TT-QEC: Transferable Transformer for Quantum Error Correction Code Decoding

Hanrui Wang[1], Pengyu Liu[2], Kevin Shao[1], Dantong Li[3], Jiaqi Gu[4], David Z. Pan[5], Yongshan Ding[3], Song Han[1]
[1]MIT [2]CMU [3]Yale University [4]Arizona State University [3]University of Texas at Austin

***Abstract*—Quantum Computing holds the promise of resolving classically unsolvable problems with superior speed and efficiency. Nonetheless, the prevalent error rate in current quantum devices surpasses the tolerable threshold for executing meaningful quantum algorithms by a significant magnitude. Quantum error correction (QEC) is the technique to enhance the error resilience of quantum systems by incorporating redundancy, whereby quantum information is distributed across multiple data qubits. Additionally, syndrome qubits are implemented to monitor data qubit parity, thereby detecting potential errors. The syndrome information is processed by a decoder to predict data qubit errors. However, accurate decoding is challenging because errors occur not only on data qubits but on syndrome qubits and syndrome extraction operations, which cause complex syndrome patterns. Furthermore, the same syndrome pattern could result from different underlying errors. Therefore, it is important for the decoder to consider all syndromes collectively. Given that a single code family can possess varying code distances, a generalizable decoding approach capable of handling different code distances is highly desirable. Machine learning (ML) decoders are considered promising candidates with multi-layer perceptron (MLP) or convolution neural network (CNN) based ones being proposed recently. However, most existing ML decoders only process a *local* region of syndromes without global receptive field, and necessitate costly *retraining* for different code distances.**

To overcome these challenges, we introduce TT-QEC, a transformer based QEC decoder that performs self-attention across all input syndromes, thus acquiring global receptive field. It also employs a *mixed loss* training mechanism that combines the loss from local physical errors and the loss from the global parity labels. Furthermore, leveraging the capability of transformer to handle arbitrary length of inputs and outputs, we propose an efficient *transfer learning* that can produce a decoder for different code distance based on a model of existing distance.

Evaluation on six code distances and ten different error configurations demonstrates that our model consistently outperforms non-ML decoders, such as Union Find (UF) and Minimum Weight Perfect Matching (MWPM), and other ML decoders, thereby achieving best logical error rates. Moreover, the transfer learning can save over 10× of training cost.

I. Introduction

Quantum Computing (QC) has been garnering substantial research interest as an emergent computational model designed to address problems previously deemed unsolvable with enhanced efficiency. A multitude of sectors and academic disciplines stand to gain from the potentialities of QC, notably cryptography [1], database search [2], combinatorial optimization [3], molecular dynamics [4], and machine learning [5]–[14] applications, etc.

Advancements in physical implementation technologies have spurred the rapid progression of QC hardware over the past two decades. These advancements have facilitated the release of QC systems boasting up to 433 qubits [15]–[19], representing the cutting-edge of current QC capabilities. The continuing evolution of QC holds promise for the development of even more efficient algorithms, fostering its broader adoption across numerous domains of application.

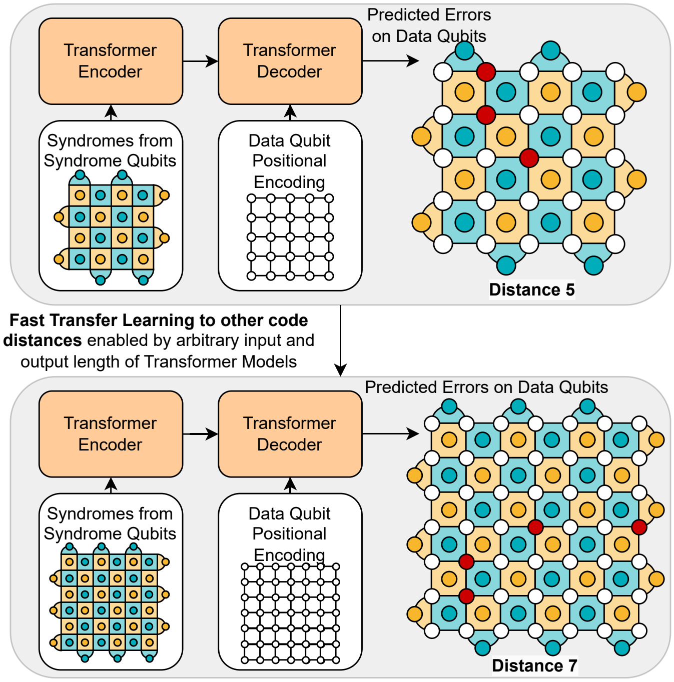
Transformer Encoder
Transformer Decoder
Predicted Errors on Data Qubits
Syndromes from Syndrome Qubits
Data Qubit Positional Encoding
Distance 5
Fast Transfer Learning to other code distances enabled by arbitrary input and output length of Transformer Models
Transformer Encoder
Transformer Decoder
Predicted Errors on Data Qubits
Syndromes from Syndrome Qubits
Data Qubit Positional Encoding
Distance 7

Fig. 1. Transformer for Error correction decoding overview. The Transformer takes syndrome inputs and processes them through both the transformer model encoder and decoder. The output of this process consists of error predictions. One notable advantage is its ability to be seamlessly applied to different code distances due to the transformer model's flexible input and output size.

Despite the exciting advancements, the qubits and quantum gates on current quantum machines suffer from high error rates of 10^{-3} to 10^{-2}, preventing us from executing applications that demand significantly lower error rates (below 10^{-10}) [20]–[22]. Therefore, reducing quantum error is of pressing demand to close the gap. Quantum Error Correction (QEC), an essential solution to this challenge, lowers the error rate by integrating redundancy, a process where the information from a single logical qubit is distributed across multiple physical qubits. By increasing the redundancy of the QEC code, the logical error rate plummets exponentially, assuming that the physical error rate p stays below a certain limit. Therefore, by skillfully controlling the redundancy, QEC enables us to achieve the required error rate for executing specific applications.

Quantum error correction integrates both quantum and classical elements in its design. On the quantum aspects, we observe a logical qubit's role in encoding quantum information within the collective state of several data qubits. Error detection involves the strategic positioning of parity qubits among these data qubits with one example in Figure 1 top right. Through periodic execution of a syndrome extraction circuit, each parity qubit retrieves the parity data of certain data qubit subsets. Subsequently, it transfers any data qubit errors into discernible, individual discrepancies. This iterative process will be repeated for many times (cycles), spanning from qubit initialization to the final measurement of data qubits. A syndrome represents the accumulated measurement results of all parity qubits within a cycle. For the classical elements of QEC, we come across the decoder's role. By analyzing syndromes, the decoder detects potential errors and determines the most suitable corrections for the data qubits. The logical error rate relies on factors such as the physical error rate and decoder performance. Since the same syndrome could be caused by different set of errors, the decoder typically need to process a large region of syndromes instead of a small receptive field.

