HammerFilter: Robust Protection and Low Hardware Overhead Method for Row-Hammering

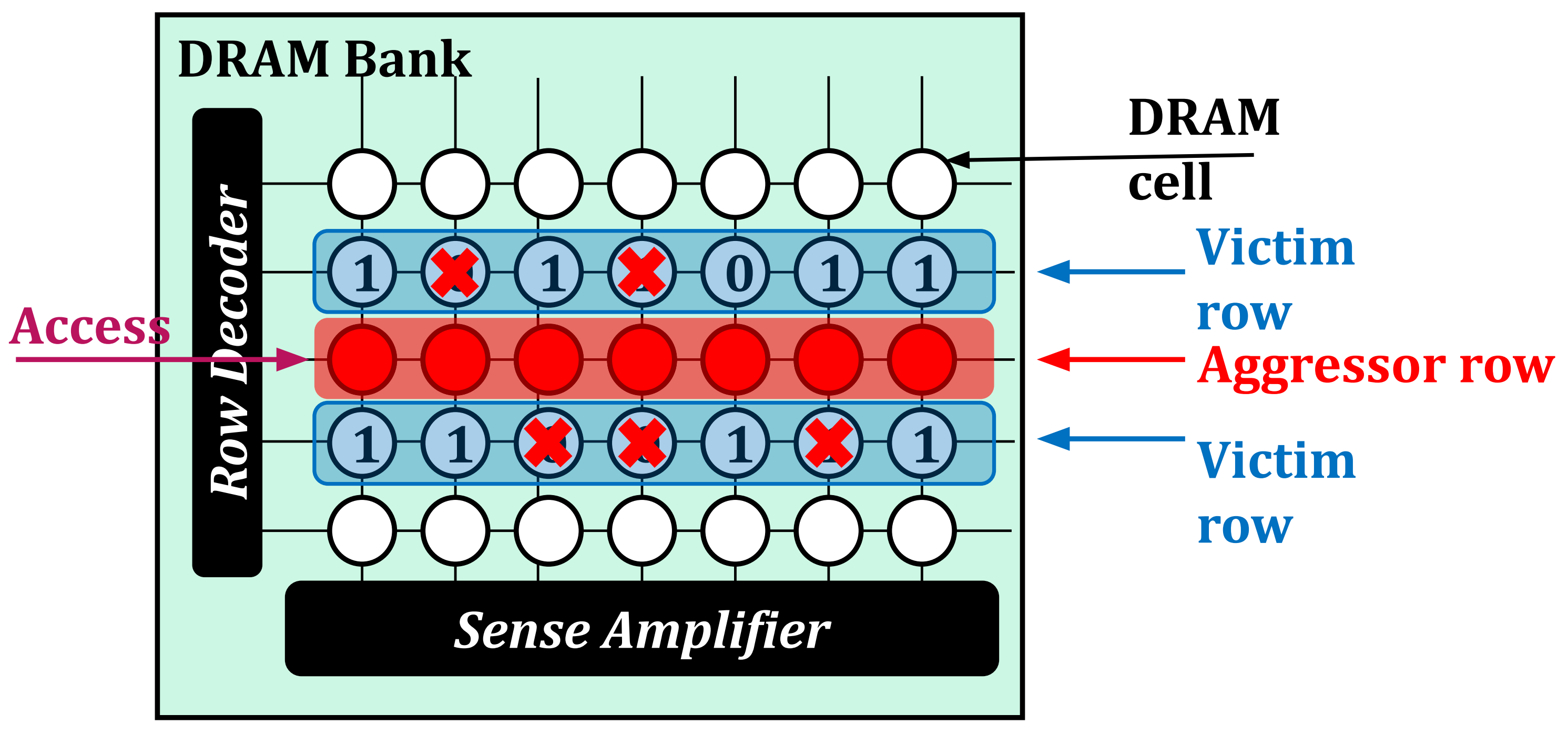
Kwangrae Kim
Department of ECE
Hanyang University

Junsu Kim
Department of ECE
Hanyang University

Jeonghyun Woo
Department of ECE
University of Illinois at Urbana-Champaign

Ki-Seok Chung
Department of ECE
Hanyang University

1) Background: Row-hammering



Repeatedly Access to a DRAM row (aggressor row) causes bit-flips in nearby rows (victim row)

