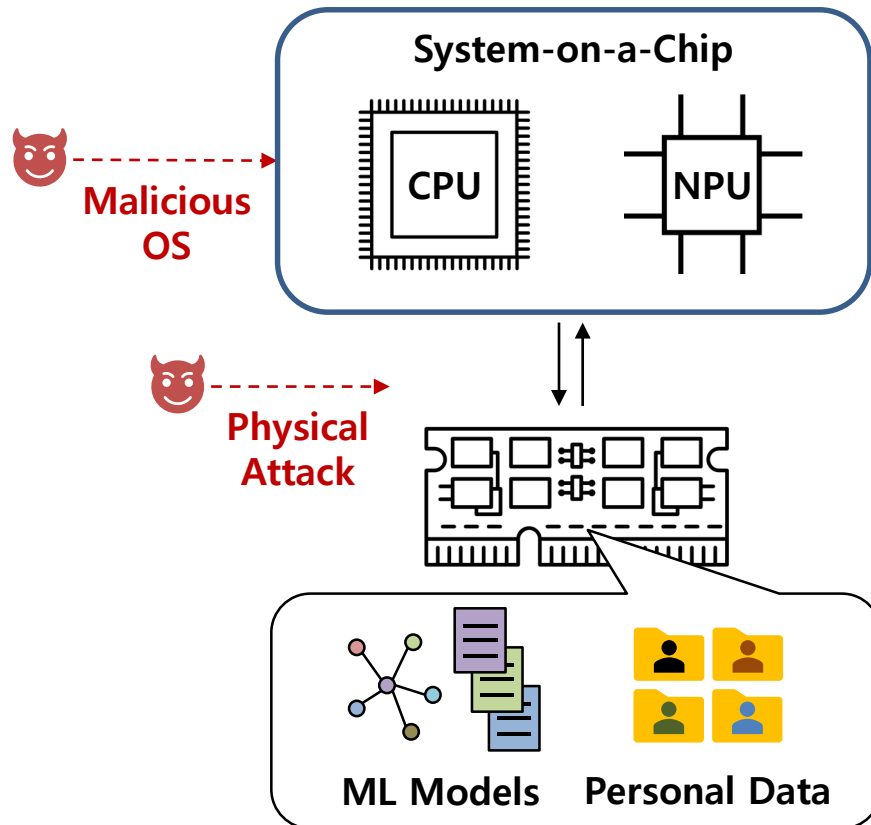# TNPU: Supporting Trusted Execution with Tree-less Integrity Protection for Neural Processing Unit

Sunho Lee, Jungwoo Kim, Seonjin Na,
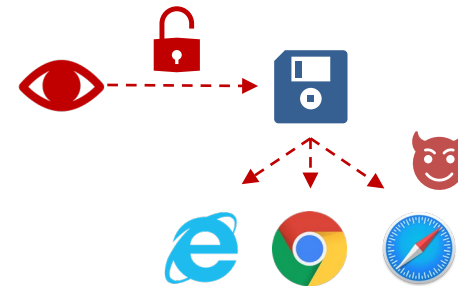Jongse Park, and Jaehyuk Huh

KAIST
School of **Computing**
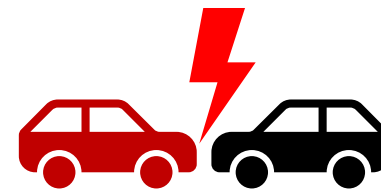
CASYS
Computer Architecture and Systems Lab

Vulnerabilities of integrated NPU

- NPU is widely used in the form of <u>System-on-a-Chip</u>.
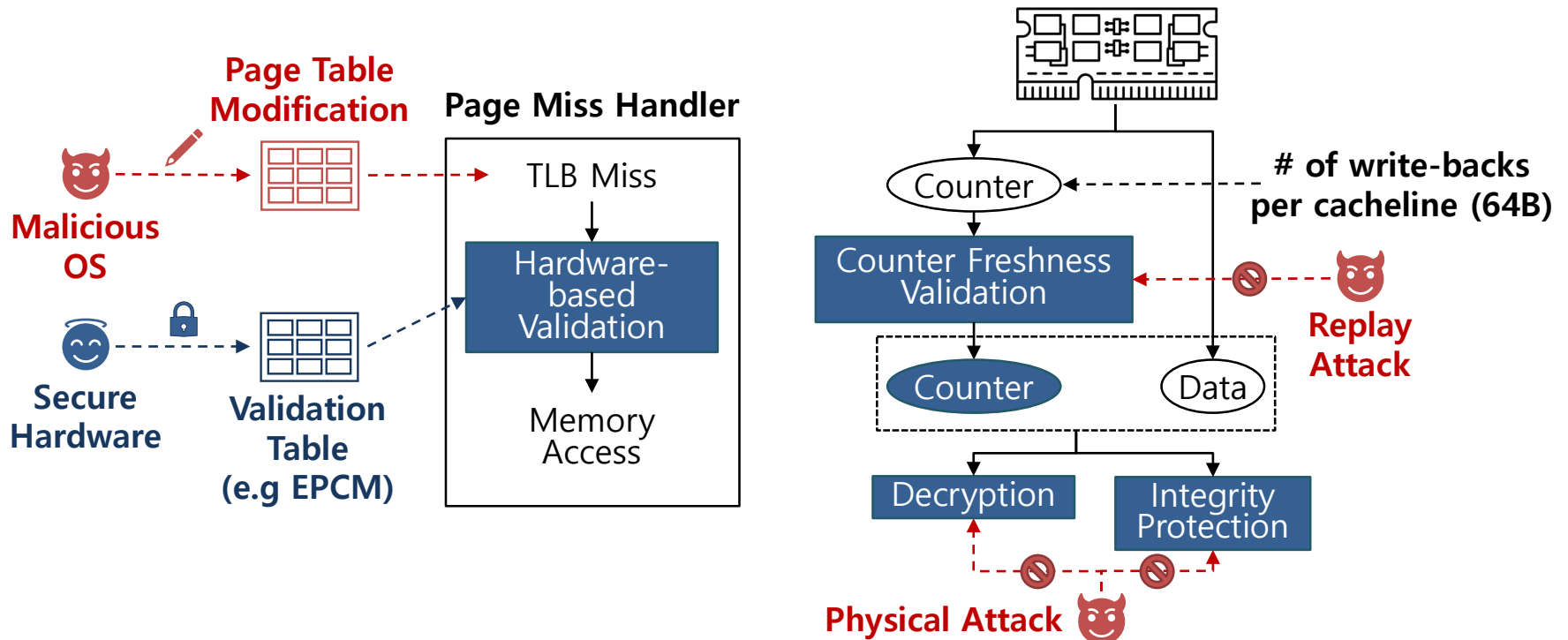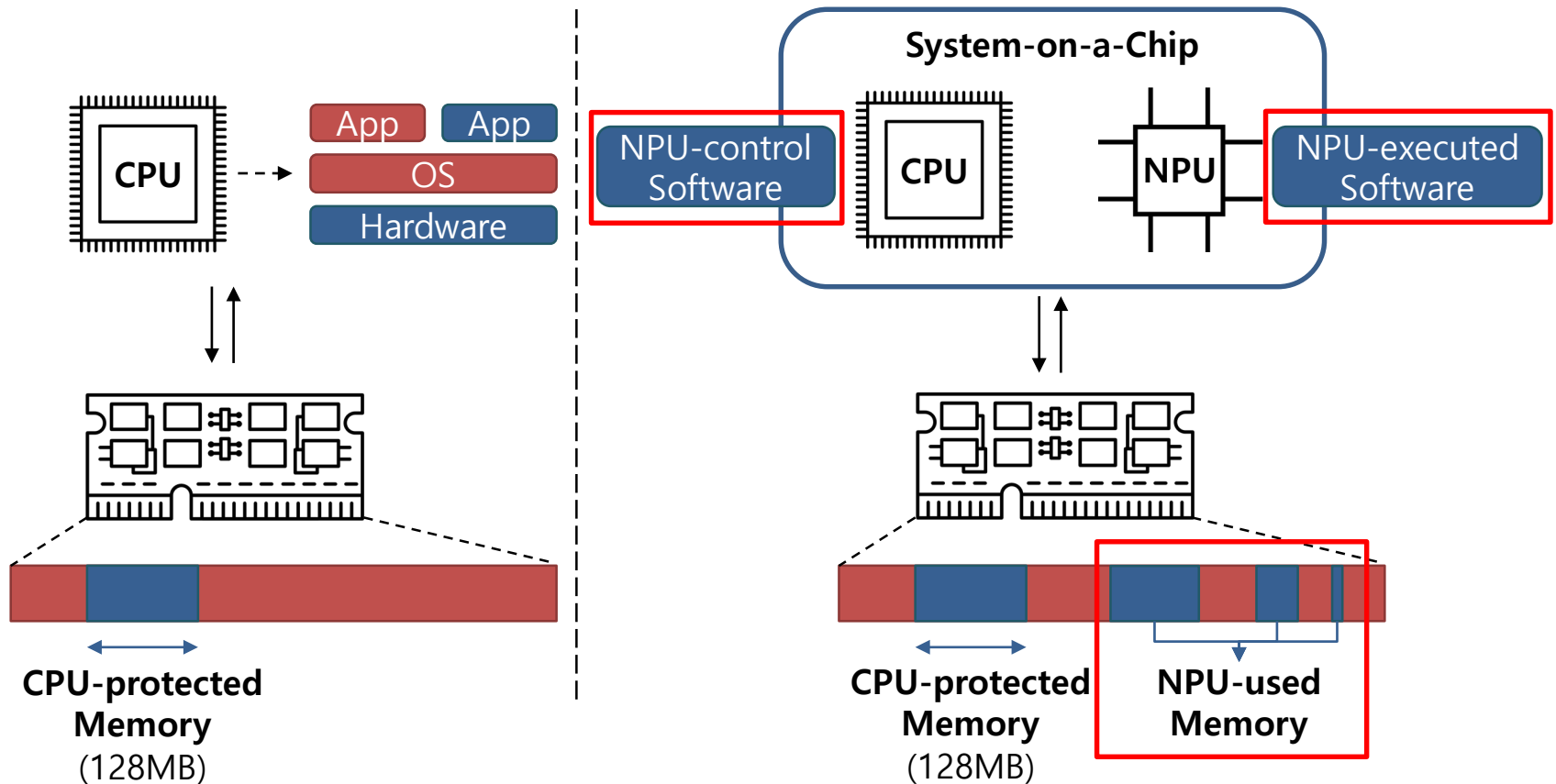
Trusted Execution Environment (CPU)