The rotated surface code [23] is a promising candidate for realizing fault tolerance. Well-established algorithms for surface code include Minimum Weight Perfect Matching (MWPM) and Union Find (UF). Recently, machine learning (ML), especially neural network (NN) based decoders have gained attention due to a few desirable characteristics. First, they generally run in constant time, which is necessary to prevent a backlog of syndrome outcomes. Second, unlike MWPM, they are capable of learning both correlations between physical errors (such as the correlation between X and Z error in depolarizing errors) as well as learning hidden and potentially changing underlying physical error distributions.

However, ML based decoders also bring significant challenges. First, several models, such as those grounded in convolutional neural networks (CNN), are limited by a small receptive field. This constraint may hinder their ability to accurately pinpoint long error chains. Second, many models, like the multi-layer perceptron (MLP), has a fixed size for both input and output. As a result, changes in code distance would mandate retraining of an entirely new model, leading to considerable overhead.

Therefore, to solve these challenges, we propose TT-QEC, a transferable transformer model designed for accurate and efficient decoding of surface code, as illustrated in Figure 1. For the sake of simplicity, the figure only depicts two dimensions, but in reality, the input syndromes include an additional temporal dimension – *round*. Our proposed model employs a transformer structure, incorporating both an encoder and a decoder to process the syndromes. Binary features on each syndrome qubit are projected to token embeddings and augmented with a 3D sinusoidal positional encoding, informing the model about the location of each qubit. The embeddings of the 3D inputs are then flattened to 1D input sequence and processed by the transformer encoder layer. Thanks to the global interaction capability brought by attention layer, all input syndromes can be considered holistically which boosts accuracy. The decoder then uses the positional encoding of the data qubits to predict the X or Z errors on each of them. Moreover, we propose a *mixed loss* that combines the loss from the local physical error of each qubit with the loss from global parity prediction.

In order to reduce the cost of model training associated with different code distances and leverage the knowledge from trained models, we further propose implementing transfer learning across code distances. Specifically, given that the transformer model input can be of arbitrary length, we could directly reuse the weights of a trained model to a new code distance by simply altering the input sequence. As such, after training a model (e.g. for distance 5), we can directly apply it to a different distance (e.g 7 or 9) and perform a quick fine-tuning to improve the performance at the target distance. This approach reduces the cost by a factor of 10 compared to training from scratch in our settings.

We extensively evaluate TT-QEC across six code distances, 3, 5, 7, 9 and compare it with MWPM, UF and MLP baselines under 10 different error rates. Our results demonstrate that TT-QEC consistently surpasses these baselines, achieving the lowest logical error rates. In summary, TT-QEC makes four key contributions:

- **A novel transformer-based model** for surface code decoding which uses the syndrome with positional encoding as inputs and predict errors.
- **A mixed loss** approach combined loss from local physical error prediction and global parity prediction improves the model's trainability and performance.
- **Transfer learning across different code distances**. For the first time, we propose to transfer the knowledge learn on one distance to another, thus reducing costs.
- **Extensive evaluations** on different physical error rates and distances demonstrates that our model consistently outperforms baselines such as MWPM, UF and MLP.

II. Background

A. Quantum Basics

Qubits and Quantum Circuit. The potency of quantum computation is derived from its fundamentally distinct approach to storing and manipulating information [24], [25]. A quantum bit, known as a *qubit*, deviates from a conventional bit in its capability to exist in a linear combination of the two basis states 0 and 1: $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, with $\alpha, \beta \in \mathbb{C}$, fulfilling the condition $|\alpha|^2 + |\beta|^2 = 1$. This distinctive feature of generating a "superposition" of basis states facilitates the representation of a linear combination of 2^n basis states using an n-qubit system. This stands in contrast to a classical n-bit register that can only store one of the 2^n states. Quantum computation on a quantum system involves the manipulation of the state

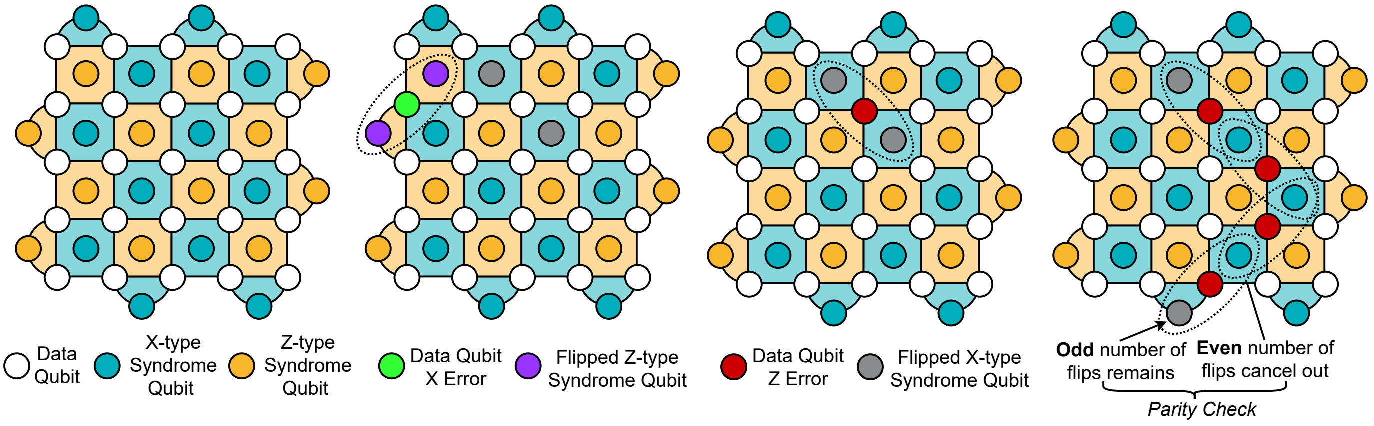
Data Qubit
X-type Syndrome Qubit
Z-type Syndrome Qubit
Data Qubit X Error
Flipped Z-type Syndrome Qubit
Data Qubit Z Error
Flipped X-type Syndrome Qubit
Odd number of flips remains
Even number of flips cancel out
Parity Check

Fig. 2. Surface Code. The surface code contains data qubits and two kinds of syndrome qubits. X-type syndrome qubits in green checks Z errors while Z-type syndrome qubits in yellow check X errors. When error occurs on data qubits, the nearby syndromes may be flipped depending on the parity of data qubits. When multiple error occurs, the syndrome patterns will be more difficult to decode.

of qubits by employing a *quantum circuit*. A quantum circuit comprises a series of operations termed *quantum gates*, which facilitate the transition of one quantum state to another.

