Atom: Low-Bit Quantization for Efficient and Accurate LLM Serving

Yilong Zhao, **Chien-Yu Lin**, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, Baris Kasikci

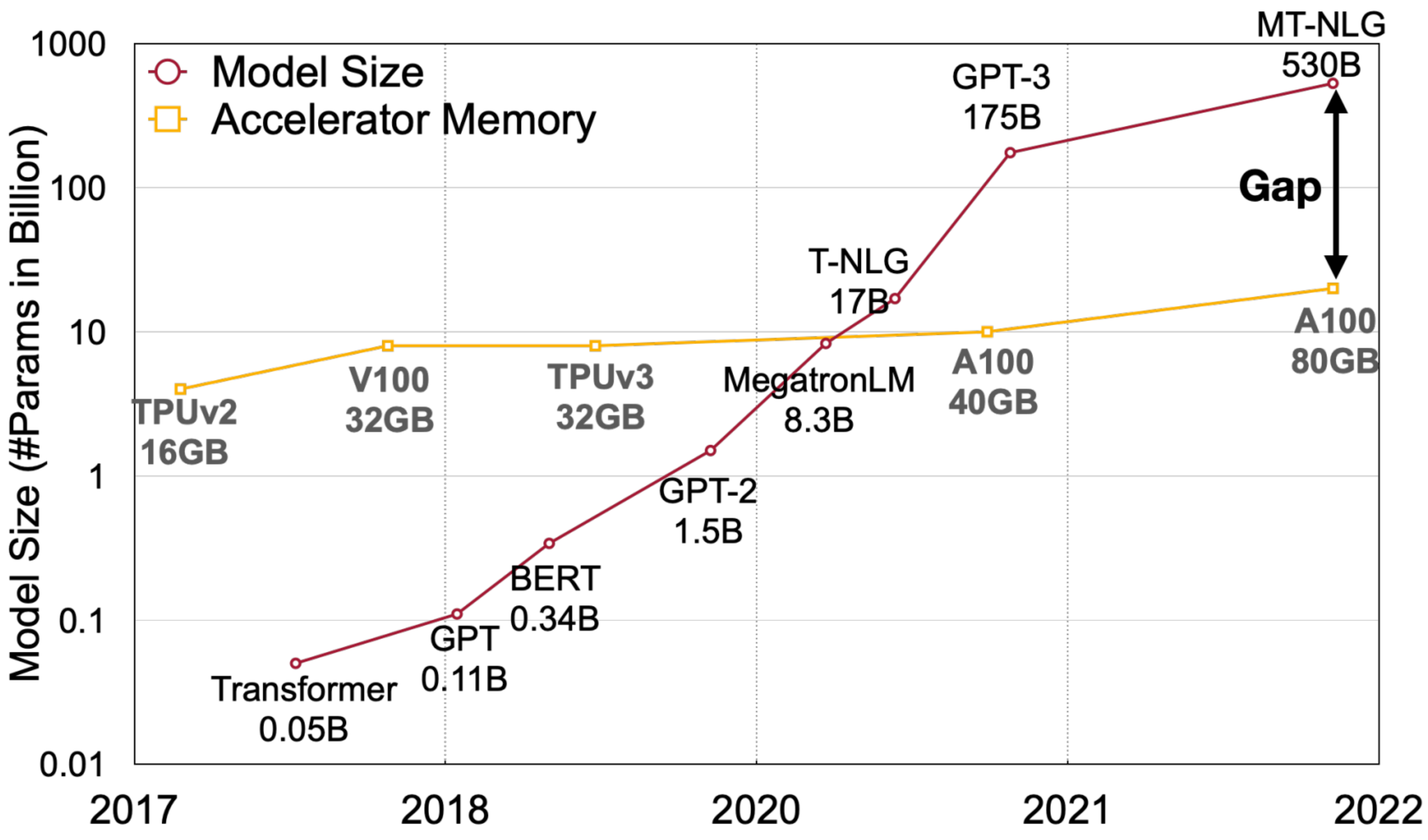
MLSys, 2024
Santa Clara, CA

SHANGHAI JIAO TONG UNIVERSITY 1896
PEKING UNIVERSITY 1898
Carnegie Mellon University

Challenges for LLM Serving

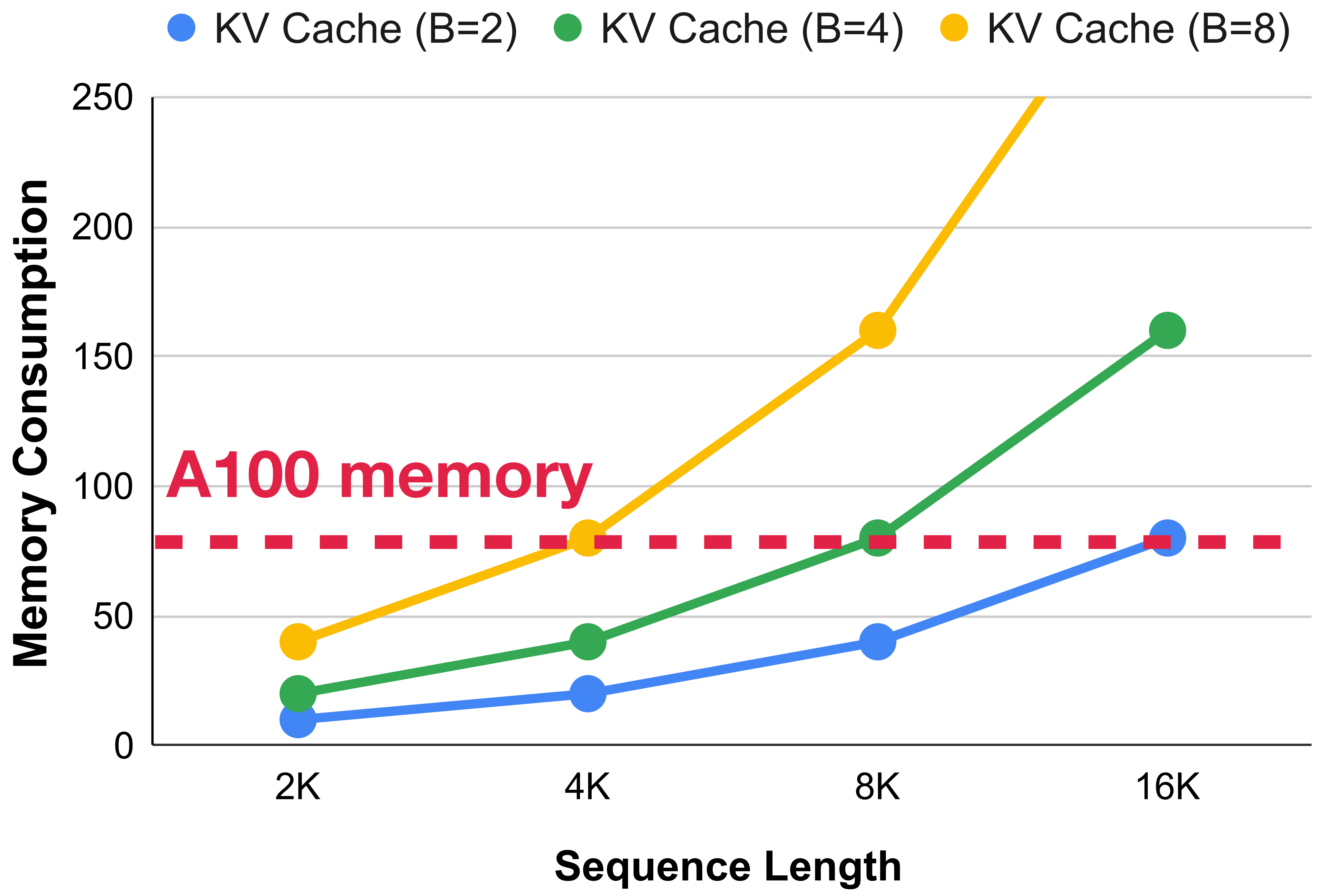
Large memory usage

Large Model weights



LLM size and accelerator memory

Large KV Cache



KV Cache size for Llama-65B

SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models, ICML 2023

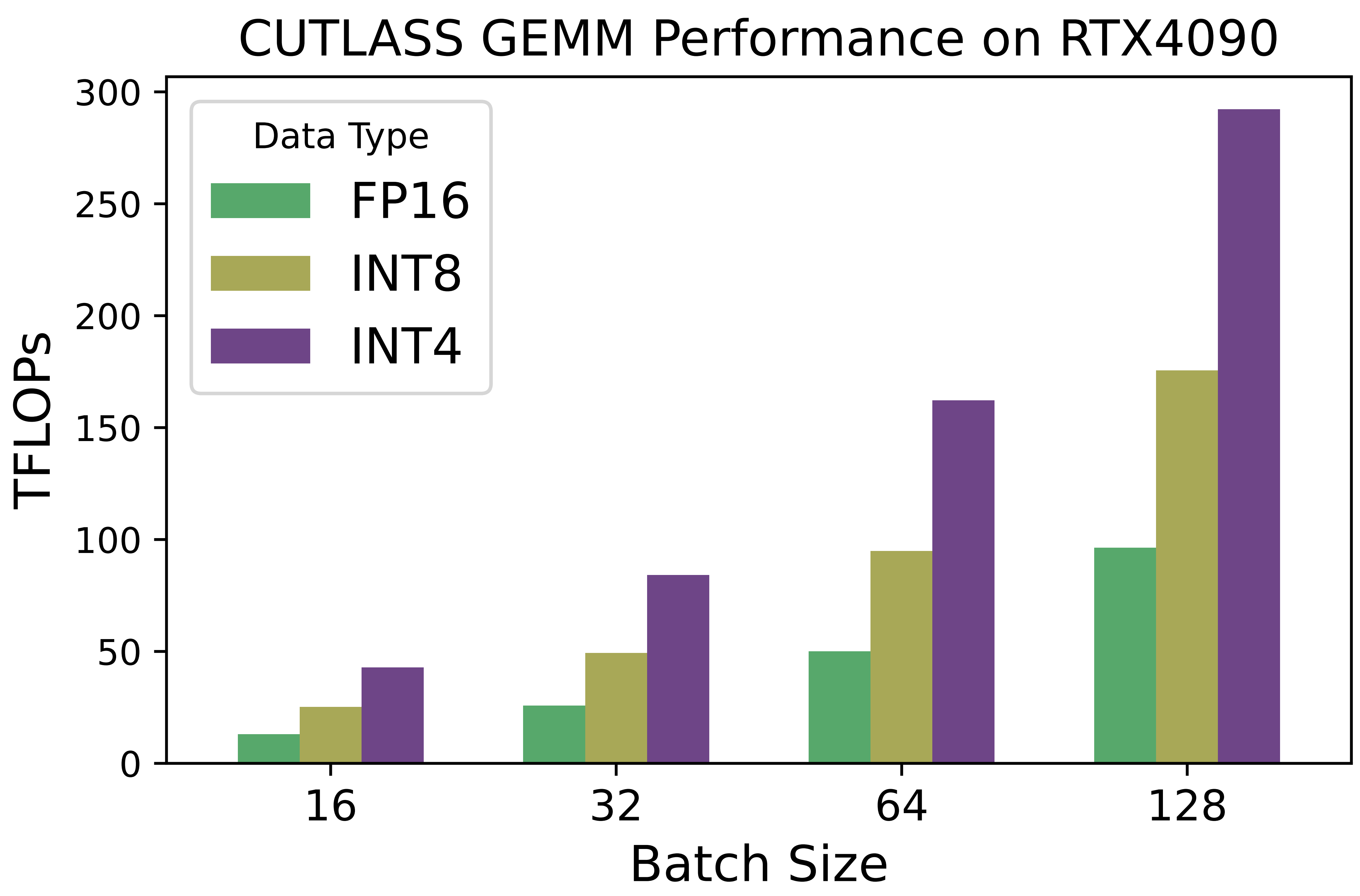
Challenges for LLM Serving

Low compute utilization

Max Batch Size for Llama-65B
(With 4xA100 80GB)

