Spatio-Temporal Attention Gated Transformer for Memory Uncorrectable Error Prediction

Jianli Ding[0009-0004-8724-1372], Guohao Chao[0009-0006-6101-5005], and Jing Li*[0000-0002-7940-5582]
School of Computer Science and Artificial Intelligence, Civil Aviation University of China, Tianjin, China
Email: {jlding, 2024052077}@cauc.edu.cn; 17674776513@163.com

***Abstract*—Uncorrectable Errors (UEs) in memory are a critical trigger for hardware failures and service disruptions in modern data centers. Existing prediction methods typically rely on rule-based feature engineering or traditional machine learning models, which struggle to effectively capture the complex spatio-temporal evolution patterns of memory errors while ignoring the spatial correlation of memory errors in bit-level physical topology. To address this, this paper proposes an end-to-end deep learning model named Spatio-Temporal Attention Gated Transformer (STAGT). Firstly, the model adopts decoupled positional encoding to explicitly preserve physical topology information. Secondly, it constructs a spatio-temporal decoupled attention architecture, leveraging spatial attention and adjacency masks to accurately capture local correlations between error bits. On this basis, a spatial global gating mechanism is introduced to adaptively adjust feature weights according to the overall spatial state. Finally, temporal attention is integrated to model the progressive evolution patterns of failures. Experimental results on real production environment datasets demonstrate that STAGT outperforms existing mainstream methods in all performance metrics.**

***Index Terms*—spatio-temporal attention, gating mechanism, Transformer, uncorrectable errors**

I. Introduction

As modern data centers serve as the core infrastructure for cloud computing and big data services, their system reliability is directly related to Quality of Service (QoS) and user experience. Under such large-scale distributed environments, hardware failures have become a major bottleneck restricting system stability [1]. Among these, memory component failures account for 37% of all hardware failures in data centers [2]. Although Error-Correcting Code (ECC) technology can alleviate some Correctable Errors (CEs), when error accumulation evolves into Uncorrectable Errors (UEs), they often lead to severe server downtime and data loss [3]. Therefore, accurately predicting the occurrence of UEs by utilizing historical data (CEs) is crucial for the transition from "passive repair" to "proactive maintenance". Although existing prediction methods (ranging from traditional machine learning to deep learning) have achieved certain progress, they still face two key challenges in practical production deployment. Firstly, existing models typically process spatio-temporal information in a mixed manner, struggling to effectively decouple temporal evolution and spatial distribution. They overlook the explicit bit-level modeling of DRAM, resulting in difficulty in capturing spatial local correlations [4], [5]. To verify this phenomenon, this paper conducts an empirical analysis of production environment data (as shown in Fig. 1). Statistical results indicate that the relative probability of Uncorrectable Error (UE) occurrence is significantly higher when adjacent bit errors exist compared to the scenario without adjacent bit errors. Secondly, existing architectures lack macroscopic perception of the global spatial state. Faced with complex and diverse bit-level error patterns [6], [7], existing methods cannot adaptively adjust the model's focus according to the severity of the error distribution, thereby reducing the model's robustness to various fault patterns.

To address the aforementioned challenges, this paper proposes an end-to-end deep learning model—Spatio-Temporal Attention-Gated Transformer (STAGT). Through innovative architectural design, this model aims to accurately capture the spatio-temporal evolution patterns of memory failures. The main contributions of this paper are summarized as follows:

(1) Proposing a Decoupled Dual Attention Mechanism: A physically topology-aware spatial attention is designed to model error propagation, and an independent temporal attention is developed to learn fault evolution. This mechanism effectively avoids mutual interference between spatio-temporal features and significantly enhances the model's ability to represent complex spatio-temporal patterns.

(2) Designing a Spatial Global Gating Architecture: By performing temporal modeling on the spatial states of 32 bits, a gating mechanism is introduced to adaptively adjust feature weights. This enables the model to dynamically adjust response strategies according to the overall fault severity, enhancing robustness to diverse fault patterns.

I will provide my data/code after acceptance.

