# Understanding RowHammer Under Reduced Refresh Latency: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions


**Yahya Can Tuğrul**

A. Giray Yağlıkçı  İsmail Emir Yüksel

Ataberk Olgun  Oğuzhan Canpolat  Nisa Bostancı

Mohammad Sadrosadati  Oğuz Ergin  Onur Mutlu

SAFARI

ETH zürich

kasırga

TOBB ETÜ

University of Economics & Technology

# Executive Summary

**Problem:**

- **Read disturbance in DRAM (e.g. RowHammer) worsens** with technology node scaling
- Existing solutions perform **preventive refresh**, inducing **significant overheads**

**Motivation: Reducing preventive refresh latency to reduce its overheads.**

- No prior work studies **i) the effect of preventive refresh latency** on RowHammer or **ii) the implications of reducing preventive refresh latency** on existing solutions

**Goal:**

- To understand the **impact of preventive refresh latency on RowHammer**
- To leverage this understanding to **reduce the overheads of existing solutions**

**Experimental Characterization: 388 DDR4 DRAM chips** from **three major vendors**

- The latency of a vast majority of preventive refresh can **be significantly reduced** without jeopardizing the data integrity of a DRAM chip

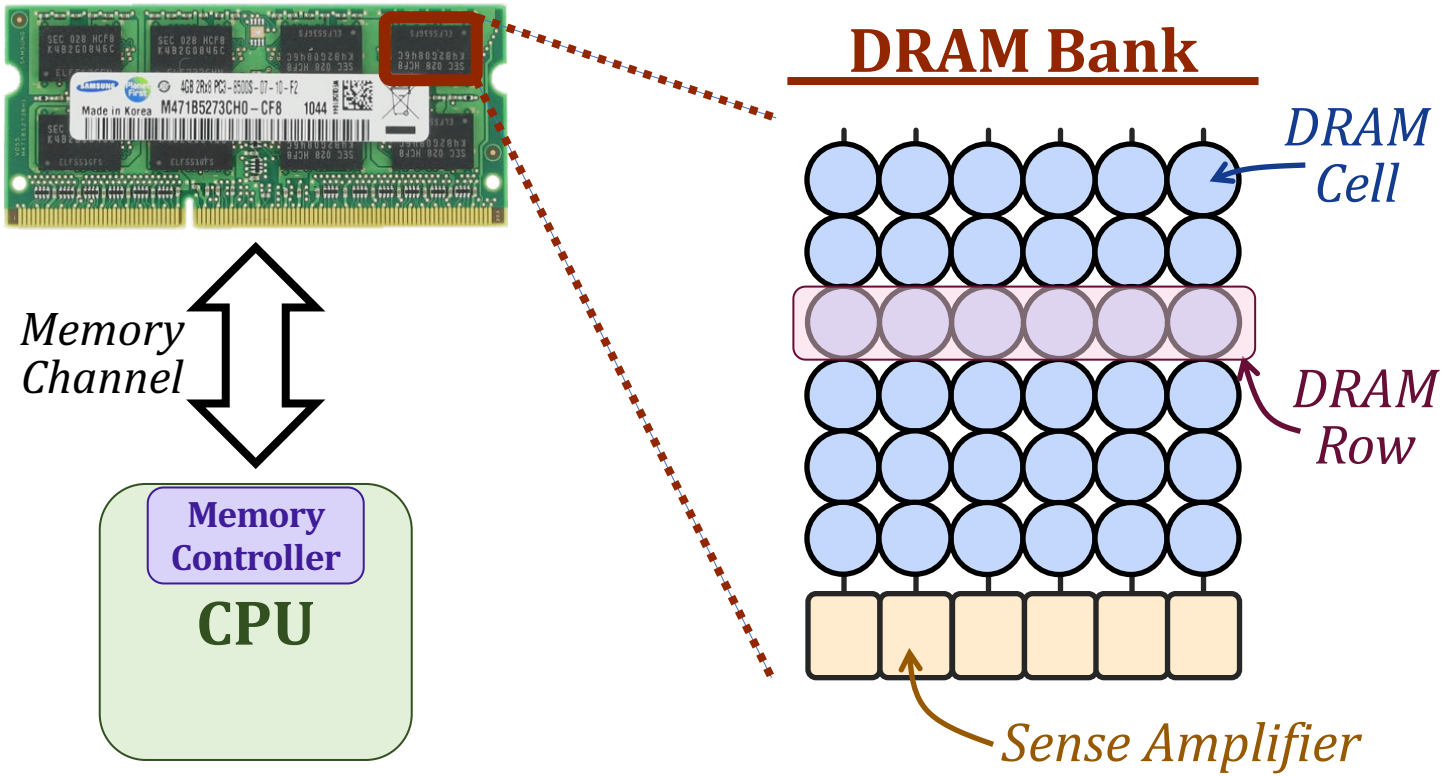
**PaCRAM: Partial Charge Restoration for Aggressive Mitigation**

- **Reduces the latency of preventive refreshes** issued by the existing solution
- **Adjusts the aggressiveness** of the existing solution

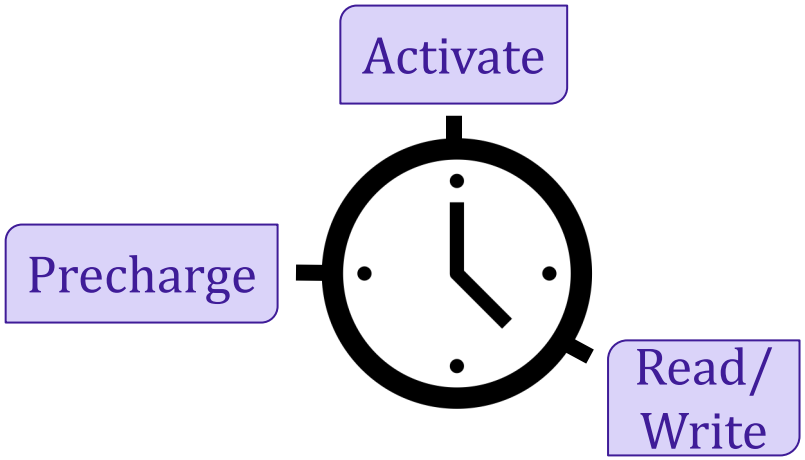
**Evaluation:** By reducing the existing solutions' performance and energy overheads, **improves system performance** and **energy efficiency** with **small additional area cost**

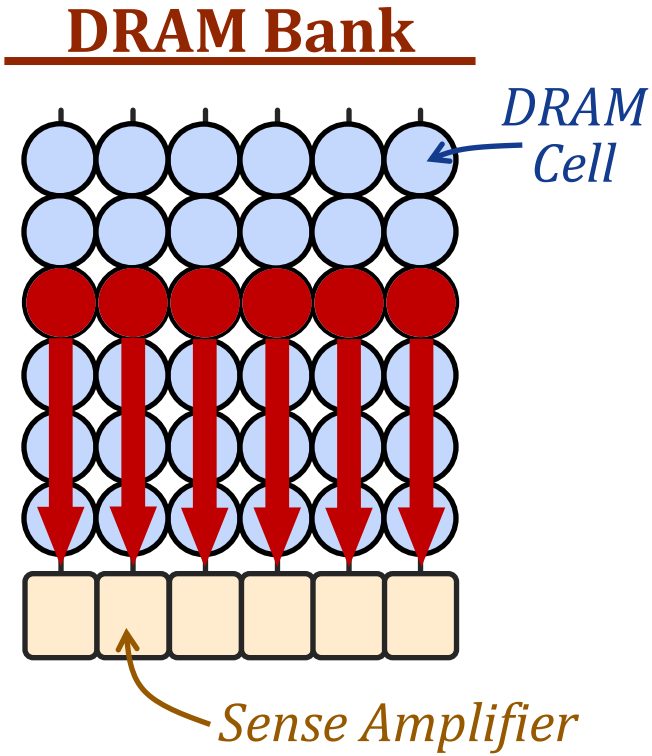
# Outline

# DRAM Organization



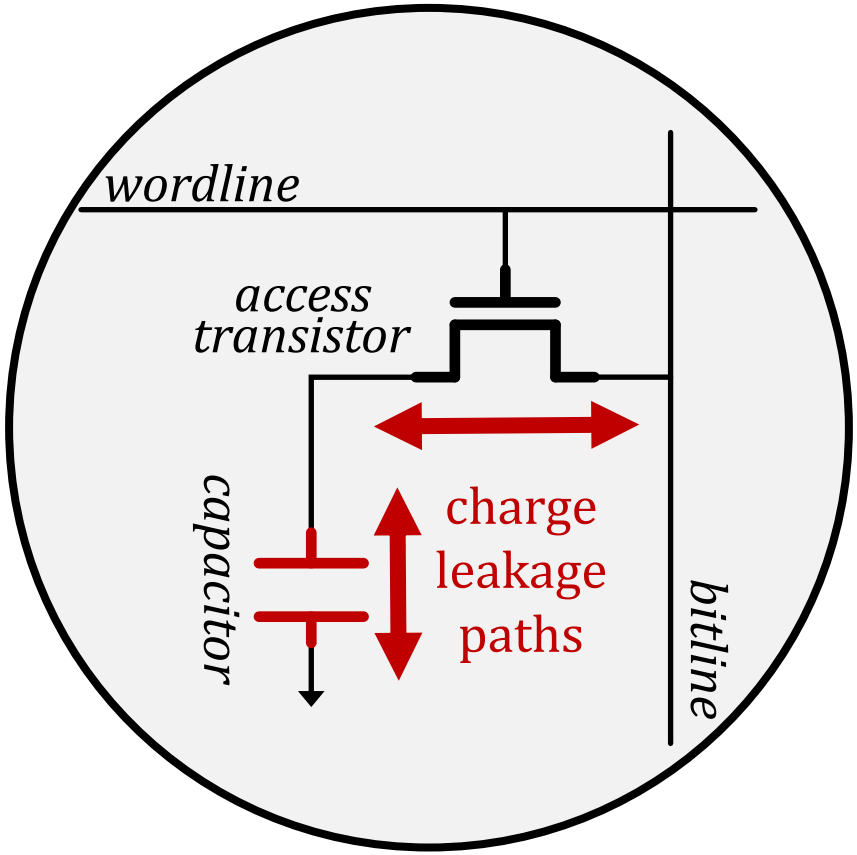
# DRAM Access


Activate
Precharge
Read/
Write


DRAM Bank
DRAM
Cell
Sense Amplifier

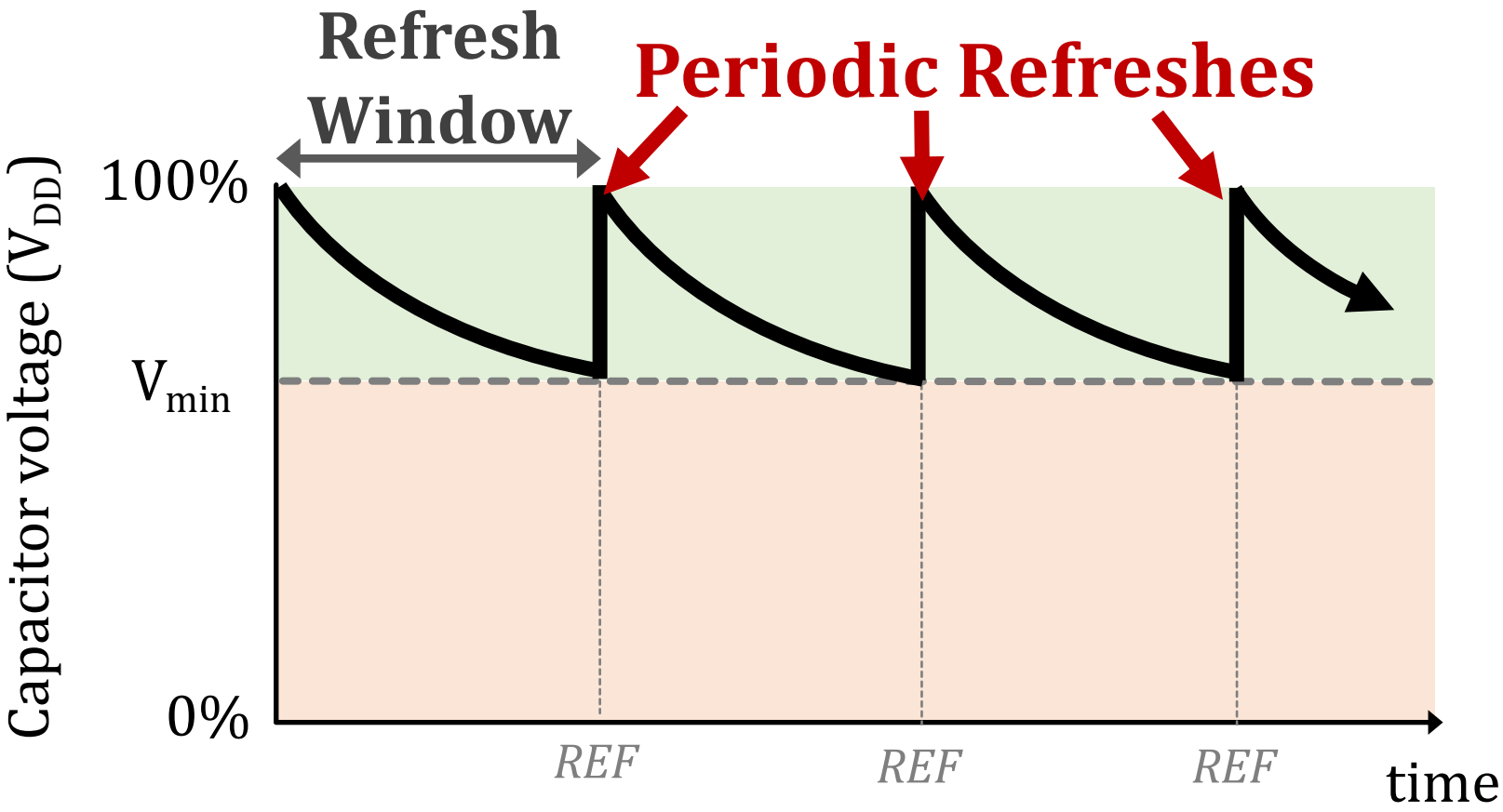
# DRAM Cell Leakage

Each cell encodes information in **leaky** capacitors



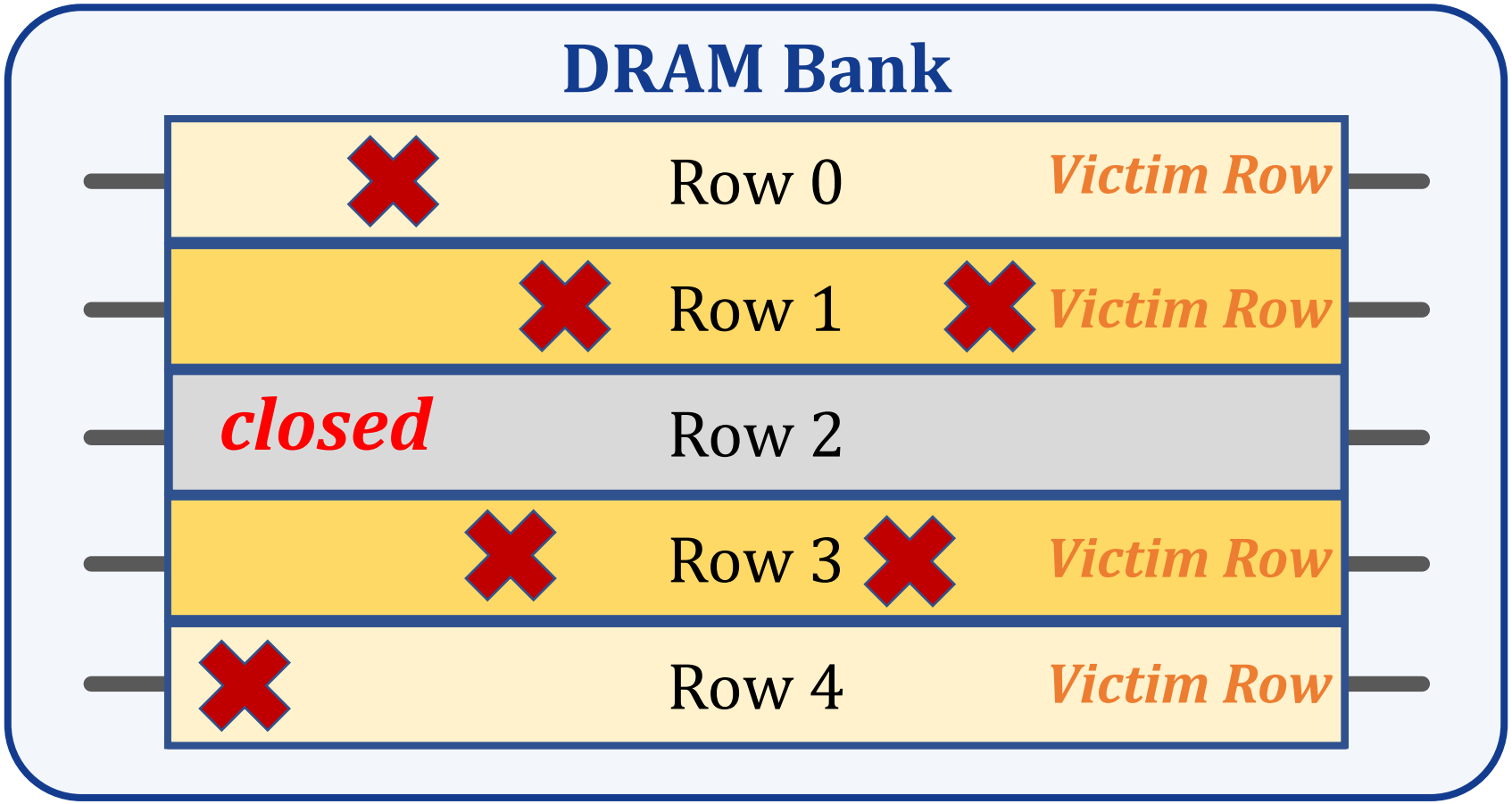

Stored data is **corrupted** if too much charge leaks (i.e., the capacitor voltage degrades too much)

