

CE2Graph: A Diagnosis-Oriented Spatio-Temporal Graph Framework for DRAM Failure Prediction

Cong Ma¹, Xinyi He², Jianwang Zhai¹, and Kang Zhao¹

{mc, zhaijw, zhaokang}@bupt.edu.cn

hexinyi@sdic.com.cn

¹Beijing University of Posts and Telecommunications

²China Electronics Engineering Design Institute Co., Ltd.



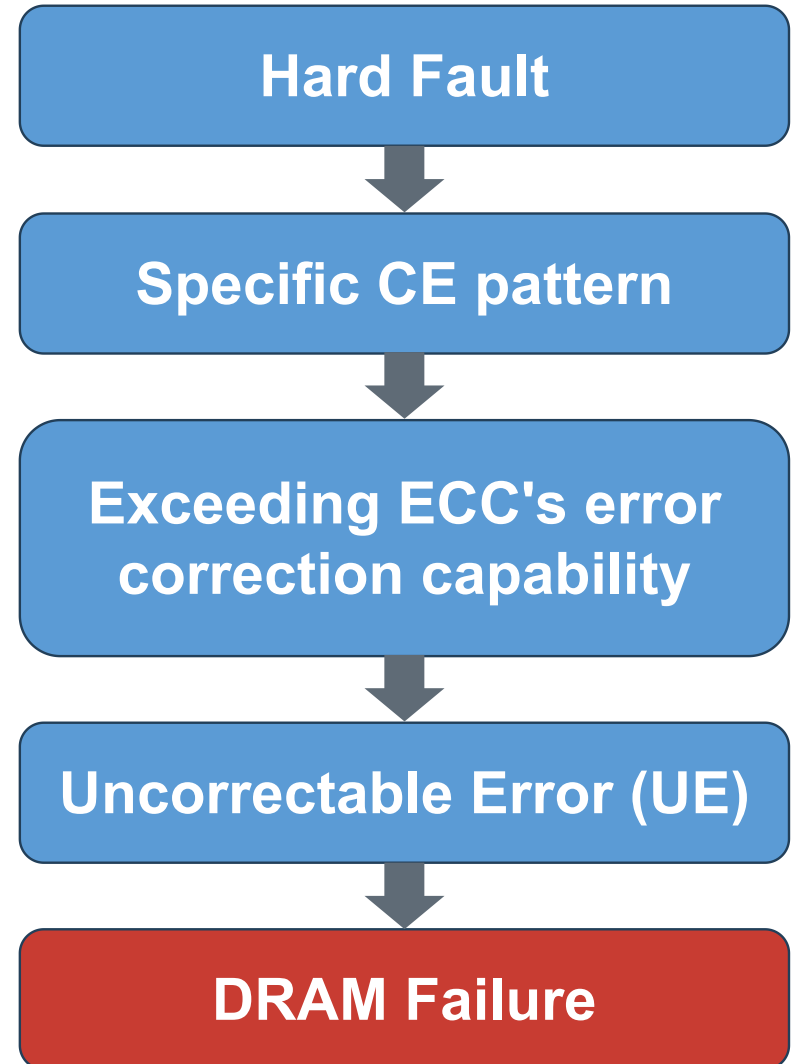
Contents

- **Introduction**
- **Preliminaries**
- **Methodology**
- **Evaluation**
- **Conclusion**

Introduction

Correctable Error (CE) and DRAM Failure

- Latent hardware defects develop into hard fault
- ECC corrects CE within its correction capability
- Specific CE pattern exceeding ECC capability becomes UE
 - Recurring CE
 - Bursty CE
 - CE storm
 - Spatially clustered CE
 -
- UE can cause DRAM Failure



Motivation: Preserving SpatioTemporal Structure

➤ Information in CE Logs

- Temporal evolution: recurring CEs, bursts, and CE storms
- Spatial locality across ranks, chips, banks, rows, and columns
- Bit-level patterns from DQ-Beat information
- Other static information such as manufacturer, frequency, etc.

