

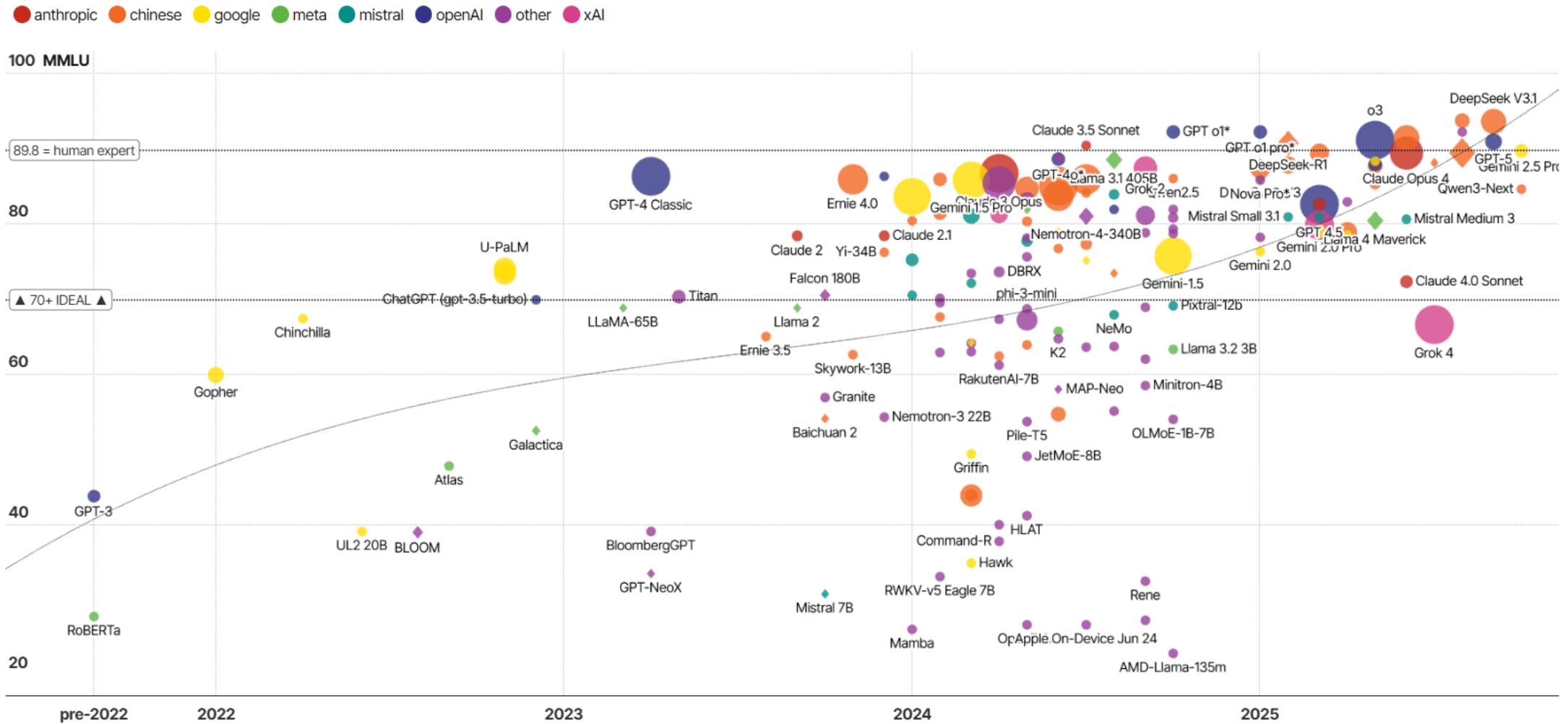
# 大模型量化剪枝：挑战、方法与机遇

——从参数分布视角谈起

柏昊立 华为香港研究所

2025-11-19

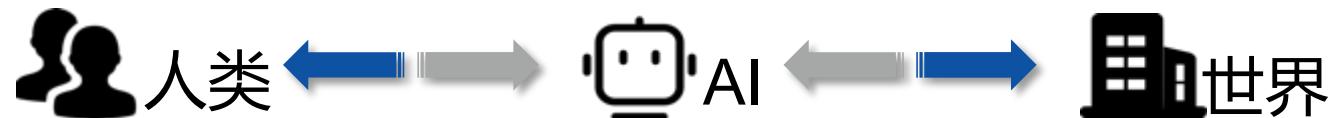
# 大模型演进：尺寸 vs. 智能



1. <https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-langs-like-chatgpt/>

# 大模型推理应用挑战

模型尺寸, 输入长度  $\uparrow$   $\rightleftharpoons$  推理开销 (时延、存储、能耗)  $\uparrow$



Deployment



云端

端侧

现实场景中的常见约束指标

High  
Throughput

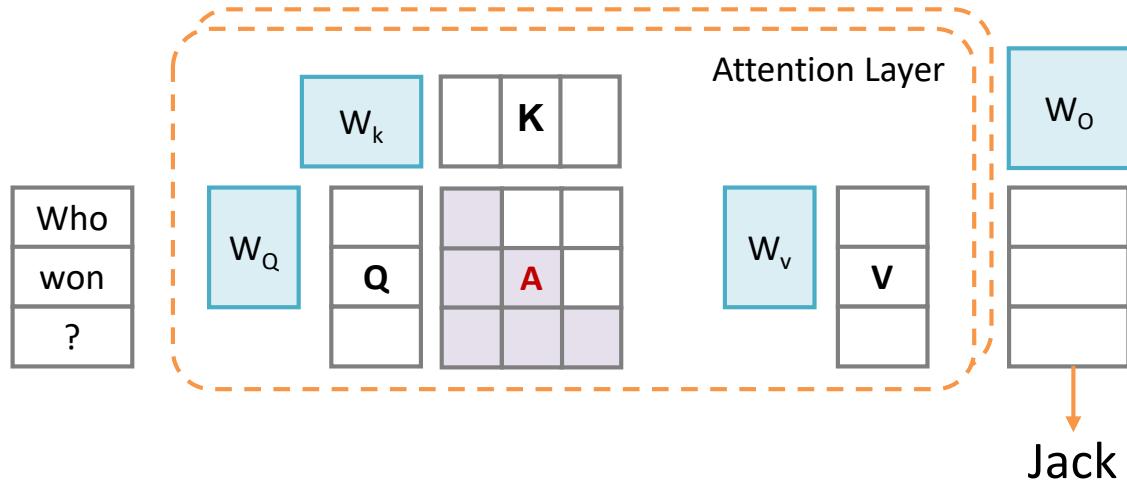
Low  
Latency

Small  
Storage

Low Energy  
Consumption

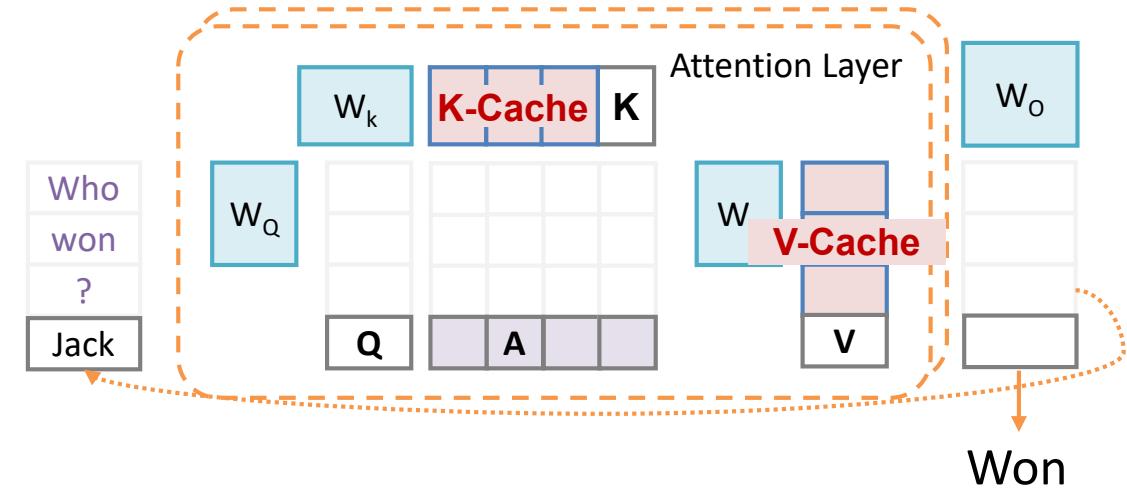
# 大模型推理范式

## 预填充 (Prefill)



- 大模型计算并存储用户输入N个token的KV Cache
- 计算N个token对应的query, key, value
- 注意力矩阵:  $N \times N$  full attention

## 解码 (Decode)



- 大模型计算, 存储用户所有输入token的KV Cache
- 只计算1个query, key, value, 读取N个KV Cache
- 注意力矩阵:  $1 \times (N+1)$

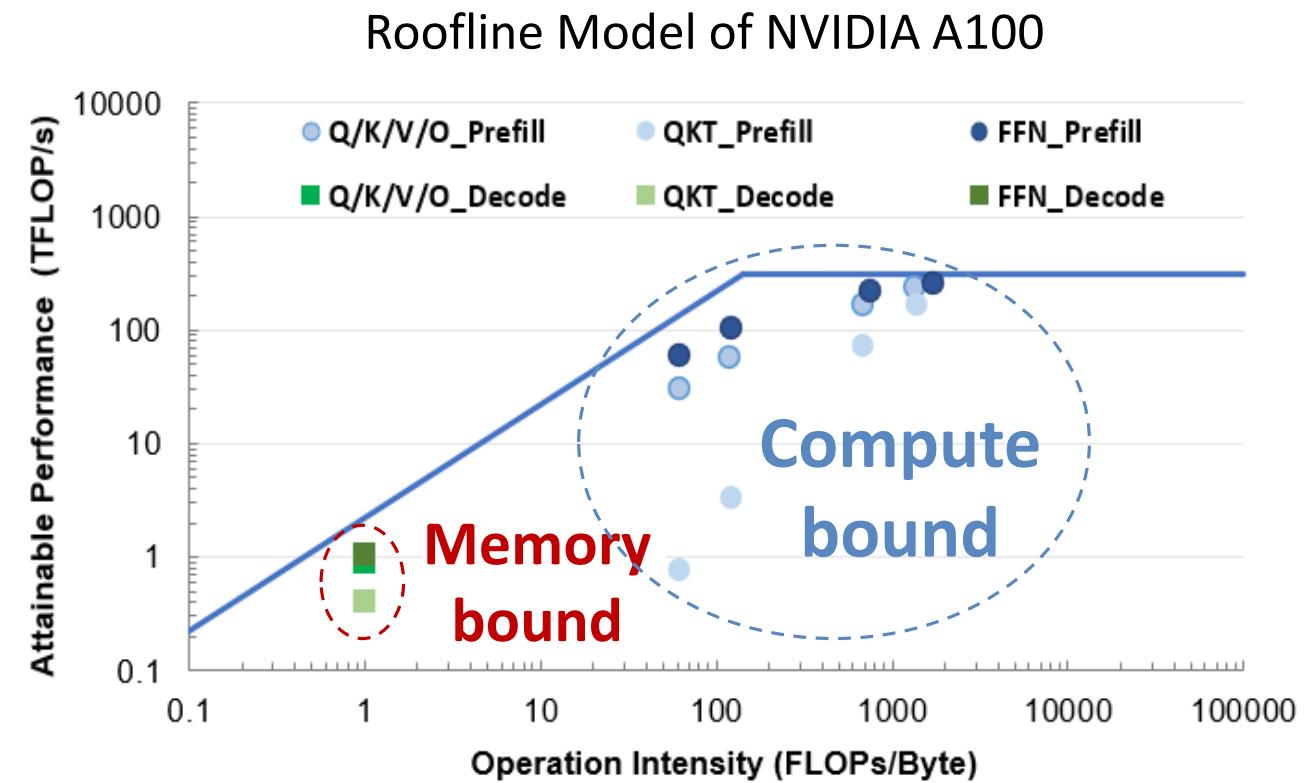
# The Roofline Model

Prefill: 计算密集型  
(Compute-bound)

Decode: 访存密集型  
Memory-bound

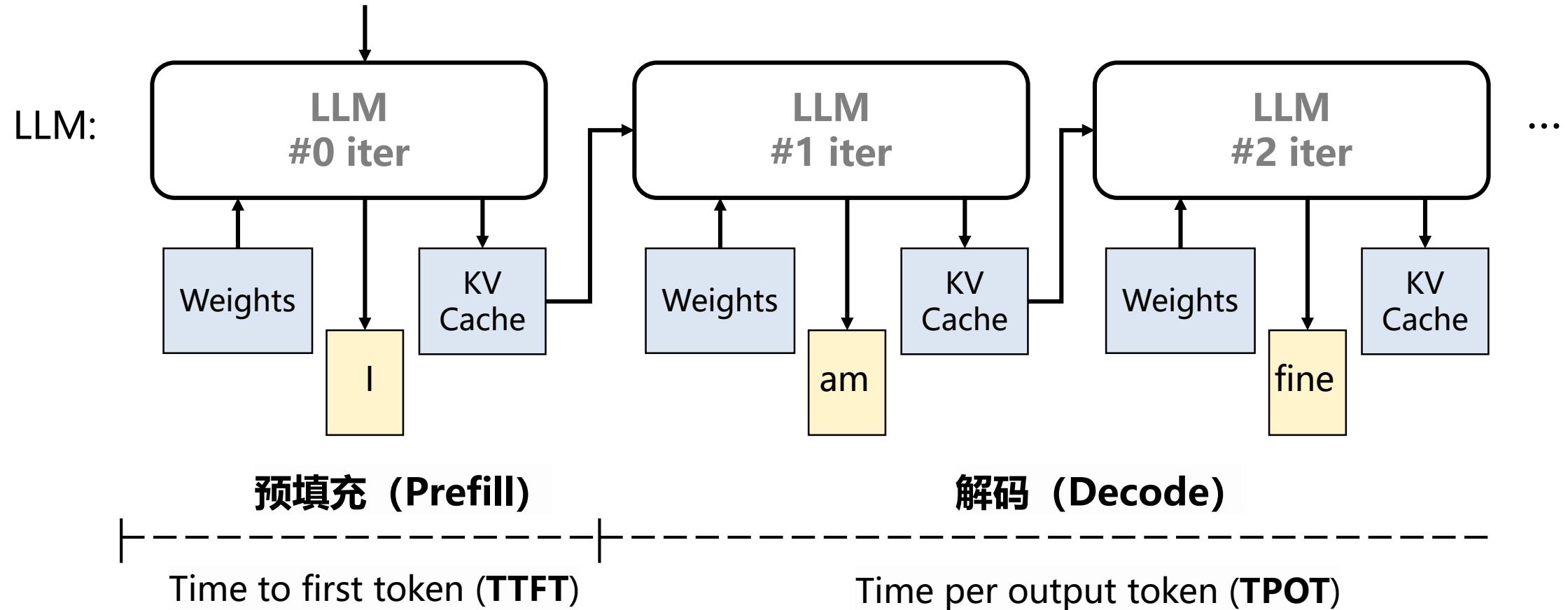
	Prefill	Decode
$Q/K/V/O$	$\frac{ld}{l+d}$	$\frac{d}{d+1}$
$QK^T$	$l$	$\frac{l}{l+1}$
$AttenV$	$l$	$\frac{l}{l+1}$
$FFN$	$\frac{ld_{FFN}}{l+d_{FFN}}$	$\frac{d_{FFN}}{d_{FFN}+1}$

Operational intensity at different stages  
(FLOPs/Byte)



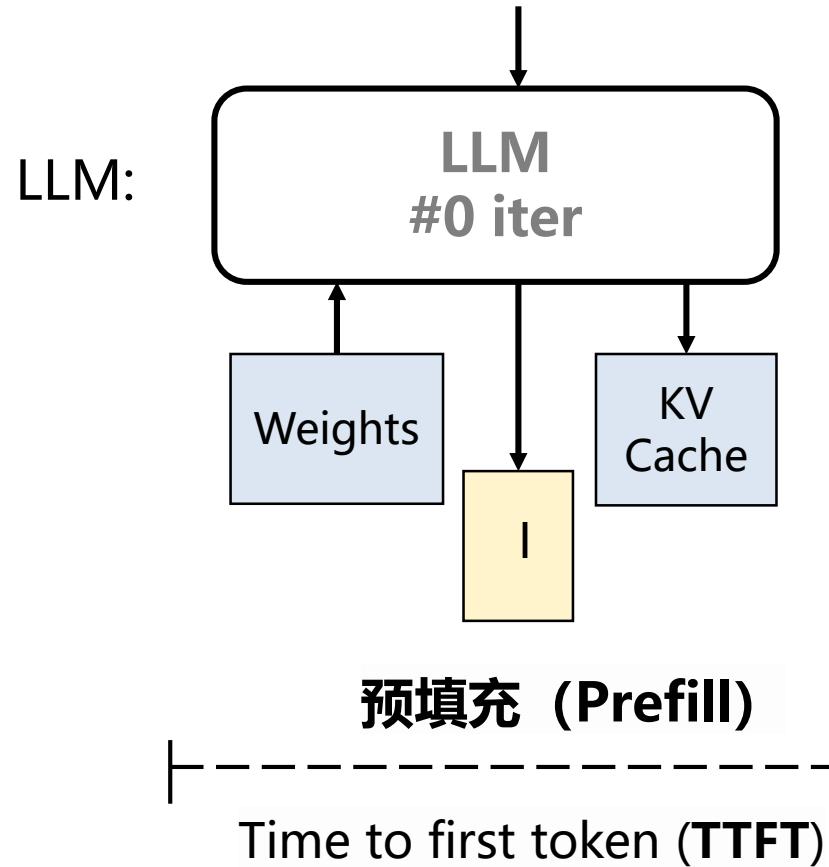
# 大模型推理范式

用户: How are you today?