# DRAM Refresh



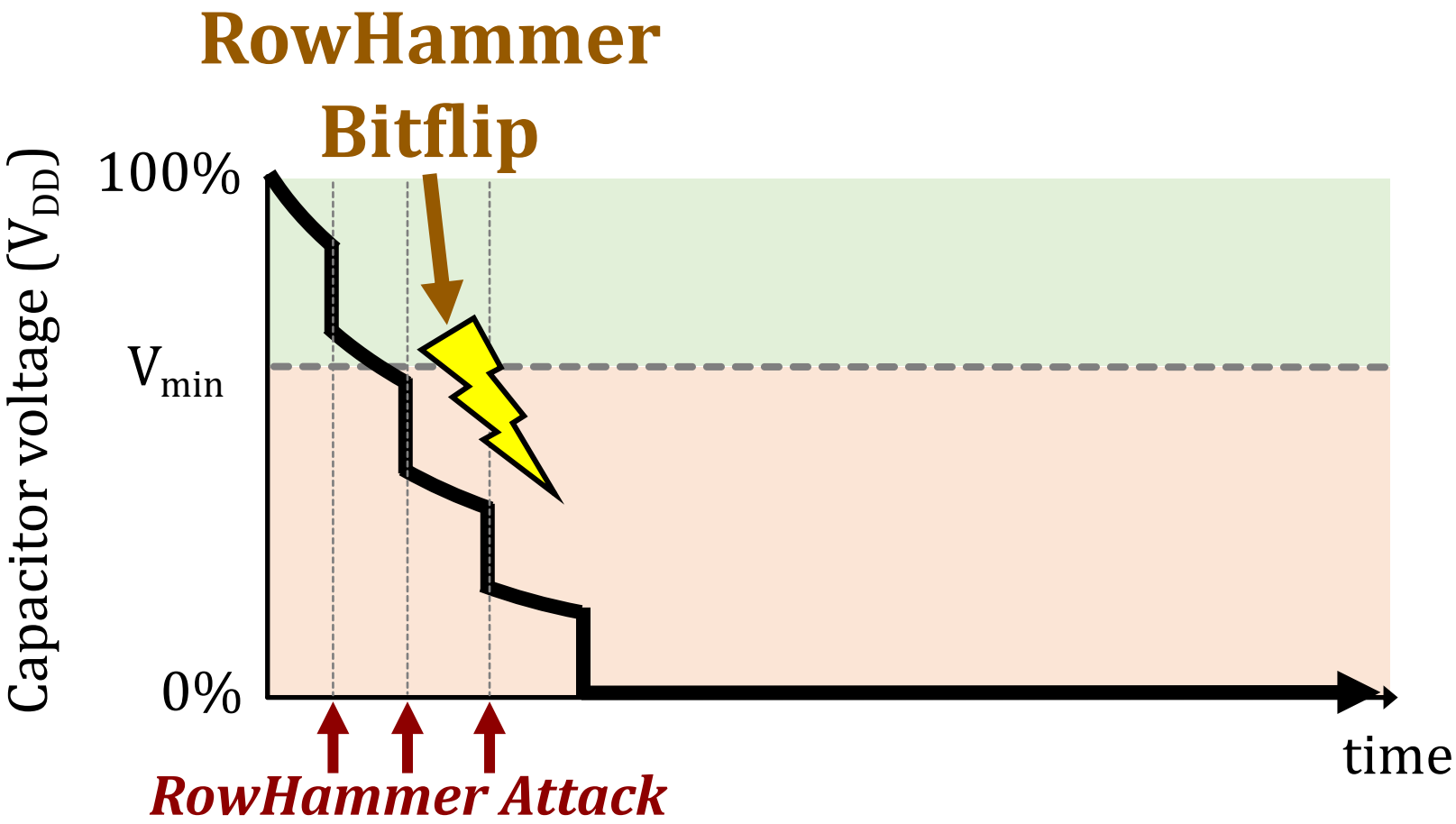

**Periodic refreshes** preserve stored data

# RowHammer Vulnerability

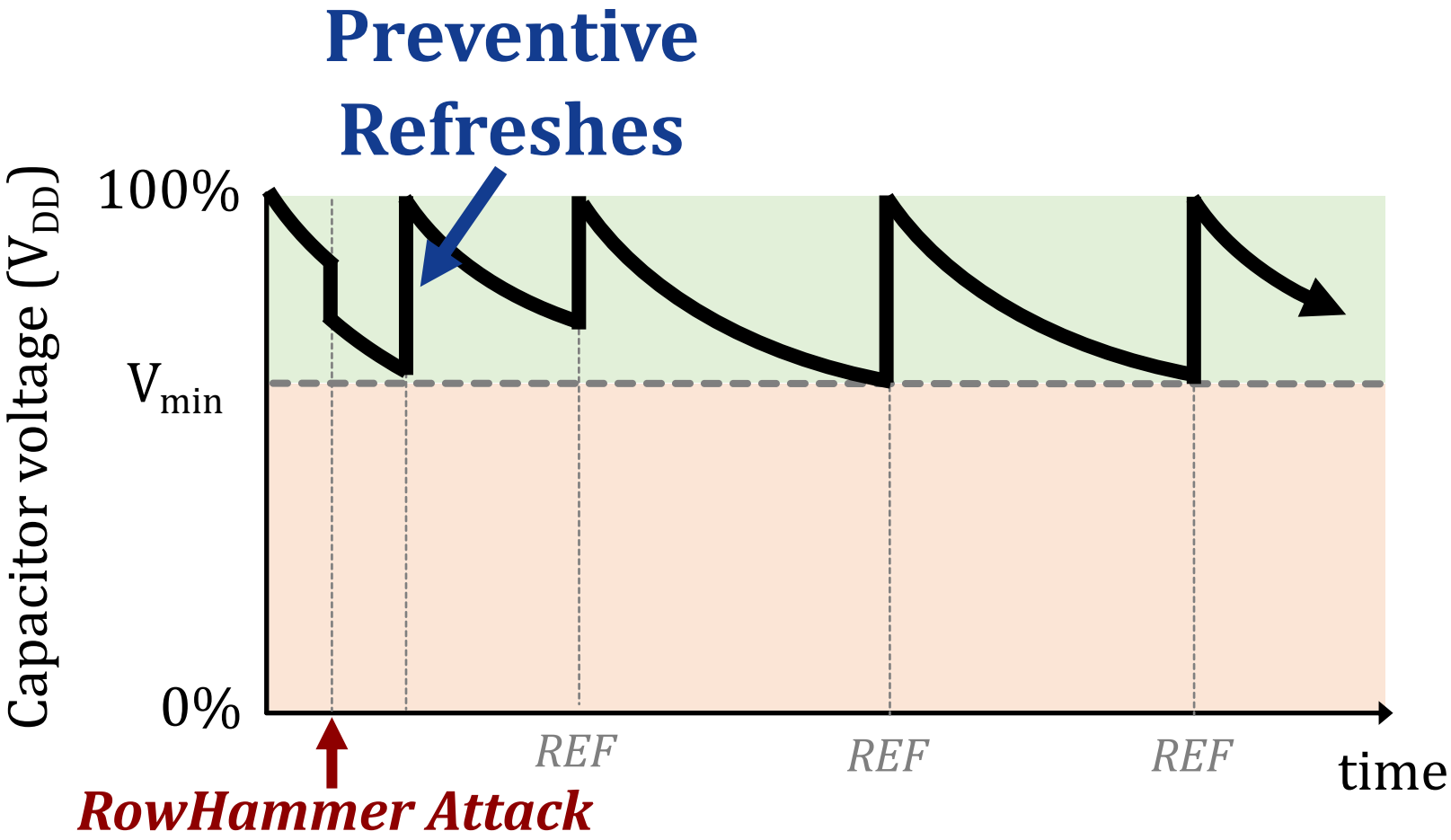



Repeatedly **opening** (activating) and **closing** (precharging) a DRAM row causes **RowHammer bitflips** in nearby cells

# RowHammer Bitflips

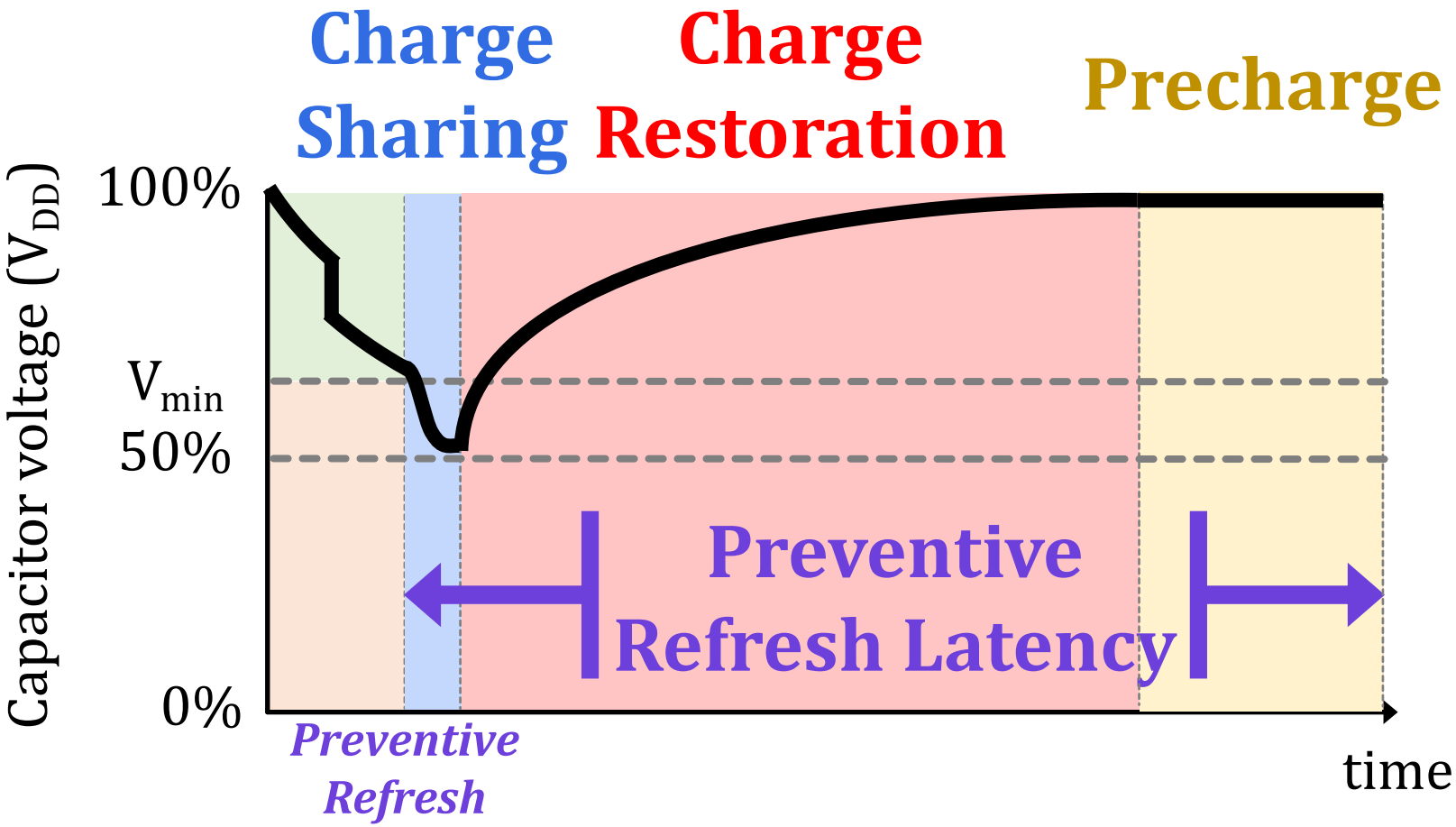


# RowHammer-Preventive Refresh




**Preventively refreshing** potential victim rows **mitigates** RowHammer bitflips

# A Closer Look




Preventive Refresh has a **non-negligible latency**

# Outline

Background

Problem and Motivation

Experimental Characterization of Real DRAM Chips

PaCRAM: Partial Charge Restoration for Aggressive Mitigation

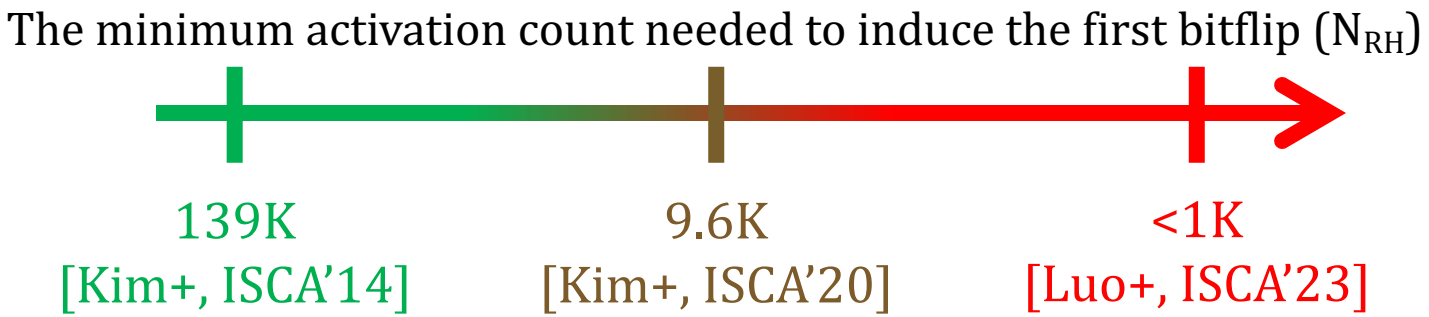
Evaluation

Conclusion

# Problem

- **RowHammer worsens** as **DRAM chip density increases**

- RowHammer bitflips occur at much lower activation counts

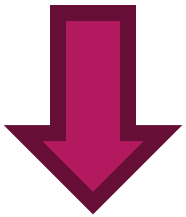



Mitigation mechanisms against RowHammer attacks
**induce higher overheads** as **RowHammer worsens**

# Reducing Mitigation Overhead

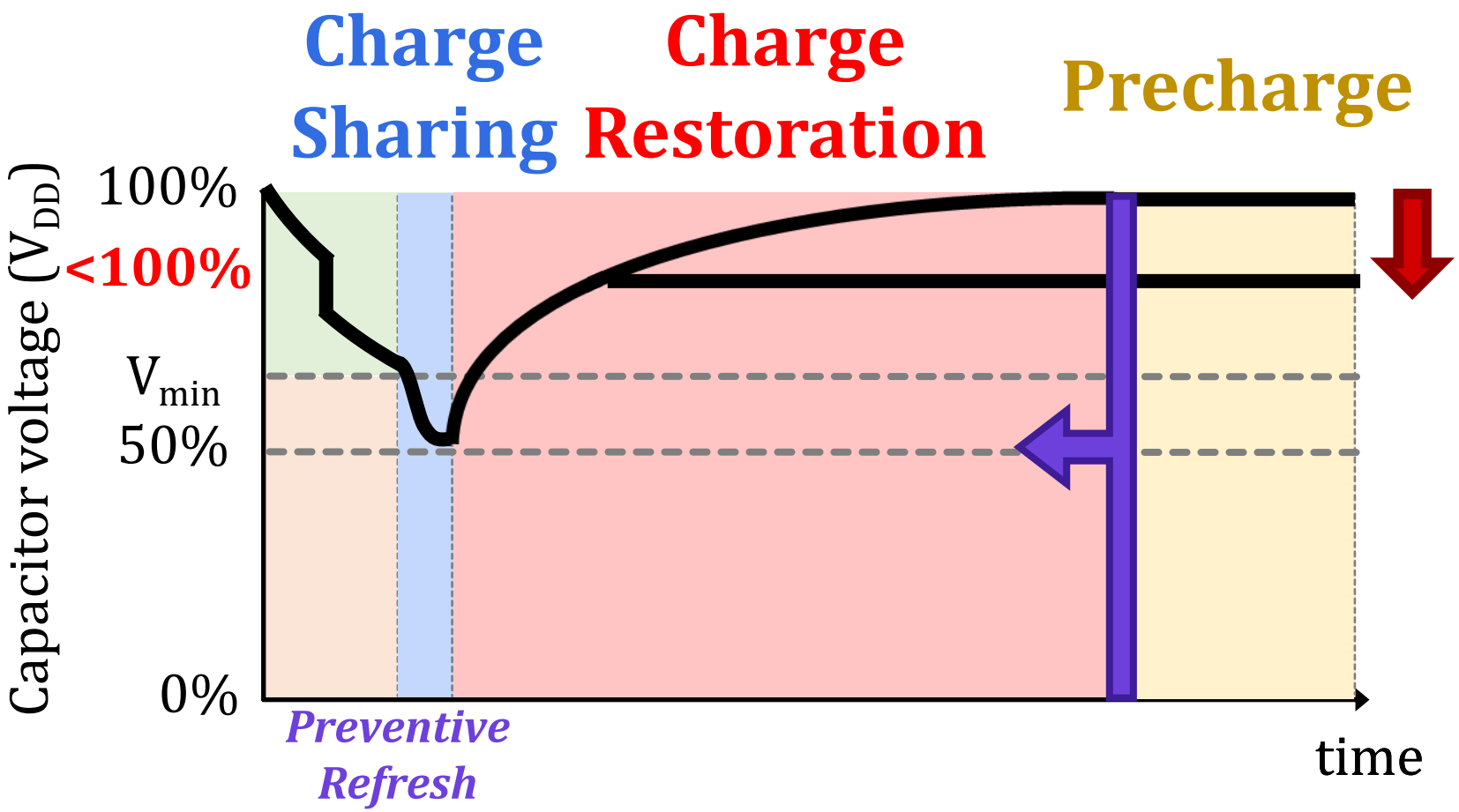
**Reduced Charge Restoration Latency**

**Reduced Preventive Refresh Latency**



**Reduced Preventive Refresh Overhead**

# Reducing Charge Restoration Latency




The charge of the cell are **partially restored**

# Partial Charge Restoration

**Partial charge restoration**
might **affect the voltage level** of the cell
and thus **affect the RowHammer vulnerability**

