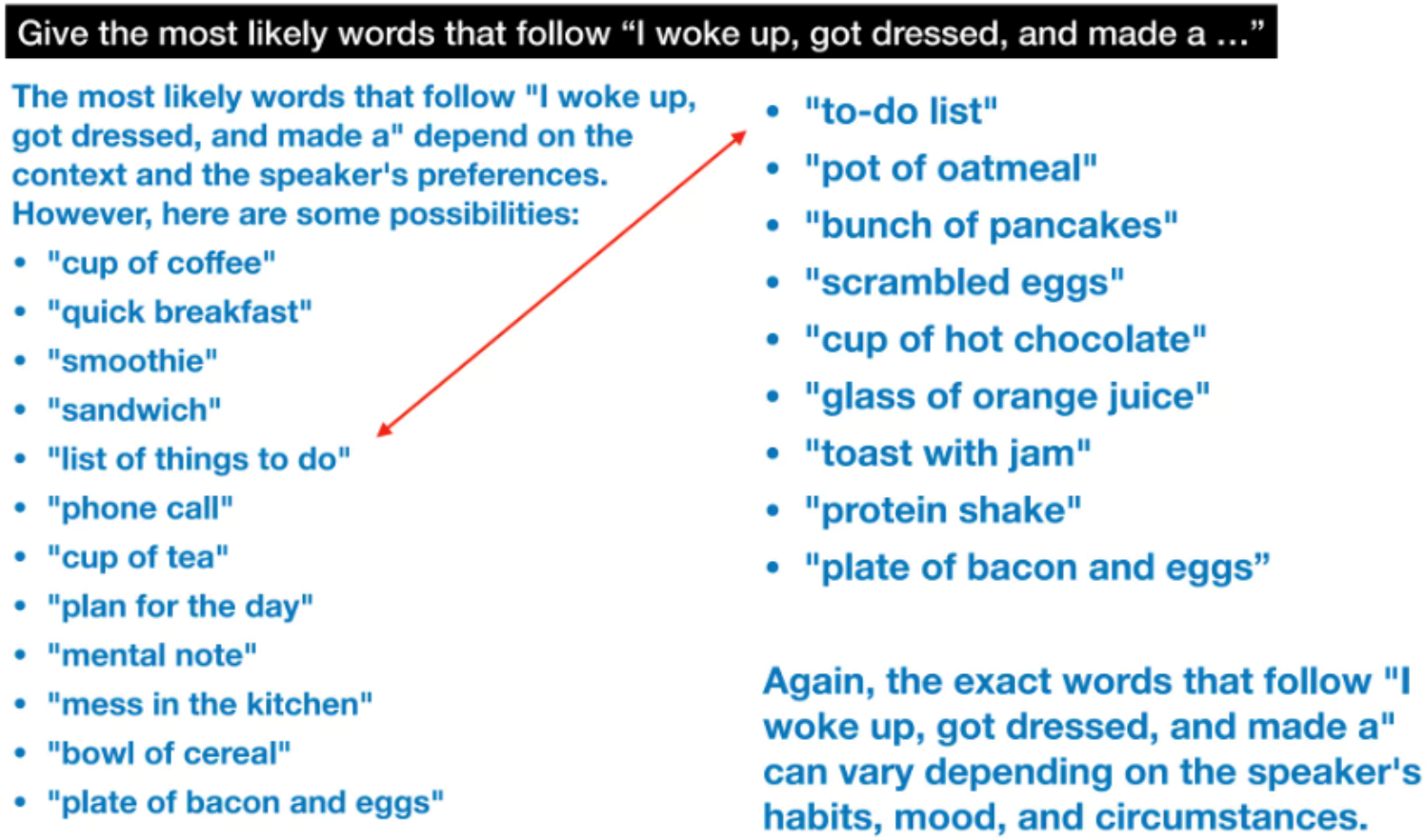
Third Place in AICAS 2024 Challenge: From LLMs Compression to Deployment on ARM CPUs