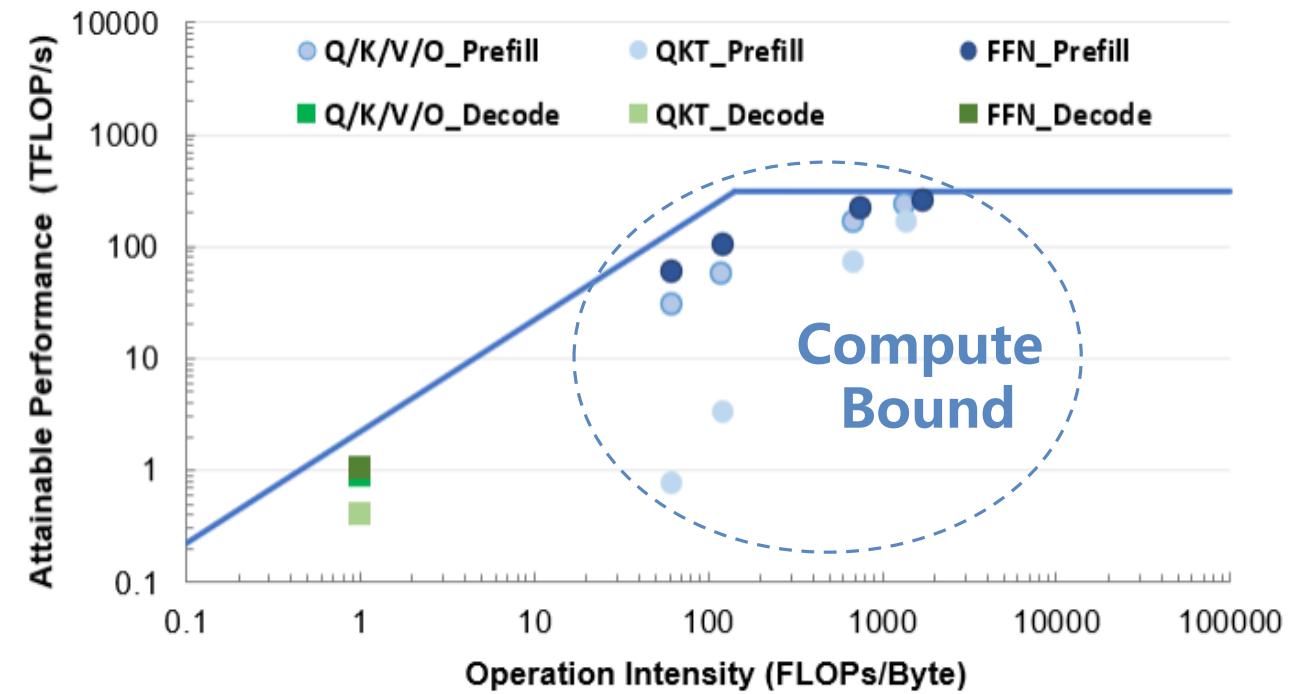


# 大模型推理范式

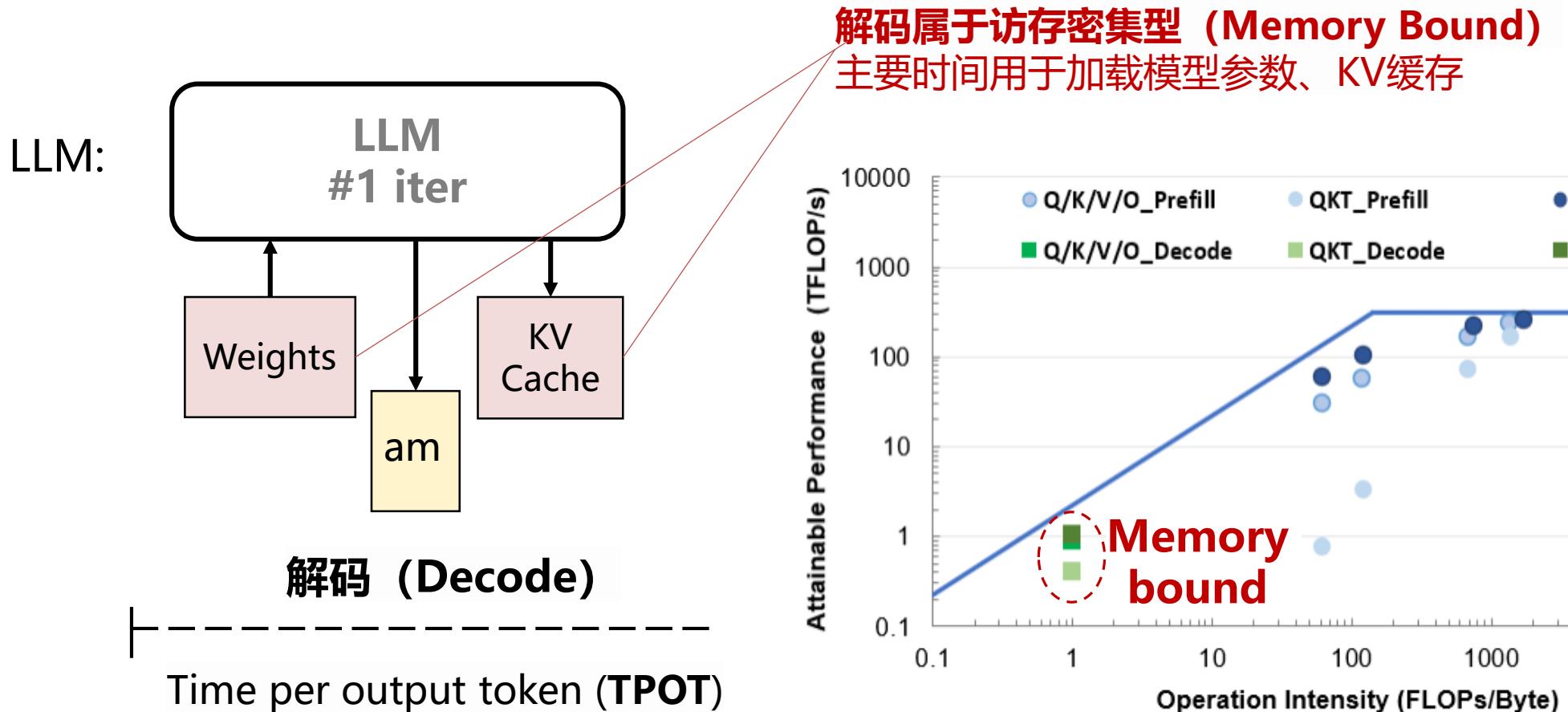
用户: How are you today?



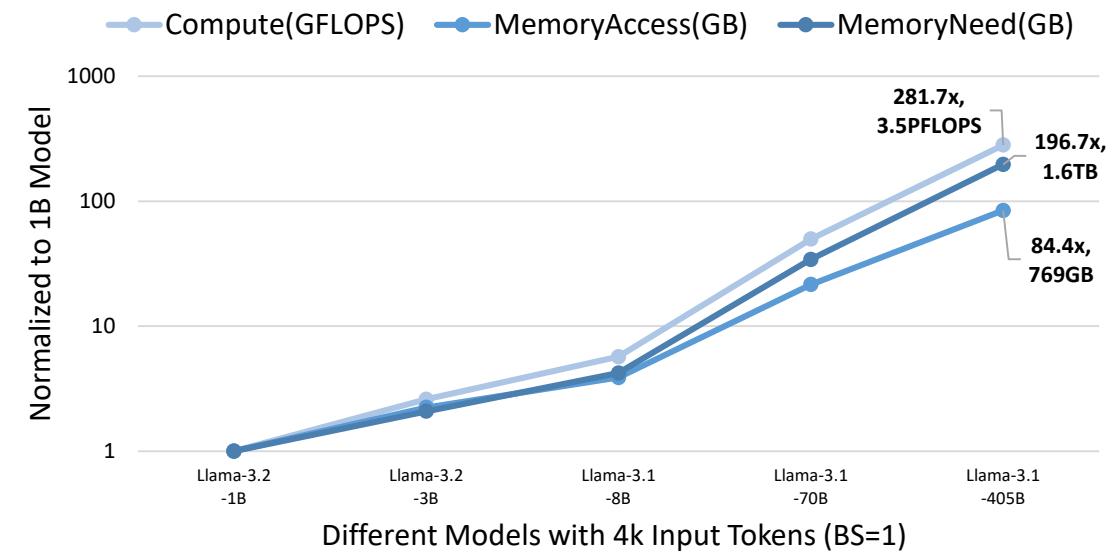
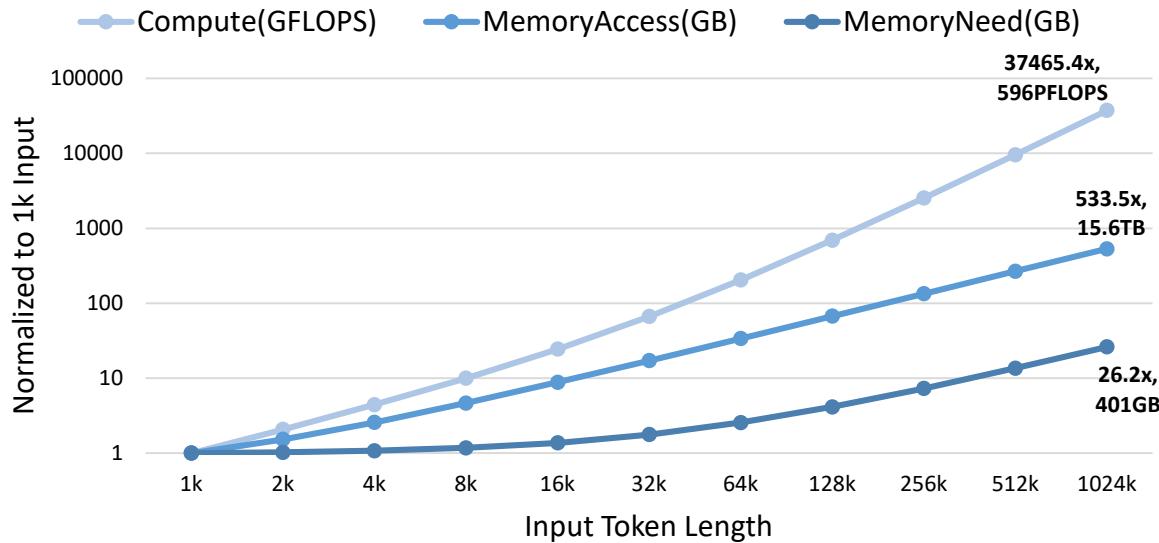
预填充属于计算密集型 (Compute Bound)  
主要时间用于计算用户请求下的self-attention、FFN



# 大模型推理范式



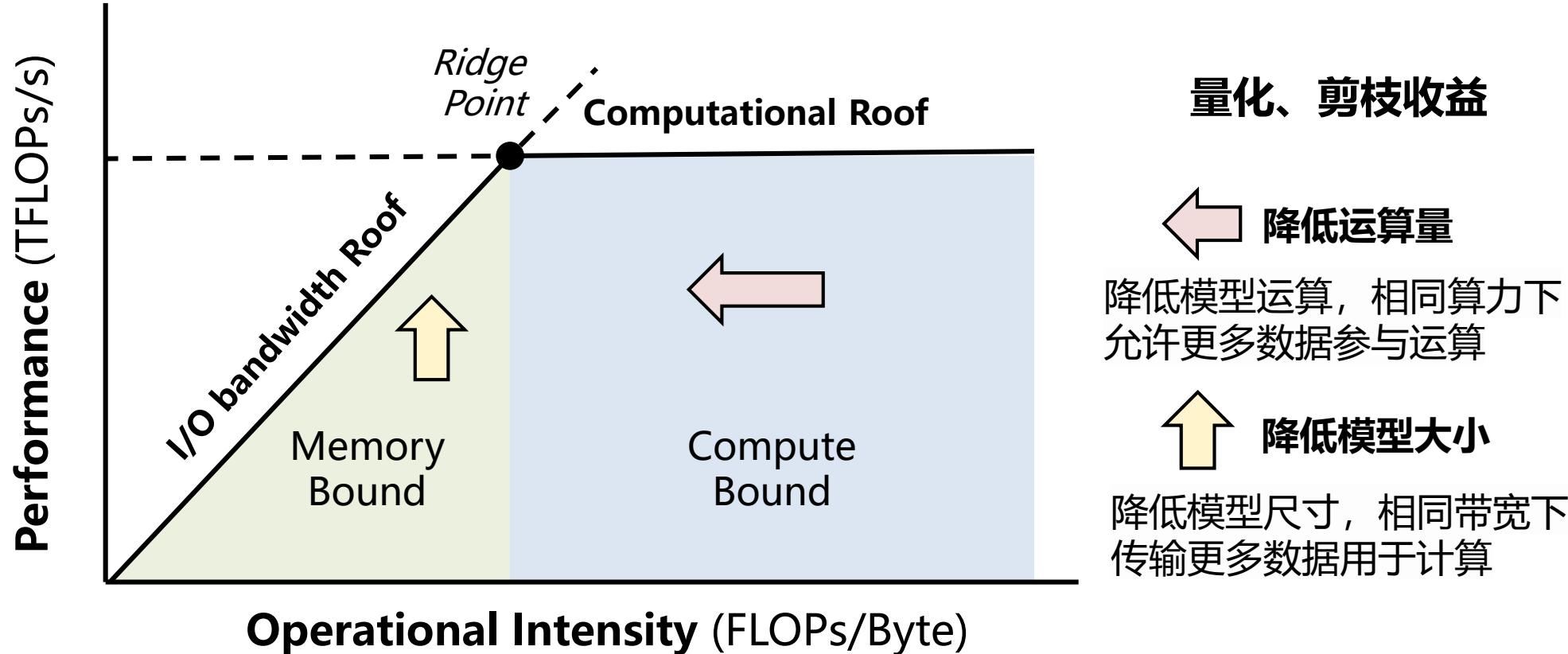
# 大模型推理成本



8B Model	Compute Amount	Memory Access	VRAM Consume	GPU	TTFT*	TPOT*
1K token	15.9 TFLOPs	29.2 GB	15.3 GB	1xA100	73~170 ms	10~24 ms
1M token	597 PFLOPs	15.6 TB	401 GB	6xA100	7.6~17.7 min	32~75 ms

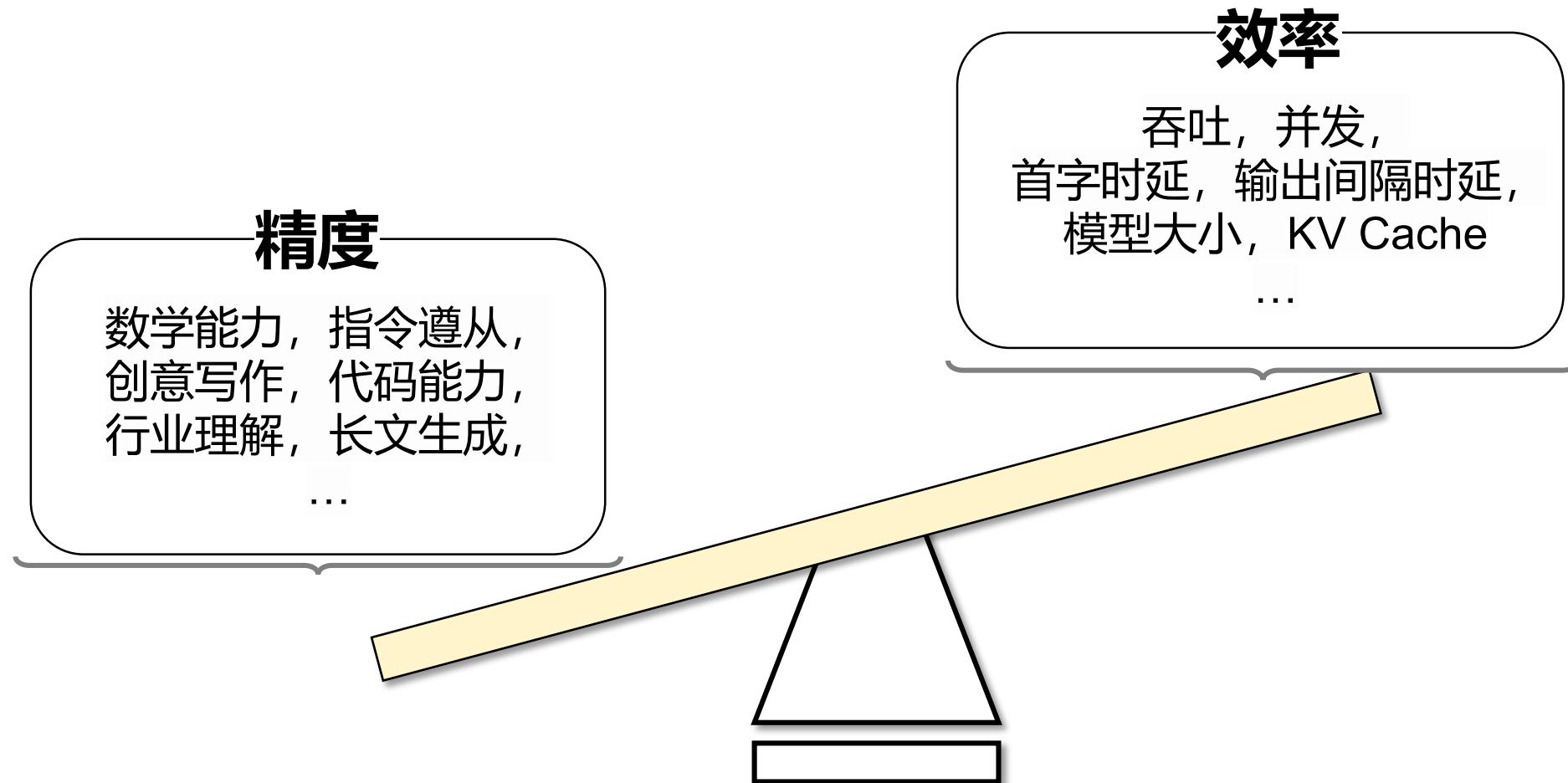
4K Tokens	Compute Amount	Memory Access	VRAM Consume	GPU	TTFT*	TPOT*
1B Model	12.4 TFLOPs	19.2 GB	15.3 GB	1xA100	57~132 ms	1.8~4.3 ms
405B Model	3.5 PFLOPs	1.6 TB	769 GB	10xA100	1.6~3.7 s	55~128 ms

# 大模型量化剪枝：Roofline模型视角



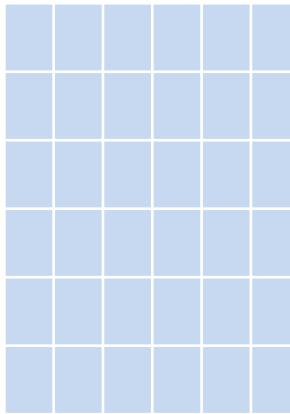
1. Dao T, Fu D, Ermon S, et al. Flashattention: Fast and memory-efficient exact attention with io-awareness[J]. Advances in neural information processing systems, 2022, 35: 16344-16359.

# 大模型部署权衡 - 效率与精度



# 大模型量化剪枝：减少访存消耗，降低运算量

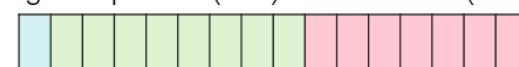
原始模型



参数/激活按BF16数值存储

sign exponent (8bit) fraction bits (7bit)

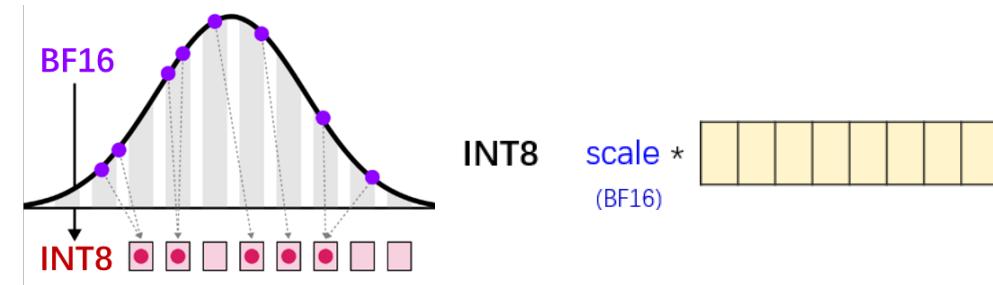
BF16



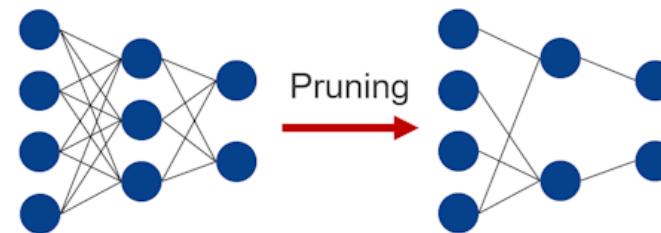
2	3	1	0	1	3
.	.	.			

0	0	0
0	.	.