Seqlen	512	1024	2048	4096
Max Batch	160	80	40	20

GPU Performance w/ Batch

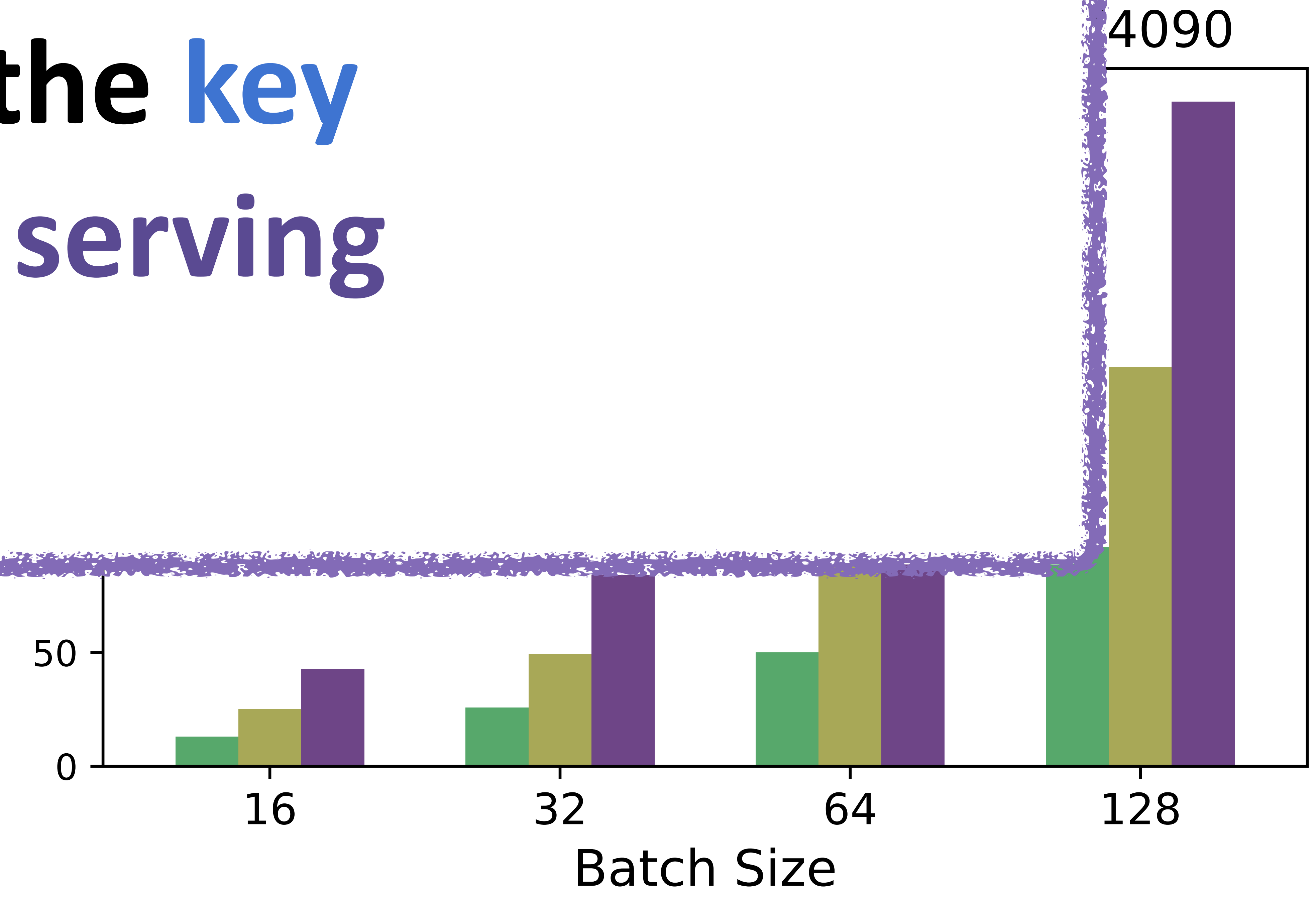


Challenges for LLM Serving

Low compute utilization

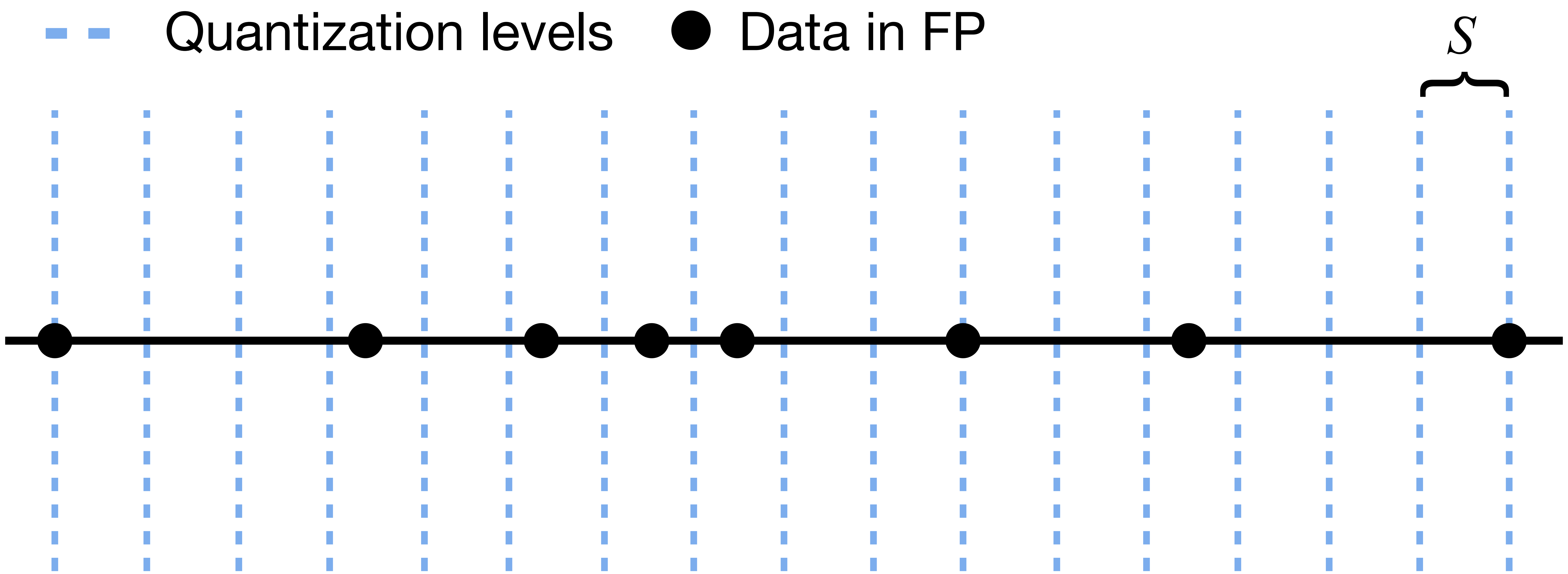
Compression is the key for efficient LLM serving

Seql				
Max Batch	160	80	40	20



Background: What is Quantization?

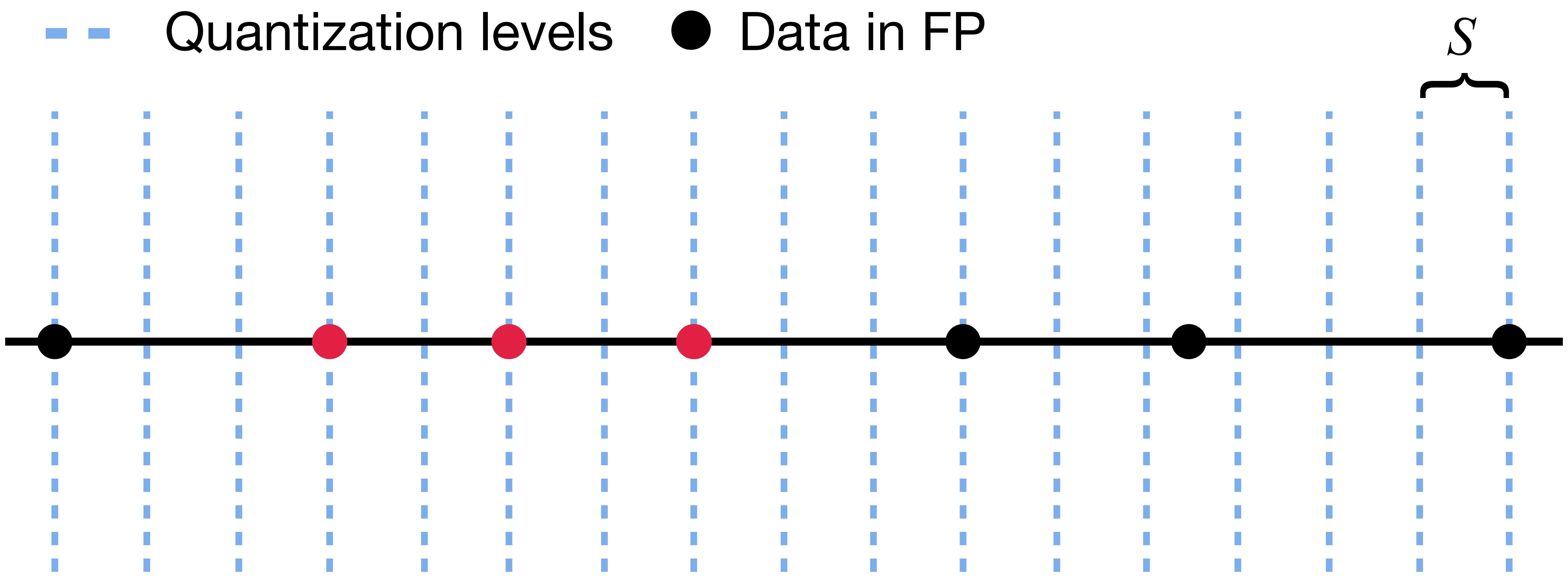
- Map data to a lower resolution
- Reduce #bits to store each element



$$\mathbf{x}_{\text{int}} = \text{clamp}\left(\left\lfloor \frac{\mathbf{x}}{s} \right\rceil; -2^{b-1}, 2^{b-1} - 1\right)$$

Background: What is Quantization?

- Map data to a lower resolution
- Reduce #bits to store each element



$$\widehat{\mathbf{x}} = s\,\mathbf{x}_{\text{int}}$$

Quantization Type

Weight-only Quantization

- Mainstream methods (AWQ, QMoE, GPTQ, SqueezeLLM, QUIP…)
- Speedup from reducing memory loading
- Dequantize weights to **high-bit** for computation

#Bit/Model	FP16	INT8	INT4
Mistral-7B	16G	8G	**4G**
Llama2-70B	140G	70G	**35G**
GPT3.5-175B	330G	165G	**83G**

LLM Sizes in different precision

Quantization Type

Weight-only Quantization

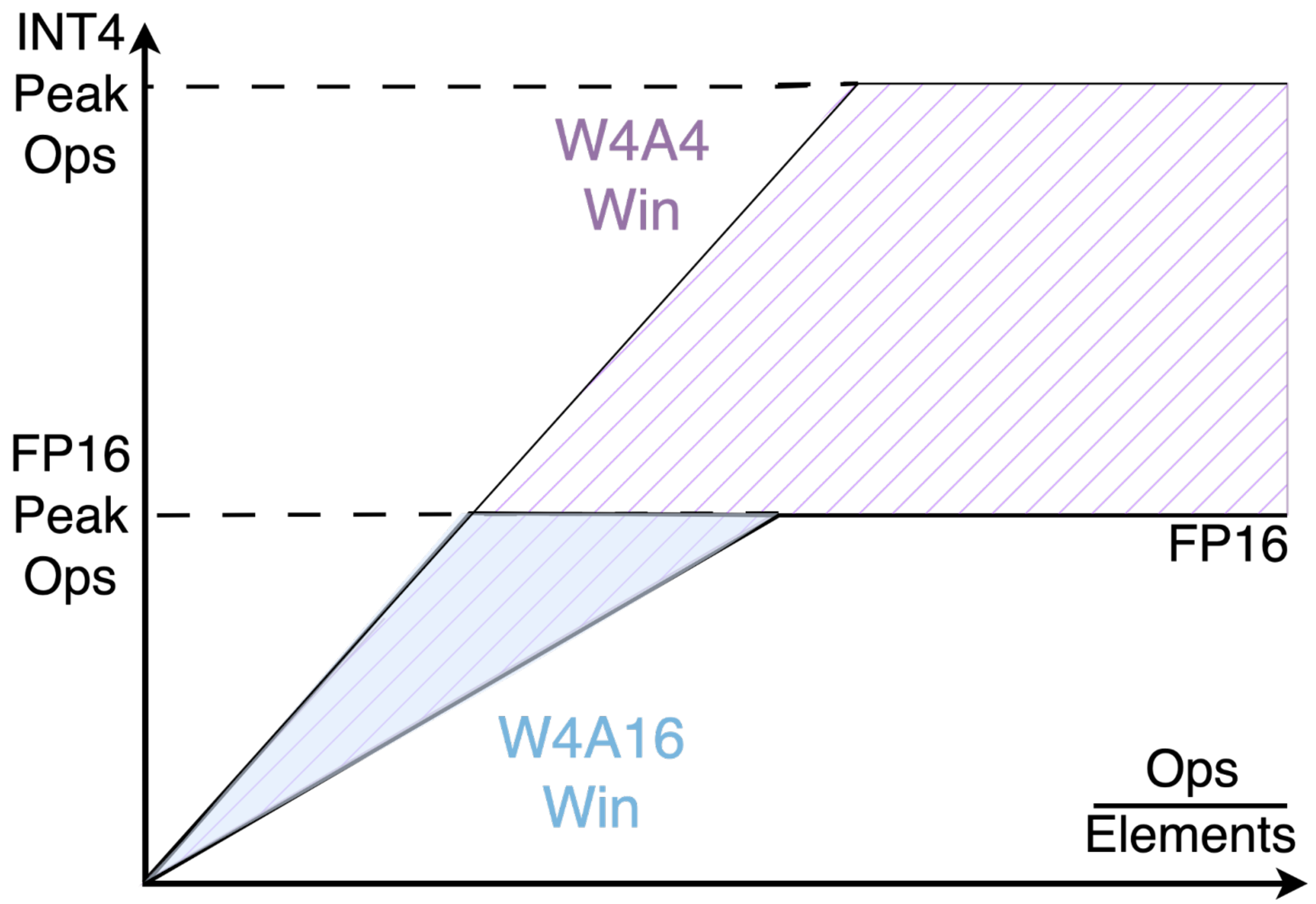
- Mainstream methods (AWQ, QMoE, GPTQ, SqueezeLLM, QUIP…)
- Speedup from reducing memory loading
- Dequantize weights to **high-bit** for computation

#Bit/Model	FP16	INT8	INT4
Mistral-7B	16G	8G	**4G**
Llama2-70B	140G	70G	**35G**
GPT3.5-175B	330G	165G	**83G**

LLM Sizes in different precision

Weight-Activation Quantization

- Use efficient **low-bit** arithmetic for computation
- Cont. increasing throughput when batch is larger
- **Prior works can not maintain accuracy at 4-bit**