Ziqiang Chen, Renhuang Huang, Xuedian Hu

Zhejiang Lab & Ant Group

2024 IEEE International Conference on Artificial Intelligence Circuits and Systems

Introduction to LLMs

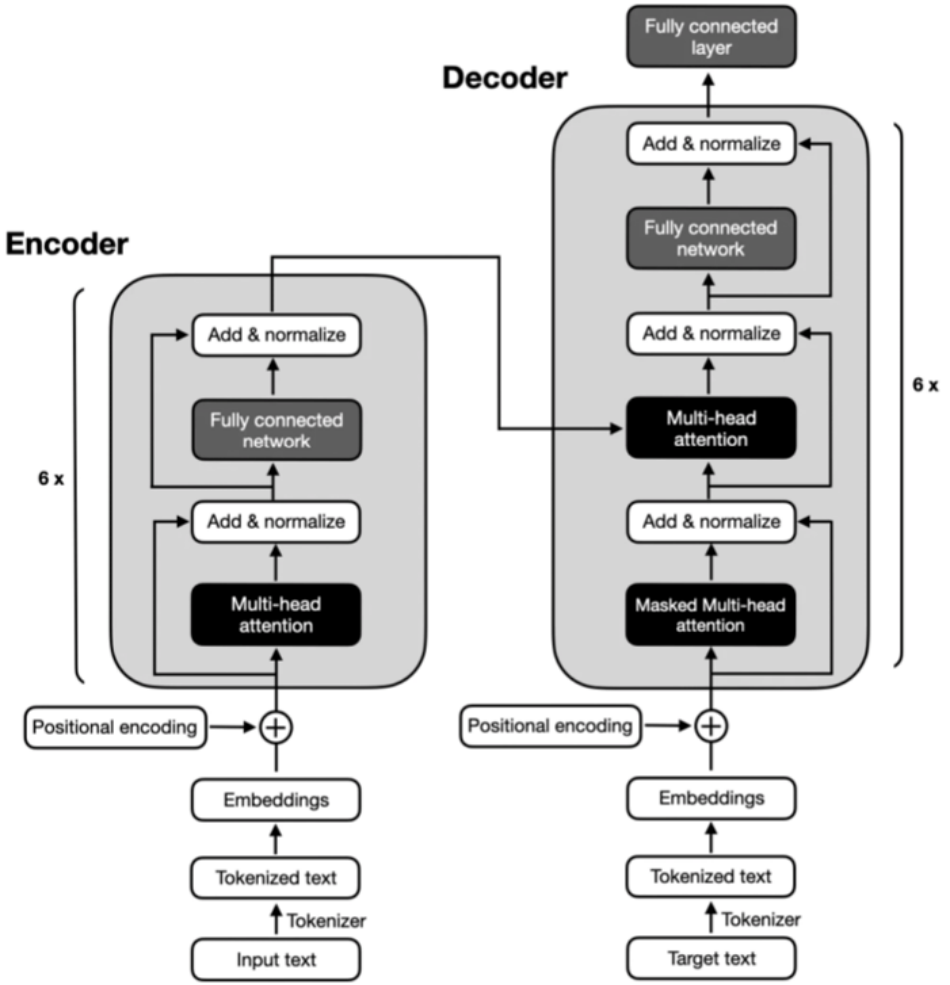


What are large language models(LLMs) ?