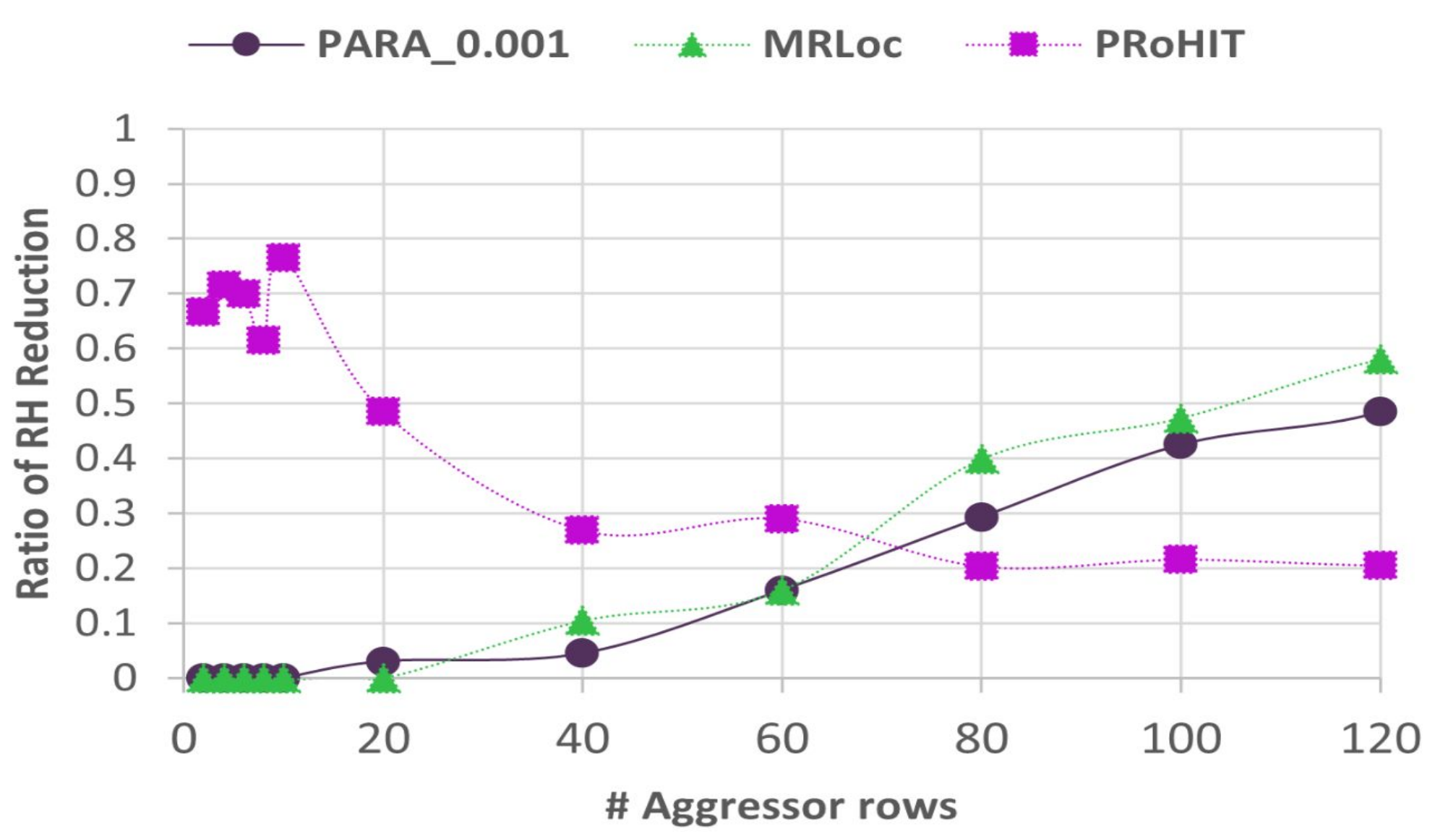
3) Contribution

1. We propose a novel data structure, named HammerFilter, which provides guaranteed protection with a low area cost against row-hammering attack
2. We evaluated the proposed method with intricate row-hammering attack patterns (five patterns)
3. HammerFilter incurs minimal performance overhead in the benign applications

