CNN-Based RowHammer Vulnerability Prediction: Inter-DIMM Generalization and Intra-DIMM Completion from Partial Row Observations

Vineet Suresh Kumar Mohammad Farmani

Florida Polytechnic University

RowHammer vulnerability characterization requires exhaustive profiling of the DRAM array, which is time-consuming for modern high-density modules, often requiring several hours of hammering per DIMM. This paper evaluates the use of row-wise Convolutional Neural Networks (CNNs) to predict RowHammer susceptibility from partial spatial observations. We treat the distribution of bit flips as a spatial signal and employ a multi-scale convolutional architecture to identify vulnerability patterns across different manufacturers. Using linear interpolation to manage missing data, we perform window-level predictions of vulnerability severity, i.e., predicting how severe bit flips will be in unmeasured row ranges of a DIMM. Experimental results on 19 DDR4 modules from Samsung, Hynix, and Micron demonstrate that the model achieves 97.0 percent classification accuracy on a representative held-out test set, with a population-wide mean accuracy of 95.6 percent plus or minus 6.1 percent. A masked prediction strategy allows us to infer window-level vulnerability severity from as little as 10 percent of the rows, reducing profiling time 10-fold while preserving high predictive fidelity. These findings demonstrate that spatial autocorrelation in bit-flip distributions can be exploited to accelerate DRAM security testing for memory vendors and system integrators.

1. Introduction

Dynamic Random Access Memory (DRAM) scaling reduces the RowHammer threshold (HC_{min}), the minimum number of activations required to induce bit flips in physically adjacent rows [1, 2]. These disturbance errors result from electromagnetic interference causing charge leakage in victim rows. As DRAM technology nodes shrink, the decreasing HC_{min} increases the vulnerability of modern memory modules, necessitating robust characterization and mitigation strategies [2].

For memory vendors and system integrators, characterizing RowHammer vulnerability is critical for quality assurance and production screening. However, identifying vulnerabilities requires exhaustive profiling of the memory array, which involves hammering each row under various data patterns and environmental conditions. Profiling is extremely time-consuming for high-capacity memory; a full profile of an 8 gigabyte DIMM can take several hours, where it takes approximately 38 minutes per GB at $HC = 6.8 \times 10^5$ and standard JEDEC DDR4-2400T timing based on our measurements. Such overhead makes frequent auditing in cloud environments or comprehensive per-device screening during manufacturing impractical.

This paper evaluates the use of row-wise Convolutional Neural Networks (CNNs) to accelerate RowHammer vulnerability characterization by predicting susceptibility from partial spatial observations. We treat bit-flip distributions as spatial signals, allowing the model to identify vulnerability patterns in unobserved memory regions based on measurements from neighboring rows. Beyond reducing profiling time, this approach enables "tier-aware" defensive strategies: memory controllers can use predicted vulnerability maps to prioritize refresh resources for Severe and Moderate regions, reducing the performance overhead of mitigations like Target Row Refresh (TRR) while maintaining strong security guarantees.

While primarily intended for defensive auditing and QA, this capability also provides a mechanism for attackers to identify high-risk regions with reduced profiling effort. We evaluate a row-wise CNN architecture that captures spatial periodicities in bit-flip patterns and predicts severity tiers using a dynamic percentile-based thresholding method. We test the generalization of these predictions across DIMMs from different vendors and evaluate an intra-DIMM completion task to predict the vulnerability of unobserved rows from partial measurements.

1.1. Threat Model

An attacker aims to identify RowHammer-vulnerable memory regions with reduced profiling time. This identification allows the attacker to find high-severity regions for exploitation without exhaustive characterization. The attacker requires only a single vulnerable region to compromise the system, which provides an asymmetric advantage over defensive implementations that must protect all vulnerable rows.

The attacker executes a double-sided RowHammer sequence with a hammer count (HC) of 6.8×10^5 activations per aggressor row, corresponding to a 64 milliseconds refresh interval (t_{REFW}). The attacker uses the row-wise CNN to predict the vulnerability of unobserved rows based on measurements from a subset of the DIMM.

In this work, we disable host-side ECC to observe raw bit flips at the DRAM array level. On-die ECC introduced in DDR5 is outside our DDR4 scope and discussed in Section 6.3. We accounted for Target Row Refresh (TRR) mechanisms by performing characterization within the 64 milliseconds refresh interval. This approach identifies vulnerabilities that persist despite internal DRAM mitigations [3, 4].