- Access control

- Counter-based memory protection

Trusted Execution Environment (NPU)

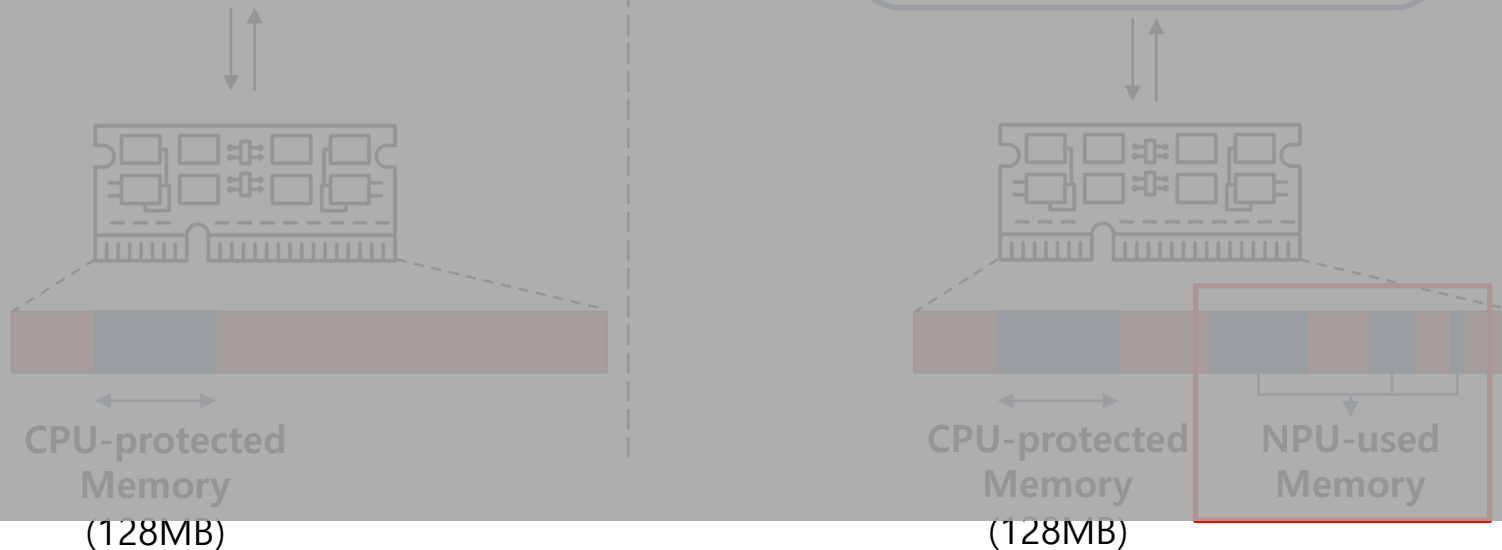- CPU: On-chip hardware and related software

- TNPU: + NPU-related hardware/software

Trusted Execution Environment (NPU)

- CPU: On-chip hardware and related software

- TNPU: + NPU-related hardware/software

System-on-a-Chip

App   App

NPU-control   NPU-executed

1) Access control, 2) Memory Protection for NPU

CPU-protected
Memory
(128MB)

CPU-protected
Memory
(128MB)

NPU-used
Memory

Validate Access from NPU-MMU

- Access control
  - CPU MMU: Traditional validation table
  - NPU IOMMU

Page Miss Handler

System-on-a-Chip

TLB Miss

Hardware-based Validation

Memory Access

Validation Table (CPU-side)