➤ Limitations of Existing Representations

- Rely on manually engineered or flattened statistics, such as:
 - CE counts or rates within fixed time windows
 - Counts of affected cells, rows, columns, and banks
 - Multi-DQ, and Multi-DQ-Beat rules
 - Other statistical features
- Lose temporal dependencies and topology-induced relations between CE events

Need: Preserve temporal evolution, spatial locality, and bit-level patterns

Related Work

➤ Empirical DRAM Studies [4-6, 13-15]:

- *Finding:* Field studies reveal spatial clustering, temporal concentration, and risky DQ-Beat patterns.

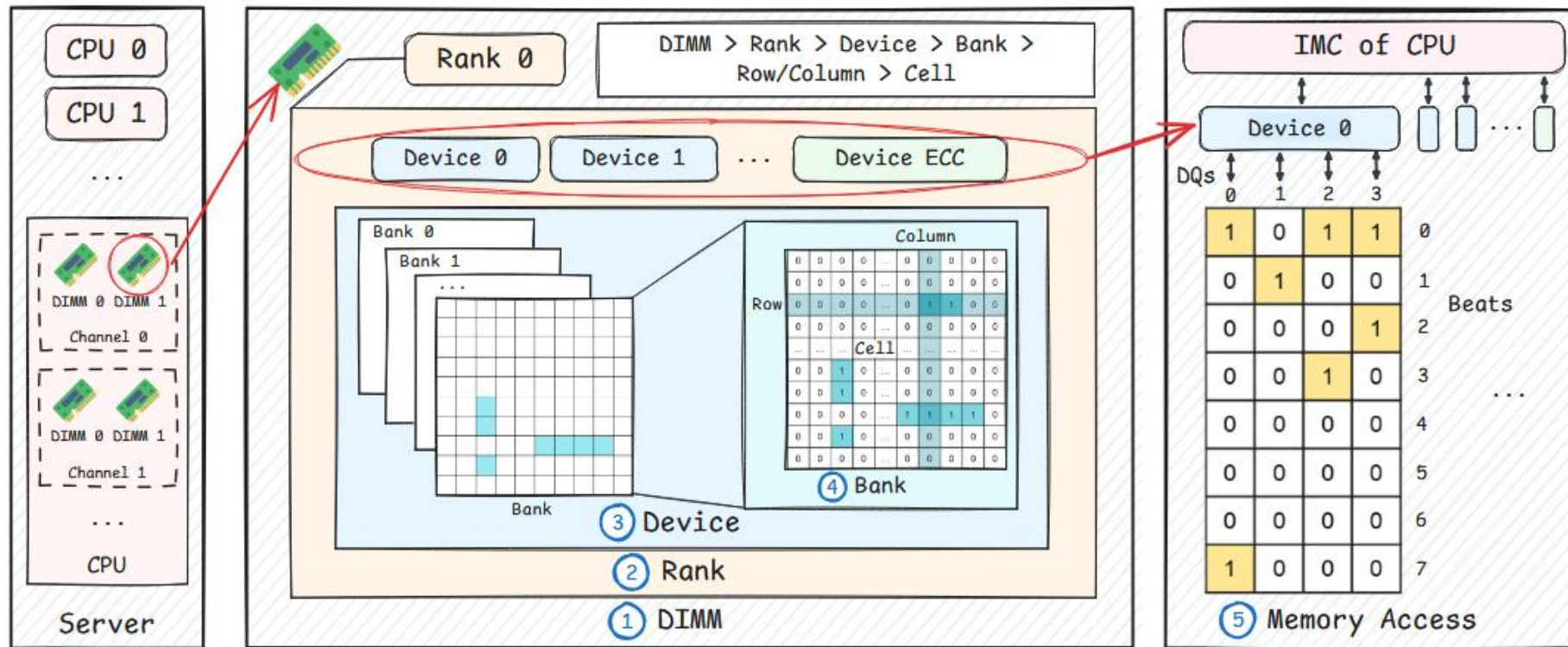
➤ ML-based Failure Prediction Models [8, 16, 17]:

- *Advancement:* Tree-based models naturally handle heterogeneous tabular features and learn nonlinear rules from feature combinations.
 - *Limitation:* Rely on manually engineered and flattened features, losing spatial and temporal relations among CE events.
- ## ➤ Sequential Failure Prediction Models [7, 17]:
- *Advancement:* Capture evolutionary trends within time sequence.
 - *Limitation:* RNN/LSTM can be overly sensitive to specific time delays, and existing sequential models still ignore spatial relations.

