NeuPIMs: NPU-PIM Heterogeneous Acceleration for Batched LLM Inferencing

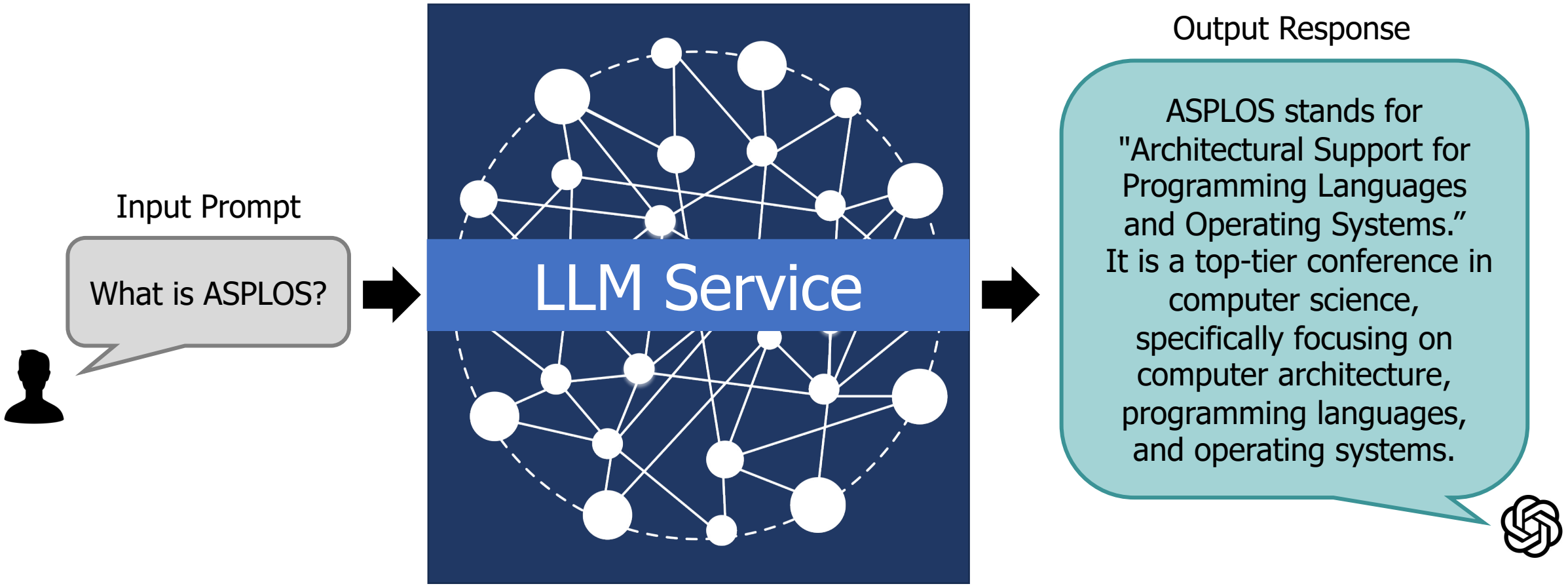
Guseul Heo
Sangyeop Lee
Jaehong Cho
Hyunmin Choi
Sanghyeon Lee
Hyungkyu Ham[†]
Gwangsun Kim[†]
Divya Mahajan[§]
Jongse Park

KAIST

POSTECH[†]

Georgia Institute of Technology[§]

ASPLOS 2024


Input Prompt
What is ASPLOS?
LLM Service
Output Response
ASPLOS stands for "Architectural Support for Programming Languages and Operating Systems." It is a top-tier conference in computer science, specifically focusing on computer architecture, programming languages, and operating systems.