Operational Noises. In real QC, a myriad of errors can transpire due to factors such as imperfect control signals, undesired interactions between qubits, or external environmental interference [26]–[28]. Consequently, qubits undergo *decoherence error* over time, while quantum gates impart *operation errors*, such as coherent or stochastic errors, into the system. To mitigate the impact of noise, these systems require frequent characterization [28] and calibration [29].

B. Quantum Error Correction

Quantum Error Correction (QEC) is a technique that enhances the reliability of quantum information by encoding logical qubits into a larger number of physical data qubits. The error correction process involves two steps. First, a syndrome extraction circuit runs on the qubits to produce an error signature, known as a syndrome. Second, a decoder identifies and corrects any errors in the data qubits based on this syndrome. Additional qubits, known as syndrome qubits, are used to detect errors without disturbing the quantum state of the data qubits. If the error rate of the physical qubits is below a certain threshold, QEC can effectively lower the logical qubit error rate at the cost of using more physical qubits. The errors are categorized into a discrete set of Pauli errors - bit-flip (Pauli-X), phase-flip (Pauli-Z), or both. In order to correct an arbitrary error, it is sufficient to be able to correct Pauli-X (bit-flip) and Pauli-Z (phase-flip) errors [24]. Since there are two types of errors to account for, the techniques for quantum error correction, despite being similar to their classical counterparts, are often more complex. Moreover, considerations regarding physical realizations render only certain coding schemes viable for Noisy Intermediate Scale Quantum (NISQ) devices. In this paper, we focus on the rotated surface code.

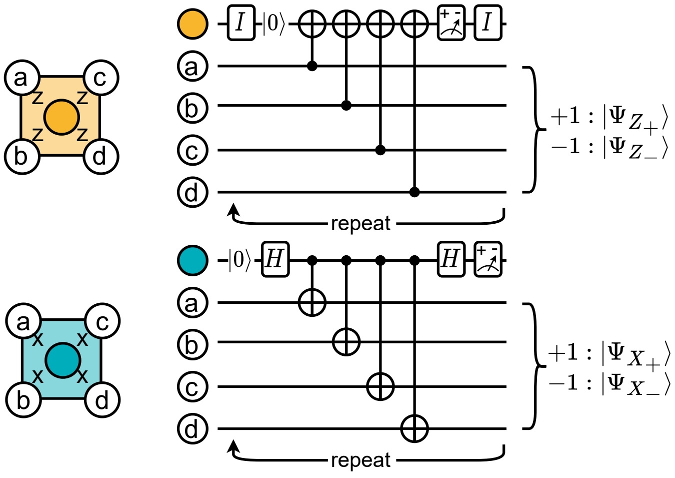
a
b
c
d
Z
X
I
|0⟩
H
$+1 : |\Psi_{Z_+}\rangle$
$-1 : |\Psi_{Z_-}\rangle$
$+1 : |\Psi_{X_+}\rangle$
$-1 : |\Psi_{X_-}\rangle$
repeat

Fig. 3. Syndrome extraction circuit. Top: Z-type syndrome qubits. Bottom: X-type syndrome qubits.

C. Surface Code

The Surface code is a prominent QEC scheme that encodes a logical qubit into a two-dimensional lattice of alternating data and syndrome (parity) qubits. It is characterized by its high error threshold and requirement for only nearest-neighbor connectivity, making it a practical option for real-world quantum systems. The logical qubit is encoded in a lattice of size dependent on the 'code distance' denoted by D, with larger distances offering increased error tolerance. Errors on data qubits are detected by adjacent parity qubits using a stabilizer circuit, as illustrated in Fig. 3 which measures a four-qubit operator, leading to detection of X, Z, or Y (combination of X and Z) errors. The surface code can correct error chains up to length $\lfloor \frac{D-1}{2} \rfloor$. In practice, a simpler variant of the original Toric code [30], the 'rotated' surface code as in Fig. 2 is often preferred due to its more compact layout, reducing the physical qubit and gate overheads. The $[[D^2, 1, D]]$ stabilizer code has become a prime candidate for

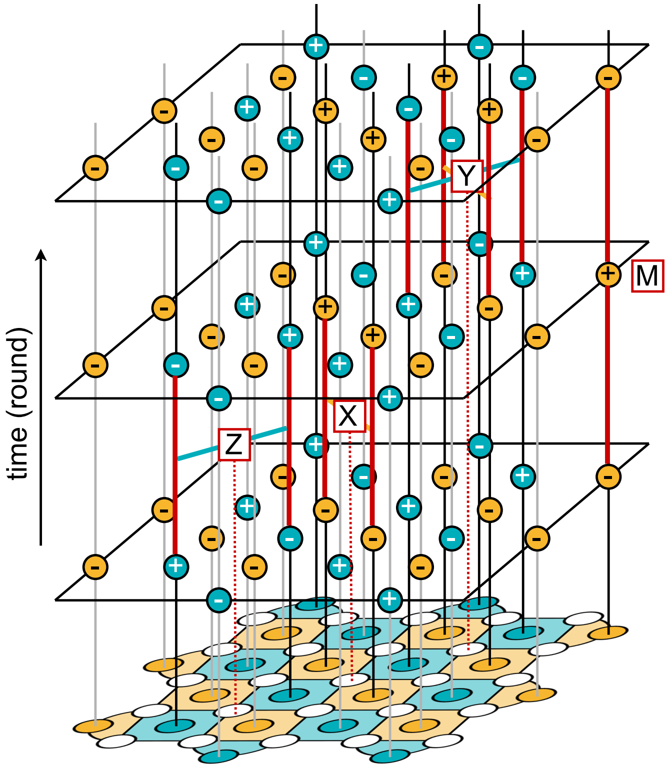

Fig. 4. Multiple rounds of surface code measurement. The progression of time is depicted by moving upwards from the array at the base, with each horizontal plane representing a step in the measurement process. In reality, errors will also occur in the syndrome extraction circuit and syndrome qubits, necessitating the need to repeat multiple rounds for decoding. On the right side, the measurement error on the syndrome qubit will also flip the syndrome.

near-term fault-tolerant quantum computation. Being amenable to single qubit operations transversally, it has also been shown that the two qubit CNOT gate can be applied only by merging and splitting codes in a technique called lattice surgery [31], rendering physical realizations much more feasible.