**To robustly perform partial charge restoration**,
we need to understand the DRAM chip's limit
with **rigorous characterization**

# Our Goal

To understand the **impact of charge restoration latency on RowHammer**

To leverage this understanding to **reduce the overheads of existing solutions**

# Outline

Background

Problem and Motivation

Experimental Characterization of Real DRAM Chips

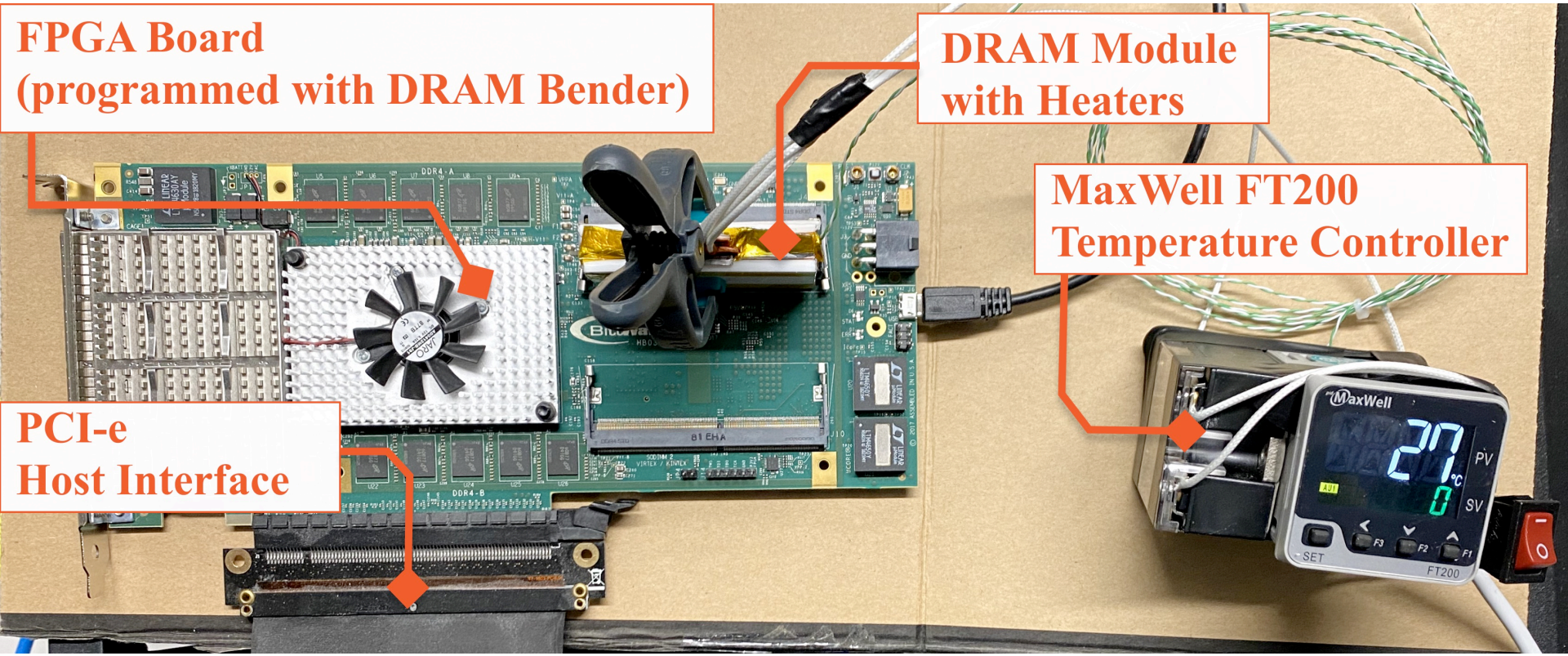
PaCRAM: Partial Charge Restoration for Aggressive Mitigation

Evaluation

Conclusion

# DRAM Testing Infrastructure

- DRAM Bender* on a Xilinx Alveo U200




Fine-grained control over **DRAM commands**, **timing parameters (±1.5ns)**, and **temperature (±0.5°C )**

*Olgun et al., TCAD, 2023

https://github.com/CMU-SAFARI/DRAM-Bender

# Tested DRAM Chips

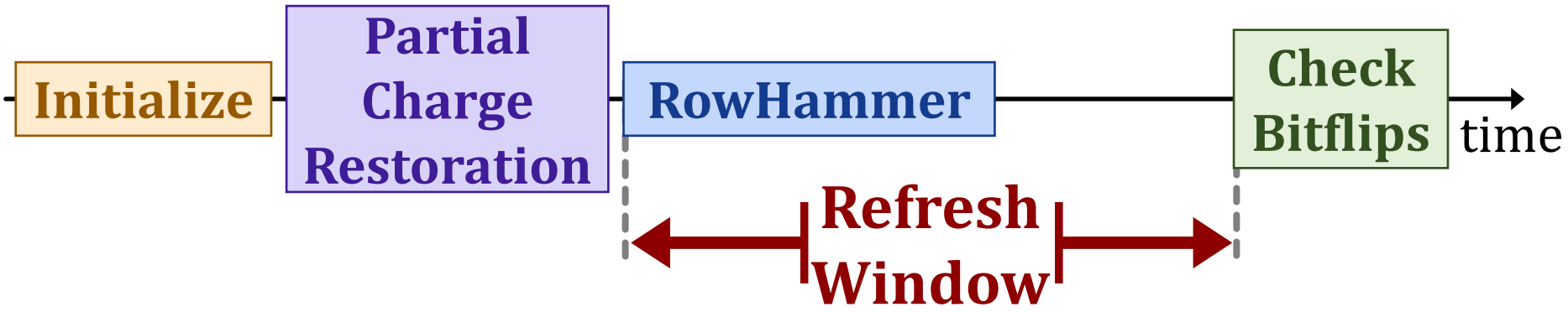
388 DDR4 DRAM chips from SK Hynix, Micron, and Samsung

| Chip Mfr. | Module IDs | #Chips (#Modules) | Form Factor | Die Density | Die Rev. | Chip Org. | Date Code |
|---|---|---|---|---|---|---|---|
| Mfr. H (SK Hynix) | H0 | 8 (1) | SO-DIMM | 4Gb | M | x8 | N/A |
| | H1 | 8 (1) | SO-DIMM | 4Gb | X | x8 | N/A |
| | H2 | 8 (1) | SO-DIMM | 4Gb | A | x8 | N/A |
| | H3 | 32 (1) | R-DIMM | 8Gb | M | x4 | N/A |
| | H4-5 | 32 (2) | R-DIMM | 8Gb | D | x8 | 2048 |
| | H6 | 32 (1) | R-DIMM | 8Gb | A | x4 | N/A |
| | H7-8 | 32 (2) | U-DIMM | 16Gb | C | x8 | 2136 |
| Mfr. M (Micron) | M0-1-2 | 48 (3) | R-DIMM | 8Gb | B | x4 | N/A |
| | M3 | 16 (1) | SO-DIMM | 16Gb | F | x8 | 2237 |
| | M4 | 4 (1) | SO-DIMM | 16Gb | E | x16 | 2046 |
| | M5 | 32 (1) | R-DIMM | 16Gb | E | x4 | 2014 |
| | M6 | 4 (1) | SO-DIMM | 16Gb | B | x16 | 2126 |
| Mfr. S (Samsung) | S0-1 | 32 (2) | U-DIMM | 4Gb | F | x8 | N/A |
| | S2-3-4 | 24 (3) | SO-DIMM | 4Gb | E | x8 | 1708 |
| | S5 | 4 (1) | SO-DIMM | 4Gb | C | x16 | N/A |
| | S6-7-8-9 | 32 (4) | U-DIMM | 8Gb | D | x8 | 2110 |
| | S10 | 16 (1) | R-DIMM | 8Gb | C | x8 | 1809 |
| | S11 | 8 (1) | R-DIMM | 8Gb | B | x8 | 2052 |
| | S12 | 8 (1) | U-DIMM | 8Gb | A | x8 | 2212 |
| | S13 | 8 (1) | U-DIMM | 16Gb | B | x8 | 2315 |

# Key Takeaway from Real Chip Experiments

Charge restoration latency
can **be significantly reduced**
for **a vast majority of preventive refreshes**

# Experiment Algorithm

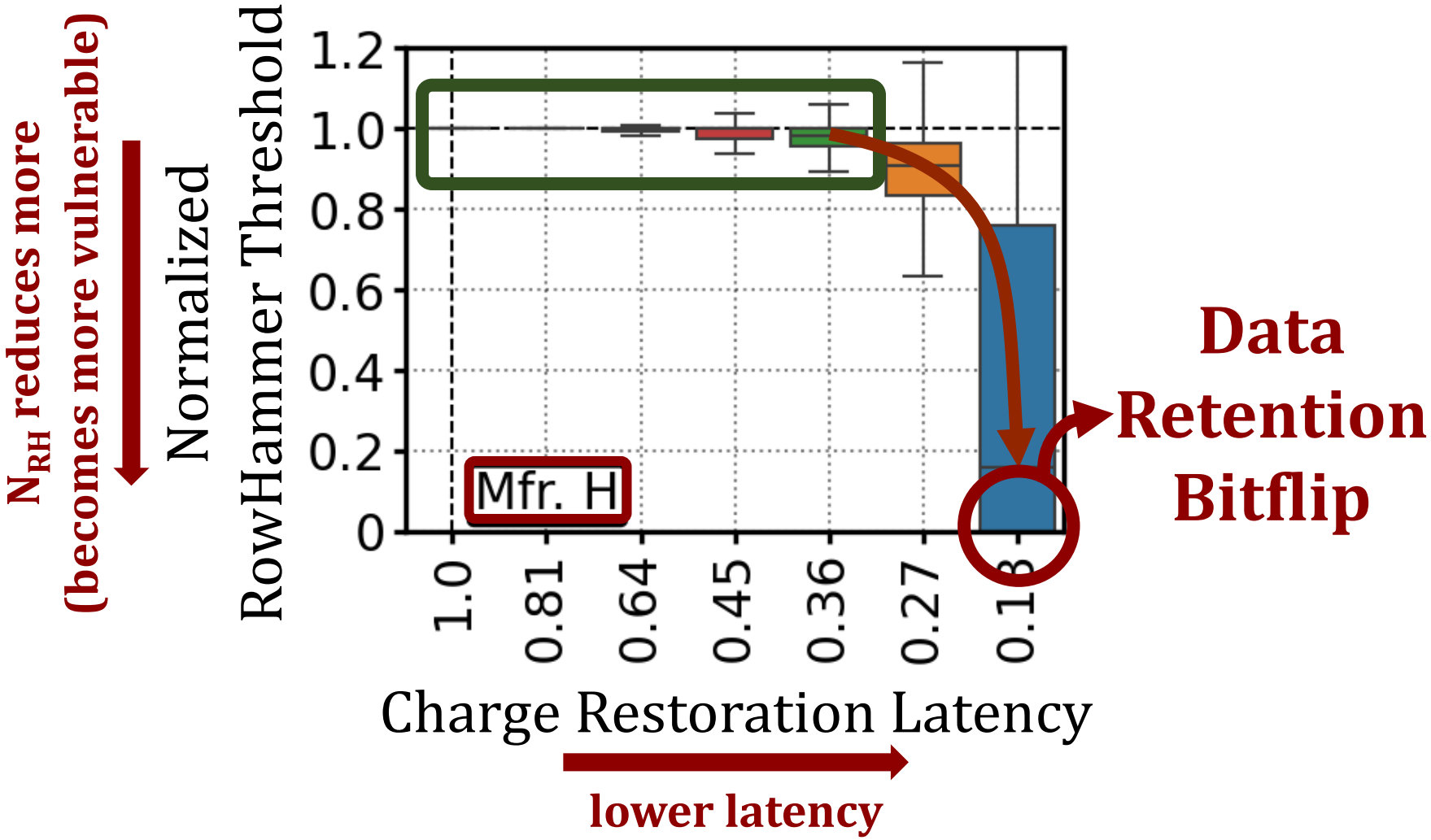


# RowHammer Vulnerability Metric

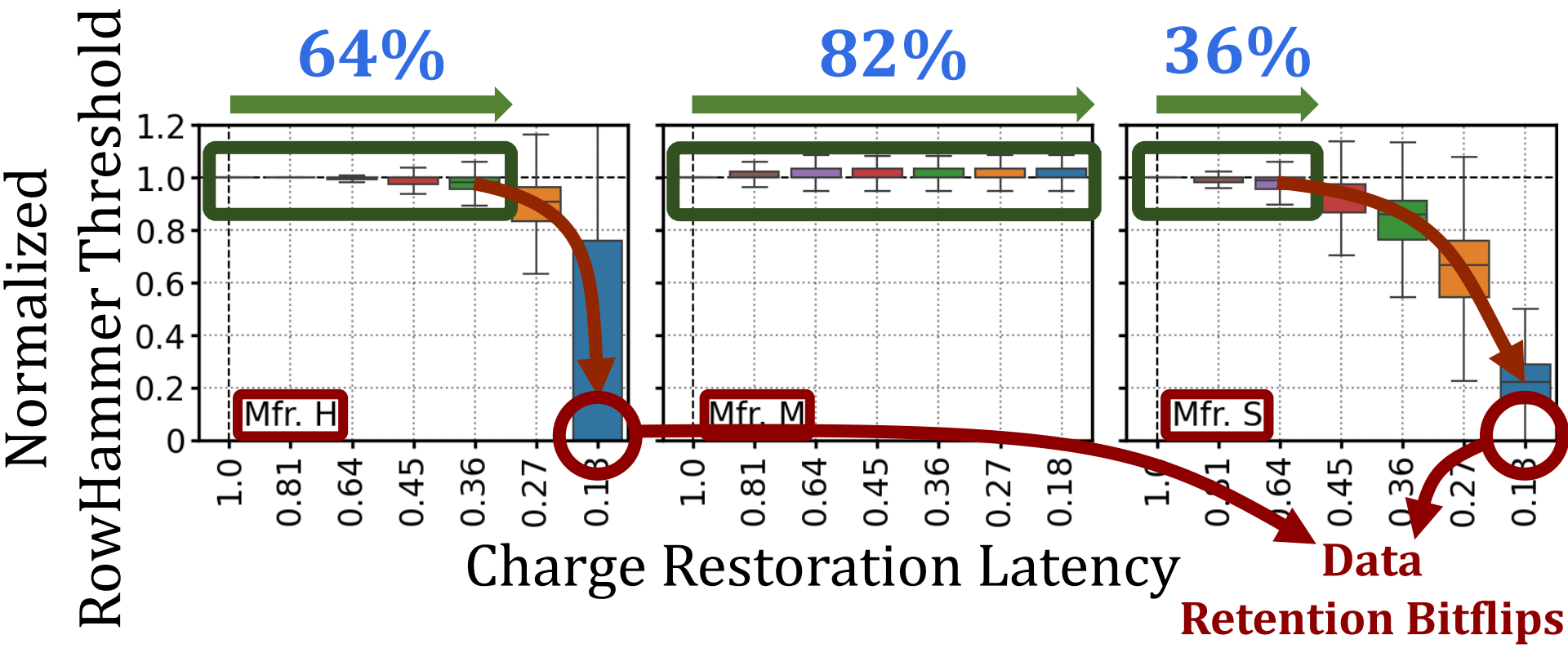
## Normalized RowHammer Threshold

$$\text{Norm}_{RH} = \frac{N_{RH} \textit{ with Partial Charge Restoration}}{N_{RH} \textit{ with Full Charge Restoration}}$$