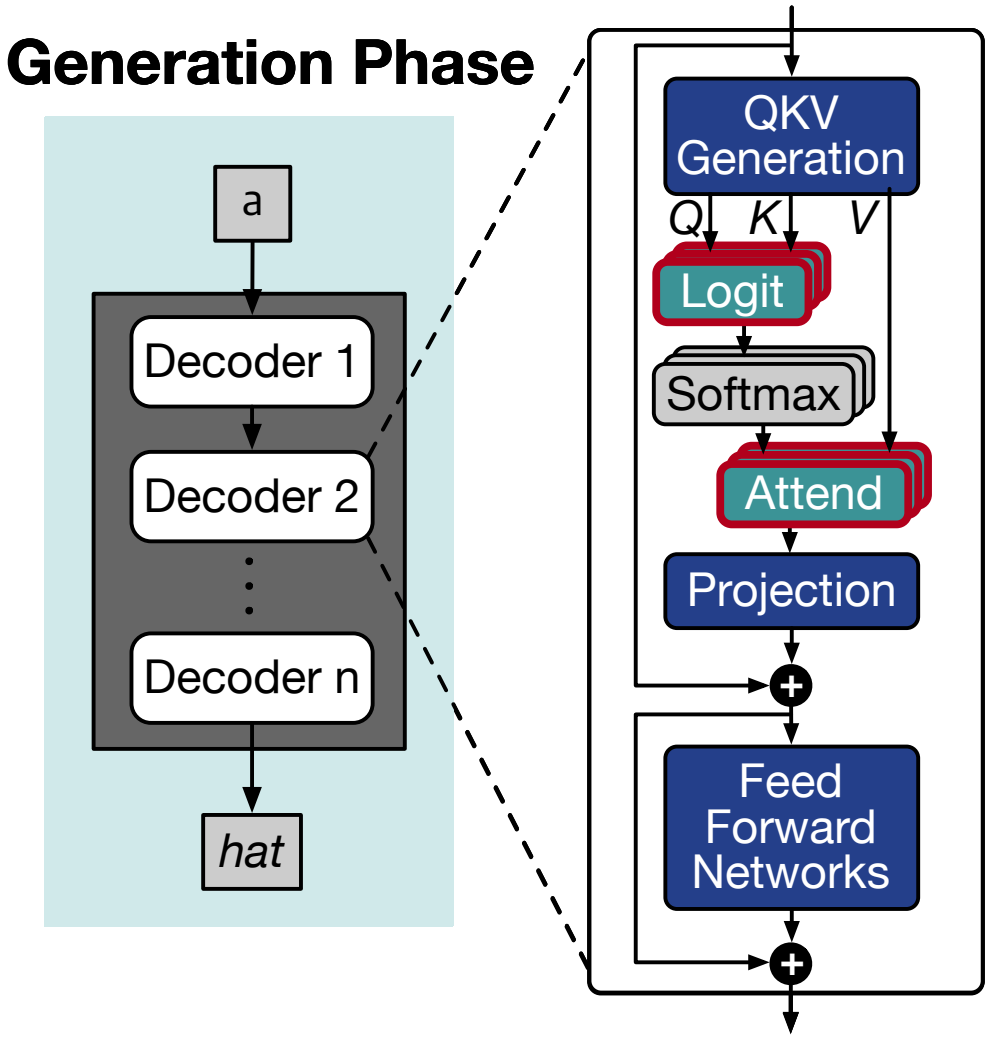
Input Prompt

Summarization Phase **x 1**

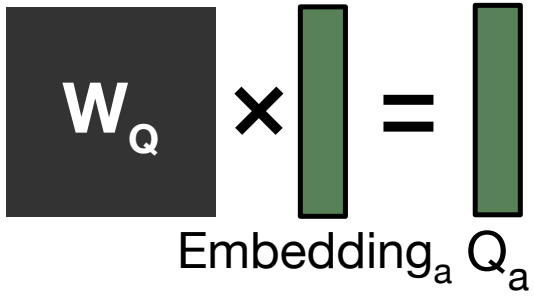
Generation Phase **x N**

Output Response

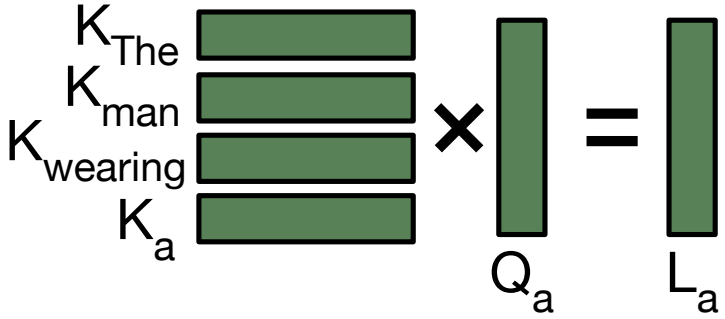
GEMV in Generation Phase



1. Weight-activation operation

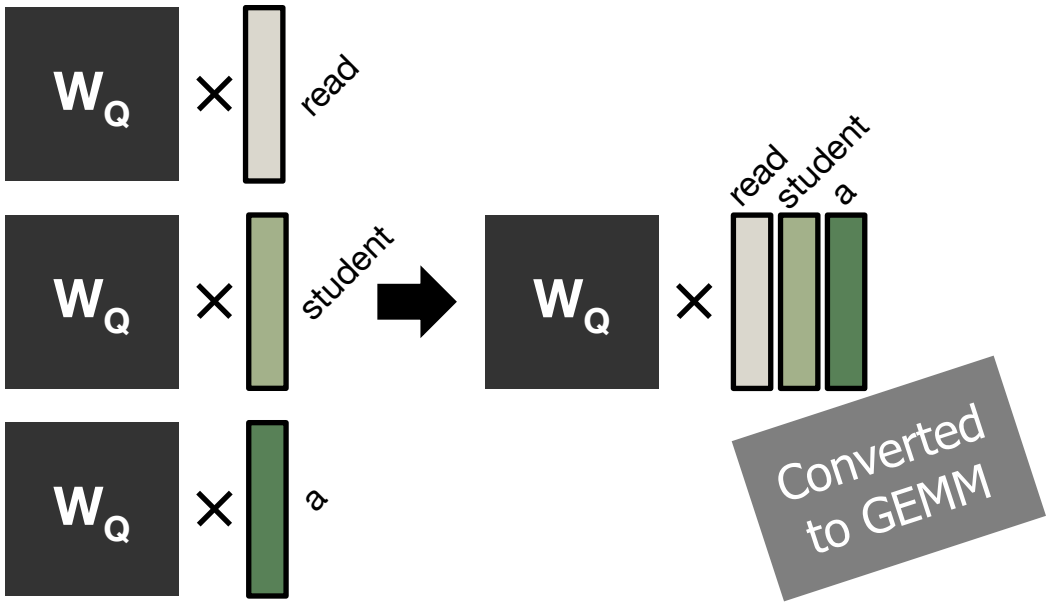


2. Activation-activation operation



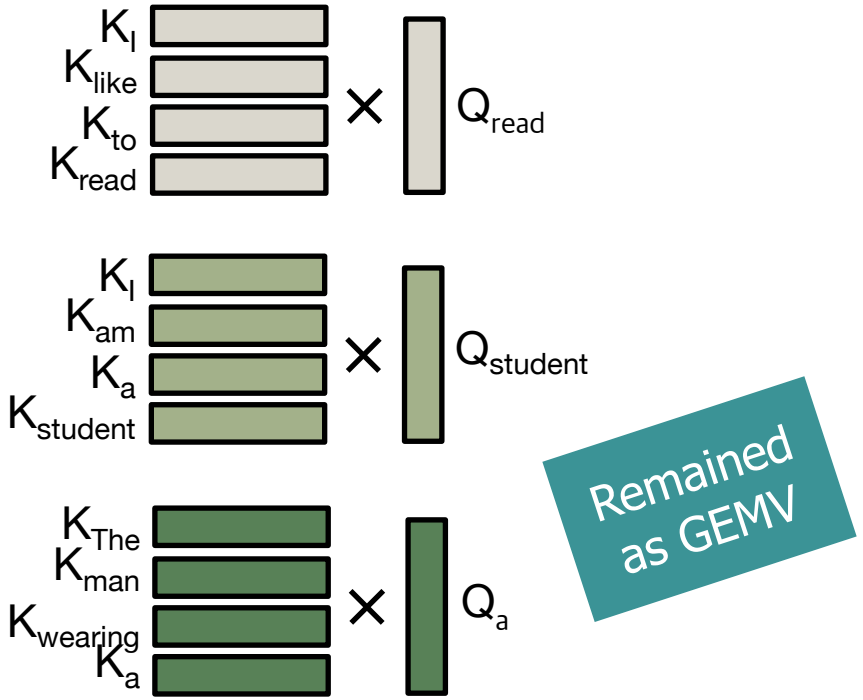
Batched Inference

Weight-activation operation



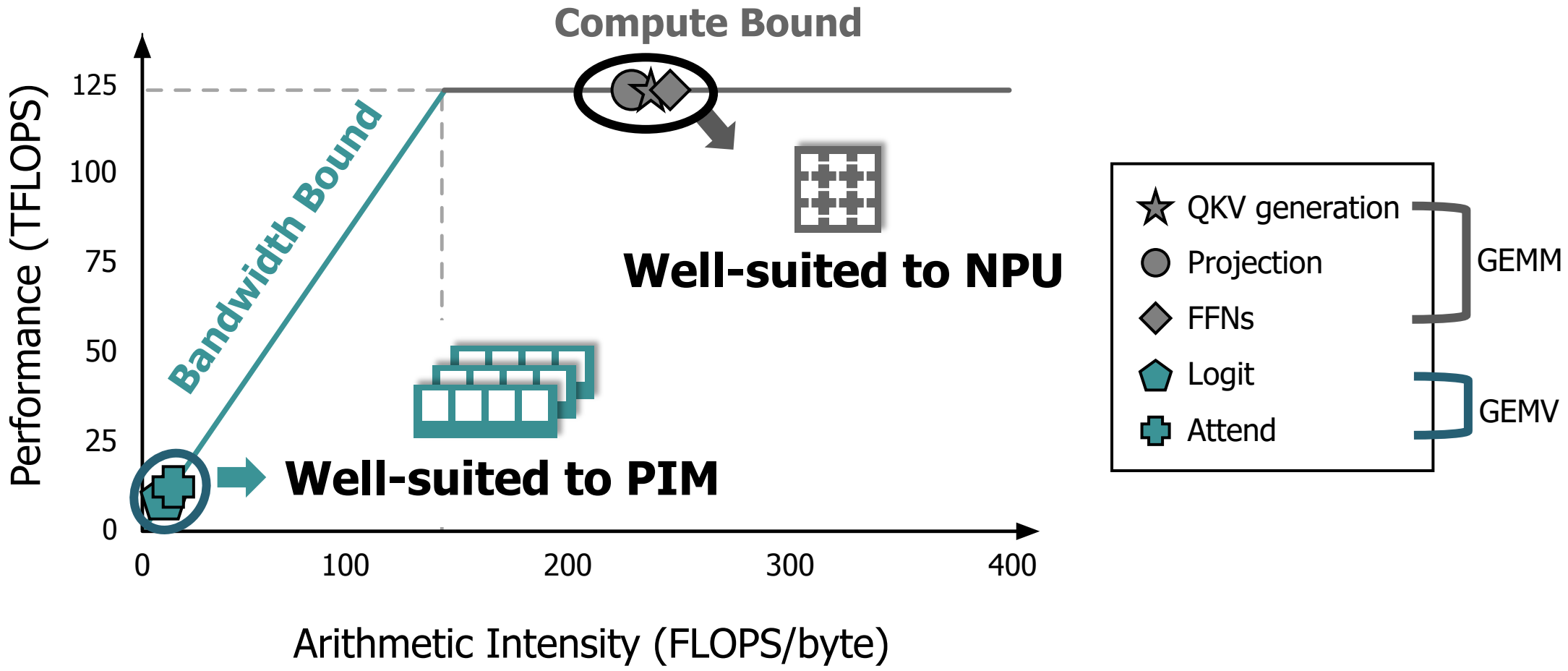
- Model weight parameter are **reusable**
- Possible to batch

