Tender: Accelerating Large Language Models via Tensor Decomposition and Runtime Requantization

Jungi Lee*, Wonbeom Lee*, Jaewoong Sim
Seoul National University

* Equal Contribution

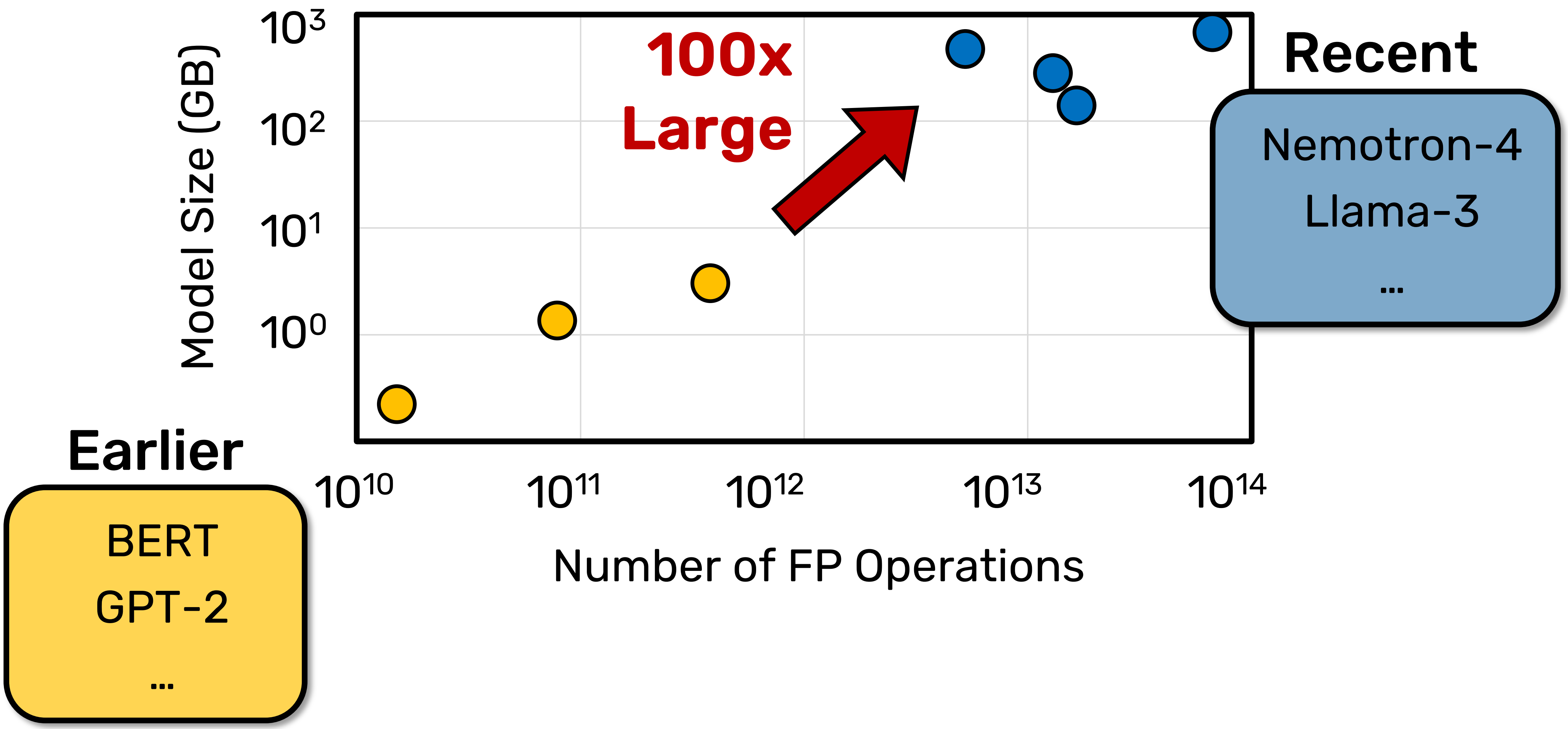
ISCA-51 | July 2024

Department of ECE
Seoul National University

Outline

- **Motivation**
  - Challenges in Efficient LLM Inference
  - Limitations of Prior Works
- Tender**: Algorithm-Hardware Co-design for Efficient LLM Inference**
  - Tensor Decomposition
  - Rescaling Operation
- **Evaluation**
- **Conclusion**

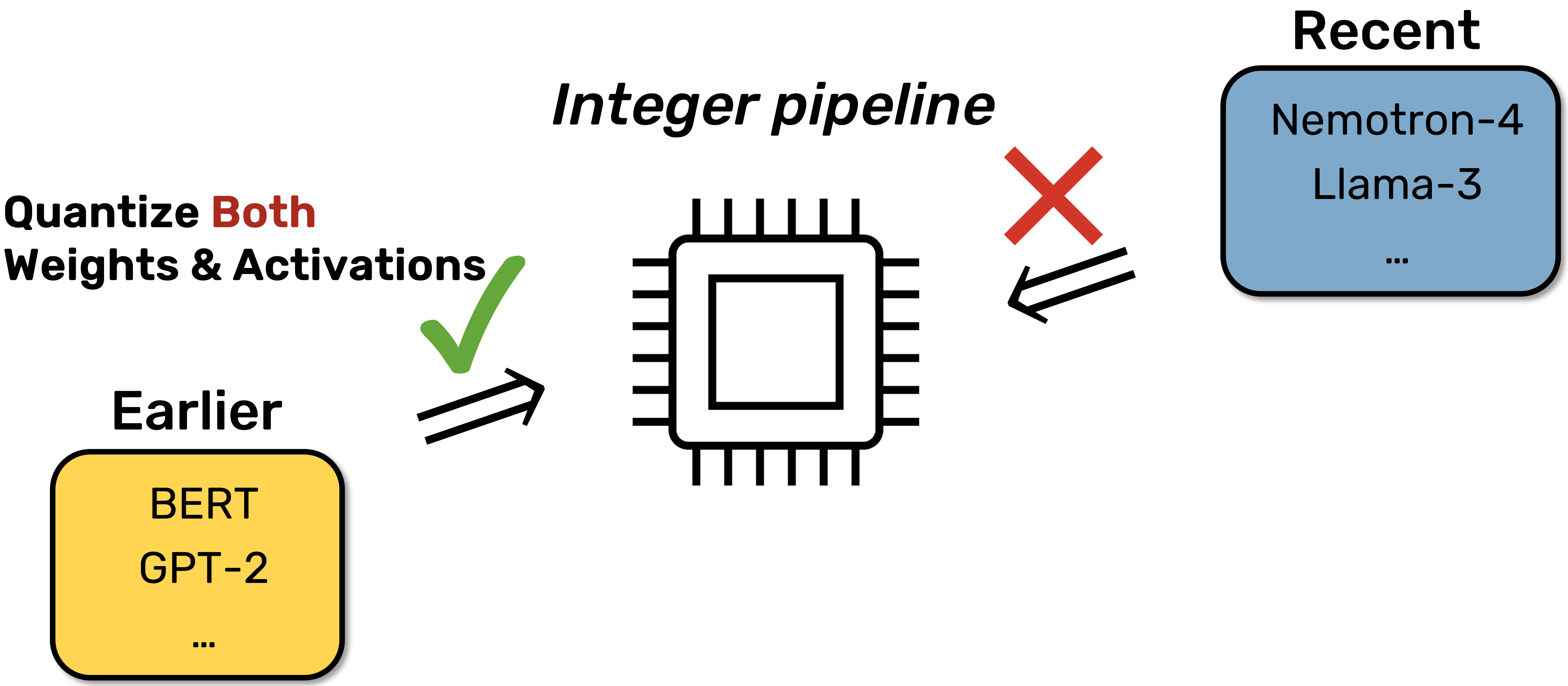
Challenges in LLM Inference



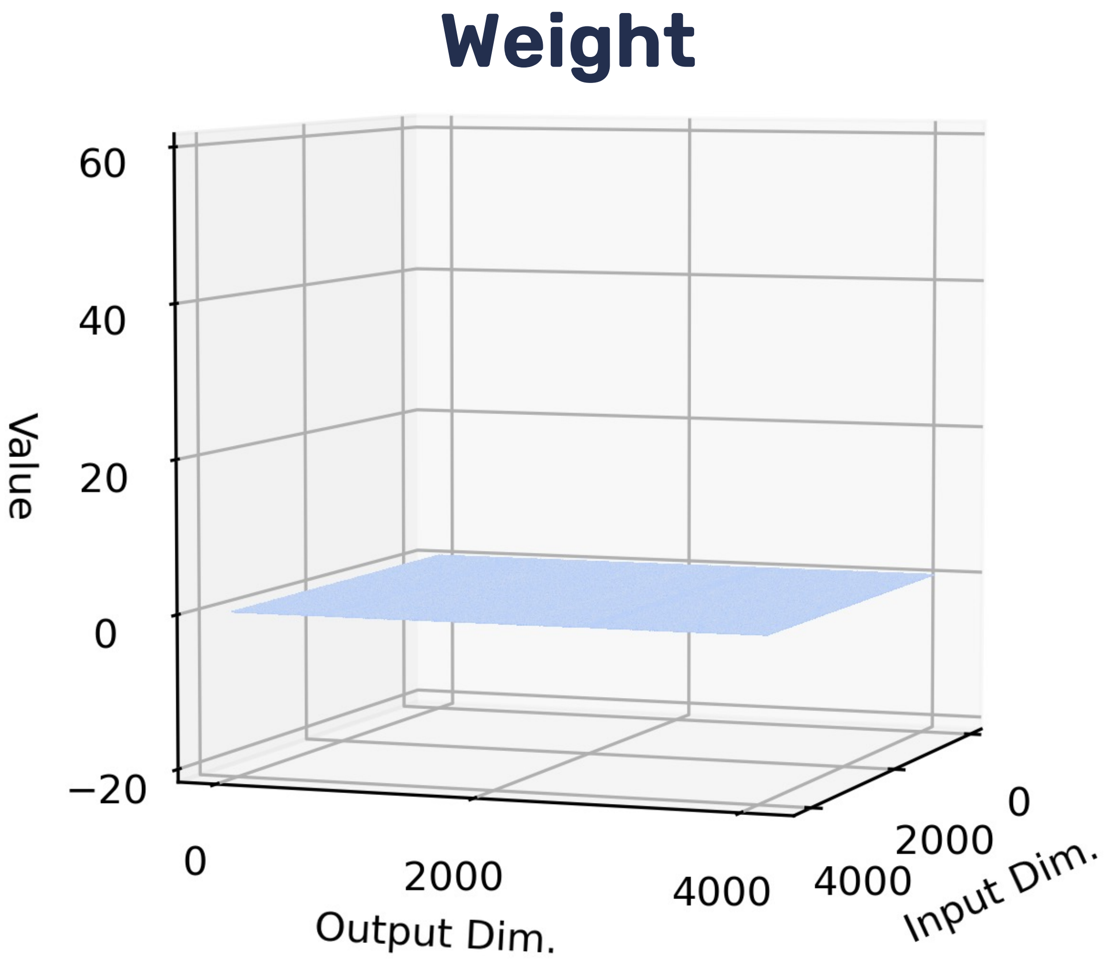
Challenges in LLM Inference

How do we serve LLMs efficiently?