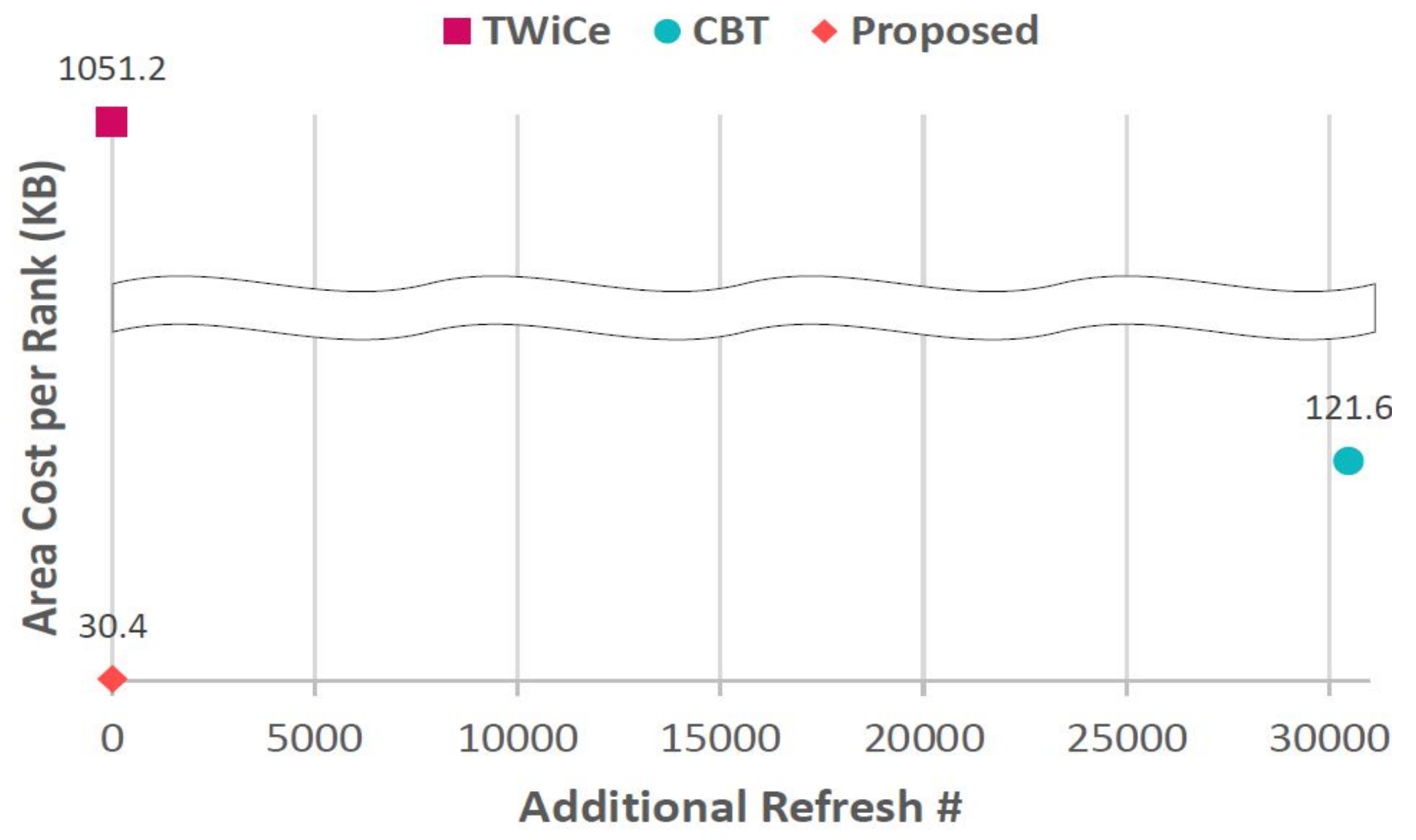
2) Motivation

- The continuous scaling-down of the DRAM process makes DRAM cells more vulnerable to row-hammering
- There are two types of hardware-based protection schemes for row-hammering attack: a probabilistic method and a counter-based method

1) The probabilistic schemes have poor protection against complex row-hammering attacks
   - PARA[Kim+, ISCA'14], MRLoc[You+, DAC'19], and PRoHIT[Son+, DAC'17] still cannot prevent well in complex row-hammering attacks
2) The counter-based schemes guarantee strong protection, but they suffer from significant area overhead or extreme additional refreshes or even both
   - CBT[Seyedzadeh+, ISCA'18] carries out considerable additional refreshes to prevent row-hammering attacks
   - TWiCe[Lee+, ISCA'19] incurs significant area overhead to operate it