模型量化：用低比特整数近似表达原BF16参数/激活值



模型剪枝：剪除模型中不重要的参数（通道/模块）

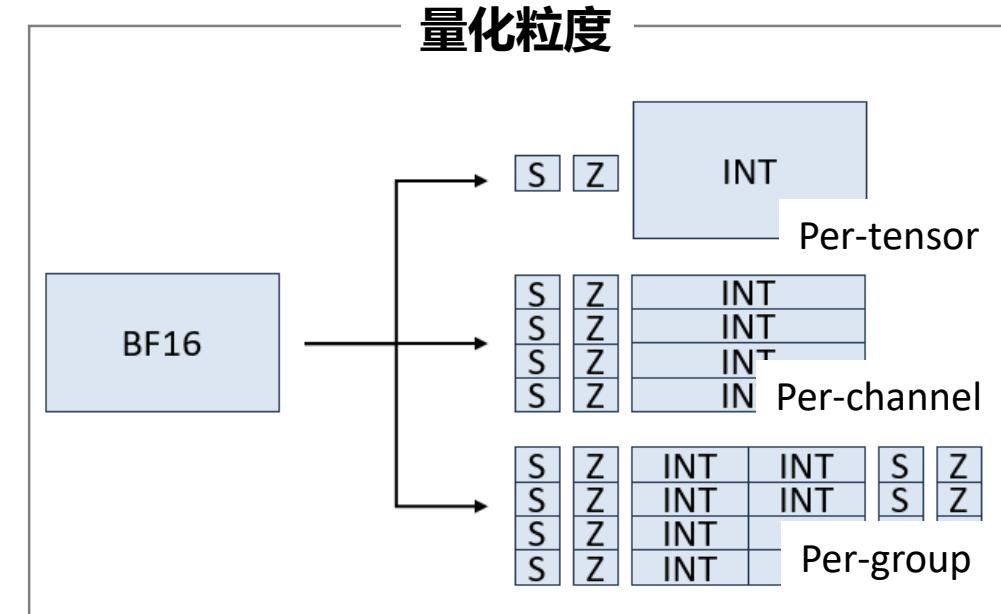
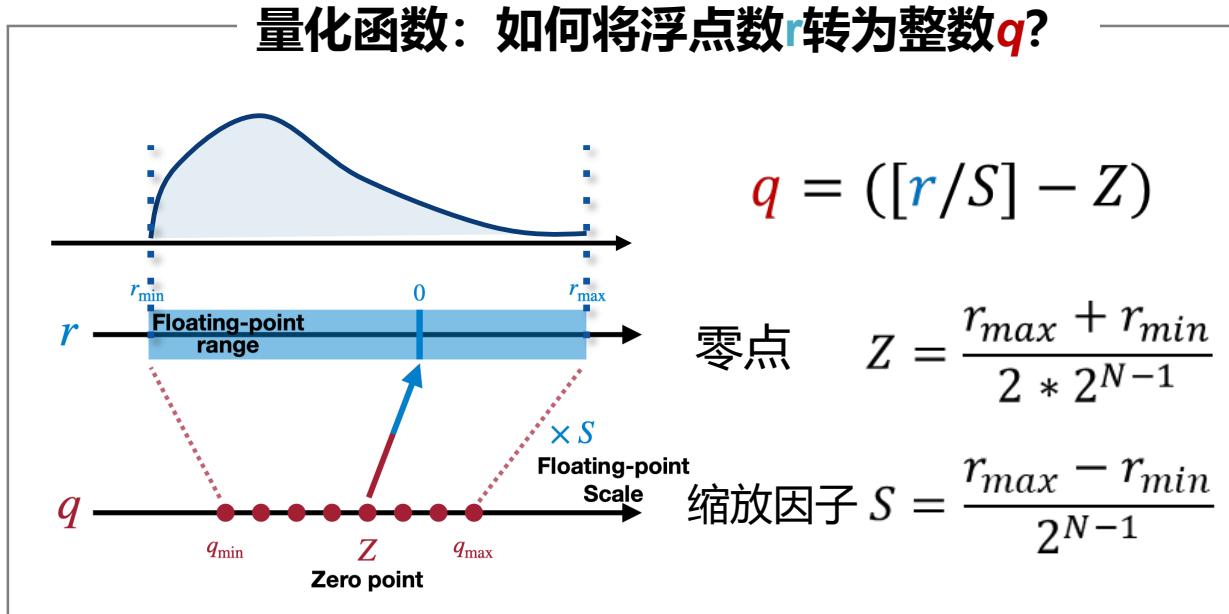


**Part 1: 大模型量化**

**Part 2: 大模型剪枝**

**Part 3: 未来展望**

# 模型量化回顾

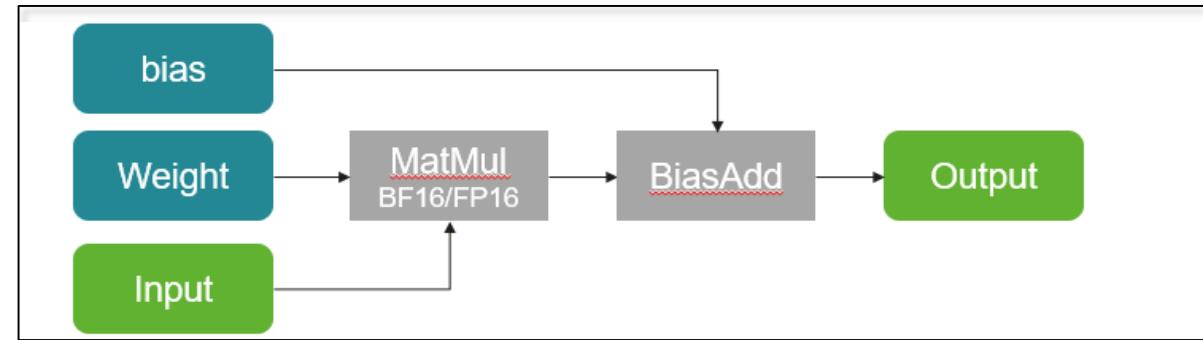


1. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, Han et al., ICLR 2016.

# 模型量化回顾

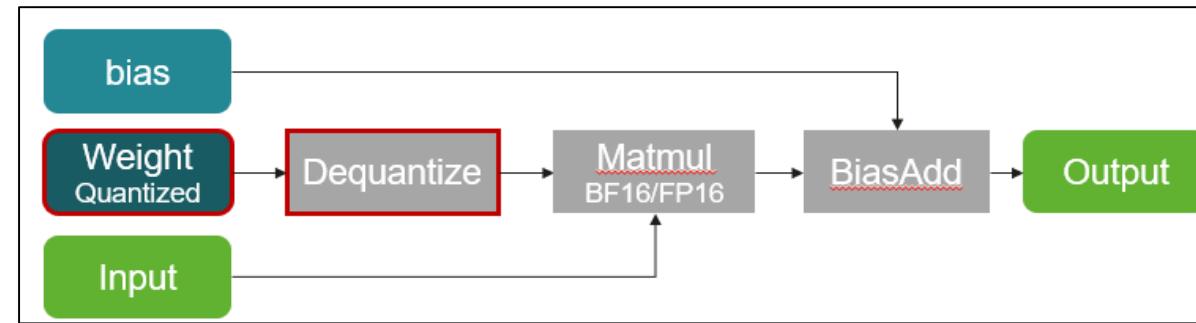
## 正常推理

模型参数&激活值: BF16



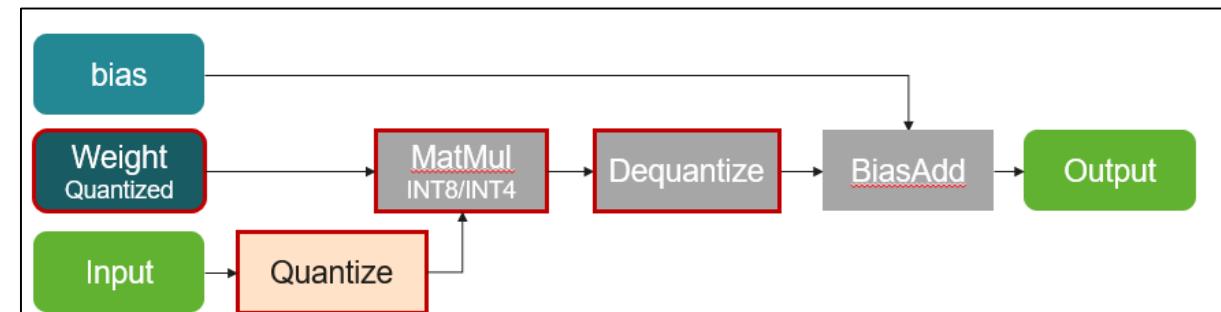
## 权重量化

降低访存, 加速Decode



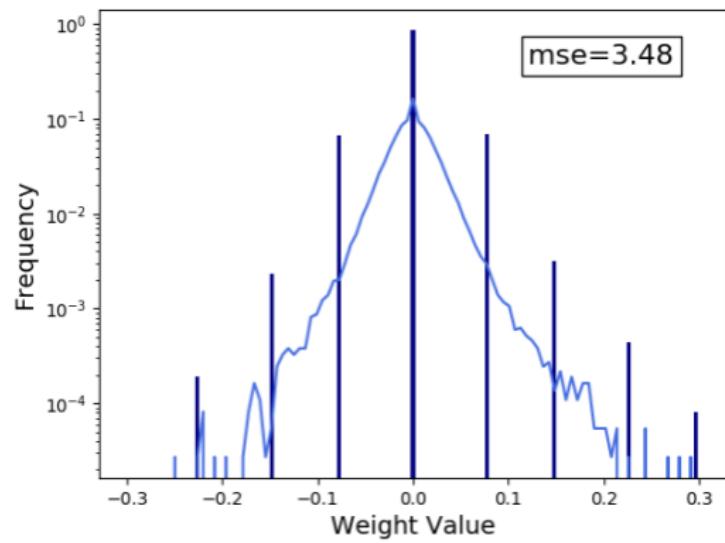
## 权重-激活量化

降低访存、计算量  
加速Prefill & Decode



# 选择合适量化步长，降低量化误差

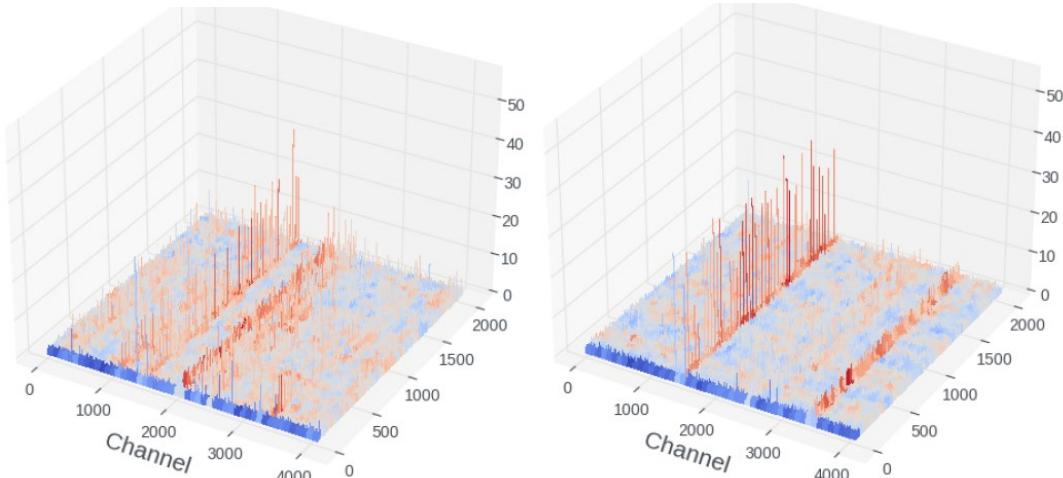
量化研究核心：选择合适量化步长，拟合参数/激活值分布



# 挑战：大模型Activation的离群值

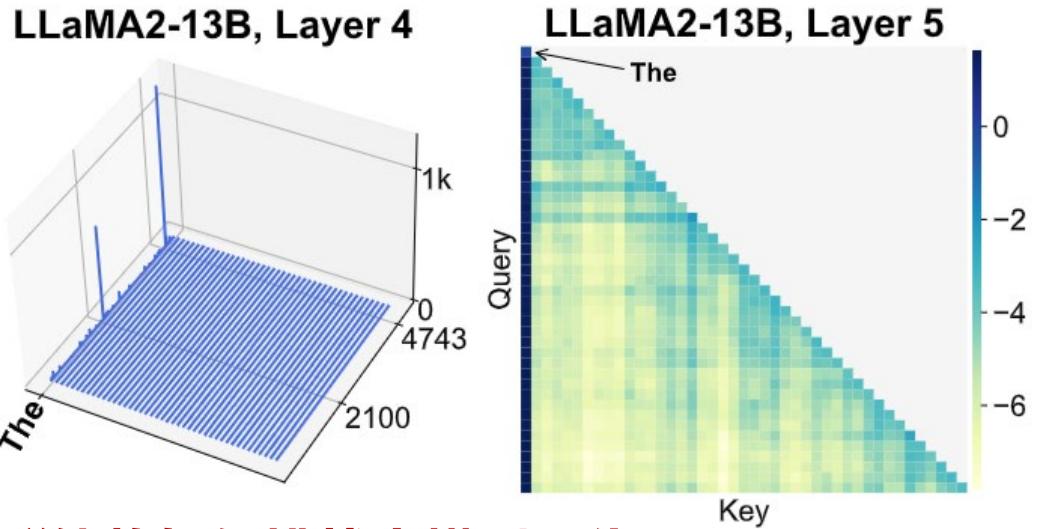
## Type I: 固定离群值

- 沿着某通道固定存在并可跨层，输入不变性
- 幅值相对大( $1e1 \sim 1e2$ )，对精度影响高



## Type II: 极大离群值 (Massive Activations)

- 在特定token出现：[BOS], 标点符号、分隔符
- 激活值远大于其他元素 ( $>1e3$ )，注意力得分集中

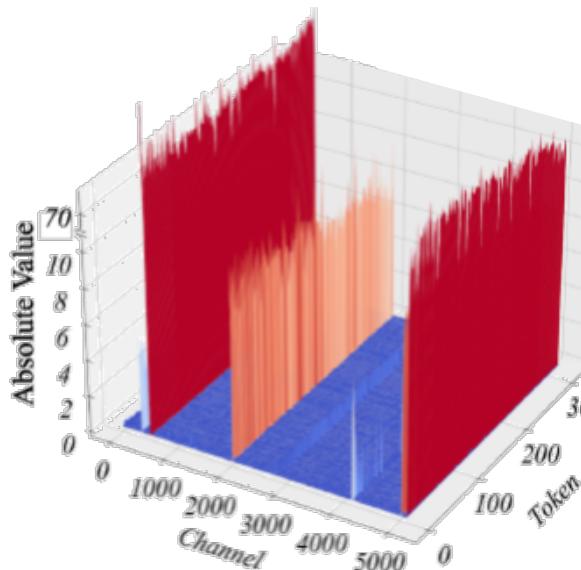


—— 这些离群值将如何挑战大模型量化？

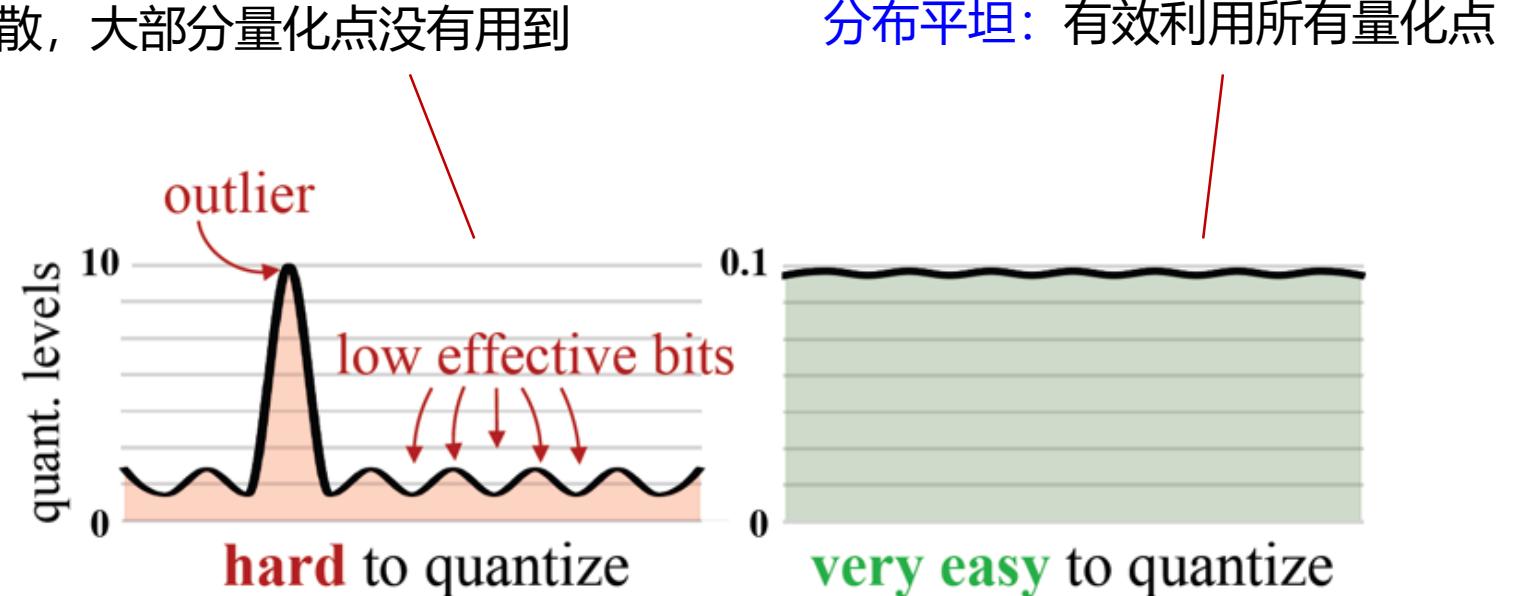
1. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. NeurIPS 2022.
2. Liu Z, Zhao C, Fedorov I, et al. Spinquant: Llm quantization with learned rotations[J]. ICLR 2025.
3. Mingjie Sun, Xinlei Chen, J. Zico Kolter, Zhuang Liu. Massive Activations in Large Language Models. arXiv preprint: 2402.17762.

# 量化挑战：Type I 固定离群值

- 量化步长选择不易！大模型中普遍存在Type I固定离群值



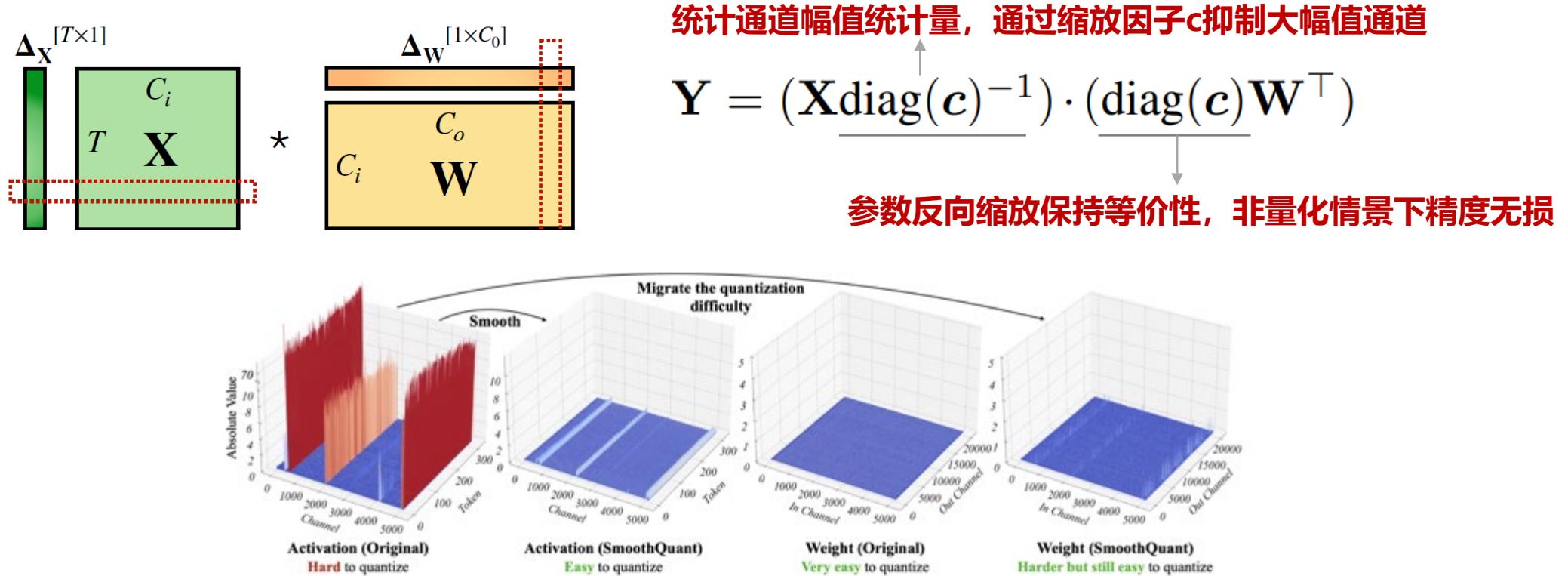
**分布畸形：**离群值的存在导致量化点过于分散，大部分量化点没有用到



1. Xiao G, Lin J, Seznec M, et al. Smoothquant: Accurate and efficient post-training quantization for large language models, ICML 2024.