D. Decoder

Decoders function by interpreting the syndrome, which is the outcome of ancilla measurements, as illustrated in Fig. 4 to establish necessary corrections for data qubits. Errors of the X-type and Z-type are adjusted separately, which automatically amends Y-type errors. For a decoder to be effective in large-scale fault-tolerant quantum computers (FTQCs), it must meet three key requirements: accuracy, latency, and scalability [32]. Accuracy refers to the decoder's ability to reliably identify errors. Latency stipulates that the decoder must operate within one cycle of syndrome extraction. Scalability demands efficient implementation of decoders with minimal hardware resources to function in hardware-limited settings. Typically, more accurate decoders take longer to operate.

III. Related Works

A. ML based decoder

Quantum computing's landscape has been vastly improved by Machine Learning (ML) based decoders, focusing on Neural Network (NN) models, Reinforcement Learning (RL) methods, and innovative designs for decoder scalability. NN models have seen significant advances, starting with [33]'s first use of a Boltzmann machine for ML-based decoding in toric codes. [34], [35] expanded this by applying a Multi-Layer Perceptron (MLP) decoder with comparable performance to previous algorithms. [36] introduced the use of Long Short-Term Memory (LSTM) for decoding surface code measurements. RL techniques form another key branch. [37] translated the decoding problem into an RL environment for a circuit-level noise model. [38] and [39] made strides with RL decoders for specific error types, displaying superior performance through error correlation learning. [40] enhanced performance further with a unique reward mechanism. Regarding decoder scalability, [41] proposed a scalable ML-based decoder with a low-depth Convolutional Neural Network (CNN), expanded upon by [42] and [43]. [44]–[47] ventured into multilevel decoder architectures to improve performance and training, while ensuring execution time independence from code distance. Other noteworthy contributions include [48]'s ML-based decoder for additional logical corrections, [49]'s concept of a distributed neural network, and [50]'s fusion of ML and non-ML-decoder benefits. Together, these significant studies have directed our focus towards Transformer-based Neural Networks for local decoding, given their inherent capability for spacetime volume processing of syndromes. Our research aspires to introduce new NN decoder architectures and enhance scalability through a local-global two-level design.

B. Non-ML based decoder

Numerous decoding methods for Quantum Error Correction (QEC) have been proposed, each presenting unique strengths. A key approach in QEC is Minimum Weight Perfect Matching (MWPM), typically applied for topological error correction. This algorithm leverages Edmonds' method to identify and correct errors via the shortest pairing of error syndromes [51]. MWPM algorithms are used to find the most probable error configuration, with the blossom algorithm [52] assisting in determining the optimal pairing with minimum weight. Upon establishing the matching, suitable correction operations are applied, restoring the original quantum state and effectively reducing error impact.

The Union Find (UF) decoder is another crucial QEC algorithm, characterized by its linear-time complexity which allows for efficient error identification and correction in toric and surface codes [53]. Lookup Table (LUT) decoders, alternatively, operate by learning a set of error patterns from classical error correction codes, enabling real-time correction using a predetermined set of error patterns [54]. The Tensor Network (TN) decoder, a newer approach, employs the tensor network structure to detect and correct errors in topological codes effectively. This graph-based method provides a relatively high error threshold [55].

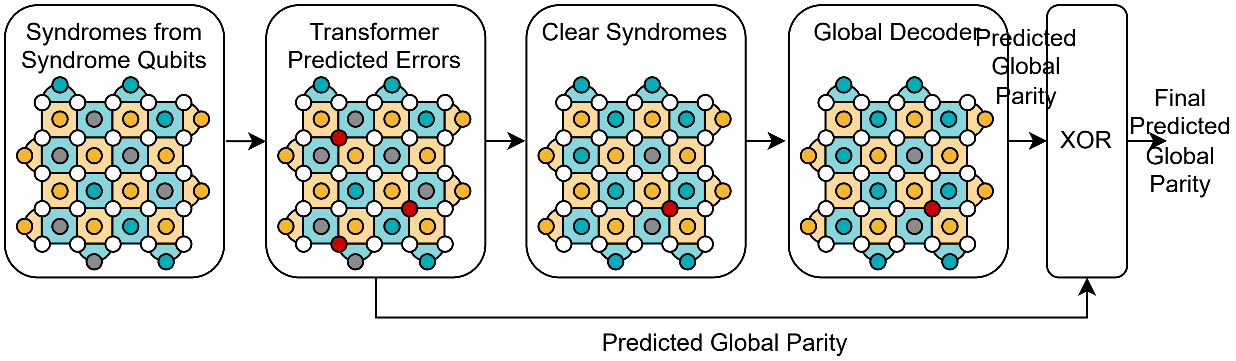
Syndromes from Syndrome Qubits
Transformer Predicted Errors
Clear Syndromes
Global Decoder
Predicted Global Parity
XOR
Final Predicted Global Parity
Predicted Global Parity

Fig. 5. Overall workflow of TT-QEC. The syndromes are firstly processed by the transformer model to predict the errors. Since the errors may not fully clear all syndromes, we will pass the cleared syndromes to a global decoder to predict a global parity. The final global parity is the XOR of the global parity from transformer predicted physical error and that predicted by global decoder.

C. Transformer Models

Transformer models have redefined the field of natural language processing (NLP), demonstrating outstanding performance in tasks such as language generation and text classification. Transformers utilize a multi-head self-attention mechanism to discern relationships among tokens, differing from traditional recurrent or convolutional neural networks. This mechanism enables the model to attend to relevant contextual information from various positions in the input sequence, capturing long-range dependencies effectively due to low inductive bias.

The Vision Transformer (ViT), a Transformer variant tailored for visual tasks, carries the success of Transformers in NLP over to computer vision, surpassing state-of-the-art CNNs, particularly in dense vision tasks requiring global context learnability. ViT processes input 2-D/3-D images into a grid of equal patches, projecting each into a sequence of feature vectors. The Transformer integrates position embeddings to encode spatial coordinates, which, together with the patch embeddings, are fed into the Transformer encoder, allowing the model to comprehend the spatial context of the input sequence/image.

IV. Methodology

In this section, we will first outline the error correction workflow in our TT-QEC framework, before delving into the details of the transformer model and transfer learning framework.

A. Overall Workflow

As mentioned earlier, the iterative syndrome extraction process produces syndrome measurement outcomes at each round. The decoder predicts error occurrences based on these outcomes. Since purely ML-based decoders directly predict errors on data qubits, the predicted errors may not always align exactly with the syndromes. Therefore, an ML decoder is typically paired with a non-ML decoder to clear all the syndromes, as illustrated in Figure 5. Specifically, once the ML decoder has predicted all errors, these predictions are used to clear the syndromes and obtain the global parity of the predicted errors. The syndrome clearing process involves flipping the syndromes linked to errors once again. Ideally, given a highly accurate ML decoder, all syndromes could be cleared. However, any remaining errors are passed to a global decoder, like MWPM, which guarantees clearance of all syndromes and subsequently produces another global parity. The final output is the XOR of two predicted global parities.