Method	Representation	Temporal	Spatio	bit-level patterns
ML-based Models	Flattened heterogeneous tabular features	Window-level statistics	Not explicit	Only statistical feature
Sequential Models	One-dimensional feature sequences	Sequence modeling	Not explicit	Not explicit
This Work	Spatiotemporal CE-event graph	Temporal edges	Spatio edges	DQ-Beat matrix encoding

Preliminaries

Memory System Structure



Source: Xie et al., "M2-MFP," KDD 2025, Fig. 2.

- **Server level:** CPUs access multiple DIMMs through memory channels.
- **DIMM level:** A DIMM is hierarchically organized into ranks, devices (chips), banks, rows, columns, and cells.
- **Error information:** CE logs record the timestamp, physical location, and affected DQ–Beat positions.

Problem Formulation

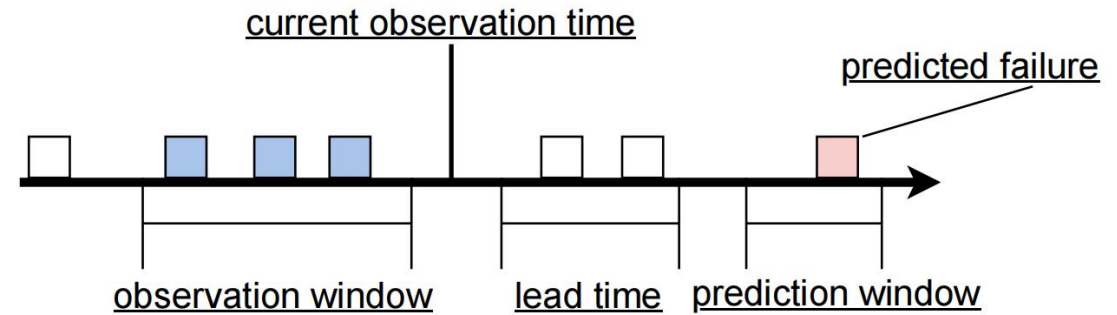
- **Input – observed CE events:** For a target DIMM d at reference time t , we collect a set of CE events within the observation window:

$$E_d(t) = \{e_i = (t_i, L_i, B_i) \mid t_i \in [t - \Delta t_{\text{obs}}, t]\}_{i=1}^N$$

Each event e_i contains its timestamp t_i , physical location L_i , and DQ–Beat pattern B_i .