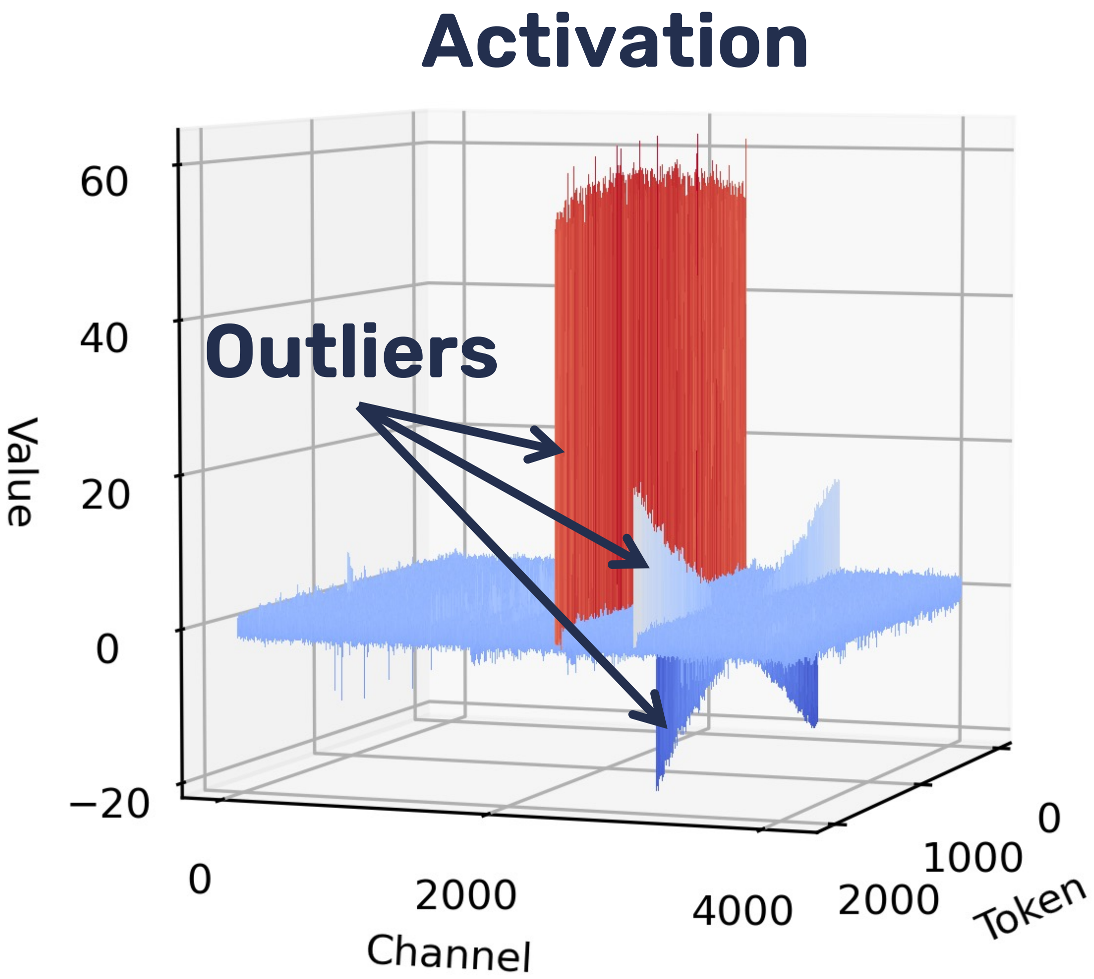
Challenges in LLM Inference



Activation Outliers in LLMs

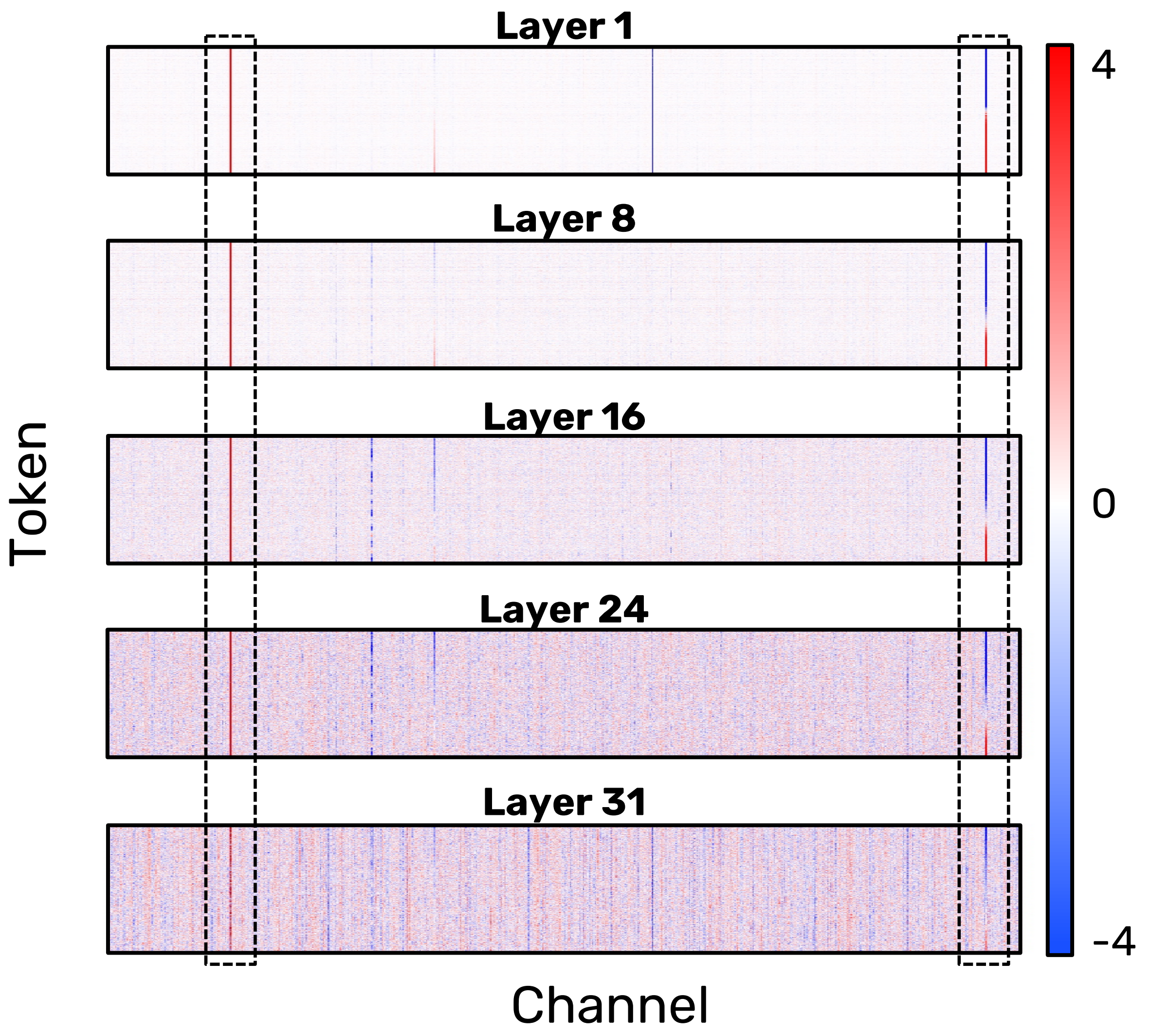


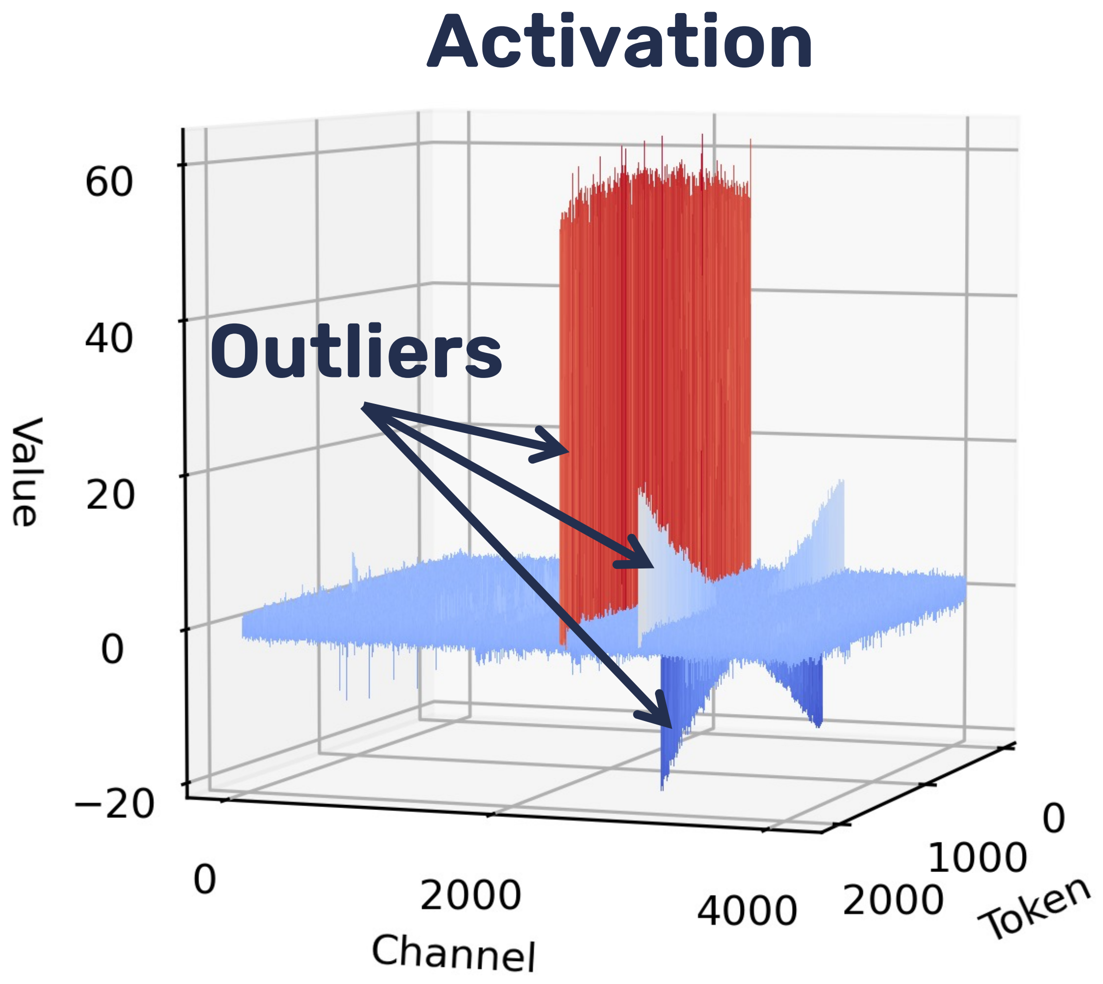
EASY to quantize!



HARD to quantize!

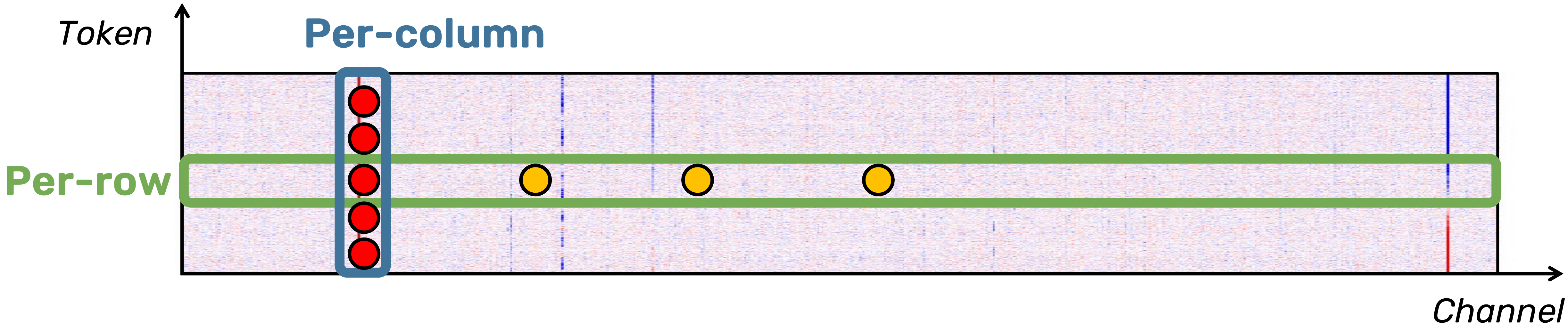
Activation Outliers in LLMs



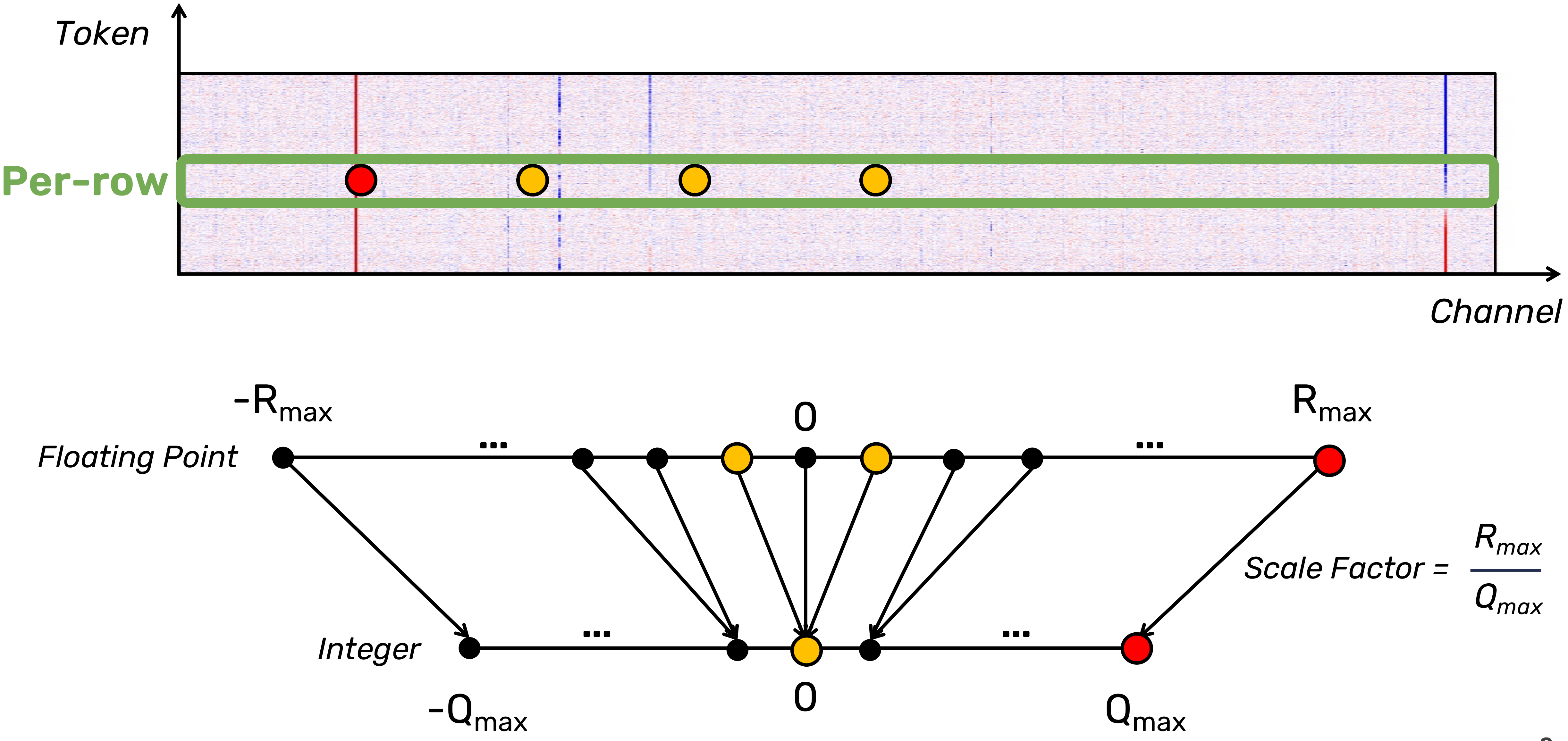


HARD to quantize!