Outline

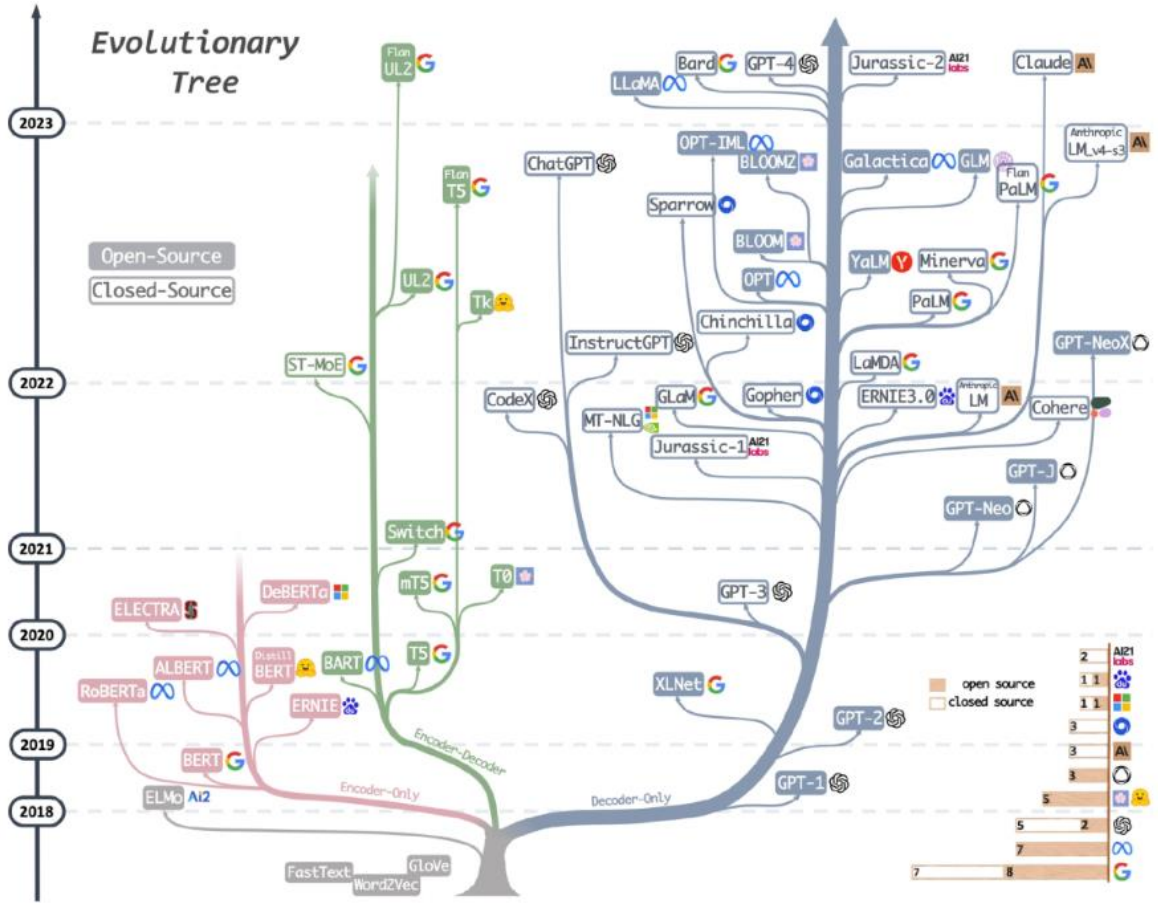
- Introduction to LLMs
- Model Compression
- Low-bit Quantization
- Kernel Optimization
- Results
- Conclusion And Future Work

Introduction to LLMs

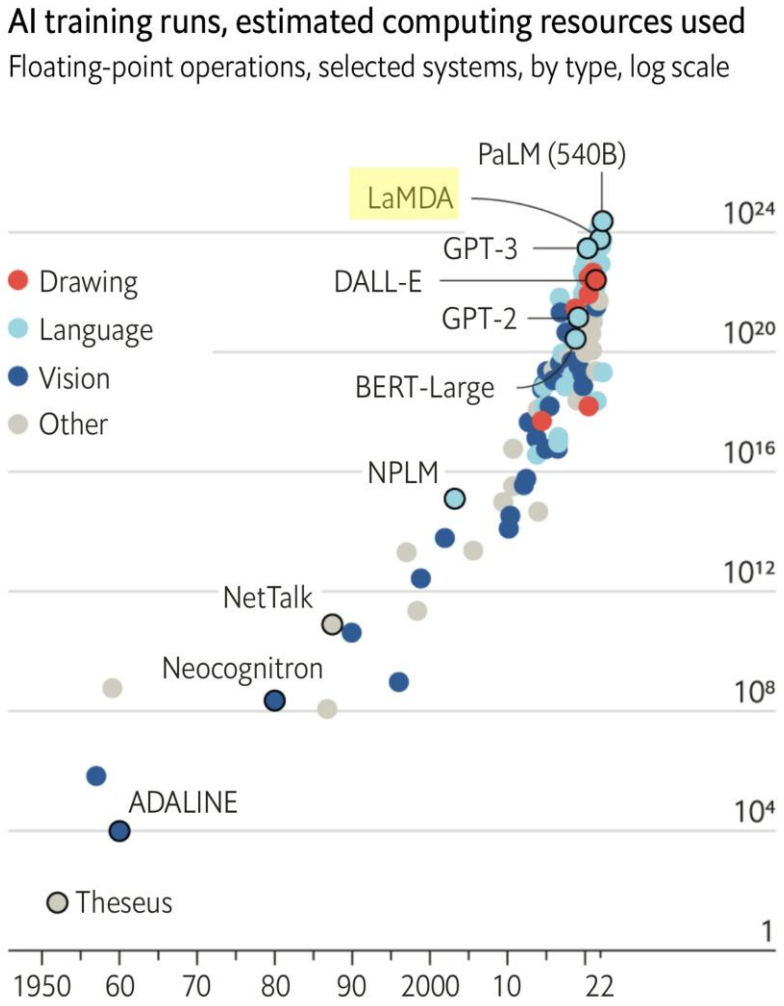


Transformer-based LLMs have revolutionized NLP!