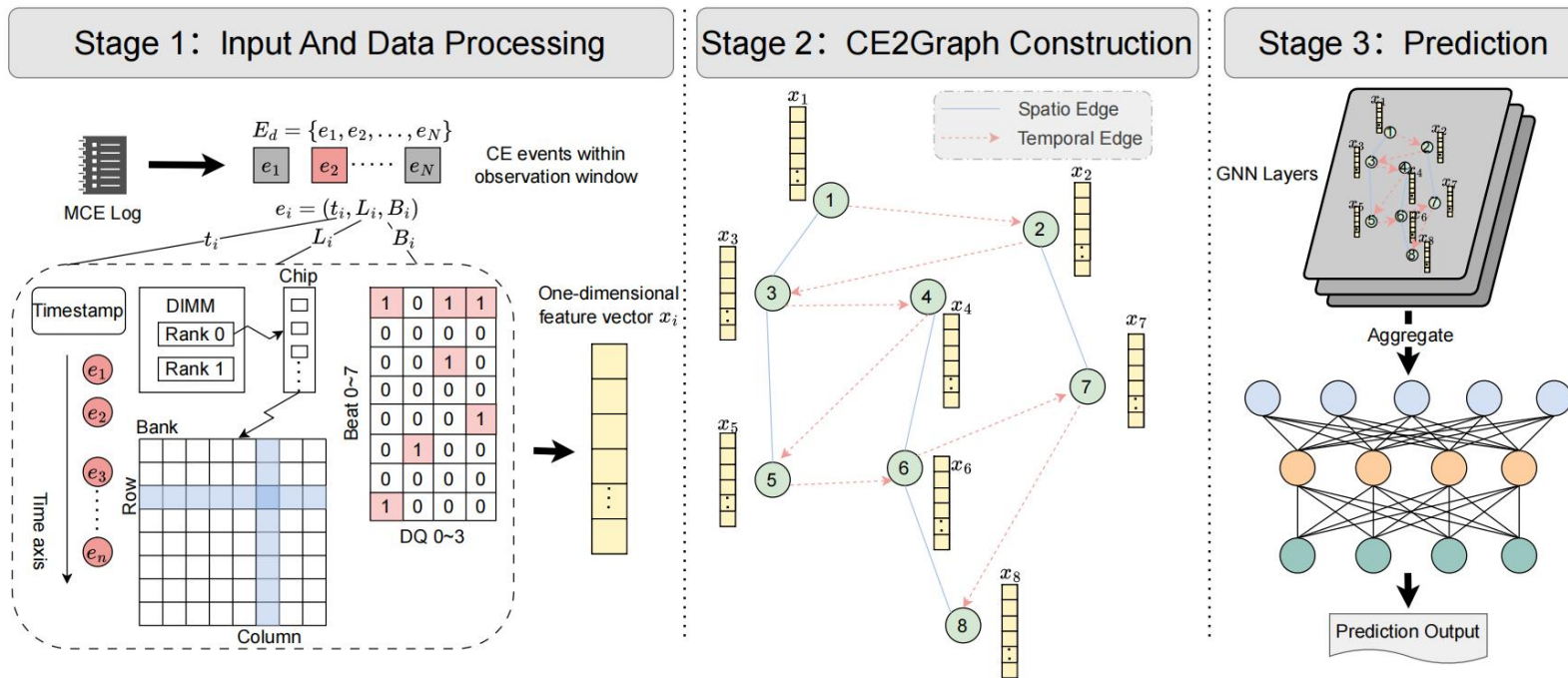
- **Target – future DRAM failure label:** A sample is labeled positive if DIMM d is predicted failed within the future prediction window:

$$y_d(t) = \begin{cases} 1, & \text{if DIMM } d \text{ fails within} \\ & [t + \Delta t_{\text{lead}}, t + \Delta t_{\text{lead}} + \Delta t_{\text{pred}}] \\ 0, & \text{otherwise.} \end{cases}$$



Methodology

CE2Graph Framework



- **Stage 1 — Encode CE events as nodes**

Map each CE event to one node using its DQ–Beat pattern and physical-location encoding.

- **Stage 2 — Construct the spatio-temporal graph**

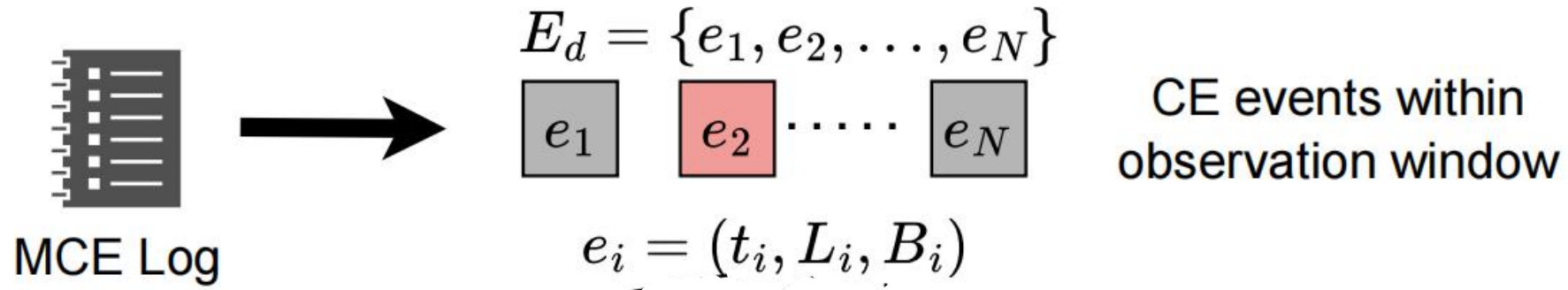
Add spatial edges for shared DRAM hierarchical structures and directed temporal edges for nearby events.