Roofline model with different precision

Quantization Type

Weight-only Quantization

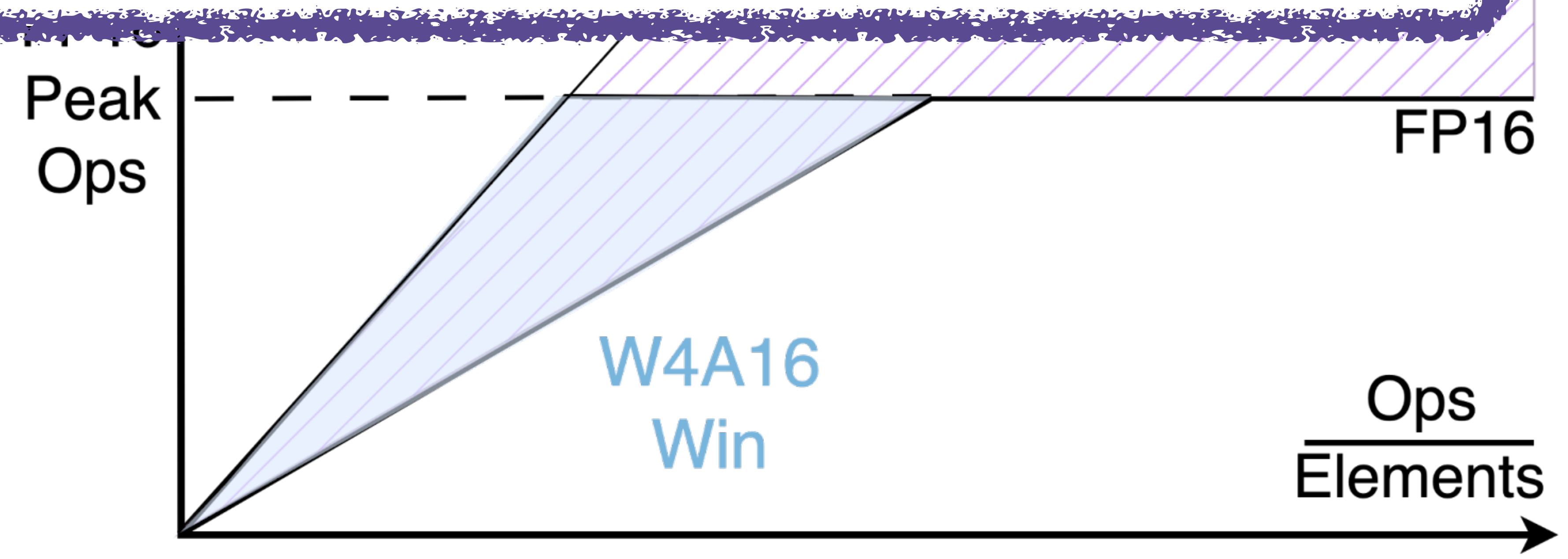
- Mainstream methods (AWQ, QMoE, GPTQ, SqueezeLLM, QUIP…)
- Speedup
- Dequant

#Bit/M			
Mistra			
Llama2-70B	140G	70G	**35G**
GPT3.5-175B	330G	165G	**83G**

LLM Sizes in different precision

Weight-Activation Quantization

- Use efficient **low-bit** arithmetic for computation
- Cont. increasing throughput when batch is larger
- **y at 4-**



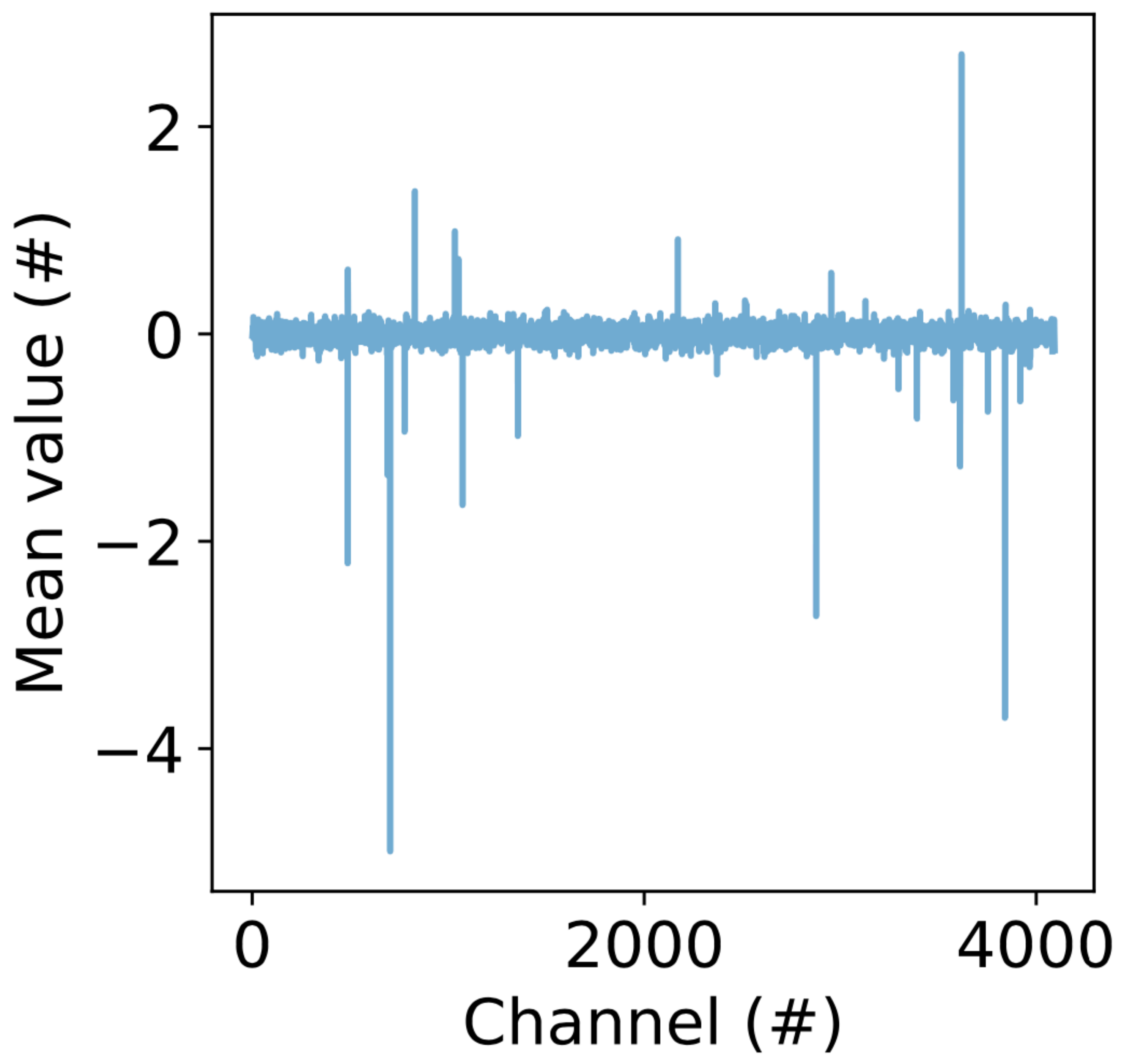
Roofline model with different precision

Atom

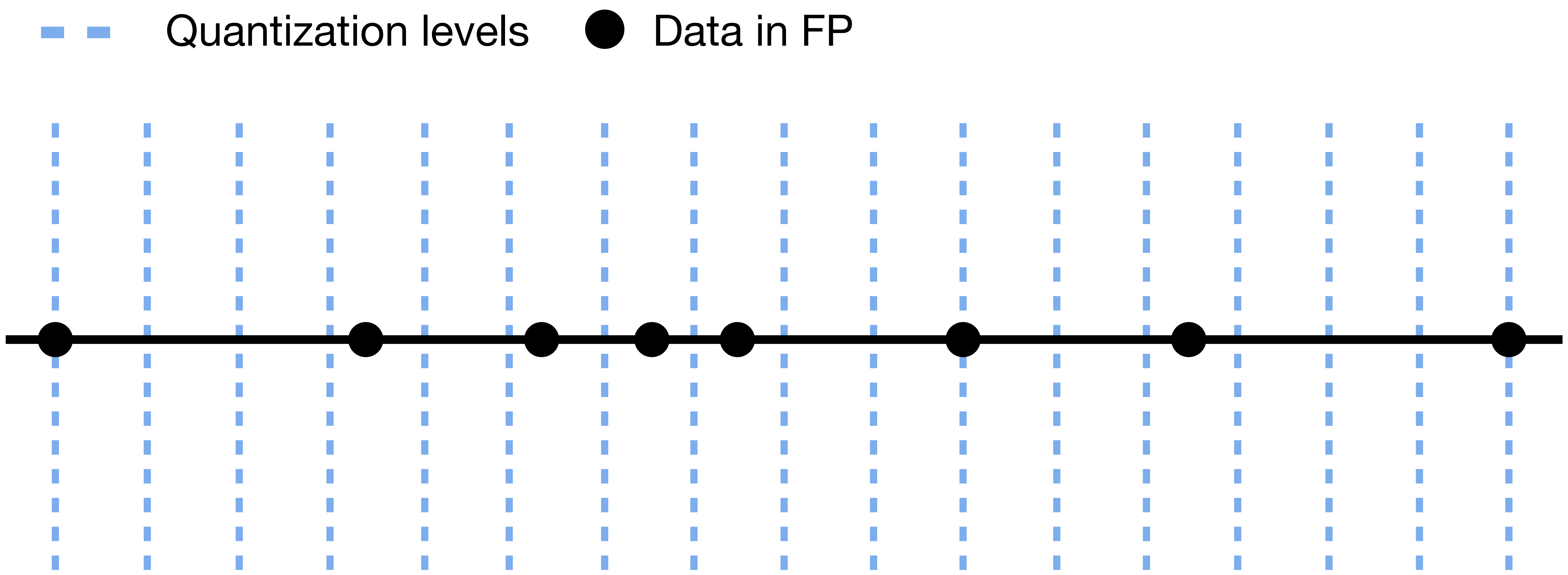
Maintaining LLM accuracy at W4A4 with a **quantization-system co-design**

LLM Quantization Challenges: Outliers

- **Few** activation channels are **consistently larger** than others
- Outliers **ruin** quantization accuracy