B. Transformer Model

Considering that the speed of global decoders, such as MWPM, is typically proportional to the number of non-zero syndromes, it is beneficial to have an ML decoder that clears as many syndromes as possible. Hence, we propose a novel Transformer-based decoder, as depicted in Figure 6.

We use a cubic grid to encode input syndromes. For a surface code of distance D, we utilize a $D+1$ square to ensure that each syndrome qubit is at an intersection. Conventional settings determine the number of rounds to be equivalent to the code distance. We introduce an additional layer for the final measurement, thereby making the round dimension $D+1$ as well. Hence, the features form a $D+1$ cubic grid. Each grid cell comprises a feature vector of length six. The first two channels denote the locations of the X check and Z check syndrome qubits, respectively, as shown in Figure 6 and below.

Location encoding for the X check syndrome qubits:

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Location encoding for the Z check syndrome qubits

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

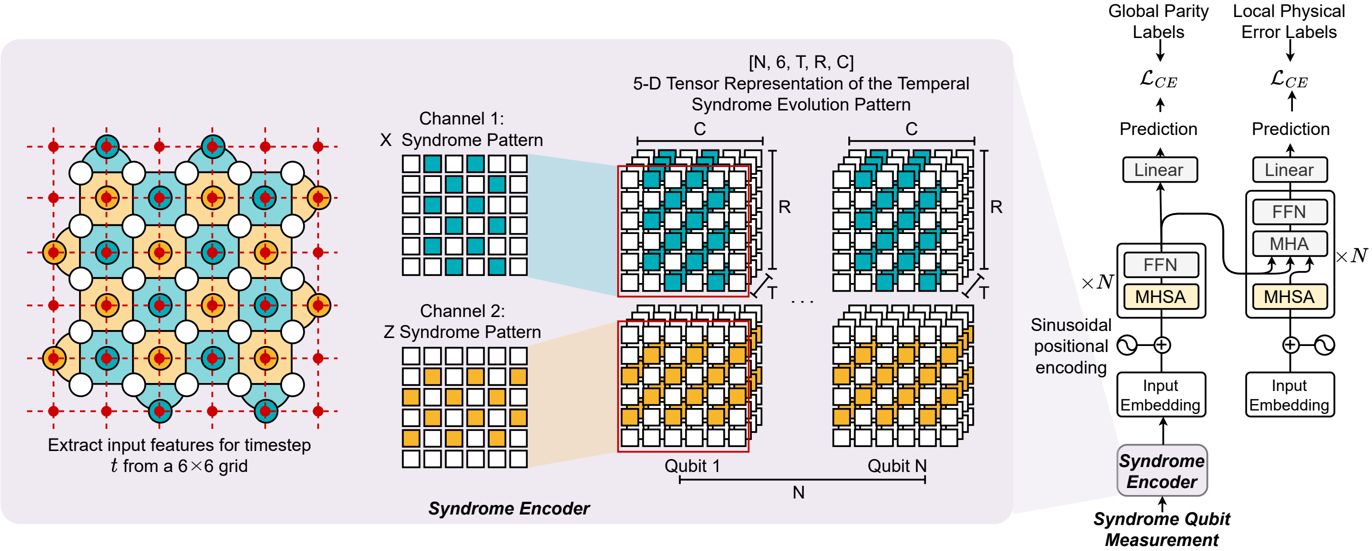

Fig. 6. Transformer model architecture. The input of the syndromes will be encoded by a $(D+1)$ cubic grid. The input will go through the transformer encoder with self attention and FFN layers. Then the transformer decoder will produce the physical error predictions by processing the positional encoding of data qubits with size D cubic.

Following the positional information, the next two channels of the feature vector are the syndromes, which vary across different rounds. In the final stage of error correction, we achieve perfect error correction by measuring the data qubits in a specific basis. Defining the temporal boundaries of the lattice is crucial for the network to generalize to different syndrome measurement rounds. The fifth channel of the dataset is set to 1 for the first round and 0 for subsequent rounds. Similarly, the sixth channel is set to 1 for the last round and 0 for all the other rounds.

The features are then projected to a higher-dimensional space by a learnable embedding layer. To inform the model about the location of each qubit, we add 3-dimensional sinusoidal positional encoding to the embedded inputs. With the position information added, we can safely flatten the 3D embeddings sequence to 1D and send it to the Transformer encoder. The encoder consists of multiple layers, each containing one multi-head self-attention (MHSA) and one Feed-forward network (FFN) layer. The MHSA allows each syndrome feature to attend to any syndrome in the entire 3D grid, thereby enabling better awareness of long-range error chains. The FFN layer contains two fully-connected layers that project the embedding to an even higher dimension, apply an activation function, and project back.

To predict physical errors, we use the Transformer decoder layers. The inputs are directly the positional encoding of the data qubit positions. We then predict the errors on a 3D grid of data qubits with the decoder layers. Inside each decoder layer, there is a layer of self-attention and one layer of cross-attention between the encoder and decoder. The queries for cross-attention come from the decoder layer inputs while the keys and values come from the encoder. This mechanism allows the layer to have access to all the previous syndrome information. Following this, we have an FFN layer and a prediction layer that outputs the logits. Given that false positives are more detrimental than false negatives, we use a confidence threshold to predict an error. Typically, the confidence after Sigmoid needs to be larger than 0.95 for a positive prediction.

C. Mixed Loss

During the training procedure, we propose a novel loss function that combines losses from two sources. One source is the loss from predicting local physical errors. Another source arises from the prediction of global parity, which is obtained through a global average pooling of encoder output embeddings, followed by a prediction layer as shown in the Figure 6 top right. The global parity provides additional information on the final parity of the syndromes, which serves as auxiliary information that can improve the generalization of different syndrome patterns.

D. Transfer Learning

Lastly, for each type of code, there is a family of codes with different distances. Depending on the quantum algorithm in use, the required logical error rate will differ, thus requiring different code distances. Existing work often necessitates retraining for different code distances, which is costly in terms of both data collection and training time. Hence, in TT-QEC, we propose a transfer learning scheme to reuse the knowledge already learned. The reasoning behind this is the considerable similarity between different code distances; for example, handling syndromes of distance 5 code would be similar to handling a sub-block for distance 7 code. For a new distance, we directly apply the existing trained model, and fine-tune it on the new dataset. This ability to leverage transfer learning essentially stems from the transformer's capability to process input and output sequences of arbitrary sizes. The only

TABLE I

COMPARISON OF LOGICAL ERROR RATES UNDER DIFFERENT CODE DISTANCE AND PHYSICAL ERROR RATES.

