively prevents model training failure caused by vanishing gradients during the training process, thereby maintaining the model's accuracy and stability throughout training. We tested our decoder at distances $d = 3$, 5, 7 and 9 and, after training under specific noise conditions. The success rate of error correction can be expressed as a complex function that incorporates the effects of noise, and the formula is defined as follows:

$$P_{\text{success}}(p, d) = \exp\left(-\alpha d \cdot \left(\frac{p}{p_{\text{thr}}} \right)^{\beta d} \cdot (1 + \gamma \cdot f(\eta)) \right), \tag{21}$$

where d represents the code distance. Finally, we obtained the error success rate at each distance, and the results are shown in Fig. 7.

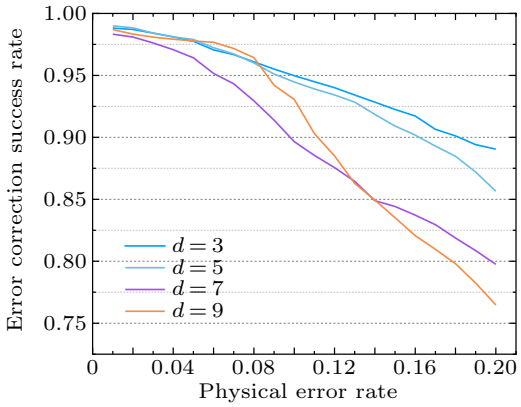


Fig. 7. The simulation results demonstrate the error correction success rate under different physical error rate conditions. We calculated the error correction success rates for code distances $d = 3$ (blue), $d = 5$ (cyan), $d = 7$ (purple), $d = 9$ (orange).

In Fig. 7 it can be seen that under lower physical error rates the error correction success rates at each distance remain relatively high. When the physical error rate is below 0.04, the success rates at distances $d = 3$ and $d = 5$ are relatively high, while that for $d = 7$ is slightly lower. When the error rate increases by 1%, the error correction success rate drops by about 10%. When the error rate is between 0.05 and 0.08, the success rate is highest for code distance $d = 9$ and lowest for $d = 7$. When the error rate is between 0.08 and 0.2, initially

$d = 9$ has a higher success rate than $d = 7$, but as the error rate further increases the success rate for $d = 9$ eventually becomes the lowest. As the physical error rate continues to increase, the error correction success rates for all four code distances show a declining trend. With increasing code distance, the error correction threshold decreases, leading to a decline in the success rate. Therefore, error correction codes with smaller code distances typically exhibit higher success rates.

4.2. Logical error rate

As the code distance increases, the ability of the system to detect and correct errors improves significantly. This is because larger code distances can provide stronger error correction capabilities. However, at higher physical error rates, multiple qubits may experience errors simultaneously, exceeding the system's error correction capacity and leading to uncorrectable errors. This situation is particularly evident in quantum systems because quantum noise is more complex and unpredictable. The logical error rate refers to the probability of an uncorrectable error occurring during the encoding of information. It also describes the system's performance when facing multiple simultaneous errors. The formula is as follows:

$$P_{\mathrm{L}} = \sum_{k=0}^{n} (\frac{n}{j})(1-p)^{n-k} p^{k} (1-\gamma_k), \tag{22}$$

$$\gamma_k = 1 - \frac{1}{p^k} \sum_{j=0}^{k} \binom{k}{j} p^{k-j} \sum_{\beta=0}^{k-j} \binom{k-j}{\beta} p_X^{\beta} p_Y^{k-j-\beta}, \tag{23}$$

where γ_k is the probability that an error of weight k is correctly decoded. When the physical error rate is high, the probability of multiple qubits experiencing errors simultaneously increases, and if these errors cause the original codeword to transform into another valid codeword or become an uncorrectable error, the logical error rate will rise significantly. Figure 8 illustrates the changes in logical error rates for different code distances under varying physical error rates.

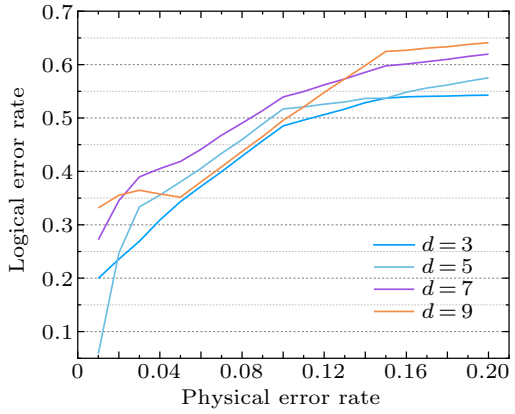


Fig. 8. Simulation results demonstrating the logical error rate under different physical error rate conditions. We calculated the the logical error rates for code distances $d = 3$ (blue), $d = 5$ (cyan), $d = 7$ (purple) and $d = 9$ (orange).