# 如何得到平坦量化分布?

- SmoothQuant 逐通道缩放：每个通道乘以一个缩放因子，抑制离群值



1. Xiao G, Lin J, Seznec M, et al. Smoothquant: Accurate and efficient post-training quantization for large language models[C]//International Conference on Machine Learning. PMLR, 2023: 38087-38099.

# 如何得到平坦量化分布?

- **AWQ 逐通道缩放**: 每个通道乘以一个缩放因子, 抑制离群值

$$\mathbf{s}^* = \arg \min_{\mathbf{s}} \mathcal{L}(\mathbf{s}), \quad \mathcal{L}(\mathbf{s}) = \|Q(\mathbf{W} \cdot \mathbf{s})(\mathbf{s}^{-1} \cdot \mathbf{X}) - \mathbf{WX}\|$$

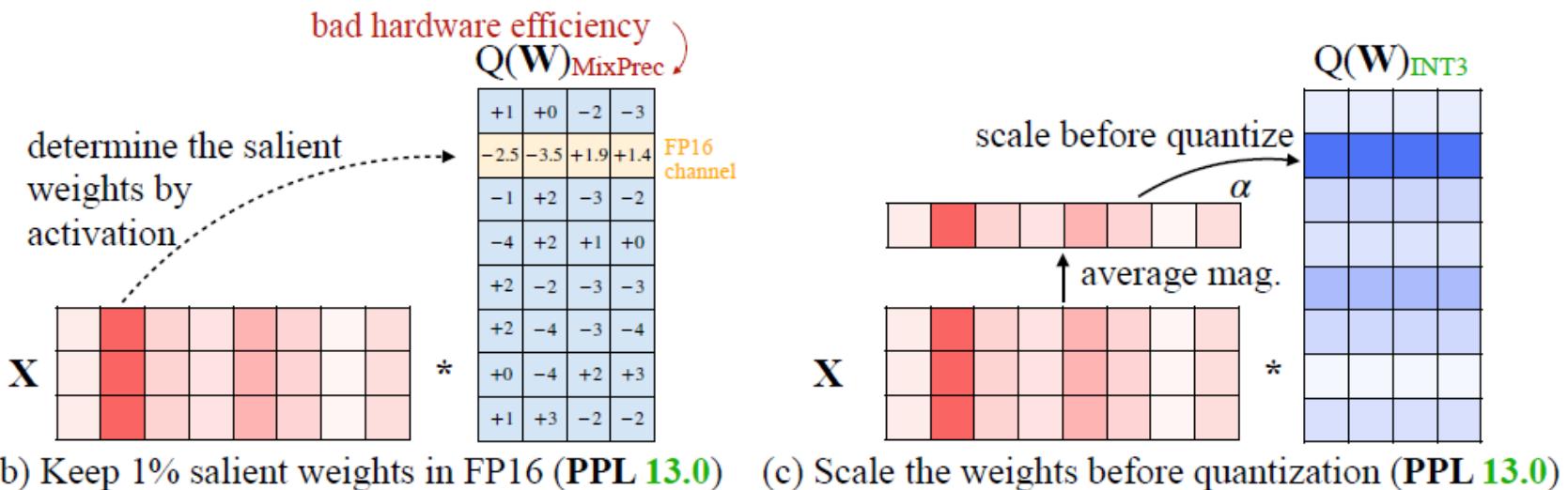
**W<sub>FP16</sub>**

+1.2	-0.2	-2.4	-3.4
-2.5	-3.5	+1.9	+1.4
-0.9	+1.6	-2.5	-1.9
-3.5	+1.5	+0.5	-0.1
+1.8	-1.6	-3.2	-3.4
+2.4	-3.5	-2.8	-3.9
+0.1	-3.8	+2.4	+3.4
+0.9	+3.3	-1.9	-2.3

**Q(W)<sub>INT3</sub>**

+1	+0	-2	-3
-3	-4	+2	+1
-1	+2	-3	-2
-4	+2	+1	+0
+2	-2	-3	-3
+2	-4	-3	-4
+0	-4	+2	+3
+1	+3	-2	-2

→



1. Xiao G, Lin J, Seznec M, et al. Smoothquant: Accurate and efficient post-training quantization for large language models[C]//International Conference on Machine Learning. PMLR, 2023: 38087-38099.
2. Lin J, Tang J, Tang H, et al. Awq: Activation-aware weight quantization for on-device llm compression and acceleration[J]. Proceedings of Machine Learning and Systems, 2024, 6: 87-100.

# 如何得到平坦量化分布?

- **旋转变换**: 在旋转后空间进行量化, 利用正交变换等价性

$$\mathbf{Y} = \mathbf{XW}^\top = (\mathbf{XH})(\mathbf{H}^\top \mathbf{W}^\top)$$

- QuaRot [1] Hadamard变换

$$\mathbf{H}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \mathbf{H}_{2^n} = \mathbf{H}_2 \otimes \mathbf{H}_{2^{n-1}}.$$

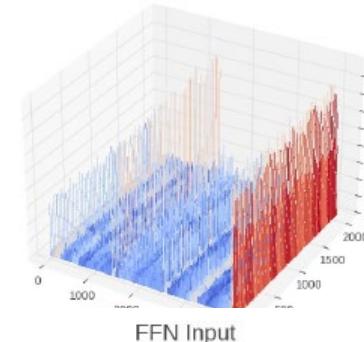
- SpinQuant [2] 可学习的旋转变换  $\mathbf{R}$



还有更好的变换方式吗?

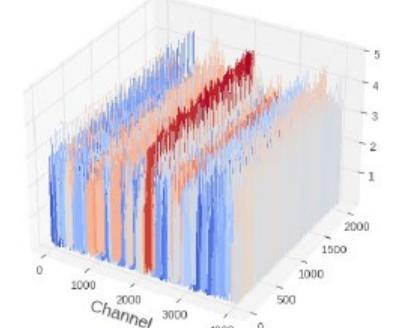
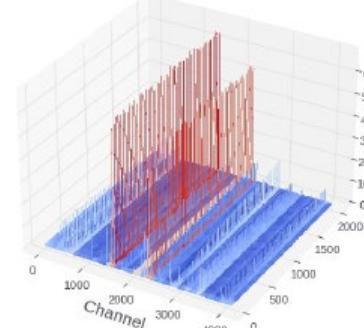
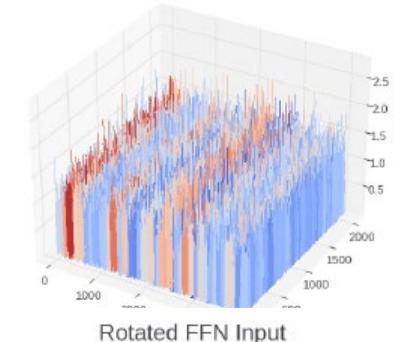
旋转前分布

MHSA Input



旋转后分布

Rotated MHSA Input



1. Ashkboos S, Mohtashami A, Croci M, et al. Quarot: Outlier-free 4-bit inference in rotated l1ms[J]. Advances in Neural Information Processing Systems, 2024, 37: 100213-100240.
2. Liu Z, Zhao C, Fedorov I, et al. Spinquant: Llm quantization with learned rotations[J]. ICLR 2025.

# FlatQuant: 大模型的平坦量化

- **可学习的线性变换:** 利用可逆矩阵降低量化误差, 为每个线性层定制变换方式

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \|\mathbf{Y} - \mathcal{Q}(\mathbf{X}\mathbf{P})\mathcal{Q}(\mathbf{P}^{-1}\mathbf{W}^\top)\|_F^2,$$

- 所有线性层都需要存储变换矩阵P, 存储和计算开销过大

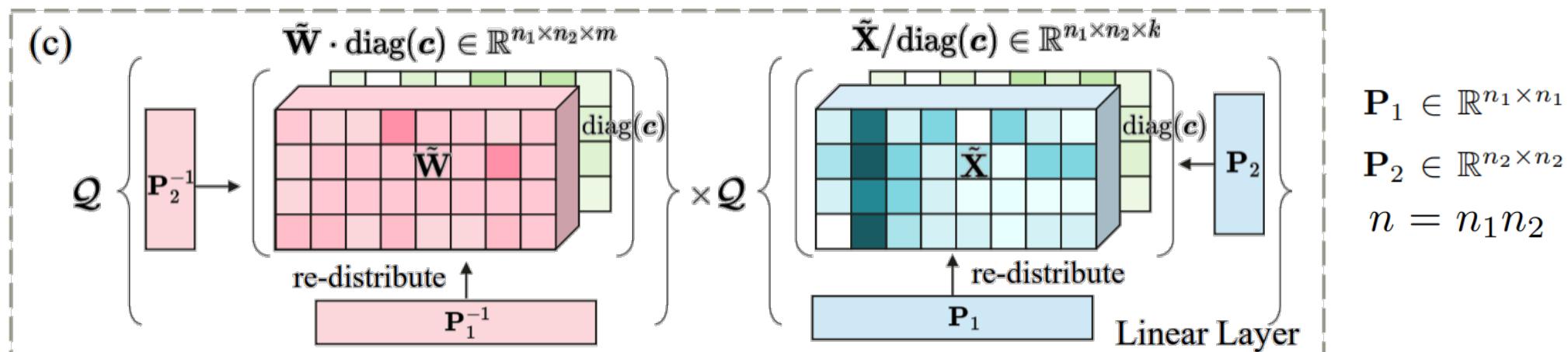
# FlatQuant: 大模型的平坦量化

- 可学习的线性变换: 利用可逆矩阵降低量化误差, 为每个线性层定制变换方式

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \|\mathbf{Y} - \mathcal{Q}(\mathbf{X}\mathbf{P})\mathcal{Q}(\mathbf{P}^{-1}\mathbf{W}^\top)\|_F^2,$$

- 所有线性层都需要存储变换矩阵 $\mathbf{P}$ , 存储和计算开销过大
- Kronecker Product 降低线性变换开销

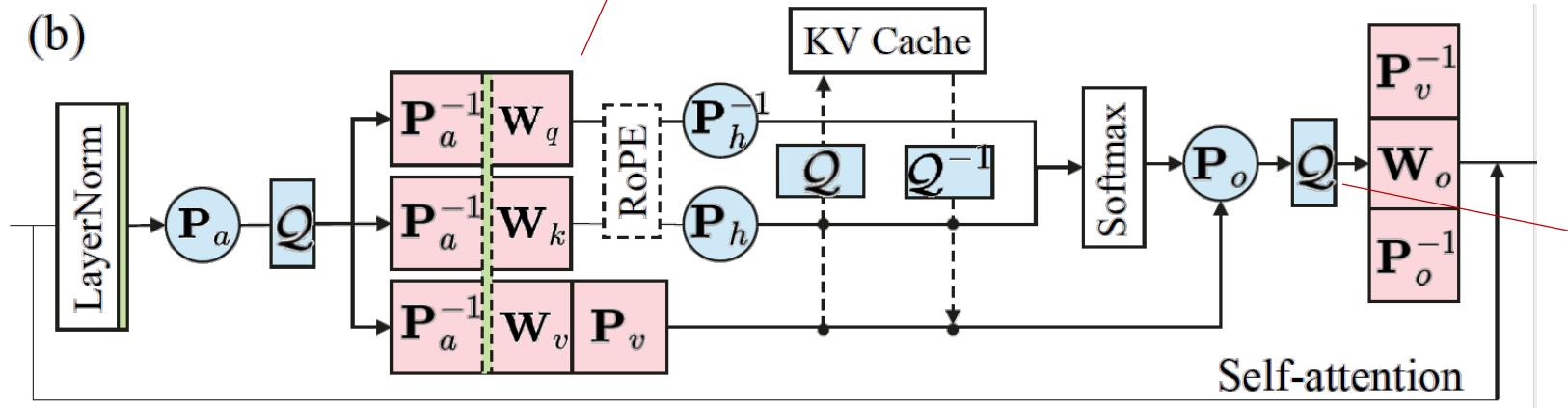
$$\mathcal{Q}(\mathbf{X}\mathbf{P})\mathcal{Q}(\mathbf{P}^{-1}\mathbf{W}^\top) = \mathcal{Q}(\mathbf{P}_1^\top \times_1 \tilde{\mathbf{X}} \times_2 \mathbf{P}_2) \mathcal{Q}(\mathbf{P}_1^{-1} \times_1 \tilde{\mathbf{W}} \times_2 (\mathbf{P}_2^{-1})^\top)^\top,$$



# FlatQuant: 大模型的平坦量化

线性变换参数融合至模型参数，没有额外开销

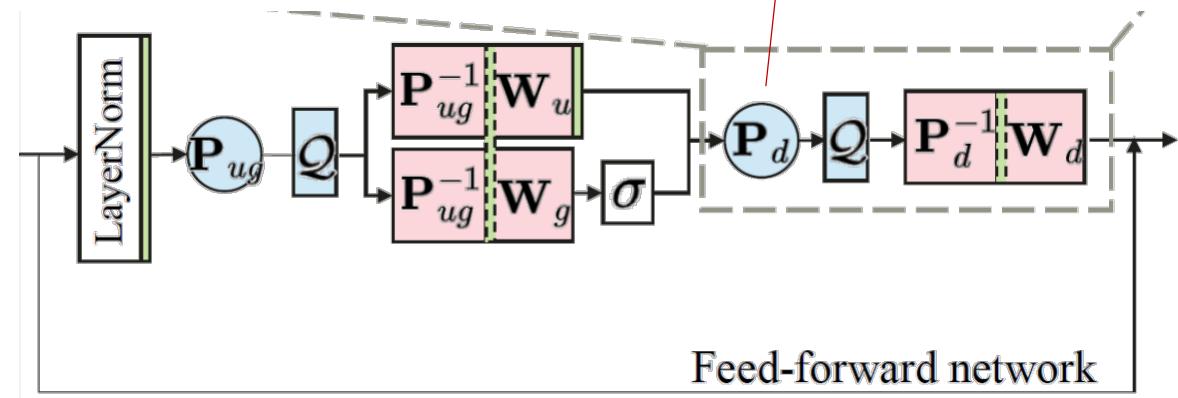
(b)



不可融合部分需要在线变换，  
存在一定推理开销

(a)

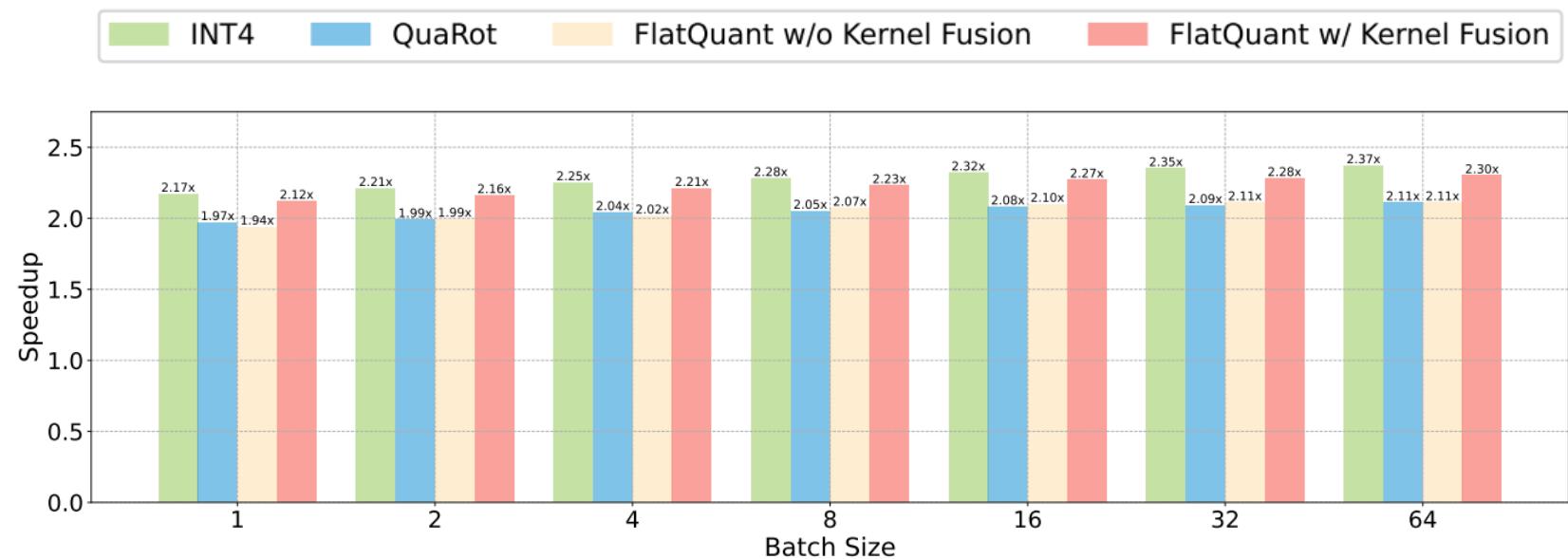
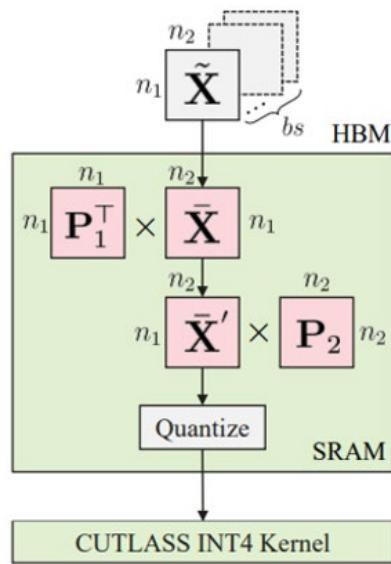
- $\mathbf{P}^{-1}$   $\mathbf{W}$  : Merged Weight (Quantized)
- ... : Pair-wisely Merged Scaling Vector
- $\mathcal{Q}$  /  $^{-1}$ : Quantization/Dequantization
- $\mathbf{P}$  : Affine Transformation Matrix