- **Stage 3 — Predict DRAM failure**

GG-NN message passing and graph readout estimate future failure probability.

Key idea: One CE becomes one node, physical and temporal relations become edges.

Node Representation



- **One-to-one mapping**

Each CE event e_i in the observation window corresponds to graph node v_i .

- **Three event attributes**

$e_i = (t_i, L_i, B_i)$: timestamp t_i , physical location L_i , and DQ–Beat error pattern B_i .

Node Representation

Micro-level DQ–Beat Matrix Representation

- **Micro-level binary encoding**

$B_i \in \{0, 1\}^{8 \times 4}$, with eight rows for burst beats and four columns for DQ lanes.

- **Position-specific error pattern**

$B_i(r, c) = 1$ indicates an error at corresponding Beat (r) and DQ (c).

- **Failure-prediction relevance**

Patterns spanning multiple DQs or Beats are more likely to evolve into a UE.

Beat 0~7

1	0	1	1
0	0	0	0
0	0	1	0
0	0	0	0
0	0	0	1
0	1	0	0
0	0	0	0
1	0	0	0

DQ 0~3

Node Representation

Node Feature Formulation

- **Bit-level feature encoding**

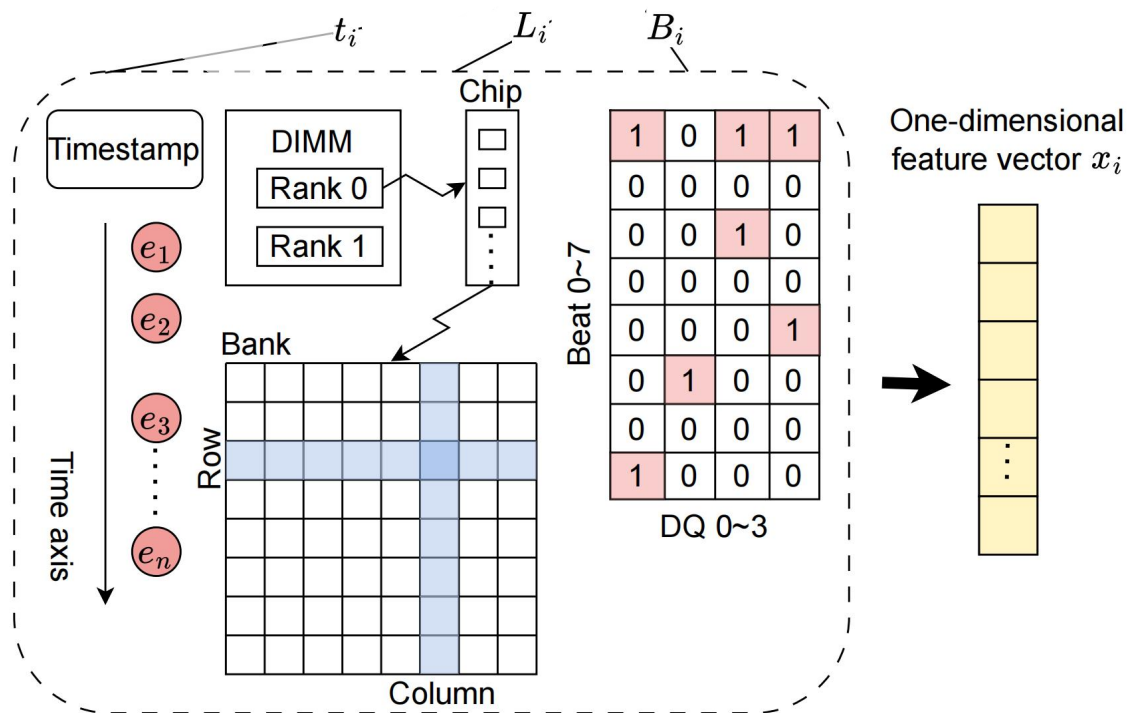
The 8×4 DQ–Beat matrix is flattened into a 32-dimensional binary vector $vec(B_i) \in \{0, 1\}^{32}$.