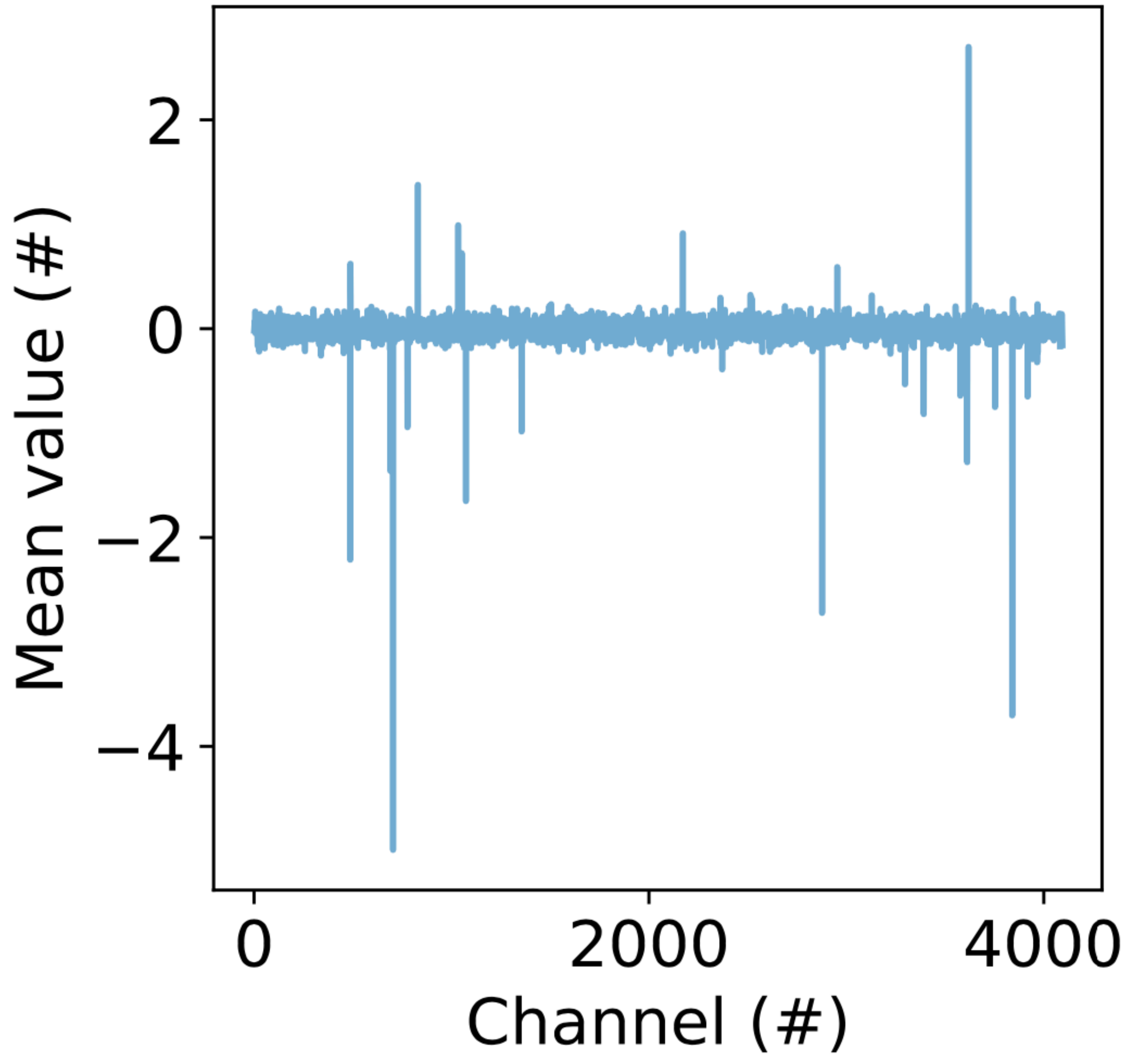


Activations sampled from Llama-7B

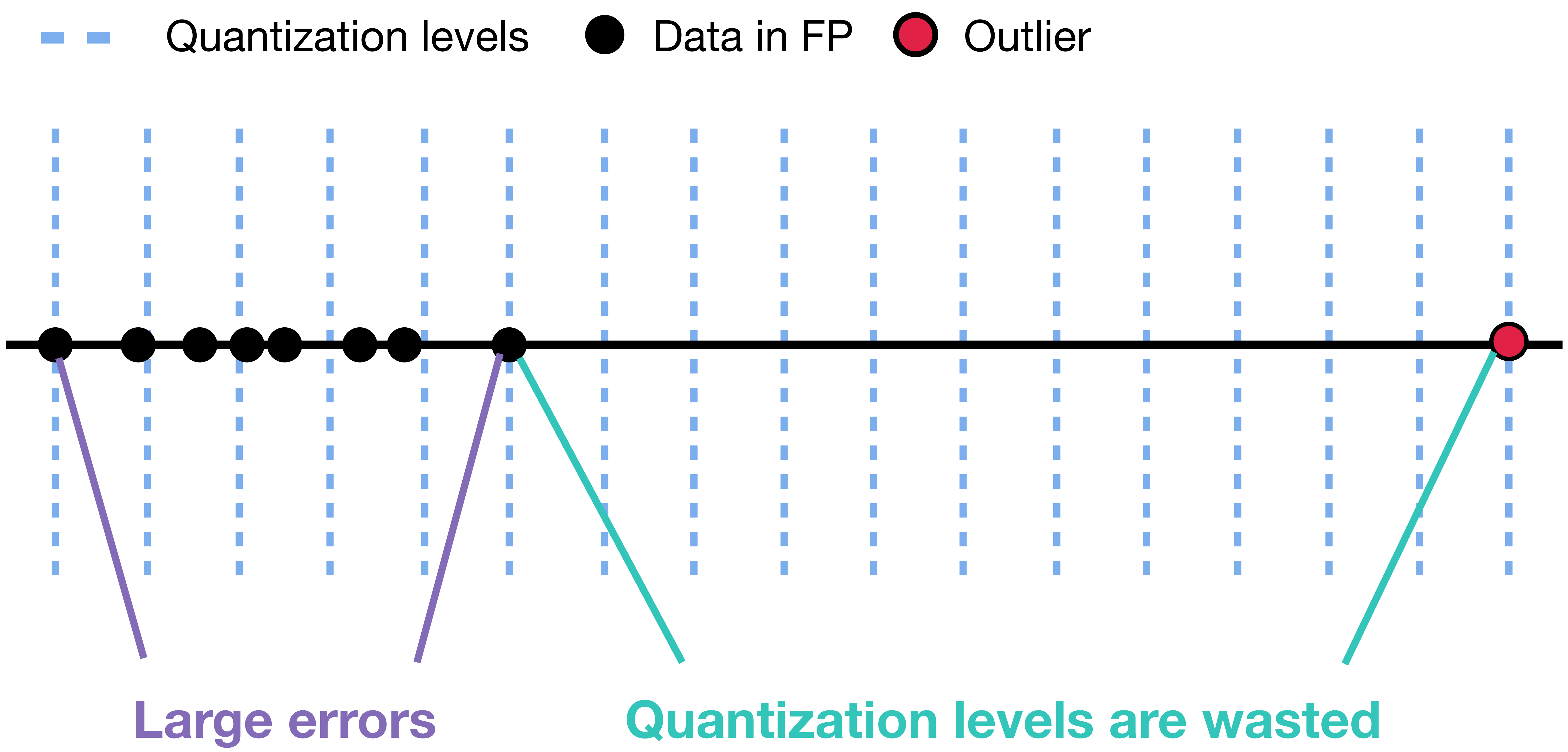


LLM Quantization Challenges: Outliers

- **Few** activation channels are **consistently larger** than others
- Outliers ruin quantization accuracy

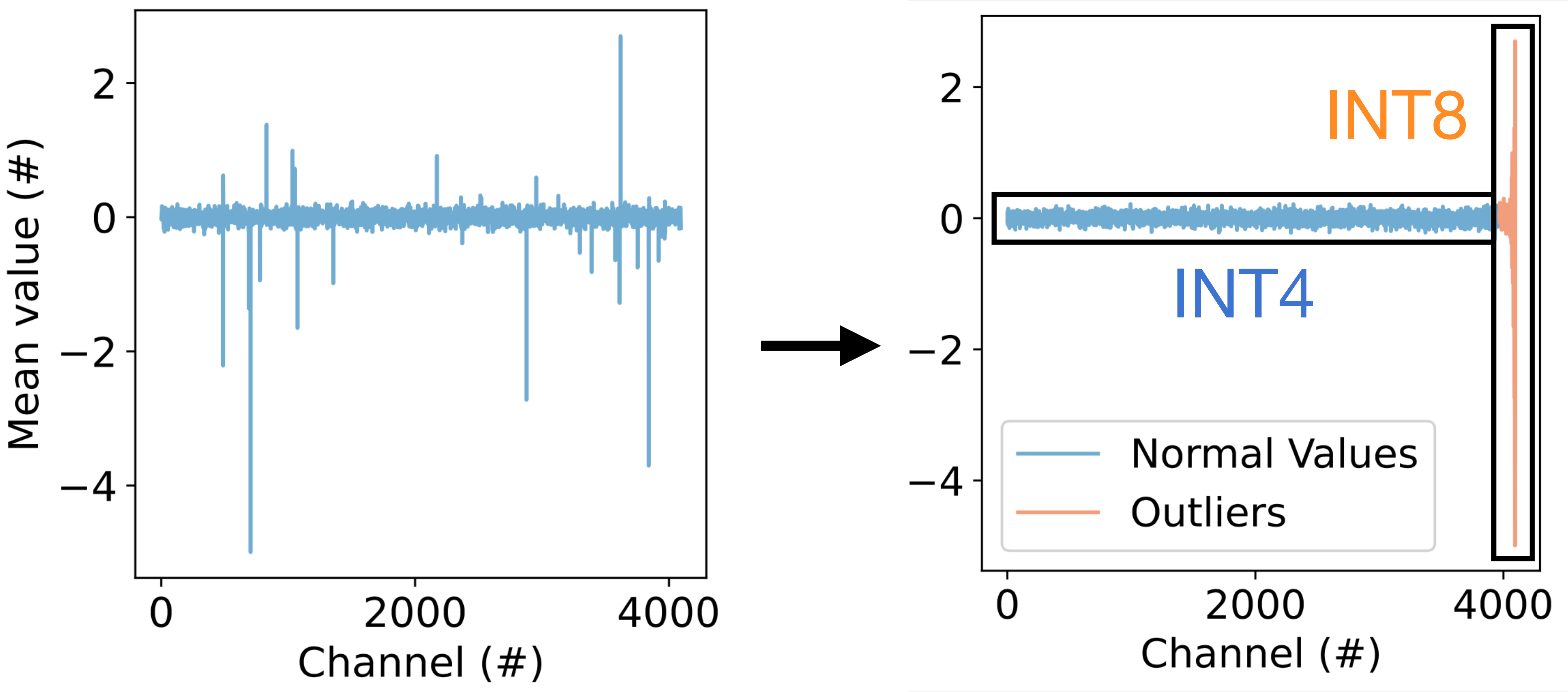


Activations from Llama-7B

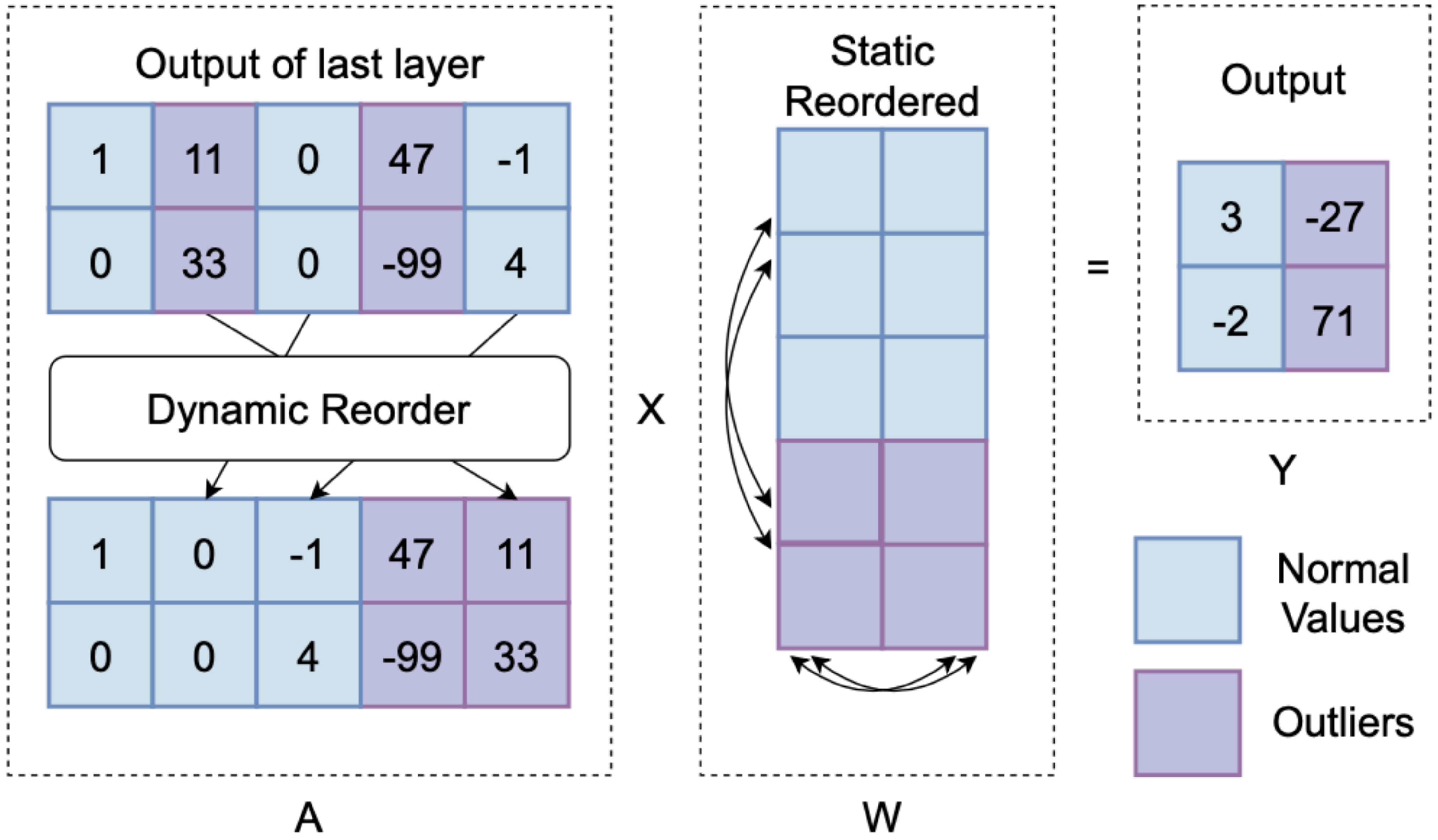


Reorder-Based Mixed Precision

- Keep outlier channels in INT8, quantize others to INT4
- **Reorder** outlier channels for regular memory accessing
- **Hide** activation reordering overhead in previous layer