In Fig. 8 we can observe that in the initial stage, when the physical error rate is relatively low, the logical error rate at a code distance of $d = 5$ is the lowest among the four distances. This indicates that the code at this distance has better error correction capability when facing low error rates. When the error rate reaches 0.03, the system at $d = 3$ shows the lowest logical error rate, while the system at $d = 7$ shows the highest, with a difference of approximately 12%. As the physical error rate continues to increase, the logical error rates at all code distances rise significantly. When the physical error rate reaches 0.15, the logical error rate at $d = 9$ is the highest, peaking at 62.4%. At this point, the system's error correction ability is nearly saturated and many logical errors can no longer be corrected. However, as the physical error rate continues to increase further, the rise in the logical error rate gradually slows down and eventually stabilizes. This demonstrates that the system has reached its noise threshold. While it can no longer avoid the occurrence of logical errors, the error rate will level off and the system's performance becomes more predictable.

Through in-depth analysis of the error correction success rate and logical error rate, we can conclude the following: under conditions of a low error rate, the decoder we used performs similarly to commonly seen decoders in terms of decoding accuracy, achieving relatively high decoding precision and exhibiting good error correction performance. However, for larger code distances, as the physical noise increases, the system's error correction ability declines significantly. This is because with the increase in code distance the complexity of the logical space expands, leading to a significant increase in the complexity of the error space, and multiple simultaneous errors become more difficult to correct. Therefore, the error success rate will drop into a predictable range under such noisy conditions, presenting greater challenges for decoding accuracy.

4.3. Decoding threshold

In this section, we present the decoding threshold results of the transformer decoder. To evaluate the decoding performance, it is crucial to analyze the relationship between the error rates of logical qubits and physical qubits. The threshold p represents a critical physical error rate and is an important indicator of the performance of a decoding algorithm. When the error rate at the physical layer is below this threshold, errors can be effectively suppressed using topological encoding, and the probability of errors occurring will decrease exponentially as the code distance increases. This behavior can be expressed by the formula

$$\varepsilon_d = \left(\frac{p}{p_{\mathrm{thr}}} \right)^{(d+1)/2}. \tag{24}$$

Here, ε_d is the logical error rate and p_{thr} is the threshold of the code. When the physical error rate $p/p_{\mathrm{thr}} < 1$, the logical error rate ε_d decreases rapidly with the increase in d, indicating the enhanced ability of the code to correct errors. This is

because larger code distances allow for more complex decoding strategies, thereby effectively reducing the occurrence of logical errors. When the physical error rate $p/p_{\text{thr}} > 1$, the logical error rate ε_d increases with the increase in d. Excessively high physical error rates will lead to the accumulation of logical errors during error correction. When $\varepsilon_d = 1/2^{(d+1)/2}$, the critical threshold is reached.

We tested the decoder under different physical error rates. As shown in Fig. 9, we first performed decoding for code distances $d = 3$, 5 and 7. The threshold determination is based on the definition of the logical error rate, which is the probability of successful decoding. In this case, we obtained a decoding threshold of 0.05, corresponding to a logical error rate of 0.1455. To better demonstrate the decoding performance, we compared the threshold behavior of the transformer decoder with that of the MWPM decoder under the same noise model and code distances $d = 3$, 5 and 7. Figure 10 shows the threshold results of the surface code using the MWPM decoding algorithm. Under these conditions, the threshold of the MWPM decoder is approximately 0.0275, which is noticeably lower than that achieved by the transformer decoder. Compared with the MWPM threshold of 0.0275, our transformer decoder's threshold is improved by about 45%.