# Effect of Partial Charge Restoration



# Effect of Partial Charge Restoration

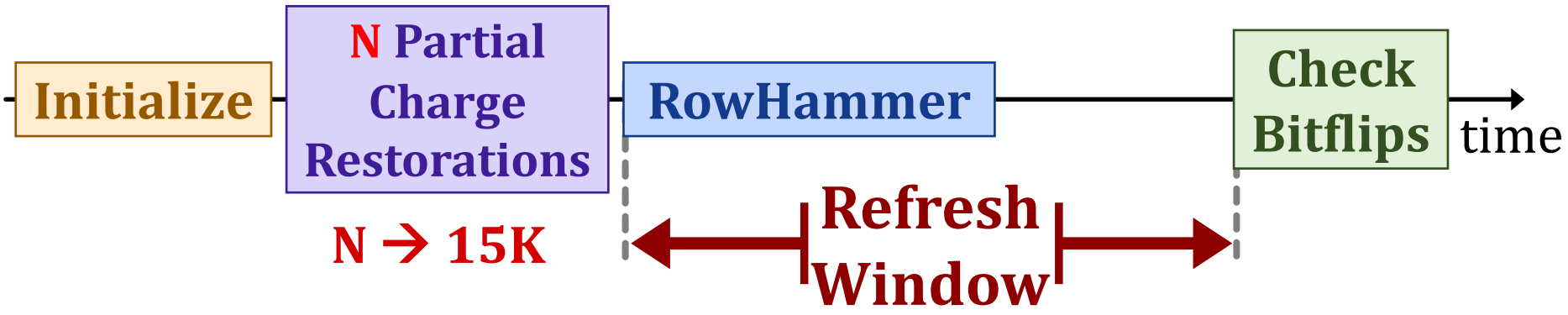



Charge restoration latency
can **be significantly reduced**

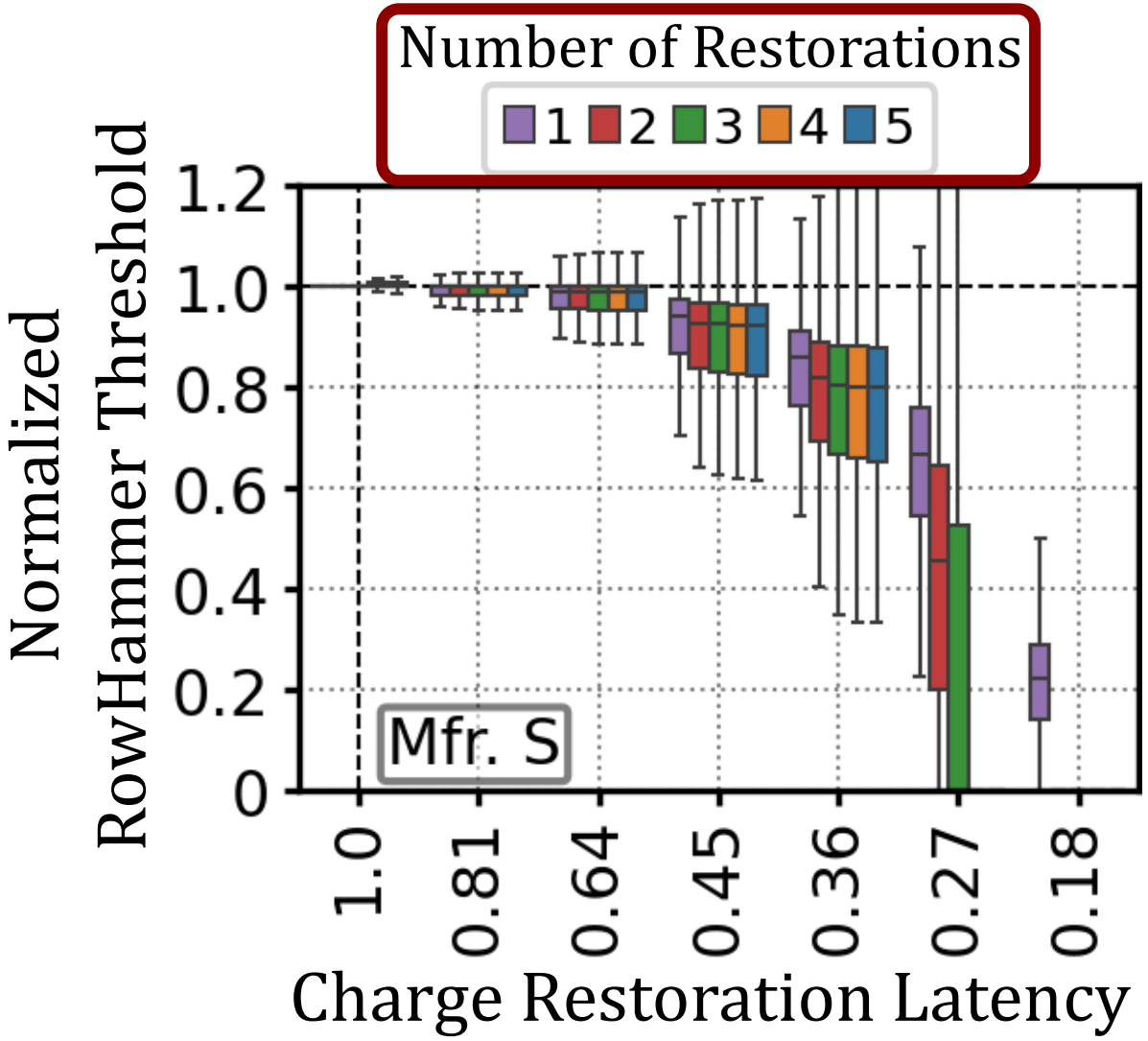
# Repeated Partial Charge Restorations

**Partial Charge Restoration** causes cells to **be partially restored**

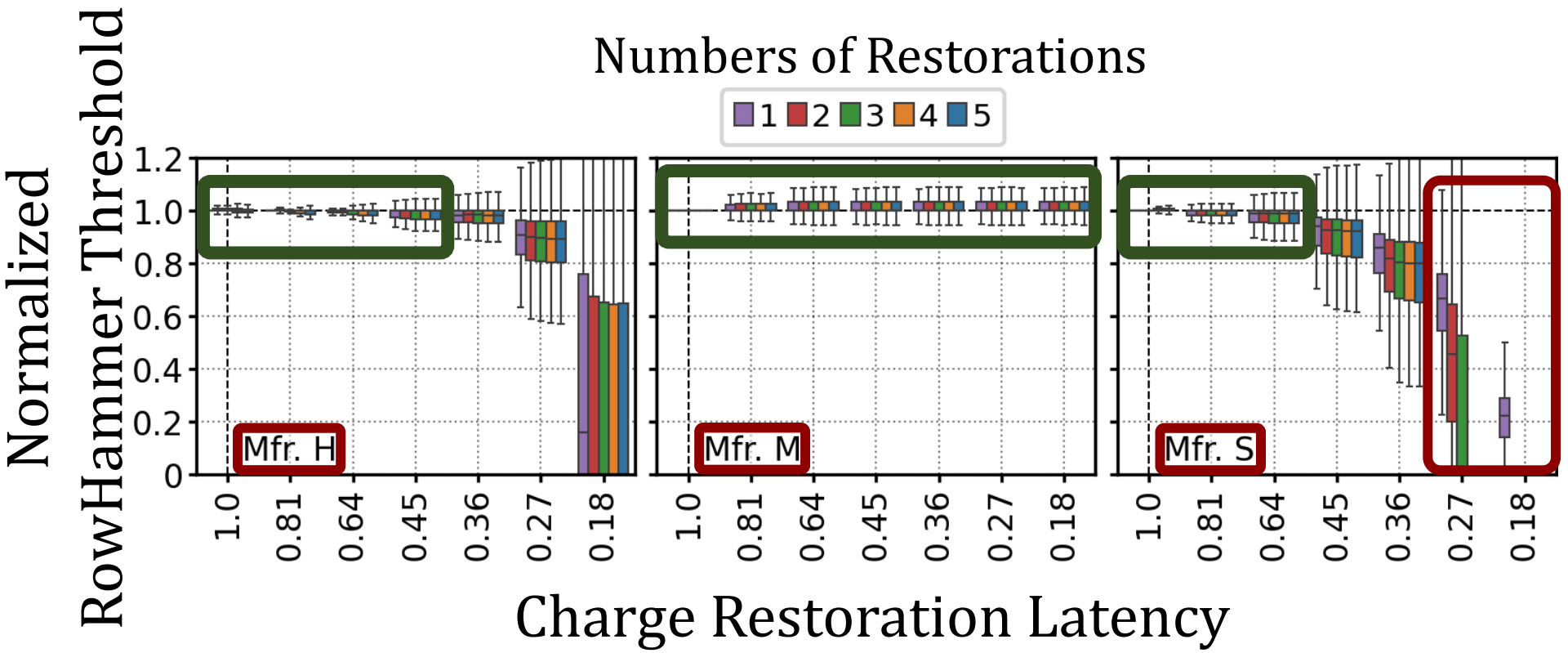
**Repeated Partial Charge Restorations** might **induce bitflips**



# Repeated Partial Charge Restorations

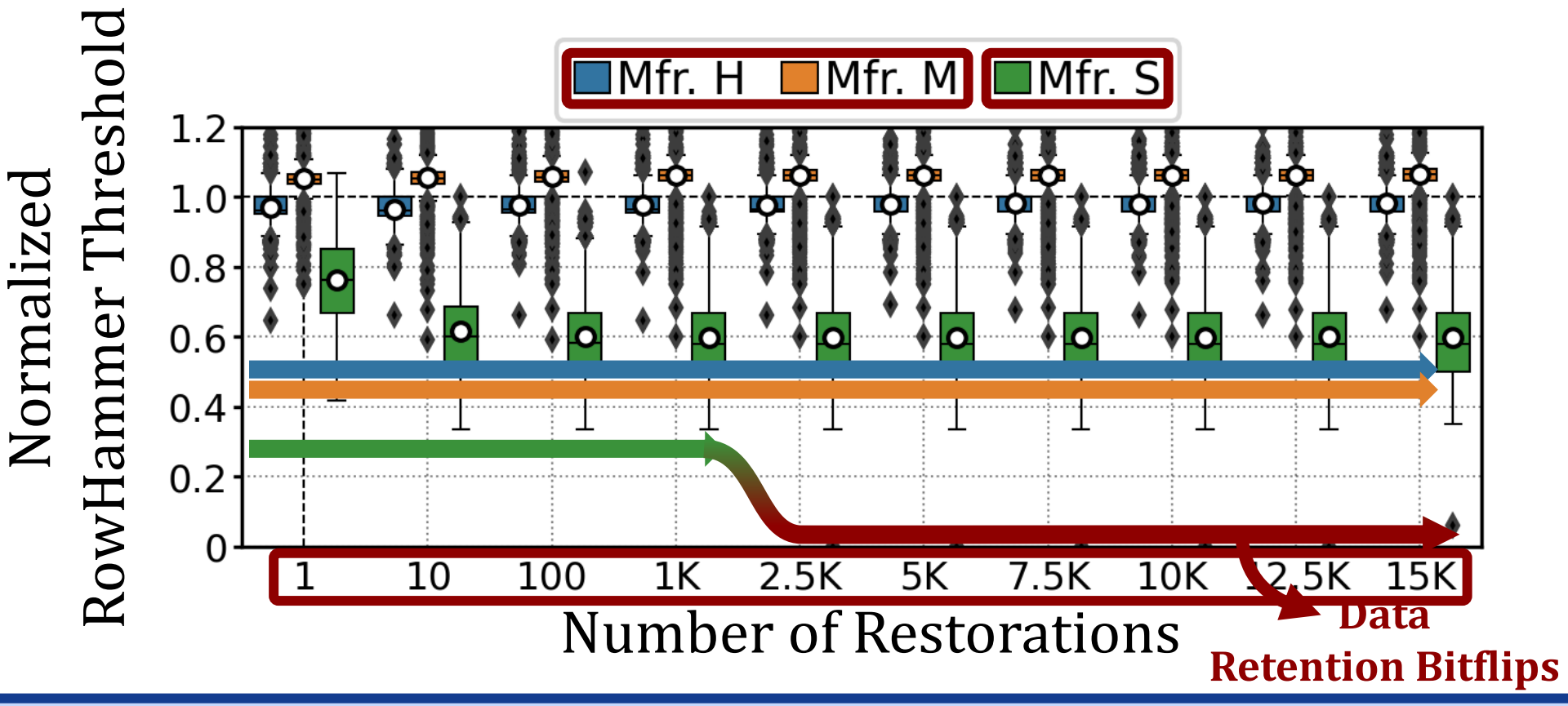


# Repeated Partial Charge Restorations




Partial charge restoration
can **be consecutively performed many times**

# Repeated Partial Charge Restorations




After 15K/15K/1K partial charge restoration, a single **full charge restoration** is needed

# Key Takeaway from Real Chip Experiments

Charge restoration latency
can **be significantly reduced**
for **a vast majority of preventive refreshes**

# Outline

Background

Problem and Motivation

Experimental Characterization of Real DRAM Chips

PaCRAM: Partial Charge Restoration for Aggressive Mitigation

Evaluation

Conclusion

# PaCRAM: Partial Charge Restoration for Aggressive Mitigation

- **Key Experimental Takeaway:** Charge restoration latency can be significantly reduced for a vast majority of preventive refreshes

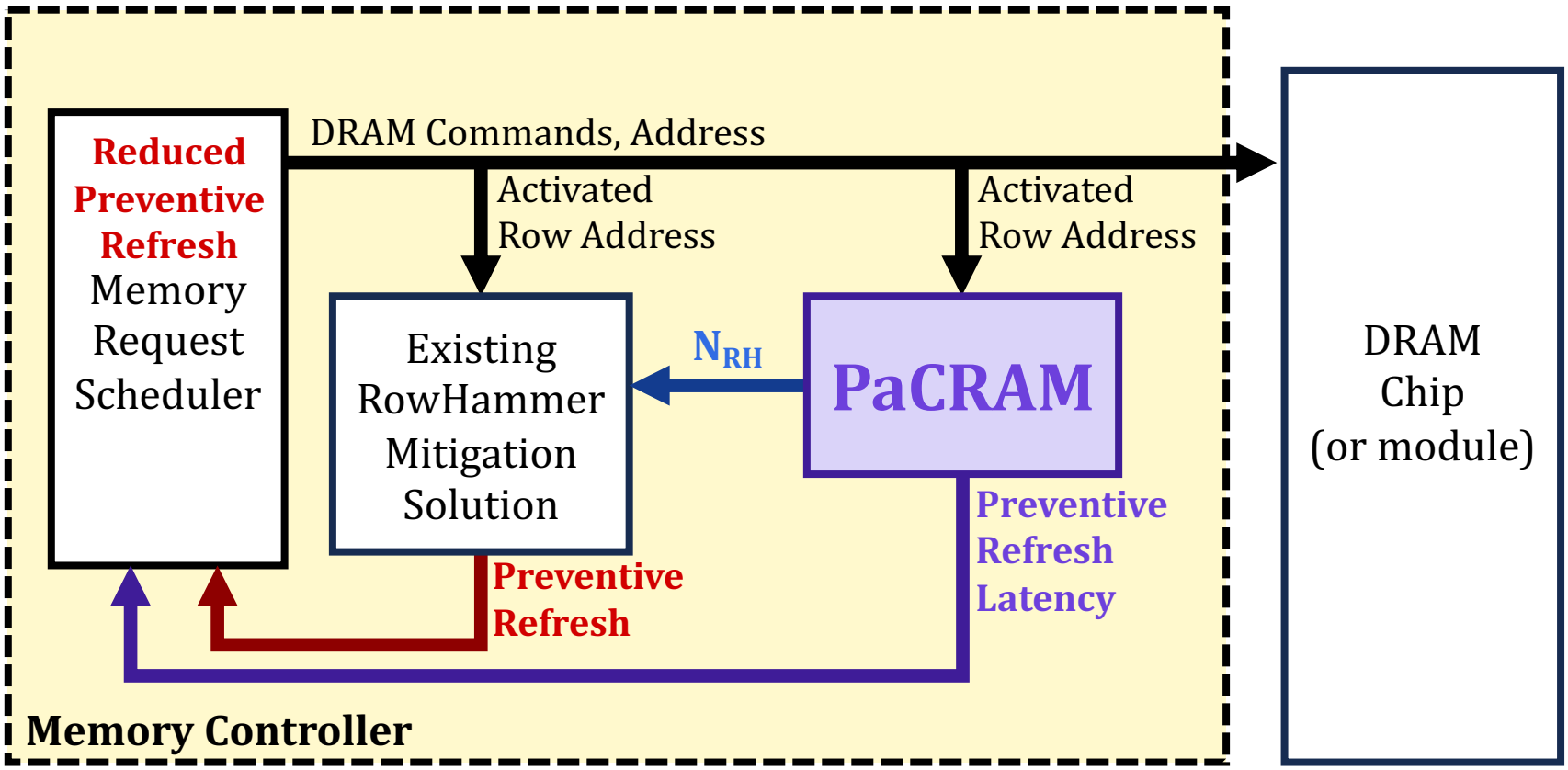
- **Key Idea:** Reduce preventive refresh overhead by reducing charge restoration latency of preventive refreshes

- **PaCRAM: Partial Charge Restoration for Aggressive Mitigation**
  - **Reduces the latency of preventive refreshes** issued by the existing solution
  - **Configures** the existing solution with **the reduced RowHammer threshold**
  - **Guarantees** that any victim is **not refreshed** by more than a safe number of **consecutive partial charge restorations**

**PaCRAM** significantly **reduces the performance and energy overhead** of existing solutions

# PaCRAM: Partial Charge Restoration for Aggressive Mitigation

- PaCRAM is implemented in the memory controller



# Outline

Background

Problem and Motivation

Experimental Characterization of Real DRAM Chips

PaCRAM: Partial Charge Restoration for Aggressive Mitigation

Evaluation

Conclusion

# Evaluation Methodology

- Cycle-level simulations using **Ramulator 2.0** [Luo+, CAL 2023]
- **System Configuration**:

| | |
|---|---|
| **Processor** | 1 or 4 cores, 3.2GHz clock frequency,<br>4-wide issue, 128-entry instruction window |
| **DRAM** | DDR5, 1 channel, 2 rank, 8 bank groups,<br>2 banks/bank group, 64K rows/bank |
| **Memory Ctrl.** | 64-entry read and write requests queues,<br>Scheduling policy: FR-FCFS [228, 229]<br>Address mapping: MOP [230] |
| **Last-Level Cache** | 2MB per core |