CPU

NPU

Not Support
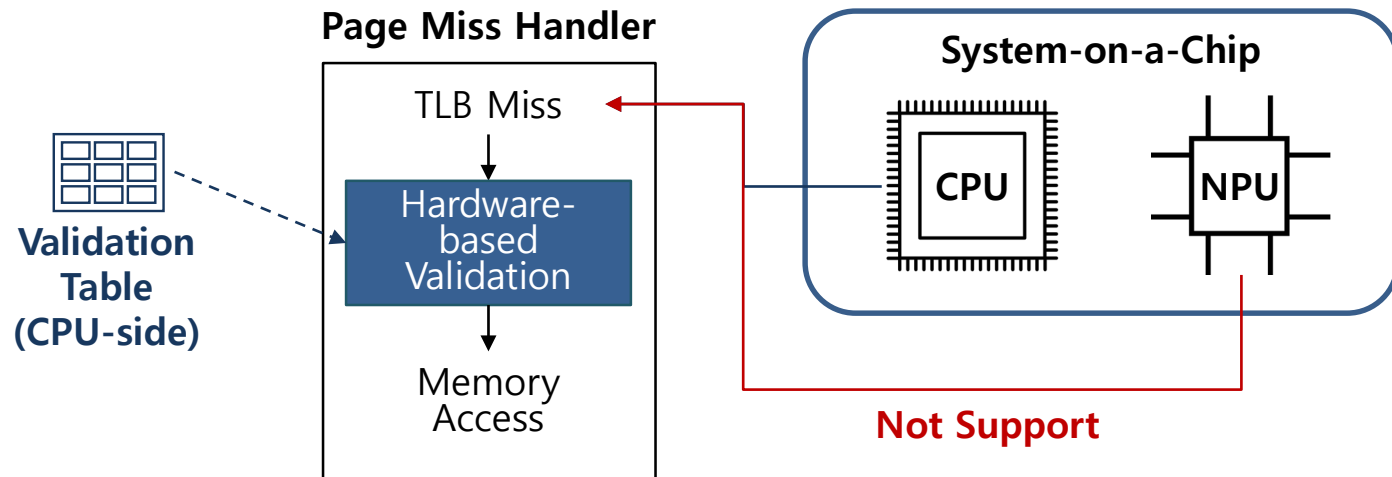
Validate Access from NPU

- Access control: **Extended validation table (EEPCM)**

  - CPU MMU: Traditional validation entries
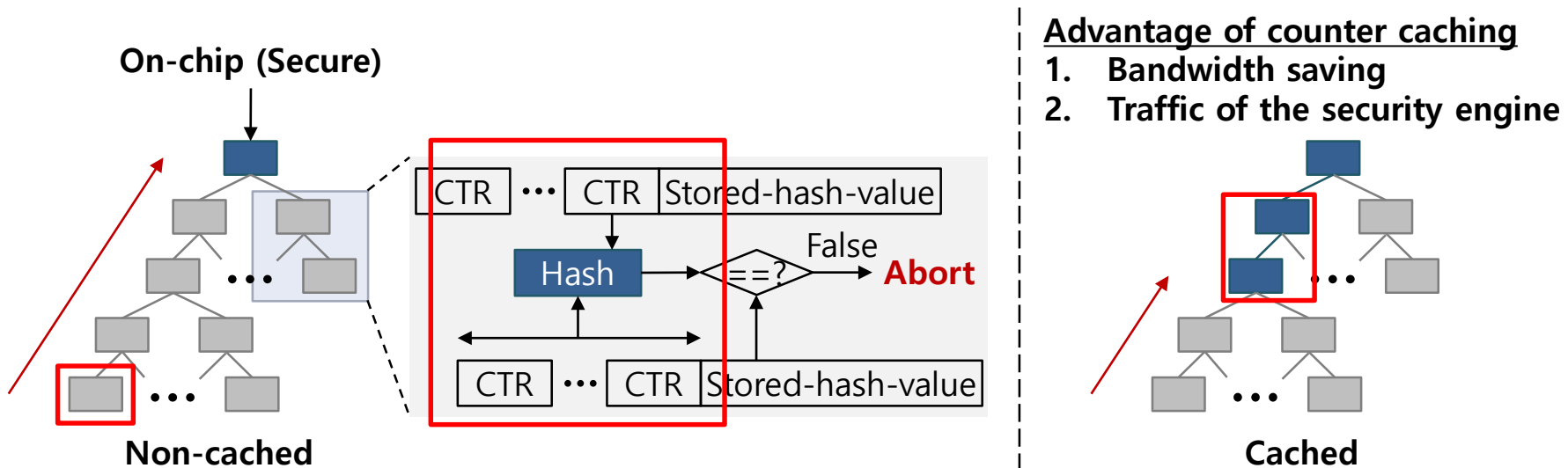
  - NPU IOMMU: **Additional validation entries**

EEPCM

Validation Entries (NPU-side) Validation Entries (CPU-side)