To further improve the decoder's threshold, we extended the code distance to $d = 9$, as shown in Fig. 11, and obtained a decoding threshold of 0.0605, corresponding to a logical error rate of 0.1005. Meanwhile, we also increased the code distance of the MWPM decoder to 9, as shown in Fig. 12. Under these conditions, the decoding threshold is 0.0372, with a corresponding logical error rate of 0.2125. This threshold remains noticeably lower than that achieved by our transformer decoder. Compared with the MWPM threshold of 0.0372, the threshold of our transformer decoder is improved by approximately 38.5%. When comparing time complexities, the use of a transformer-based self-attention mechanism could be considered for surface code decoding. For a surface code with distance d, the measurement data size typically scales as $O(d^2)$. Since the complexity of self-attention is $O(N^2)$ with respect to the sequence length N, setting $N = O(d^2)$ results in a single layer of self-attention having a computational complexity of approximately $O(d^4)$. In contrast, for decoding the same code distance using the MWPM algorithm, the number of defects also scales as $O(d^2)$. The classical MWPM complexity is $O(N^3)$ for a graph with N vertices, so we get an MWPM complexity of $O(d^6)$.

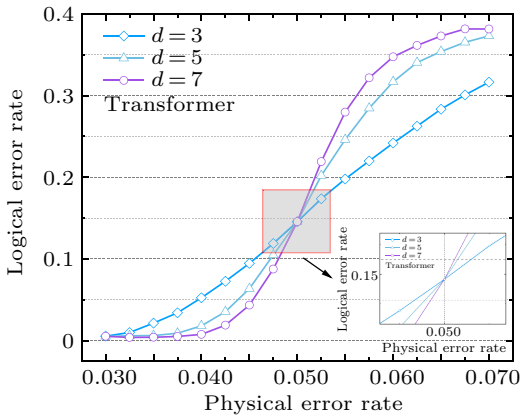


Fig. 9. The performance of the transformer-based decoder at code distances $d = 3$, $d = 5$ and $d = 7$. The intersection of the curve and the dashed line represents the decoder's threshold, which is approximately 0.05 in this case.

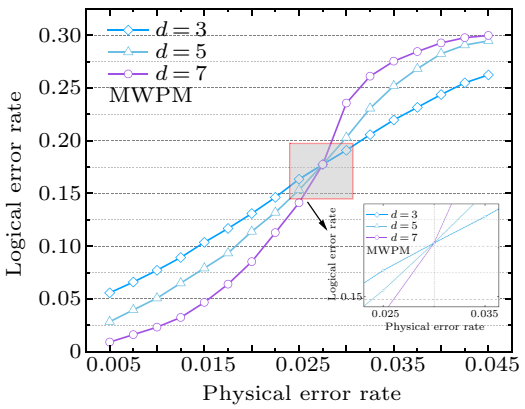


Fig. 10. The performance of the MWPM decoder at code distances $d = 3$, $d = 5$ and $d = 7$. The intersection of the curve and the dashed line represents the decoder's threshold, which is approximately 0.0275 in this case.

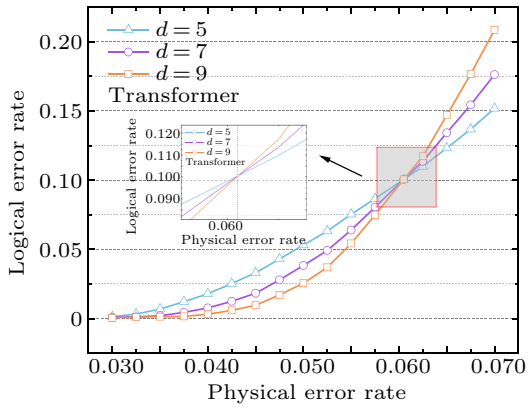


Fig. 11. The performance of the transformer-based decoder at code distances $d = 5$, $d = 7$ and $d = 9$. The intersection of the curve and the dashed line represents the threshold of the decoder, which is approximately 0.0605 in this case.

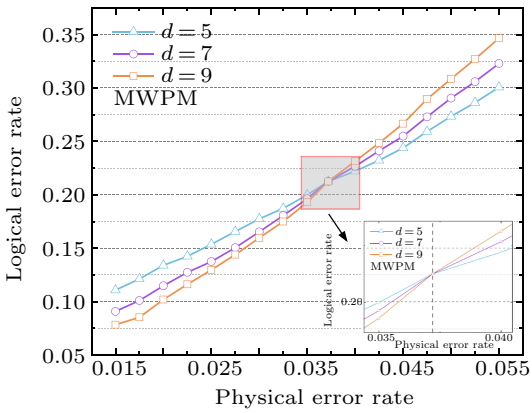


Fig. 12. The performance of the MWPM decoder at code distances $d = 5$, $d = 7$ and $d = 9$. The intersection of the curve and the dashed line represents the threshold of the decoder, which is approximately 0.0372 in this case.