- **Workloads**: **62 single-core** and **60 four-core** multiprogrammed workloads from **5 benchmark suites**
- **$N_{RH}$**: {1K, 512, 256, 128, 64, 32} hammers
  The **minimum hammer count** needed to induce **the first bitflip**

# Configurations

- Paired with 5 state-of-the-art solutions
  - **PARA** [Kim+, ISCA'14]
  - **RFM** [JEDEC 2020]
  - **PRAC** [JEDEC 2024]
  - **Hydra** [Qureshi+, ISCA'22]
  - **Graphene** [Park+, MICRO'20]
- Configured using **experimental characterization data**

Mfr. S → **PaCRAM-S** → Charge Restoration Latency

Mfr. H → **PaCRAM-H** → Charge Restoration Latency

Mfr. M → **PaCRAM-M** → Charge Restoration Latency

# Configurations

- Paired with 5 state-of-the-art solutions
  - **PARA** [Kim+, ISCA'14]
  - **RFM** [JEDEC 2020]
  - **PRAC** [JEDEC 2024]
  - **Hydra** [Qureshi+, ISCA'22]
  - **Graphene** [Park+, MICRO'20]
- Configured using **experimental characterization data**

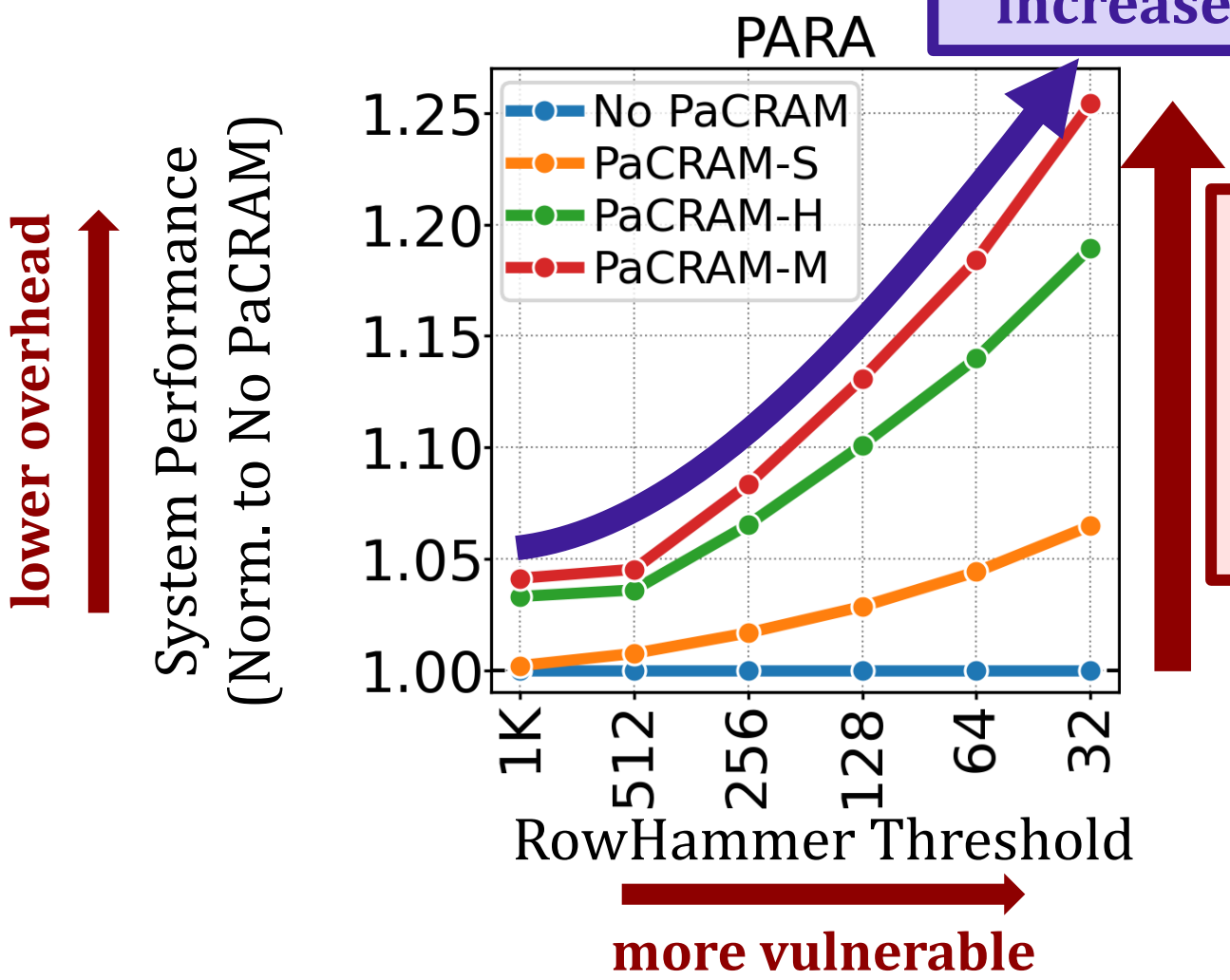
Mfr. S → **PaCRAM-S** → Charge Restoration Latency → **55%**

Mfr. H → **PaCRAM-H** → Charge Restoration Latency → **64%**

Mfr. M → **PaCRAM-M** → Charge Restoration Latency → **82%**

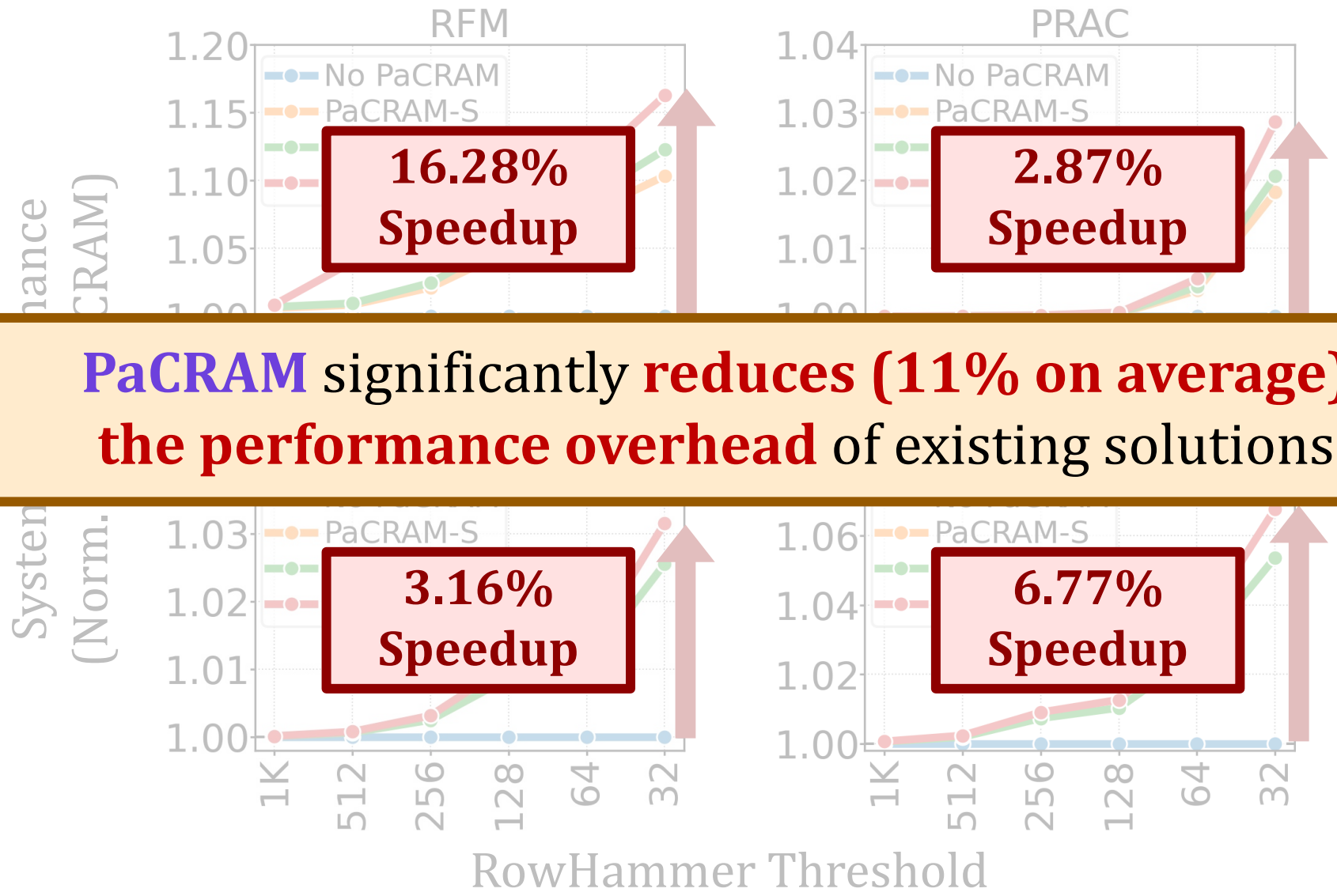
# Implication on Future Solutions

**PaCRAM's benefits increase** as $N_{RH}$ reduces



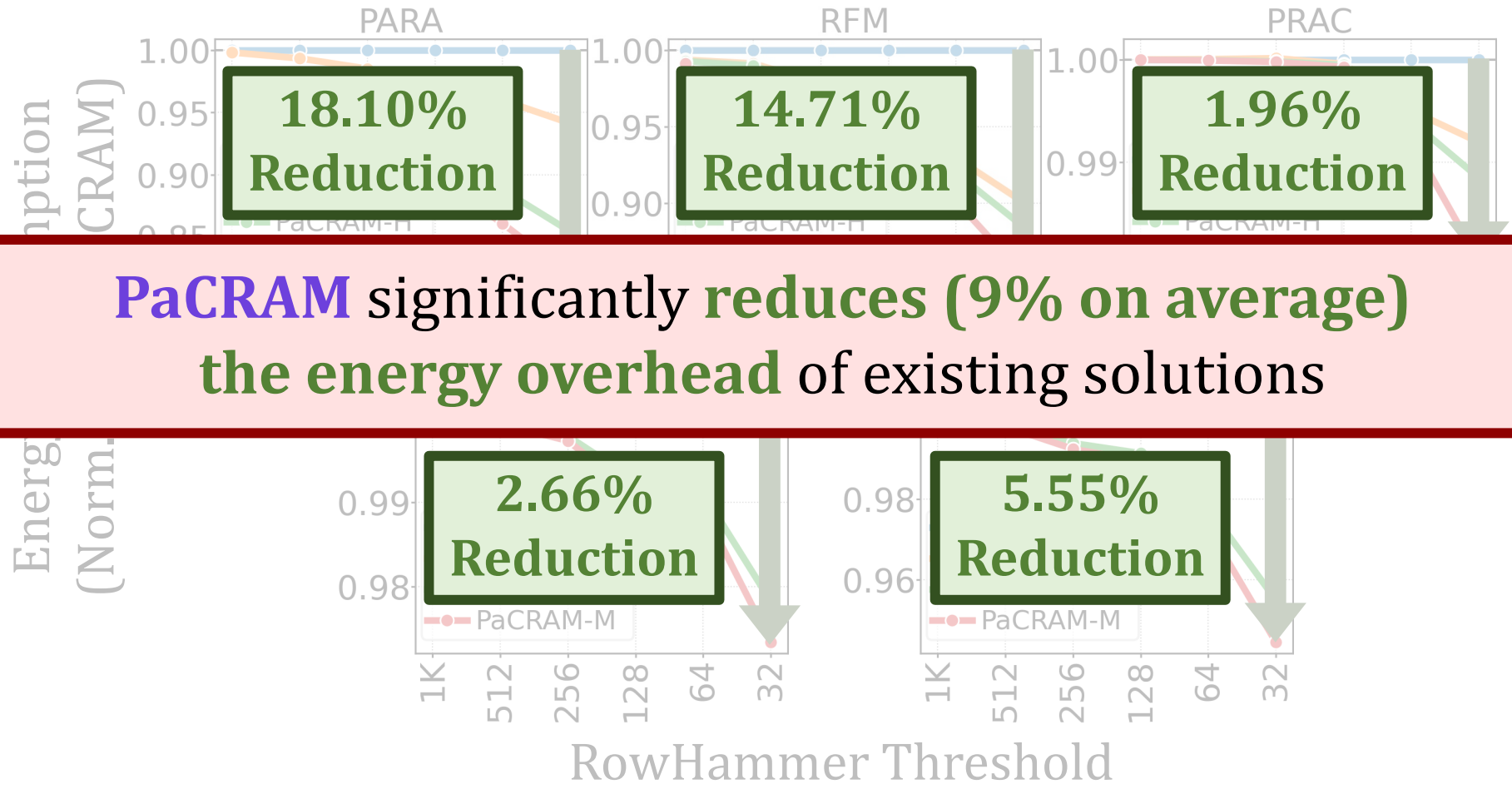

Reduction in mitigation overhead **25.43% Speedup**

# Implication on Future Solutions




**PaCRAM** significantly **reduces (11% on average) the performance overhead** of existing solutions

# Implication on Future Solutions




**PaCRAM** significantly **reduces (9% on average) the energy overhead** of existing solutions

# More in the Extended Version

- Effect of **partial charge restoration** on
  - Lowest observed $N_{RH}$ of DRAM modules
  - RowHammer Bit-Error-Rate
  - Half-Double access pattern
  - Data retention time
- Combined effect of **partial charge restoration** and **temperature**
- **Algorithms and details** of our experiments
- **Detailed implementation of PaCRAM**
  - Hardware complexity analysis
    (0.09% area overhead on a high-end Intel Xeon Processer)
  - Profiling cost analysis
- Reducing charge restoration latency for **periodic refreshes**

# More in the Extended Version

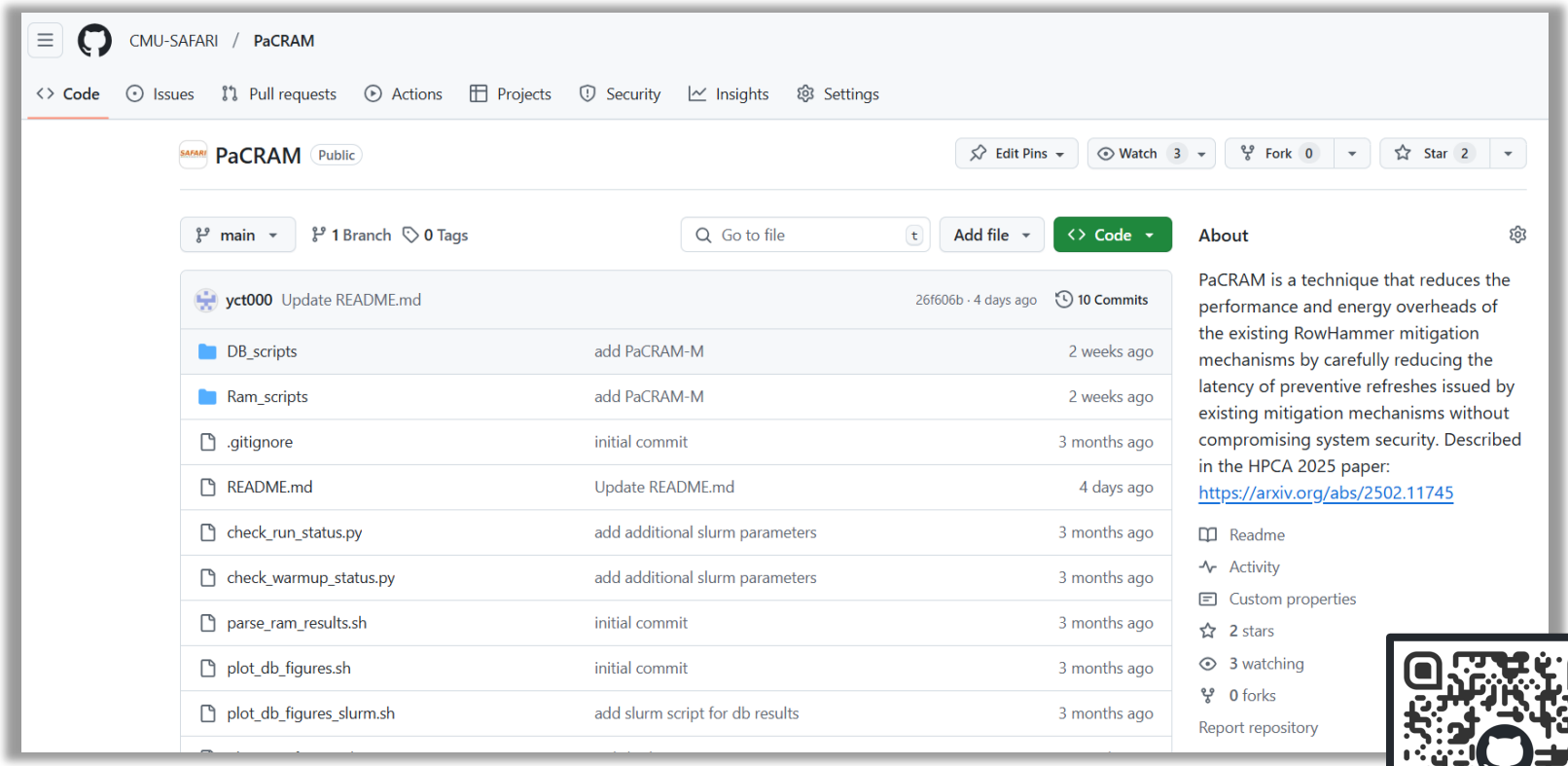
**Understanding RowHammer Under Reduced Refresh Latency: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions**