# 算子融合提升推理效率

- 线性变换乘法融合成单算子

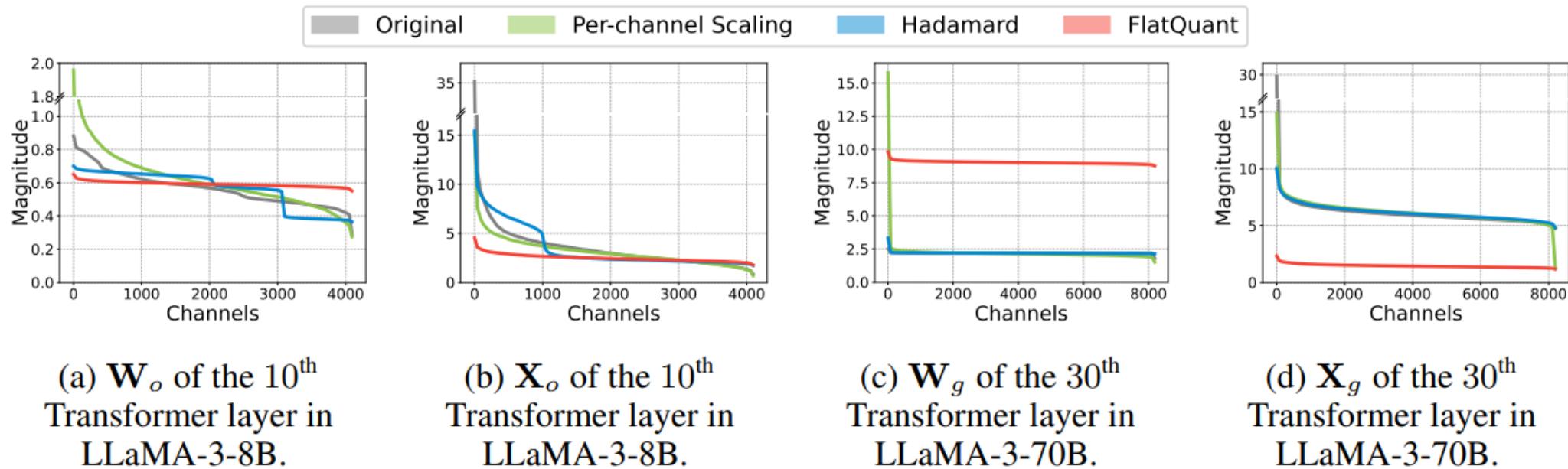
- 线性变换操作本身是访存瓶颈
- 减少HBM, L1/L2 Cache数据搬运，数据搬运从3次降为1次
- 线性变换优化后提速3-5倍，4bit量化Prefill端到端提速高达2.3倍



<https://github.com/ruikangliu/FlatQuant>

# 分布可视化

- 参数、激活按通道统计幅值
  - 从大到小排列，曲线越平缓，分布越平坦



# 主要结果

Model	Method	W Quantizer	ARC-C	ARC-E	HellaSwag	LAMBADA	PIQA	Winogrande	Avg
3-8B	FP16	-	53.50	77.57	79.12	75.51	80.74	72.93	73.23
	QuaRot	RTN	38.65	66.54	68.82	57.20	71.82	65.04	61.34
	SpinQuant	RTN	45.73	71.38	74.07	67.67	76.66	66.38	66.98
	FLATQUANT	RTN	50.00	75.80	76.80	72.91	79.16	72.69	71.23
	QuaRot	GPTQ	45.73	70.83	72.97	62.70	75.35	67.17	65.79
	SpinQuant	GPTQ	47.27	74.20	74.55	70.29	77.37	68.51	68.70
3-70B	FLATQUANT	GPTQ	50.51	75.88	76.49	73.20	79.00	72.93	71.33
	FP16	-	64.25	85.94	84.93	79.37	84.44	80.74	79.95
	QuaRot	RTN	22.18	34.30	32.15	13.35	57.67	52.49	35.36
	SpinQuant	RTN	44.03	69.07	74.57	63.34	76.99	65.98	65.66
	FLATQUANT	RTN	62.12	84.97	83.95	78.73	84.28	80.03	79.01
	QuaRot	GPTQ	49.49	74.37	77.22	71.69	78.89	71.03	70.45
3-130B	SpinQuant	GPTQ	51.96	77.40	77.29	71.90	79.33	72.06	71.66
	FLATQUANT	GPTQ	61.95	84.47	83.87	77.99	83.95	79.24	78.58

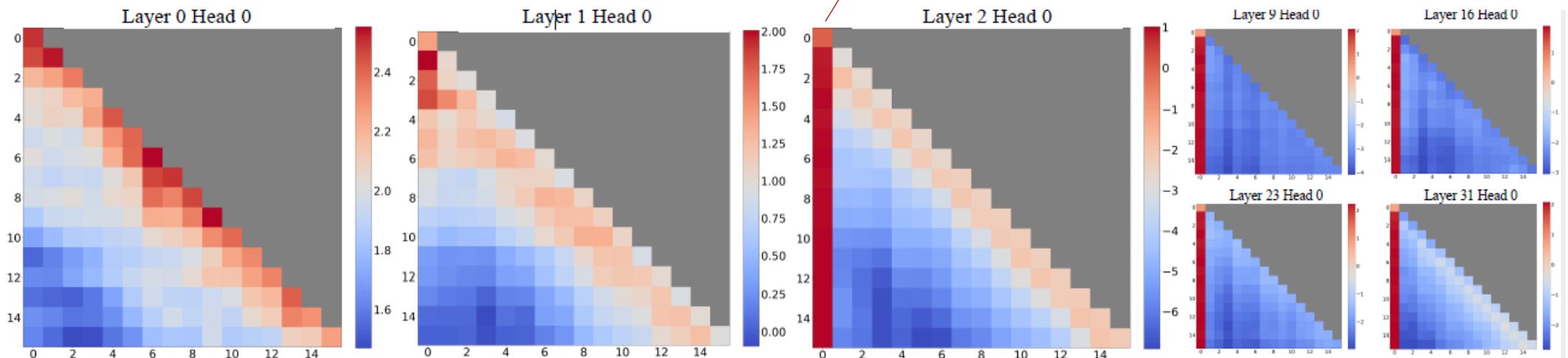
- 精度显著提升：W4A4KV4后量化精度损失控制在1%-2%
- 基于FlatQuant， RTN可以得到与GPTQ同样有竞争力的结果，并且更快（减少1-2小时）

1. <https://github.com/ruijangliu/FlatQuant>. FlatQuant已支持LLaMA, Qwen, DeepSeek v3系列模型，兼容vLLM推理框架，实际部署量化流程已开源

# 量化挑战：Type II 特大离群值

- **注意力槽现象：**某些token占据绝大部分注意力得分
  - 与[2]同期发现：Type II特大离群值和注意力槽现象有密切管理

自注意力图中某些词元占据90%以上注意力得分

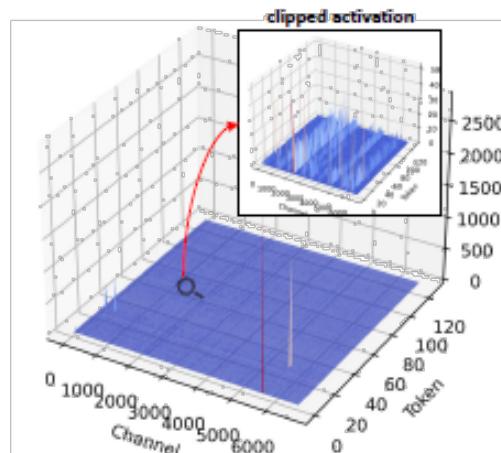


主要是哪些词元？[BOS], \n, \t, 标点符号等

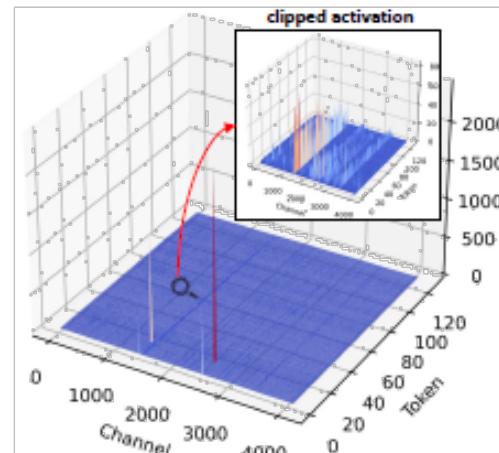
1. Xiao G, Tian Y, Chen B, et al. Efficient streaming language models with attention sinks[J]. arXiv preprint arXiv:2309.17453, 2023.
2. Mingjie Sun, Xinlei Chen, J. Zico Kolter, Zhuang Liu. Massive Activations in Large Language Models. arXiv preprint: 2402.17762.

# 量化挑战：Type II 特大离群值

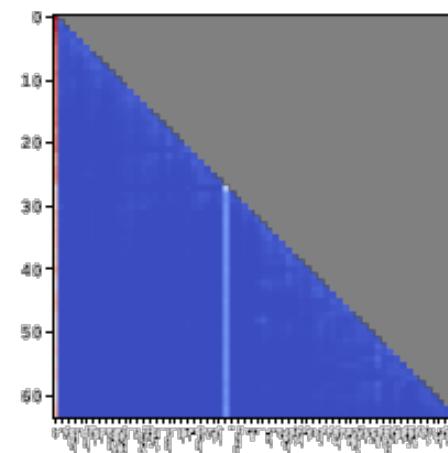
- 注意力槽导致特大异常值 ( $\sim 1e3$ )
  - Attention output & FFN output
  - 对量化有何影响？



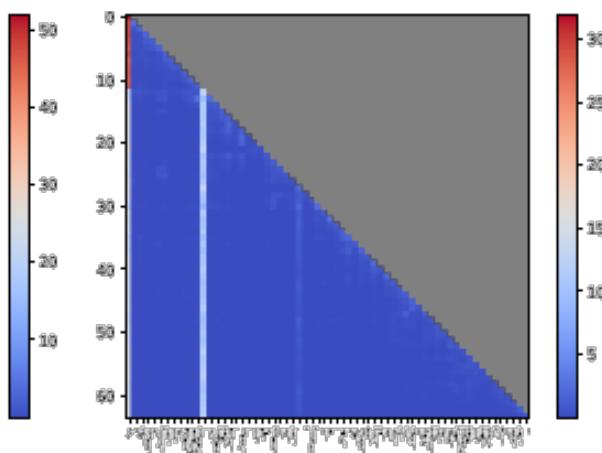
(a) Output activations of  
LLaMA-30B Layer 24



(b) Output activations of  
LLaMA-2-7B Layer 24



(c) Attention map of  
LLaMA-30B Layer 24

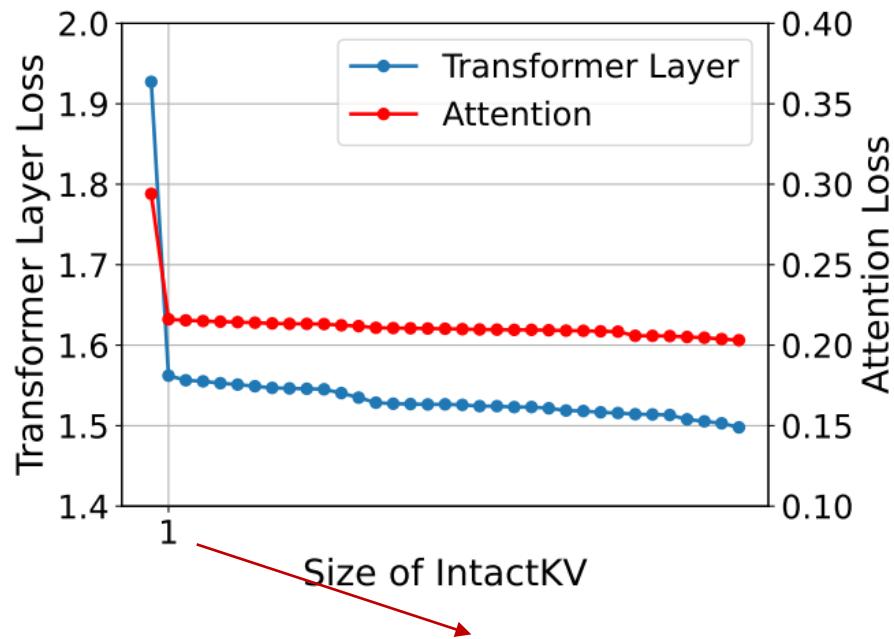


(d) Attention map of  
LLaMA-2-7B Layer 24

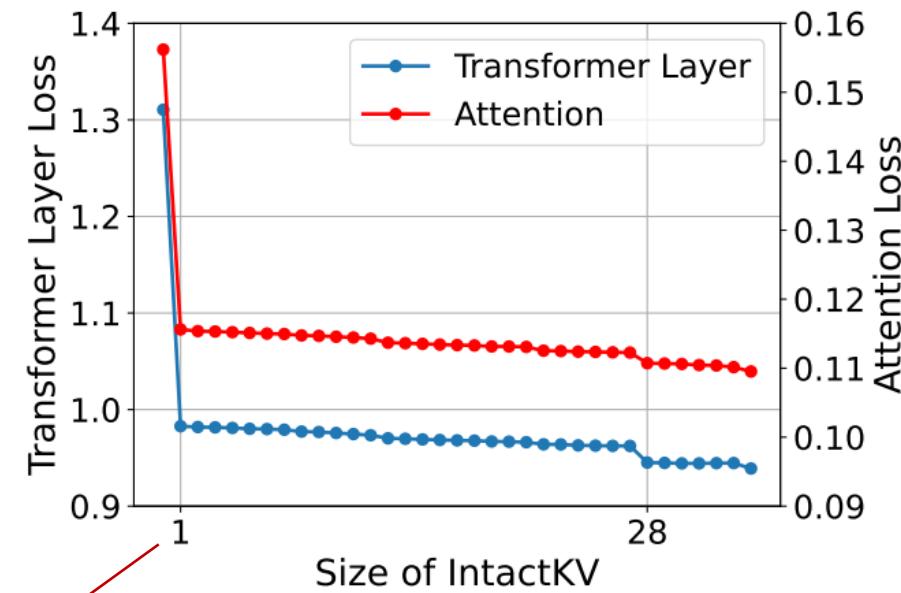
1. Ruikang Liu, Haoli Bai†, Haokun Lin, Yuening Li, Han Gao, Zhengzhuo Xu, Lu Hou, Jun Yao, Chun Yuan. IntactKV: Improving Large Language Model Quantization by Keeping Pivot Tokens Intact. Findings of ACL, 2024.

# 量化挑战：Type II 特大离群值

- 注意力槽导致特大异常值 ( $\sim 1e3$ )
  - Attention output & FFN output
  - 对量化有何影响？

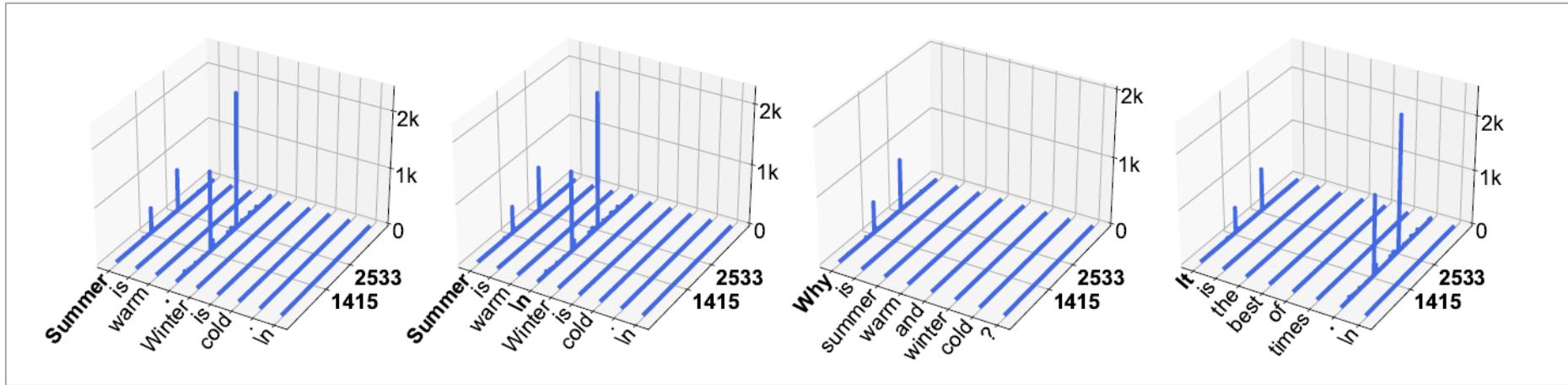


跳过BOS token，量化误差显著降低

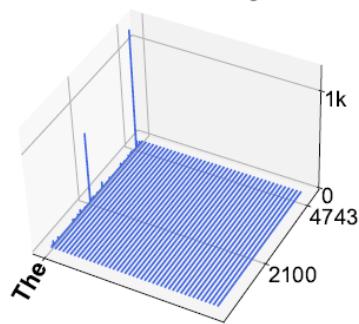


1. RuiKang Liu, Haoli Bai†, et.al. IntactKV: Improving Large Language Model Quantization by Keeping Pivot Tokens Intact. Findings of ACL, 2024.

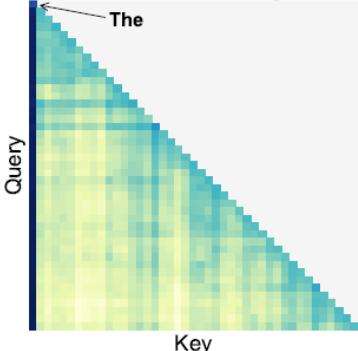
# 同期工作: Massive Activations



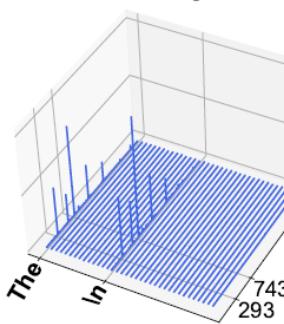
LLaMA2-13B, Layer 4



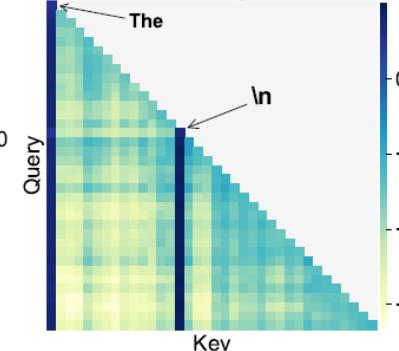
LLaMA2-13B, Layer 5



Phi-2, Layer 8



Phi-2, Layer 9



极大激活值对应注意力槽token  
(attention sink)

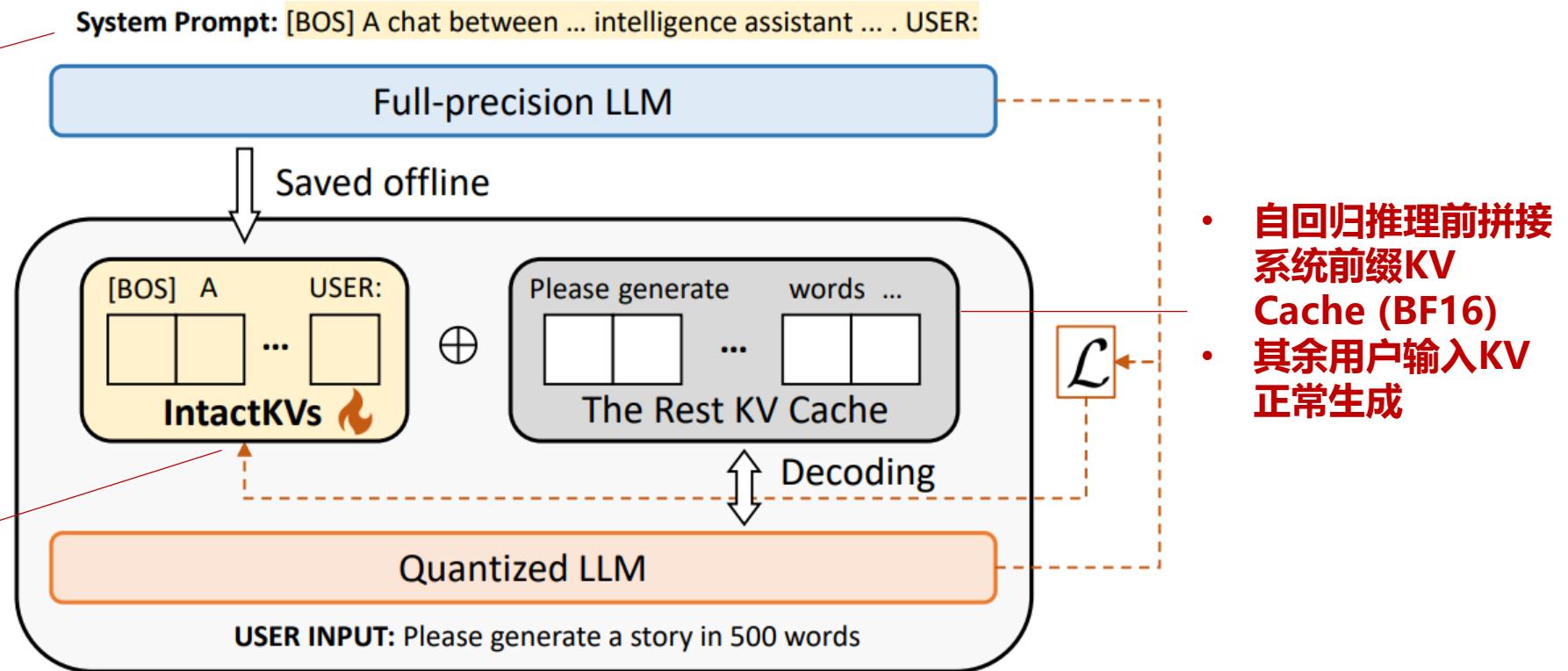
(a) LLaMA2-13B

(b) Phi-2

1. Mingjie Sun, et al. Massive Activations in Large Language Models. COLM 2024

# 隔离Type II离群值： IntactKV关键词元无损量化

- 离线缓存整个系统前缀的KV Cache (BF16)
- 大概率包含极大值 token
- 离线存储的KV Cache作为共享前缀参数，可以额外训练，弥补后续量化损失



1. Ruikang Liu, Haoli Bai†, et.al. IntactKV: Improving Large Language Model Quantization by Keeping Pivot Tokens Intact. Findings of ACL, 2024.