Distance	Phys. Err. Rate	Logical Error Rate ↓			
		UF	MWPM	MLP	**TT-QEC**
3	0.0500	0.16745	0.14063	0.14794	**0.13005**
	0.0100	0.01039	0.00800	0.00903	**0.00784**
5	0.0500	0.24120	0.17279	0.20888	**0.17232**
	0.0100	0.00406	0.00268	0.00443	**0.00254**
7	0.0500	0.29813	0.20178	0.28454	**0.20590**
	0.0100	0.00113	0.00064	0.00197	**0.00059**
9	0.0500	0.35250	0.23161	0.32770	**0.23144**
	0.0100	0.00028	0.00002	0.00017	**0.00001**

aspect that needs careful handling is the positional encoding under the new distance.

V. EVALUATION

A. Evaluation Methodology

Benchmarks: We have selected the rotated surface code with distances of 3, 5, 7, 9. The round is set to be the same as the distance. The phenomenological error model [56] we use encompasses errors on syndrome measurement and data qubits. Each syndrome qubit experiences a measurement error with a probability p. The errors on data qubits are depolarizing errors, which occur with a probability p, causing Pauli X, Y, or Z errors with equal probability. As assumed in previous work [57], the error probabilities of these two types are considered to be equal. We choose values of p from the set 0.05, 0.01. The Google Stim package is used to construct the circuit and perform stabilizer simulations.

Baselines: Our three baselines include the Union Find decoder, the Minimum Weight Perfect Matching (MWPM) decoder as implemented in [52], and a Multi-Layer Perceptron (MLP) architecture. Following [58], our MLP architecture has two hidden layers, with the dimensions of these layers set empirically. As the MLP requires fixed-size inputs and generates fixed-size outputs, it does not facilitate transfer learning like the Transformer does.

Training Settings: Our main model is a Transformer with 6 layers, an embedding dimension of 256, 8 heads, and a feed-forward network (FFN) hidden dimension of 512. This model contains 7.9 million parameters. We also have a smaller model with 6 layers, an embedding dimension of 64, 2 heads, and an FFN hidden dimension of 128, which includes 0.5 million parameters. For training, we collect a dataset of 1,000,000 samples with a 1% error rate. We use a learning rate of 0.001 with linear warmup and cosine decay, a weight decay with lambda 0.0001, and we train for 100 epochs. We utilize the Adam optimizer with a weighted binary cross-entropy loss for local physical errors, and a normal binary cross-entropy loss for global parity errors. For the MLP model, we use a physical error rate of 1% for $d = 3, 5$ and 2.5% for $d = 7, 9$. Like the initial Transformer model, we train for 100 epochs with the Adam optimizer. Training is conducted on a single NVIDIA A6000 GPU.

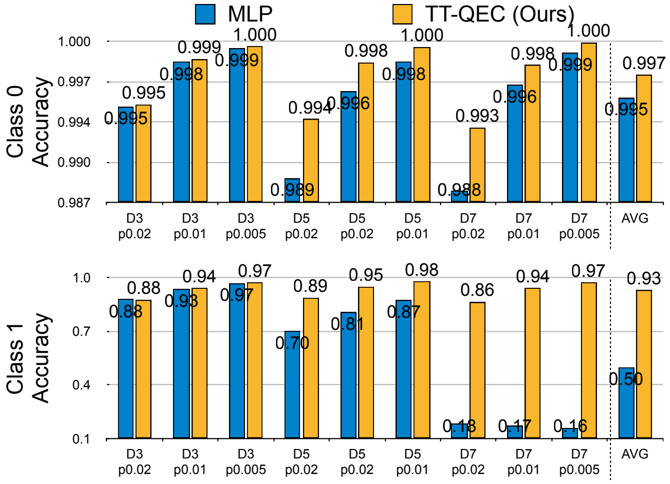

Fig. 7. Accuracy comparison between the TT-QEC an MLP baseline. Class 0 accuracy is the accuracy of correctly identify a no error data qubit as no error (True Negative). Class 1 accuracy is the accuracy of correctly identify an error when the data qubit has error (True Positive).

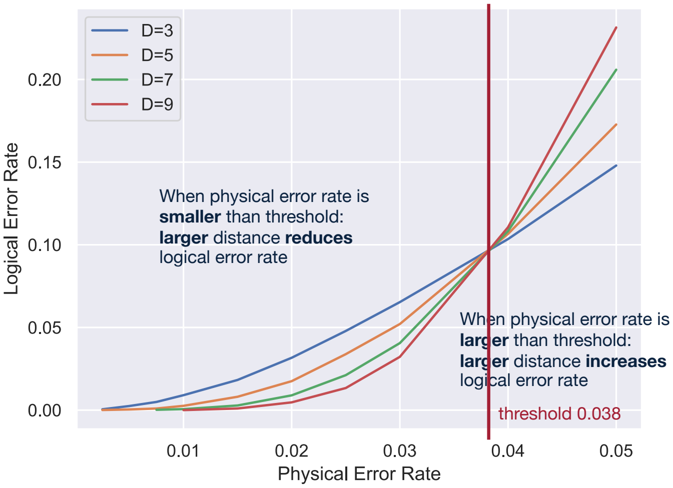

Fig. 8. Threshold of transformer based decoder. The threshold indicated the largest acceptable physical error rate for which using QEC can reduce error rate. Transformer obtains about 0.038 threshold.

Transfer Learning Settings: The distance 5 model, trained from scratch, is used as the source model for transfer learning. For all other distances, we use a constant learning rate of 0.0005 and train for 10 epochs. All other settings remain identical to the training from scratch.

B. Experiment Results

Main Results: Table I presents our primary results for varying code distances and physical error rates. Notice that all the models are transferred from the distance 5 model. In general, TT-QEC achieves a lower logical error rate for all benchmarks. The improvements over Union Find and MLP decoders are considerably more significant than the MWPM. This is likely because the global decoder in TT-QEC's frame-

TABLE II
COMPARISON OF LOGICAL ERROR RATE WITH GLOBAL LOSS.