2. Background and Related Work

RowHammer is a disturbance phenomenon where rapid activation of an aggressor row induces bit flips in adjacent victim

rows [1]. As DRAM scales, reduced cell spacing increases electromagnetic coupling and intensifies vulnerability [2, 3]. DRAM defects are non-uniformly distributed due to manufacturing variations. Recent techniques like HiFi-DRAM [5] highlight the importance of physical layout. DRAM-Profiler [6] uses four-level security classifications, and SVARD [7] documents subarray-level periodicities. While machine learning is primarily applied to RowHammer for attack detection or fingerprinting (e.g., FP-Rowhammer [8]), our work applies CNNs directly to the spatial topology of bit flips to reconstruct spatial vulnerability profiles from partial observations, accelerating characterization.

3. Experimental Setup and Dataset

We used the DRAM Bender framework [9] on a Xilinx Alveo U200 Field Programmable Gate Array (FPGA) to perform double-sided attacks on 19 Registered Dual In-line Memory Modules (RDIMMs) from Samsung, Hynix, and Micron. We profiled 16 banks across three data patterns (RowStripe, Checkered, ColumnStripe), yielding 912 unique maps partitioned into 28,272 windows of 2048 rows. Experiments used standard 64 milliseconds auto-refresh.

To prevent leakage, we employ strict partitioning. For inter-DIMM generalization, we train on 15 modules and evaluate on 4 representative held-out modules. Additionally, we perform Leave-One-DIMM-Out (LODO) cross-validation, where each fold holds out one DIMM for testing and trains on the remaining 18, and we report the mean accuracy across all 19 folds (mean accuracy = 0.9580 ± 0.032). For intra-DIMM completion, we partition each bank into disjoint row-ranges: rows 0–8192 for training and the remainder for validation.

4. Methodology

The methodology uses a row-wise CNN which identifies RowHammer vulnerability patterns from spatial bit-flip distributions.

4.1. Data Representation and Preprocessing

The input data consists of bit-flip counts recorded during RowHammer profiling. Each DRAM bank is represented as a density map where each element is the total number of bit flips in a specific row. We handle the dynamic range of bit-flip counts by applying a log-transformation: $x' = \ln(x + 1)$, where x represents the raw bit-flip count and x' denotes the log-transformed density.

The memory array is partitioned into windows of 2048 rows. We employ a masked prediction strategy where a subset of rows within each window is sampled according to a probe budget (10 percent, 25 percent, or 50 percent). The input is a two-channel tensor containing linearly interpolated bit-flip densities and a binary mask indicating measured rows. The model objective is a window-level prediction of the log-sum of bit flips in unobserved rows (Fig. 1).

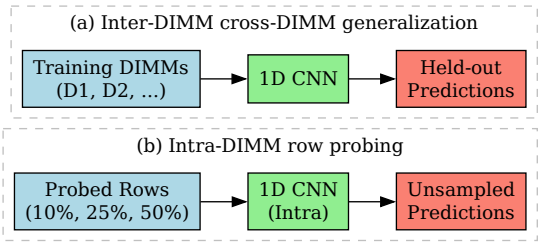

Figure 1: Overview of the proposed RowHammer vulnerability prediction framework. The workflow maps partial spatial observations to full-window severity profiles.

4.2. Row-wise Convolutional Neural Network Architecture

The row-wise CNN uses a multi-scale discovery layer which captures spatial features across different resolutions. This layer consists of four parallel convolutional blocks, each with 16 output channels and kernel sizes of 128, 256, 512, and 1024 elements. The outputs of these parallel blocks are concatenated to form a 64-channel feature map. This architecture identifies periodicities and clustering patterns that span different row-addressing granularities, as visualized in Fig. 2 and Fig. 3.

The multi-scale layer is followed by Batch Normalization, a Rectified Linear Unit (ReLU) activation function, and a Max Pooling layer with a kernel size of two. Two subsequent convolutional layers use 128 and 256 channels with kernel sizes of five and three elements, respectively. A Global Average Pooling layer reduces the final feature map to a 256-element feature vector.