Introduction to LLMs

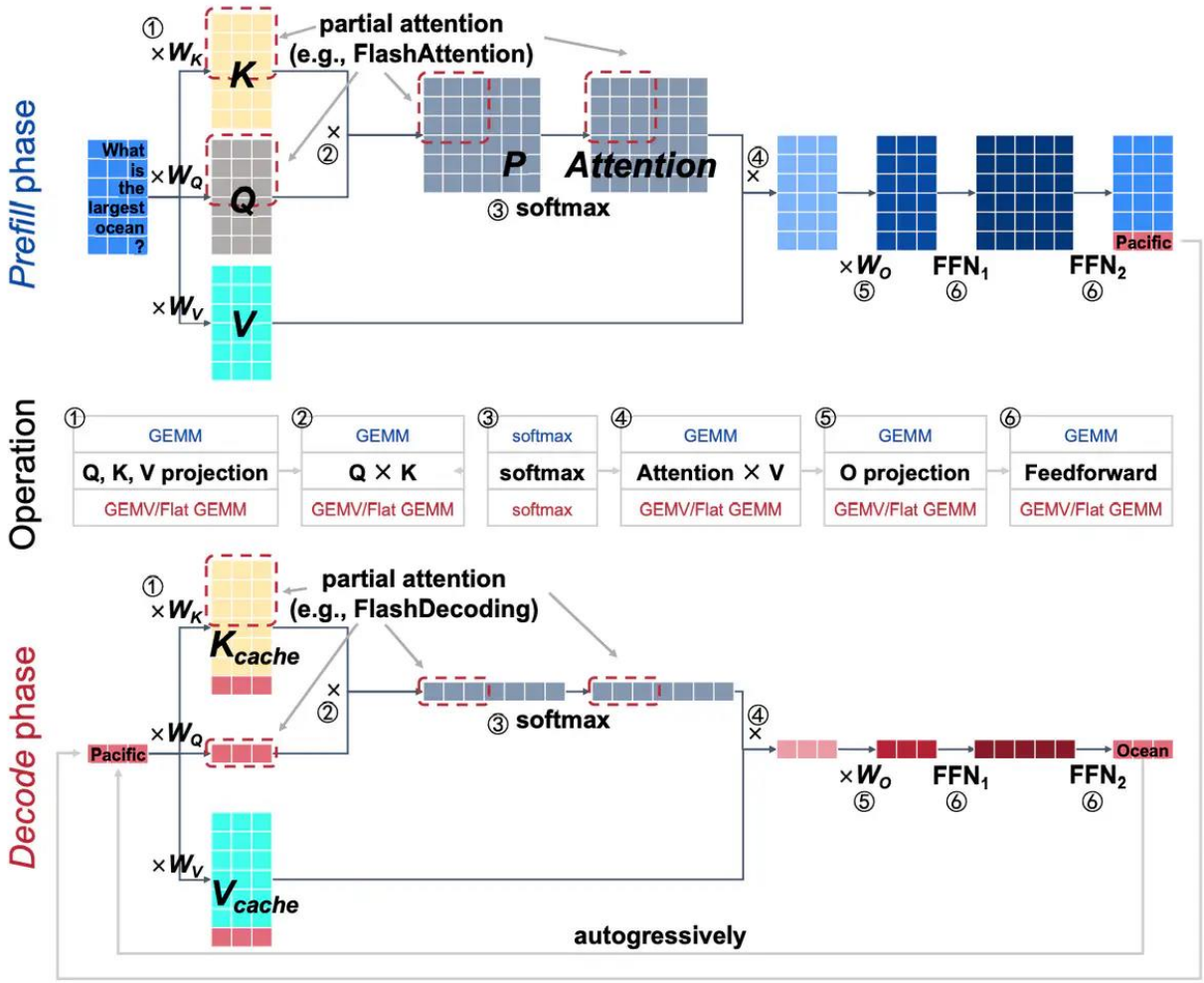


Bigger is Better !