Error Rate	0.0500	0.0400	0.0300	0.0250	0.0200
Local loss	0.17276	**0.10659**	0.05207	**0.03384**	0.01751
+ Global loss	**0.17232**	**0.10659**	**0.05196**	**0.03384**	**0.01744**
Error Rate	0.0150	0.0100	0.0075	0.0050	0.0025
Local loss	0.00808	0.00259	**0.00097**	0.00039	0.00007
+ Global loss	**0.00802**	**0.00254**	0.00103	**0.00035**	**0.00005**

work also employs an MWPM. The MLP model can surpass the Union Find but is generally not as good as MWPM and TT-QEC, even though the MLP models are trained individually for each code distance. This result highlights the effectiveness of our proposed transfer learning techniques. Furthermore, in Figure 7, we show the physical error prediction accuracy for baseline MLP and our TT-QEC. The class 0 accuracy means the ratio of predicted 0 when the ground truth is 0 (true negative). The class 1 accuracy means the ratio of predicted 1 when the ground truth is 1 (true positive). We can see that the accuracy for class 0 is in general much higher then class 1 because of the imbalance of training dataset. Moreover, our TT-QEC can achieve 43% higher accuracy on the class 1 which means the TT-QEC model can identify errors with much higher reliability. That is beneficial when we desire the low level decoder to clear as many as syndromes as possible and speedup the end-to-end process.

Evaluation of the Threshold: Figure 8 shows the threshold evaluation of TT-QEC, with the X-axis as physical and Y-axis as logical error rates. The curves of different distances intersect at a point where the physical error rate is 0.0038 and the logical error rate is around 0.09. When p is smaller than the threshold, larger code distances reduce the logical error rate. However, when p is larger than the threshold, larger distances do not help. Instead, we observe larger logical error rates. This trend can be attributed to the increased error introduced by larger system sizes, which eclipses the benefits of greater redundancy with more qubits.

Effectiveness of Mixed Loss: To evaluate the mixed loss function, we perform an ablation study on the distance 5 code, as shown in Table II. Each column shows the comparison of the logical error rate under a specific physical error rate. The performance with both local and global loss can achieve better or equivalent performance for nine out of ten cases. This demonstrates that the global parity loss provides valuable guidance during the model's training process.

Ablation on Model Size: We evaluate two models with different sizes but the same training setting in Table III under code distance 5 and varying physical error rates. It is evident that the larger model, with approximately 8 million parameters, outperforms the smaller model with 500 thousand parameters. The larger model is not overfitted to the training set and performs poorly on testing. This outcome is mainly due to the large size of the training set.

TABLE III
COMPARISON OF LOGICLA ERROR RATES UNDER DIFFERENT MODEL SIZE.

Error Rate	0.0200	0.0150	0.0100	0.0075	0.0050
503K Params	0.01812	0.00860	0.00290	0.00127	0.00045
7,911K Params	**0.01744**	**0.00802**	**0.00254**	**0.00103**	**0.00035**

VI. CONCLUSION

In conclusion, our study introduces an innovative and potent quantum error correction (QEC) decoder for rotated surface codes, harnessing the capabilities of machine learning and transformer model architecture. Rigorous evaluations reveal our decoder consistently surpasses existing benchmarks, exhibiting enhanced error correction for a range of code distances. Moreover, the transformer architecture also enables fast transfer learning between different code distance, amortizing the cost for model training. The integration of a global decoder and utilization of larger Transformer models are key in attaining these notable outcomes. This investigation lays the groundwork for future progress in ML-based Transformer decoders for stabilizer codes, fostering precision and promptness in quantum computations. Consequently, it significantly contributes to the evolution of dependable and proficient quantum computing systems during the NISQ era and beyond.

REFERENCES

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *STOC*, 1996, pp. 212–219.

[3] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[4] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, no. 1, pp. 1–7, 2014.

[5] J. Biamonte *et al.*, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[6] S. Lloyd *et al.*, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv preprint arXiv:1307.0411*, 2013.

[7] P. Rebentrost *et al.*, "Quantum support vector machine for big data classification," *Physical review letters*, vol. 113, no. 13, p. 130503, 2014.

[8] Z. Liang *et al.*, "Can noise on qubits be learned in quantum neural network? a case study on quantumflow," in *ICCAD*. IEEE, 2021, pp. 1–7.

[9] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, and S. Han, "Quantumnas: Noise-adaptive search for robust quantum circuits," in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2022, pp. 692–708.

[10] H. Wang, J. Gu, Y. Ding, Z. Li, F. T. Chong, D. Z. Pan, and S. Han, "Quantumnat: Quantum noise-aware training with noise injection, quantization and normalization," *arXiv preprint arXiv:2110.11331*, 2021.

[11] H. Wang, Z. Li, J. Gu, Y. Ding, D. Z. Pan, and S. Han, "Qoc: quantum on-chip training with parameter shift and gradient pruning," in *DAC*, 2022, pp. 655–660.

[12] Z. Liang, H. Wang, J. Cheng, Y. Ding, H. Ren, X. Qian, S. Han, W. Jiang, and Y. Shi, "Variational quantum pulse learning," *arXiv preprint arXiv:2203.17267*, 2022.

[13] Z. Liang, J. Cheng, H. Ren, H. Wang, F. Hua, Y. Ding, F. Chong, S. Han, Y. Shi, and X. Qian, "Pan: Pulse ansatz on nisq machines," *arXiv preprint arXiv:2208.01215*, 2022.

[14] H. Wang, P. Liu, J. Cheng, Z. Liang, J. Gu, Z. Li, Y. Ding, W. Jiang, Y. Shi, X. Qian *et al.*, "Quest: Graph transformer for quantum circuit reliability estimation," *arXiv preprint arXiv:2210.16724*, 2022.

[15] IBM, "Ibm unveils 400 qubit-plus quantum processor and next-generation ibm quantum system two."

[16] ——, "Ibm unveils breakthrough 127-qubit quantum processor."

[17] Rigetti, "Rigetti quantum," https://www.rigetti.com/.

[18] J. Kelly, "A preview of bristlecone, google's new quantum processor," https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html.

[19] J. Hsu, "Ces 2018: Intel's 49-qubit chip shoots for quantum supremacy."

[20] J. Lee *et al.*, "Even more efficient quantum computations of chemistry through tensor hypercontraction," *PRX Quantum*, vol. 2, no. 3, p. 030305, 2021.

[21] I. Kivlichan *et al.*, "Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via trotterization," *Quantum*, vol. 4, p. 296, 2020.

[22] C. Gidney and M. Ekerå, "How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits," *Quantum*, vol. 5, p. 433, 2021.

[23] C. Horsman *et al.*, "Surface code quantum computing by lattice surgery," *New Journal of Physics*, vol. 14, no. 12, p. 123011, 2012.

[24] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.

[25] Y. Ding and F. T. Chong, "Quantum computer systems: Research for noisy intermediate-scale quantum computers," *Synthesis Lectures on Computer Architecture*, vol. 15, no. 2, pp. 1–227, 2020.