# IntactKV 实验结果

MT-bench GPT4 Evaluation

Method	Vicuna-v1.5-7B	Vicuna-v1.5-13B
FP16	5.31	5.52
RTN	4.34	5.13
+INTACTKV <sub>[P]</sub>	4.72	5.27
+INTACTKV <sub>[P]</sub> +Cal	<b>4.73</b>	<b>5.30</b>
OmniQuant	4.78	5.05
+INTACTKV <sub>[P]</sub>	<b>4.94</b>	5.10
+INTACTKV <sub>[P]</sub> +Cal	4.85	<b>5.24</b>
AWQ	4.74	5.17
+INTACTKV <sub>[P]</sub>	4.68	5.34
+INTACTKV <sub>[P]</sub> +Cal	<b>4.84</b>	<b>5.44</b>

Zero-shot MMLU

Method	MMLU (0 shot)				
	Hums	STEM	Social	Others	Avg
FP16	47.89%	39.96%	58.86%	57.34%	50.77%
RTN	42.06%	32.87%	47.61%	49.51%	43.02%
+INTACTKV <sub>[B]</sub>	42.49%	35.35%	50.37%	52.44%	44.98%
GPTQ	45.06%	35.88%	52.23%	51.26%	46.09%
+INTACTKV <sub>[B]</sub>	44.72%	35.42%	52.94%	52.07%	46.22%
OmniQuant	43.51%	36.85%	52.16%	53.05%	46.18%
+INTACTKV <sub>[B]</sub>	44.19%	36.61%	53.33%	53.52%	46.72%
AWQ	45.14%	36.18%	52.55%	53.79%	46.84%
+INTACTKV <sub>[B]</sub>	45.91%	36.65%	53.75%	54.60%	<b>47.64%</b>

- 与各类量化方案解耦，对精度有直接提升
- 与公共前缀 (common-prefix) 方案共同组合，还可降低首token时延 (TTFT)

## **Part 1：大模型量化**

- ✓ 大模型中的两种离群值：一种沿固定通道，一种绑定注意力槽
- ✓ 通过线性变换、隔离极大离群值可以有效缓解离群值的影响

## **Part 2：大模型剪枝**

## **Part 3：未来展望**

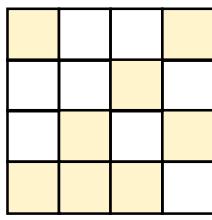
## Part 1：大模型量化

- ✓ 大模型中的两种离群值：一种沿固定通道，一种绑定注意力槽
- ✓ 通过线性变换、隔离极大离群值可以有效缓解离群值的影响

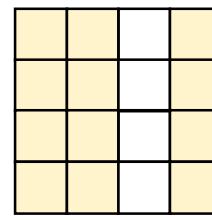
## Part 2：大模型剪枝

## Part 3：未来展望

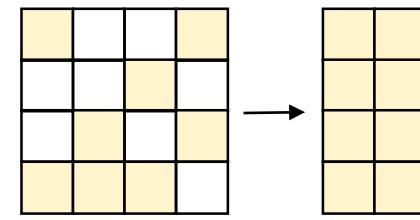
# 大模型剪枝



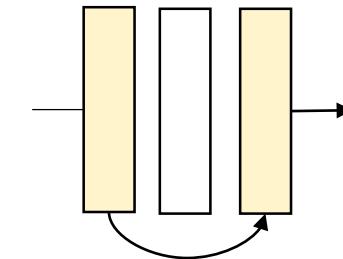
非结构化剪枝



结构化剪枝

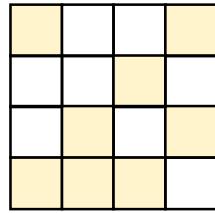


半结构化剪枝 (N:M稀疏)

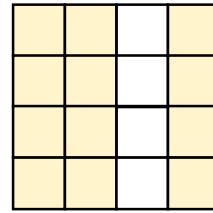


层剪枝

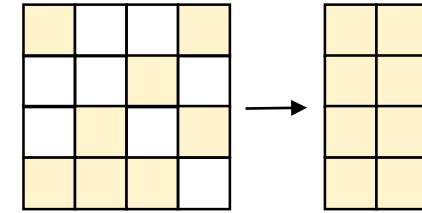
# 大模型剪枝



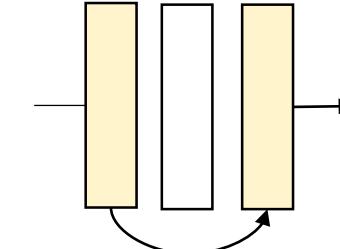
非结构化剪枝



结构化剪枝



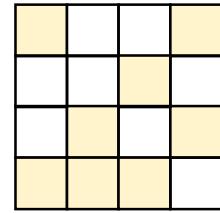
半结构化剪枝 (N:M稀疏)



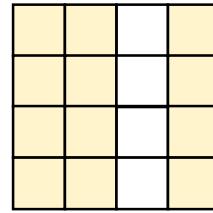
层剪枝

	访存		计算量	
	模型尺寸	吞吐	预填充	解码
非结构化剪枝	↓	↑	-	-
结构化剪枝	↓	↑	↓	↓
半结构化剪枝	↓	↑	↓	↓
层剪枝	↓	↑↑	↓↓	↓↓

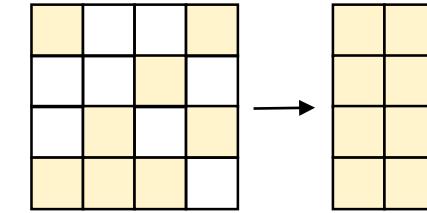
# 大模型剪枝



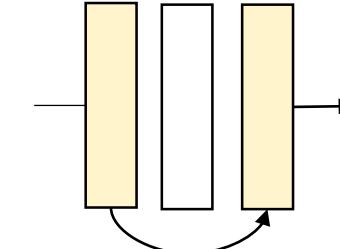
非结构化剪枝



结构化剪枝



半结构化剪枝 (N:M稀疏)

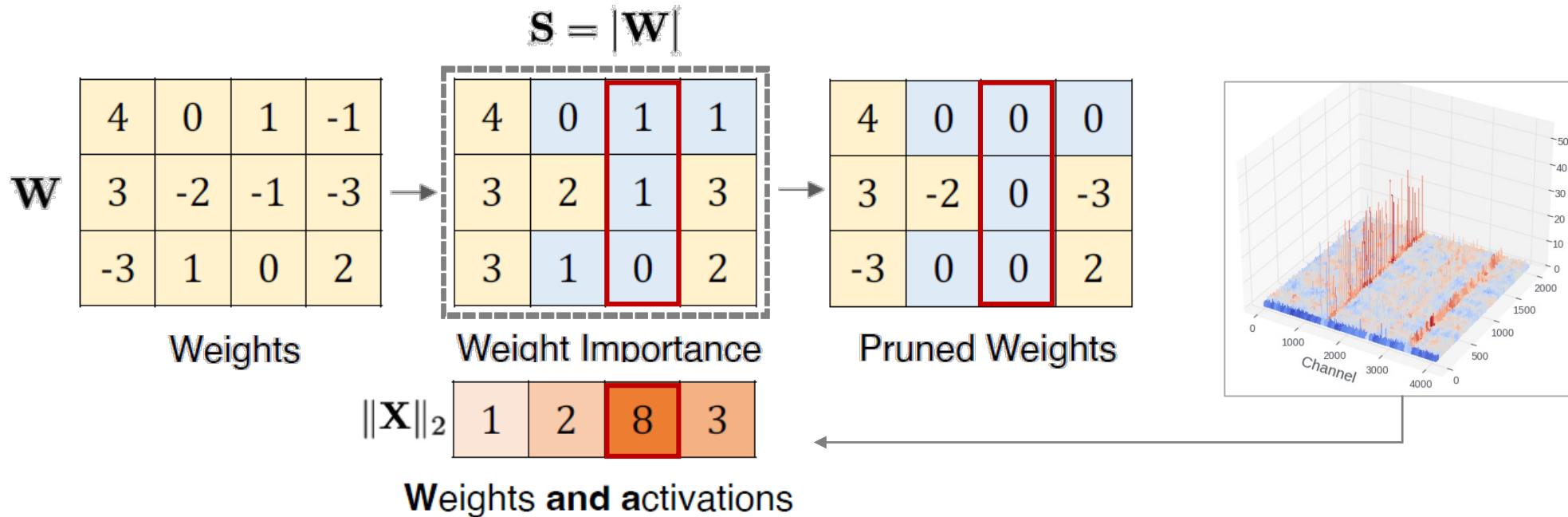


层剪枝

	访存		计算量	
	模型尺寸	吞吐	预填充	解码
非结构化剪枝	↓	↑	-	-
结构化剪枝	↓	↑	↓	↓
半结构化剪枝	↓	↑	↓	↓
层剪枝	↓	↑↑	↓↓	↓↓

# 模型剪枝：忽略激活值或导致误判

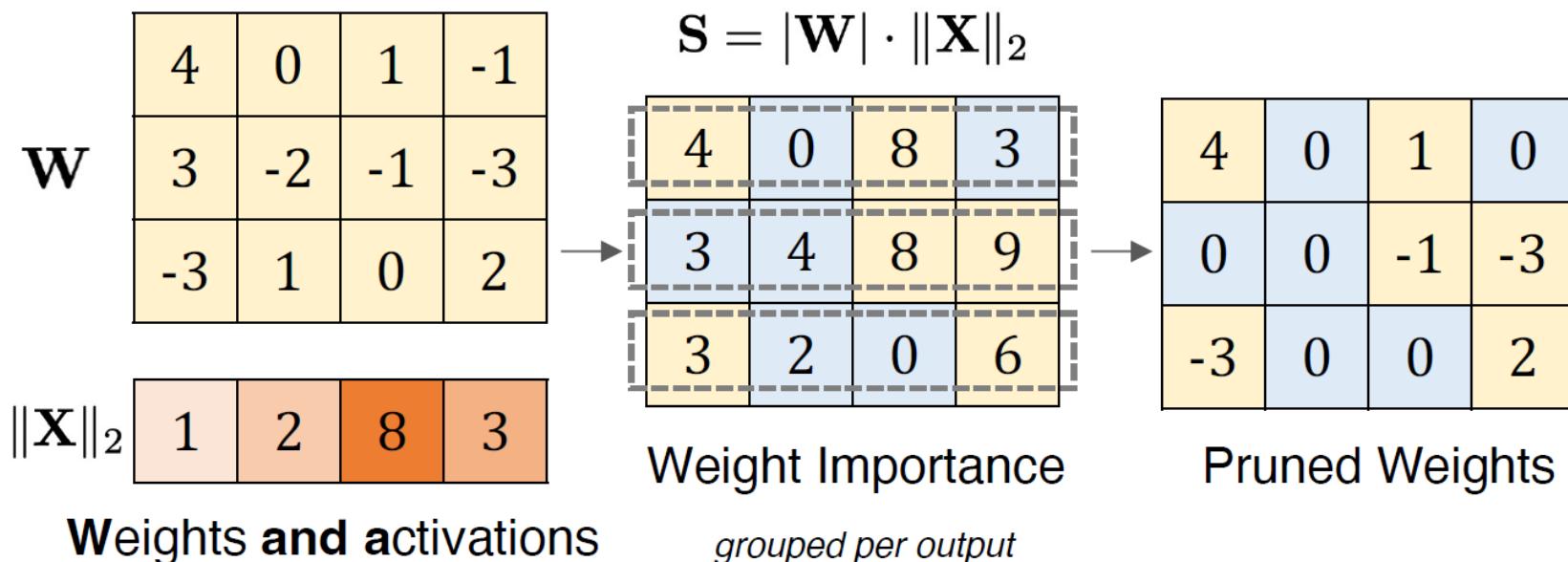
- 如何判断参数重要性？参数幅值大小
  - 模型剪枝只对于参数进行裁剪，与激活值无关
  - 离群值主要集中在激活值



如何考虑激活值对参数剪枝的重要性判断？

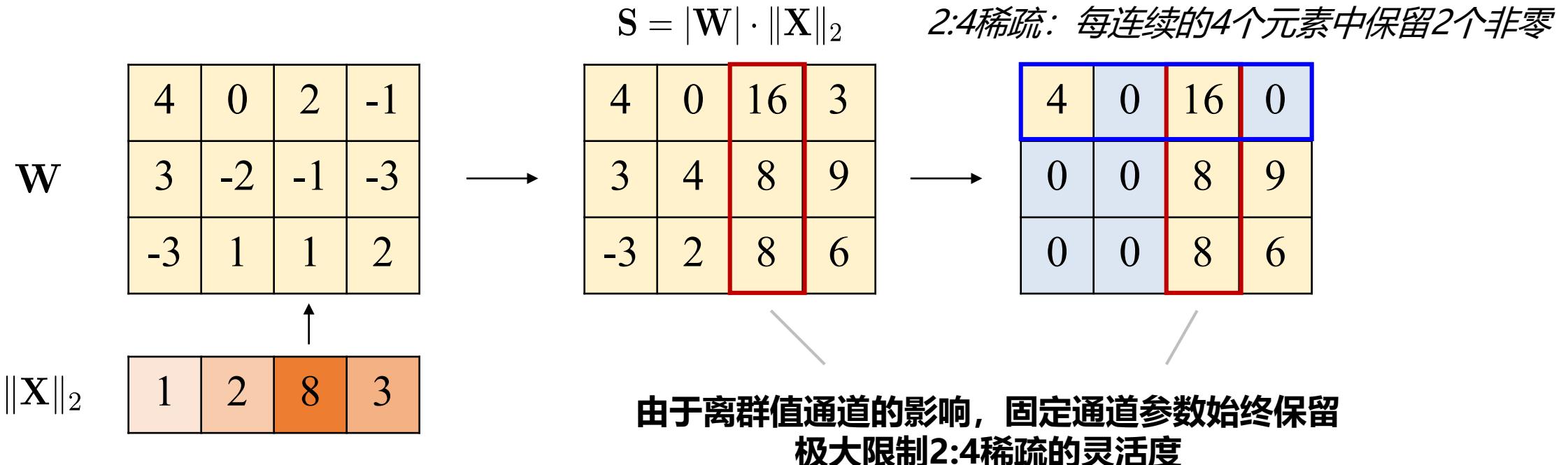
# 模型剪枝：忽略激活值或导致误判

- 如何把激活值的重要性迁移到模型参数上?
  - 把激活值上统计的重要性缩放进模型参数里 [1, 2]



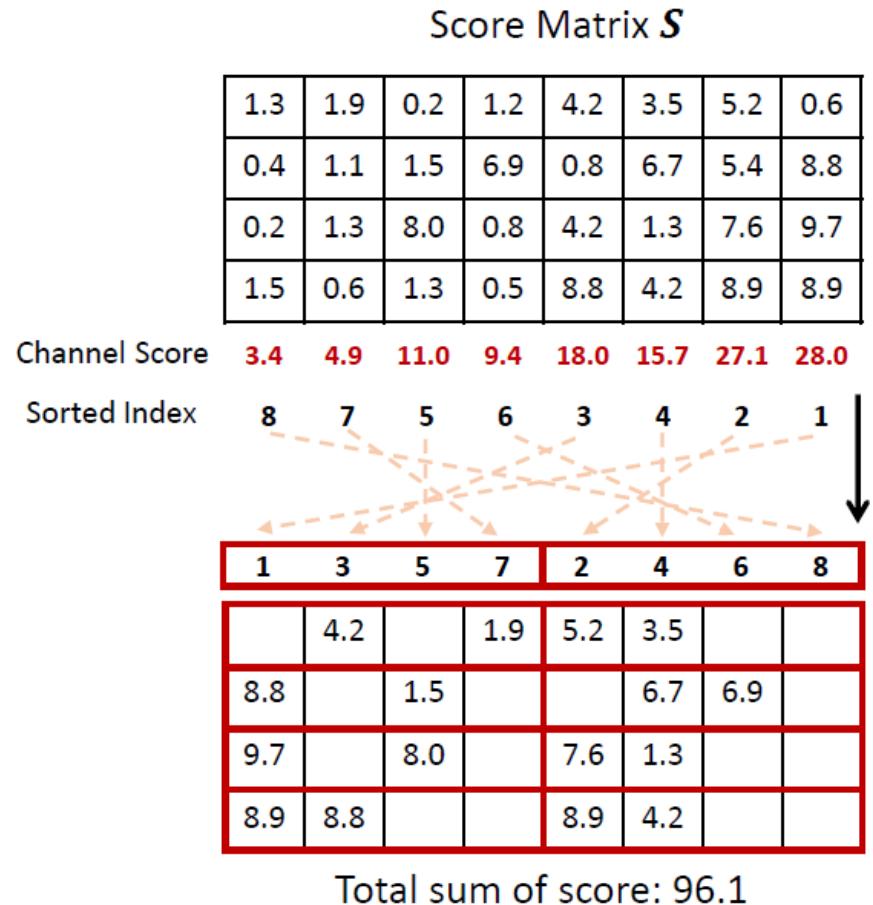
1. Mingjie Sun, et. al., A Simple and Effective Pruning Approach for Large Language Models, arXiv preprint arXiv:2306.11695
2. Yingtao Zhang, Haoli Bai, et. al., Plug-and-Play: An Efficient Post-training Pruning Method for Large Language Models. ICLR 2024.