Activation-activation operation



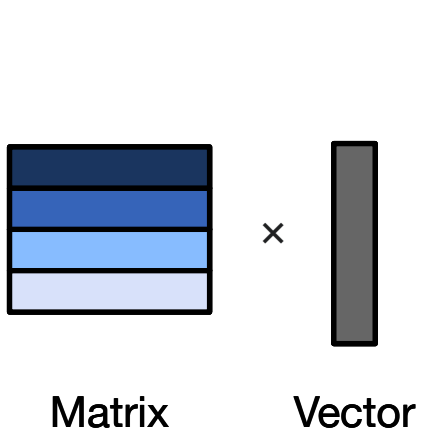
- **No reusable parameters**
- Cannot be batched

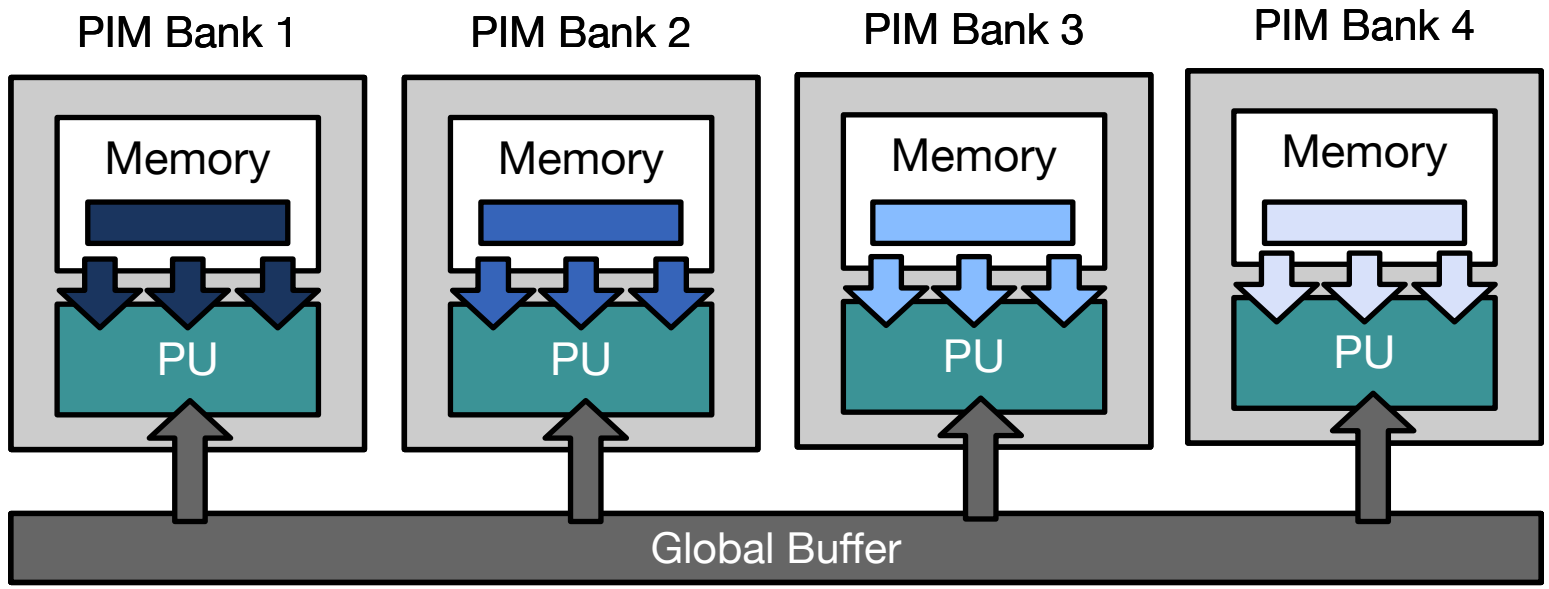
Arithmetic Intensity



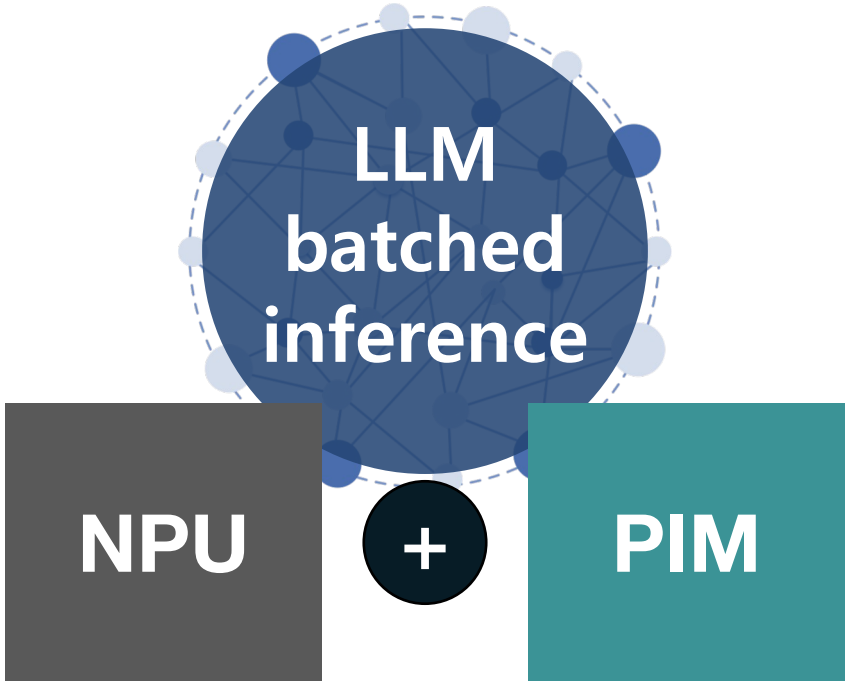
Processing in Memory (PIM) for GEMV

- Processing unit in memory, utilizing **high internal bandwidth**
- Efficient to bandwidth-bound operations such as GEMV