II. Related Work

A. Traditional Prediction Methods

Early research on memory fault prediction mainly relied on threshold-based heuristic rules. The pioneering work by Schroeder and Gibson [8] systematically analyzed hardware failure patterns in large-scale systems for the first time. Building on this, Schroeder [9] further observed from production environment data that when the cumulative number of CEs exceeds a threshold within a specific time window, the probability of UEs occurring increases significantly. Based on this finding, many data centers have deployed static CE count threshold strategies as early warning mechanisms. However,

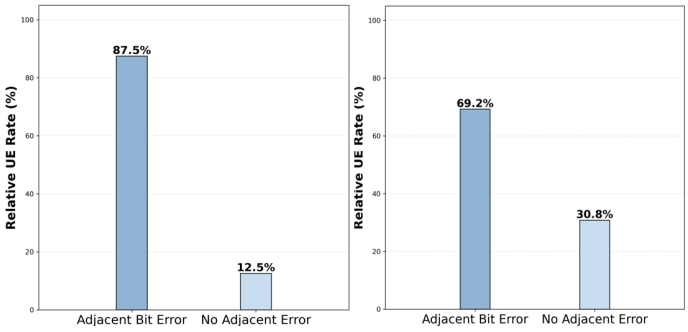

Fig. 1. The left subfigure corresponds to Dataset 1, and the right subfigure corresponds to Dataset 2. Analysis of the impact of adjacent bit errors on the occurrence probability of Uncorrectable Errors (UEs) across different datasets.

this method has poor generalization ability—research by Meza et al. [10] shows that the setting of the optimal threshold varies drastically across different hardware environments and workloads, making it difficult to standardize. To overcome the limitations of static thresholds, subsequent studies have gradually shifted to traditional machine learning methods, including cost-sensitive learning [11], online learning [12], workload-aware prediction [13], and proactive replacement-oriented prediction strategies [14]. Although these methods improve the accuracy to a certain extent, it is difficult to effectively capture the complex nonlinear spatio-temporal dependencies in memory failures.

B. Prediction Methods Based on Fine-Grained Features

In recent years, fault prediction using fine-grained spatio-temporal features has emerged as a research hotspot. Li et al. [6] and Yu et al. [7] have thoroughly explored the bit-level information in the evolution process from Correctable Errors (CEs) to Uncorrectable Errors (UEs), demonstrating that the utilization of fine-grained error bit patterns can effectively improve prediction performance. Based on these findings, Yu et al. [15] proposed a hierarchical intelligent memory failure prediction model (HiMFP), attempting to integrate multi-level features for system-level prediction; Gu et al. [20] further analyzed the differences in fault patterns under the X86 and ARM architectures, and proposed a multi-grained fault prediction framework MemSeer. Furthermore, to address the interference of hardware and software redundancy technologies on error features in production environments, Zheng et al. [21] proposed a novel UPH indicator system.

Although the aforementioned methods based on fine-grained feature engineering have achieved remarkable progress, they lack the ability to perform end-to-end representation learning on raw error sequences, which limits the model's ability to perceive the underlying spatio-temporal evolution patterns. To address this limitation, researchers have begun to explore end-to-end deep learning architectures. The Transformer architecture [16], with its self-attention mechanism, has demonstrated outstanding performance in addressing long-sequence dependency issues. The STIM model proposed by Liu et al. [17] pioneered the application of the Transformer to UE prediction, and by directly processing the DQ-Beat matrix, it has outperformed traditional machine learning methods in terms of performance. However, the STIM model still has notable limitations: firstly, the standard positional encoding it adopts fails to distinguish between time steps and spatial positions, resulting in entangled spatio-temporal features; secondly, this architecture lacks explicit modeling of the internal physical topology of DRAM. These shortcomings limit the model's ability to capture complex fault propagation patterns and have provided a clear direction for improvement for the Spatio-Temporal Attention Gated Transformer (STAGT) proposed in this paper.

III. Method

A. Method Overview

The overall architecture of STAGT is illustrated in Fig. 2. We formalize memory fault prediction as a temporal binary classification task. The input is the error feature sequence within the observation window, denoted by $X = [x_1, x_2, ..., x_T] \in \mathbb{R}^{T \times F}$, where T is the time step (in this study, the 72-hour observation window is aggregated hourly, i.e., $T = 72$), and $F = 44$ represents the feature dimension. The feature vector of each time step $x_t \in \mathbb{R}^F$ includes: 32-dimensional bit features, obtained by expanding the 4×8 DQ-Beat matrix, recording the error distribution at each position on the memory chip; 12-dimensional micro features, covering error statistics, server parameters, and component-level fault information. The goal of the model is to predict whether the DIMM module will experience UEs within the future prediction window.