The model employs multi-task learning with two output heads. A regression head predicts the log-normalized bit-flip count for the unobserved rows. A classification head predicts the severity tier across four classes: Severe, Moderate, Low, and Safe.

4.3. Severity Classification and Training

Bit-flip counts vary across DRAM vendors and manufacturing nodes. To align with the four-level security classifications used in prior work like DRAM-Profiler [6] and to provide balanced boundaries among vulnerable tiers, we use a dynamic percentile-based thresholding mechanism. We compute the

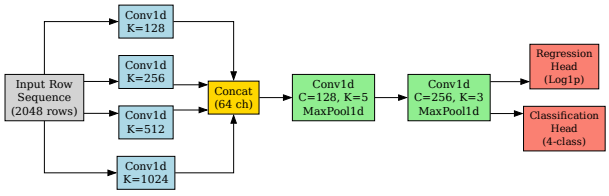

Figure 2: The multi-scale row-wise CNN architecture, featuring four parallel discovery kernels (16 channels each) and dual regression/classification output heads.

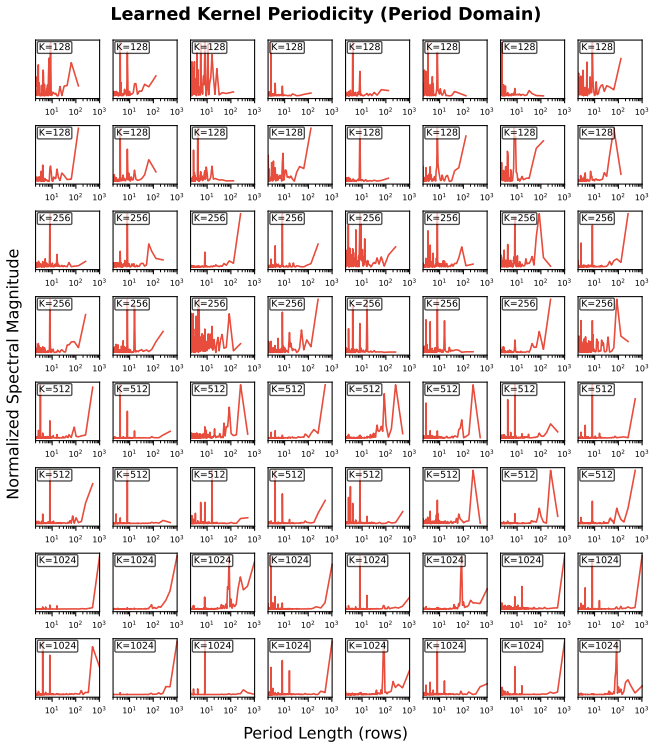

Figure 3: Frequency domain representation of learned kernels showing structural periodicity detection (X-axis represents period length in rows).

25th and 75th percentiles of the non-zero bit-flip counts in the training set to define the boundaries between Low, Moderate, and Severe classes. These thresholds are computed once globally on the training set and applied statically to all held-out DIMMs during evaluation, ensuring that the model generalizes to the underlying vulnerability distribution rather than adapting to per-device statistics. Regions with zero bit flips are categorized as Safe.

The model is trained using a joint loss function combining Mean Squared Error (MSE) for regression (weighted by 0.1) and Cross-Entropy (CE) for classification. We use the Adam optimizer (learning rate 0.001, batch size 256). Training was distributed across two AMD Radeon RX 7900 XTX GPUs via ROCm, consuming 4 gigabytes VRAM each.

5. Results

5.1. Inter-DIMM Generalization Performance

On our validation set, the model achieves 97.0 percent accuracy. Table 1 breaks down precision, recall, and F1-score for inter-DIMM and intra-DIMM tasks, with 95 percent confidence intervals from bootstrapping ($N = 1000$). The model achieves 100 percent recall for vulnerable tiers in the inter-DIMM task and for the Severe tier in the intra-DIMM task, ensuring high-risk regions are identified. Lower precision for some tiers reflects conservative behavior, preferable for security auditing where false negatives are more costly. Leave-one-DIMM-out (LODO) validation yields a mean accuracy of $95.6\% \pm 6.1\%$, macro F1-score of 0.851 ± 0.212, recall of 0.874 ± 0.190, regression R^2 of 0.729 ± 0.282, and MAE (log-scale) of 2.001 ± 1.129. While legacy modules (e.g., *hynix02*) show lower Moderate recall (47 percent), recent nodes maintain near-perfect performance. This reduced recall is due to threshold misalignment: the older process node for *hynix02* produces bit-flip distributions that differ from the training population, so the globally computed Moderate tier boundary misclassifies borderline windows.