Yahya Can Tuğrul[§†] A. Giray Yağlıkçı[§] İsmail Emir Yüksel[§] Ataberk Olgun[§]
Oğuzhan Canpolat[§†] Nisa Bostancı[§] Mohammad Sadrosadati[§] Oğuz Ergin[‡†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology* [‡]*University of Sharjah*

*Read disturbance in modern DRAM chips is a widespread weakness that is used for breaking memory isolation, one of the fundamental building blocks of system security and privacy. RowHammer is a prime example of read disturbance in DRAM where repeatedly accessing (hammering) a row of DRAM cells* cell causes physically nearby DRAM cells to lose their charge (i.e., charge leakage) and exhibit bitflips. *RowHammer* is a prime example of such DRAM read disturbance phenomena where a row of DRAM cells (i.e., a DRAM row) can experience bitflips when another physically nearby DRAM row (i.e.,

https://arxiv.org/pdf/2502.11745

# PaCRAM is Open Source and Artifact Evaluated




https://github.com/CMU-SAFARI/PaCRAM

# Outline

Background

Problem and Motivation

Experimental Characterization of Real DRAM Chips

PaCRAM: Partial Charge Restoration for Aggressive Mitigation

Evaluation

Conclusion

# Conclusion

**The first rigorous experimental characterization** on the effect of preventive refresh latency on RowHammer

- **388 DDR4 DRAM chips** from **three major vendors**

The latency of **a vast majority of preventive refresh** can **be significantly reduced** without jeopardizing the data integrity of a DRAM chip

**PaCRAM: Partial Charge Restoration for Aggressive Mitigation**

- **Reduces the latency of preventive refreshes** issued by the existing solution
- **Adjusts the aggressiveness** of the existing solution

**PaCRAM** significantly **reduces the performance and energy overheads** of existing solutions

# Understanding RowHammer Under Reduced Refresh Latency: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions


**Yahya Can Tuğrul**

A. Giray Yağlıkçı  İsmail Emir Yüksel

Ataberk Olgun  Oğuzhan Canpolat  Nisa Bostancı

Mohammad Sadrosadati  Oğuz Ergin  Onur Mutlu

SAFARI

ETH zürich

kasırga

TOBB ETÜ

University of Economics & Technology

# Backup Slides


**Yahya Can Tuğrul**

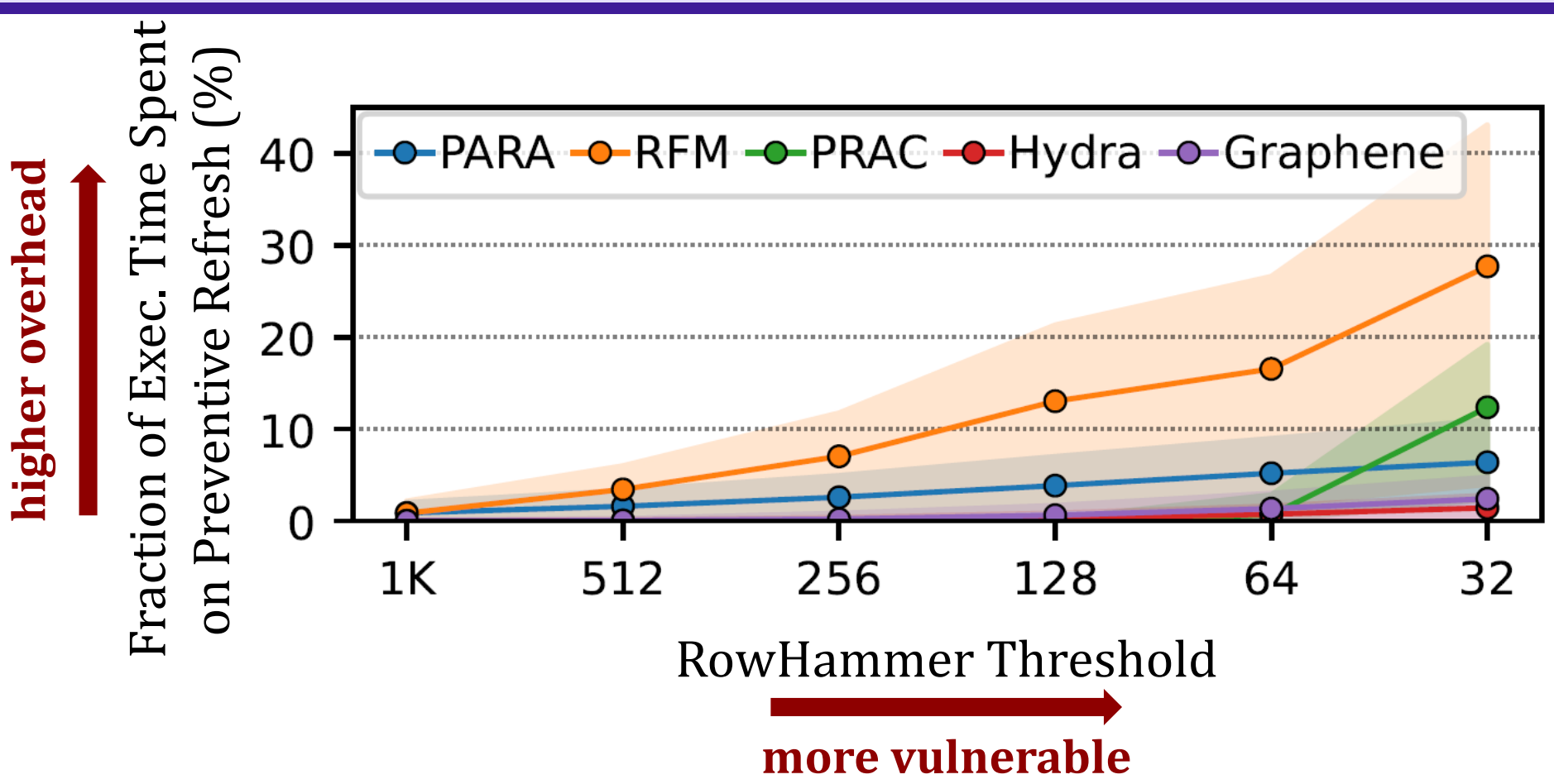
A. Giray Yağlıkçı  İsmail Emir Yüksel

Ataberk Olgun  Oğuzhan Canpolat  Nisa Bostancı

Mohammad Sadrosadati  Oğuz Ergin  Onur Mutlu

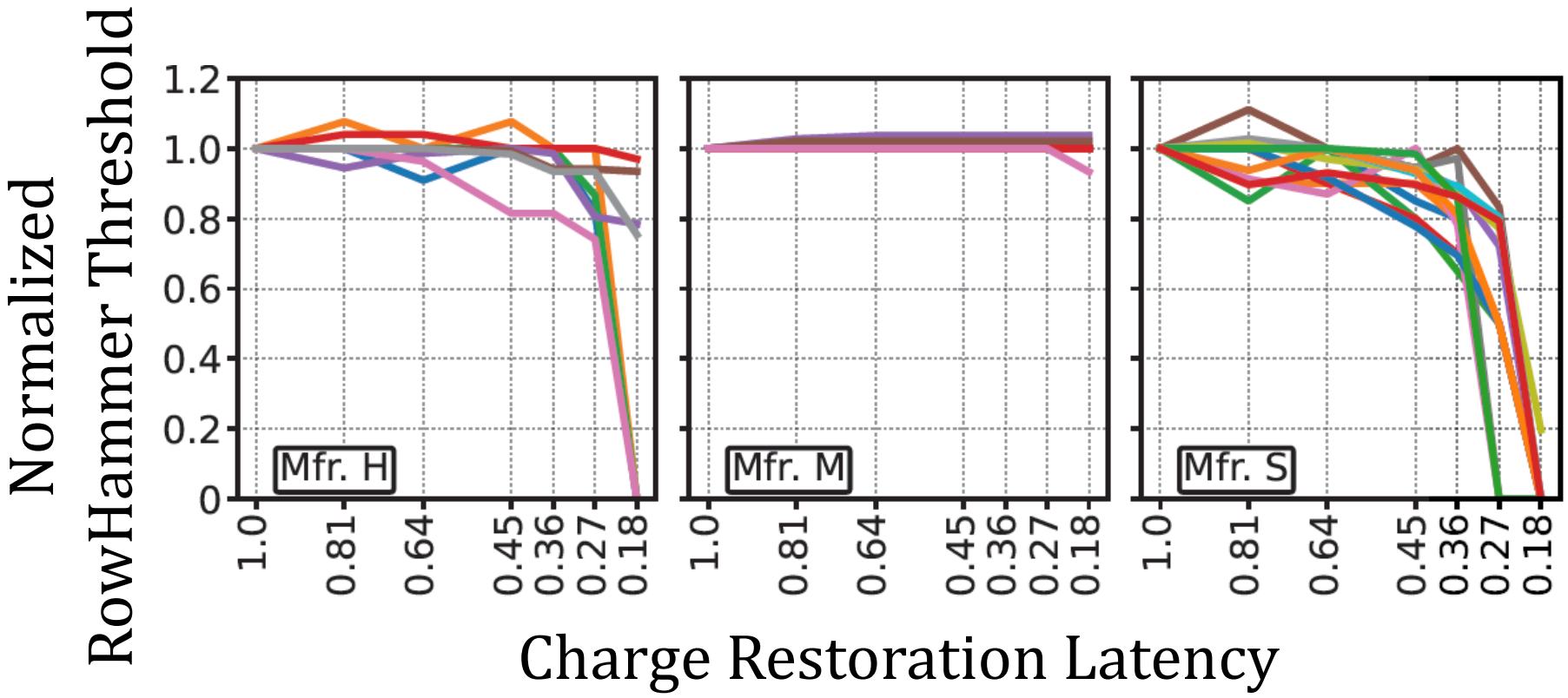


# Preventive Refresh Overhead



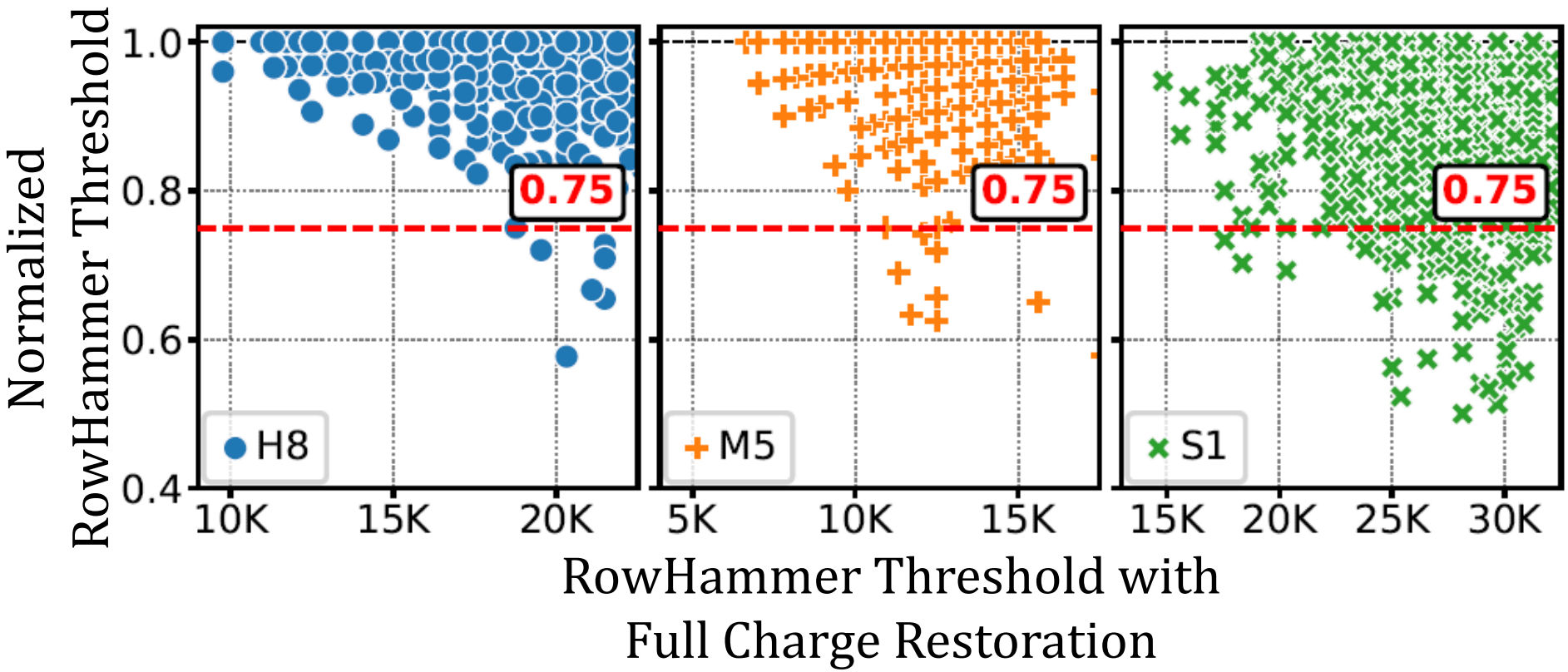

Mitigation mechanisms against RowHammer attacks
**induce higher overheads** as **RowHammer worsens**

# Effect of Partial Charge Restoration on Lowest Observed $N_{RH}$ of DRAM Modules



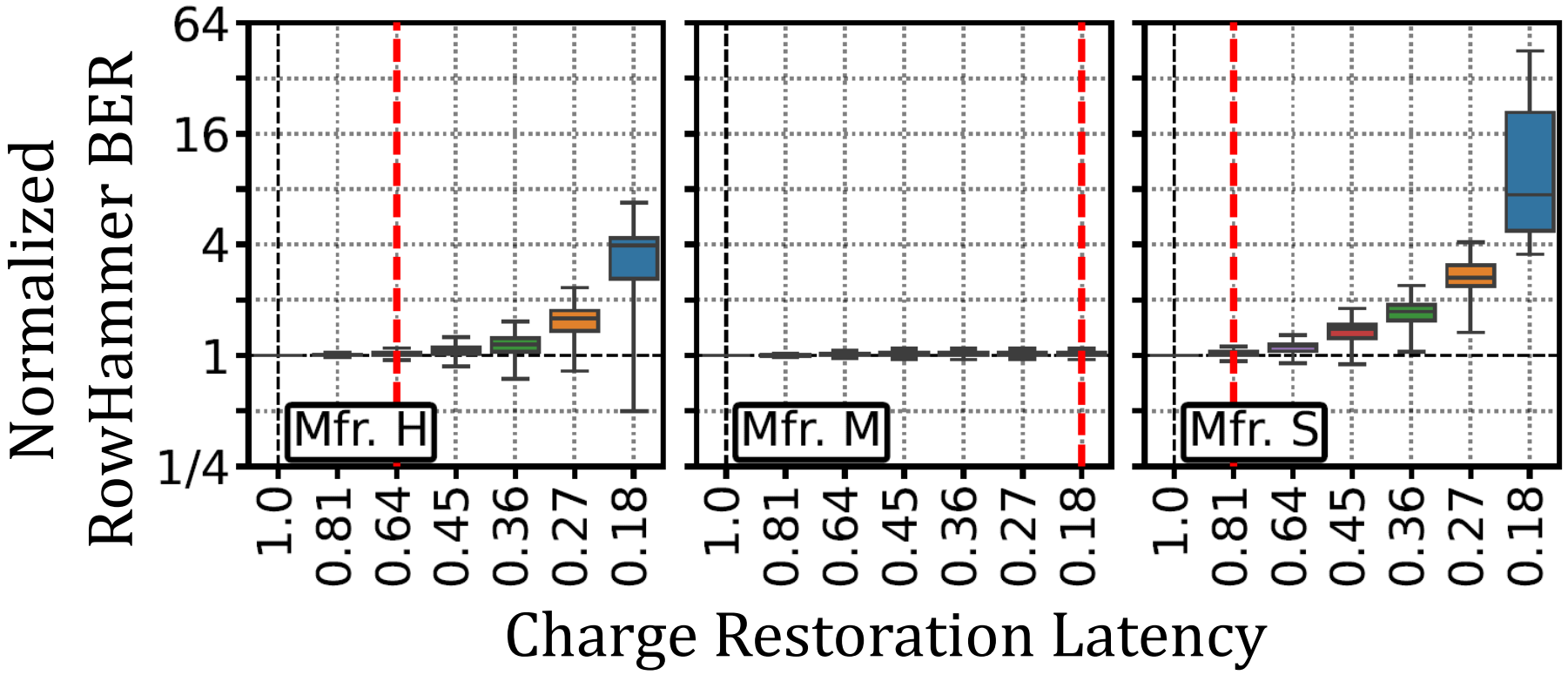

Charge restoration latency
can **be reduced without significantly affecting the Lowest Observed $N_{RH}$ of DRAM Modules**