Page Miss Handler

TLB Miss

Hardware-based Validation

Memory Access

System-on-a-Chip

CPU   NPU

Support
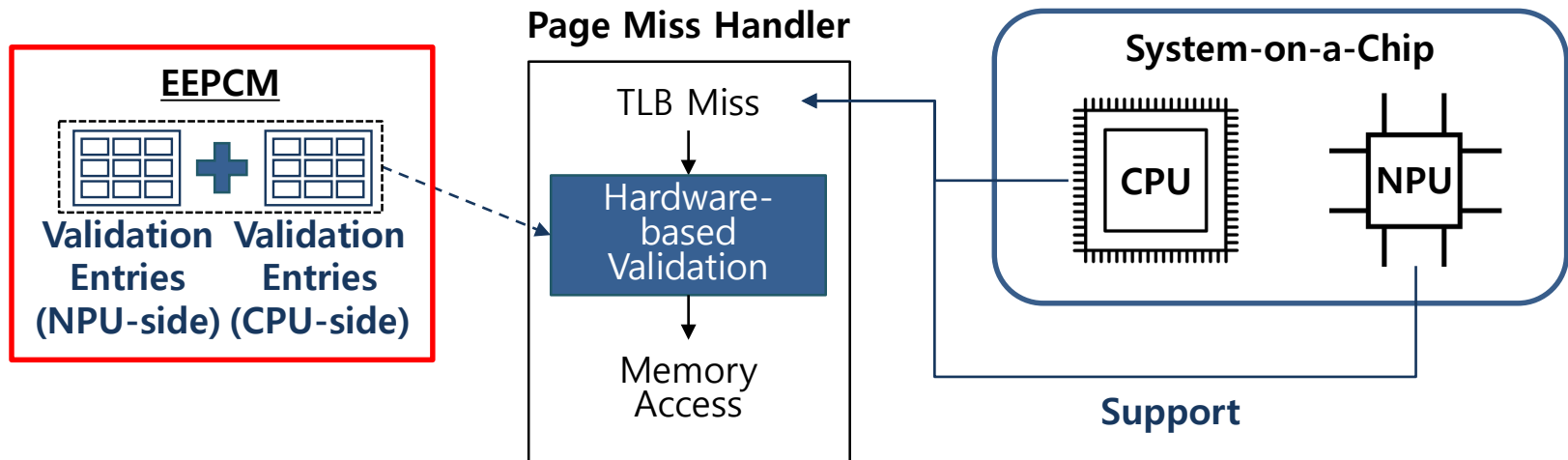
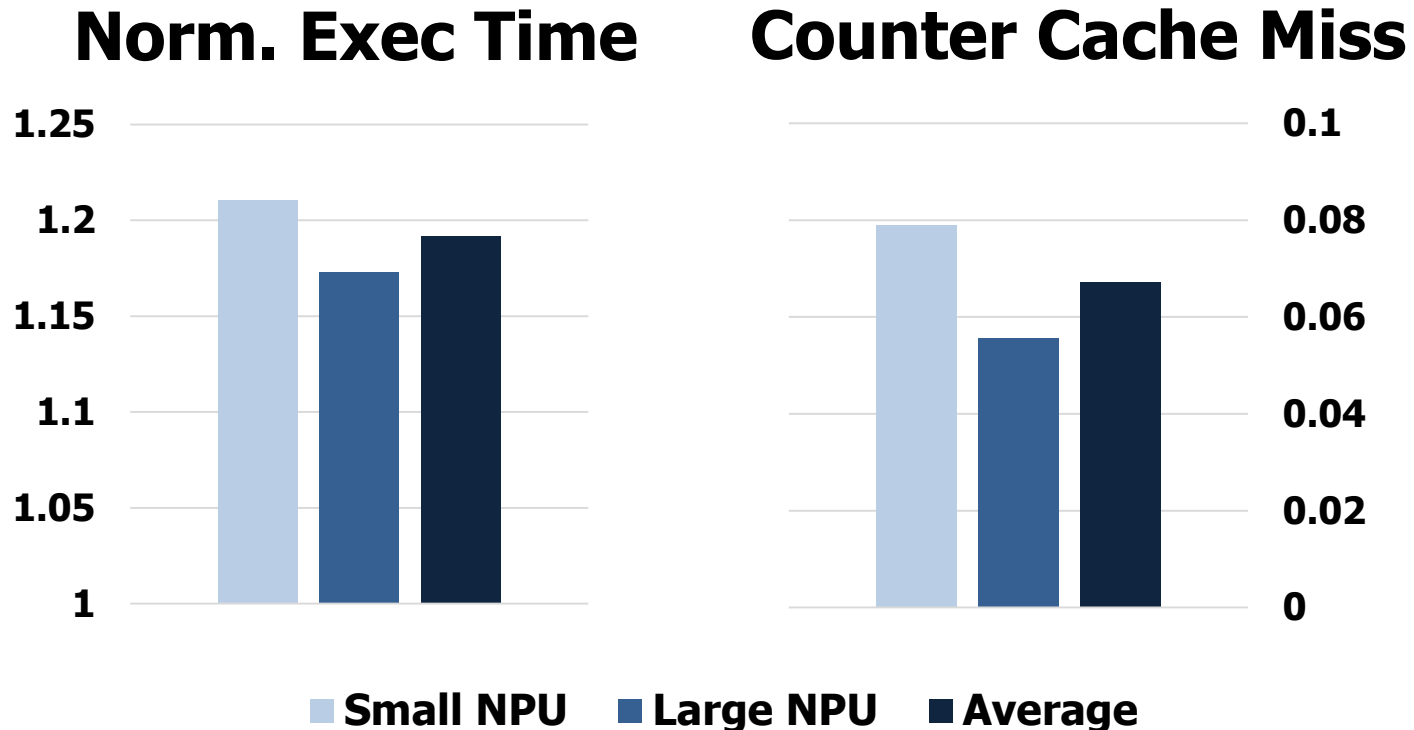# Naive Memory Protection to NPU

- Memory protection
  - Counter-based encryption & integrity protection
  - **Counter Freshness Validation**

On-chip (Secure)

CTR ⋯ CTR Stored-hash-value

Hash → ==? → False **Abort**

CTR ⋯ CTR Stored-hash-value

Non-cached

Advantage of counter caching
1. **Bandwidth saving**
2. **Traffic of the security engine**

Cached

Naive Memory Protection to NPU

- Average **19.2%** performance degradation

- Reason: Counter-cache miss rate (**7.9%**)

Norm. Exec Time

Counter Cache Miss

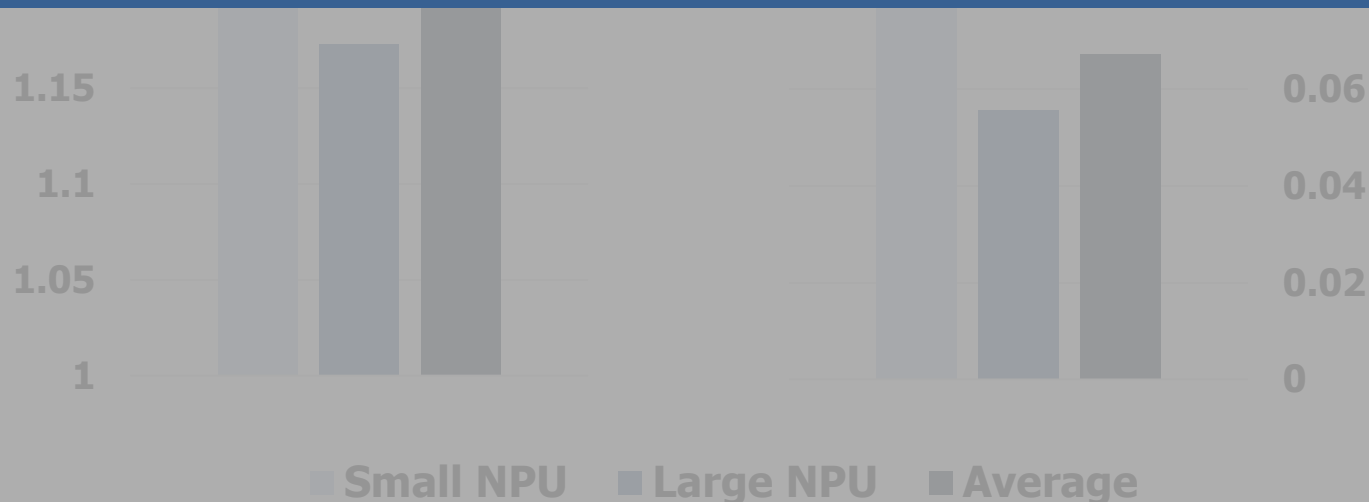■ **Small NPU** ■ **Large NPU** ■ **Average**

Naive Memory Protection to NPU

- Average **19.2%** performance degradation

- Reason: Counter-cache miss rate (**7.9%**)

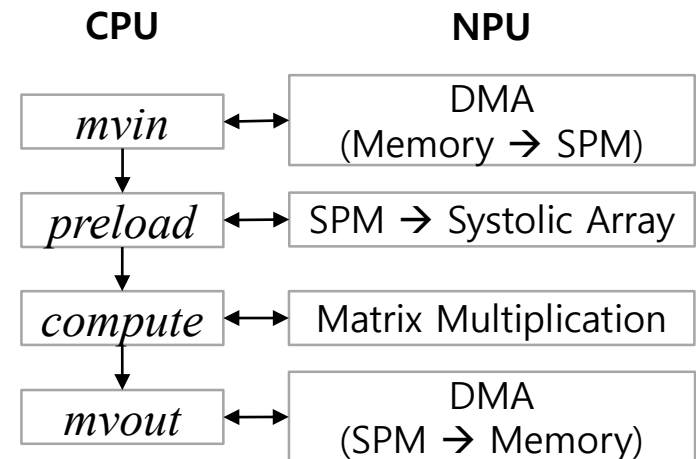**Norm. Exec Time**   **Counter Cache Miss**

A novel memory protection technique for NPU is necessary!