NPU+PIM Integration



Compute-centric accelerator
Computation power
Effective for **GEMM**

Bandwidth-centric accelerator
High internal bandwidth
Effective for **GEMV**

NPU+PIM Integration

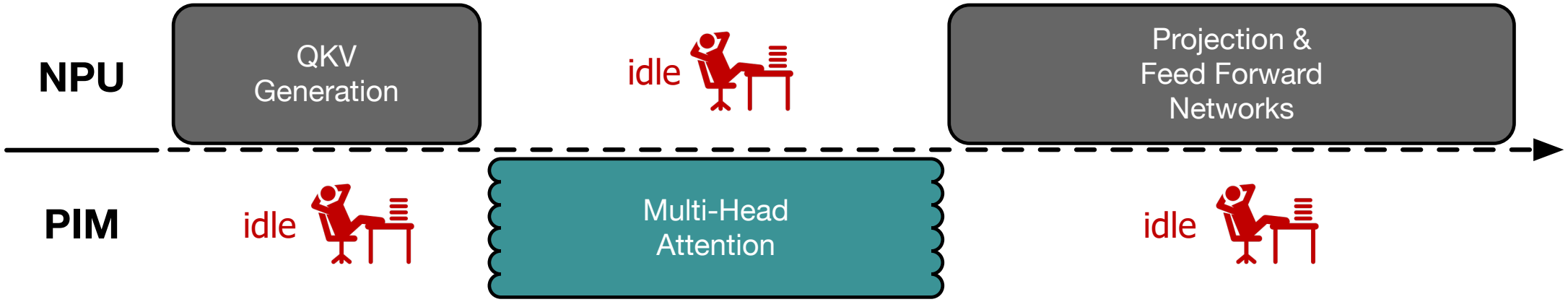
We aim to leverage **NPU+PIM heterogeneous acceleration** for efficient batched inference of LLM

Compute-centric accelerator
Computation power
Effective for **GEMM**

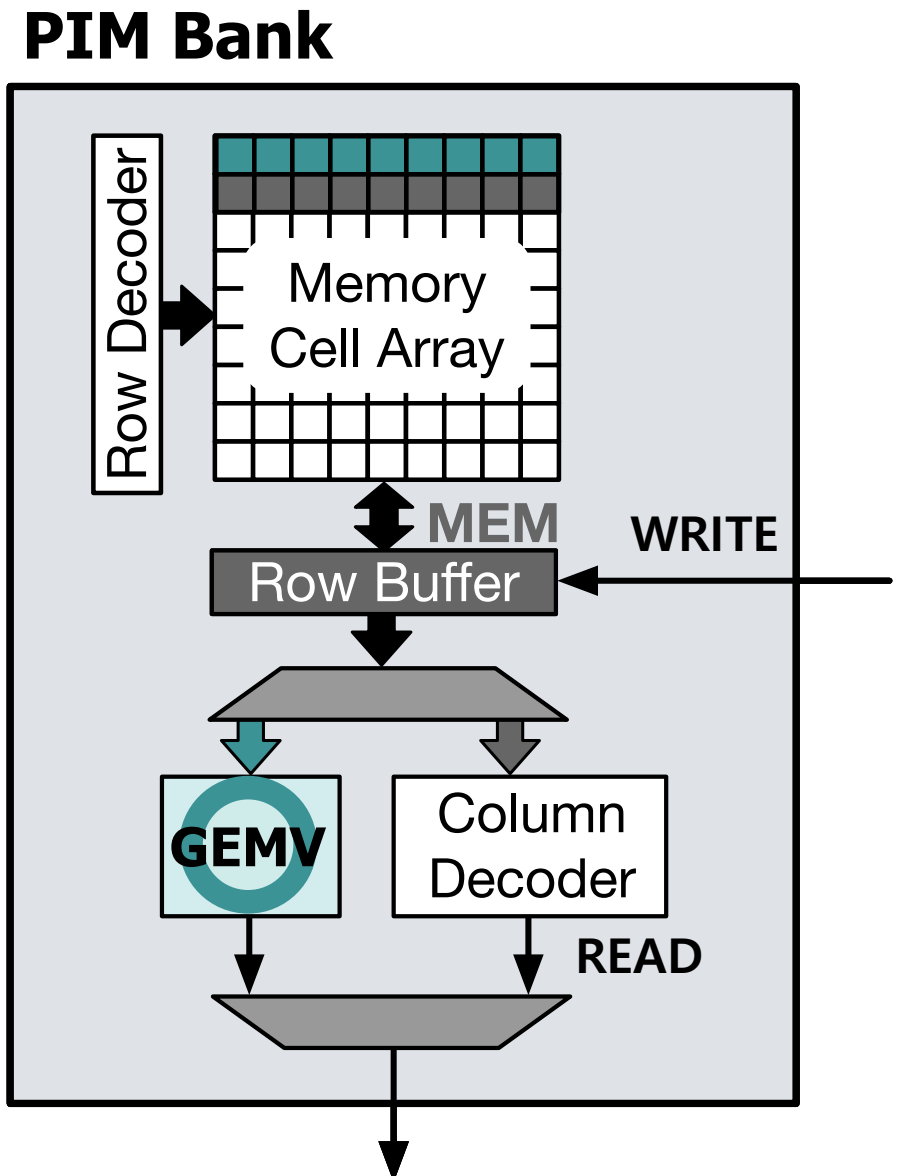
Bandwidth-centric accelerator
High internal bandwidth
Effective for **GEMV**