The scatter plot in Fig. 5 further demonstrates a strong correlation between predicted and actual log-flip counts ($R^2 = 0.84$). This performance shows that the multi-scale kernels successfully capture cross-DIMM spatial features.

To contextualize this performance, Table 2 compares the row-wise CNN against naive baselines. The CNN significantly outperforms both a global mean baseline and a vendor-specific mean baseline, demonstrating its capacity to learn meaningful spatial features rather than merely memorizing averages.

However, analyzing the Mean Absolute Error (MAE) reveals significant vendor-specific variance in absolute flip counts. Samsung exhibits the highest error (raw MAE: 169,967; log MAE: 1.76), followed by Hynix (raw MAE: 28,904; log MAE: 1.57) and Micron (raw MAE: 21,187; log MAE: 1.45). This discrepancy highlights that while the model successfully learns the spatial distribution topology required for accurate tier classification across vendors, predicting exact raw bit-flip counts remains challenging due to manufacturing-specific vulnerability baselines.

5.2. Intra-DIMM Masked Completion

Table 3 provides a granular per-DIMM breakdown of performance for the masked completion task. We compare the Mean Absolute Error (log-scale) and Classification Accuracy of Next-Window (NW) prediction against masked completion at varying probe budgets for both the linear interpolation baseline (I) and the row-wise CNN (C).

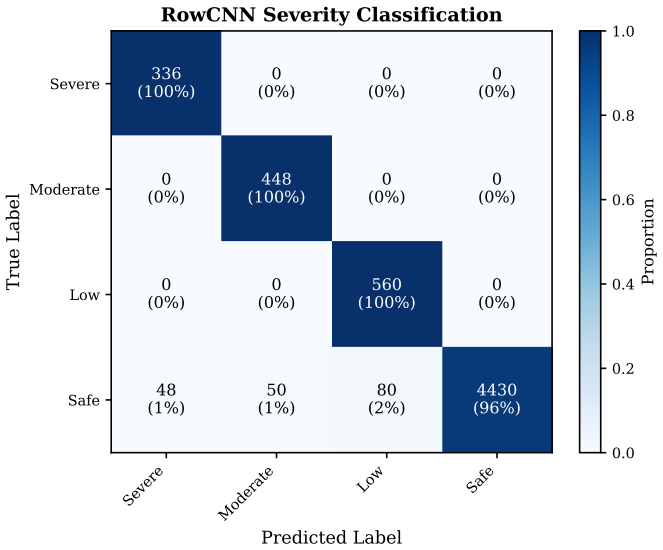

Figure 4: Confusion matrix for inter-DIMM severity classification on representative held-out modules.

Table 1: Severity Classification Performance with 95 percent Confidence Intervals (Bootstrap, $N = 1000$)

	Inter-DIMM (Next Window)			Intra-DIMM (Next Window)		
Category	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Severe	0.875 [0.84, 0.91]	1.000 [1.00, 1.00]	0.933 [0.91, 0.95]	0.871 [0.85, 0.89]	1.000 [1.00, 1.00]	0.931 [0.92, 0.94]
Moderate	0.899 [0.87, 0.93]	1.000 [1.00, 1.00]	0.947 [0.93, 0.96]	0.914 [0.90, 0.93]	0.928 [0.91, 0.94]	0.921 [0.91, 0.93]
Low	0.875 [0.85, 0.90]	1.000 [1.00, 1.00]	0.933 [0.92, 0.95]	1.000 [1.00, 1.00]	0.820 [0.79, 0.85]	0.901 [0.88, 0.92]
Safe	1.000 [1.00, 1.00]	0.961 [0.96, 0.97]	0.980 [0.98, 0.98]	N/A	N/A	N/A

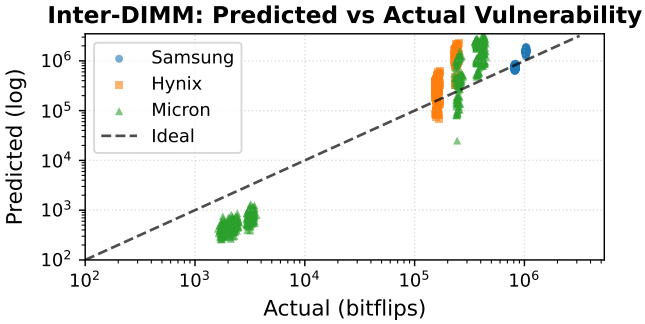

Figure 5: Scatter plot of predicted versus actual log-normalized bit-flip counts for the inter-DIMM generalization task. Color indicates vendor.