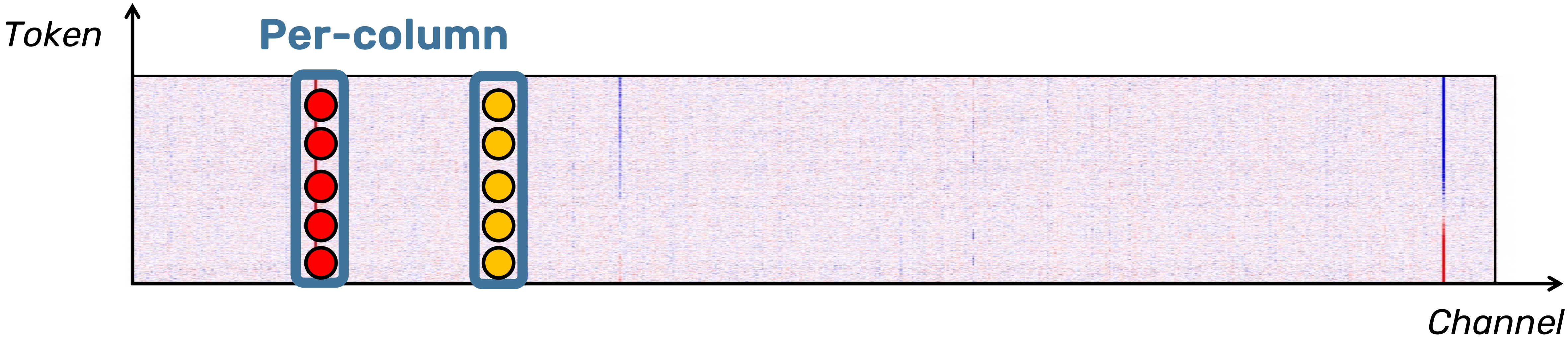
Activation Outliers in LLMs

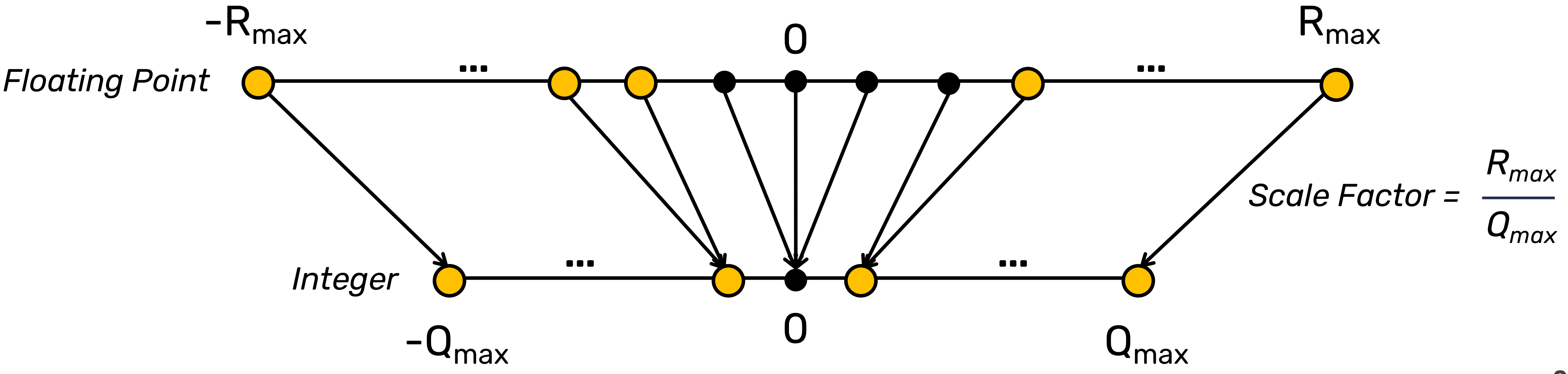


Activation Outliers in LLMs

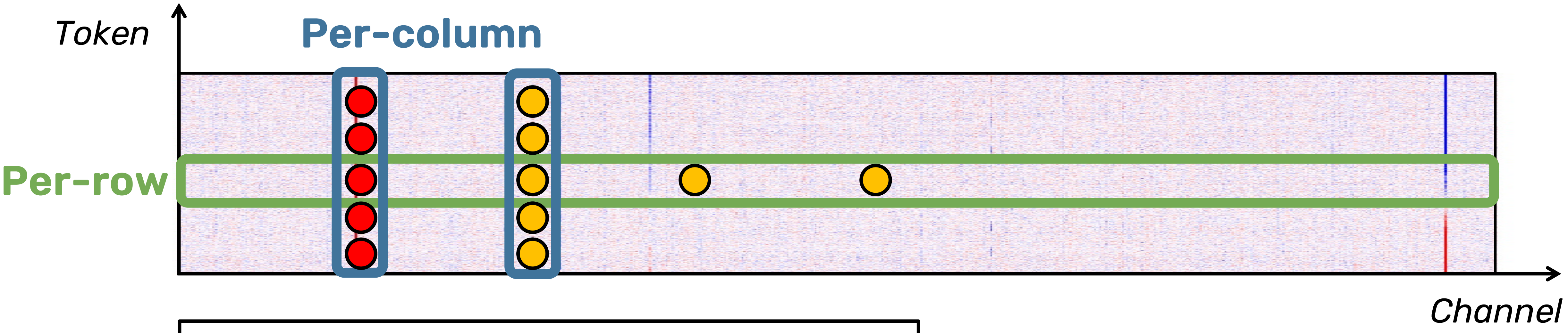


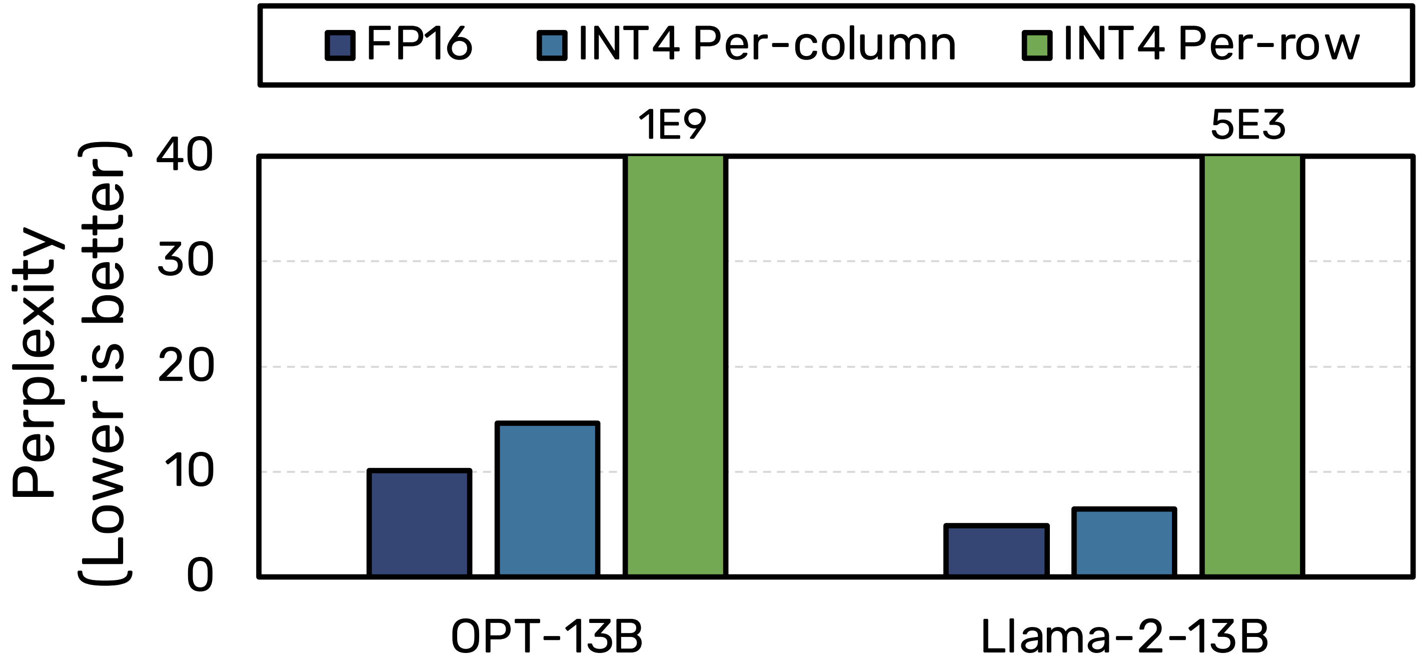
Activation Outliers in LLMs





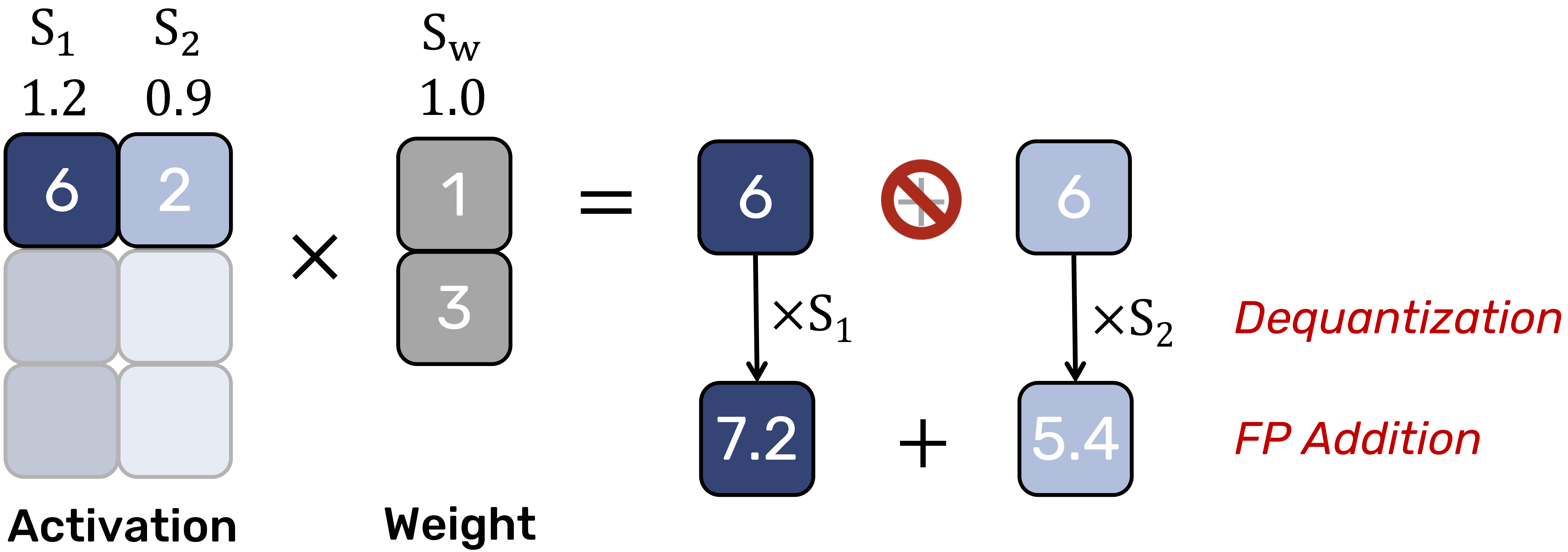
Activation Outliers in LLMs





Per-column isolates outliers
→ **Smaller Error**

Performance of Per-column Quantization

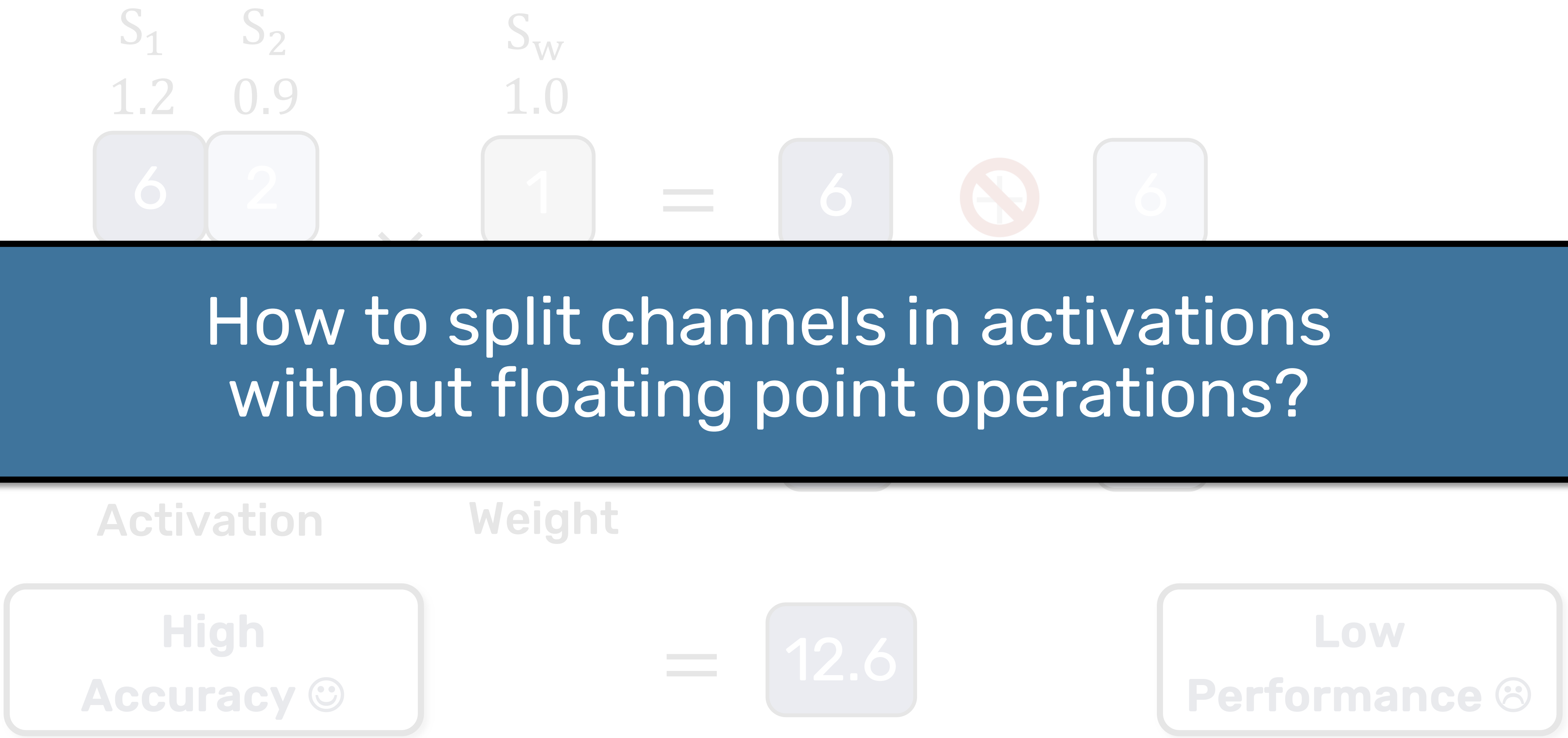


High Accuracy ☺

= 12.6

Low Performance ☹

Performance of Per-column Quantization



How to split channels in activations without floating point operations?

Limitations of Prior Works

Mixed Precision

LLM.int8() [NeurIPS’22]

OLAccel [ISCA’18]

DRQ [ISCA’20]

Custom & Multiple Types

OliVe [ISCA’23]

Limitations of Prior Works

Baseline

vs

Mixed Precision

LLM.int8() [NeurIPS'22]

OLAccel [ISCA'18]

DRQ [ISCA'20]

Custom & Multiple Types

OliVe [ISCA'23]

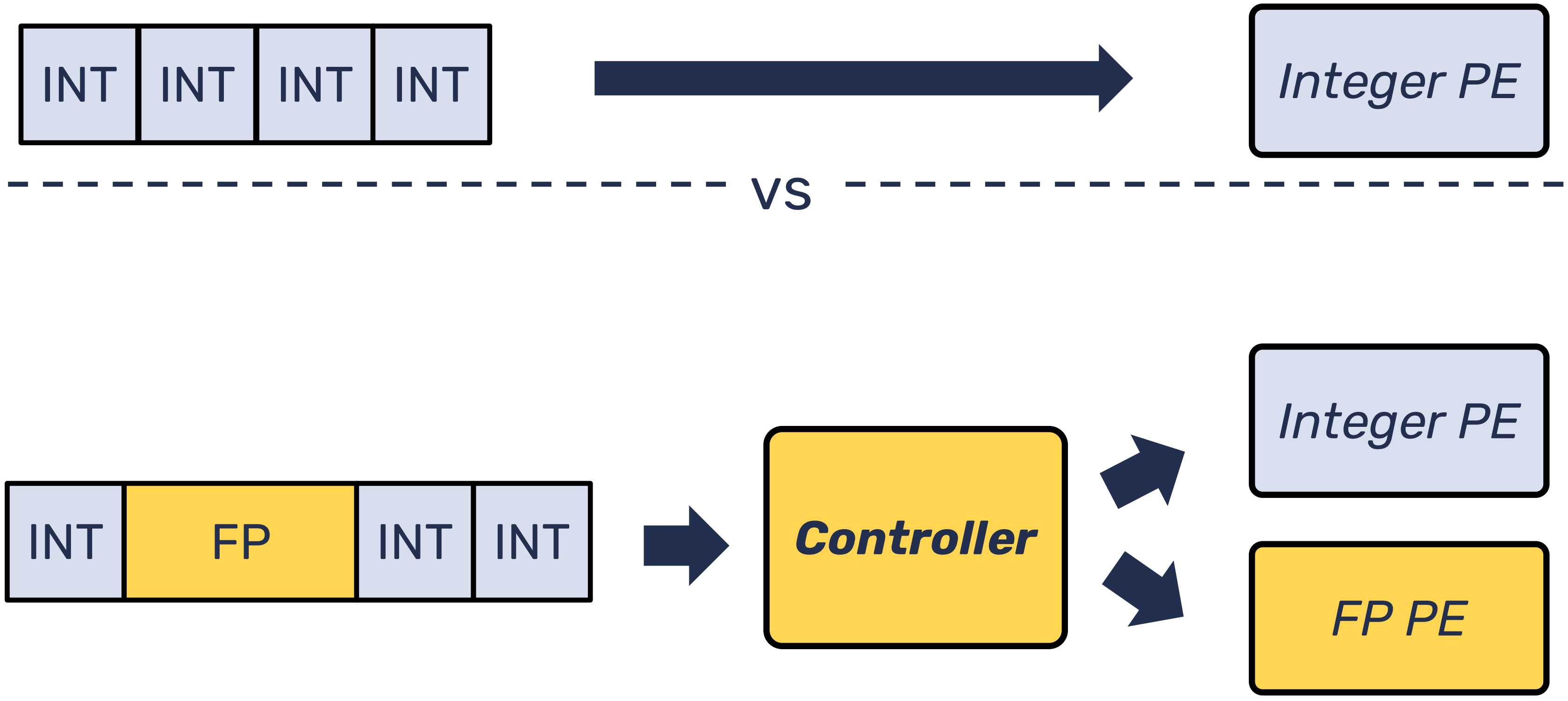
Limitations of Prior Works

Baseline

vs

Mixed Precision

LLM.int8() [NeurIPS'22]
OLAccel [ISCA'18]
DRQ [ISCA'20]