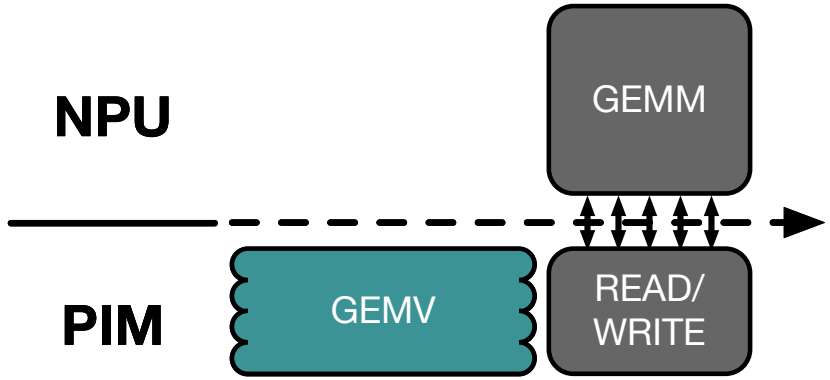
NPU+PIM Integration for LLM

- NPU+PIM integration suffers from resource underutilization

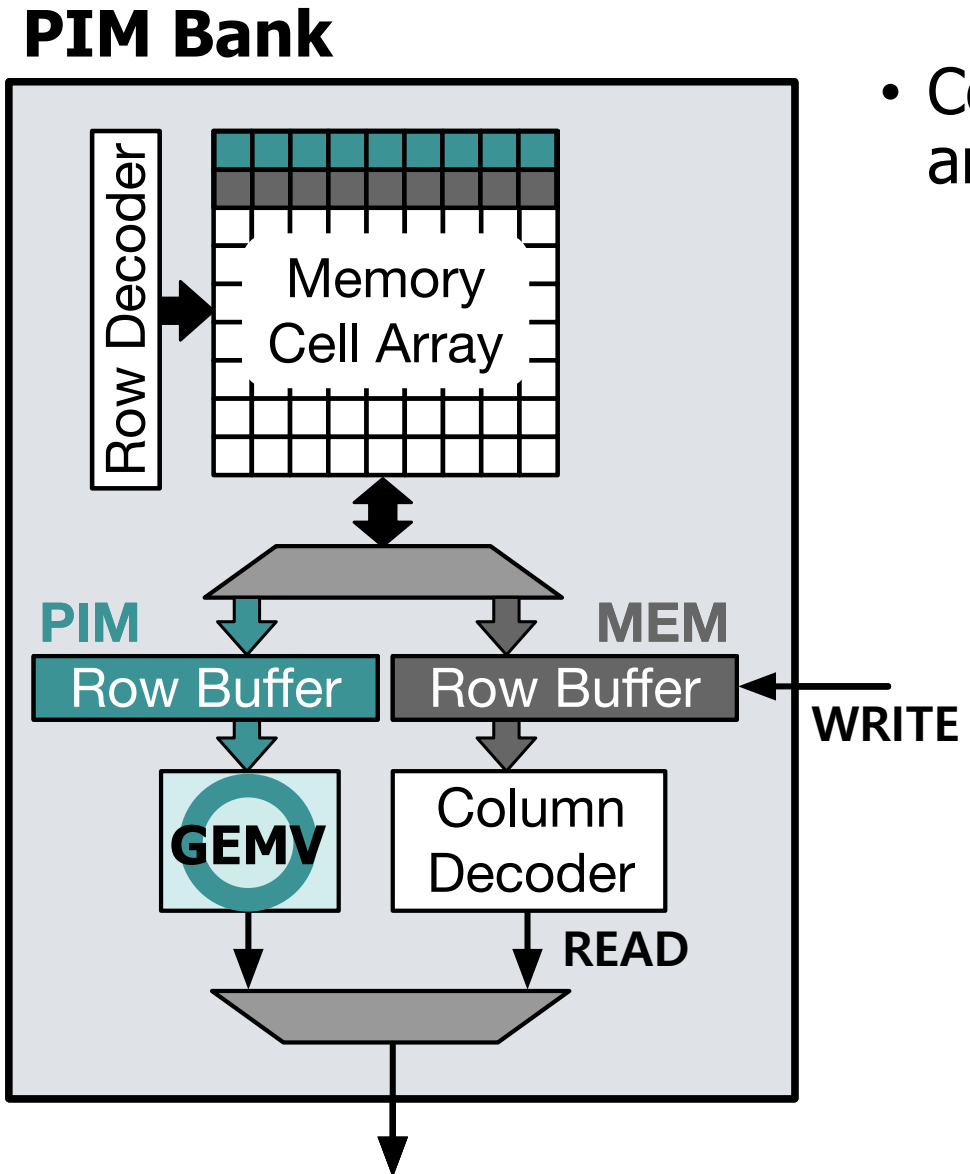


Challenge 1: Blocked Mode PIM

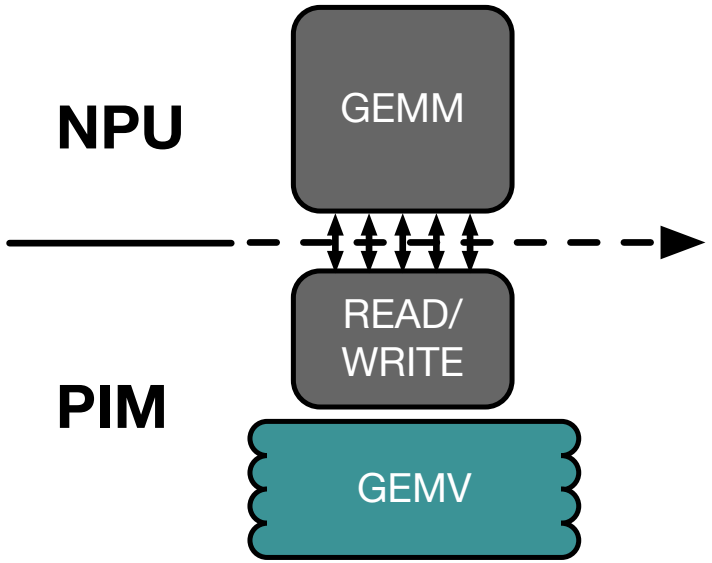




Solution 1: Dual Row Buffer PIM

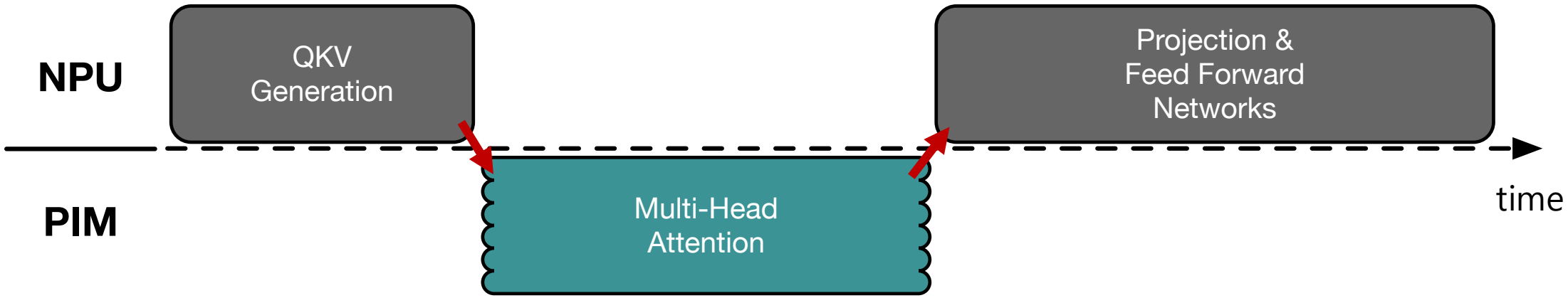


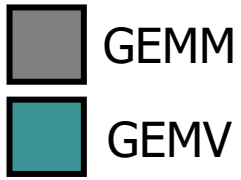
- Concurrent execution of PIM computation and memory access for NPU