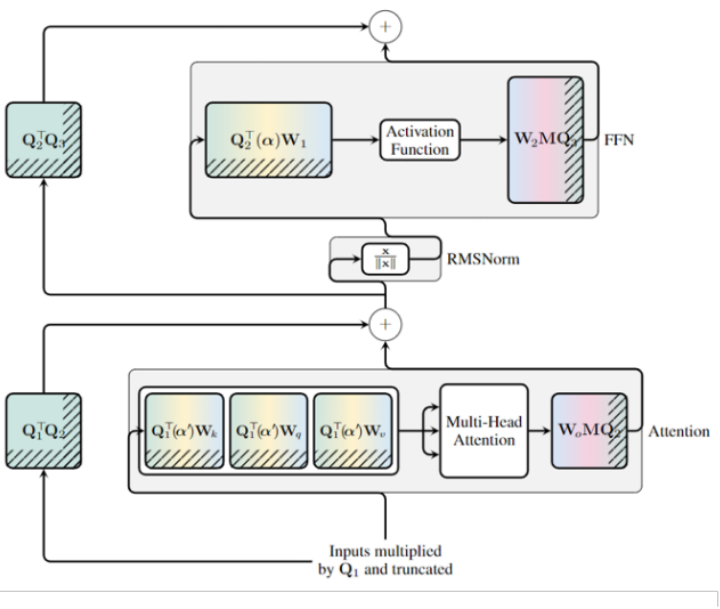
Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data

Introduction to LLMs



***Prefill* phase & *Decode* phase**

Model Compression



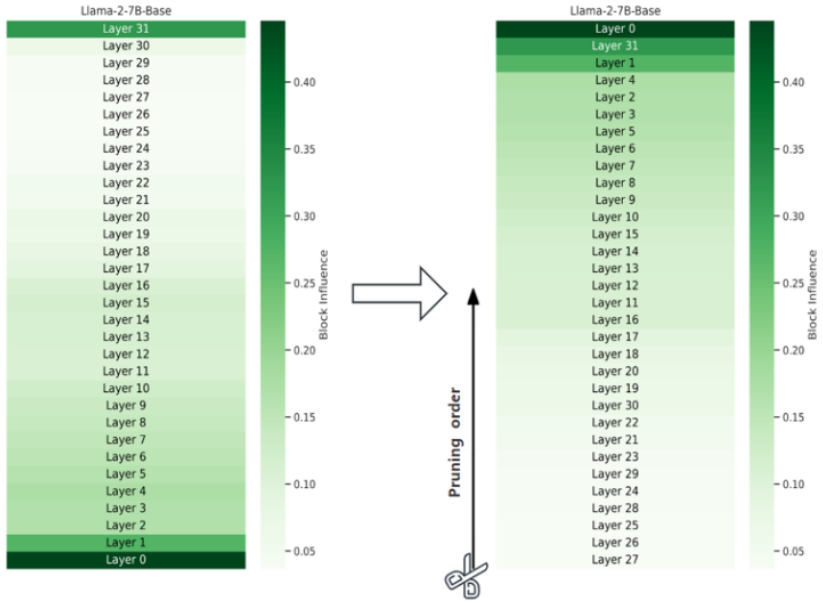


Fig. 2. Left: SliceGPT - Making LLMs Leaner, Right: ShortGPT - Making LLMs Shorter.

TABLE II
COMPARISON OF PRUNING METHODS ON PIQA. IN SHORTGPT-N, 'N' REFERS TO THE NUMBER OF PRUNED BLOCK LAYERS.

Method	Memory (MB)	PIQA
Dense(F16)	3503	0.733
SliceGPT(15%)	2999	0.659
SliceGPT(25%)	2625	0.611
ShortGPT-2	3300	0.712
ShortGPT-4	3000	0.691