# $N_{RH}$ Reduction Distribution



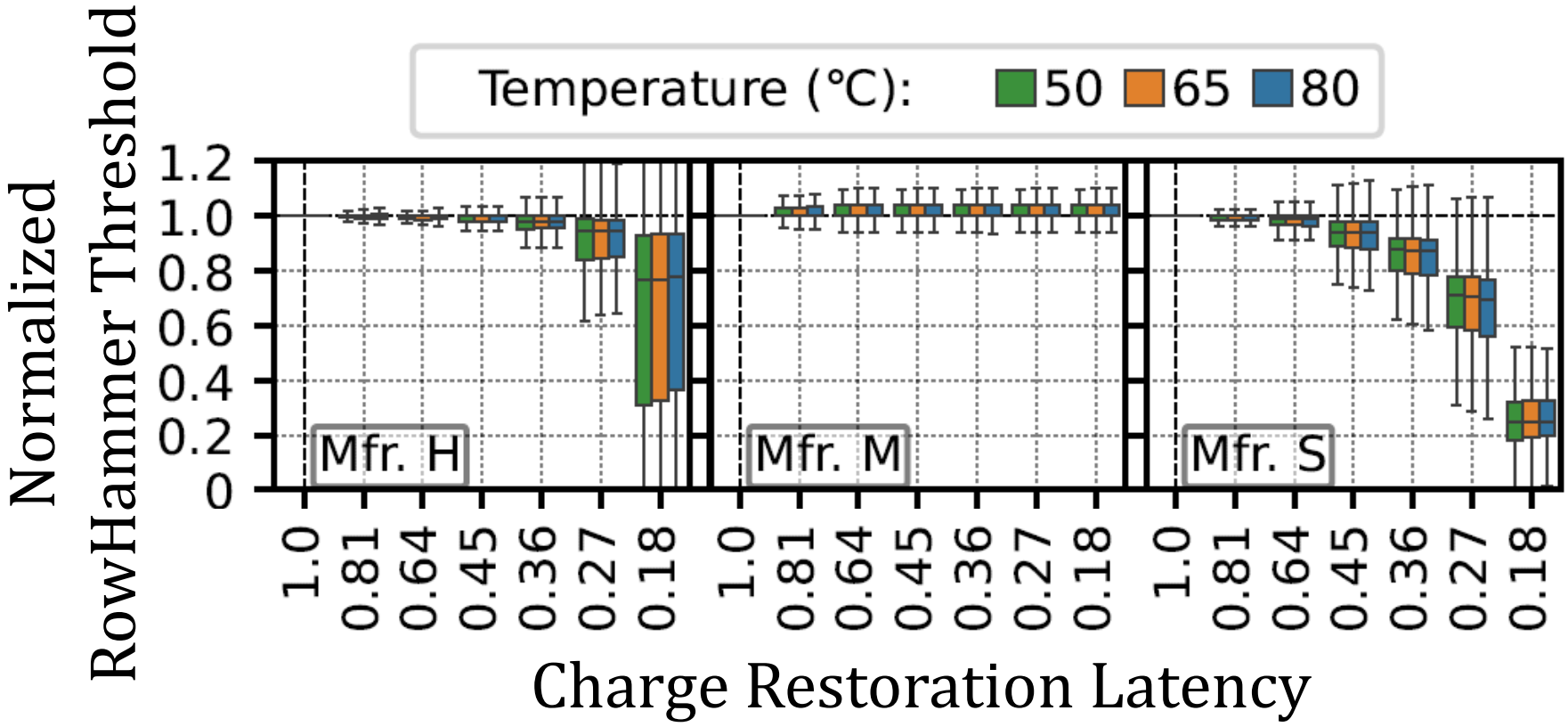

Rows that are highly vulnerable with full charge restoration **do not exhibit the largest $N_{RH}$ reductions** with partial charge restoration

# Effect of Partial Charge Restoration on RowHammer Bit-Error-Rate (BER)



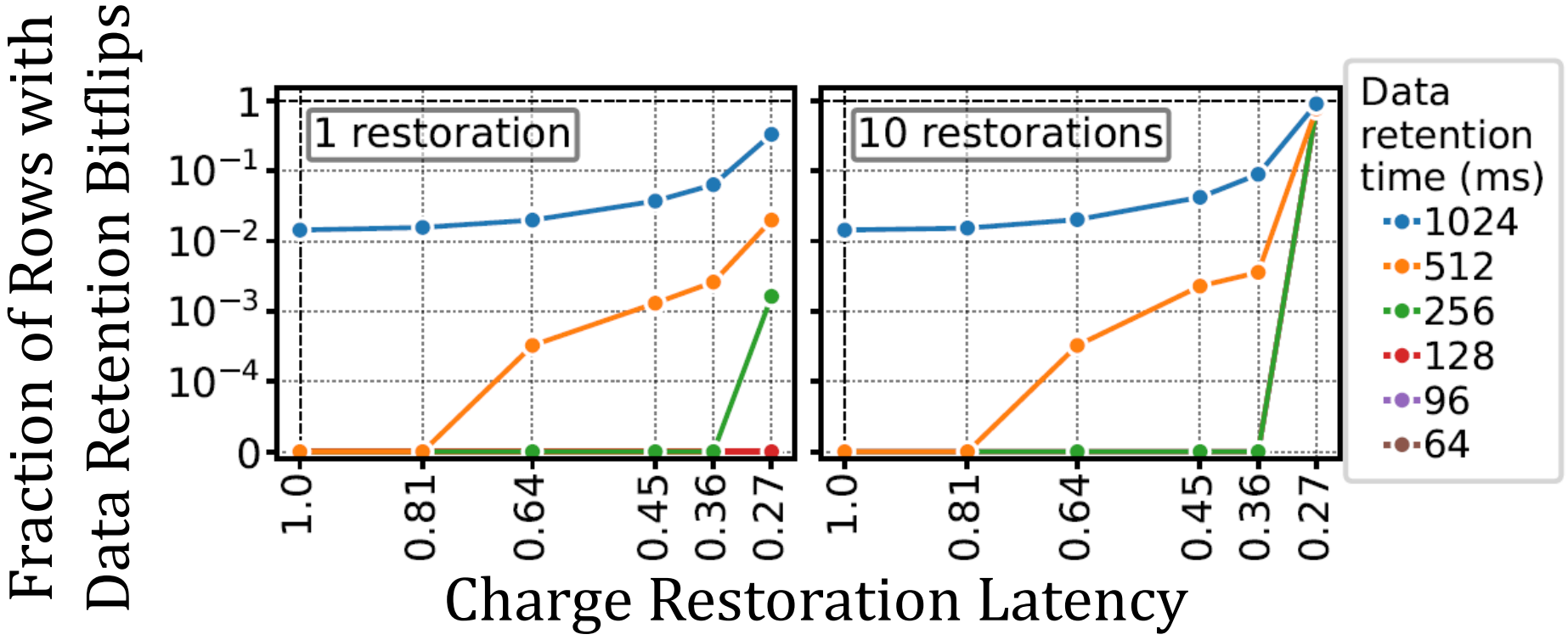

Charge restoration latency
can **be reduced without significantly affecting RowHammer Bit-Error-Rate**

# Combined Effect of Partial Charge Restoration and Temperature



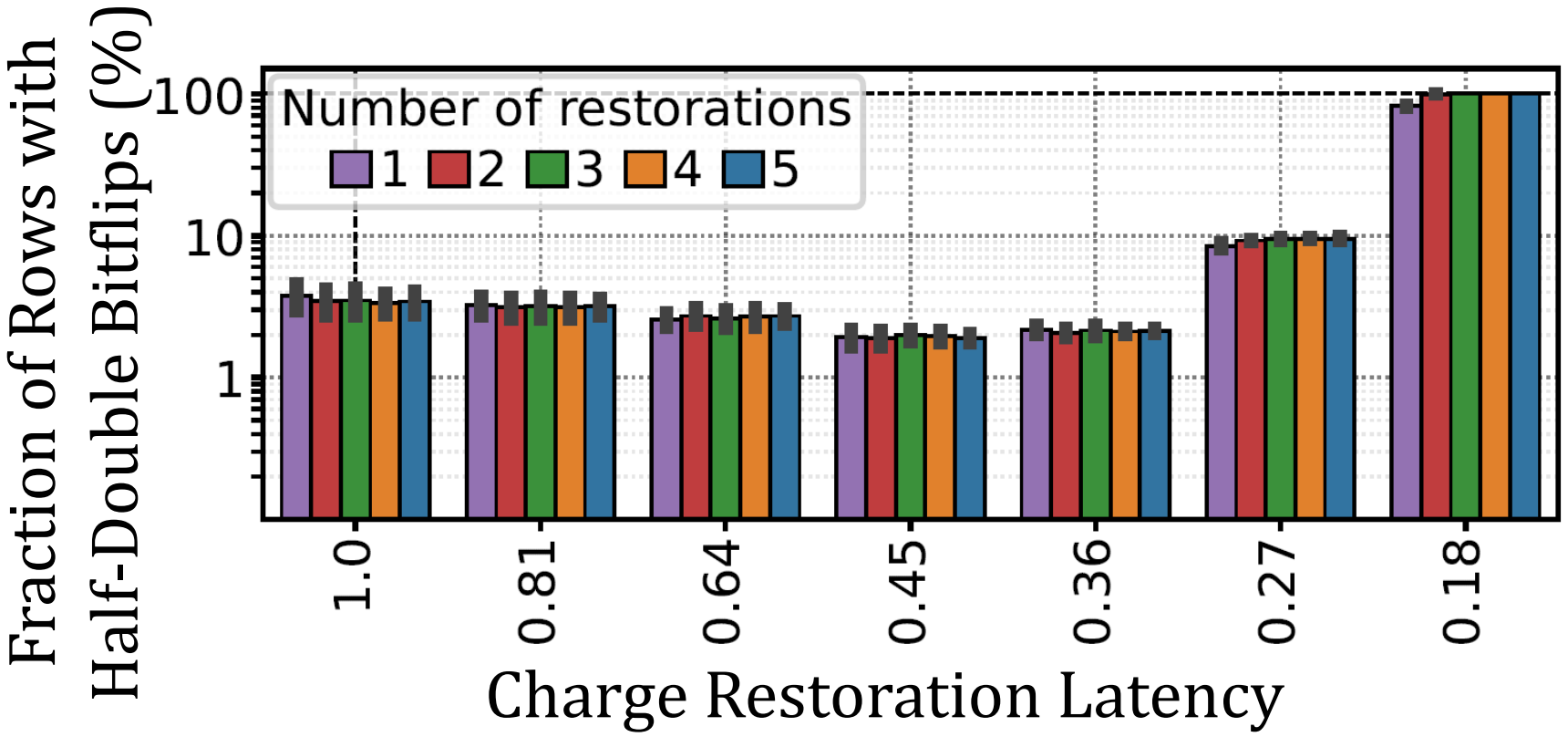

**No significant impact of temperature**
on the effect of charge restoration latency
on RowHammer vulnerability

# Effect of Partial Charge Restoration on Data Retention Time



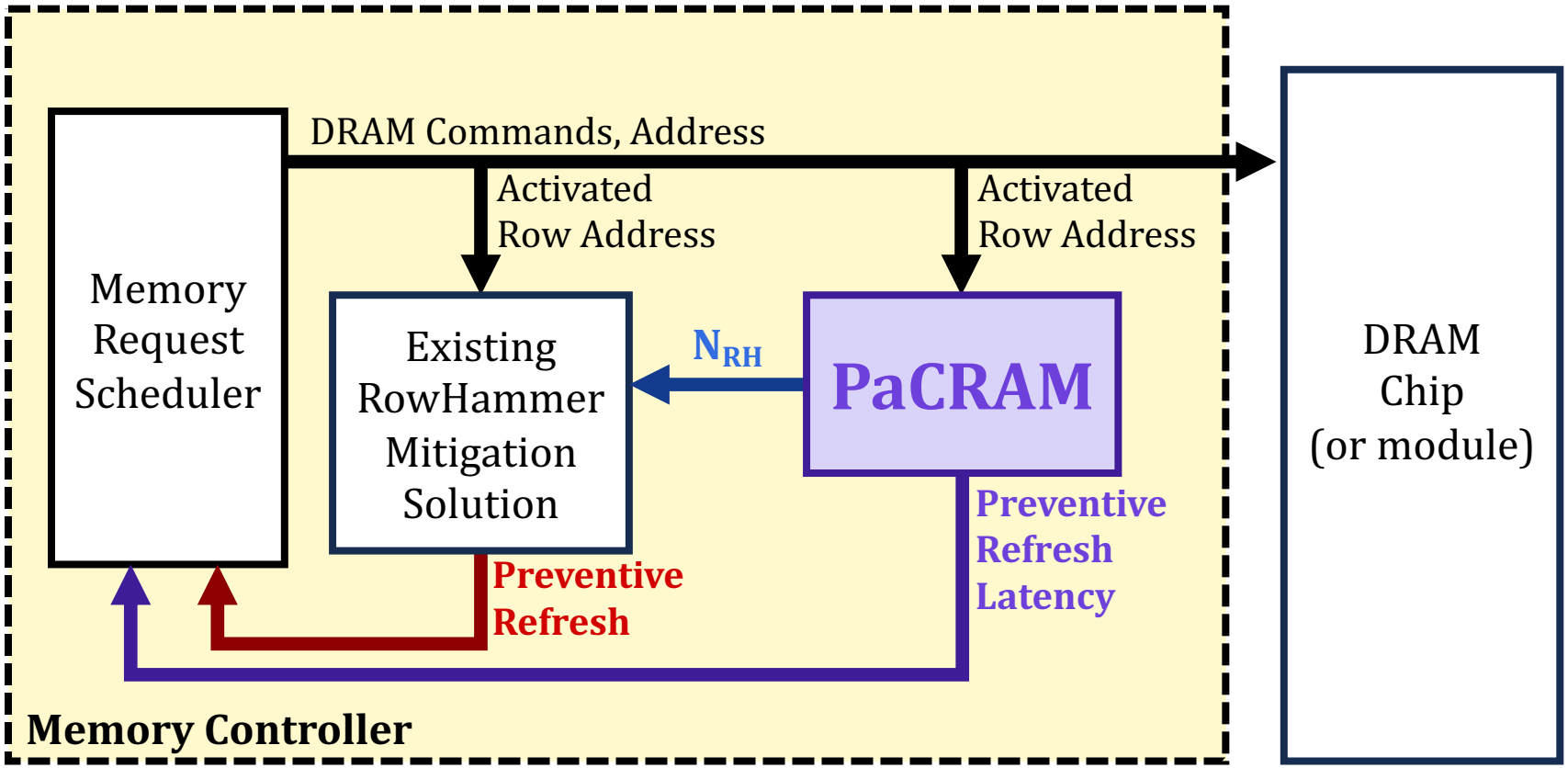

Charge restoration latency can **be significantly reduced** to a safe minimum value **without causing data retention failures**