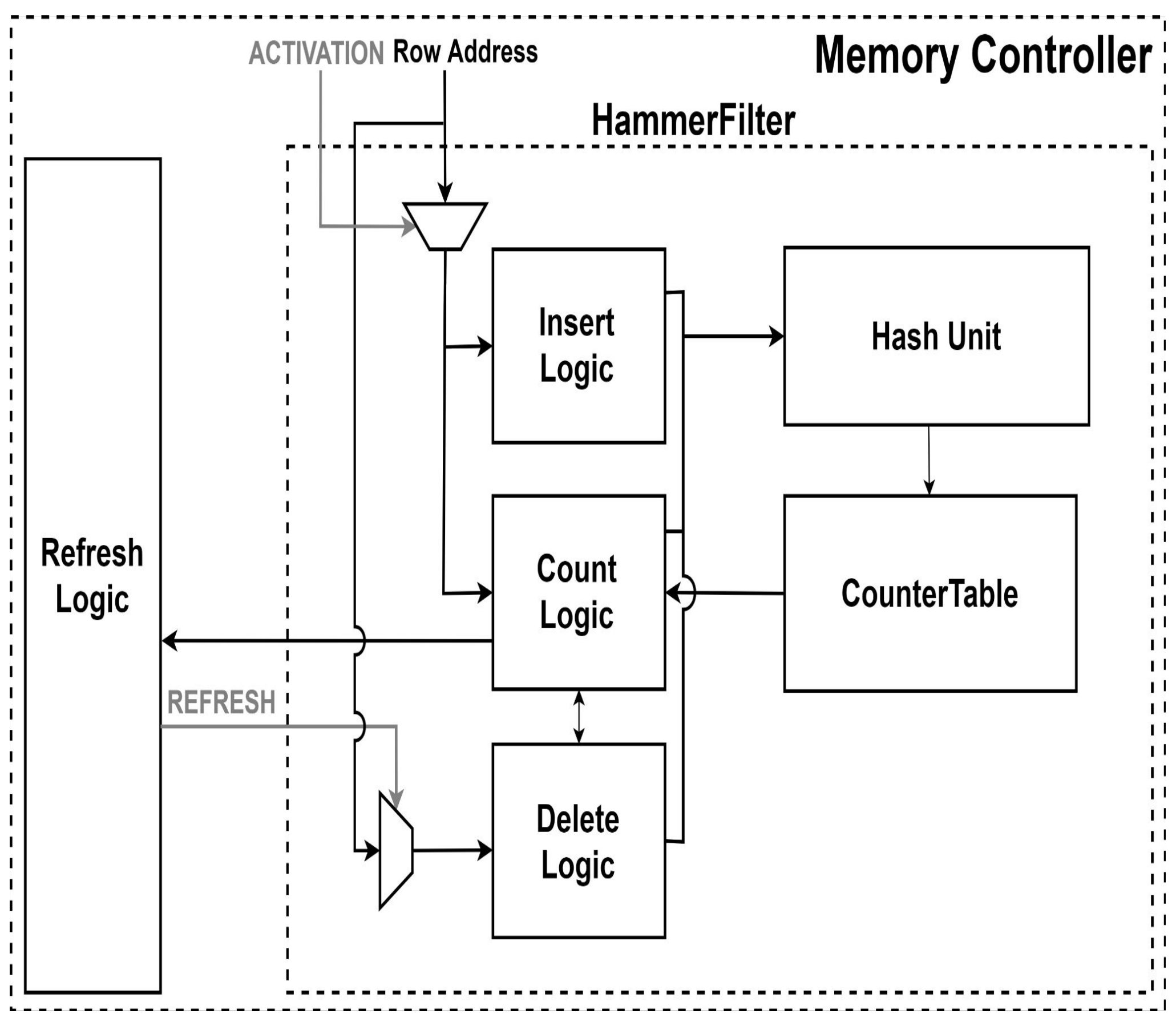


Row-hammering Reduction of Probabilistic Models



Overhead analysis of Counter-based Models

4) HammerFilter Design Overview

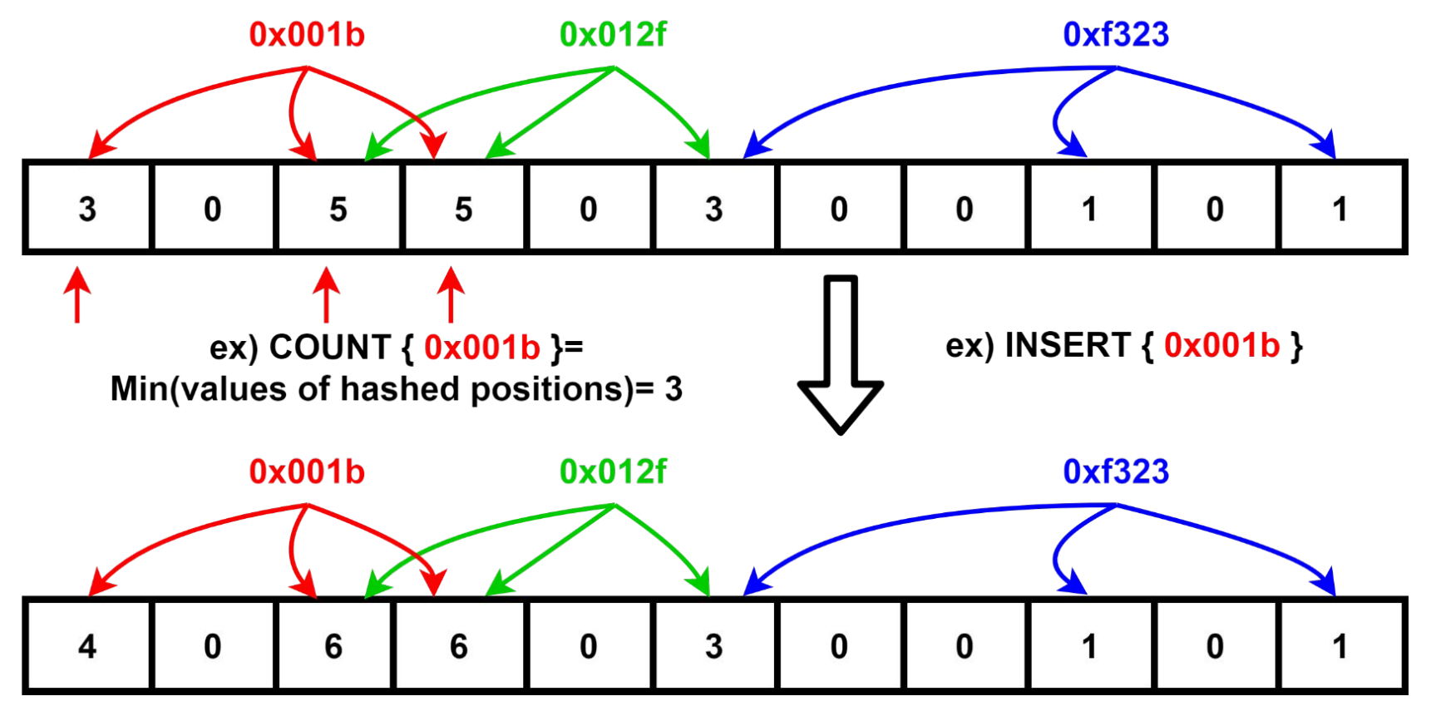


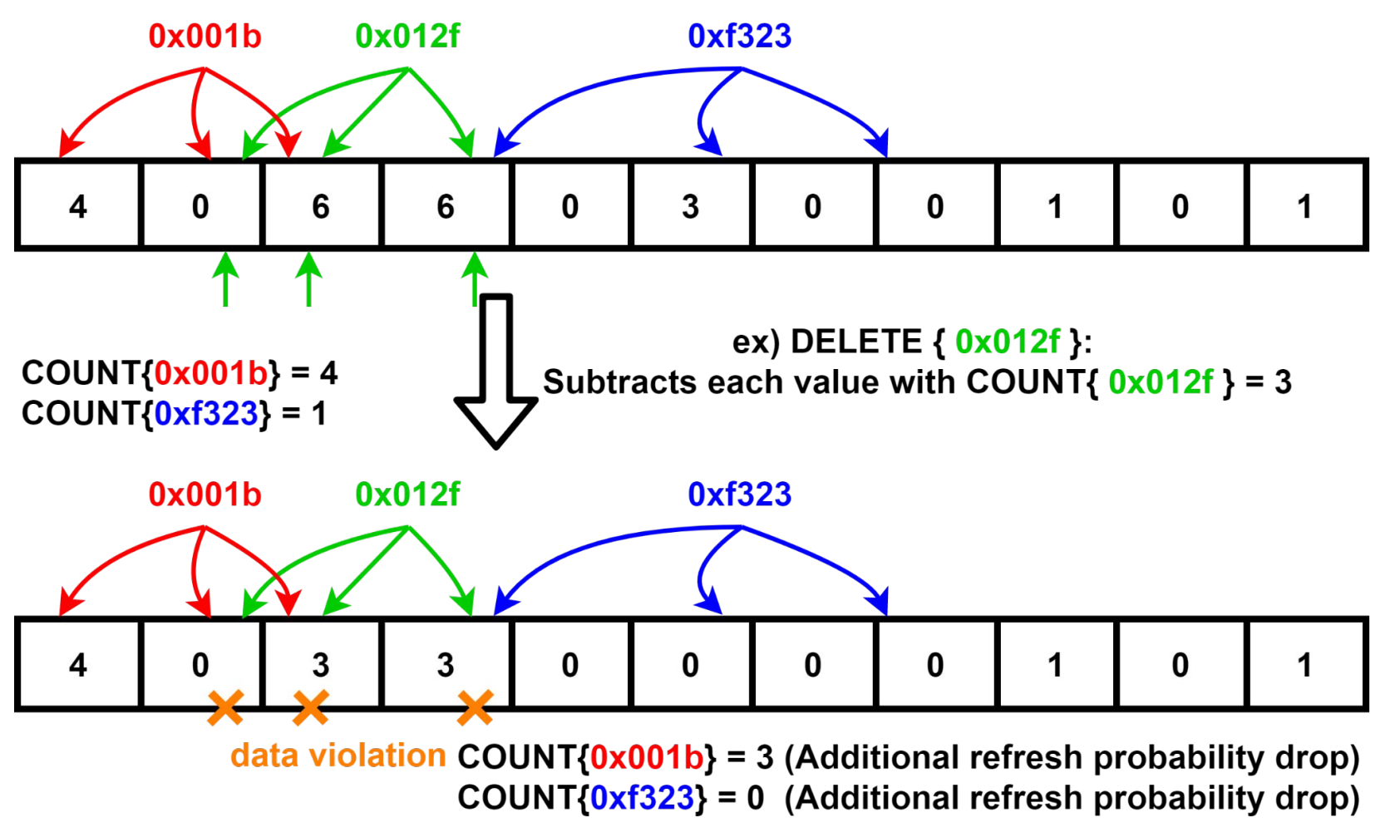
a) Architecture