The STAGT model adopts an end-to-end deep learning architecture, consisting of five main components: input embedding layer, decoupled positional encoding, dual attention mechanism, feature fusion layer, and classifier. First, the original feature sequence within the 72-hour observation window is divided into bit features and micro features, which are converted into high-dimensional representations through independent linear projections. Then, the decoupled positional encoding module injects positional information into different types of features: bit features are injected with dual positional encoding of time and space to characterize both their temporal position and physical coordinates; while micro features only add temporal positional encoding due to the lack of spatial topology attributes. Next, the dual attention mechanism models the dependencies in spatial and temporal dimensions respectively: spatial attention captures the local correlation patterns between bit features, and temporal attention learns the evolution patterns of all features within the observation window. The feature fusion layer integrates the attention-enhanced features to generate a unified representation. Finally, the classifier predicts UE/CE through a multi-layer perceptron.

B. Decoupled Positional Encoding Mechanism

Traditional positional encoding usually couples spatio-temporal information into a single vector, making it difficult

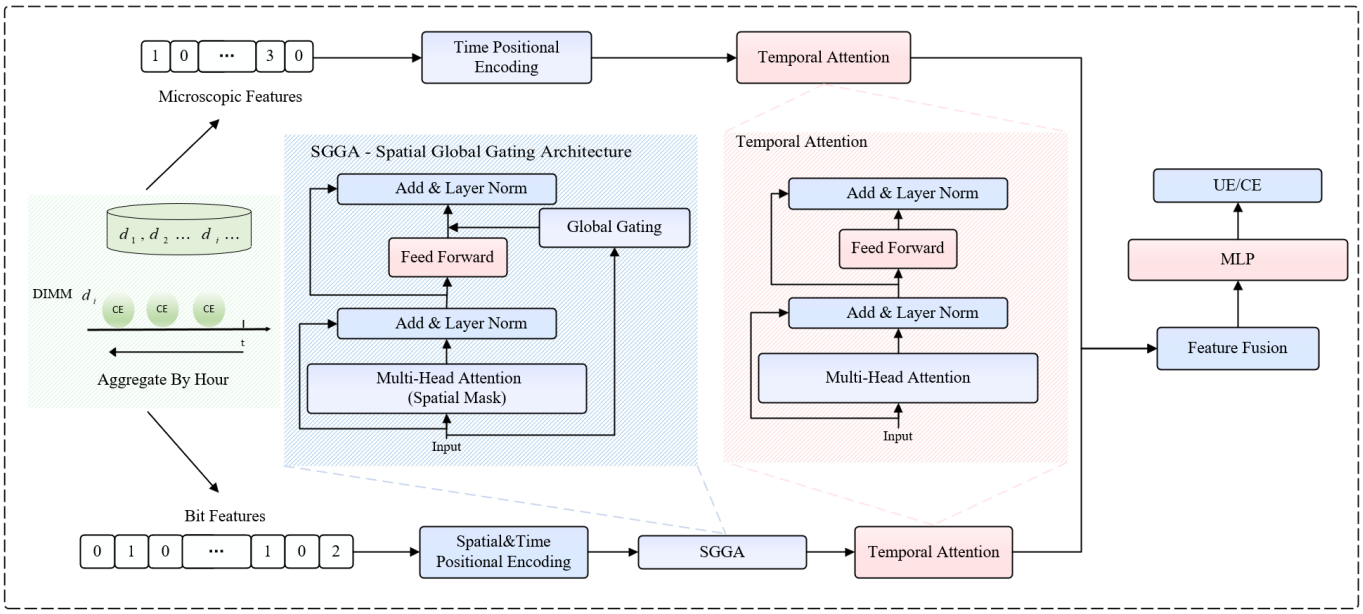

Fig. 2. The processing flow and network structure of STAGT.

to effectively characterize the complex physical topology of memory. Given that the temporal evolution patterns and spatial distribution characteristics of memory errors follow completely different physical mechanisms, this paper designs Decoupled Positional Encoding to model temporal and spatial dimensions independently.

1) Temporal Positional Encoding: For a sequence of length T, the temporal positional encoding is defined as:

$$\mathrm{PE}_{\text{temporal}}(t,i) = \begin{cases} \sin\left(\frac{t}{10000^{2i/d}}\right) & \text{if } i \text{ is even} \\ \cos\left(\frac{t}{10000^{2(i-1)/d}}\right) & \text{if } i \text{ is odd} \end{cases} \tag{1}$$

where t is the time step index, i is the dimension index, and d is the encoding dimension.