Challenge 2: GEMM-GEMV Dependency

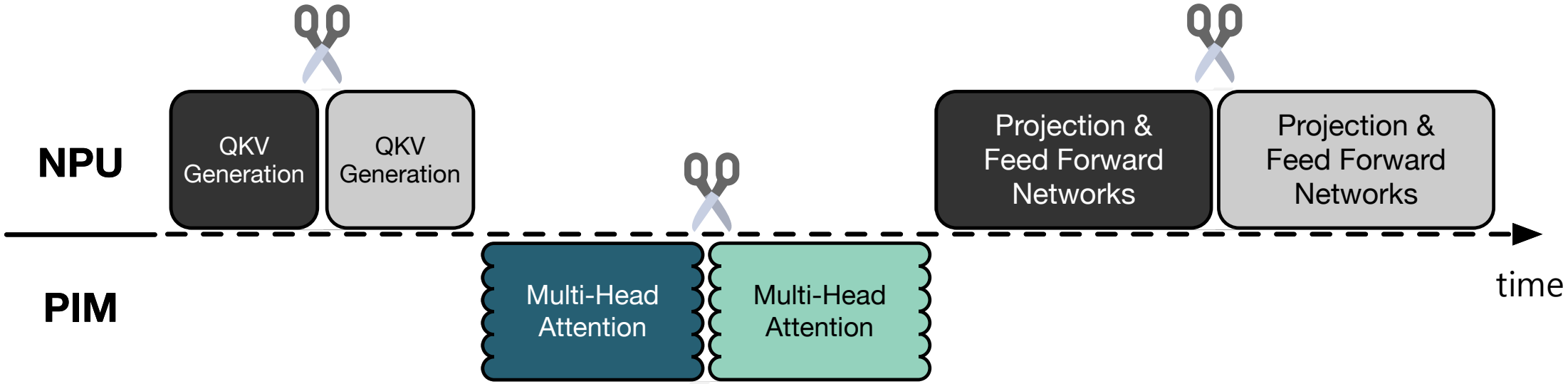
- GEMM and GEMV dependency in single batch of LLM inference