# 离群值倾向固化N:M稀疏选择



如何在考虑激活离群值的前提下，探索更多N:M稀疏方案？

# 通道重排序寻找更优N:M稀疏解



## 通道重排序

1. 计算参数权重要性
  2. 统计各通道 (channel) 得分
  3. 把通道按每组M个元素，分成K组
  4. 将通道得分从高到低依次分别放入K组
  5. 对于每组内部，进行N:M稀疏剪枝
- 通道重排序可以保证计算等价性
  - 重排序+calibration: 10分钟内完成

1. Yingtao Zhang, Haoli Bai, et. al., Plug-and-Play: An Efficient Post-training Pruning Method for Large Language Models. ICLR 2024.

# 主要结果

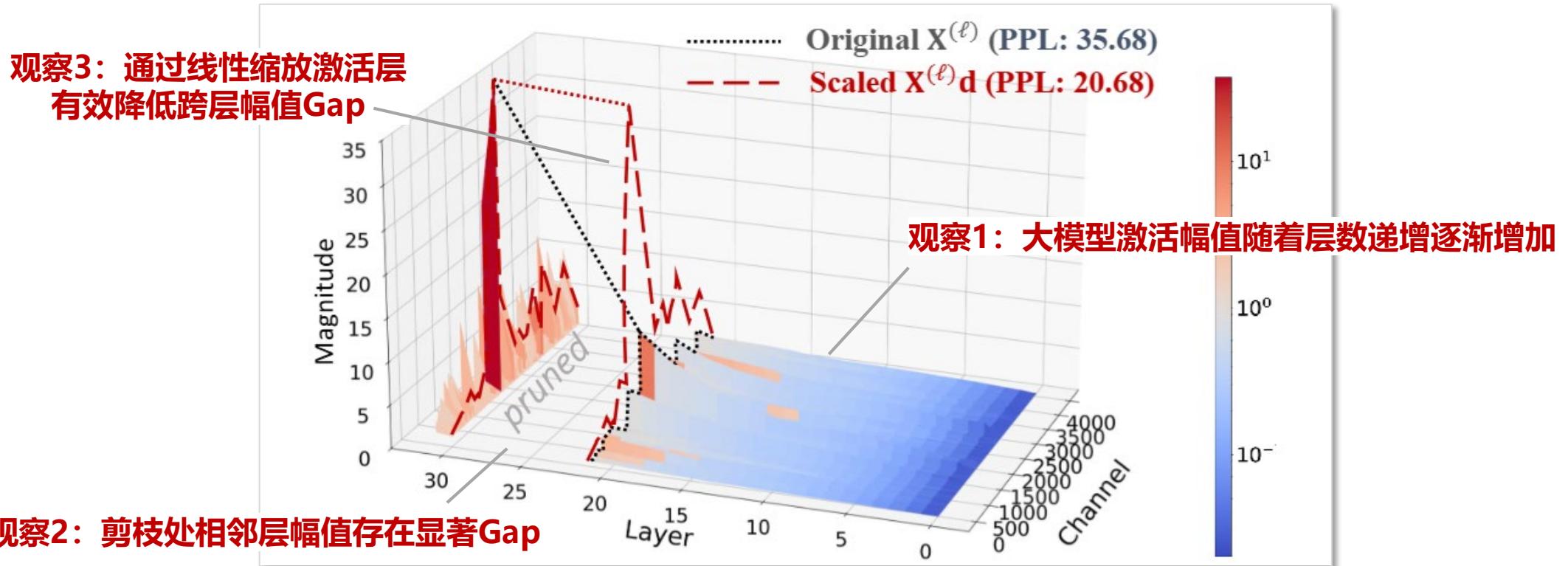
Method	Hellaswag	BoolQ	ARC-C	MNLI	RTE	AVG
Dense	64.77	83.70	54.44	45.81	67.87	63.32
Wanda (2:4)	57.35	81.44	46.01	37.69	68.59*	58.22
Wanda (2:4+CP)	59.37	84.50*	48.55	43.09	66.43	60.39
Wanda (4:8+CP)	<b>60.86</b>	82.73	49.94	40.15	67.87	60.51
RIA (2:4)	57.13	82.78	46.76	37.39	69.31*	58.68
RIA (2:4+CP)	58.48	<b>85.14*</b>	49.15	<b>49.08*</b>	68.95*	62.16
RIA (4:8+CP)	60.44	83.58	<b>50.43</b>	48.69*	<b>70.04*</b>	<b>62.64</b>

- 通道重排序可以与当前SOTA方案正交组合
- 重排序方案显著提升精度效果，LLaMA-2-7B

1. Mingjie Sun, et. al., A Simple and Effective Pruning Approach for Large Language Models, arXiv preprint arXiv:2306.11695
2. Yingtao Zhang, Haoli Bai, et. al., Plug-and-Play: An Efficient Post-training Pruning Method for Large Language Models. ICLR 2024.

# 大语言模型层剪枝

- 大模型不同层间幅值存在显著差异

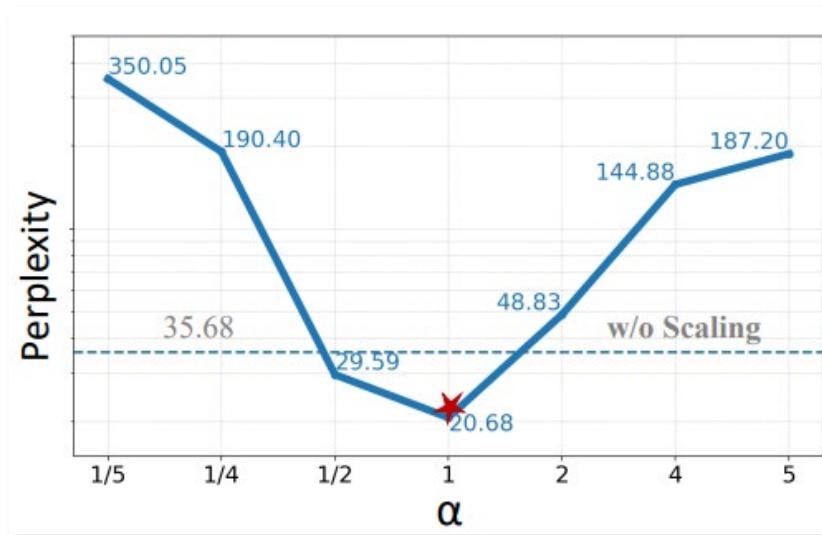


1. Xinrui Chen, Haoli Bai\*, et. al. A Simple Linear Patch Revives Layer-Pruned Large Language Models. NeurIPS 2025.

# 两种大模型的层间幅值差异

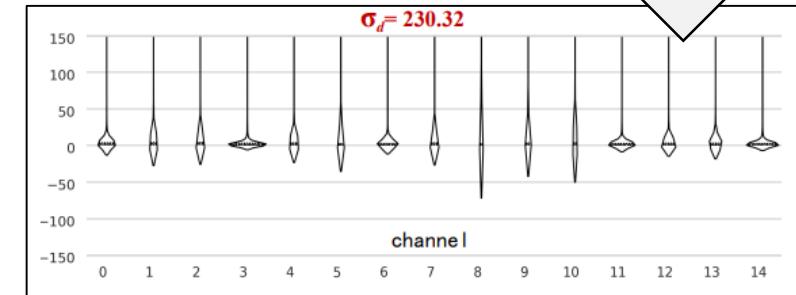
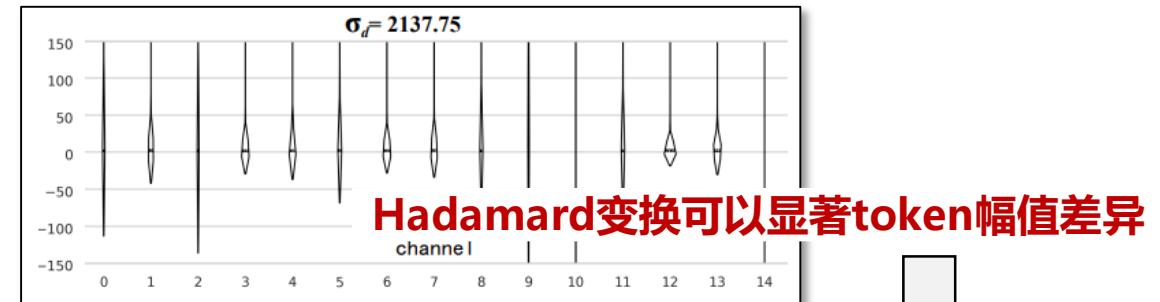
① 相邻层间通道幅值差异  $X \in \mathbb{R}^{n \times k}$

$$d_k(\ell^*, \ell^* + n) = \|\mathbf{X}_{:,k}^{(\ell^*+n)}\|_1 / \|\mathbf{X}_{:,k}^{(\ell^*)}\|_1.$$
$$\mathbf{X}^{(\ell^*+n)} = \alpha \mathbf{X}^{(\ell^*)} \cdot \mathbf{d} + f(\alpha \mathbf{X}^{(\ell^*)} \cdot \mathbf{d}, \theta^{(\ell^*+n)}).$$

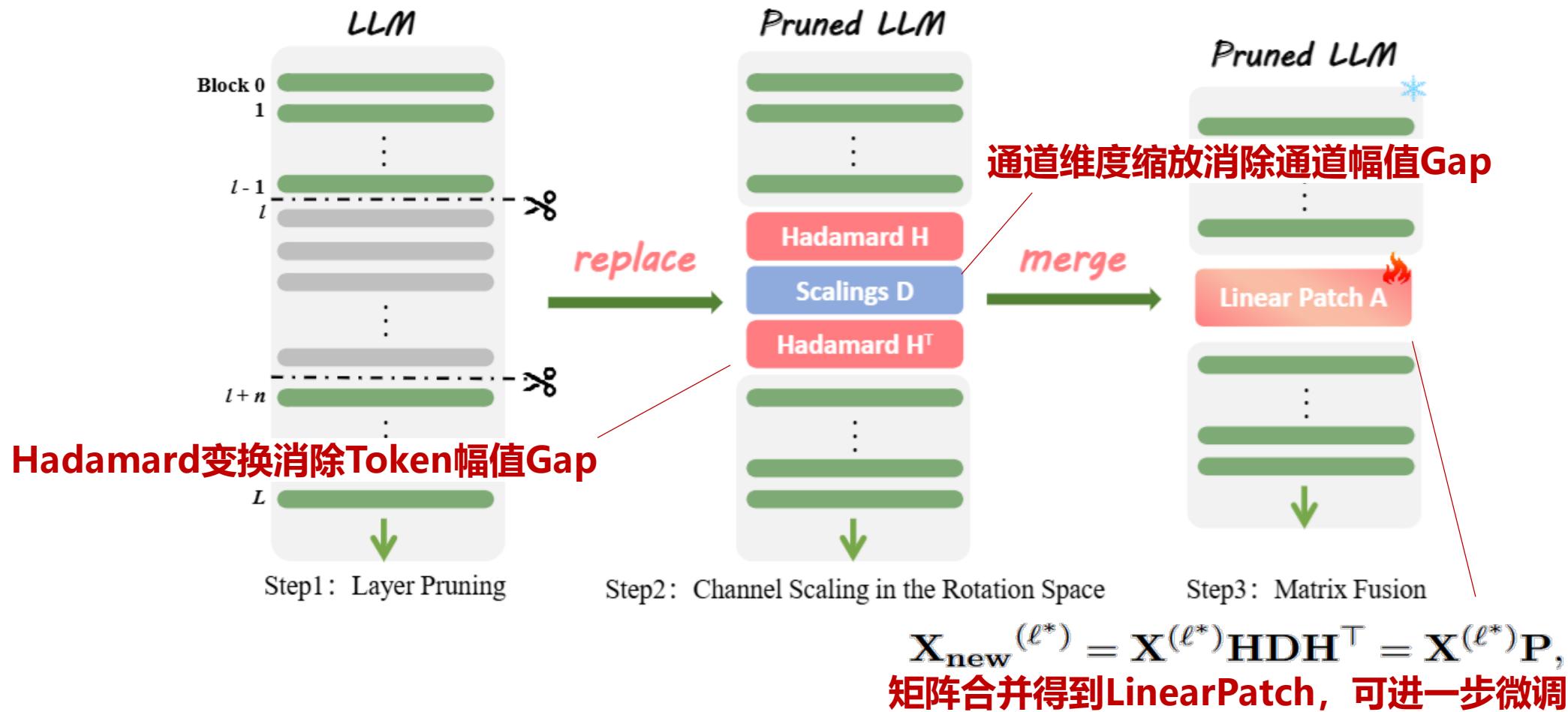


② 相邻层间token幅值差异  $X \in \mathbb{R}^{n \times k}$

$$\sigma_d(\ell^*, \ell^* + n) = \frac{1}{BC} \sum_{i=1}^B \sum_{k=1}^C \sigma \left( \frac{|\mathbf{X}_{i,k}^{(\ell^*+n)}|}{|\mathbf{X}_{i,k}^{(\ell^*)}|} \right)$$



# 在层剪枝处打上一个线性补丁



1. Xinrui Chen, Haoli Bai\*, et. al. A Simple Linear Patch Revives Layer-Pruned Large Language Models. NeurIPS 2025.

# 主要结果

Model	Method	WIKI-2	C4	PTB	PPL avg.
LLaMA-2-7B	Dense	5.47	6.97	22.51	11.65
	SLEB	9.14	11.21	38.45	19.60
	+LINEARPATCH	8.77	10.66	38.30	<b>19.24</b>
	Taylor+	18.45	20.99	62.18	33.87
	+LINEARPATCH	13.84	15.28	48.26	<b>25.79</b>
	ShortGPT	18.45	20.99	62.18	33.87
	+LINEARPATCH	13.22	14.58	45.97	<b>24.59</b>
	LLM-Streamline (None)	18.45	20.99	62.18	33.87
	+LINEARPATCH	13.22	14.58	45.97	<b>24.59</b>
LLaMA-3-8B	Dense	6.14	8.88	10.59	8.54
	SLEB	13.12	16.76	21.04	16.97
	+LINEARPATCH	11.97	15.74	19.55	<b>15.75</b>
	Taylor+	2287.86	1491.38	4741.90	2840.38
	+LINEARPATCH	208.88	235.63	264.97	<b>236.49</b>
	ShortGPT	57.76	50.13	67.39	58.43
	+LINEARPATCH	25.67	28.38	31.22	<b>28.42</b>
	LLM-Streamline (None)	2287.73	1491.37	4738.81	2839.30
	+LINEARPATCH	69.82	96.68	88.79	<b>85.10</b>

1. <https://github.com/chenxinrui-tsinghua/LinearPatch>

① LinearPatch作为可插拔插件，正交组合现有层剪枝方案

② 在各类方法上带来精度提升

③ 线性补丁的推理成本可忽略

## Part 1：大模型量化

- ✓ 大模型中的两种离群值：一种沿固定通道，一种绑定注意力槽
- ✓ 通过线性变换、隔离极大离群值可以有效提升量化效果

## Part 2：大模型剪枝

- ✓ N:M剪枝：结合激活幅值、通道重排序有效规避离群值影响
- ✓ 层剪枝：通过一个线性补丁有效缝合不同层间幅值，提升层剪枝精度

## Part 3：未来展望

## Part 1：大模型量化

- ✓ 大模型中的两种离群值：一种沿固定通道，一种绑定注意力槽
- ✓ 通过线性变换、隔离极大离群值可以有效提升量化效果

## Part 2：大模型剪枝

- ✓ N:M剪枝：结合激活幅值、通道重排序有效规避离群值影响
- ✓ 层剪枝：通过一个线性补丁有效缝合不同层间幅值，提升层剪枝精度

## Part 3：未来展望

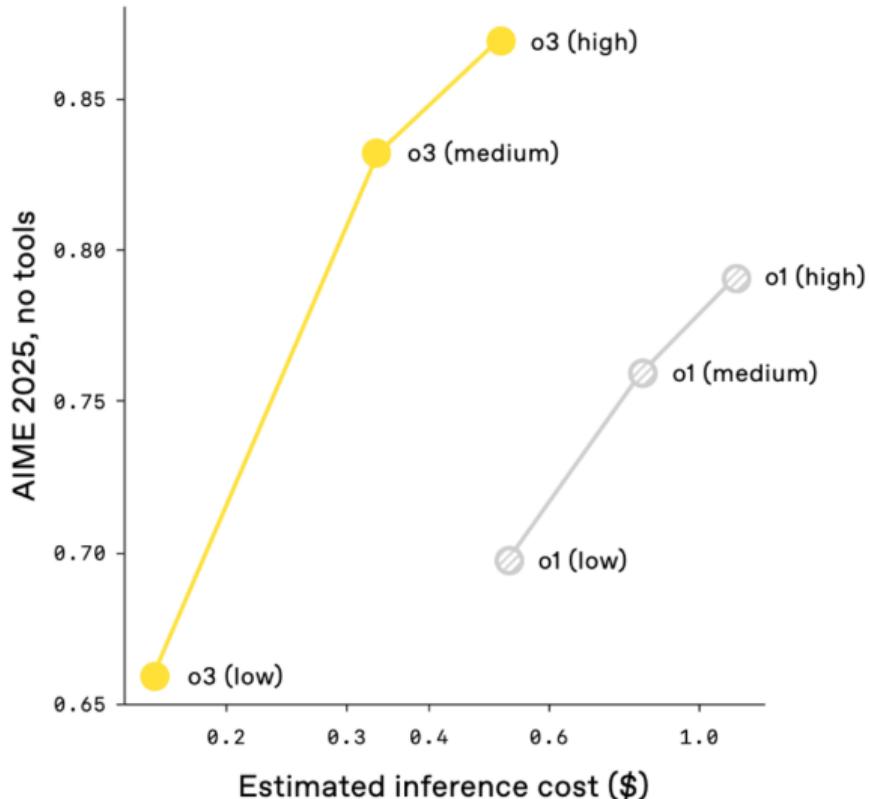


接下来，量化稀疏还有什么新机遇？

# 机遇1：新模型，新任务



Thought for 12m 3s



**Problem:** Let  $A = (1, -11, 2)$ ,  $B = (3, -4, 1)$ , and  $C = (-2, 1, -1)$ . Compute  $\angle ABC$ , in degrees.

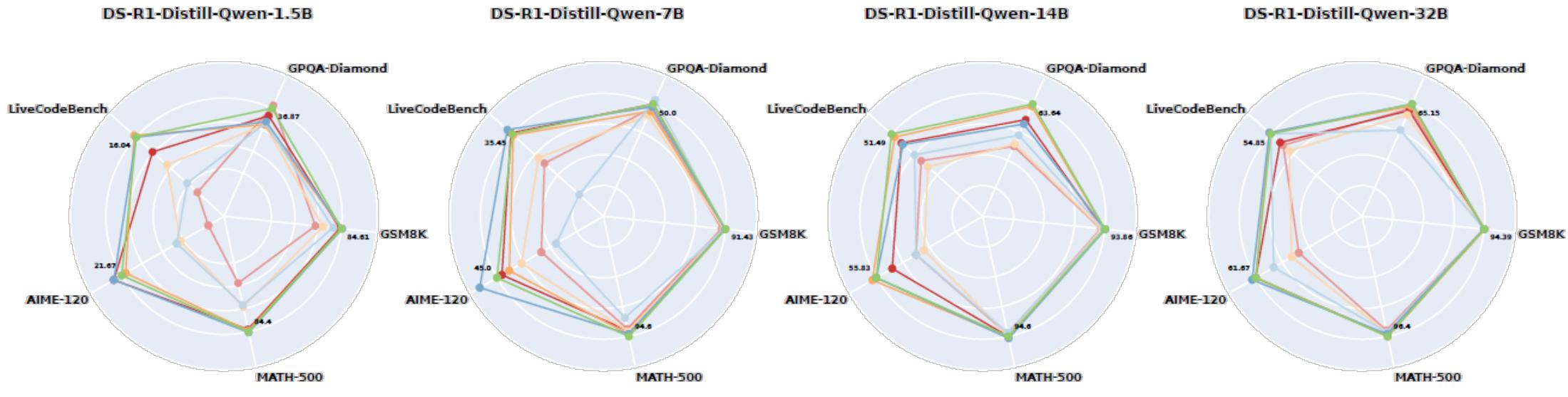
**Reasoning:** Okay, so I need to find the angle at point B in triangle ABC. The points given are  $A = (1, -11, 2)$ ,  $B = (3, -4, 1)$ , and  $C = (-2, 1, -1)$ . Wait, hold on, the coordinates for point C are written as  $(-, 1, -1)$ . That doesn't look right. Is that a typo or a mistake? Maybe it should be something like  $(-1, 1, -1)$  or  $(-2, 1, -1)$  with a specific number? Hmm, maybe it's a typo and supposed to be  $(-1, 1, -1)$ . I'll assume that for now, so point C is  $(-1, 1, -1)$ . I think that makes sense. So, points are  $A = (1, -11, 2)$ ,  $B = (3, -4, 1)$ , and  $C = (-1, 1, -1)$ . I need to compute angle ABC, which is the angle at point B between points A, B, and C. So, angle ABC is the angle at vertex B, formed by sides BA and BC. So, to find this angle, I need to find the vectors BA and BC, then compute the angle between these two vectors using the dot product formula ...