Table 2: Inter-DIMM Regression Baselines vs Row-wise CNN

Model	**MAE (Log)**	**R^2 Score**
Naive Mean Baseline	4.4071	-0.0173
Vendor Mean Baseline	4.3675	-0.0097
Row-wise CNN	**1.5634**	**0.8395**

5.3. Intra-DIMM Performance Tradeoffs

Table 3 shows a tradeoff between interpolation and CNN modeling. Interpolation achieves high R^2 values by extrapolating magnitude trends, but it misclassifies severity tiers at low budgets. The CNN produces more accurate and consistent classifications. On *hynix01* at 10 percent budget, the CNN reaches 100.0 percent accuracy while interpolation achieves only 70.8 percent. This 29.2 percentage point gap persists at 50 percent, where interpolation collapses to 0.0 percent while the CNN maintains 100.0 percent accuracy. The CNN captures spatial vulnerability patterns that linear interpolation overlooks. This explains why the CNN classifies severity tiers correctly while interpolation only tracks magnitude. On *samsung05*, the CNN achieves 81.9 percent accuracy at 10 percent budget while interpolation falls to 60.4 percent. Across all 19 DIMMs, the CNN consistently outperforms interpolation in classification accuracy, with the largest margins observed on Samsung and Hynix modules where vulnerability distributions exhibit more complex spatial structure.

The presence of 0.000 or 1.000 accuracy entries in Table 3 results from the interaction between threshold-based classification and highly localized spatial distributions. In cases with 0.000 accuracy (e.g., *hynix01* at 50 percent budget for interpolation), the linear model consistently under-estimates or over-estimates the flip count by a small margin that nonetheless crosses a class boundary for all evaluated windows. Conversely, 1.000 accuracy indicates that the model's magnitude estimations, even if not perfectly accurate in raw count, remain securely within the correct severity tier boundaries.

5.4. Zero-Shot and Cross-Manufacturer Evaluation

We evaluate vulnerability categorization on held-out devices without observed bit-flip data. We compare manufacturer-specific models, a unified general model, and a linear interpolation baseline using sparse random samples (10 or 25 percent). We use unified thresholds for all evaluations to ensure fair comparison. Table 4 presents these results.

Results support the hypothesis that specialist models match or exceed general model performance. For Samsung and Micron, specialists achieve 100 percent accuracy at 10 percent budget. Specialists benefit from learning vendor-specific periodicities consistent within product lines. For Samsung DIMMs at 10% probe budget, the CNN's log-scale MAE is much larger than interpolation (Table 3). This occurs because Samsung modules have higher absolute variance in their vulnerability distributions, which inflates MAE; nevertheless, the CNN still maintains competitive severity-tier accuracy, so the security-relevant conclusions remain valid.

Both CNN approaches vastly outperform linear interpolation for severity categorization. While interpolation achieves high R^2 (above 0.99), it fails to map magnitudes to tiers at low budgets, dropping to 0.745 accuracy. This demonstrates the CNN identifies underlying structural features of vulnerability rather than performing pixel-wise completion.

6. Discussion

6.1. Profiling Efficiency and the Probe-Budget Paradox

The proposed strategy enables up to a 10-fold speedup in data acquisition while maintaining an aggregate R^2 of 0.89. Including model inference (less than 10 milliseconds), characterization time is reduced by over 90 percent compared to an exhaustive scan. We observed a "probe-budget paradox" (Fig. 6) where relative prediction sensitivity increases at higher budgets as target variance decreases, suggesting predictive characterization is most effective at high sparsity (e.g., 10 percent) where spatial autocorrelation provides the strongest signal. We observe that prediction stability is highest at 10 percent probe budget and briefly dips at intermediate budgets, likely due to variance shifts in the targets; we leave deeper analysis to future work.