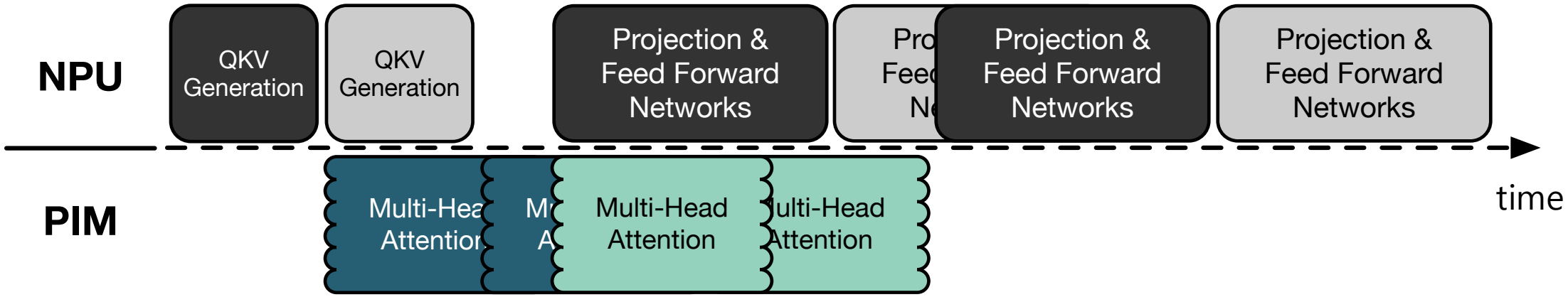
Solution 2: Sub-batch Interleaving

- Divide one large batch into two sub-batches and alternate them



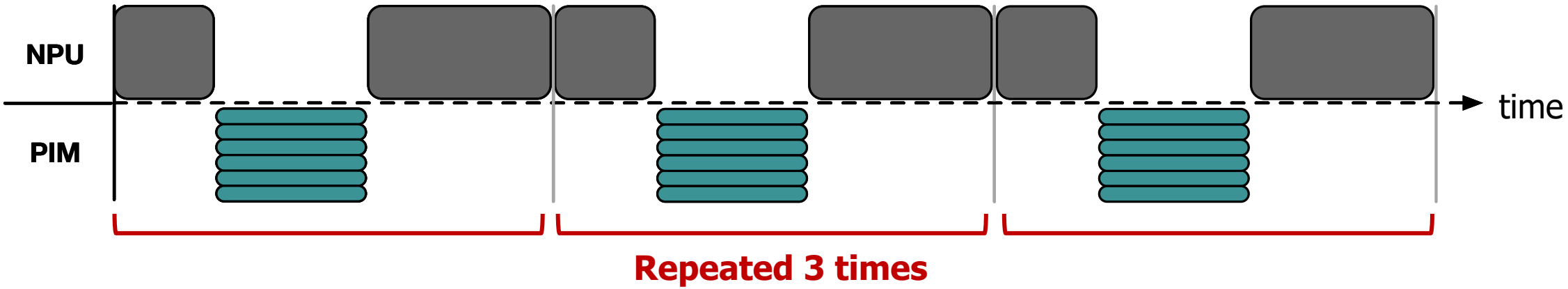
Solution 2: Sub-batch Interleaving

- Divide one large batch into two sub-batches and alternate them

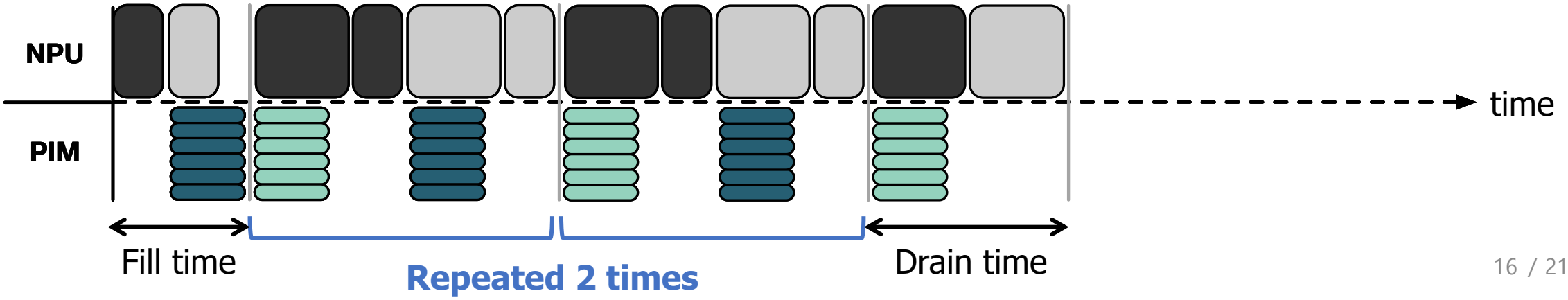


Example of 3 Decoder Blocks

NPU+PIM non interleaving

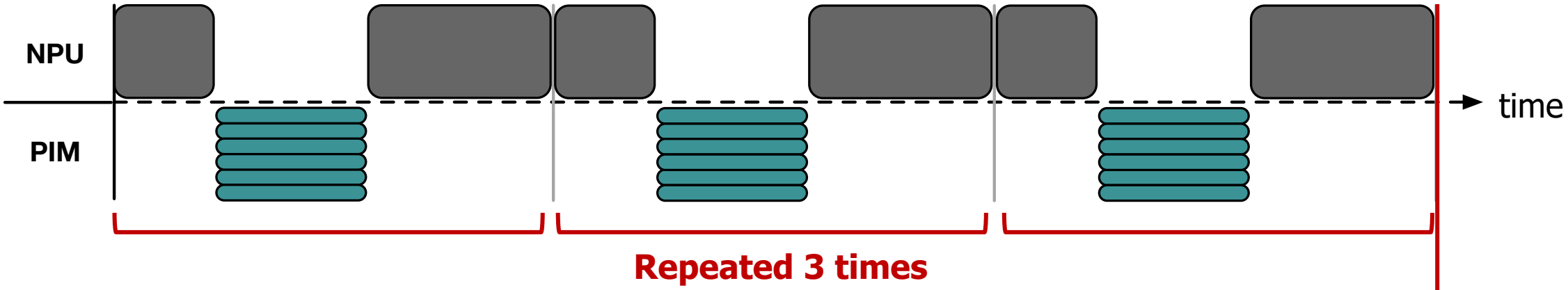


Sub-batch interleaving

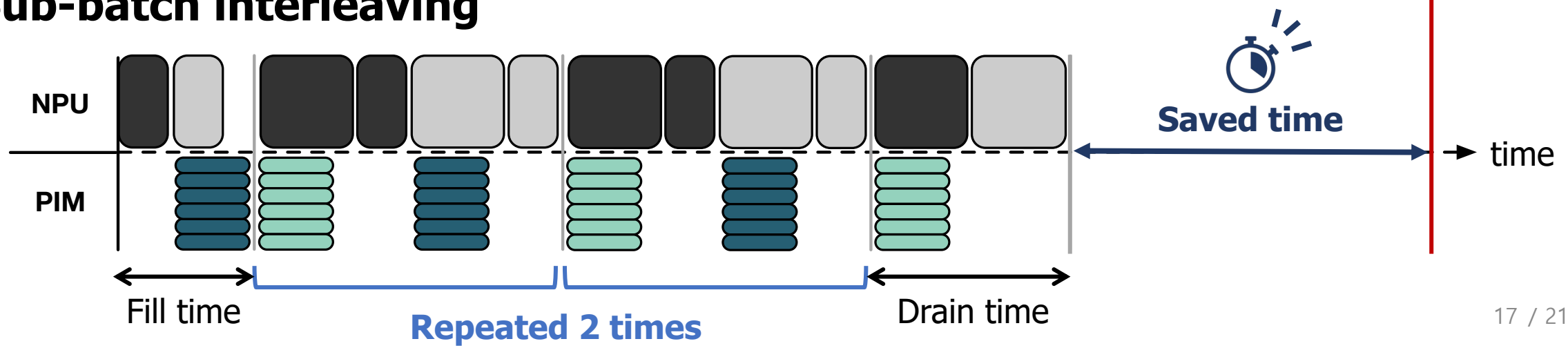


Example of 3 Decoder Blocks

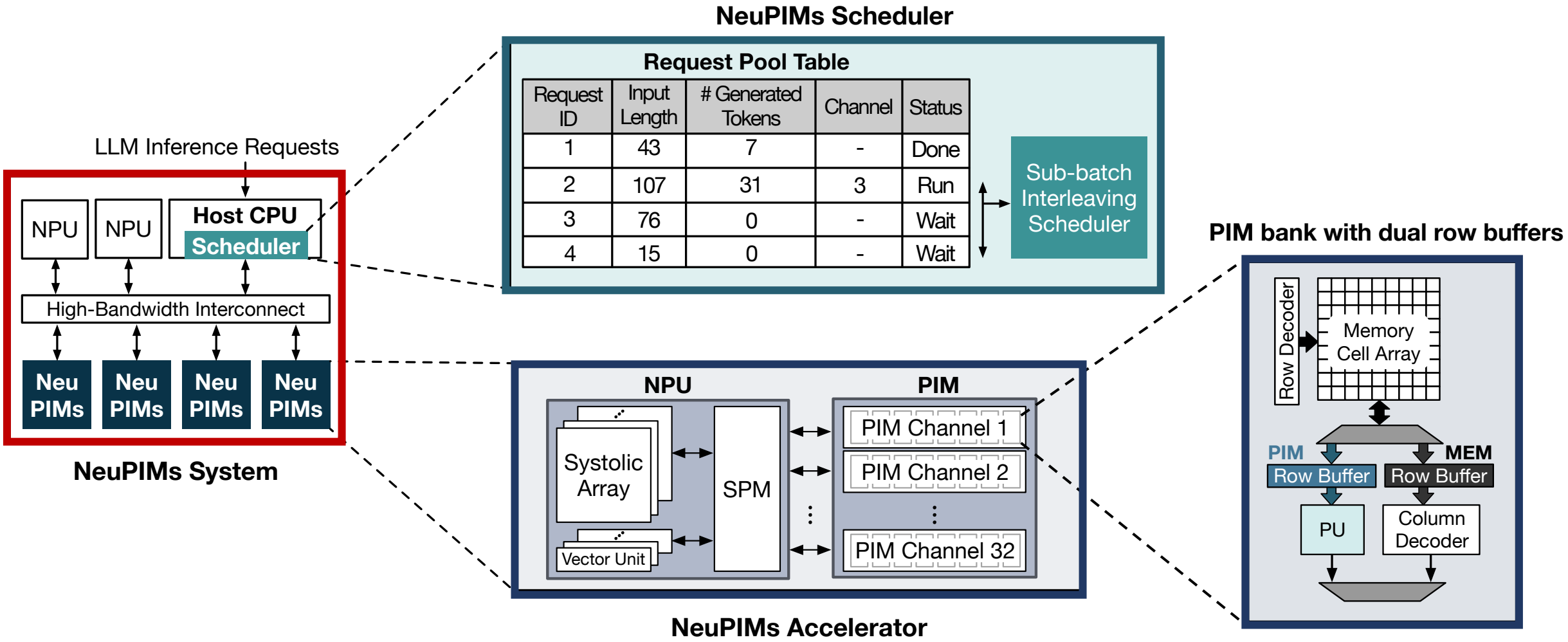
NPU+PIM non interleaving



Sub-batch interleaving

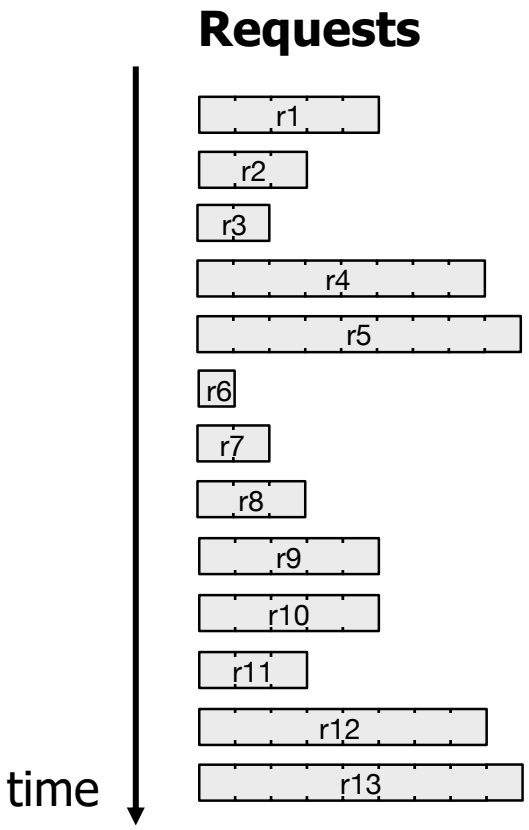


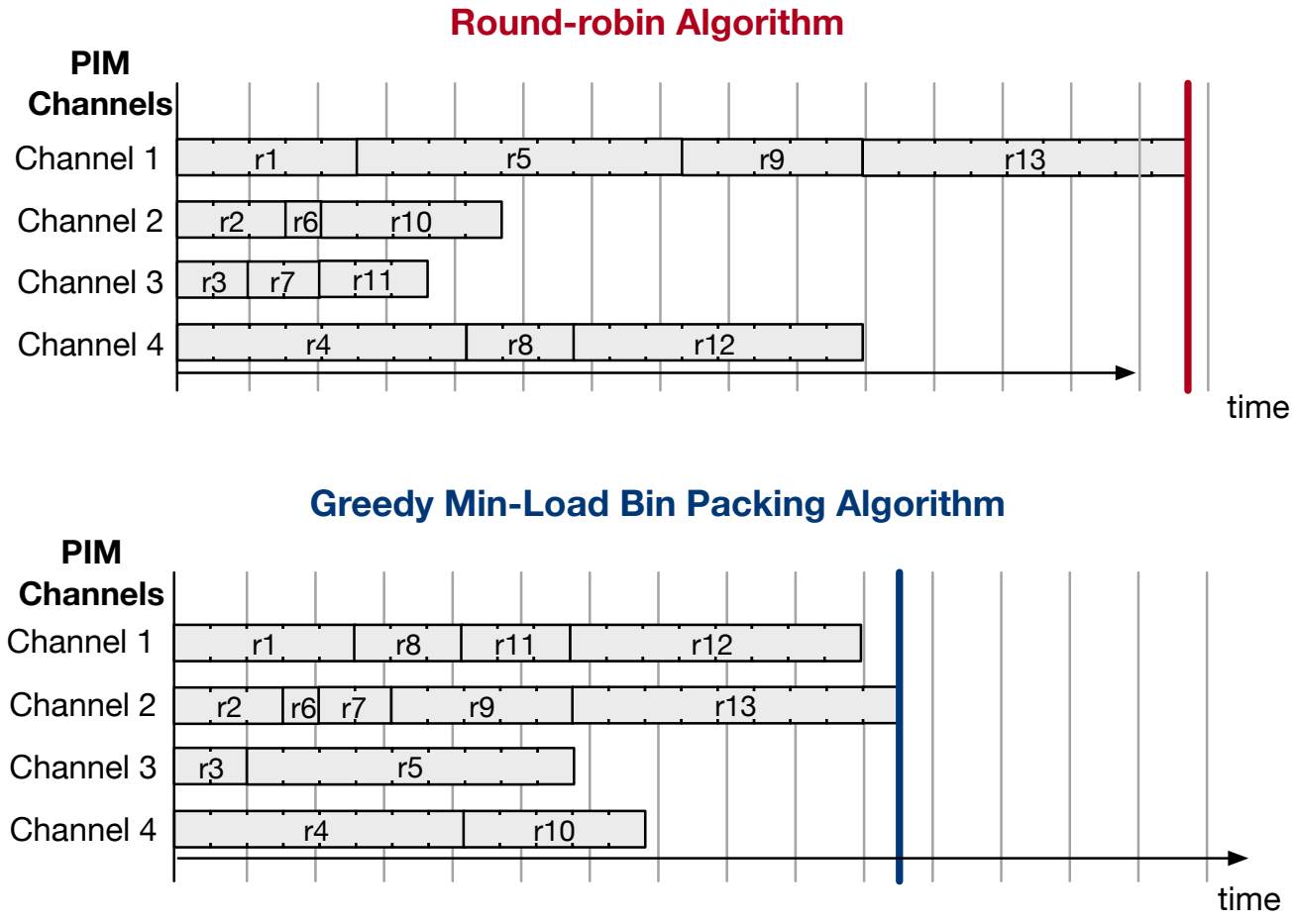
NeuPIMs System



Greedy Min-Load Bin Packing Algorithm

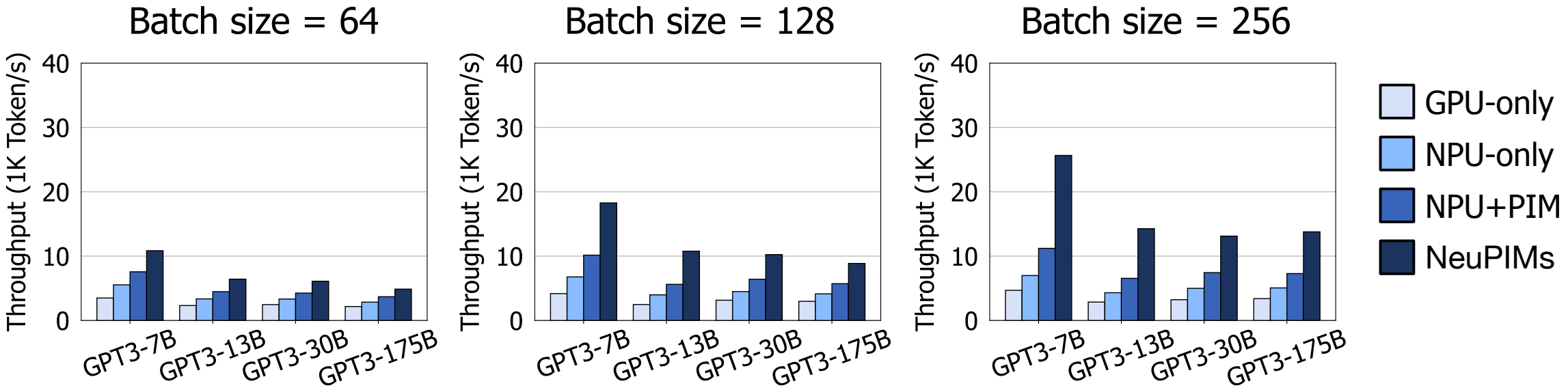
- Distribute requests to PIM channels





NeuPIMs Performance

- Dataset: ShareGPT