2) Spatial Positional Encoding: For the 4×8 grid structure of 32-bit features, spatial positional encoding needs to explicitly model the two-dimensional physical topology relationship. For the k-th bit ($k \in \{0,1,...,31\}$), its position in the grid can be represented as (r_k, c_k). Then the spatial positional encoding is defined as:

$$\mathrm{PE}_{\text{spatial}}(r_k,c_k,i) = \begin{cases} \frac{r_k}{4}, & \text{if } i < \frac{d}{2} \\ \frac{c_k}{8}, & \text{if } i \geq \frac{d}{2} \end{cases} \tag{2}$$

where $r_k \in \{0,1,2,3\}$ represents the row index, $c_k \in \{0,1,...,7\}$ represents the column index. The normalization terms $\frac{r_k}{4}$ and $\frac{c_k}{8}$ aim to map spatial coordinates to a numerical range similar to temporal encoding, to ensure numerical stability during model training. The final positional encoding is as follows:

$$\begin{aligned} \mathrm{PE}_{\text{bit}}(t,r,c) &= \mathrm{PE}_{\text{temporal}}(t) + \mathrm{PE}_{\text{spatial}}(r,c) \\ \mathrm{PE}_{\text{micro}}(t) &= \mathrm{PE}_{\text{temporal}}(t) \end{aligned} \tag{3}$$

C. Fused Spatio-Temporal Attention Mechanism

Memory failures exhibit significant spatio-temporal dual nature. Traditional fully connected attention mechanisms overlook this physical constraint, leading the model to learn spurious long-range dependencies. To address this, this paper designs a dual attention architecture that decouples the spatio-temporal modeling process into two cascaded stages: first, bit features capture neighborhood correlations on the 4×8 grid through spatial attention to generate a spatially enhanced feature sequence, after which the temporal evolution of these patterns is modeled. Simultaneously, micro features directly capture the progressive evolution process of UEs through the temporal attention mechanism.

1) Spatial Attention and Adjacency Mask: To capture the bit-level local correlations of memory errors, this paper designs a spatial adjacency mask $M_{\text{spatial}} \in \mathbb{R}^{32\times32}$ to constrain the attention calculation. The element $M_{\text{spatial}}[i,j]$ of the mask matrix denotes whether the i-th bit can attend to the j-th bit ($i,j \in \{0,1,...,31\}$). The adjacency mask is defined as:

$$\mathbf{M}_{\text{spatial}}(i,j) = \begin{cases} 0, & \text{if } |r_i - r_j| + |c_i - c_j| \leq 1 \\ -\infty, & \text{otherwise} \end{cases} \tag{4}$$

As shown in Fig. 3, this mask ensures that each bit can only attend to its 4-neighborhood (up, down, left, right) and itself.

The calculation formula of spatial attention is:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + M_{\text{spatial}}\right)V \tag{5}$$

where Q, K, V are the query, key, and value matrices respectively.

2) Spatial Global Gating Mechanism: In memory fault prediction tasks, the spatial error distribution across 32 bits exhibits high complexity and diversity. Traditional Transformer

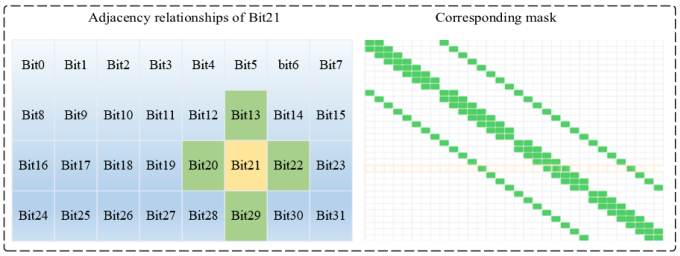

Fig. 3. The left side shows the 4-neighborhood structure centered on Bit 21 (up, down, left, right, and itself); the right side shows the corresponding sparse mask matrix visualization, with green blocks indicating the positions where attention interactions are allowed.