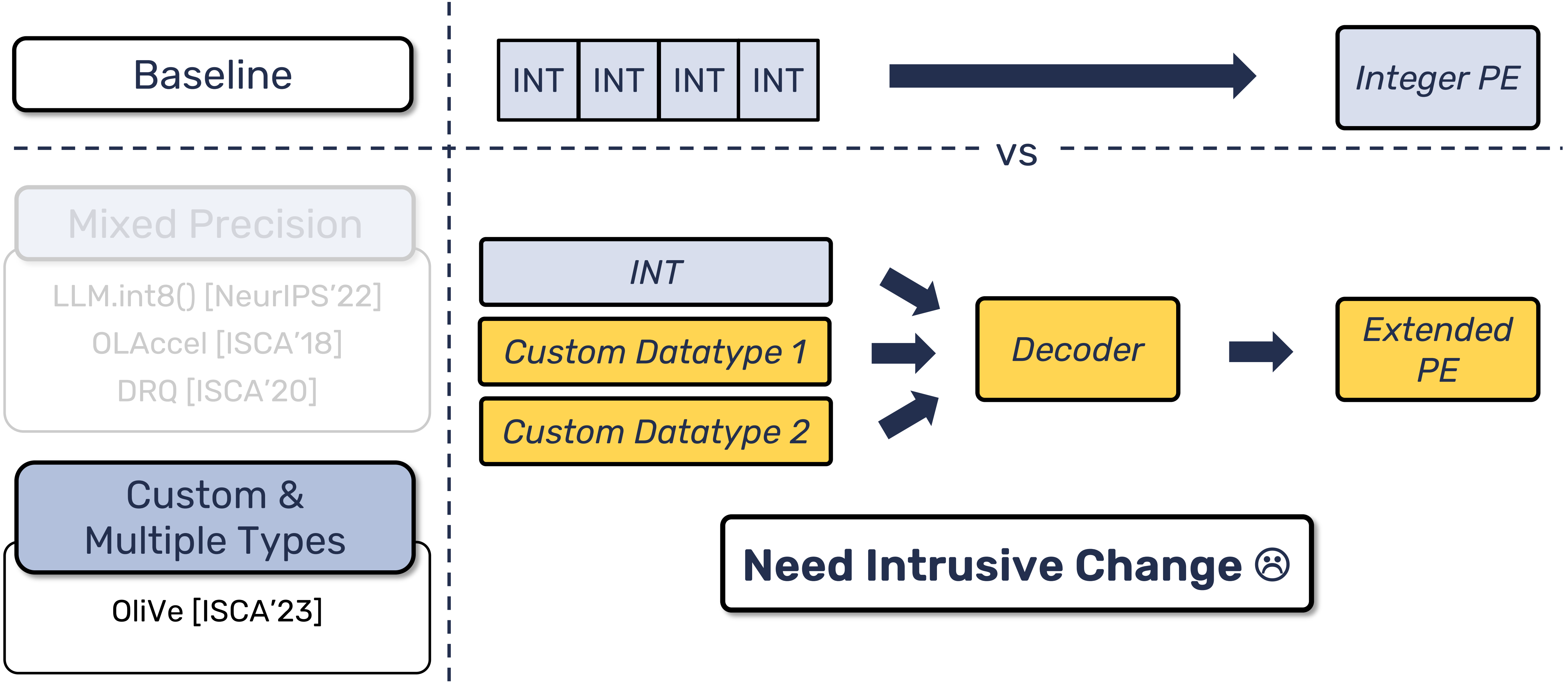


Complex Hardware ☹

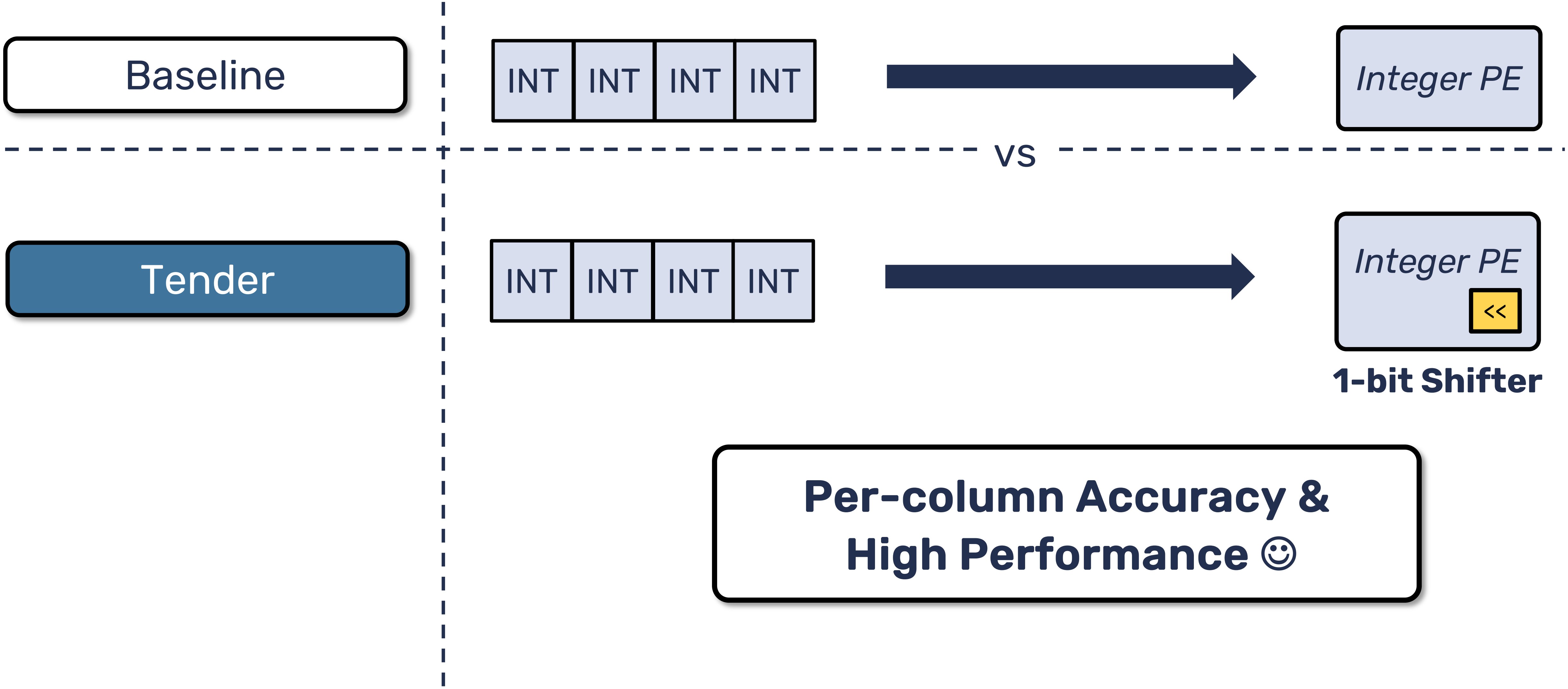
Custom & Multiple Types

OliVe [ISCA'23]

Limitations of Prior Works



Tender Overview



Outline

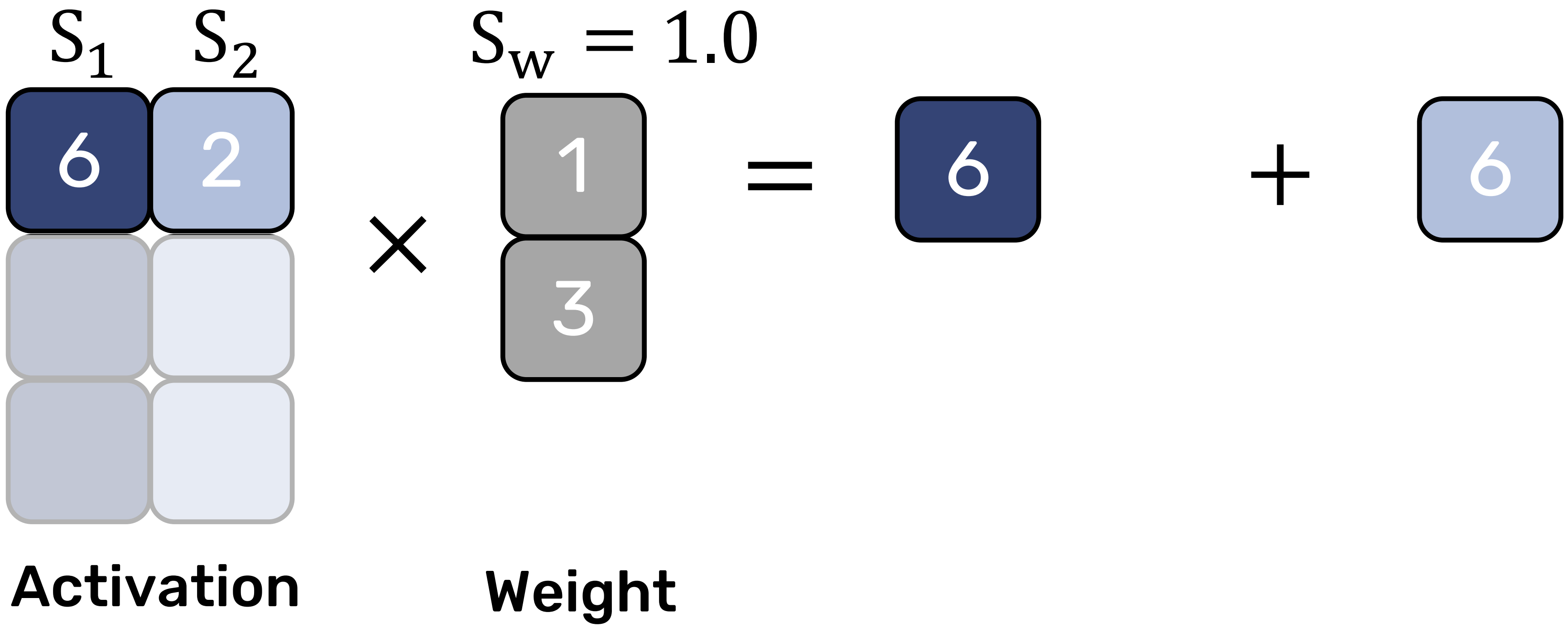
- **Motivation**
  - Challenges in Efficient LLM Inference
  - Limitations of Prior Works

- **Tender: Algorithm-Hardware Co-design for Efficient LLM Inference**
  - Tensor Decomposition
  - Rescaling Operation

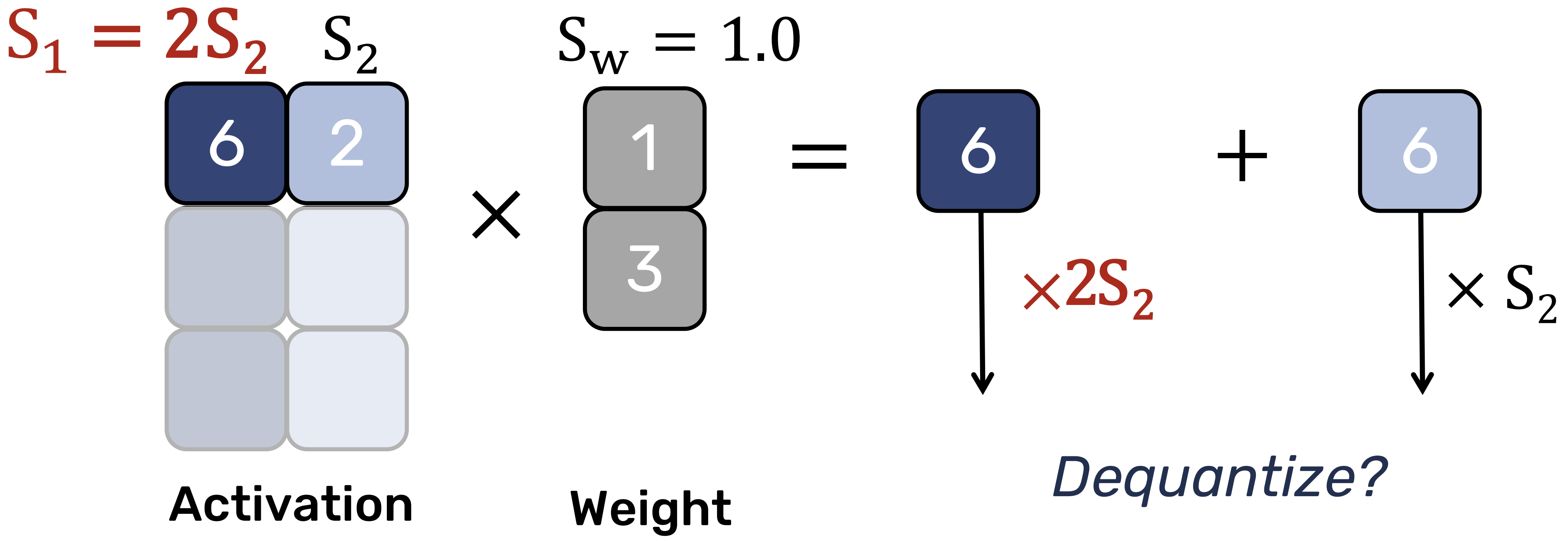
- **Evaluation**

- **Conclusion**

Key Insight

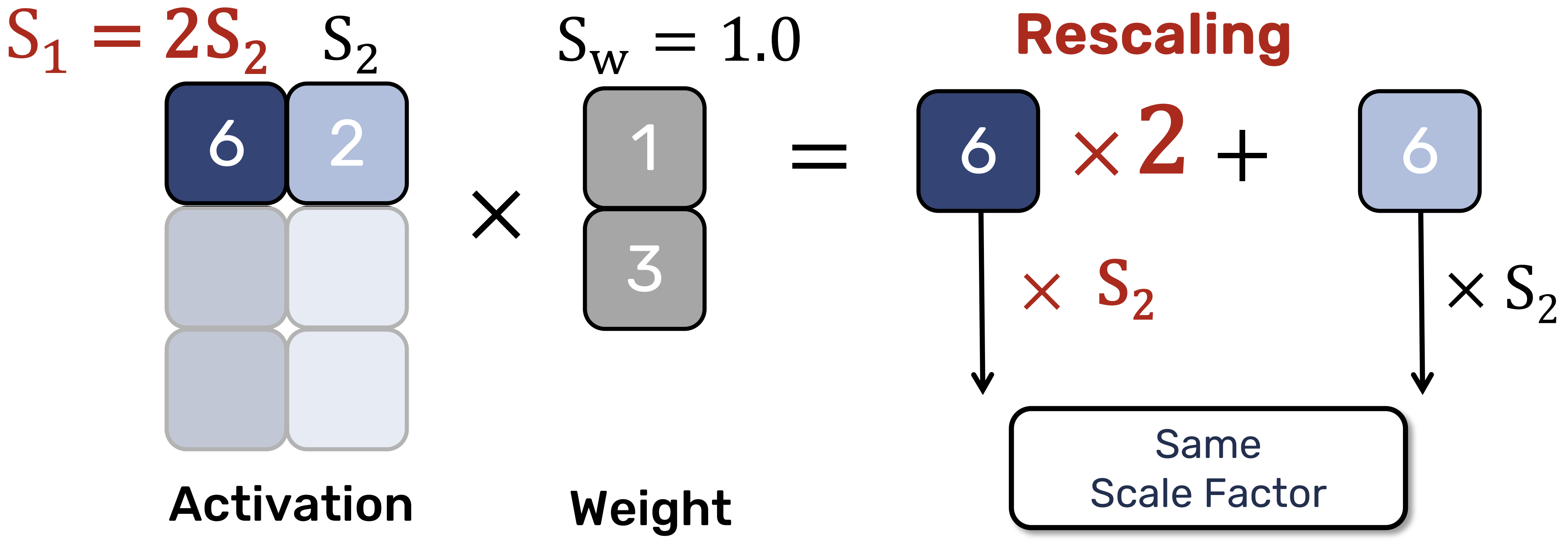


Key Insight

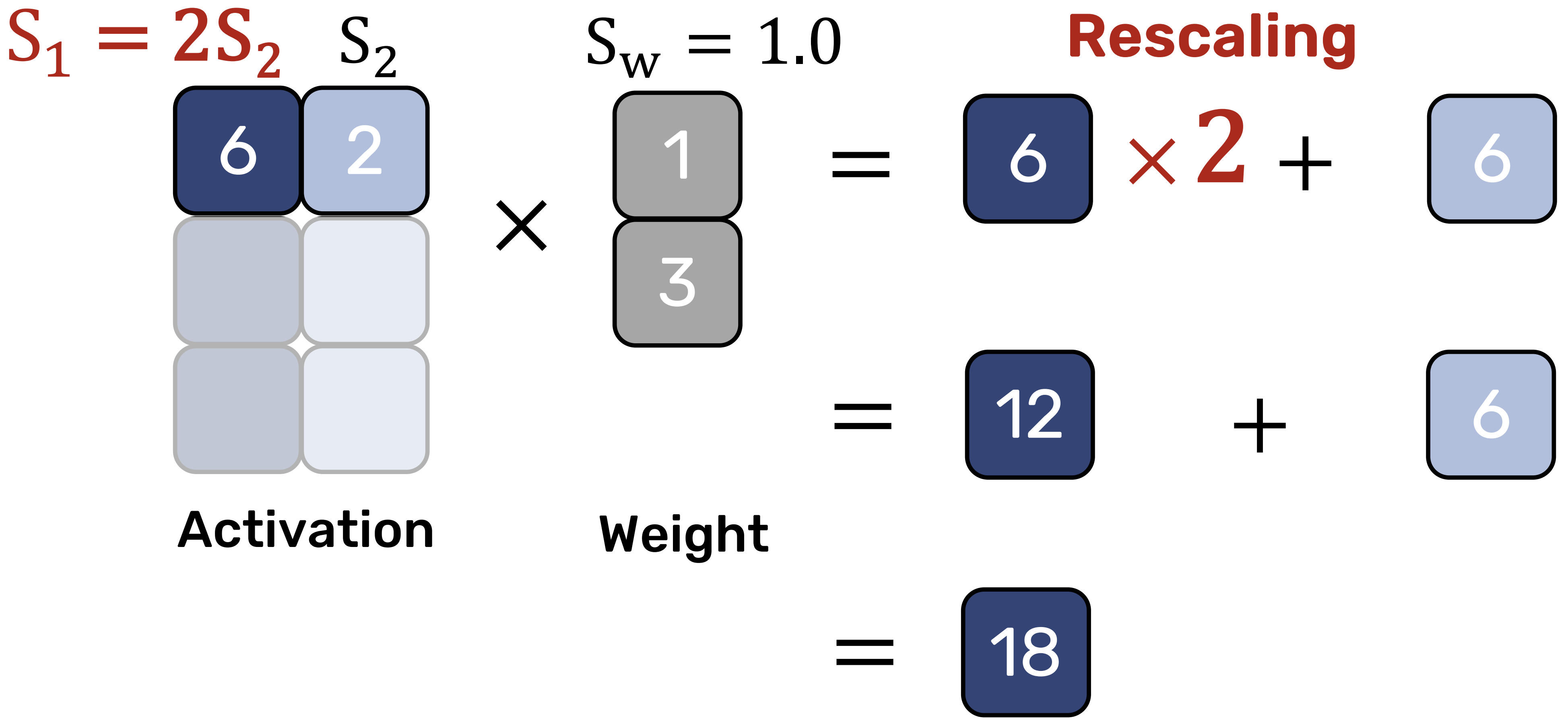


$S_1 = 2S_2$
S_2
$S_w = 1.0$
6
2
×
1
3
=
6
+
6
$\times 2S_2$
$\times S_2$
Activation
Weight
Dequantize?