- **Physical-location feature encoding**

The location tuple $L_i = (Rank_i, Chip_i, Bank_i, Row_i, Col_i)$ is passed through a learnable MLP to obtain the location representation $MLP_\theta(L_i)$.

- **Joint feature representation**

The features are concatenated into a one-dimensional initial node feature vector $x_i = vec(B_i) \oplus MLP_\theta(L_i)$, then the timestamp is added to the vector.



Spatio-Temporal Graph Construction

- **CE events as graph nodes**

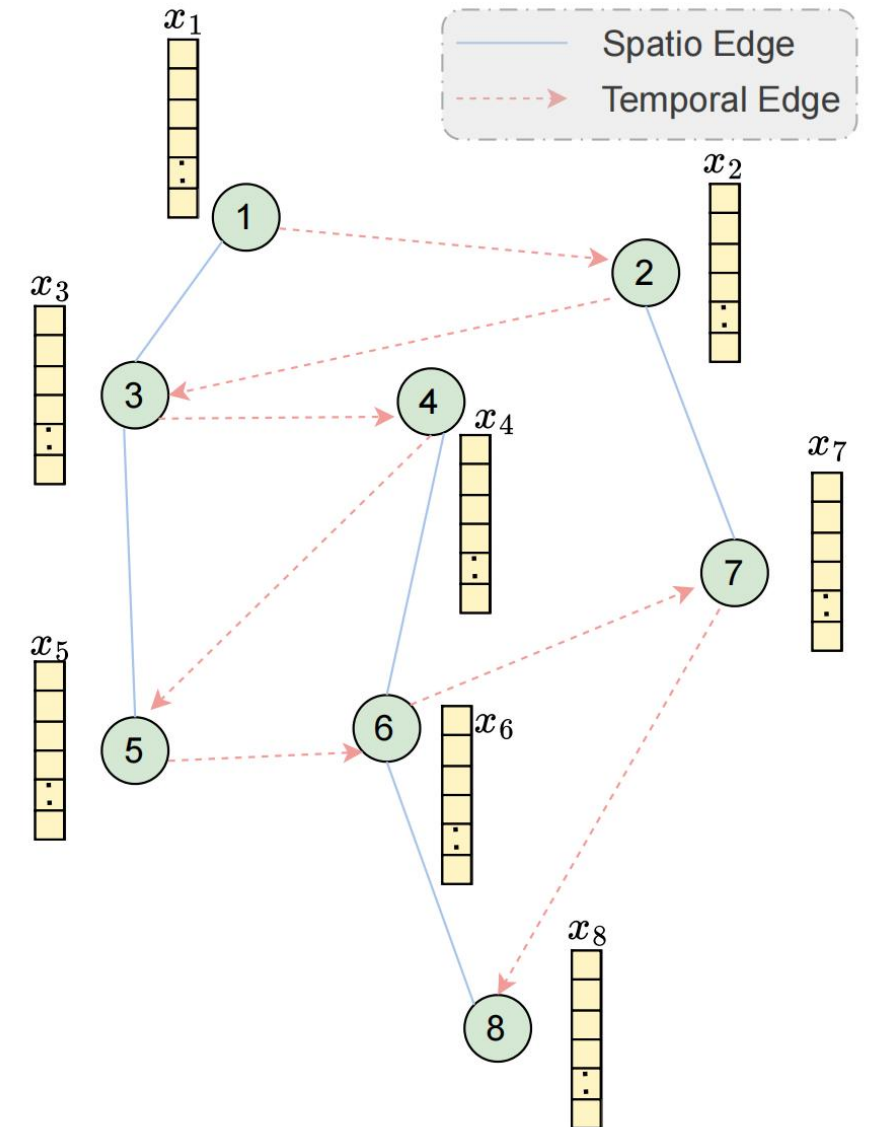
Each CE event is represented as a node v_i , initialized with its encoded feature vector x_i .