Table 3: Intra-DIMM Performance Comparison across 19 DIMMs: Interpolation Baseline (I) vs CNN (C)

DIMM	NW-MAE/Acc	I10-MAE/Acc	C10-MAE/Acc	I25-MAE/Acc	C25-MAE/Acc	I50-MAE/Acc	C50-MAE/Acc
hynix01	0.310/1.000	0.130/0.708	1.157/1.000	0.128/0.333	2.662/0.993	0.165/0.000	0.170/1.000
hynix02	0.218/1.000	0.109/0.361	1.004/0.576	0.138/0.333	2.713/0.375	0.175/0.000	0.151/1.000
hynix03	0.336/0.944	0.124/0.931	0.708/0.944	0.139/0.618	2.549/0.944	0.187/0.611	0.074/0.944
hynix04	0.268/0.507	0.120/0.333	0.903/0.410	0.138/0.243	2.636/0.333	0.183/0.000	0.115/0.806
hynix05	0.304/0.750	0.124/0.736	0.774/0.757	0.139/0.479	2.602/0.750	0.184/0.417	0.110/0.778
hynix06	0.370/1.000	0.125/0.931	0.601/0.993	0.142/0.674	2.524/0.993	0.182/0.667	0.076/1.000
micron02	0.449/1.000	0.137/1.000	0.411/1.000	0.147/1.000	0.185/1.000	0.179/1.000	0.103/1.000
micron03	0.371/1.000	0.150/1.000	0.182/1.000	0.146/1.000	0.351/1.000	0.173/1.000	0.169/1.000
micron04	0.399/1.000	0.146/1.000	0.320/1.000	0.149/1.000	0.222/1.000	0.170/1.000	0.115/1.000
micron11	0.338/1.000	0.170/0.958	1.422/1.000	0.198/0.799	1.845/1.000	0.230/0.333	0.074/1.000
micron12	0.325/1.000	0.157/0.993	1.403/1.000	0.184/0.868	1.899/1.000	0.235/0.306	0.070/1.000
micron13	0.318/1.000	0.158/0.965	1.378/1.000	0.180/0.785	1.911/0.993	0.237/0.236	0.080/1.000
micron14	0.293/1.000	0.158/0.938	1.274/1.000	0.192/0.667	1.907/1.000	0.249/0.014	0.099/1.000
samsung01	0.565/0.333	0.119/0.681	2.150/0.729	0.147/0.667	2.672/0.333	0.144/0.667	0.123/1.000
samsung02	0.624/0.576	0.102/0.646	2.272/0.694	0.127/0.438	2.535/0.562	0.117/0.438	0.180/0.639
samsung03	0.628/1.000	0.109/0.340	2.363/0.556	0.113/0.000	2.356/1.000	0.113/0.000	0.254/1.000
samsung04	0.629/0.764	0.107/0.438	2.258/0.500	0.124/0.250	2.561/0.750	0.120/0.250	0.180/0.764
samsung05	0.552/1.000	0.104/0.604	2.426/0.819	0.119/0.174	2.223/1.000	0.117/0.000	0.216/1.000
samsung06	0.576/1.000	0.101/0.778	2.433/0.931	0.116/0.333	2.173/1.000	0.109/0.000	0.275/1.000

Table 4: Zero-shot performance comparison between Manufacturer-Specific, Unified, and Interpolation models (Balanced Evaluation).

	10 percent Budget		25 percent Budget	
Model/Scenario	Acc.	Vuln Acc.	Acc.	Vuln Acc.
Linear Interpolation	0.883	0.843	0.745	0.660
Unified (Mixed)	0.918	0.891	**1.000**	**1.000**
Samsung Specialist	**1.000**	**1.000**	0.988	0.982
Hynix Specialist	0.810	0.711	**1.000**	**1.000**
Micron Specialist	**1.000**	**1.000**	**1.000**	**1.000**

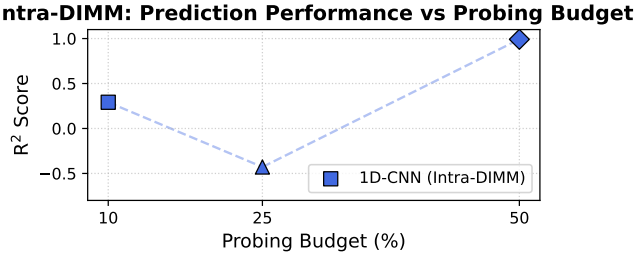

Figure 6: Aggregate R^2 for masked intra-DIMM prediction across all DIMMs.