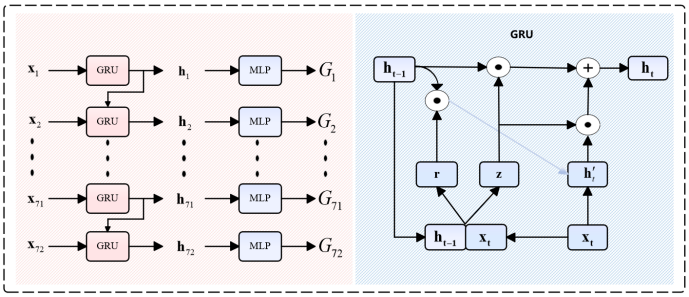

Fig. 4. On the left is the historical accumulation pattern that uses GRU to capture spatial states, and on the right are the internal details of the GRU.

architectures directly apply residual connections after spatial attention; this fixed-weight fusion approach lacks global awareness of the overall spatial state and cannot dynamically adjust the contribution of the Feed-Forward Network (FFN) based on the current spatial state. To address this, this paper proposes a spatial global gating architecture, which adopts a "pre-evaluation and post-adjustment" strategy: first, based on the spatial distribution of raw bit features, a Gated Recurrent Unit (GRU) is used to capture the historical cumulative patterns of spatial states and generate temporal gating scores; subsequently, the contribution weight of the FFN is dynamically adjusted. This design ensures that the GRU computation for gating scores is performed only once, while spatial pooling based on raw features can capture purer spatial distribution patterns.

Given the input bit feature $H_{\text{input}}^{(l)} \in \mathbb{R}^{T\times 32\times d}$ for the l-th layer, we first perform average pooling over the 32 bits at each time step to obtain the overall spatial representation of that time step:

$$x_t^{(l)} = \text{MeanPool}(H_{\text{input}}^{(l)}[t]) \tag{6}$$

Then, a unidirectional GRU is used to model the time series $\{x_1^{(l)}, x_2^{(l)}, ..., x_T^{(l)}\}$ (as shown in Fig. 4) to capture the historical accumulation pattern of spatial states:

$$h_t^{(l)} = \text{GRU}(x_t^{(l)}, h_{t-1}^{(l)}) \tag{7}$$

where $h_{t-1}^{(l)} \in \mathbb{R}^d$ is the hidden state of GRU at the previous time step.

Next, the GRU output is mapped to a scalar gating score $G_t^{(l)}$ through a multi-layer perceptron, and finally the gating score after exponential transformation is used to adjust the contribution of FFN output to the final representation:

$$H^{(l)}[t] = H_{\text{spatial}}^{(l)}[t] + \exp(G_t^{(l)} - 0.5) \cdot \text{FFN}(H_{\text{spatial}}^{(l)}[t]) \tag{8}$$

where $H_{\text{spatial}}^{(l)}[t]$ is the output after spatial attention. The introduction of the exponential transformation $\exp(G_t^{(l)} - 0.5)$ is designed to overcome the limitation of the conventional Sigmoid activation function $[0, 1]$, enabling the gating coefficient to adaptively enhance (> 1) or suppress (< 1) the information flow contribution of the Feed-Forward Network (FFN) according to the severity of the spatial state.

3) Temporal Attention Modeling: In memory fault occurrence, there exists a significant temporal cumulative effect: UEs are often accompanied by sustained growth in CE counts and shortening of fault intervals, and abrupt changes in error patterns (e.g., shifting from sporadic to high-frequency) are important precursors to UEs. To capture these temporal evolution features, this paper applies temporal attention separately to the bit features enhanced by spatial gating and the micro features, learning their evolution patterns within the observation window.

It is worth noting that although the GRU in the global gating also involves temporal information processing, there is an essential difference between it and the temporal attention module in this section: the GRU aims to model the temporal evolution of the global spatial state (i.e., the overall statistics of the 32 bits) to generate scalar gating weights; in contrast, temporal attention focuses on fine-grained features, independently learning the temporal evolution dependencies of each bit in the time dimension.

For the i-th feature (bit), temporal attention first computes its query Q_i, key K_i, and value V_i matrices across all time steps, then obtains the temporally enhanced feature representation via the self-attention mechanism:

$$\text{Attention}_i(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d}}\right) V_i \tag{9}$$

The same temporal attention calculation is adopted for micro features. Finally, the temporal attention outputs of bit features and micro features are used for subsequent feature fusion.