- Insert Logic: updates aggressor row candidates with predetermined probability, p_i (0.004)
- Delete Logic: reduces corresponding Counter Table's value when additional refresh is done
- Count Logic: provides the extent of danger of accessed rows to the Refresh Logic

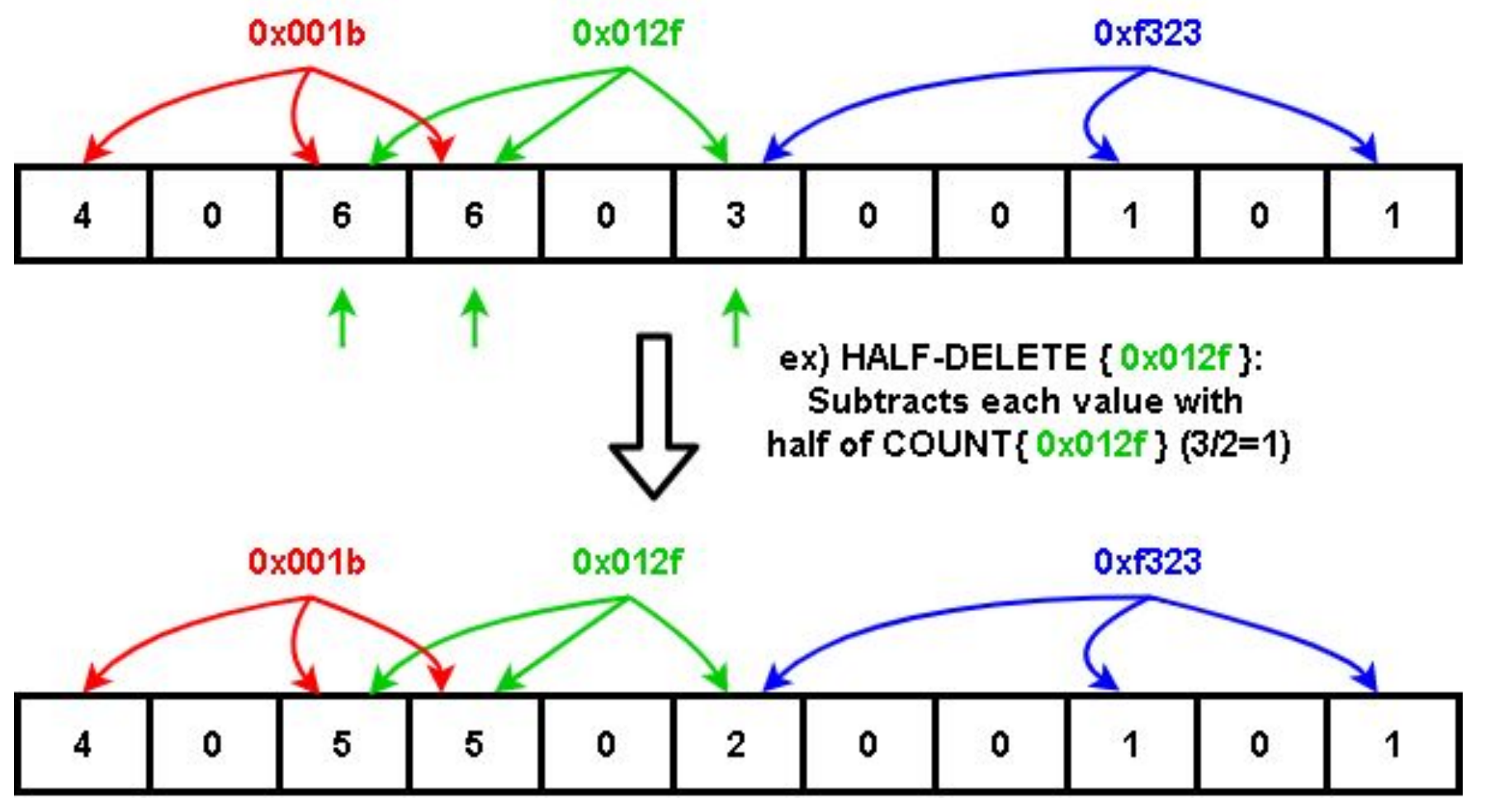
b) Description of Operations

- COUNT: considers the minimum of the values in the hashed positions as the COUNT value
- INSERT: increases the values of the hashed positions by one
- HALF-DELETE: subtracts each value in the hashed positions by one-half of the COUNT value. It is to avoid a hash-induced collision