Model Compression

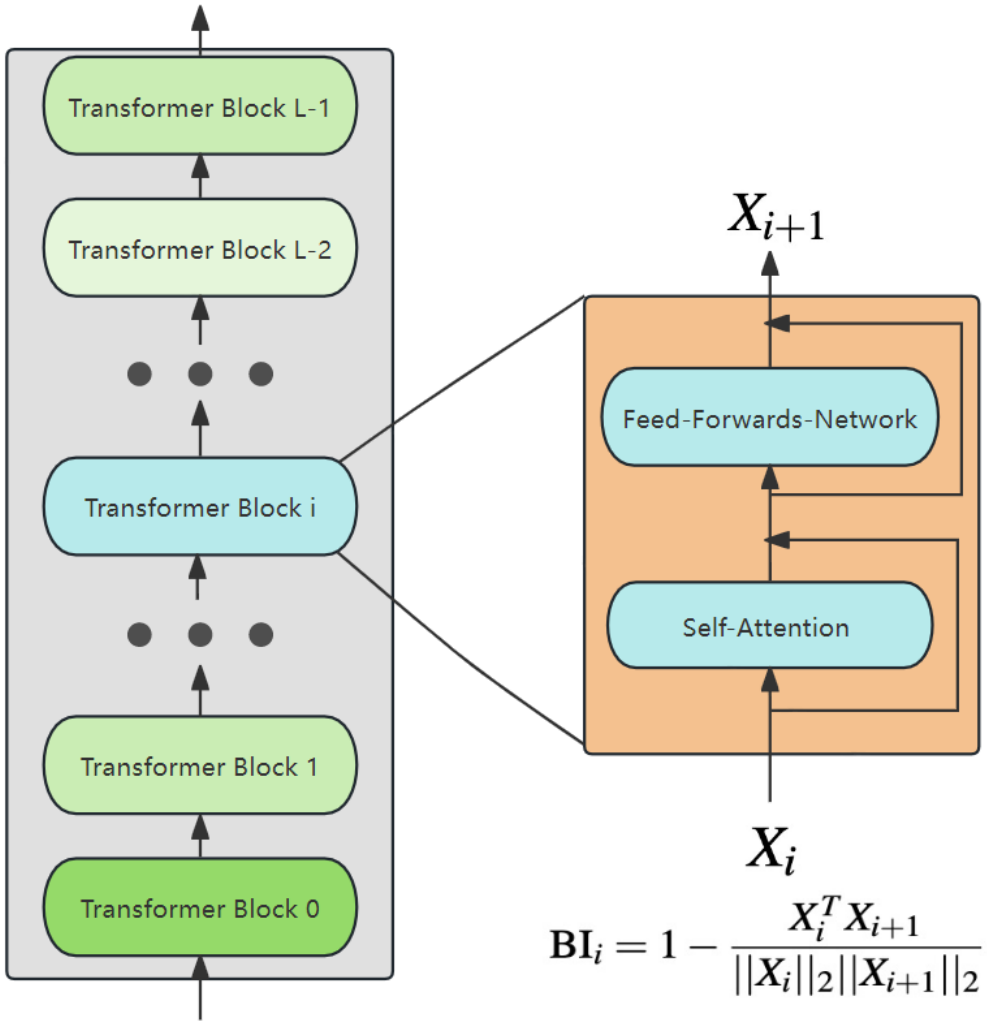


TABLE I
SETUP OF REMOVED LAYERS FOR QWEN-1.8B AND COMPARISON OF STRATEGIES FOR LAYER REMOVAL ON PIQA.

Method	Removed Layers	PIQA
RM [13]	7,9,8,10	0.6630
BI [8]	20,21,22,11	0.6512
LogRM (ours)	10,11,9,12	**0.6908**

$$RM = ||\frac{f(x)}{x + f(x)}|| \tag{3}$$

$$BI = 1 - \frac{\mathbf{X}_i^T \mathbf{X}_{i+1}}{||\mathbf{X}_i||_2 ||\mathbf{X}_{i+1}||_2} \tag{4}$$

$$LogRM = abs(log(\frac{||\frac{f(x)}{x+f(x)}||}{1 - ||\frac{f(x)}{x+f(x)}||})) \tag{5}$$