1.15					0.06
1.1					0.04
1.05					0.02
1					0

■ **Small NPU**   ■ **Large NPU**   ■ **Average**

NPU Execution Model

- Execution: $*mvin \rightarrow preload \rightarrow compute \rightarrow **mvout$
  - The **software** controls NPU data movement by commands
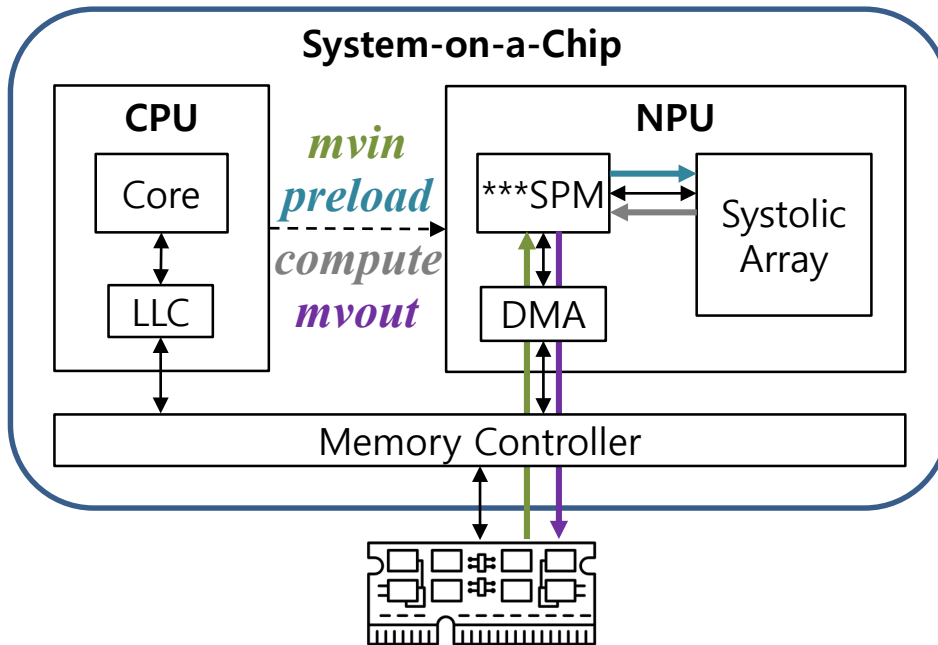
*mvin: move-in, **mvout: move-out, ***SPM: Scratchpad Memory

Tensor-based Computing
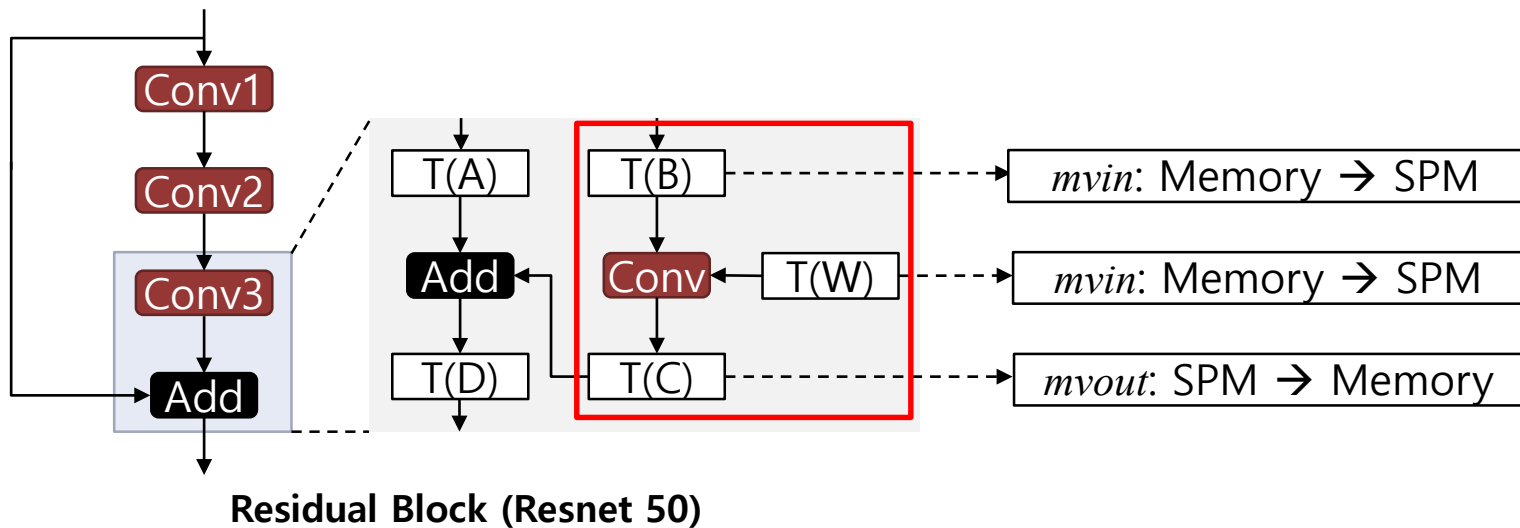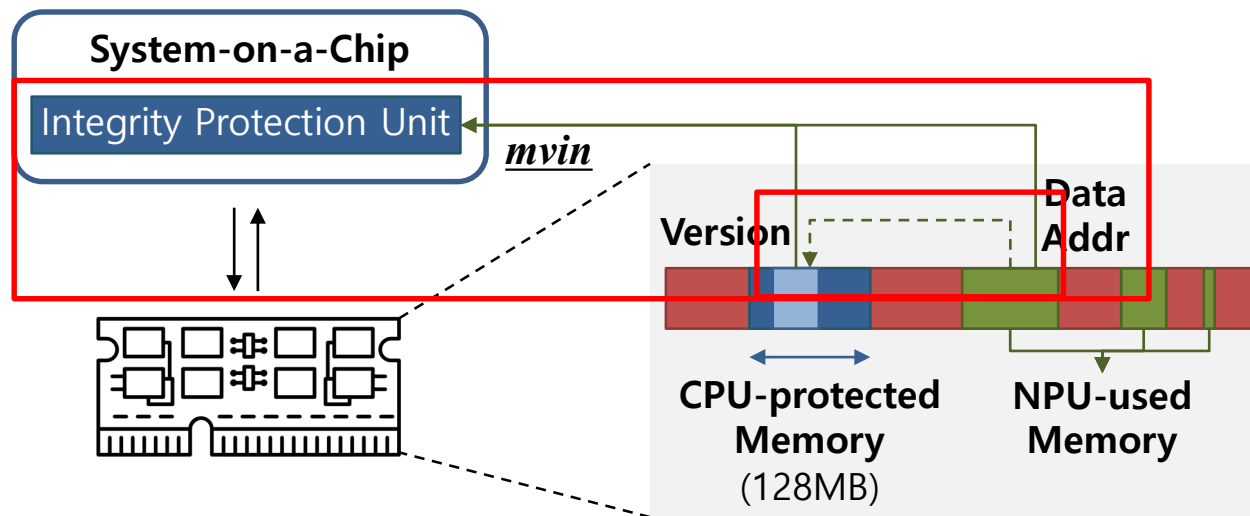
- Tensor-granular computation