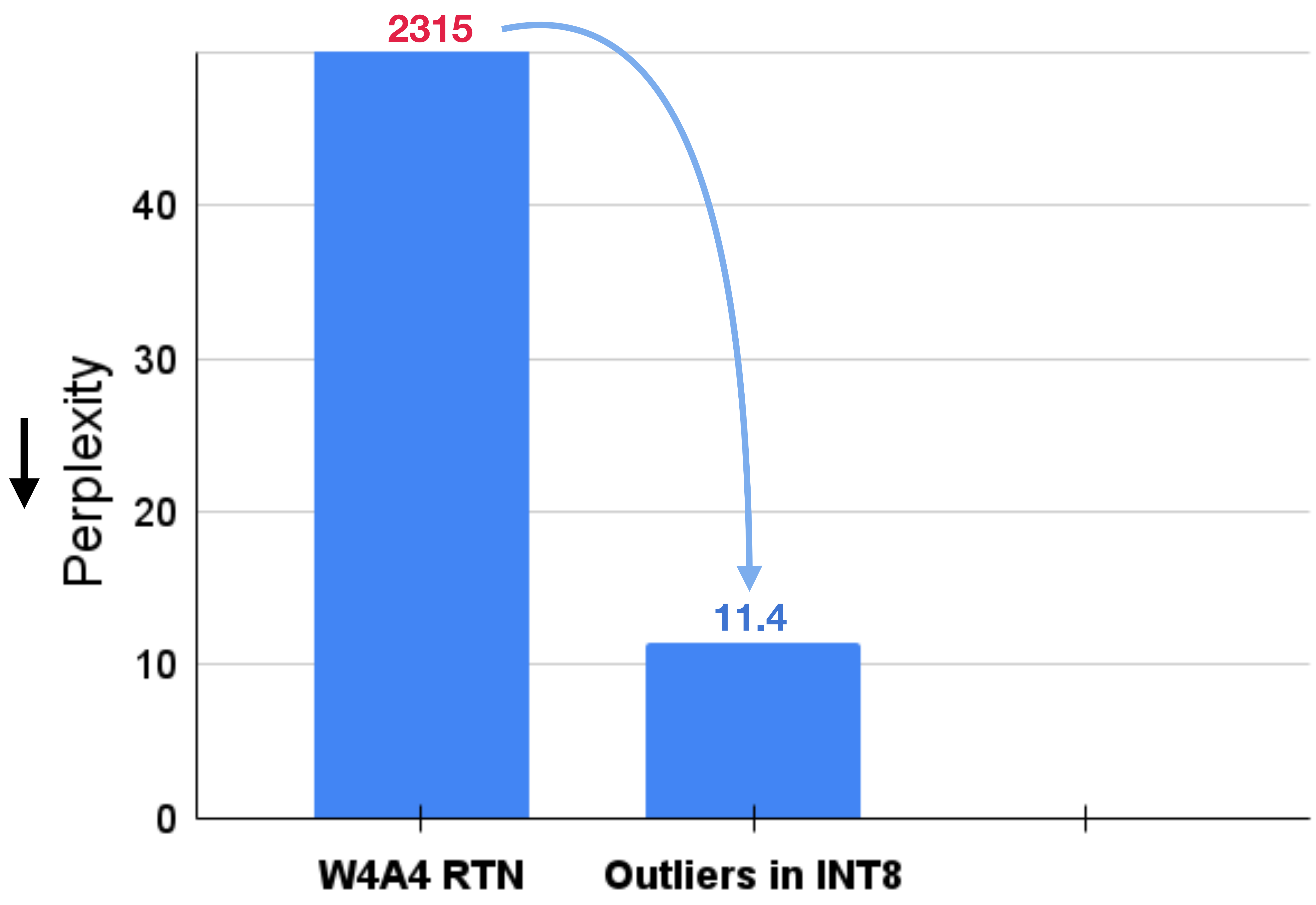


Activations after Reordering

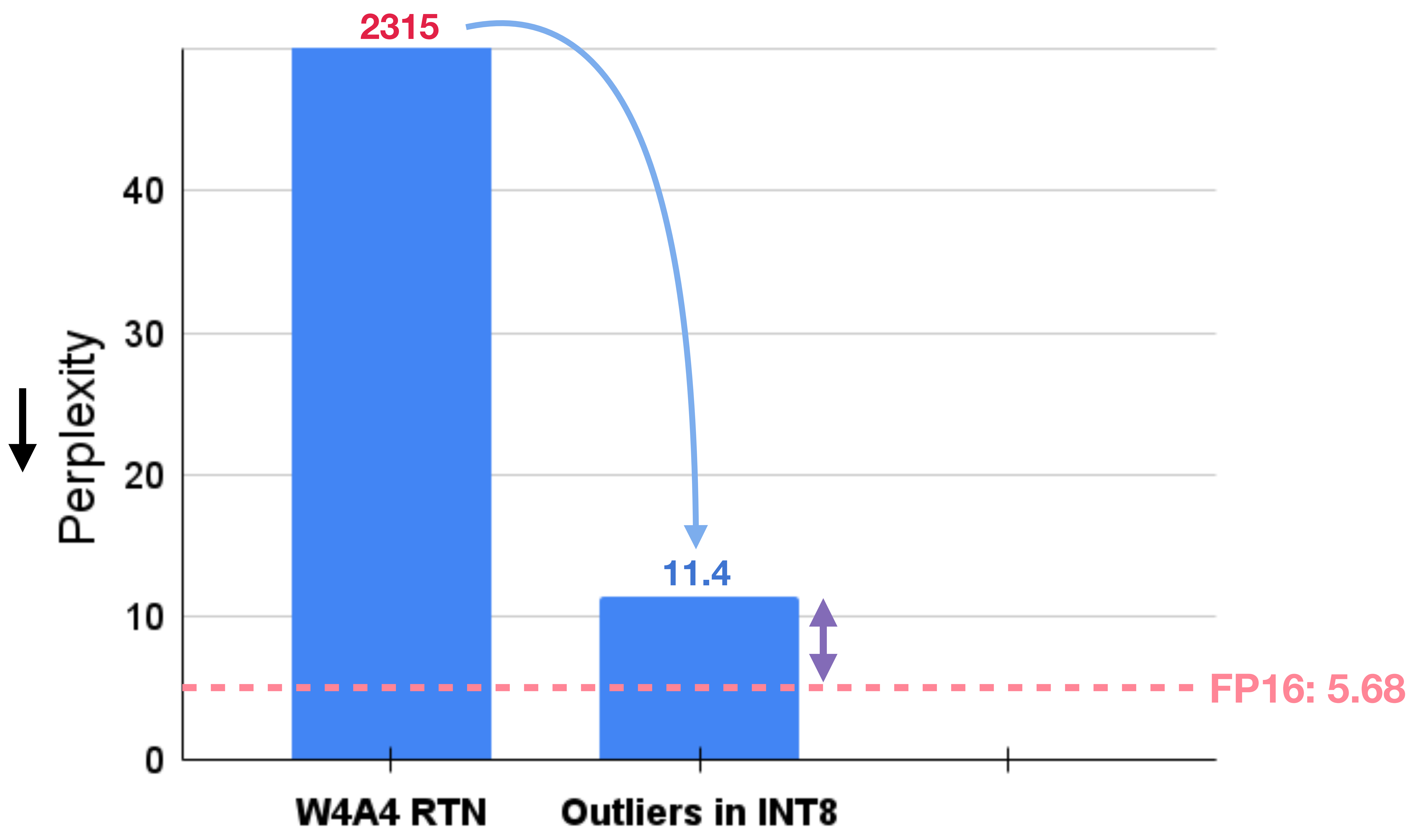


Reorder weights for accurate GEMM

Llama-7B WikiText2 Perplexity with Mixed-Precision

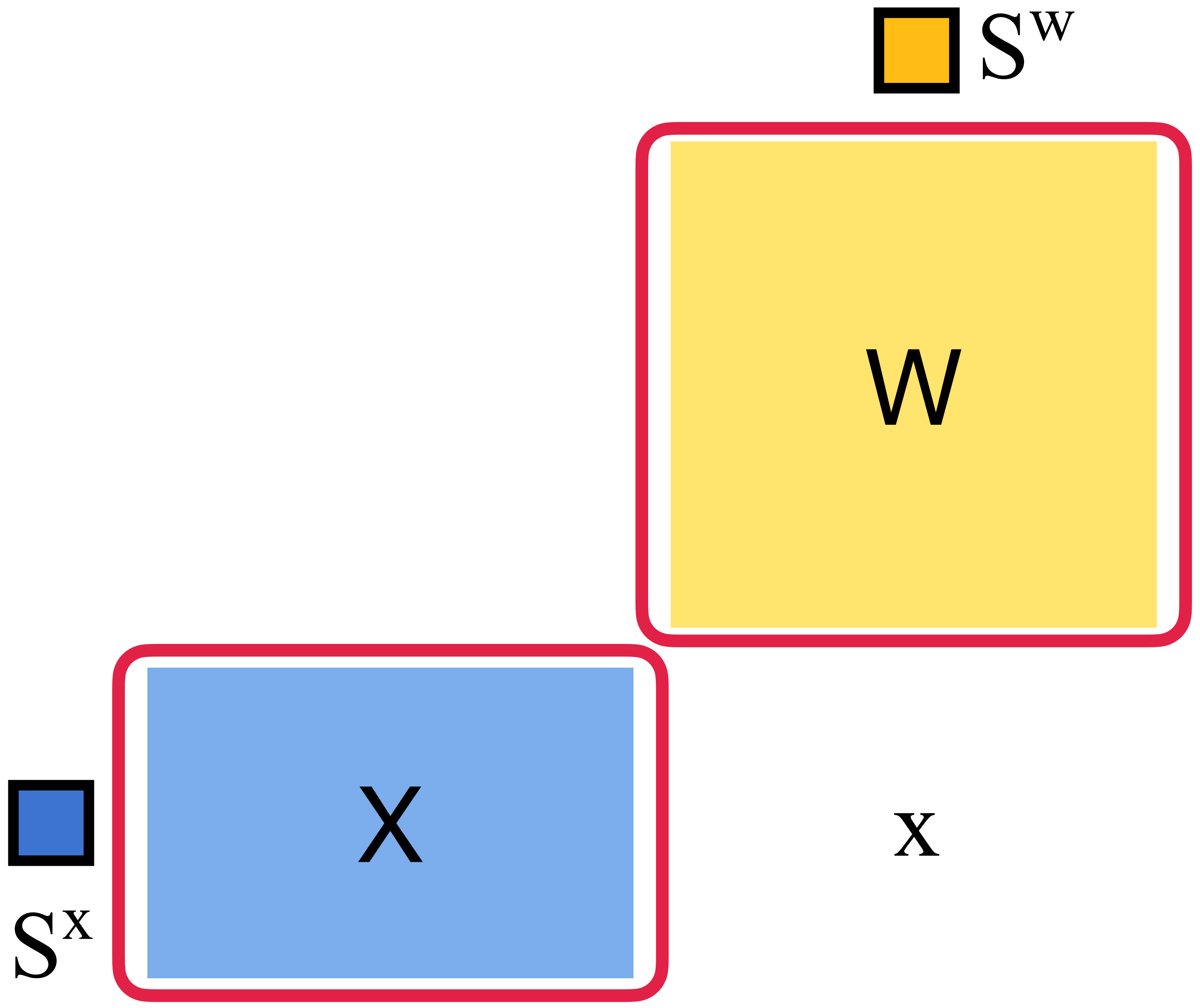


Llama-7B Perplexity with Mixed-Precision



Fine-grained Group Quantization

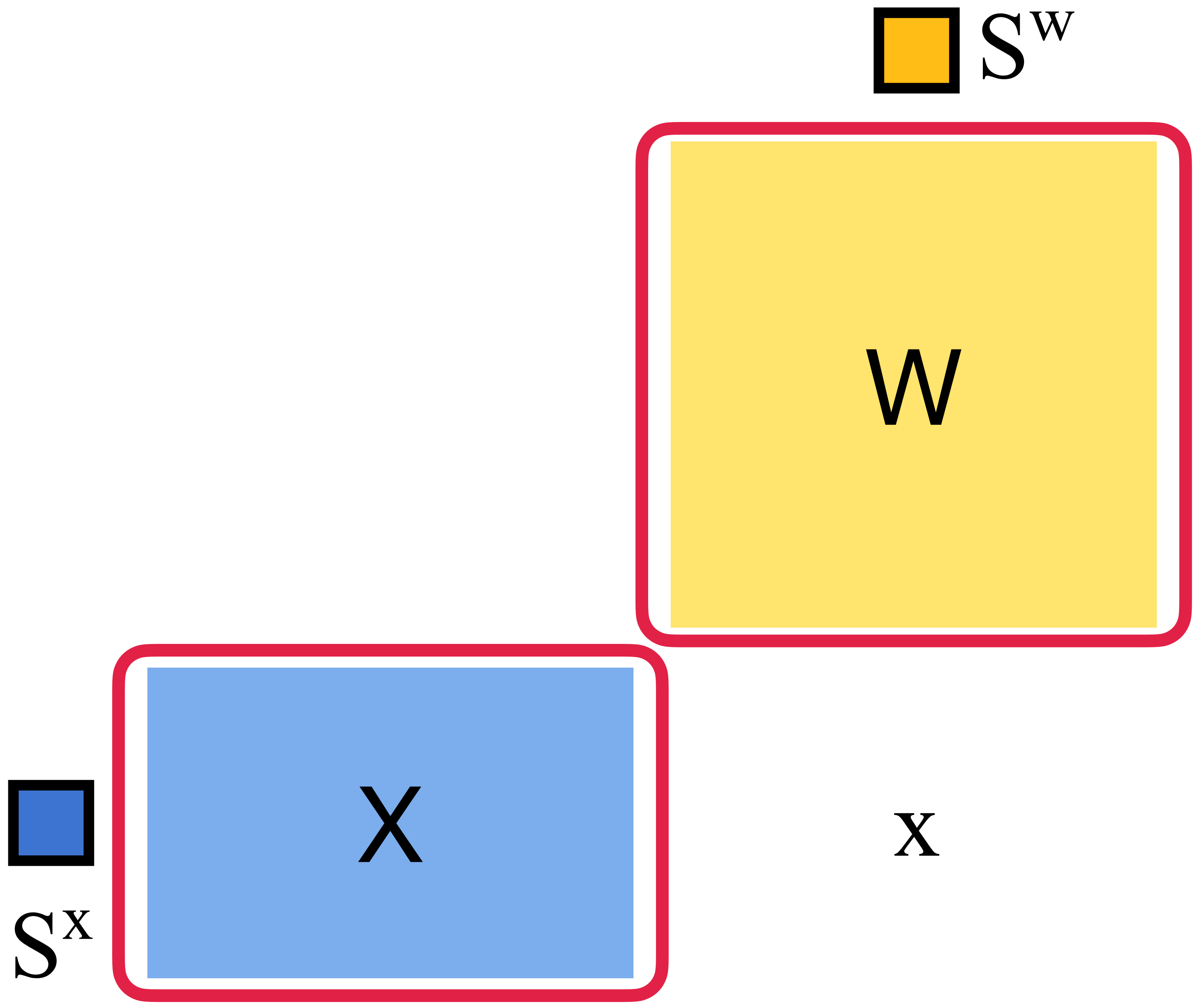
Low accuracy



Per tensor

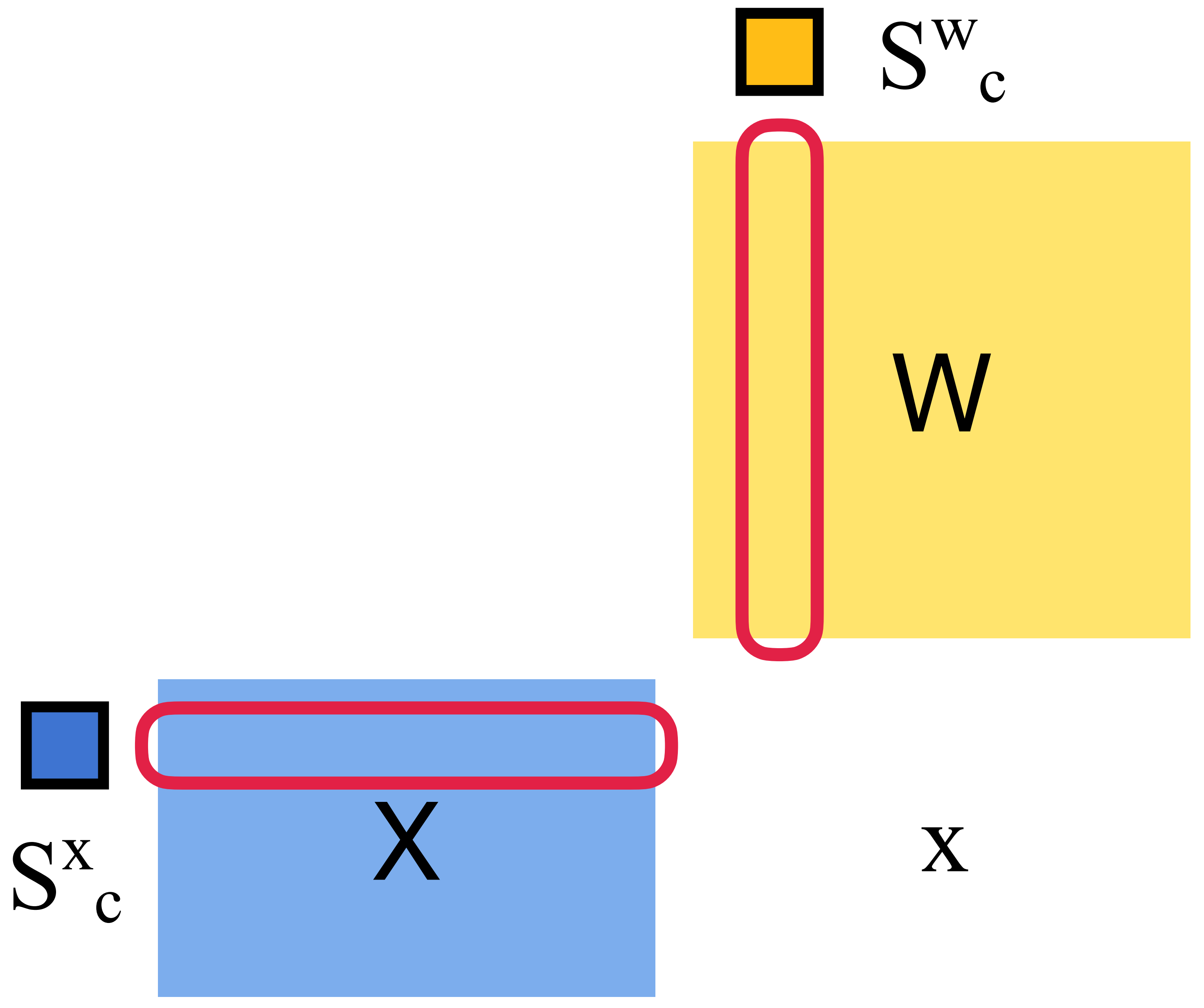