- **Spatial and Temporal relations as graph edges**

Spatial edges capture physical locality, while directed temporal edges capture short-term evolution.

- **Unified weighted graph**

The two edge types jointly define a weighted adjacency matrix $A_{i,j} = A_{i,j}^{\text{spat}} + A_{i,j}^{\text{temp}}$, measures the relation strength between nodes v_i and v_j . The resulting graph is represented as $G = (V, E, A)$.



Spatio-Temporal Graph Construction

➤ Spatial Edges

● Connection rule

Connect v_i and v_j when they belong to the same DRAM hierarchy.

● Purpose

Capture physical locality and fault propagation.

$$A_{i,j}^{\text{spat}} = I(\text{Bank}_i = \text{Bank}_j) [w_{\text{row}} I(\text{Row}_i = \text{Row}_j) + w_{\text{col}} I(\text{Col}_i = \text{Col}_j)]$$

➤ Temporal Edges

● Connection rule

Add a directed edge $v_i \rightarrow v_j$ when e_j occurs later and $(0 < t_j - t_i \leq \tau_{\text{storm}})$, where $\tau_{\text{storm}} = 60$ s.

Temporally closer events receive larger weights, controlled by a decay parameter λ .

● Purpose

Capture short-term error concentration and temporal fault escalation.

$$A_{i,j}^{\text{temp}} = \begin{cases} \exp(-\lambda(t_j - t_i)), & \text{if } 0 < t_j - t_i \\ & \leq \tau_{\text{storm}}, \\ 0, & \text{otherwise,} \end{cases}$$

GG-NN-Based Spatio-Temporal Failure Prediction

- **Max-based message aggregation**

Each node takes the element-wise maximum over adjacency-weighted neighbor messages, emphasizing severe local fault evidence.

- **GRU-based state update**

Update and reset gates combine the aggregated message with the previous node state over multiple GG-NN layers.

- **Graph-level failure prediction**

Global max pooling produces a DIMM-level representation, followed by an MLP and sigmoid to estimate the future failure probability.

- **Imbalance-aware training**

Weighted binary cross-entropy addresses the rarity of positive failure samples.

Algorithm structure

Initialization (Lines 1–3) → Message passing and GRU update (Lines 4–17)

Graph readout and prediction (Lines 18–21) → Weighted loss (Lines 22–23)

Algorithm 1: GG-NN-based spatio-temporal learning on CE2Graph

Input: Spatio-temporal graph $G_d(t) = (V, E, A)$; node features $\{x_i\}_{v_i \in V}$; number of layers L ; Model parameters $\Theta = \{W^{(l)}, W_z, U_z, W_r, U_r, W_h, U_h, \phi\}$

Output: Failure probability $\hat{p}_d(t)$ and loss $\mathcal{L}_d$

```
1 foreach  $v_i \in V$  do
2   Initialize node hidden state:  $h_i^{(0)} \leftarrow x_i$ 
3 end
4 for  $l \leftarrow 1$  to  $L$  do
5   foreach  $v_i \in V$  do
6     Aggregate neighborhood messages:
7      $a_i^{(l)} \leftarrow \max_{j \in \mathcal{N}(i)} (A_{j,i} W^{(l)} h_j^{(l-1)})$ 
8     Update gate:
9      $z_i^{(l)} \leftarrow \sigma(W_z a_i^{(l)} + U_z h_i^{(l-1)})$ 
10    Reset gate:
11     $r_i^{(l)} \leftarrow \sigma(W_r a_i^{(l)} + U_r h_i^{(l-1)})$ 
12    Candidate hidden state:
13     $\tilde{h}_i^{(l)} \leftarrow \tanh(W_h a_i^{(l)} + U_h (r_i^{(l)} \odot h_i^{(l-1)}))$ 
14    Node state update:
15     $h_i^{(l)} \leftarrow (1 - z_i^{(l)}) \odot h_i^{(l-1)} + z_i^{(l)} \odot \tilde{h}_i^{(l)}$ 
16  end
17 end
18 Graph readout by global max pooling:
19  $h_d^G \leftarrow \max_{v_i \in V} h_i^{(L)}$ 
20 Failure probability prediction:
21  $\hat{p}_d(t) \leftarrow \sigma(\text{MLP}_\phi(h_d^G))$ 
22 Weighted binary cross-entropy loss:
23  $\mathcal{L}_d \leftarrow$ 
24    $-[w_{\text{pos}} y_d(t) \log \hat{p}_d(t) + w_{\text{neg}} (1 - y_d(t)) \log(1 - \hat{p}_d(t))]$ 
25 return  $\hat{p}_d(t), \mathcal{L}$ 
```