**Final answer:**

The measure of angle ABC is 126 degrees.

# 机遇1：新模型，新任务

- 慢思考模型量化精度测评：DeepSeek-R1-Distill-Qwen 1.5B ~ 32B



研究问题：构建更新、更全的各项任务精度测评分析

—— Deep Research, Code Agent

1. Ruiyang Liu\*, Yuxuan Sun\*, Manyi Zhang\*, Haoli Bai<sup>#+</sup>, Xianzhi Yu, Tiezheng Yu, Chun Yuan, Lu Hou<sup>#</sup>. Quantization Hurts Reasoning? An Empirical Study on Quantized Reasoning Models. COLM 2025.

# 机遇2：模型压缩兼顾多维能力平衡

ChatGPT请求类型分布



研究问题

① 校准集选取与调优

② 压缩感知微调数据配比

③ Bad-case快速修复与上线

1. <https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-langs-like-chatgpt/>

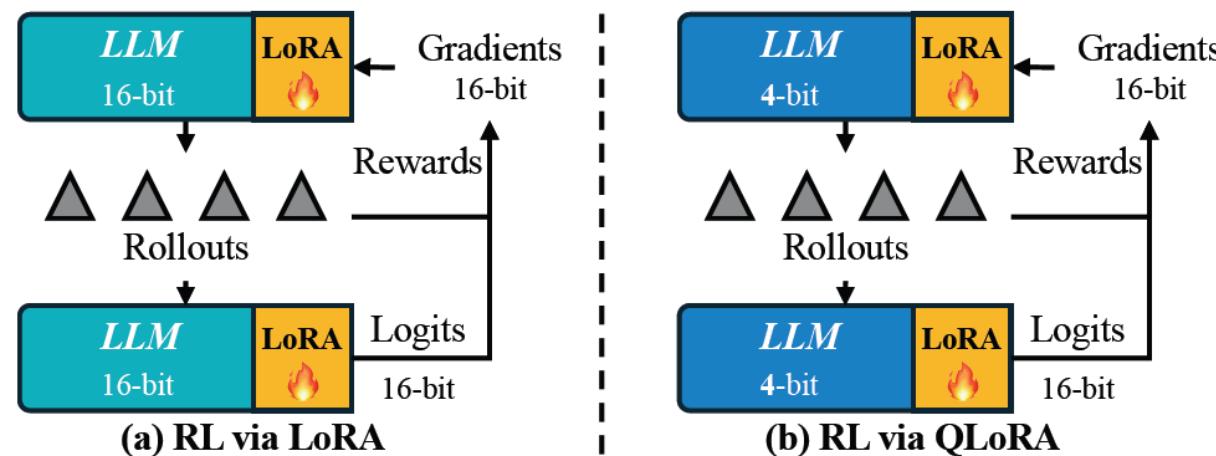
# 机遇3：模型量化与强化学习

- **视角1：提升推理模型量化精度**

- 对于量化后的通用模型精度提升，强化学习是否是有效的微调方式？
- 相对稳健的训练范式：1) 后量化; 2) 知识蒸馏; 3) 强化学习

- **视角2：提升强化学习训练效率**

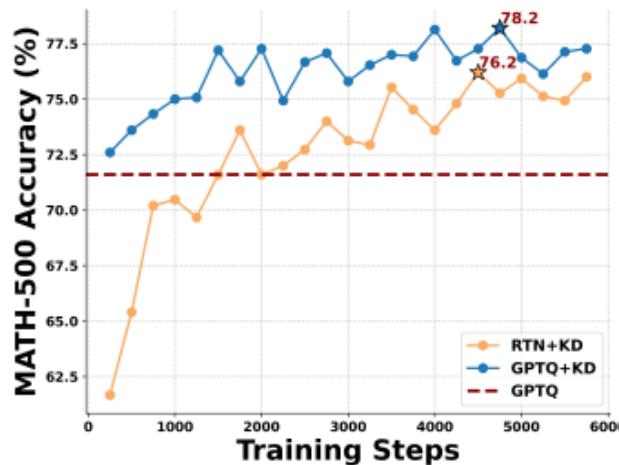
- 通过模型量化，降低推理rollout在训练系统中的占比开销



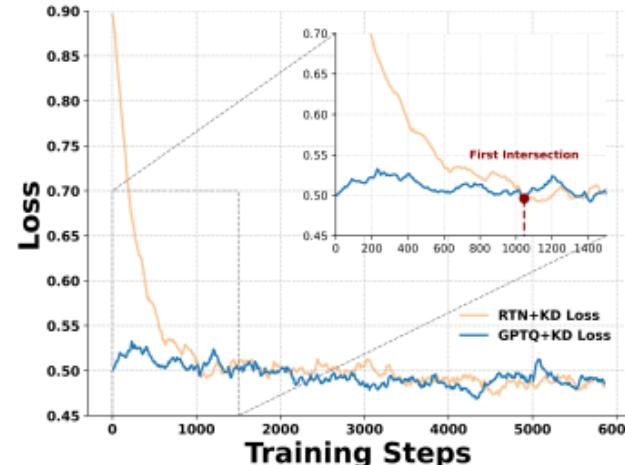
1. Huang W, Ge Y, Yang S, et al. QeRL: Beyond Efficiency--Quantization-enhanced Reinforcement Learning for LLMs[J]. arXiv preprint 2510.11696.
2. Anonymous authors, Benchmarking and Advancing Quantization-Aware Training for Reasoning Models. Under review at ICLR 2026.

# 机遇3：模型量化与强化学习

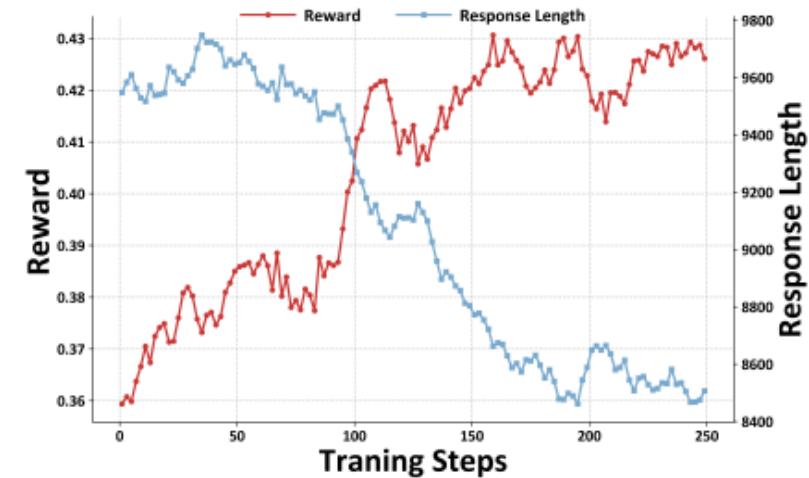
- 什么样的目标函数亲和模型训练范式 (SFT、RL) ?
- 如何更好的初始化QAT模型?
- 低bit强化学习下的训练曲线如何?



(a) Test Accuracy



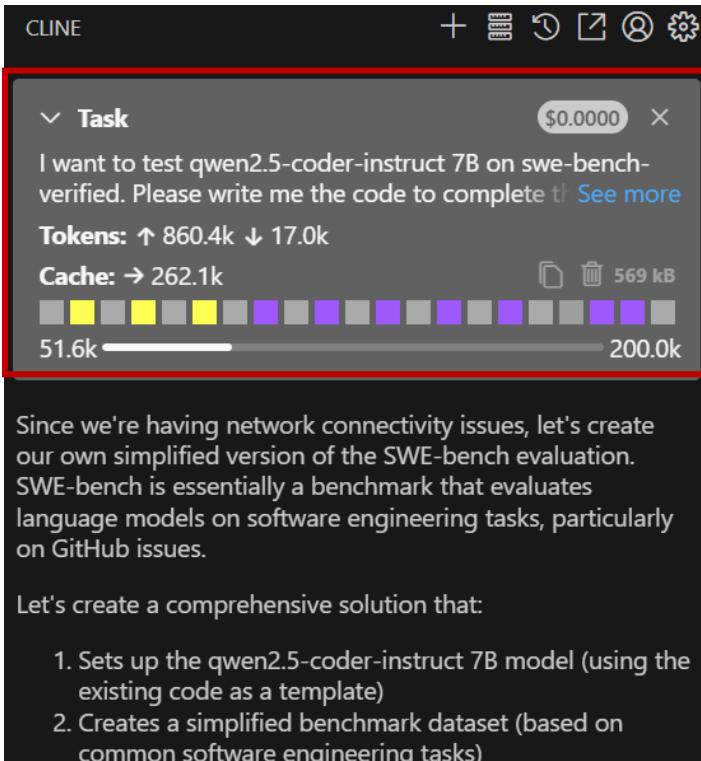
(b) Loss Value



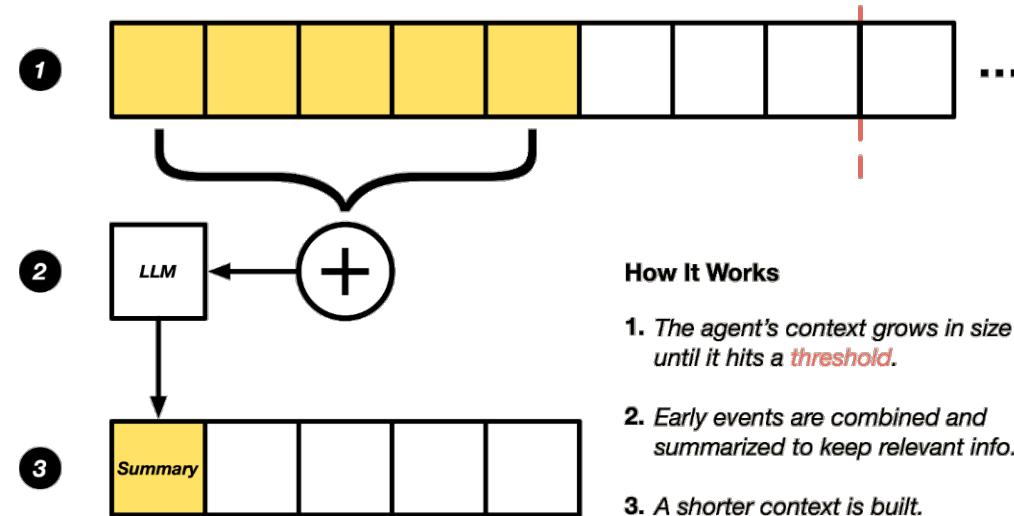
1. Anonymous authors, Benchmarking and Advancing Quantization-Aware Training for Reasoning Models. Under review at ICLR 2026.

# 机遇4：稀疏长上下文与Agentic LLM

- **Agentic AI上下文折叠：**适用于代码智能体、深度搜索智能体等



Qwen3-Coder 480B: 单请求消耗800K+ tokens

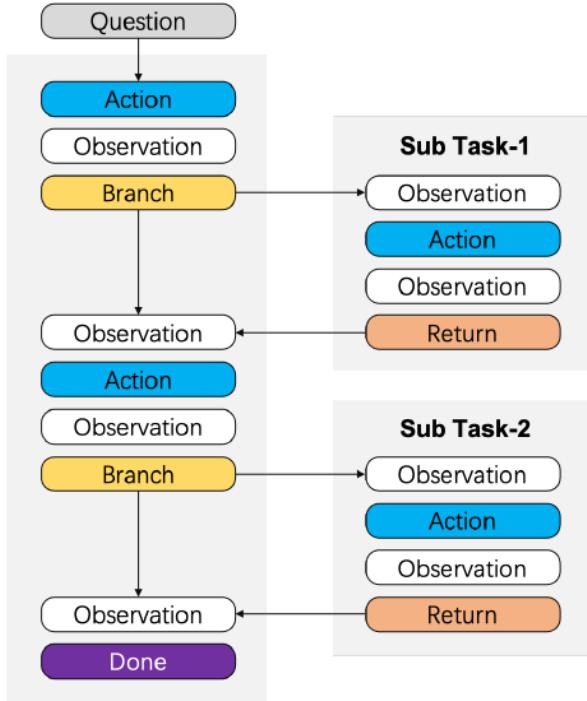


大模型对超长context做总结，压缩长上下文

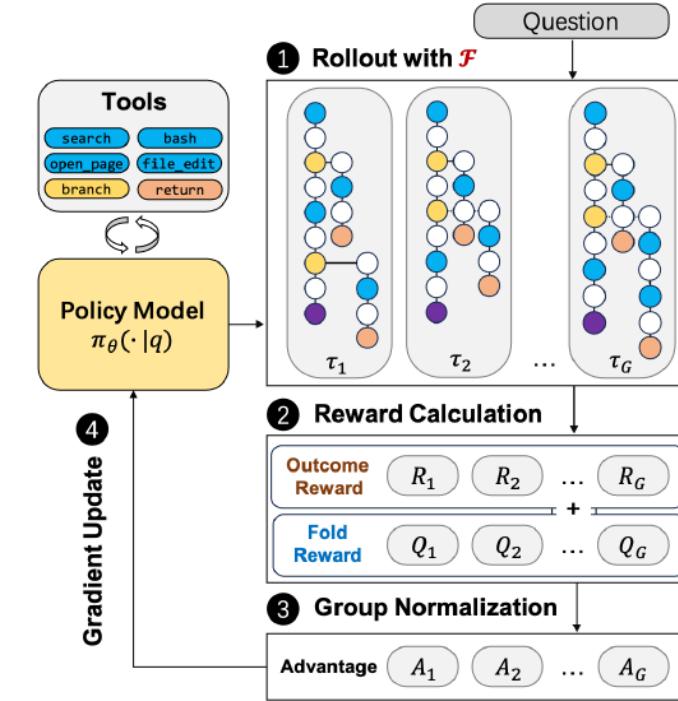
1. <https://openhands.dev/blog/openhands-context-condensation-for-more-efficient-ai-agents>

# 机遇4：稀疏长上下文与Agentic LLM

- **Agentic AI上下文折叠：**适用于代码智能体、深度搜索智能体等



(a) Context Folding



1. 学习创建，解决，总结子任务
2. 降低子任务输出的上下文窗口
3. 通过SFT/RL训练使能模型学会对上下文窗口进行折叠
4. 降低上下文窗口10+倍，在SWE、BrowseComp等榜单精度接近无损

1. <https://openhands.dev/blog/openhands-context-condensation-for-more-efficient-ai-agents>

## Part 1：大模型量化

- ✓ 大模型中的两种离群值：一种沿固定通道，一种绑定注意力槽
- ✓ 通过线性变换、隔离极大离群值可以有效提升量化效果

## Part 2：大模型剪枝

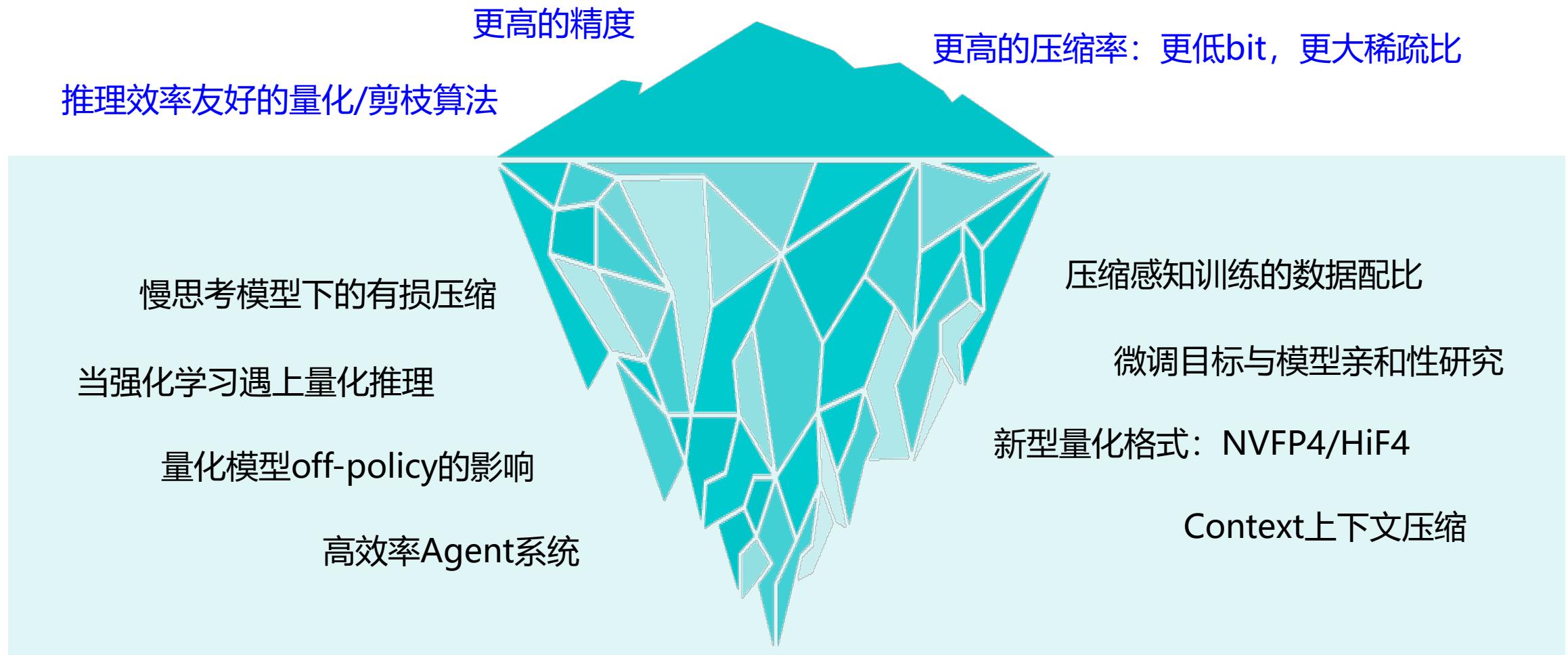
- ✓ N:M剪枝：结合激活幅值、通道重排序有效规避离群值影响
- ✓ 层剪枝：通过一个线性补丁有效缝合不同层间幅值，提升层剪枝精度

## Part 3：未来展望

- ✓ 机遇1：新模型，新任务
- ✓ 机遇2：平衡压缩模型的各项能力
- ✓ 机遇3：模型量化与强化学习
- ✓ 机遇4：稀疏长上下文与Agentic LLM

...

# 大模型量化剪枝：当前研究是冰山一角



# Q&A