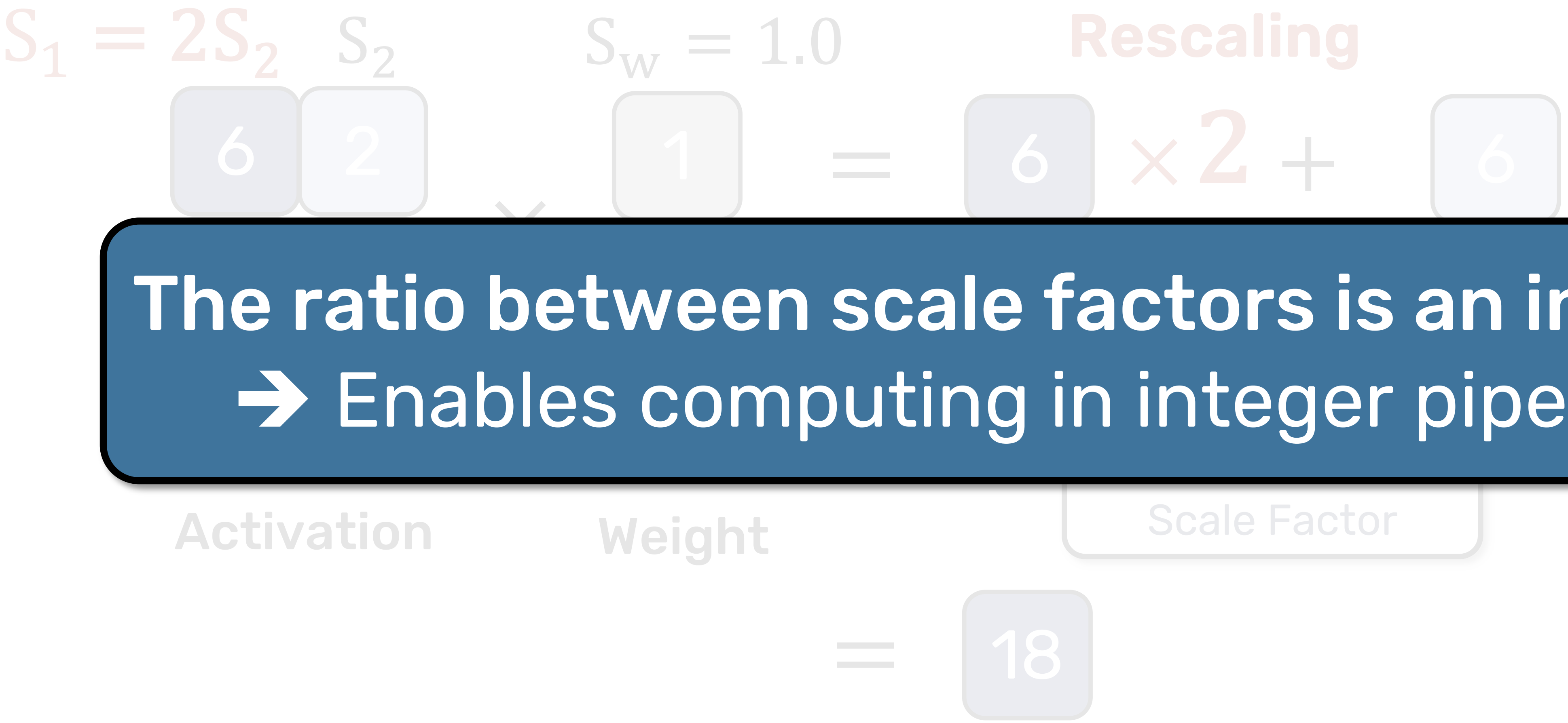
Key Insight


$S_1 = 2S_2$
S_2
6
2
Activation
×
$S_w = 1.0$
1
3
Weight
=
Rescaling
6
$\times 2 +$
6
$\times S_2$
$\times S_2$
Same
Scale Factor

Key Insight



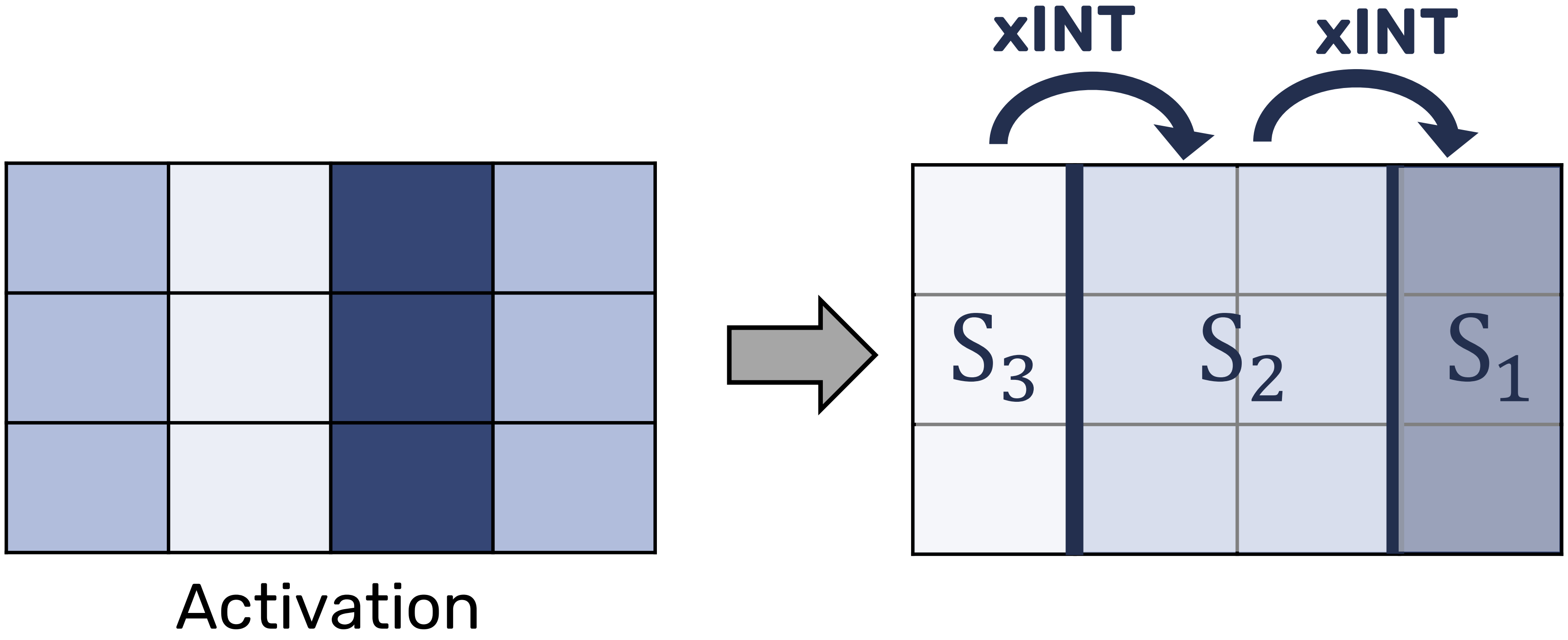
Key Insight



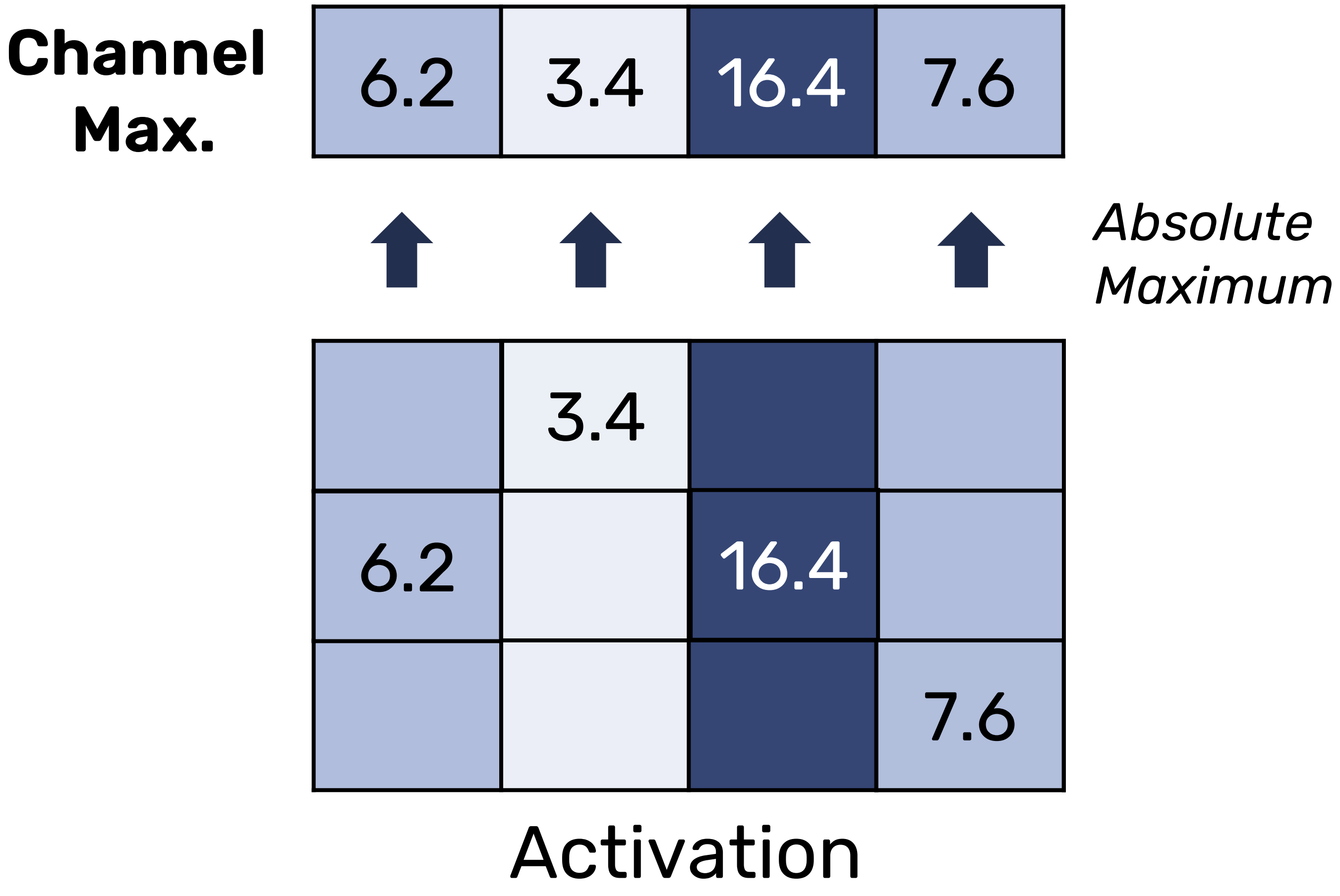
The ratio between scale factors is an integer