6.2. Comparison with Spatial Baselines

To evaluate the necessity of the convolutional architecture for intra-DIMM tasks, we compared our model against two classical spatial baselines: linear interpolation and 1D Gaussian Process (GP) regression with a Matérn kernel ($\nu = 1.5$). As shown in Table 5, linear interpolation provides a surprisingly strong baseline, achieving $R^2 > 0.99$ across all budgets. This shows that RowHammer bit-flip distributions exhibit extremely high spatial autocorrelation within a single bank, such that even sparse measurements (10 percent) can accurately reconstruct the log-density profile of unobserved rows.

Linear interpolation is very strong within a DIMM when local probed context is available, achieving higher R^2 and higher accuracy than the CNN at low probe budgets for intra-DIMM completion. However, the CNN's primary advantage is (i) tier classification at higher probe budgets and (ii) cross-DIMM generalization (Table 2), where no local context exists and interpolation cannot be applied. Linear interpolation achieved high R^2 (>0.99) and strong classification accuracy. While linear interpolation outperforms the CNN in regression fidelity (R^2), the CNN model maintains strong classification accuracy (up to 95.0 percent), which is the primary metric for security auditing. This performance gap is likely due to the CNN's joint optimization for classification and regression, which biases the model toward identifying vulnerable regions rather than minimizing pixel-wise regression error in a known spatial context. However, the CNN's primary advantage remains its ability to generalize to completely unobserved DIMMs (the inter-DIMM task), a capability that classical interpolation and GP methods cannot provide as they require local probe data for every target module.

Table 5: Comparison of spatial prediction methods for the Intra-DIMM masked completion task across varying probe budgets (10 percent, 25 percent, 50 percent).

	10 percent Budget		25 percent Budget		50 percent Budget	
Method	R^2	Acc.	R^2	Acc.	R^2	Acc.
Linear Interp.	**0.994**	**0.858**	**0.994**	**0.888**	**0.993**	**0.902**
Gaussian Process	0.939	0.552	0.968	0.738	0.961	0.794
CNN (Ours)	0.889	0.635	0.941	0.950	0.872	0.836

6.3. Architectural Insights

The model's reliance on large kernels (up to 1024 elements) shows that RowHammer vulnerability is influenced by macroscopic physical structures, such as subarray boundaries. These layouts remain consistent across product generations, explaining the high inter-DIMM generalization. While spatial patterns may reflect vendor-specific TRR mitigations, they represent the effective system-level vulnerability critical for security analy-

Table 6: Ablation study of CNN architectural components for the intra-DIMM masked completion task (25 percent budget).

Configuration	R^2	Accuracy	F1-score
Full Model (Mask + Multi-task)	**0.941**	**0.950**	**0.951**
CNN, No Mask Channel	0.099	0.879	0.884
Single Task (Classification Only)	N/A	0.913	0.916
Single Task + No Mask	N/A	0.933	0.933

sis. Future work will extend this framework to DDR5/LPDDR5 and integrate temporal covariates to handle Variable Retention DRAM (VRD) effects.

6.4. Ablation Study

To quantify the impact of our architectural choices, we performed an ablation study on the intra-DIMM masked completion task (25 percent budget), shown in Table 6. The results demonstrate that the binary mask channel is essential for accurate regression; removing it caused R^2 to drop from 0.941 to 0.099. This shows that the model uses the mask to distinguish between observed ground truth and interpolated regions. Furthermore, the multi-task learning setup provides a clear benefit for classification accuracy, which dropped from 95.0 percent to 91.3 percent when the regression head was removed. This suggests that predicting the raw bit-flip magnitude provides an auxiliary signal that regularizes the classification tiers and helps the model learn more accurate spatial representations.