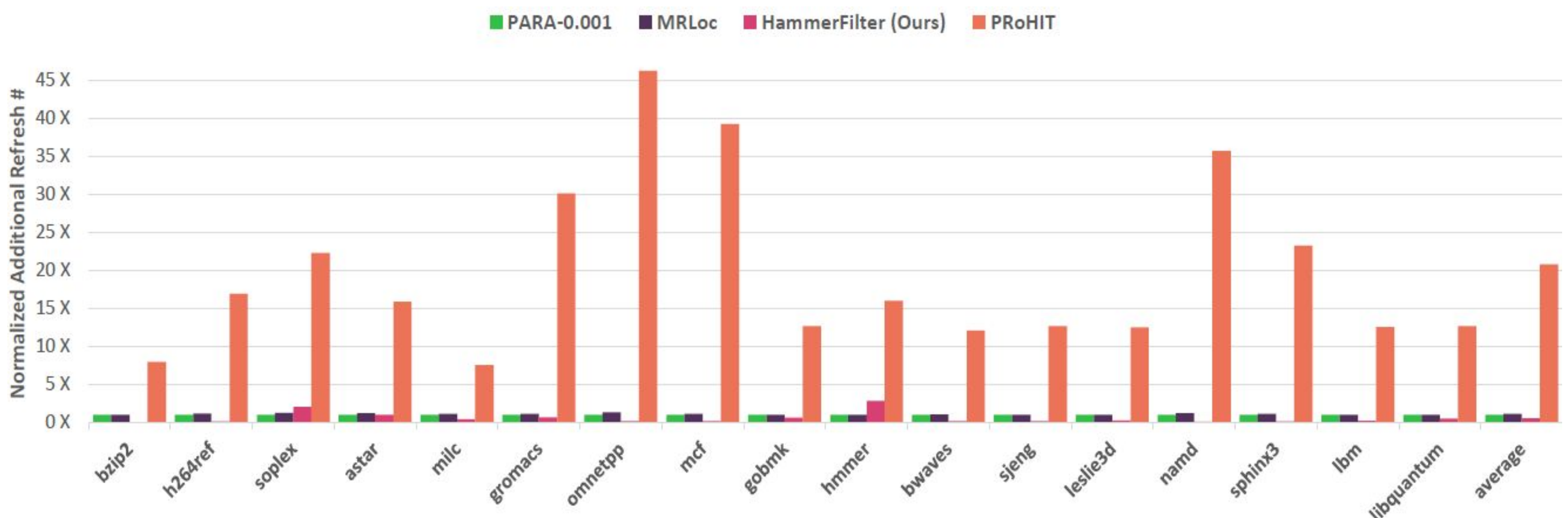


0x001b
0x012f
0xf323
4
0
6
6
0
3
0
0
1
0
1
COUNT{0x001b} = 4
COUNT{0xf323} = 1
ex) DELETE { 0x012f }: Subtracts each value with COUNT{ 0x012f } = 3
0x001b
0x012f
0xf323
4
0
3
3
0
0
0
0
1
0
1
data violation
COUNT{0x001b} = 3 (Additional refresh probability drop)
COUNT{0xf323} = 0 (Additional refresh probability drop)



0x001b
0x012f
0xf323
4
0
6
6
0
3
0
0
1
0
1
ex) HALF-DELETE { 0x012f }: Subtracts each value with half of COUNT{ 0x012f } (3/2=1)
0x001b
0x012f
0xf323
4
0
5
5
0
2
0
0
1
0
1