➔ Enables computing in integer pipeline

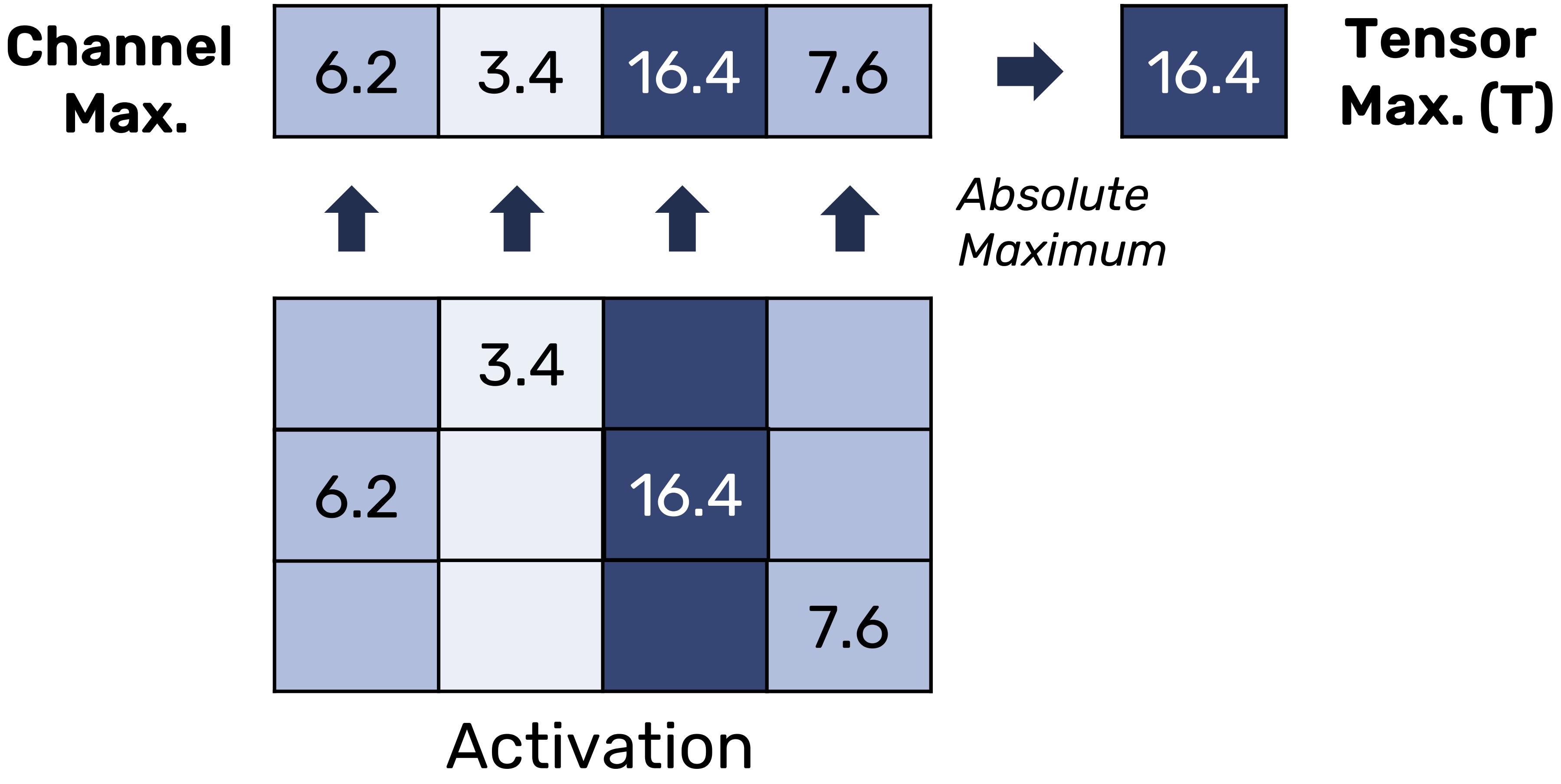
Tender: Tensor Decomposition



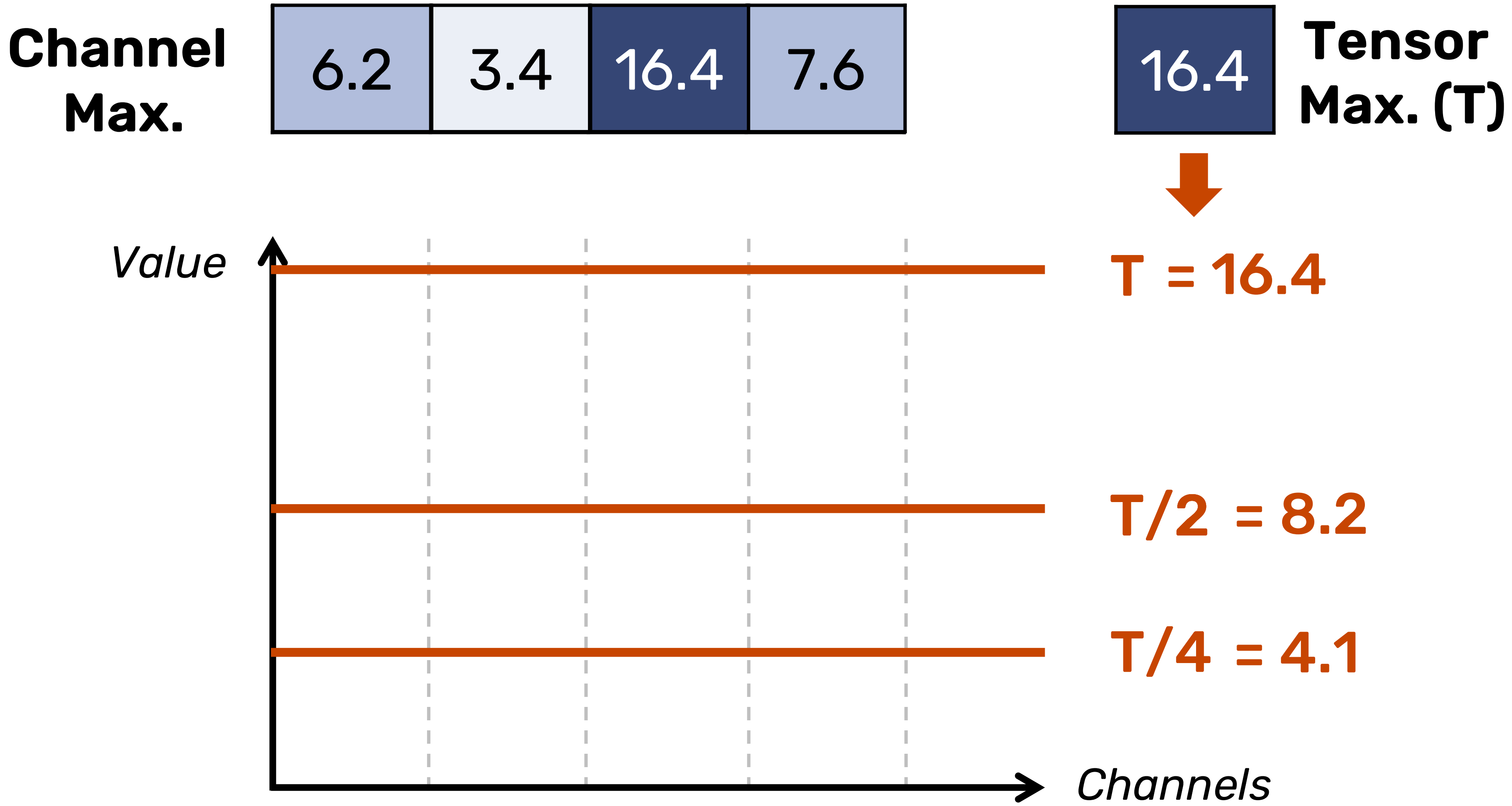
Tender: Tensor Decomposition



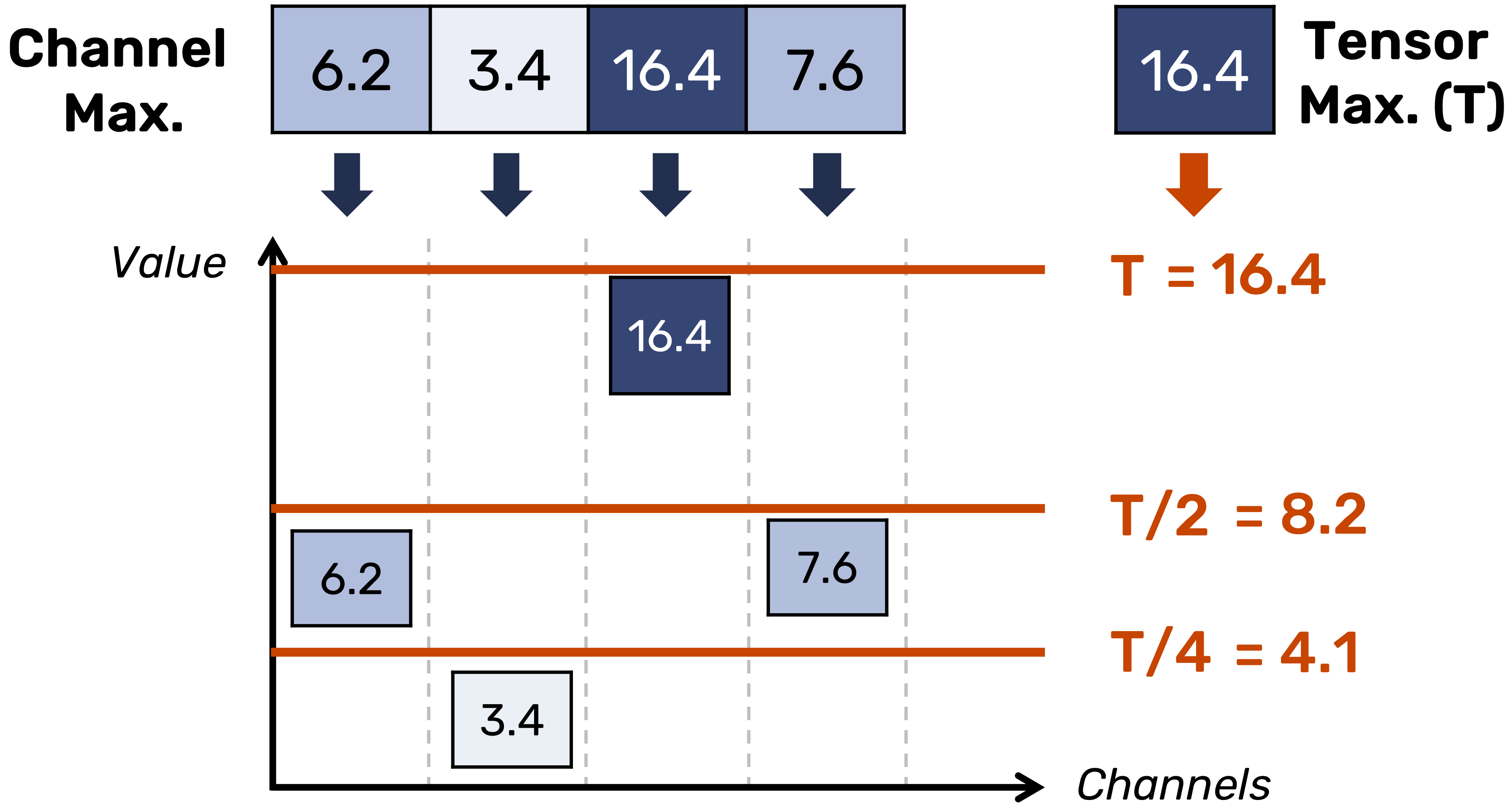
Tender: Tensor Decomposition



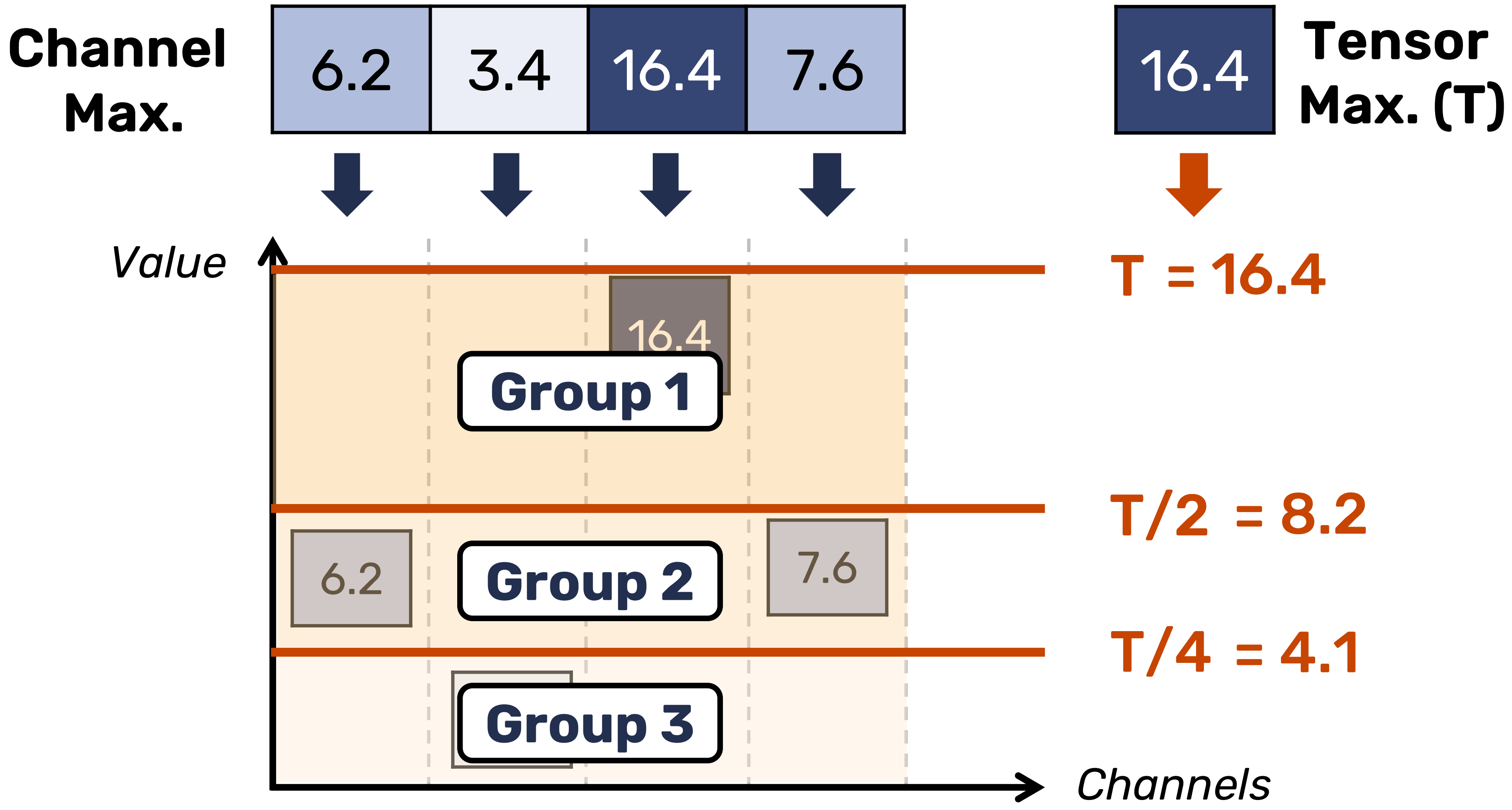
Tender: Tensor Decomposition



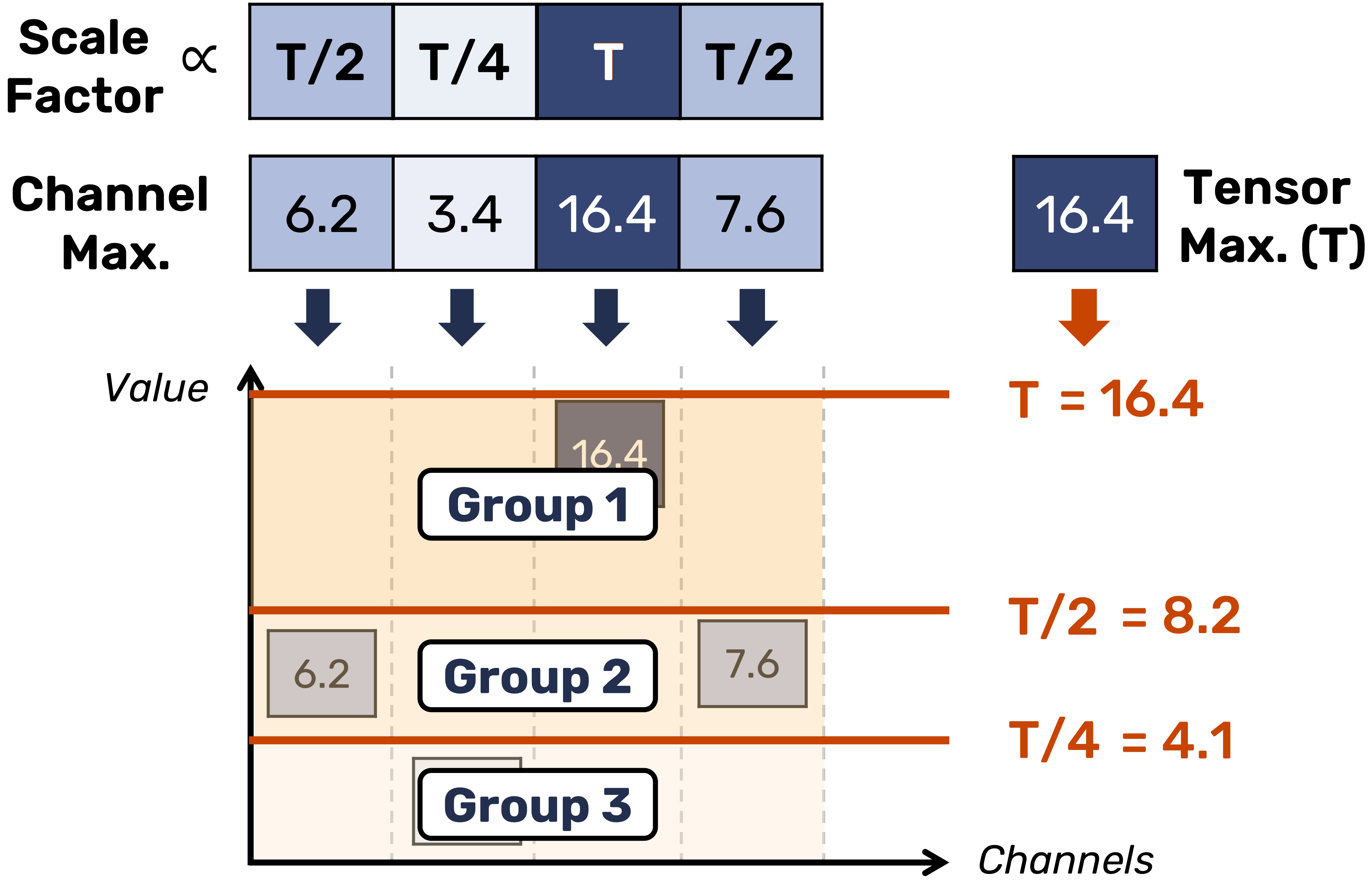
Tender: Tensor Decomposition



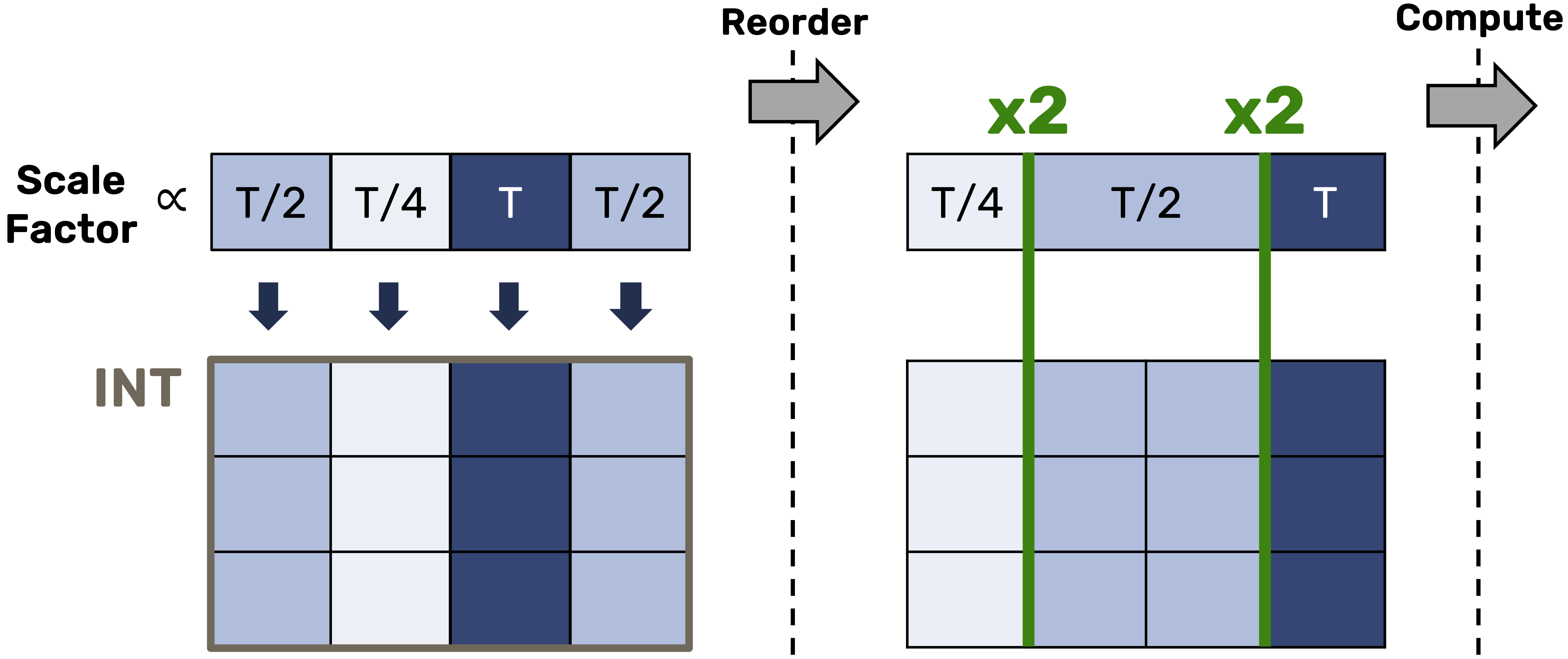
Tender: Tensor Decomposition



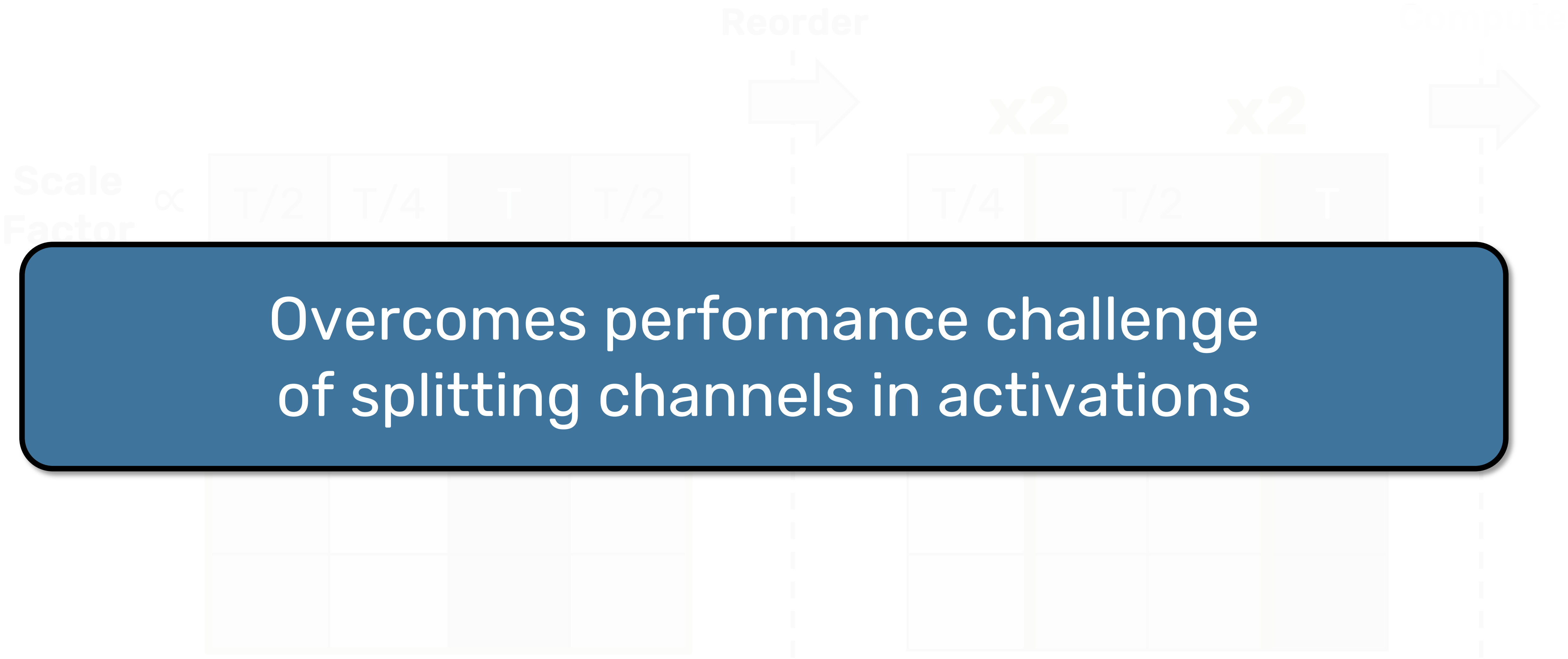
Tender: Tensor Decomposition



Tender: Tensor Decomposition