# Effect of Partial Charge Restoration on Half-Double Access Pattern

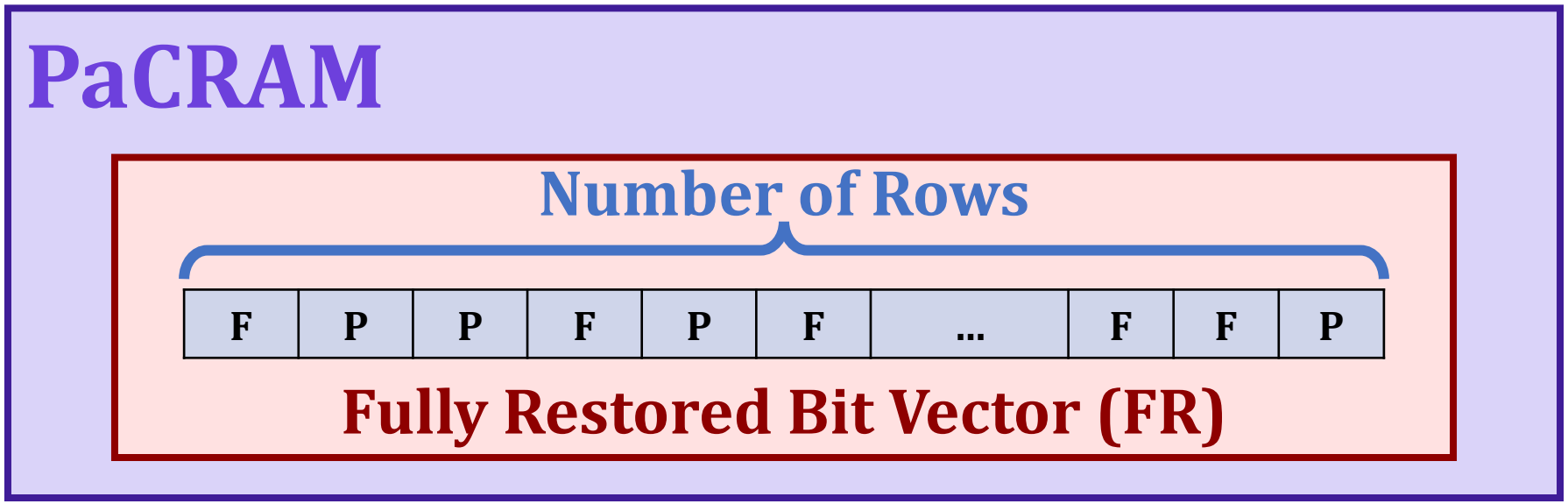



Charge restoration latency can **be significantly reduced**

# Detailed Implementation of PaCRAM

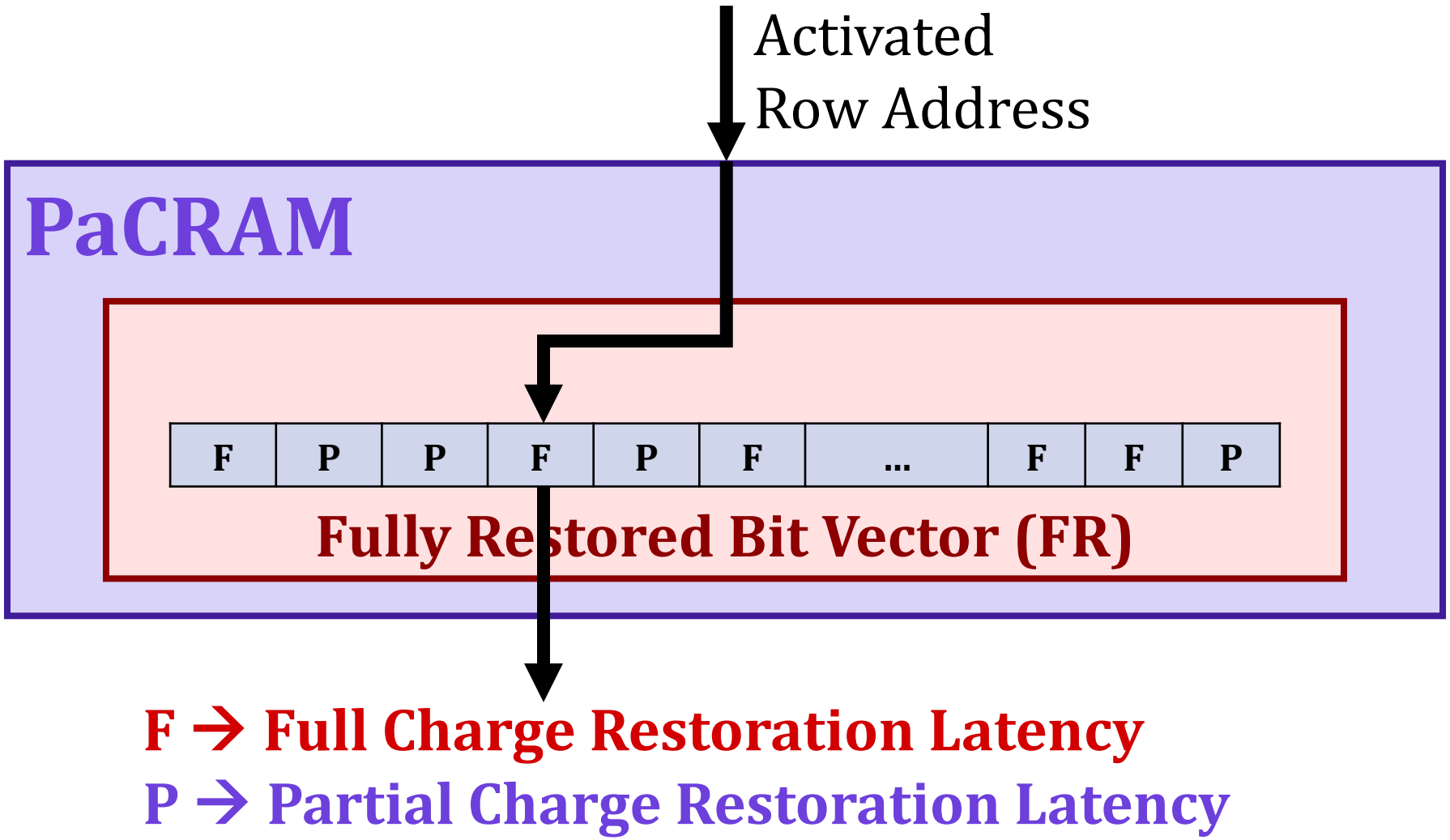


# Detailed Implementation of PaCRAM




- **F State**: the row has to be refreshed using **full charge restoration**
- **P State**: the row can be refreshed using **partial charge restoration**

# Determining Preventive Refresh Latency



# PaCRAM's Row State Transition

- **F State → P State**
  - When the row is refreshed using full charge restoration
  - After fully restoring a row, PaCRAM uses partial charge restoration

- **P State → F State**
  - PaCRAM periodically resets each rows state to F state
  - **Full Charge Restoration Interval**

# Full Charge Restoration Interval

- **To Guarantee** that any victim is **not refreshed** by more than a safe number ($N_{PCR}$) of **consecutive partial charge restorations**

- PaCRAM assumes **the worst case** where a DRAM row is accessed as frequently as possible
- The **smallest time window** that can contain $N_{PCR}$ **preventive refreshes**

**Time to Receive One Preventive Refresh**

†

**Attack Time** **Preventive Refresh Latency**

# Hardware Complexity Analysis of PaCRAM

- **Chip area** and **access latency** analysis using CACTI
- PaCRAM requires **1 bit** for each DRAM row (Fully Restored Bit Vector)
- For a DRAM module with 32 banks and 64K row/bank,
  - **256 KB of SRAM**
  - **0.09%** of a high-end Intel Xeon processor
  - **1.35%** of the memory controller area
  - Access latency of **0.27ns** (<< row activation latency)

**PaCRAM** introduces **a small additional area overhead**

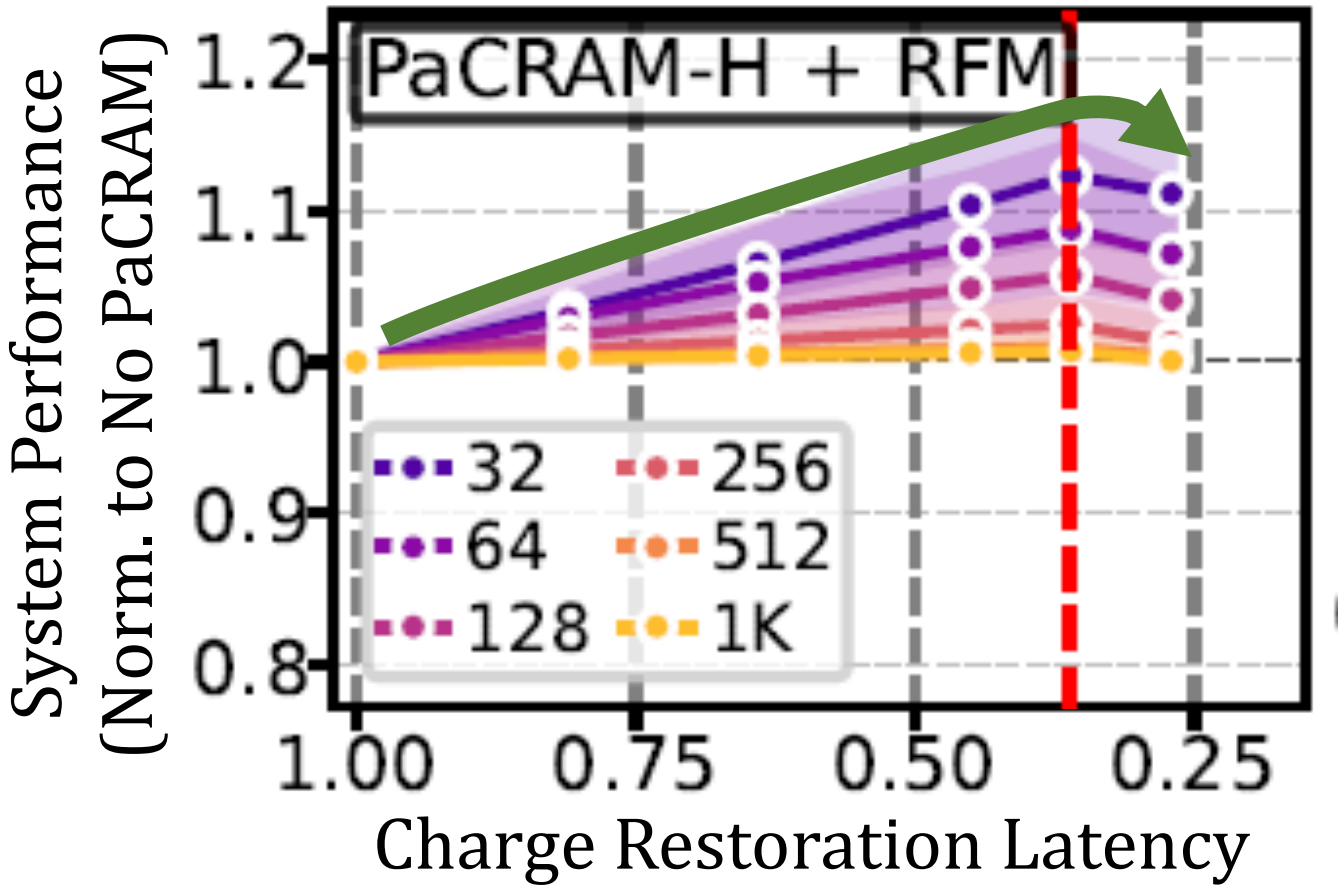
# PaCRAM with on-DRAM-die RowHammer Mitigation Mechanisms

- **On-DRAM-die mitigation mechanism**
  - **PRAC**
  - **Chronus**
  - **…**
- PaCRAM can be implemented **inside DRAM chip**
  - On-DRAM-die mitigation mechanism **inserts a Back-Off signal** to request a preventive refresh
  - PaCRAM stores **the preventive refresh latency in the mode register (MR)**
  - Memory controller checks the latency value in MR and issues **a preventive refresh with the latency provided by PaCRAM**

# Profiling Cost Analysis

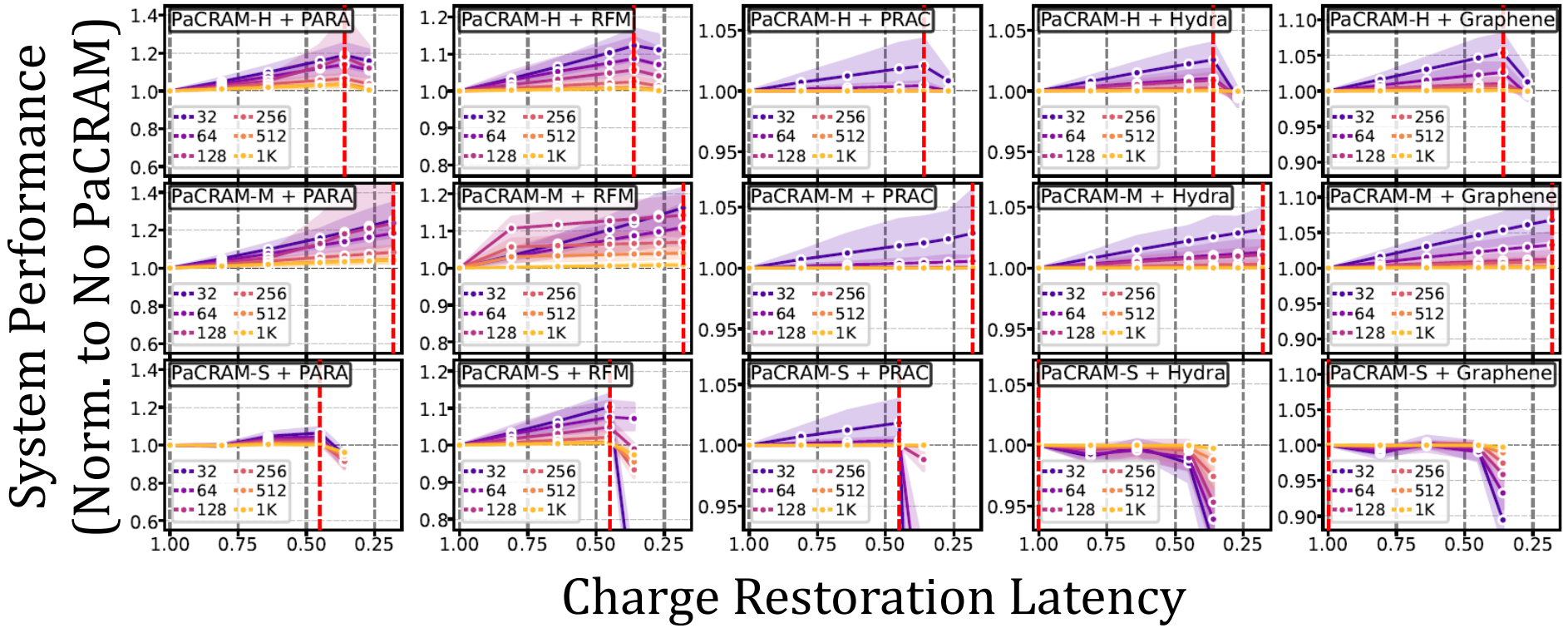
- For PaCRAM to work **robustly**,
  - How much charge restoration latency can be reduced
  - How RowHammer threshold changes with partial charge restoration
  - How many partial charge restorations we can perform consecutively
- **Profiling**
  - The system perform profiling the very first time DRAM is initialized
  - DRAM manufacturers perform profiling and store metadata in SPD
  - The system can perform online profiling

# Effect of Charge Restoration Latency on System Performance

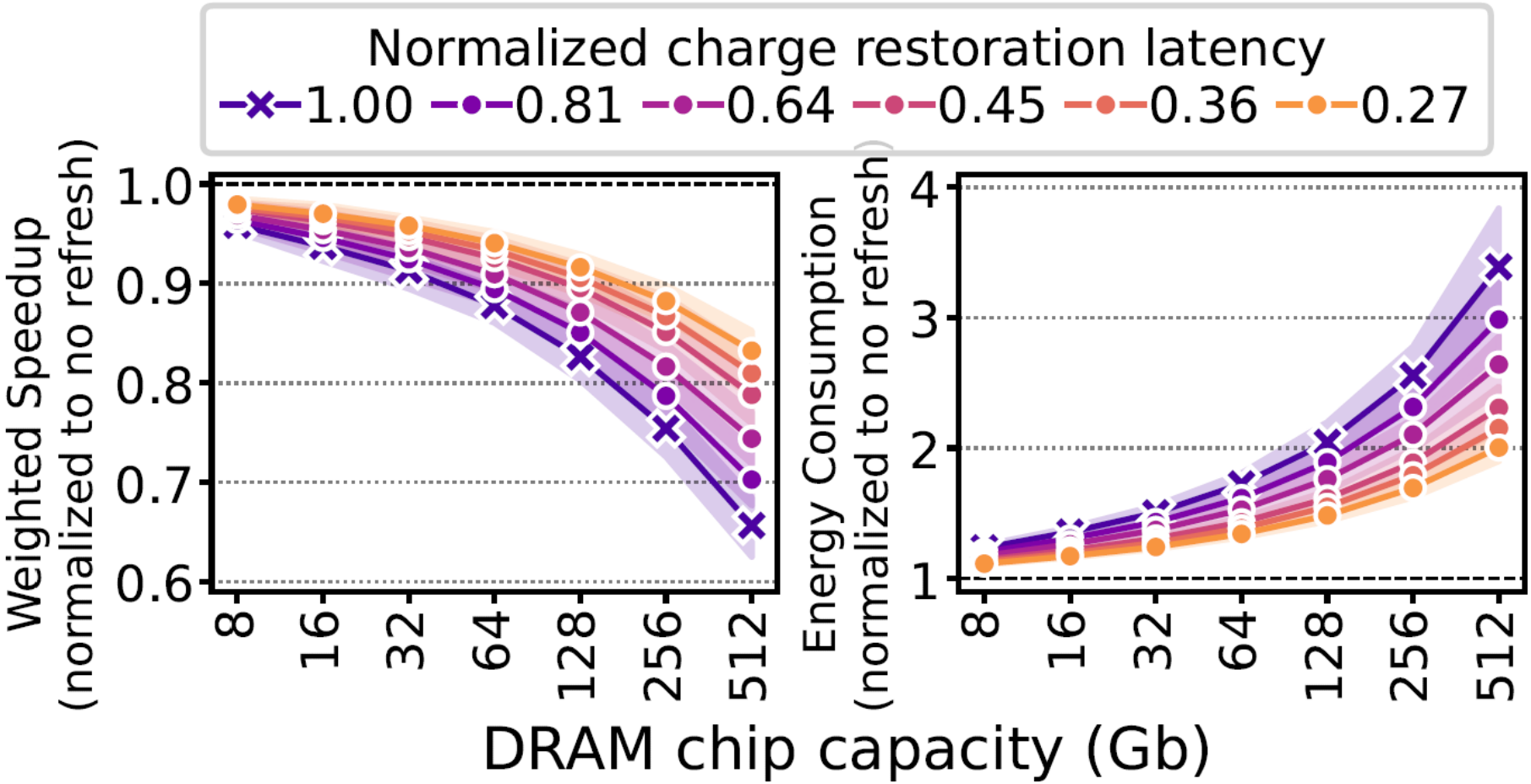



System performance **first increases**, **then decreases** as charge restoration latency decreases

# Effect of Charge Restoration Latency on System Performance



# Reducing Periodic Refresh Latency




PaCRAM improves the system performance and energy efficiency **by reducing periodic refresh overhead**

# Understanding RowHammer Under Reduced Refresh Latency: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions


**Yahya Can Tuğrul**

A. Giray Yağlıkçı  İsmail Emir Yüksel

Ataberk Olgun  Oğuzhan Canpolat  Nisa Bostancı

Mohammad Sadrosadati  Oğuz Ergin  Onur Mutlu

SAFARI

ETH zürich

kasırga

TOBB ETÜ

University of Economics & Technology