5) Experimental Results



Evaluation in Benign Applications

Description of
Row-hammering Attack Patterns

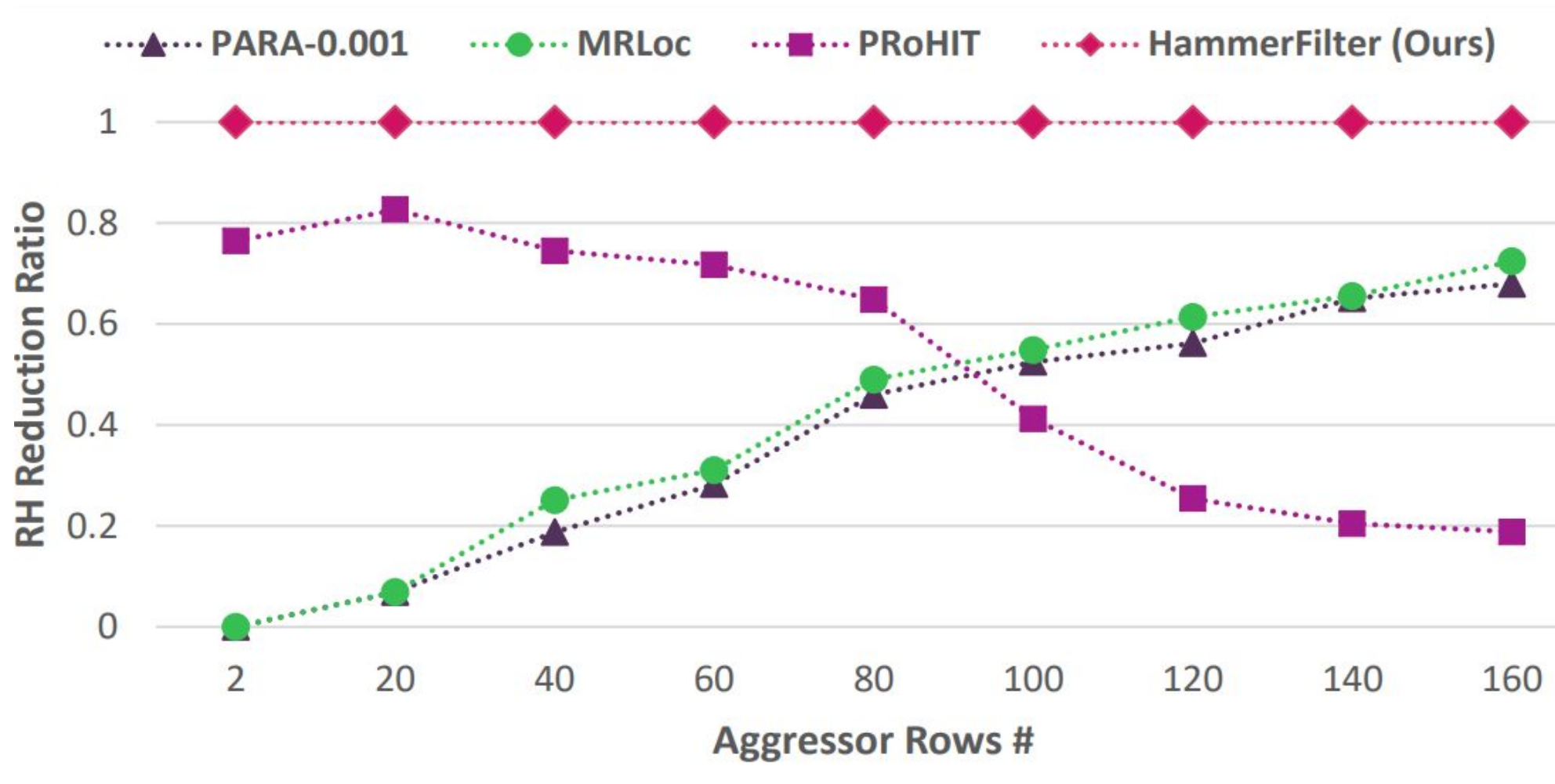
Type	Description	Access Pattern
1	Repeated Selected Rows	$(a_1, a_2, \cdots, a_N)^*$
2	Repeated Selected Rows + Random Rows	$a_1, 1315, a_2, 798, \cdots, a_N, 37, \cdots$
3	Double-sided Rows	$(a_1-1, a_1+1, \cdots, a_N-1, a_N+1)^*$
4	Double-sided Rows + Random Rows	$a_1-1, 927, a_1+1, \cdots, 109, a_N+1, \cdots$
5	Double-sided Rows + Repeated Selected Rows	$(a_1-1, b_1, a_1+1, \cdots, b_N, a_N+1)^*$

a) Evaluation in Benign Applications

- The average amount of additional refreshes in HammerFilter is 2.09X and 38.87X less than that of MRLoc and PRoHIT, respectively

b) Evaluation in Malicious Applications

- We evaluates HammerFilter with five artificial pattern of row-hammering attack
- HammerFilter achieves overwhelmingly better results with respect to all the numbers of aggressor rows than other schemes



Evaluation in Malicious Applications

c) HammerFilter's Mechanism

1) HammerFilter operates INSERT an accessed row to the CounterTable with a fixed probability, p_i.
2) It sends COUNT value to the Refresh Logic for every access
3) Refresh Logic sends additional refresh with the probability (p_r) according to the COUNT value to the victim rows

$$\begin{cases} p_r = 1 \div 2^{8-count} & if\ count > 3 \\ p_r = 0 & else \end{cases}$$

6) Conclusion

HammerFilter is a novel method to mitigate row-hammering

- Prevents all the row-hammering attacks that have various patterns and access multiple rows.
- Incurs minimal performance overhead in benign applications
- Extremely area efficient compared to counter-based methods