Tender: Tensor Decomposition



Overcomes performance challenge of splitting channels in activations

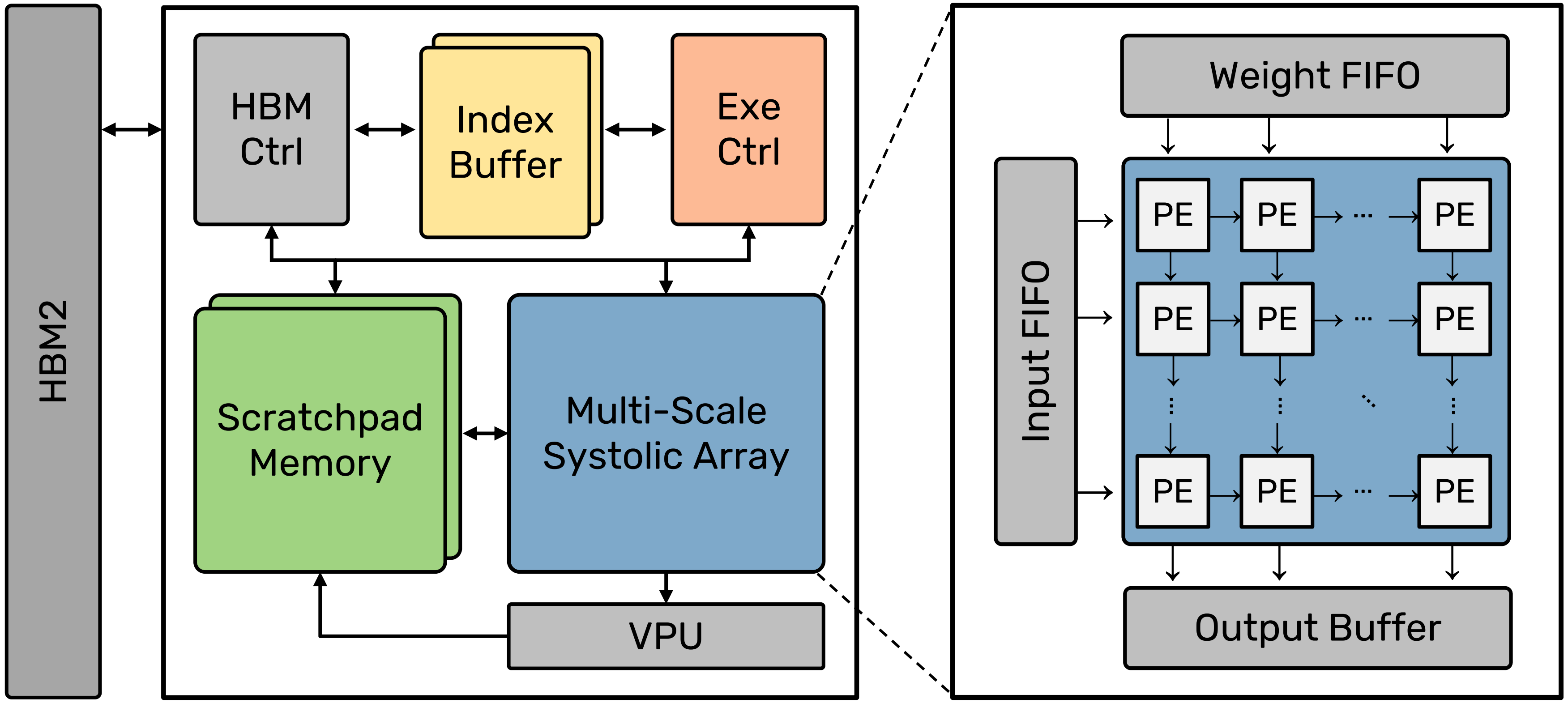
Tender: Architecture Overview

Execution Controller

- Column reordering

Multi-Scale Systolic Array (MSA)

- Computation with Rescaling



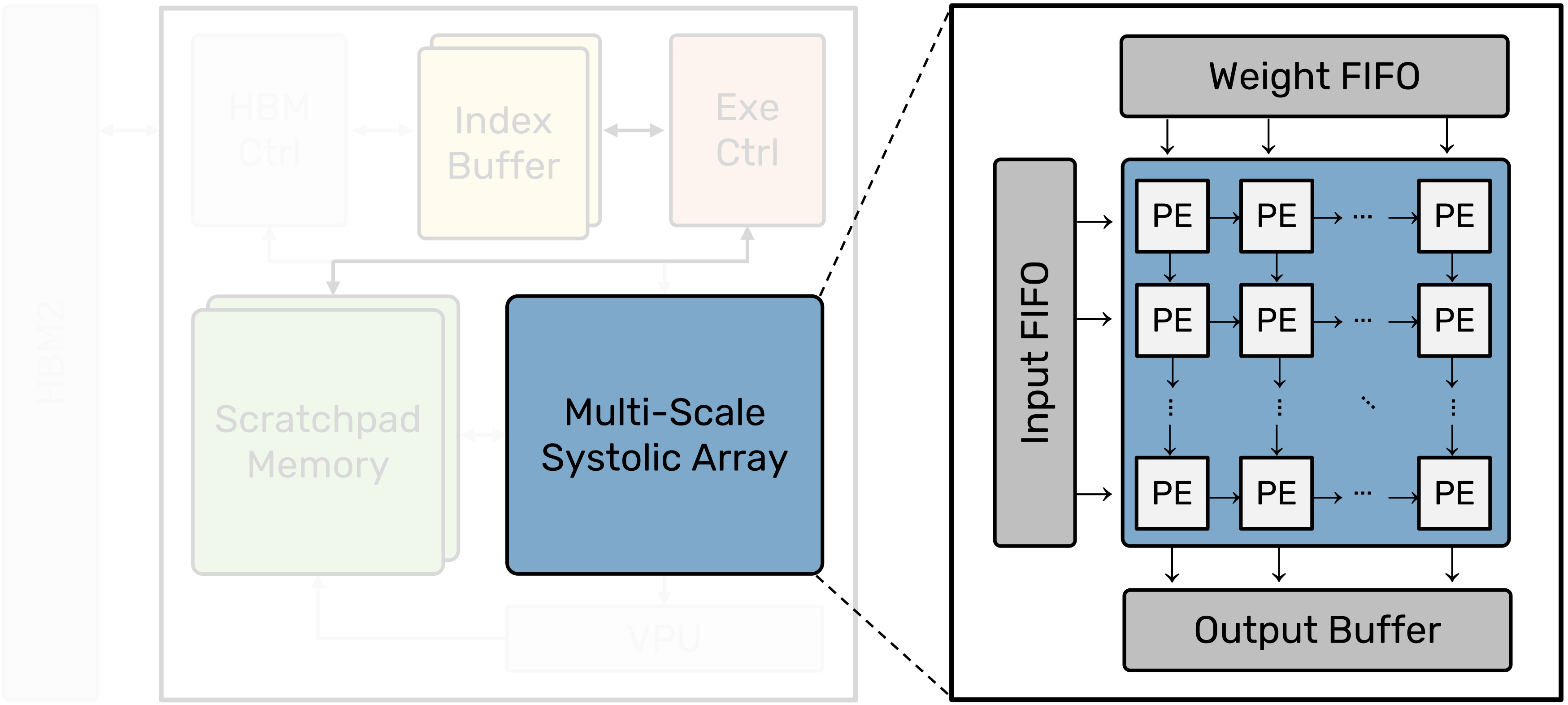
Tender: Architecture Overview

Execution Controller

- Column reordering

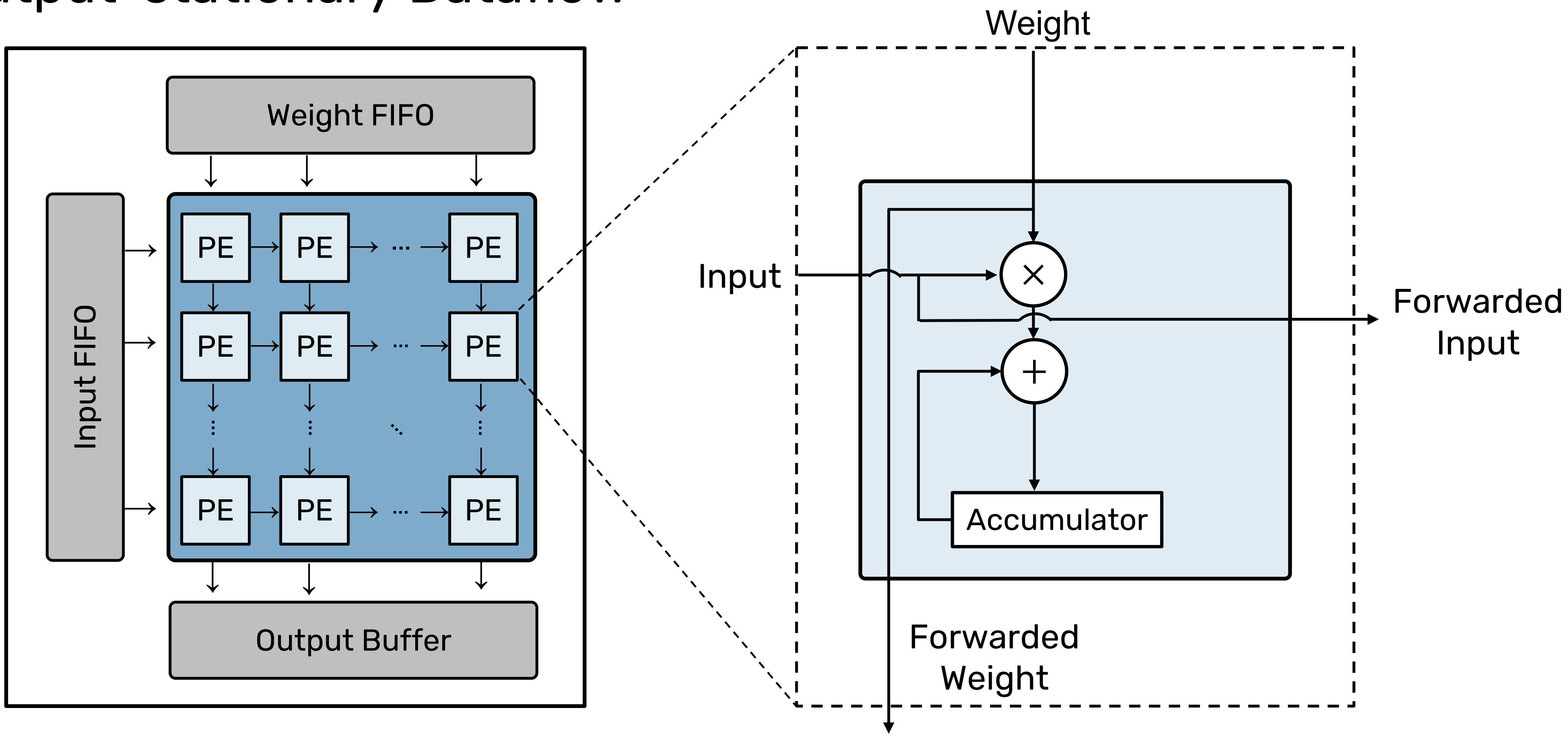
Multi-Scale Systolic Array (MSA)

- Computation with Rescaling



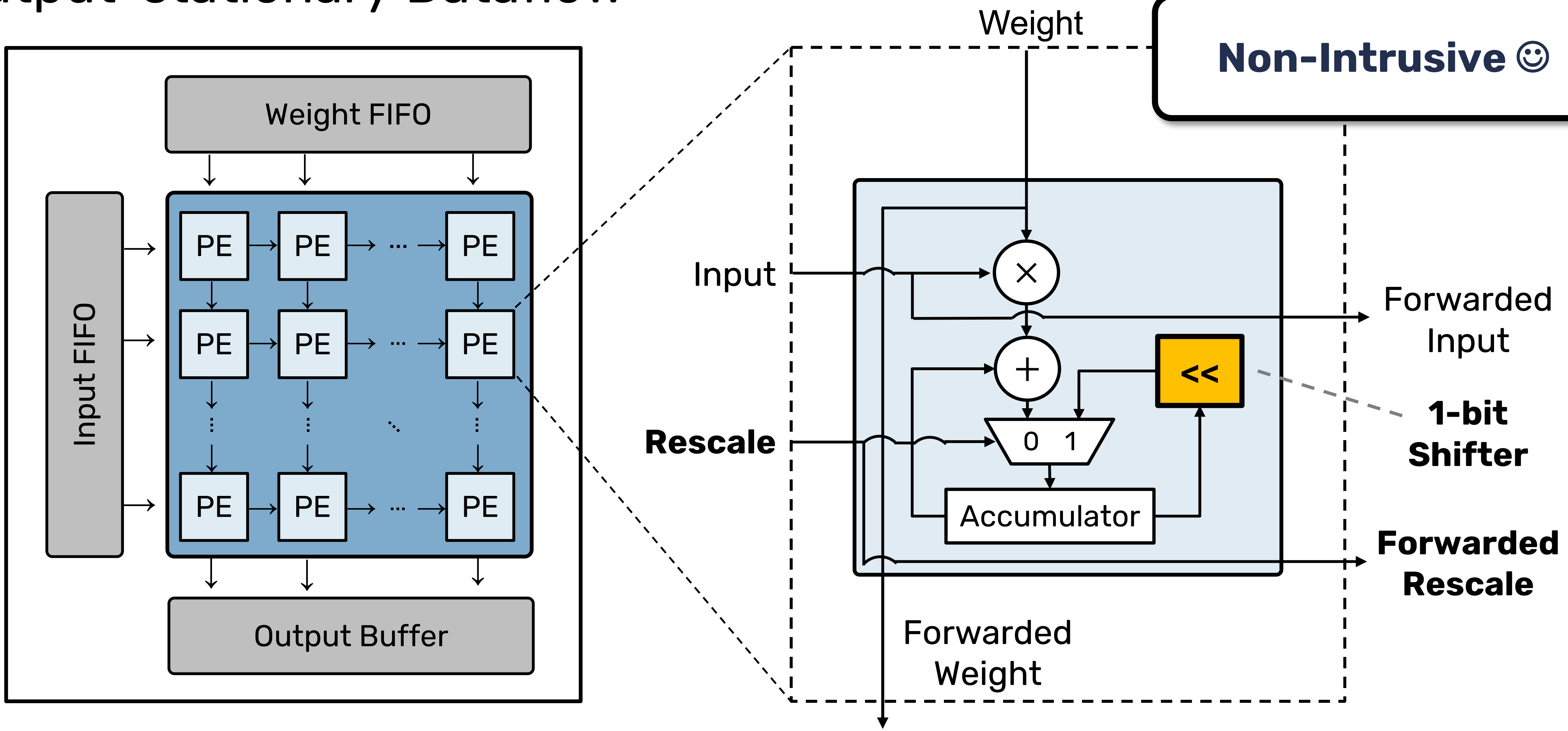
Multi-Scale Systolic Array (MSA)