6.5. Limitations and Future Work

Several research directions remain open. First, our targets are window-level log-sums of flips rather than per-row labels. Extending the approach to per-row vulnerability prediction is an objective for future work. Second, this study used 19 Registered Dual In-line Memory Modules (RDIMMs) at ambient temperature and a fixed hammer count. Testing generalization across varied temperatures, refresh intervals (t_{REFW}), and hammer counts is required to verify model reliability in diverse environments. Finally, the emergence of on-die ECC in DDR5 and LPDDR5 will require updated models that can account for internal error correction and different subarray layouts.

6.6. Defensive Integration and Security Impact

The ability to predict RowHammer vulnerability has immediate implications for memory characterization. In cloud environments, the 10-fold speedup enabled by our CNN allows for more frequent and detailed DRAM auditing without significant downtime. Furthermore, these predictive maps can inform hardware-level mitigations. Rather than applying a uniform refresh policy, a memory controller could prioritize "Severe" and "Moderate" windows. This strategy concentrates defensive resources on the most vulnerable regions. This "tier-aware" mitigation strategy could reduce the performance overhead of RowHammer defenses like TRR, which often suffer from limited internal tracking table sizes. By focusing on predicted high-risk regions, future DRAM controllers can provide stronger security guarantees with minimal throughput impact.

7. Conclusion

This work shows predictive modeling reduces RowHammer profiling effort by up to 10-fold while preserving high accuracy in identifying vulnerable regions. Treating bit-flip distributions as spatial signals allows a CNN to generalize vulnerability patterns across manufacturers. The model achieved 97.0 percent inter-DIMM accuracy, and intra-DIMM R^2 values of 0.89 (10 percent masked completion) and 0.84 (next-window). Furthermore, the model categorizes severity tiers with 0.918 overall accuracy and 0.891 accuracy on vulnerable regions in zero-shot evaluations using 10 percent of the memory rows (Table 4, Unified model). These results show the framework generalizes to unobserved devices.

References

[1] O. Mutlu, "The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser," 2017. [Online]. Available: http://arxiv.org/abs/1703.00626

[2] O. Mutlu and J. S. Kim, "RowHammer: A Retrospective," pp. 115–118, 2019. [Online]. Available: http://arxiv.org/abs/1904.09724

[3] J. S. Kim, M. Patel, A. G. Yaglikci, H. Hassan, R. Azizi, L. Orosa, and O. Mutlu, "Revisiting rowhammer: An experimental analysis of modern dram devices and mitigation techniques," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, May 2020, p. 638–651. [Online]. Available: http://dx.doi.org/10.1109/ISCA45697.2020.00059

[4] H. Hassan, Y. C. Tugrul, J. S. Kim, V. van der Veen, K. Razavi, and O. Mutlu, "Uncovering in-dram rowhammer protection mechanisms:a new methodology, custom rowhammer patterns, and implications," in *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO '21. ACM, Oct. 2021, p. 1198–1213. [Online]. Available: http://dx.doi.org/10.1145/3466752.3480110

[5] M. Marazzi, T. Sachsenweger, F. Solt, P. Zeng, K. Takashi, M. Yarema, and K. Razavi, "HiFi-DRAM: Enabling High-fidelity DRAM Research by Uncovering Sense Amplifiers with IC Imaging," in *ISCA*, 2024.

[6] R. Zhou, J. T. Liu, N. Kochar, S. Ahmed, A. S. Rakin, and S. Angizi, "DRAM-Profiler: An experimental DRAM RowHammer vulnerability profiling mechanism," *arXiv preprint arXiv:2404.18396*, 2024.

[7] A. G. Yaglikci, Y. C. Tugrul, G. F. Oliveira, I. E. Yuksel, A. Olgun, H. Luo, and O. Mutlu, "Spatial variation-aware read disturbance defenses: Experimental analysis of real DRAM chips and implications on future solutions," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2024, pp. 560–577.

[8] J. Ba, J. Park, J. S. Kim, and O. Mutlu, "Fp-rowhammer: Fingerprinting rowhammer susceptibility for device identification," in *2023 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2023.

[9] A. Olgun, H. Hassan, A. G. Yaglikci, Y. C. Tugrul, L. Orosa, H. Luo, M. Patel, O. Ergin, and O. Mutlu, "Dram bender: An extensible and versatile fpga-based infrastructure to easily test state-of-the-art dram chips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 12, pp. 5098–5112, 2023.