- Dual row buffer PIM architecture and sub-batch interleaving boosted performance
- The benefits increase with larger batch sizes
- **2.4x** speedup over NPU-only, **1.6x** speedup over NPU+PIM

More Results in Paper

- Utilization improvement
- Ablation study
- Model parallelism sensitivity
- Hardware overhead (area/power)
- Comparison with prior PIM-only solution

Conclusion

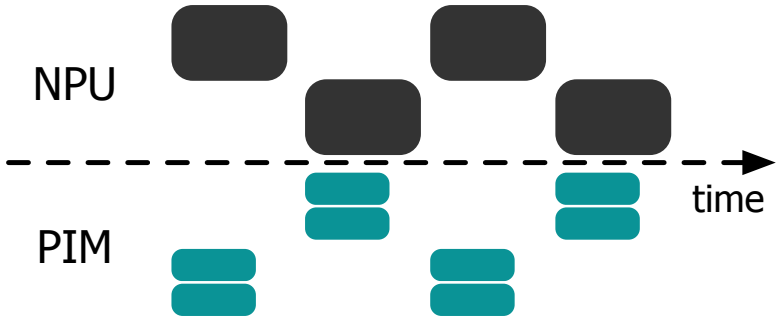
Our simulator code is available
https://github.com/casys-kaist/NeuPIMs

- **NeuPIMs**
  - NPU+PIM heterogeneous acceleration system for LLM batched inference
- **Contributions**
  - PIM microarchitecture equipped with dual row buffer for concurrent execution
  - Sub-batch interleaving technique to overlap NPU execution and PIM execution

Dual row buffer PIM

Concurrent Execution

Sub-batch interleaving



Throughput improvement

2.4×

over NPU-only

1.6×

over naïve NPU+PIM