Output-stationary Dataflow



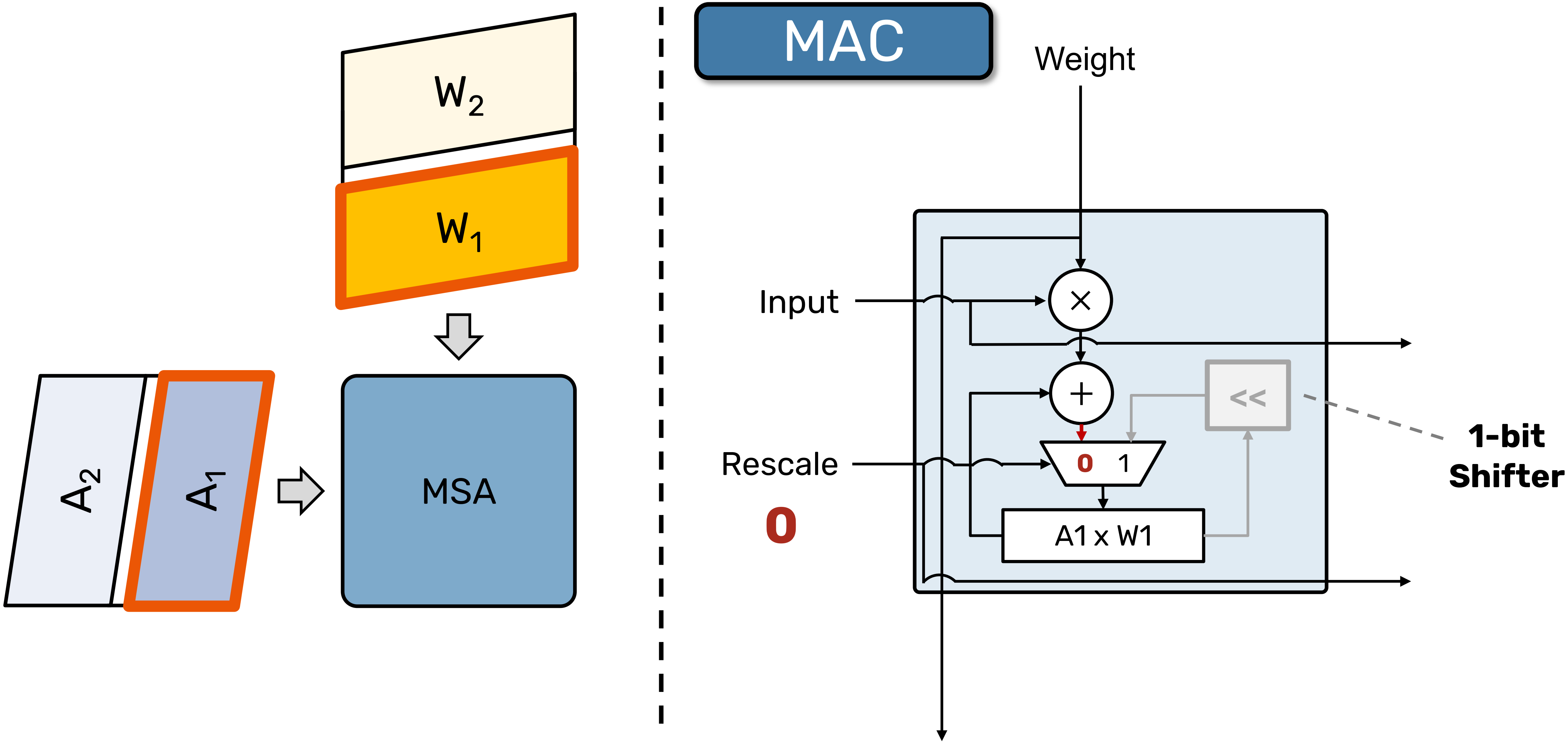
Multi-Scale Systolic Array (MSA)

Output-stationary Dataflow

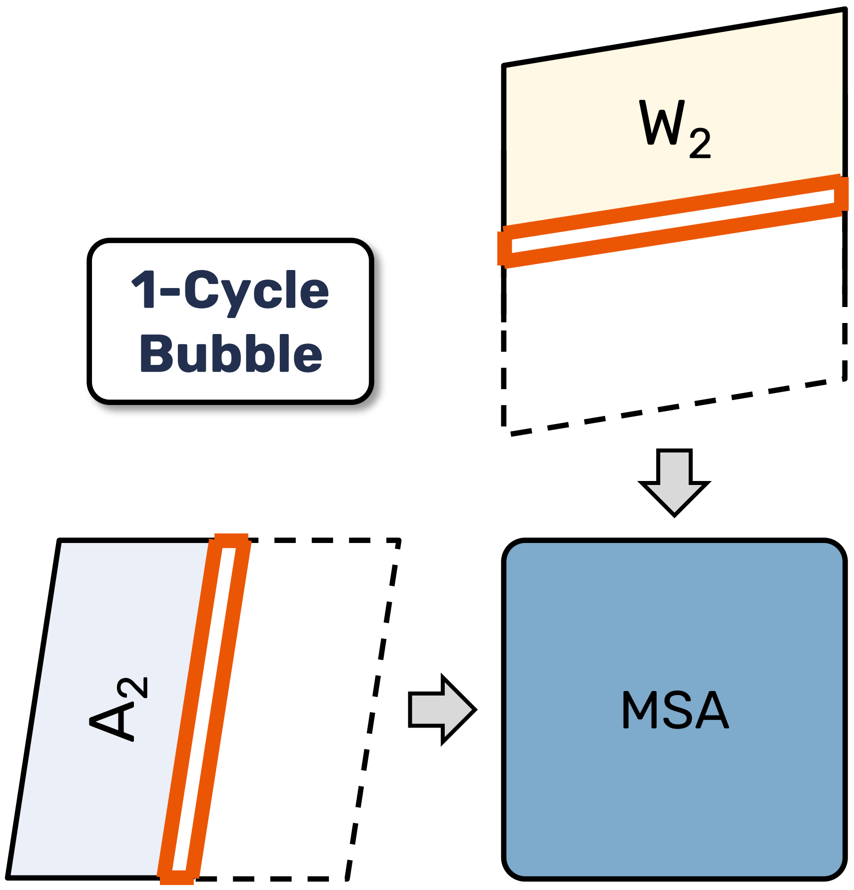


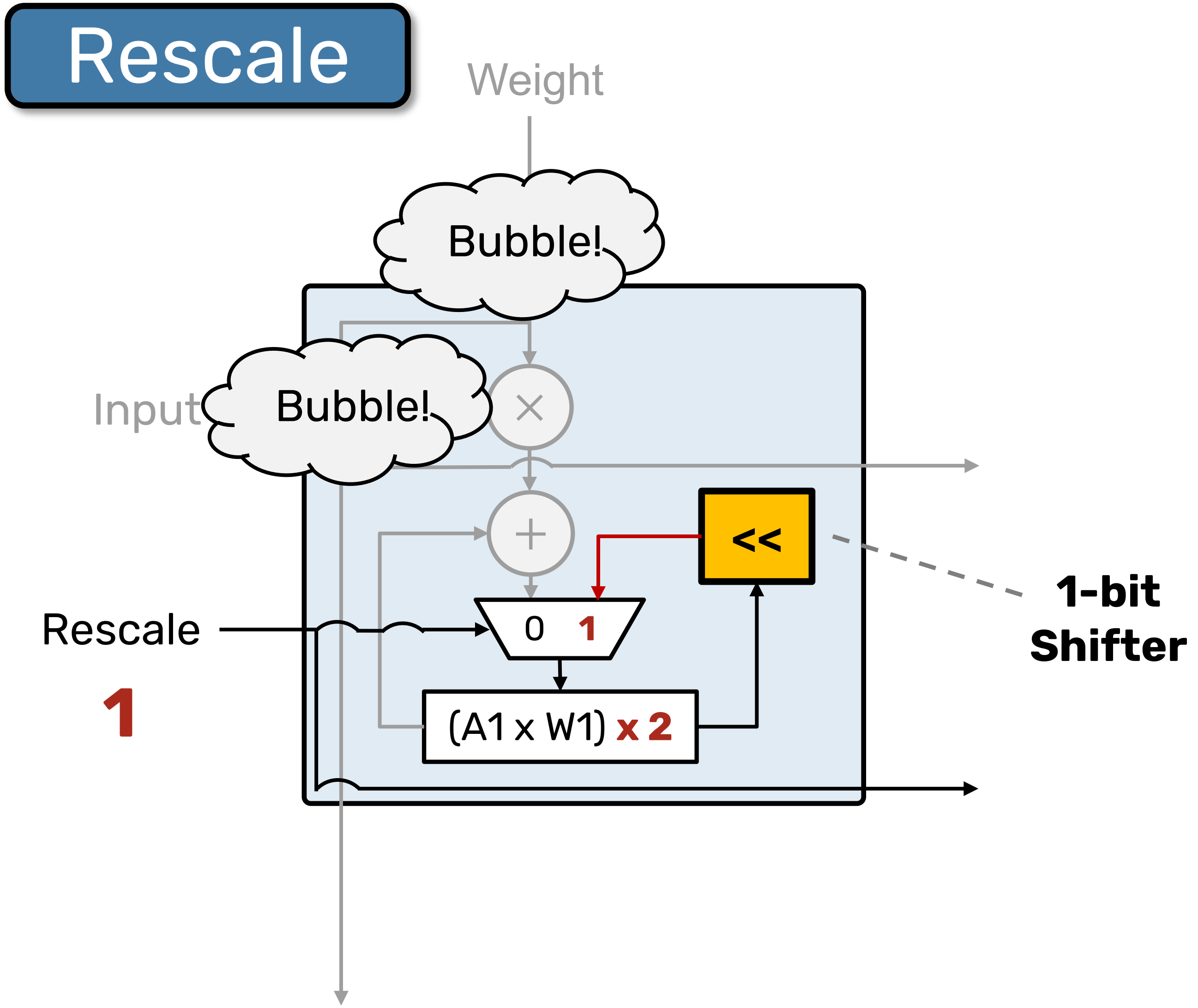
Non-Intrusive ☺

Multi-Scale Systolic Array (MSA)



Multi-Scale Systolic Array (MSA)





Outline

- Motivation
  - Challenges in Efficient LLM Inference
  - Limitations of Prior Works

- **Tender**: Algorithm-Hardware Co-design for Efficient LLM Inference
  - Tensor Decomposition
  - Rescaling Operation

- Evaluation

- Conclusion

Methodology

Models

- OPT, LLaMA, and Llama-2

Datasets

- WikiText-2 and Penn Treebank

Accuracy

- Hugging Face Library

Performance

- RTL: 28nm technology
- Cycle-level simulator

Baselines

Accuracy	
SmoothQuant	Column-wise scaling
ANT	Adaptive & Custom Types
OliVe	Adaptive & Custom Types

Performance	
OLAccel	Input - Mixed Precision
ANT	Input - Exponent & Integer
OliVe	Input - Exponent & Integer

Quantization Results

Perplexity results using *WikiText-2* dataset

* Lower is better

Precision	Scheme	OPT-66B	Llama-2-70B
FP16	Base	9.34	3.32
INT8	SmoothQuant	9.87	17.30
	OliVe	**9.43**	50.94
	Tender	**9.43**	**3.48**

Isolation of outliers

Quantization Results

Perplexity results using *WikiText-2* dataset

* Lower is better

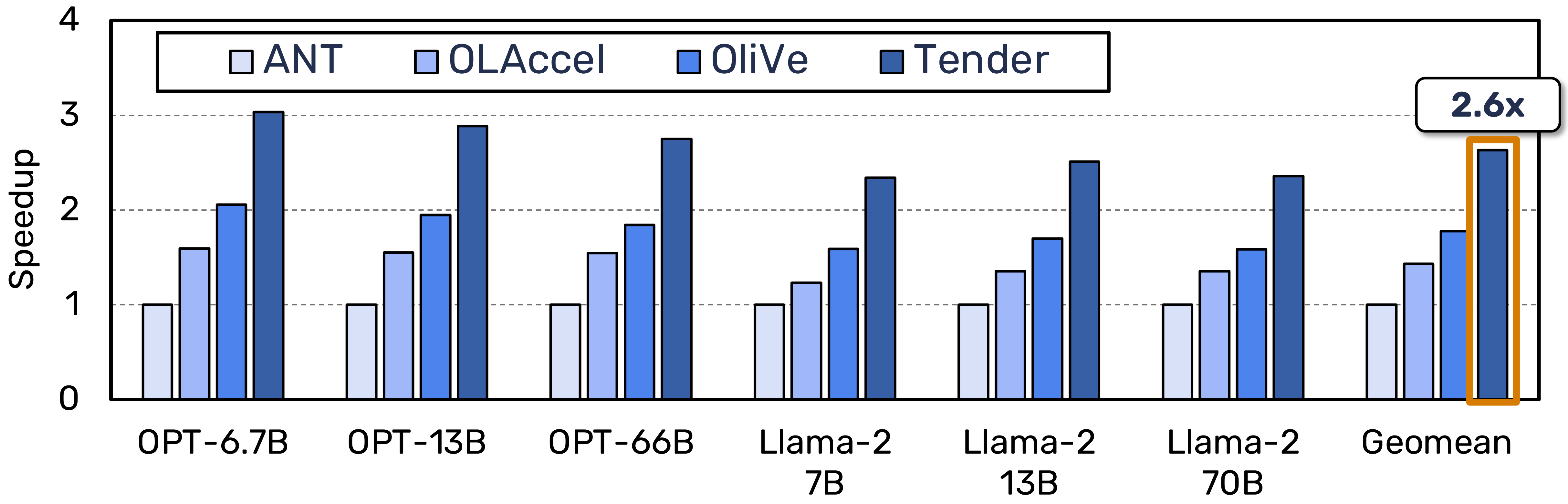
Precision	Scheme	OPT-66B	Llama-2-70B
FP16	Base	9.34	3.32
INT8	SmoothQuant	9.87	17.30
	OliVe	**9.43**	50.94
	Tender	**9.43**	**3.48**
INT4	SmoothQuant	6E+4	7E+4
	OliVe	6E+3	99.91
	Tender	**12.38**	**13.43**

Isolation of outliers

More **important** in **low-precision**

Performance

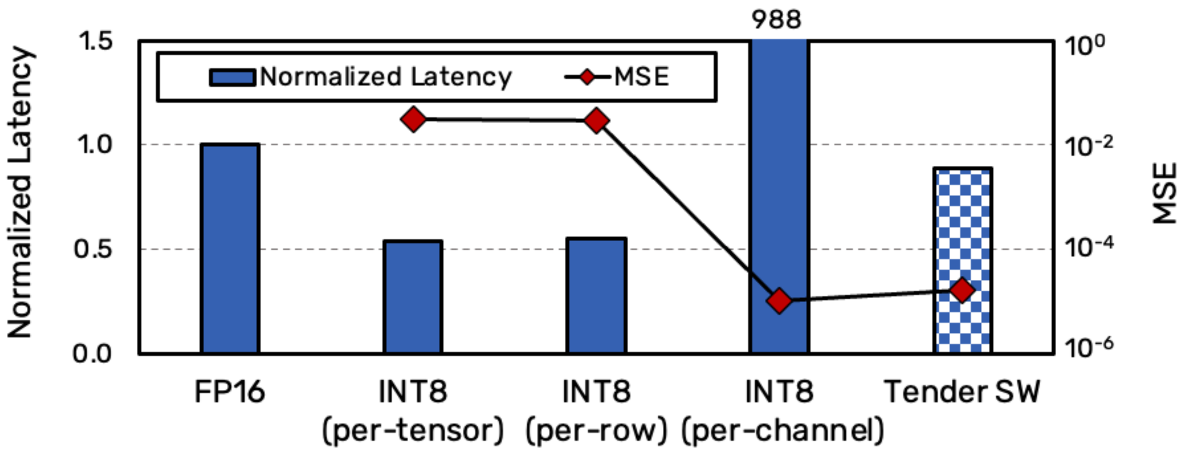
LLM inference **speedup**



→ With **higher accuracy**, Tender achieves **higher performance**

More Details in Our Paper

- GPU Implementation of Tender
- Tender on weight-stationary dataflow



- Hardware support for reordering
- Comparison with BFP variants
  - MSFP and MX formats
- Area & Energy Efficiency
- Others…

Conclusion

Problem

- Outliers make an efficient serving of LLM challenging
- Complex and intrusive design of prior works

Solution: Tender, efficient low-bit integer-based LLM inference accelerator

- Tensor decomposition while considering accuracy and performance
- Rescaling only requires a 1-bit shifter and 1-cycle latency

Result

- Tender achieves up to an average of **2.6x speedup** over the baseline with **substantially higher accuracy** ☺

Thank You!

Tender

Accelerating Large Language Models via Tensor Decomposition And Runtime Requantization

Jungi Lee (jungi.lee@snu.ac.kr)

Department of ECE
Seoul National University