IV. Experimental Setup and Result Analysis

This section aims to validate the performance of the STAGT model using real production environment data. We will elaborate on the characteristics of the dataset used in the experiments, evaluation metrics, and the configuration of baseline models in detail. Regarding the experimental setup, the Focal Loss function and AdamW optimizer are adopted, with the batch size consistently set to 32, the initial learning rate set to 8×10^{-6}, and a 10-round early stopping mechanism configured to ensure the stability and convergence of training.

TABLE I
THE DATA COMES FROM HUAWEI CLOUD'S PRODUCTION ENVIRONMENT AND COVERS TWO SERVER PLATFORMS: INTEL PURLEY AND WHITLEY.

Statistical Item	**Dataset 1**	**Dataset 2**
Server Platform	Intel Purley	Whitley
Faulty DIMMs in Training Set (Jan.-Mar.)	470	39
Faulty DIMMs in Test Set (Apr.-May)	298	29
Total Faulty DIMMs	768	69
Total DIMMs	56,403	5,821

A. Dataset Introduction

This study adopts a large-scale memory fault dataset from Huawei Cloud's production environment (see Table I). The dataset is collected from two distinct server platforms, with a time span covering January to May 2024. Each sub-dataset consists of two parts: memory error logs (CE logs) and failure tickets. Specifically, the memory error logs record CE events of each DIMM during system operation, including the timestamp of error occurrence, DIMM-level physical locations (such as Rank, Bank, Row, and Column), hardware configuration information (hardware specification parameters like DIMM model, CPU capacity, and manufacturer), and detailed error bit-level features (a 32-bit binary sequence indicating error positions in the DQ-Beat matrix of x4 DDR4 chips). A time-series splitting strategy is employed in this experiment: the data from the first three months is used as the training set, and the data from the last two months serves as the test set.

B. Performance Metrics

Memory fault prediction can essentially be formulated as a binary classification problem in the field of machine learning. Its core objective is to determine the likelihood of UEs occurring within a predefined future time window by analyzing the occurrence patterns of historical CEs, as illustrated in Fig. 5. At the current time t, the algorithm observes historical data from an observation window of Δt_d to predict faults within the prediction horizon $[t + \Delta t_l, t + \Delta t_l + \Delta t_p]$, where Δt_l denotes the minimum time interval between prediction and potential fault occurrence. This interval serves as a time buffer for the system to implement fault response and preventive measures. Within the prediction window Δt_p: True Positive (TP) refers to correctly predicted faults, False Positive (FP) denotes false alarms (incorrectly predicted faults),False Negative (FN) represents faults occurring without prior warning,True Negative (TN) indicates correctly identified non-fault states.

We evaluate the model performance using the following metrics: Precision $= \frac{\text{TP}}{\text{TP+FP}}$, Recall $= \frac{\text{TP}}{\text{TP+FN}}$, F1 $= \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}}$.

C. Comparison Experiments

This study selects the following baseline methods for comparison:(1) Risky CE [7]: A feature extraction method based on error bit risk patterns, using the LightGBM classifier;(2) HiMFP [15]: Integrates DIMM-level and bit-level count and pattern features for system-level prediction;(3) STIM [17]: A Transformer-based spatio-temporal feature learning method

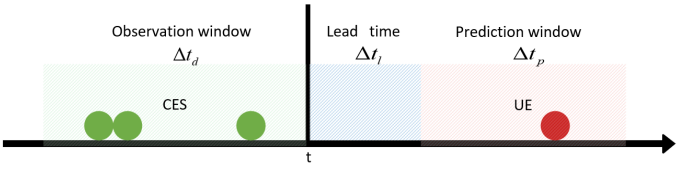

Fig. 5. Definition of the Prediction Problem. Δt_l is set to 15 minutes, Δt_d is set to 3 days, Δt_p is set to 7 days.

TABLE II
COMPARISON OF PERFORMANCE INDICATORS FOR DIFFERENT FAULT PREDICTION METHODS.