  - **<u>Per-tensor version number</u>** is sufficient: Tensor-unit memory access

Residual Block (Resnet 50)

T(B) ⋯→ *mvin*: Memory → SPM

Conv ← T(W) ⋯→ *mvin*: Memory → SPM

T(C) ⋯→ *mvout*: SPM → Memory
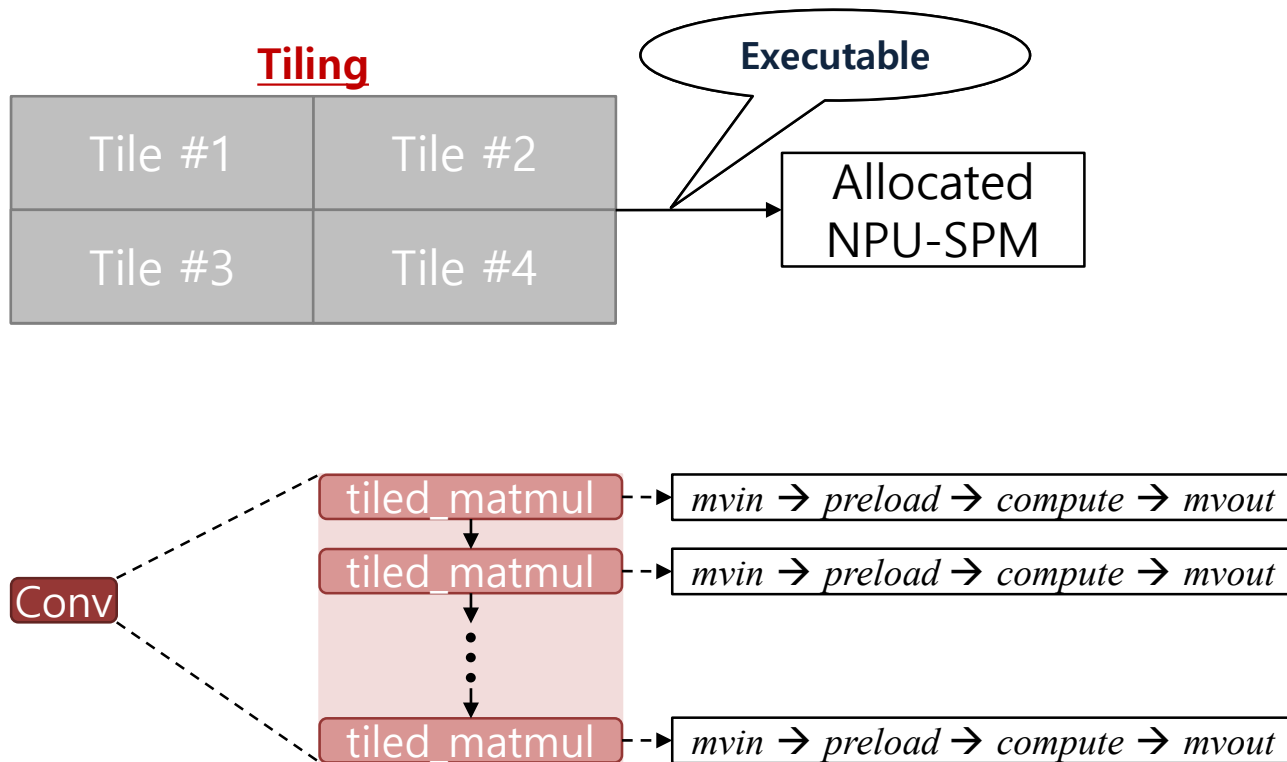
Tree-less Integrity Protection

- Counter → **Version number** controlled by **software**

  - Security granularity: Cacheline → **Tensor**

  - Storage requirement: Only **0.14KB** on average

**Problem: NPU executes layer operation at once?
(i.e Many large tensors are not fitted into SPM)**

Challenge: Intra-layer Computing

- Tensor → One or multiple tiles for intra-layer computing

Tiling

Tile #1	Tile #2
Tile #3	Tile #4

Executable

Allocated
NPU-SPM

Conv

tiled_matmul → *mvin → preload → compute → mvout*

tiled_matmul → *mvin → preload → compute → mvout*

tiled_matmul → *mvin → preload → compute → mvout*

Challenge: Intra-layer Computing

- Tensor → One or multiple tiles for intra-layer computing

Tiling

Executable

Tile #1 Tile #2

Allocated
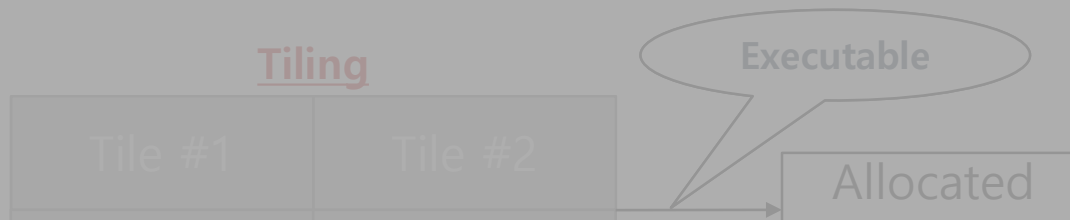
Tile-granular version number is necessary in intra-layer!