The results show that when the physical error rate is low, the probability of successful decoding increases with increase in the code distance. However, when the error rate is high, the opposite is true. The results based on the decoding threshold indicate that when the physical error rate is below this threshold, larger code distances can correct more errors, thereby reducing the logical error rate. But when the physical error rate is above this threshold, larger code distances may introduce more errors, leading to an increase in the logical error rate. This trend is likely due to the increased encoding complexity of larger code distances, which introduces additional errors, making it impossible for the system to fully correct all errors and thus limiting the overall decoding performance.

5. Conclusion

In this work we studied a transformer decoder incorporating self-attention mechanisms for QEC of surface codes. The model not only addresses data qubit errors but also considers the impact of syndrome qubits, effectively integrating syndrome information through self-attention to enhance the decoding performance. With neural network optimization, the decoder achieved a threshold of 0.0605 and an accuracy of 93.8% under the phenomenological error model. Future research directions include combining the transformer decoder with other topological QEC codes, such as toric codes and color codes. Although the transformer decoder reached threshold decoding, there is still a gap compared with the theoretical optimal threshold. This study demonstrates the potential of ML-based transformer decoders in improving the accuracy and efficiency of quantum computing.

Acknowledgements

Project supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2021MF049), Joint Fund of Natural Science Foundation of Shandong Province (Grant Nos. ZR2022LLZ012 and ZR2021LLZ001), and the Key R&D Program of Shandong Province, China (Grant No. 2023CXGC010901).