Models	**Dataset 1**			**Dataset 2**		
	Precision	Recall	F1-score	Precision	Recall	F1-score
STIM	15.29%	20.47%	17.50%	10.26%	27.59%	14.96%
HiMFP	26.27%	32.89%	29.21%	20.00%	34.48%	25.32%
Risky CE	27.27%	34.23%	30.36%	18.97%	37.93%	25.29%
LSTM	16.03%	25.50%	19.69%	12.70%	27.59%	17.39%
CNN+Transformer	22.45%	28.86%	25.26%	16.00%	27.59%	20.25%
STAGT	32.49%	38.93%	35.42%	21.28%	34.48%	26.32%

that inputs DQ-Beat error bit matrices into a Transformer encoder for UE prediction. The experimental settings of the aforementioned existing methods strictly follow the descriptions in their original papers. In addition, this paper constructs two additional deep learning baseline models:(4) CNN+Transformer: Uses a 1D-Convolutional Neural Network (1D-CNN) [18] to extract local spatial features from bit sequences, followed by integrating a Transformer to learn temporal dependencies;(5) LSTM: Adopts the Long Short-Term Memory [19] network, which excels at capturing long-range temporal dependencies and serves as a widely used recurrent neural network baseline in temporal fault prediction tasks.

Table II presents a performance comparison of different methods on two datasets. Note that the generally lower performance on Dataset 2 is attributed to the extreme scarcity of fault samples, which limits the generalization capability of all models.Although rule-based methods (e.g., Risky CE, HiMFP) exhibit strong competitiveness, they still suffer from performance bottlenecks when coping with complex and variable fault patterns. Among deep learning models, CNN+Transformer outperforms LSTM and STIM significantly; this result underscores the importance of spatial information, indicating that relying solely on temporal features or limited spatial perception is insufficient to capture complex fault patterns. The STAGT model achieves more refined modeling by introducing physically topology-aware adjacency masks, a spatio-temporal feature decoupling mechanism, and a spatial global gating mechanism. By integrating these targeted improvements into a unified end-to-end framework, STAGT achieves state-of-the-art prediction performance.

D. Ablation Experiments

To verify the effectiveness of each key component of the model, we designed progressive ablation experiments (see Table III), starting from the basic temporal attention module and gradually adding core innovative components: TA (Temporal

TABLE III
IMPACT OF PROGRESSIVE ABLATION EXPERIMENTS ON MODEL PERFORMANCE.

Models	Dataset 1			Dataset 2		
	Precision	Recall	F1-score	Precision	Recall	F1-score
TA	18.25%	24.50%	20.92%	12.50%	27.59%	17.20%
STA	22.02%	27.85%	24.59%	15.00%	31.03%	20.22%
STA+Mask	27.35%	34.23%	30.40%	17.54%	34.48%	23.26%
STAGT	32.49%	38.93%	35.42%	21.28%	34.48%	26.32%

Attention): a basic temporal attention Transformer encoder, adopting the standard multi-head self-attention mechanism to capture temporal dependencies. STA (Spatio-Temporal Attention): introducing spatial attention mechanism and decoupled positional encoding on the basis of TA to model the spatial dependencies between memory bit features. STAM (STA + Mask): adding 4×8 spatial adjacency mask constraints on the basis of STA to limit attention calculation only between adjacent bits, so as to capture the locality features of memory fault propagation. STAGT (complete model): introducing the global gating mechanism on the basis of STAM, including all components.

V. CONCLUSION AND FUTURE WORK

With the rapid advancement of cloud computing and big data technologies, memory failures have become a critical factor affecting data center reliability. The accurate prediction of UEs is of great significance for implementing preventive maintenance and improving system reliability. To address the limitations of existing memory fault prediction methods in spatio-temporal feature modeling, global awareness capability, and adaptive adjustment, this study proposes the spatio-temporal Attention-Gated Transformer (STAGT) model, establishing an end-to-end deep learning framework for memory fault prediction. However, this research still has room for improvement in terms of sample diversity and cross-platform validation. On the one hand, constrained by the extreme scarcity of fault samples in production environments, the model's robustness in few-shot scenarios still faces challenges; on the other hand, the current validation has not yet covered a wider range of heterogeneous server architectures. To address this, future work will focus on exploring data augmentation and transfer learning technologies to enhance the model's generalization capability in complex heterogeneous environments.

ACKNOWLEDGMENT

This research was funded by the Special Fund of the Fundamental Research Funds for the Central Universities at Civil Aviation University of China (Project No. 3122026053).

REFERENCES

[1] G. Wang, L. Zhang, and W. Xu, "What can we learn from four years of data center hardware failures?" in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks (DSN)*, Denver, CO, USA, 2017, pp. 25–36.

[2] "Intel MCA+MFP Helps JD Stable and Efficient Cloud Services." [Online]. Available: https://www.intel.com/content/dam/www/central-libraries/us/en/documents/memory-failure-prediction-at-jd-cloud-us.pdf [Accessed: 2026-01-19].