Conv

tiled_matmul ---▶ *mvin → preload → compute → mvout*

tiled_matmul ---▶ *mvin → preload → compute → mvout*

tiled_matmul ---▶ *mvin → preload → compute → mvout*
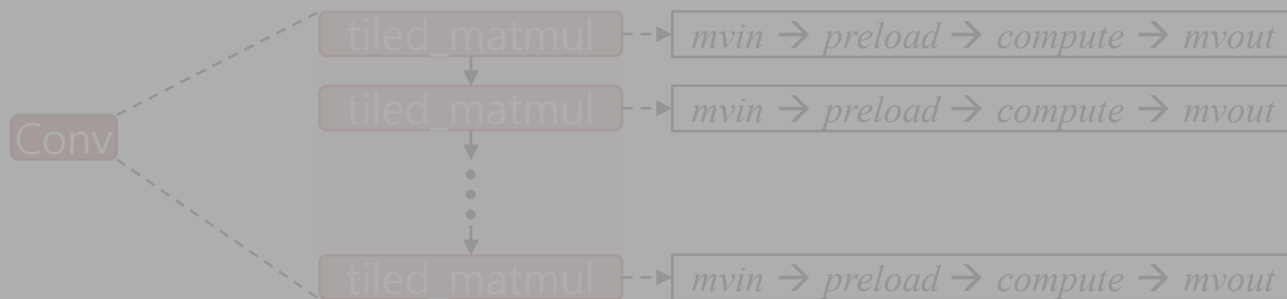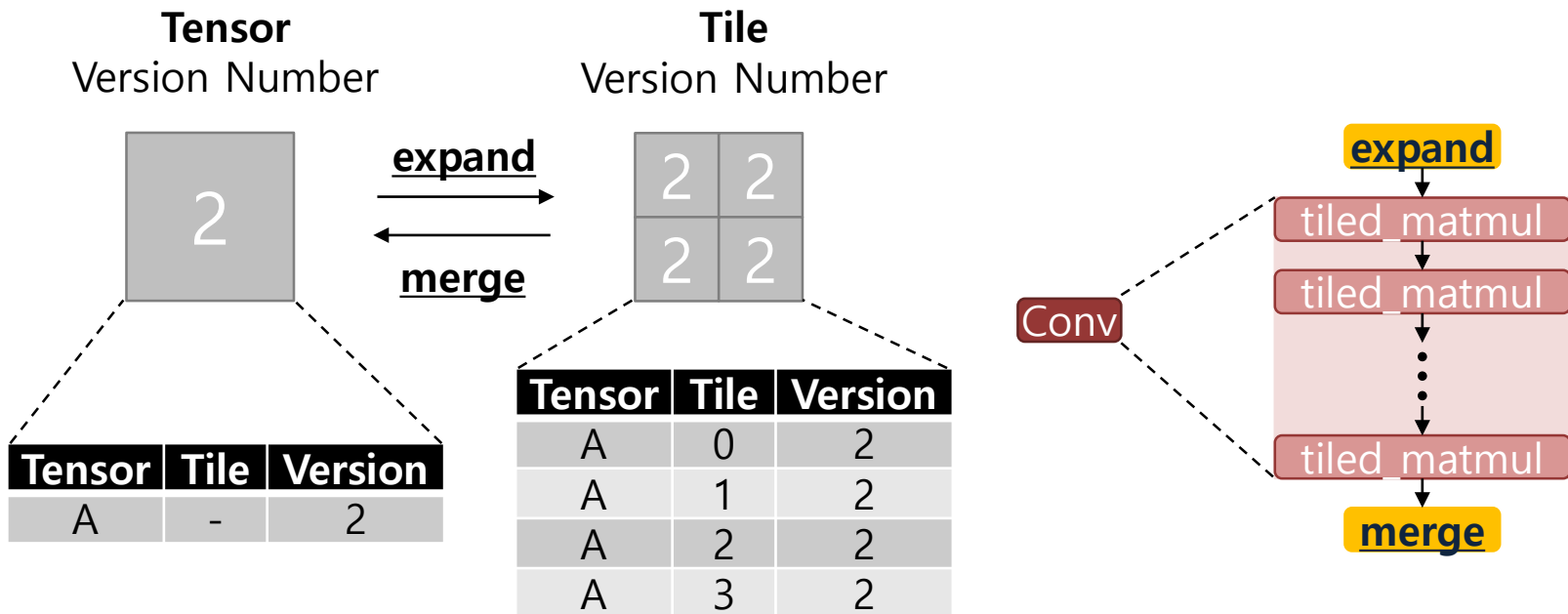
Tile-granular Version Number

- Tensor → One or multiple tiles for intra-layer computing

Tensor A   Tensor B   Tensor C
                         Version Number
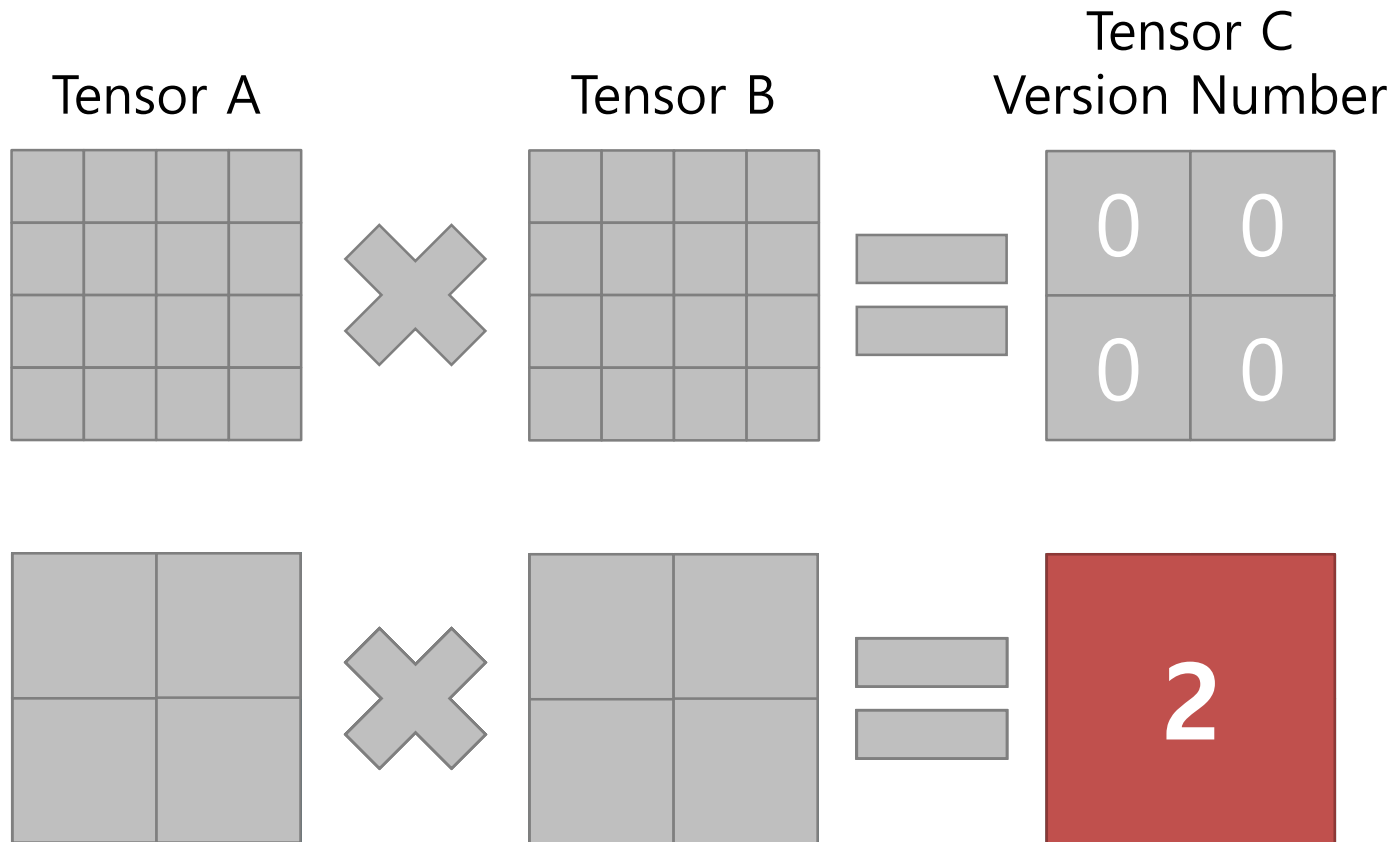
Tensor/Tile Version Number

- Tensor/Tile version number

    - Granularity: Cacheline → Tensor/**Tile (Intra-layer)**

    - Storage requirement: Only **1.3KB** on average

- **expand**, **merge**: Granularity translation operation

Tensor
Version Number

Tile
Version Number

expand

merge

Tensor	Tile	Version
A	-	2

Tensor	Tile	Version
A	0	2
A	1	2
A	2	2
A	3	2

expand
tiled_matmul
tiled_matmul
tiled_matmul

Conv

merge

Evaluation Environment

- Cycle-level simulation modified from *SCALE-Sim

- Two edge-level system-on-a-chip configurations

    - Samsung Exynos 990 (Small NPU), ARM Ethos N77 (Large NPU)