Fine-grained Group Quantization

Low accuracy



Per tensor

Medium accuracy



Per token

Fine-grained Group Quantization

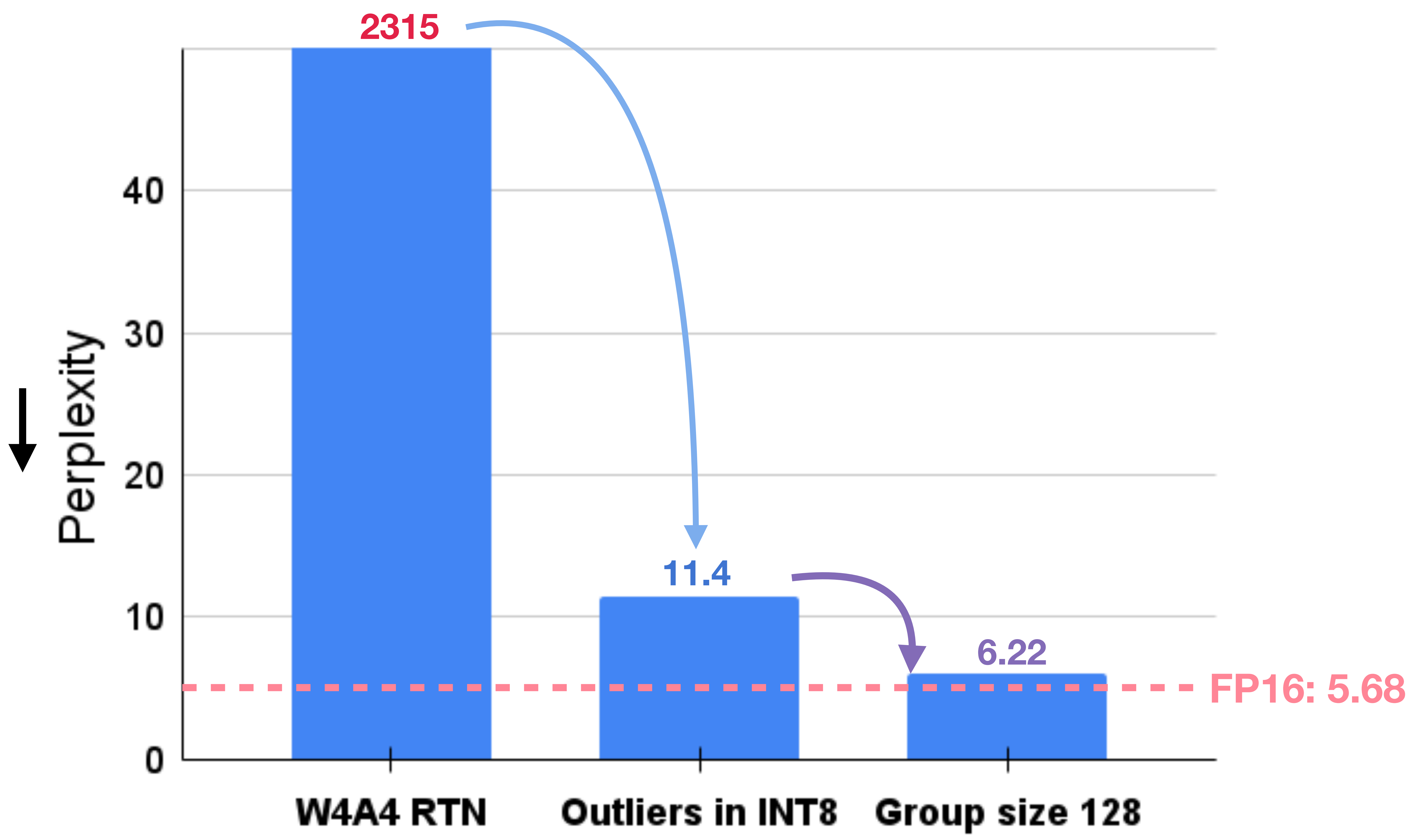
Fine-grained Group Quantization

Low accuracy | Medium accuracy | High accuracy

Prior W-Act Quants

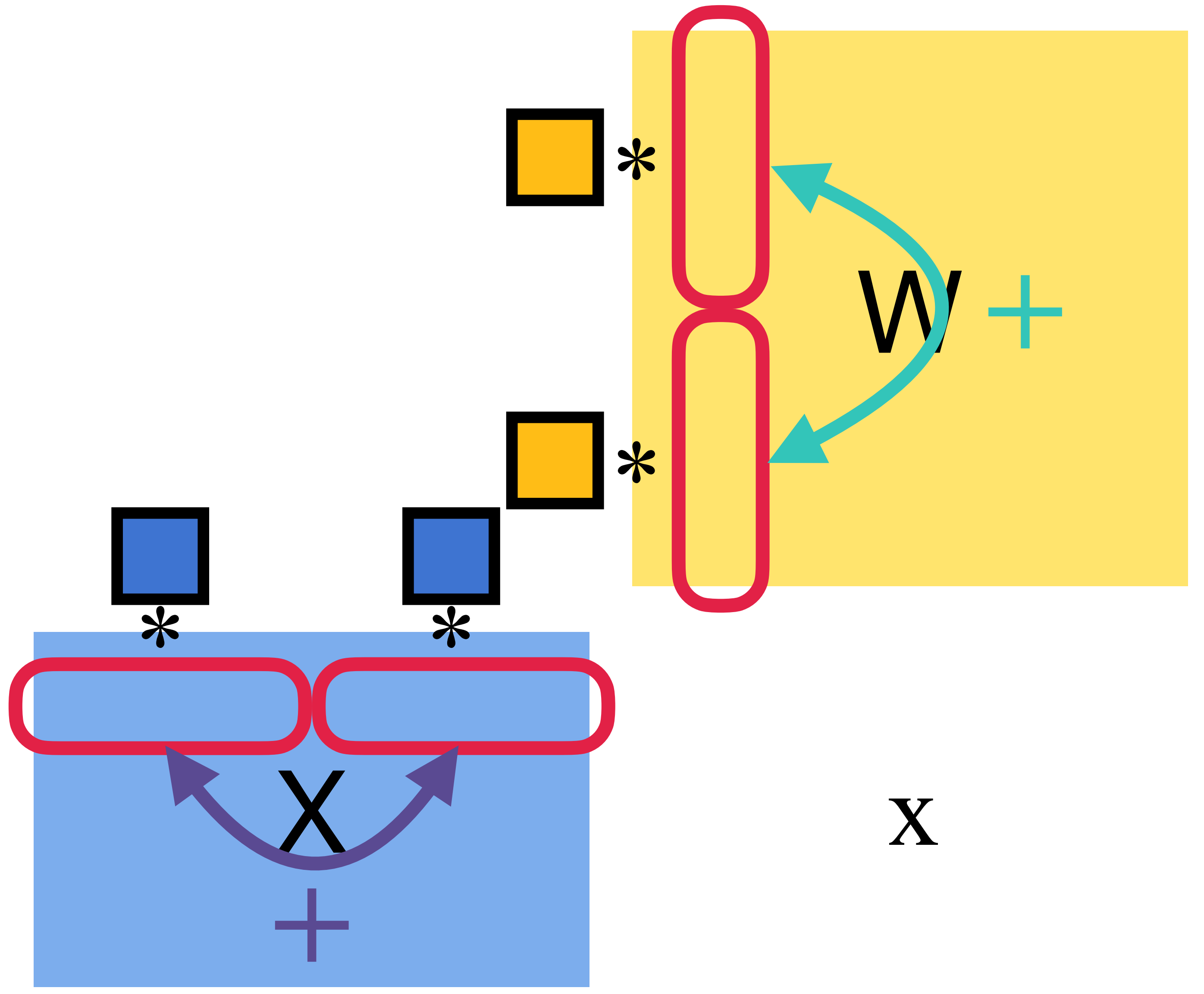
Atom

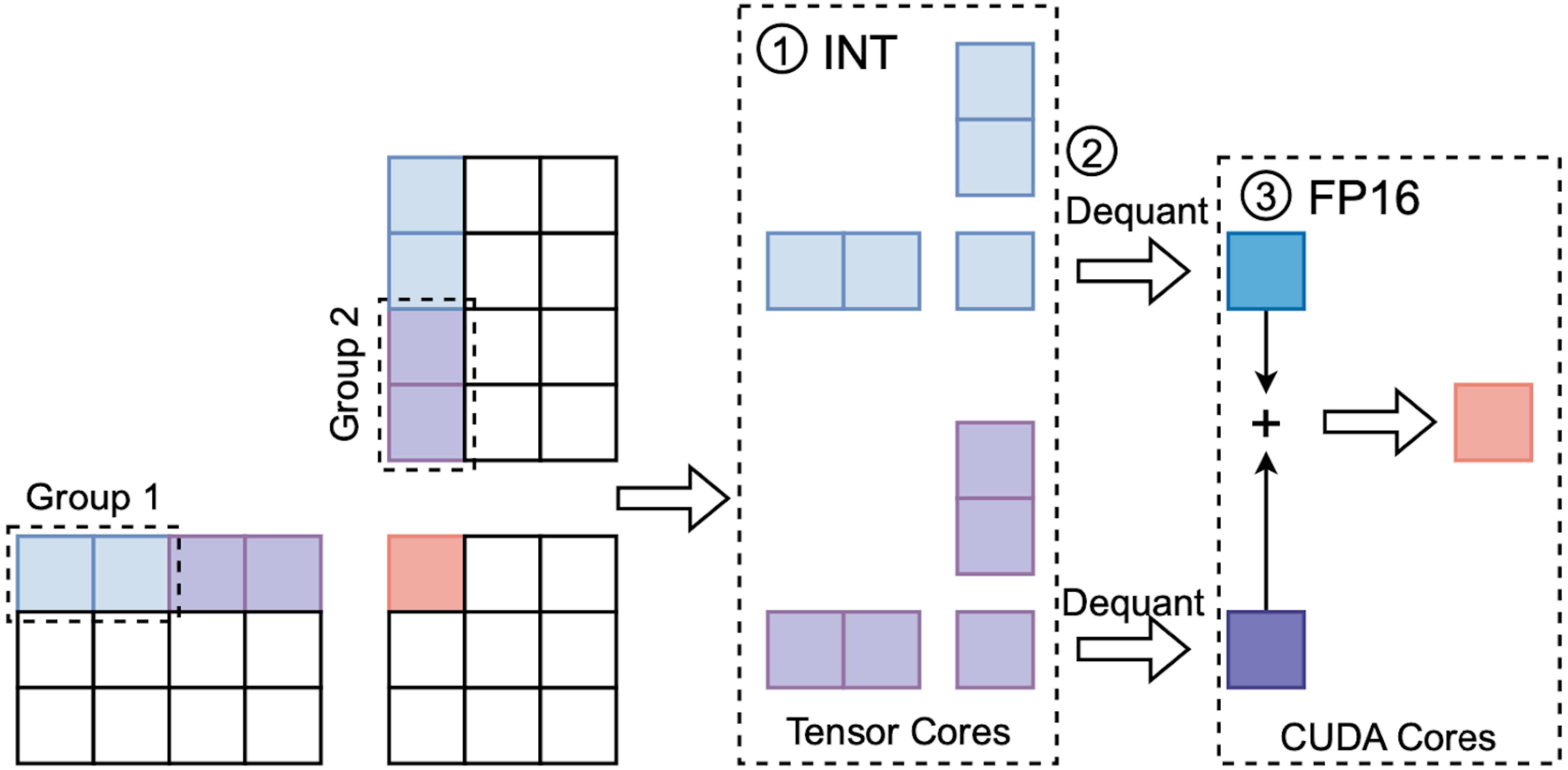
Llama-7B Perplexity with Fine-Grained Group Quant.