[3] Z. Cheng, S. Han, P. P. Lee, X. Li, J. Liu, and Z. Li, "An in-depth correlative study between DRAM errors and server failures in production data centers," in *Proc. 41st Int. Symp. Reliable Distrib. Syst. (SRDS)*, Vienna, Austria, 2022, pp. 262–272.

[4] V. Sridharan and D. Liberty, "A study of DRAM failures in the field," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. (SC '12)*, Salt Lake City, UT, USA, 2012, pp. 76:1–76:11.

[5] A. A. Hwang, I. A. Stefanovici, and B. Schroeder, "Cosmic rays don't strike twice: Understanding the nature of DRAM errors and the implications for system design," in *Proc. 17th Int. Conf. Architect. Support Program. Lang. Oper. Syst. (ASPLOS)*, London, UK, 2012, pp. 111–122.

[6] C. Li et al., "From correctable memory errors to uncorrectable memory errors: What error bits tell," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. (SC22)*, Dallas, TX, USA, 2022, pp. 1–14.

[7] Q. Yu, W. Zhang, J. Cardoso, and O. Kao, "Exploring error bits for memory failure prediction: An in-depth correlative study," in *Proc. 2023 IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, San Francisco, CA, USA, 2023, pp. 1–9.

[8] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," *IEEE Trans. Dependable Secure Comput.*, vol. 7, no. 4, pp. 337–350, 2010.

[9] B. Schroeder, E. Pinheiro, and W.-D. Weber, "DRAM errors in the wild: A large-scale field study," in *Proc. 11th Int. Joint Conf. Meas. Model. Comput. Syst. (SIGMETRICS '09)*, Seattle, WA, USA, 2009, pp. 193–204.

[10] V. Sridharan et al., "Memory errors in modern systems: The good, the bad, and the ugly," in *Proc. 20th Int. Conf. Architect. Support Program. Lang. Oper. Syst. (ASPLOS '15)*, Istanbul, Turkey, 2015, pp. 297–310.

[11] I. Boixaderas et al., "Cost-aware prediction of uncorrected DRAM errors in the field," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal. (SC '20)*, Atlanta, GA, USA, 2020, pp. 1–15.

[12] X. Du and C. Li, "Memory failure prediction using online learning," in *Proc. Int. Symp. Memory Syst. (MEMSYS '18)*, Washington, DC, USA, 2018, pp. 38–49.

[13] L. Mukhanov, K. Tovletoglou, H. Vandierendonck, D. S. Nikolopoulos, and G. Karakonstantis, "Workload-aware DRAM error prediction using machine learning," in *Proc. 2019 IEEE Int. Symp. Workload Characteriz. (IISWC)*, Orlando, FL, USA, 2019, pp. 106–118.

[14] X. Du, C. Li, S. Zhou, M. Ye, and J. Li, "Predicting uncorrectable memory errors for proactive replacement: An empirical study on large-scale field data," in *Proc. 16th Eur. Dependable Comput. Conf. (EDCC)*, Munich, Germany, 2020, pp. 41–46.

[15] Q. Yu et al., "HiMFP: hierarchical intelligent memory failure prediction for cloud service reliability," in *Proc. 53rd Annu. IEEE/IFIP Int. Conf. Dependable Syst. Networks (DSN)*, Porto, Portugal, 2023, pp. 216–228.

[16] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[17] Z. Liu, "STIM: predicting memory uncorrectable errors with spatio-temporal transformer," *Microsoft Journal of Applied Research*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:267755744 [Accessed: 2026-01-19].

[18] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *J. Syst. Eng. Electron.*, vol. 28, no. 1, pp. 162–169, 2017.

[19] J. Xu, Z. Zhang, T. Chun, J. Yang, and T. Li, "Hard drive failure prediction using LSTM networks," in *Proc. 2016 IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, Ottawa, ON, Canada, 2016, pp. 1–6.

[20] Y. Gu et al., "MemSeer: leverage memory failure distinctions and multi-grained prediction in ultra-scale heterogeneous x86/ARM clusters," in *Proc. 62nd ACM/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2025.

[21] L. Qin et al., "An effective uncorrectable memory error prediction framework by exploiting UPH indicators in production environments," in *Proc. 2025 IEEE Int. Parallel and Distributed Processing Symp. (IPDPS)*, 2025.