- Workloads: 14 models in MLPerf, DeepBench

	Small NPU (Samsung Exynos 990)	Large NPU (ARM Ethos N77)
PE	32 x 32	45 x 45
Bandwidth	11 GB/s (4 channels)	22 GB/s (4 channels)
Frequency	2.75 GHz (both processor/memory)	1 GHz (both processor/memory)
SPM	480KB in total	1MB in total
Precision	Float16	Float16

* A systematic methodology for characterizing scalability of DNN accelerators using SCALE-Sim (ISPASS 2020)
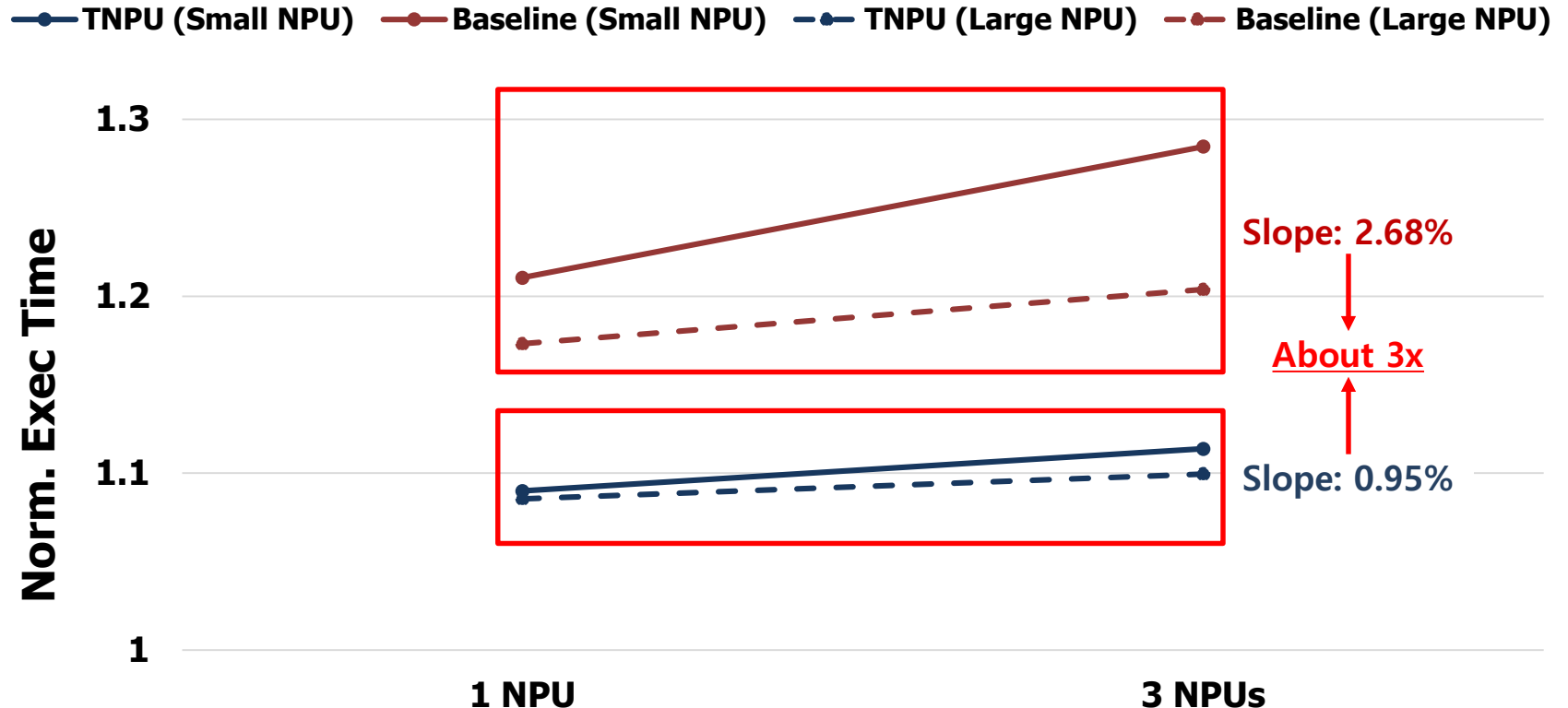
Evaluation Result (Single NPU)

- **Performance improvement: 8.75%**
  - Data traffic reduction: **7.67%**

- **Remaining performance degradation: 8.80%** (Comp. Unsecure)
  - Stored-hash-value (Message-authentication-code; MAC)
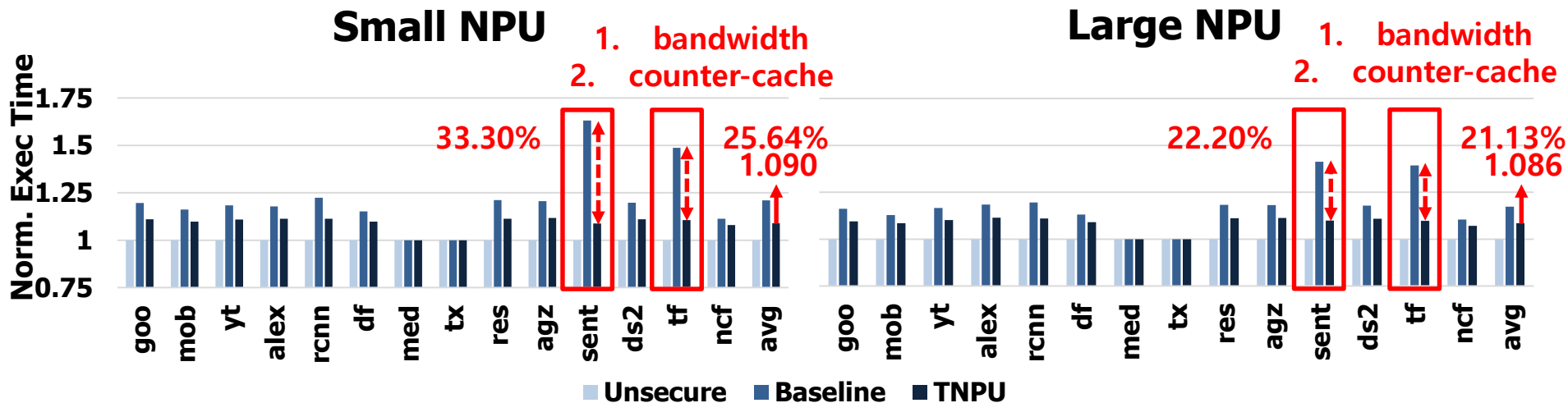
Evaluation Result (Multiple NPUs)

- Scalability: Slope (TNPU) < Slope (Baseline)

- Performance improvement: 8.75% → **11%**

Summary

- **Result**

  - Trusted Execution environment for NPU

  - Performance improvement: **8.75%** (single), **11%** (3-NPU)

- **Challenge**

  - Counter tree overhead

- **Idea**

  - Counter → Tensor/tile-granular version number

- **Further Work**

  - Stored-hash-value (MAC) optimization

Thank you