Overheads of Group Quantization

- Partial sum **between groups** can not be accumulated directly
- To accumulate: (1) dequantize partial sum to **FP16** and (2) sum up in FP16
- We design a **specialized GPU kernel** to handle GEMM with group quant
- We **fuse** low-bit and high-bit GEMM in one kernel



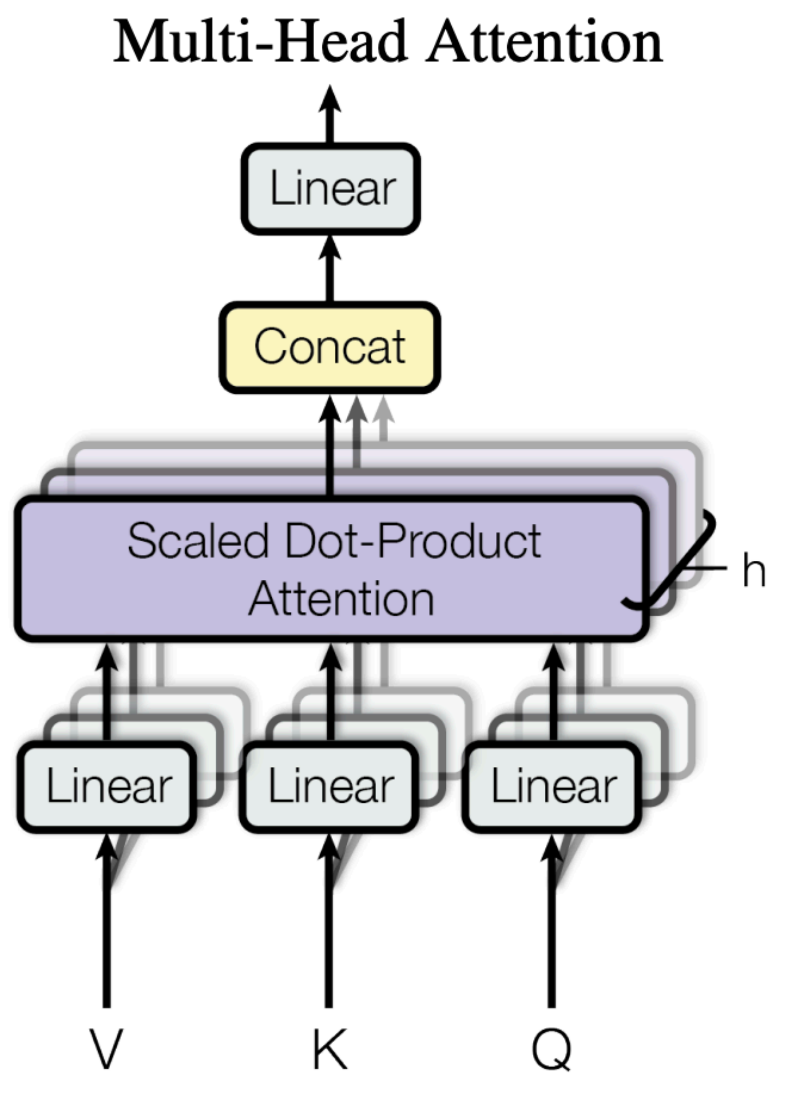


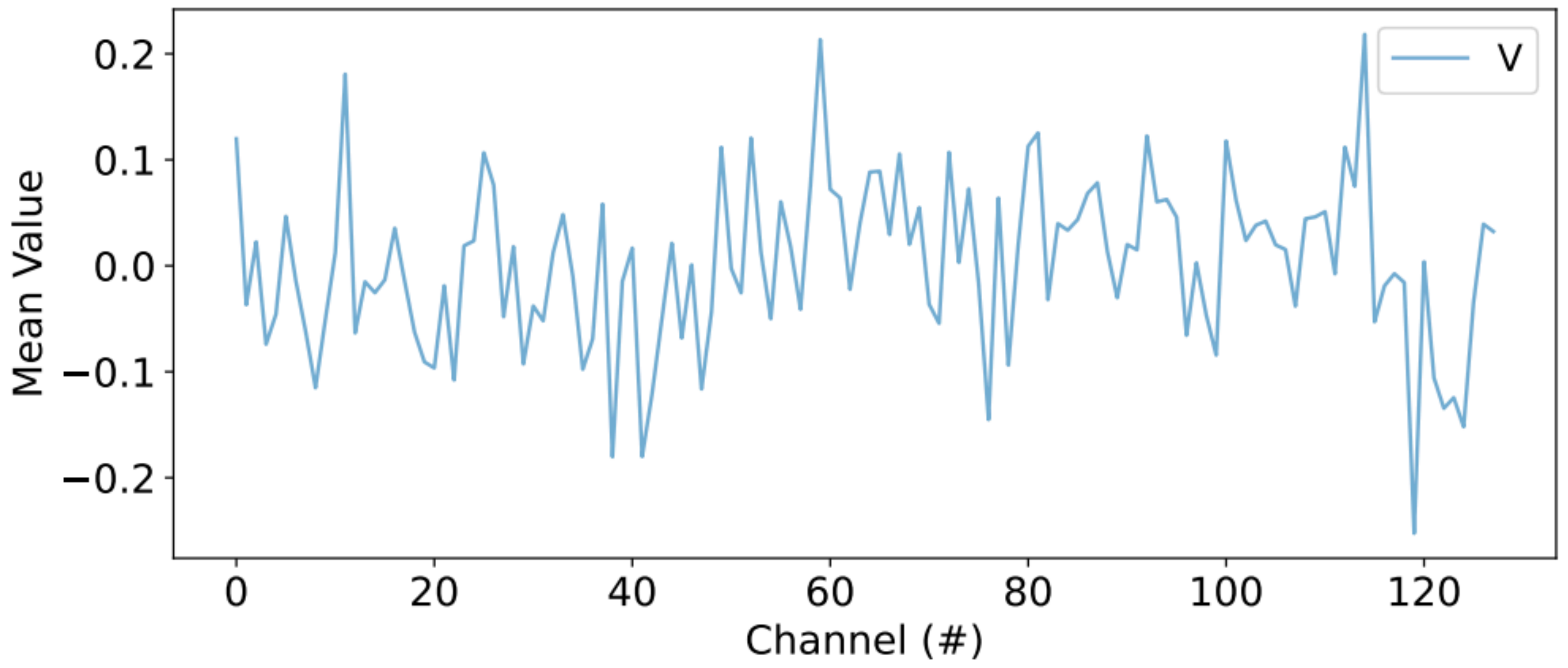
Atom GEMM kernel design

KV Cache Quantization

- KV Cache: caching key and value data for self-attention layer to save computation
- KV Cache is **relatively easy** to quant: **a simple 4-bit RTN** can maintain accuracy
- Mixed-precision, reordering, group quantization can still be applied to KV Cache

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$





V data from Llama-7B

Evaluation

Accuracy Evaluation Setup

- LLMs: Llama, Llama2, Mixtral-8x7B
- Baselines: SmoothQuant[1], OmniQuant[2], QLLM[3]
- Group size: 128
- Outliers: 128
- Calibration: 128 samples from WikiText2
- Perplexity eval: WikiText2, PTB, C4
- Zero-shot accuracy eval: six common sense tasks from **lm-evaluation-harness**[4]

[1] SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models, ICML 2023
[2] OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models, ICLR 2024
[3] QLLM: Accurate and Efficient Low-Bitwidth Quantization for Large Language Models, ICLR 2024
[4] https://github.com/EleutherAI/lm-evaluation-harness

Zero-Shot Accuracy of LLaMA-65B

- At W4A4, Atom is able to maintain accuracy with only a **1.47%** drop
- Atom's accuracy at **W3A3** is even **better** than **prior works at W4A4**

Llama	#Bits	Method	Zero-shot Accuracy ↑ PIQA	ARC-e	ARC-c	BoolQ	HellaSwag	Winogrande	Avg.	
65B	FP16	-	80.79	58.71	46.24	82.29	80.72	77.50	71.04	Baseline
	W4A4	SmoothQuant	60.72	38.80	30.29	57.61	36.81	53.43	46.28	-24.76%
		OmniQuant	71.81	48.02	35.92	73.27	66.81	59.51	59.22	-11.82%
		QLLM	73.56	52.06	39.68	-	70.94	62.90	59.83	-11.21%
		Atom	**80.41**	**58.12**	**45.22**	**82.02**	**79.10**	**72.53**	**69.57**	-1.47%
	W3A3	SmoothQuant	49.56	26.64	29.10	42.97	26.05	51.14	37.58	
		Atom	75.84	51.43	41.30	74.07	72.22	64.33	63.20	

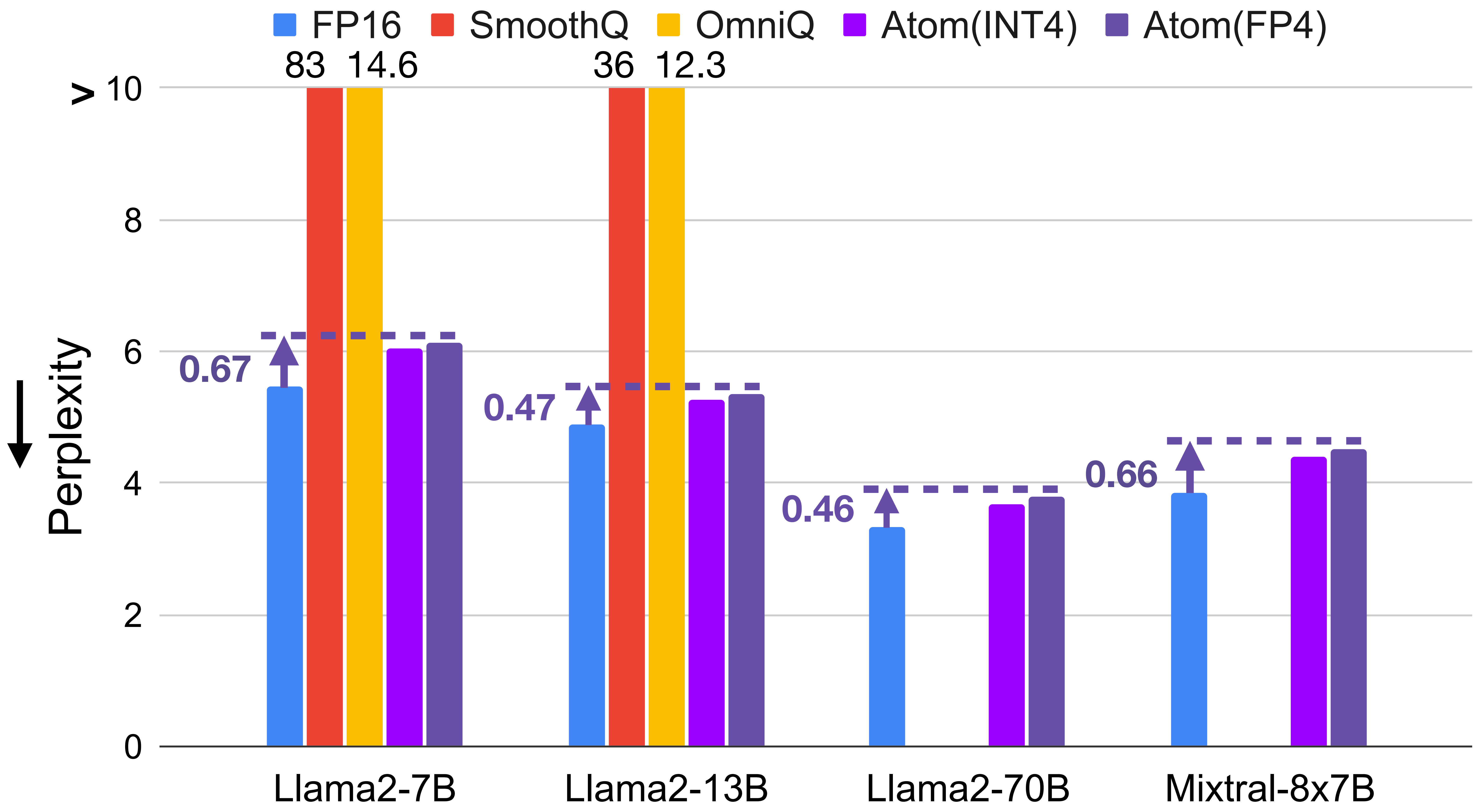
Zero-Shot Accuracy of LLaMA-65B

- At W4A4, Atom is able to maintain accuracy with only a **1.47%** drop
- Atom's accuracy at **W3A3** is even **better** than **prior works at W4A4**

Llama	#Bits	Method	Zero-shot Accuracy ↑							
			PIQA	ARC-e	ARC-c	BoolQ	HellaSwag	Winogrande	Avg.	
65B	FP16	-	80.79	58.71	46.24	82.29	80.72	77.50	71.04	Baseline
	W4A4	SmoothQuant	60.72	38.80	30.29	57.61	36.81	53.43	46.28	-24.76%
		OmniQuant	71.81	48.02	35.92	73.27	66.81	59.51	59.22	-11.82%
		QLLM	73.56	52.06	39.68	-	70.94	62.90	59.83	-11.21%
		Atom	**80.41**	**58.12**	**45.22**	**82.02**	**79.10**	**72.53**	**69.57**	-1.47%
	W3A3	SmoothQuant	49.56	26.64	29.10	42.97	26.05	51.14	37.58	
		Atom	75.84	51.43	41.30	74.07	72.22	64.33	63.20	-7.84%

Perplexity of Llama2 & Mixtral on WikiText2

- Atom is able to main accuracy across models (Llama2, Mixtral)
- Atom can be used with **FP4** quantization



Efficiency Evaluation Setup

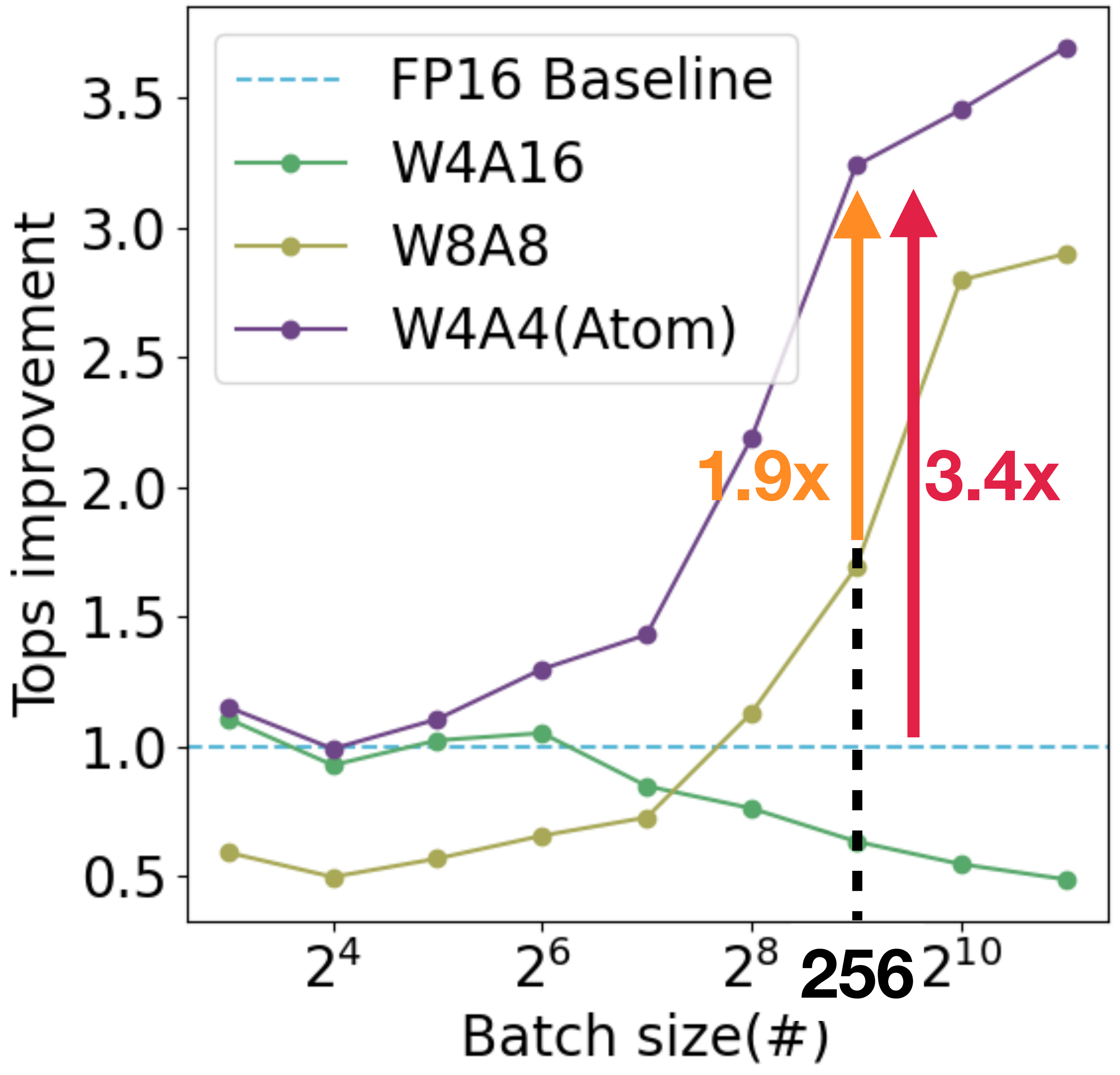
- Kernel: W4A4-G128_W8A8-O128
- Benchmark: Llama-7B
- Baseline: FP16, W4A16 (AWQ[1]), W8A8 (SmoothQuant[2])
- Workload: ShareGPT[3]
- Evaluate on RTX 4090 24GB
- Integrate into Punica[4] for end-to-end performance evaluation
- Use FlashInfer[5] as self-attention kernel and add 4-bit kernel support