Evaluation

Experimental Setup

➤ Dataset

- Contains continuous CE logs, static configurations, and failure tickets documenting actual DRAM-related failures.
- Covers 72,641 DIMM slots.

➤ Split Strategy & Parameters

- Chronological split: 86,904 training, 20,465 validation, and 20,531 test samples.
- 7-day observation window and 30-day prediction window

➤ Baselines & Metrics

- 14 baselines: 4 tree-based ML, 2 deep-learning, and 8 rule-based or handcrafted predictors.
- Evaluation metrics: Recall, Precision, and F1-score.

Overall Performance Comparison

● Main Results

- Best F1-score among all compared methods: 0.4136.
- Highest precision among all compared methods: 0.5940.
- CE2Graph provides the best overall precision-recall balance.

● Relative F1 Improvement

- 40.1% higher than XGBoost, the strongest tree-based baseline.
- 29.3% higher than Multi-DQ, the strongest rule-based baseline.

TABLE I Performance comparison.

Method	Recall	Precision	F1-score
Tree-based models			
Random Forest	0.3473	0.2131	0.2641
XGBoost	0.2934	0.2973	0.2953
CatBoost	0.3054	0.2566	0.2789
LightGBM	0.5030	0.1742	0.2588
Deep learning models			
LSTM	0.4491	0.0806	0.1366
FT-Transformer	0.5210	0.0452	0.0832
Rule-based and handcrafted-feature predictors			
Cell fault predictor	0.69	0.04	0.08
Row fault predictor	0.43	0.06	0.11
Column fault predictor	0.26	0.08	0.13
Bank fault predictor	0.30	0.12	0.17
CE overflow predictor	0.15	0.38	0.21
CE storm predictor	0.61	0.08	0.13
Multi-DQ predictor	0.77	0.20	0.32
Multi-DQ-Beat predictor	0.79	0.14	0.24
CE2Graph (Ours)	0.3173	0.5940	0.4136

Ablation Study

● Component Contributions

- Full CE2Graph achieves the best F1-score of 0.4136.
- Removing spatial relations reduces the F1-score by 14.3%, to 0.3545.
- Removing temporal relations reduces the F1-score by 6.6%, to 0.3862.
- Removing DQ-Beat features reduces the F1-score by 8.0%, to 0.3807.

TABLE II Ablation study of CE2Graph.

Variant	Recall	Precision	F1-score
Only Temporal	0.2365	0.7074	0.3545
Temporal + Bank	0.3078	0.6102	0.4092
Temporal + Row	0.2797	0.6507	0.3912
Temporal + Column	0.2578	0.7364	0.3819
Temporal + Cell	0.2722	0.6525	0.3841
Temporal + Bank + Row	0.2457	0.6332	0.3540
Only Spatio	0.2584	0.7640	0.3862
w/o DQ-Beat	0.2656	0.6716	0.3807
CE2Graph	0.3173	0.5940	0.4136

Conclusion

Conclusion

➤ **Structured Spatio-Temporal Graph**

- CE2Graph transforms irregular CE logs into structured graphs that preserve DQ-Beat morphology, physical topology, and temporal relations.

➤ **Hardware-Aware Failure Prediction**

- A max-based GGNN aggregates localized fault evidence, bridging low-level hardware symptoms and DIMM-level DRAM failure prediction.

➤ **Future Work**

- Evaluate robustness across hardware platforms and integrate prediction with online mitigation strategies.

THANK YOU!