[26] P. Krantz *et al.*, "A quantum engineer's guide to superconducting qubits," *Applied Physics Reviews*, vol. 6, no. 2, p. 021318, 2019.

[27] C. Bruzewicz *et al.*, "Trapped-ion quantum computing: Progress and challenges," *Applied Physics Reviews*, vol. 6, no. 2, p. 021314, 2019.

[28] E. Magesan *et al.*, "Characterizing quantum gates via randomized benchmarking," *Physical Review A*, vol. 85, no. 4, p. 042311, 2012.

[29] Q. IBM, Apr 2021. [Online]. Available: https://qiskit.org/textbook/ch-quantum-hardware/calibrating-qubits-pulse.html

[30] A. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2–30, jan 2003. [Online]. Available: https://doi.org/10.1016%2Fs0003-4916%2802%2900018-0

[31] H. Clare *et al.*, "Surface code quantum computing by lattice surgery," *New Journal of Physics*, vol. 14, no. 12, p. 123011, dec 2012. [Online]. Available: https://doi.org/10.1088%2F1367-2630%2F14%2F12%2F123011

[32] P. Das *et al.*, "Afs: Accurate, fast, and scalable error-decoding for fault-tolerant quantum computers," in *HPCA*. IEEE, 2022, pp. 259–273.

[33] G. Torlai and R. G. Melko, "Neural decoder for topological codes," *Phys. Rev. Lett.*, vol. 119, p. 030501, Jul 2017. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.119.030501

[34] C. Chamberland and P. Ronagh, "Deep neural decoders for near term fault-tolerant experiments," *Quantum Science and Technology*, vol. 3, no. 4, p. 044002, jul 2018. [Online]. Available: https://doi.org/10.1088%2F2058-9565%2Faad1f7

[35] S. Varsamopoulos, B. Criger, and K. Bertels, "Decoding small surface codes with feedforward neural networks," *Quantum Science and Technology*, vol. 3, no. 1, p. 015004, nov 2017. [Online]. Available: https://doi.org/10.1088/2058-9565/aa955a

[36] P. Baireuther *et al.*, "Machine-learning-assisted correction of correlated qubit errors in a topological code," *Quantum*, vol. 2, p. 48, Jan. 2018. [Online]. Available: https://doi.org/10.22331/q-2018-01-29-48

[37] S. Ryan *et al.*, "Reinforcement learning decoders for fault-tolerant quantum computation," *Machine Learning: Science and Technology*, vol. 2, no. 2, p. 025005, dec 2020. [Online]. Available: https://doi.org/10.1088/2632-2153/abc609

[38] P. Andreasson *et al.*, "Quantum error correction for the toric code using deep reinforcement learning," *Quantum*, vol. 3, p. 183, Sep. 2019. [Online]. Available: https://doi.org/10.22331/q-2019-09-02-183

[39] D. Fitzek *et al.*, "Deep q-learning decoder for depolarizing noise on the toric code," *Phys. Rev. Research*, vol. 2, p. 023230, May 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.2.023230

[40] L. Domingo Colomer, M. Skotiniotis, and R. Muñoz-Tapia, "Reinforcement learning for optimal error correction of toric codes," *Physics Letters A*, vol. 384, no. 17, p. 126353, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0375960120301638

[41] N. P. Breuckmann and X. Ni, "Scalable Neural Network Decoders for Higher Dimensional Quantum Codes," *Quantum*, vol. 2, p. 68, May 2018. [Online]. Available: https://doi.org/10.22331/q-2018-05-24-68

[42] X. Ni, "Neural network decoders for large-distance 2d toric codes," *Quantum*, vol. 4, p. 310, aug 2020. [Online]. Available: https://doi.org/10.22331%2Fq-2020-08-24-310

[43] K. Meinerz *et al.*, "Scalable neural decoder for topological surface codes," *Phys. Rev. Lett.*, vol. 128, p. 080505, Feb 2022. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.128.080505

[44] T. Wagner, H. Kampermann, and D. Bruß, "Symmetries for a high-level neural decoder on the toric code," *Phys. Rev. A*, vol. 102, p. 042411, Oct 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.102.042411

[45] V. Savvas *et al.*, "Comparing neural network based decoders for the surface code," *IEEE Transactions on Computers*, vol. 69, no. 2, pp. 300–311, feb 2020. [Online]. Available: https://doi.org/10.1109%2Ftc.2019.2948612

[46] S. Gicev, L. C. L. Hollenberg, and M. Usman, "A scalable and fast artificial neural network syndrome decoder for surface codes," *arXiv e-prints*, p. arXiv:2110.05854. [Online]. Available: https://arxiv.org/abs/2110.05854

[47] C. C. *et al.*, "Techniques for combining fast local decoders with global decoders under circuit-level noise," 2022.

[48] C. Chamberland and P. Ronagh, "Deep neural decoders for near term fault-tolerant experiments," *Quantum Science and Technology*, vol. 3, no. 4, p. 044002, jul 2018. [Online]. Available: https://doi.org/10.1088/2058-9565/aad1f7

[49] S. Varsamopoulos *et al.*, "Decoding surface code with a distributed neural network–based decoder," *Quantum Machine Intelligence*, vol. 2, no. 1, p. 2524–4914, 2020. [Online]. Available: https://doi.org/10.1007/s42484-020-00015-9

[50] M. Sheth, S. Z. Jafarzadeh, and V. Gheorghiu, "Neural ensemble decoding for topological quantum error-correcting codes," *Phys. Rev. A*, vol. 101, p. 032338, Mar 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.101.032338

[51] A. Fowler *et al.*, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A*, vol. 86, no. 3, p. 032324, 2012.

[52] Y. Wu and L. Zhong, "Fusion blossom: Fast mwpm decoders for qec," *arXiv preprint arXiv:2305.08307*, 2023.

[53] N. Delfosse and G. Zémor, "Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel," *Quantum Information & Computation*, 2017.

[54] Y. Tomita and K. M. Svore, "Low-distance surface codes under realistic quantum noise," *Phys. Rev. A*, vol. 90, p. 062320, Dec 2014. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.90.062320

[55] C. T. Chubb, "General tensor network decoding of 2d pauli codes," 2021.

[56] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4452–4505, 2002.

[57] P. Das, A. Locharla, and C. Jones, "Lilliput: A lightweight low-latency lookup-table based decoder for near-term quantum error correction," 2021.

[58] R. W. J. Overwater, M. Babaie, and F. Sebastiano, "Neural-network decoders for quantum error correction using surface codes: A space exploration of the hardware cost-performance tradeoffs," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–19, 2022. [Online]. Available: https://doi.org/10.1109%2Ftqe.2022.3174017