[1] AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration, MLSys 2024
[2] SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models, ICML 2023
[3] ShareGPT, https://sharegpt.com/
[4] Punica: Multi-Tenant LoRA Serving, MLSys 2024
[5] FlashInfer, https://github.com/flashinfer-ai/flashinfer

GEMM Throughput & Self-Attention Latency

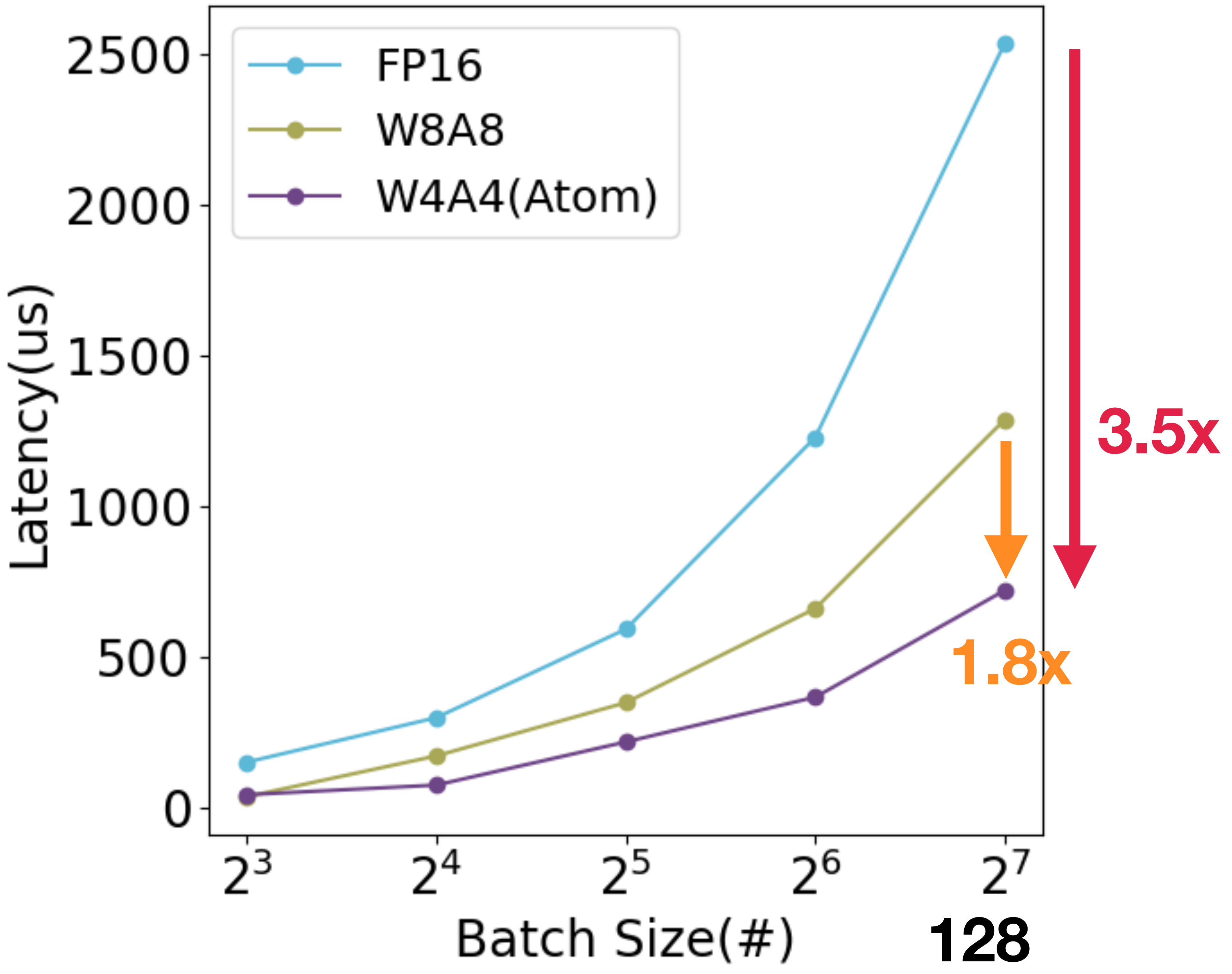
- For GEMM when B=256, Atom is **3.4x** and **1.9x** better than FP16 and W8A8
- For Self-attn when B=128, Atom is **3.5x** and **1.8x** faster than FP16 and W8A8

GEMM



Shape: Bsz x 4096 x 4096

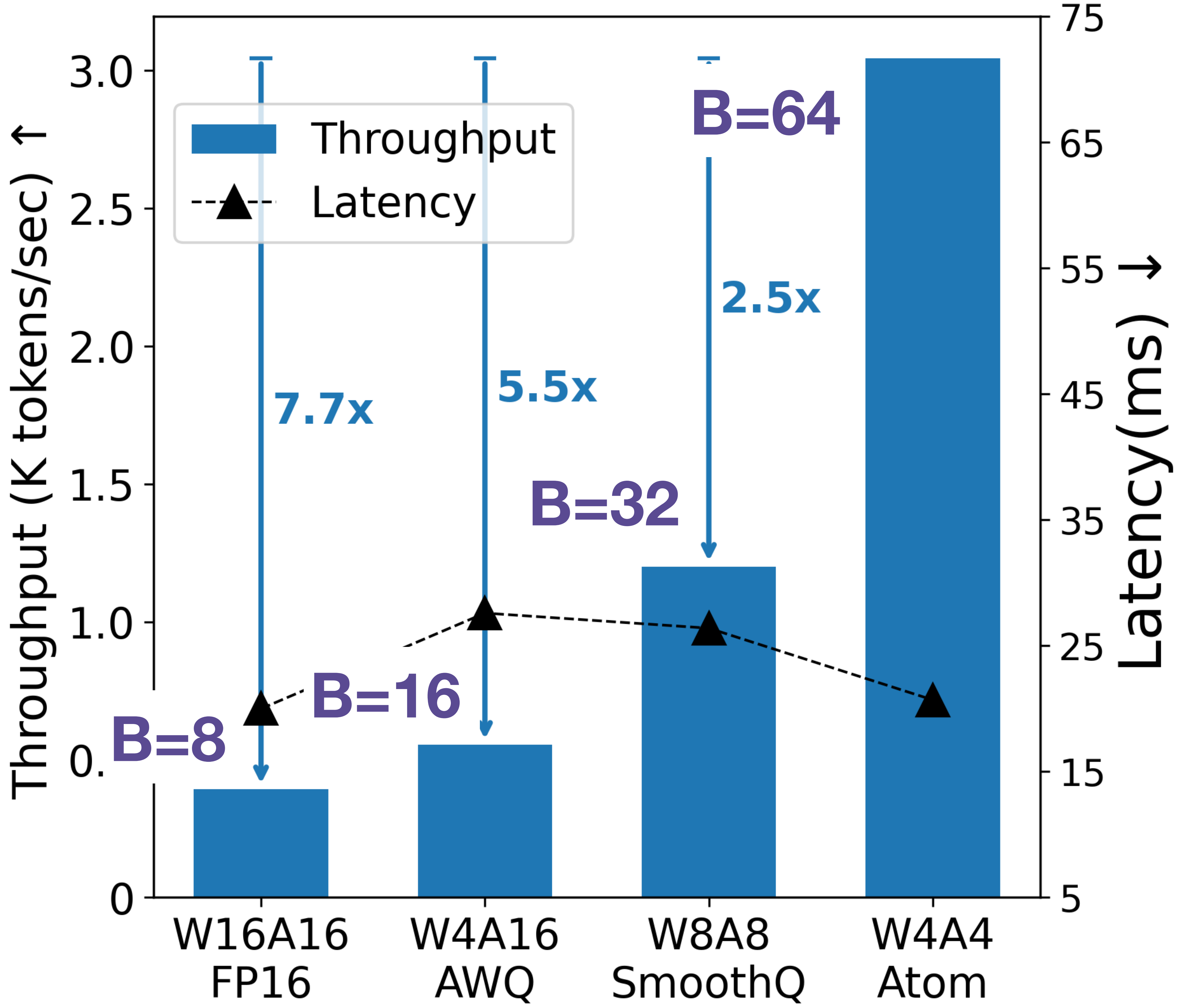
Self-attention



Sequence length: 1024

End-to-End Throughput & Latency

- Atom can boost throughput for up to 7.7x while maintaining a low latency
- Why gains are more than **4x for FP16** and **2x for W8A8**?
  Ans: Atom is able to run at **a larger batch size**



Conclusions

- Atom is an accurate and efficient low-bit weight-activation quantization for LLMs
- Atom uses (1) reorder-based mixed-precision, (2) fine-grained group quantization and (3) specialized GPU kernel
- Atom can boost end-to-end throughput for up to **7.7x** while maintaining accuracy at W4A4

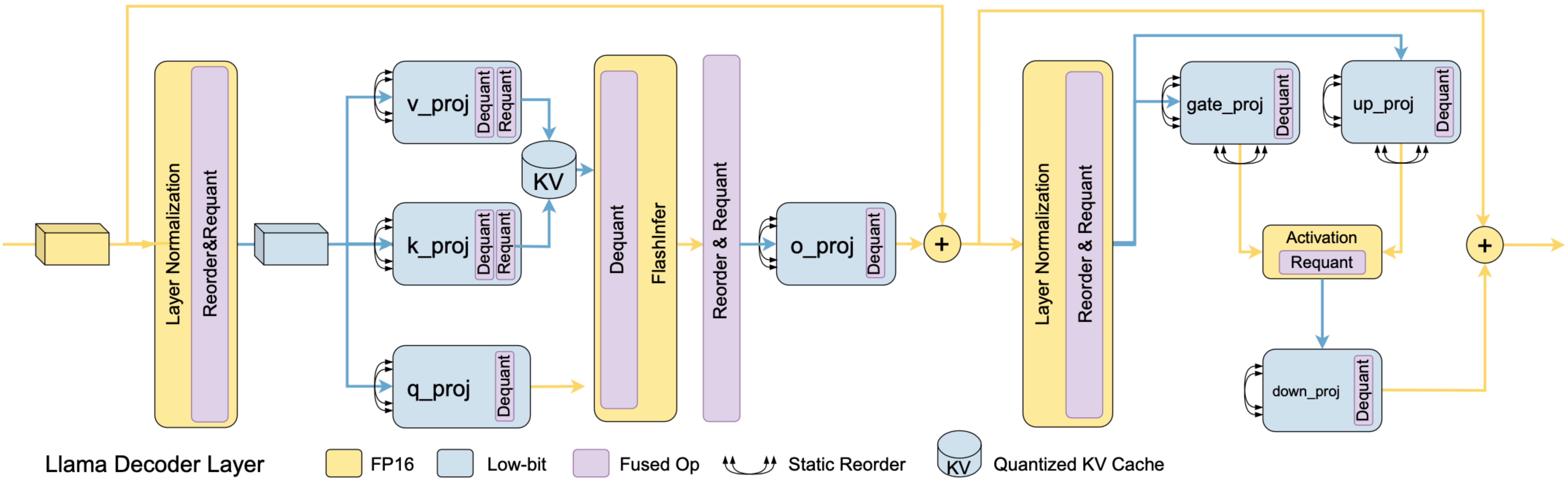
Atom: Low-Bit Quantization for Efficient and Accurate LLM Serving

Thank you!

Backup

Atom's Workflow

- Reordering and quantization are fused into LayerNorm
- De-quantization is fused into GEMM and Self-Attention kernel



Atom's Workflow for a Singe Decoder Block

Ablation on Quantization Techniques

Table 4. Ablation study on different quantization techniques used in Atom. The model used in this table is Llama-7B.

Quantization method	WikiText2 PPL↓
FP16 baseline	5.68
W4A4 RTN	2315.52
+ Keeping 128 outliers in FP16	11.34 (2304.2↓)
+ Quantizing outliers to INT8	11.39 (0.05↑)
+ Group size 128	6.22 (5.17↓)
+ Clipping	6.13 (0.09↓)
+ GPTQ	6.04 (0.09↓)
+ Quantizing KV-cache to INT4	6.16 (0.12↑)

Ablation on Reordering

Batch	16	32	64	128	256
Naive	47.58	47.25	46.74	47.64	48.14
Reorder	31.49	31.76	32.11	32.9	36.42
Speedup	**33.8%**	**32.8%**	**31.3%**	**30.9%**	**24.35%**