References

[1] Li J, Peng X, Du J and Suter D 2012 *Sci. Rep.* **2** 260
[2] Gong L H, Li M L, Cao H and Wang B 2024 *Laser Phys. Lett.* **21** 055209
[3] Daley A J, Bloch I, Kokail C, Flannigan S, Pearson N, Troyer M and Zoller P 2022 *Nature* **607** 667
[4] Qiao L F, Streltsov A, Gao J, Rana S, Ren R J, Jiao Z Q, Hu C Q, Xu X Y, Wang C Y, Tang H, Yang A L, Ma Z H, Lewenstein M and Jin X M 2018 *Phys. Rev. A* **98** 052351
[5] Pan J W, Chen Z B, Lu C Y, Weinfurter H, Zeilinger A and Żukowski M 2012 *Rev. Mod. Phys.* **84** 777
[6] Terhal B M 2015 *Rev. Mod. Phys.* **87** 307
[7] Chiaverini J, Leibfried D, Schaetz T, Barrett M D, Blakestad R B, Britton J, Itano W M, Jost J D, Knill E, Langer C, Ozeri R and Wineland D J 2004 *Nature* **432** 602
[8] Calderbank A R and Shor P W 1996 *Phys. Rev. A* **54** 1098
[9] Andersen C K, Remm A, Lazar S, Krinner S, Lacroix N, Norris G J, Gabureac M, Eichler C and Wallraff A 2020 *Nat. Phys.* **16** 875
[10] Stephens A M 2014 *Phys. Rev. A* **89** 022321
[11] Zhao Y, Ye Y, Huang H L, *et al.* 2022 *Phys. Rev. Lett.* **129** 030501
[12] Bravyi S, Englbrecht M, König R and Peard N 2018 *npj Quantum Inf.* **4** 55
[13] Liu Y H and Poulin D 2019 *Phys. Rev. Lett.* **122** 200501
[14] Yan D D, Fan X K, Chen Z Y and Ma H Y 2022 *Chin. Phys. B* **31** 010304
[15] Meinerz K, Park C Y and Trebst S 2022 *Phys. Rev. Lett.* **128** 080505
[16] Sundaresan N, Yoder T J, Kim Y, Li M Y, Chen E H, Harper G, Thorbeck T, Cross A W, Córcoles A D and Takita M 2023 *Nat. Commun.* **14** 2852
[17] Sheth M, Jafarzadeh S Z and Gheorghiu V 2020 *Phys. Rev. A* **101** 032338
[18] Bravyi S, Suchara M and Vargo A 2014 *Phys. Rev. A* **92** 032326
[19] Hammar K, Orekhov A, Hybelius P W, Wisakanto A K, Srivastava B, Kockum A F and Granath M 2022 *Phys. Rev. A* **105** 042616
[20] Fan X K, Yan D D, Liu F and Ma H Y 2022 *Sci. Sin. Inform.* **52** 539
[21] Song Z Y, Wang Y N, Wang H W and Ma H Y 2023 *Acta Electronica Sinica* **51** 2030
[22] Gicev S, Hollenberg L C L and Usman M 2023 *Quantum* **7** 1058
[23] Baireuther P, Caio M D, Criger B, Beenakker C W J and O'Brien T E 2019 *New J. Phys.* **21** 013003
[24] Varsamopoulos S, Bertels K and Almudever C G 2020 *Quantum Mach. Intell.* **2** 1
[25] Bordoni S and Giagu S 2023 *Quantum Inf. Process* **22** 151
[26] Krastanov S and Jiang L 2017 *Sci. Rep.* **7** 11003
[27] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 arXiv:1706.03762v7 [cs.CL]
[28] Cao H Y, Pan F, Wang Y J and Zhang P 2023 arXiv:2307.09025v1 [quant-ph]
[29] Wang H R, Liu P Y, Shao K, Li D T, Gu J Q, Pan D Z, Ding Y S and Han S 2023 arXiv:2311.16082v1 [quant-ph]
[30] Bausch J, Senior A W, Heras F J H, Edlich T, Davies A, Newman M, Jones C, Satzinger K, Niu M Y, Blackwell S, Holland G, Kafri D, Atalaya J, Gidney C, Hassabis D, Boixo S, Neven H and Kohli P 2024 *Nature* **635** 834
[31] Fujisaki J, Oshima H, Sato S and Fujii K 2022 *Phys. Rev. Res.h* **4** 043086
[32] Smith S C, Brown B J and Bartlett S D 2023 *Phys. Rev. Appl.* **19** 034050
[33] Reichardt B W 2021 *Quantum Sci. Technol.* **6** 015007
[34] Bhoumik D, Sen P, Majumdar R, Sur-Kolay S, Kumar K J L and Iyengar S S 2021 arXiv:2110.10896v1 [quant-ph]
[35] Bonilla Ataides J P, Tuckett D K, Bartlett S D, Flammia S T and Brown B J 2021 *Nat. Commun.* **12** 2172
[36] Versluis R, Poletto S, Khammassi N, Tarasinski B, Haider N, Michalak D J, Bruno A, Bertels K and DiCarlo L 2017 *Phys. Rev. Appl.* **8** 034021
[37] Horsman D, Fowler A G, Devitt S and Meter R V 2012 *New J. Phys.* **14** 123011
[38] Fowler A G, Mariantoni M, Martinis J M and Cleland A N 2012 *Phys. Rev. A* **86** 032324
[39] Varbanov B M, Serra-Peralta M, Byfield D and Terhal B M 2023 arXiv:2307.03280v2 [quant-ph]
[40] Kubica A and Delfosse N 2023 *Quantum* **7** 929
[41] Vedral V and Plenio M B 1998 *Progress in Quantum Electronics* **22** 1
[42] Nagy M and Akl S G 2006 *International Journal of Parallel, Emergent and Distributed Systems* **21** 1
[43] Hu T J, Wu J D and Li Q 2024 *IEEE Network* **38** 155
[44] Ercan H E, Ghosh J, Crow D, Premakumar V N, Joynt R, Friesen M and Coppersmith S N 2018 *Phys. Rev. A* **97** 012318
[45] Xu M X, Dai W R, Liu C M, Gao X, Lin W Y, Qi G J and Xiong H K 2020 arXiv:2001.02908v2 [eess.SP]
[46] Yun S, Jeong M, Yoo S, Lee S, Yi S S, Kim R, Kang J and Kim H J 2022 *Neural Networks* **153** 104
[47] Fowler A G, Stephens A M and Groszkowski P 2009 *Phys. Rev. A* **80** 052312
[48] Stephens A M 2014 *Phys. Rev. A* **89** 022321
[49] Dür W and Briegel H J 2007 *Rep. Prog. Phys.* **70** 1381
[50] Huo M X and Li Y 2017 *New J. Phys.* **19** 123032
[51] Bravyi S and Haah J 2013 *Phys. Rev. Lett.* **111** 200501