Low-bit Quantization

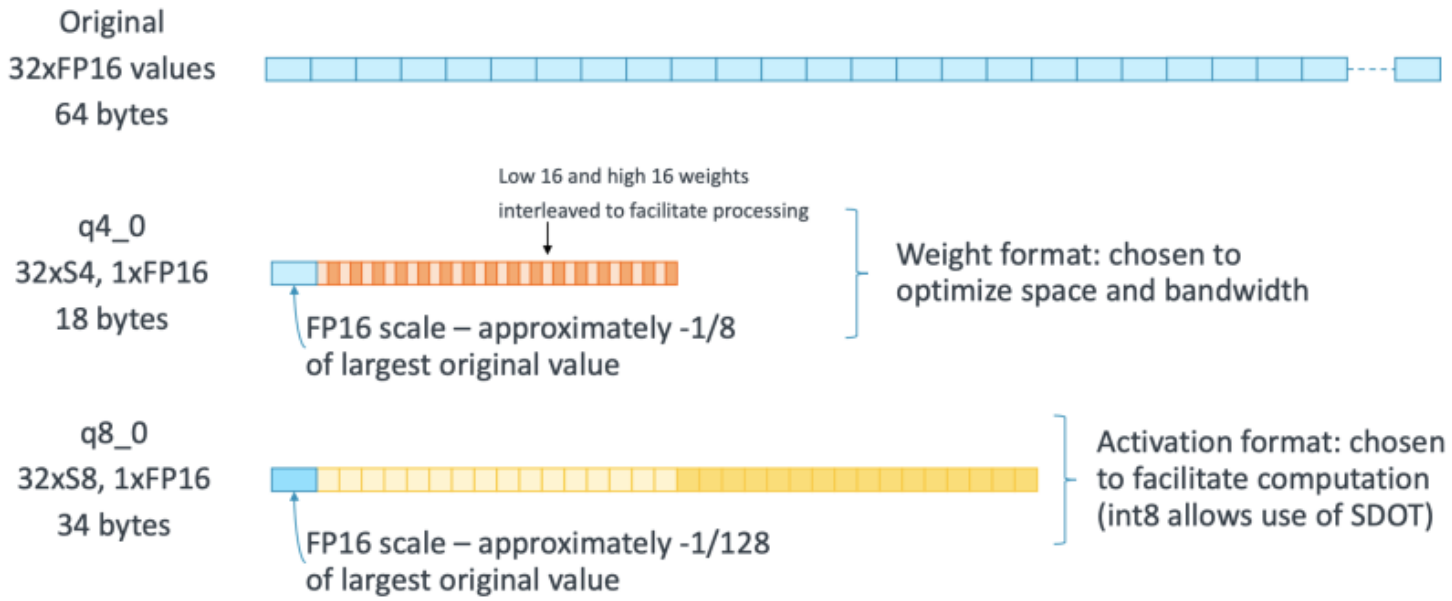


Fig. 3. Block Quantized Formats

TABLE III
COMPARISON OF K-QUANTS METHODS ON QWEN-1.8B

Method	Memory (MB)	PIQA
Dense(FP16)	3420.0	0.7312
Q2_K	791.9	0.6502
Q4_0	1040.0	0.7144
Q4_K_M	1150.0	0.7198
Q8_0	1820.0	0.7265

Kernel Optimization

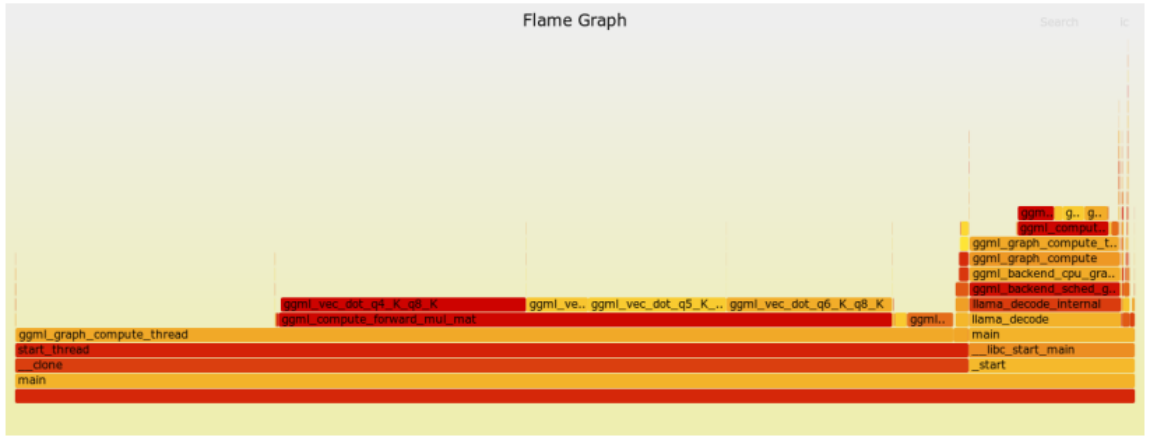
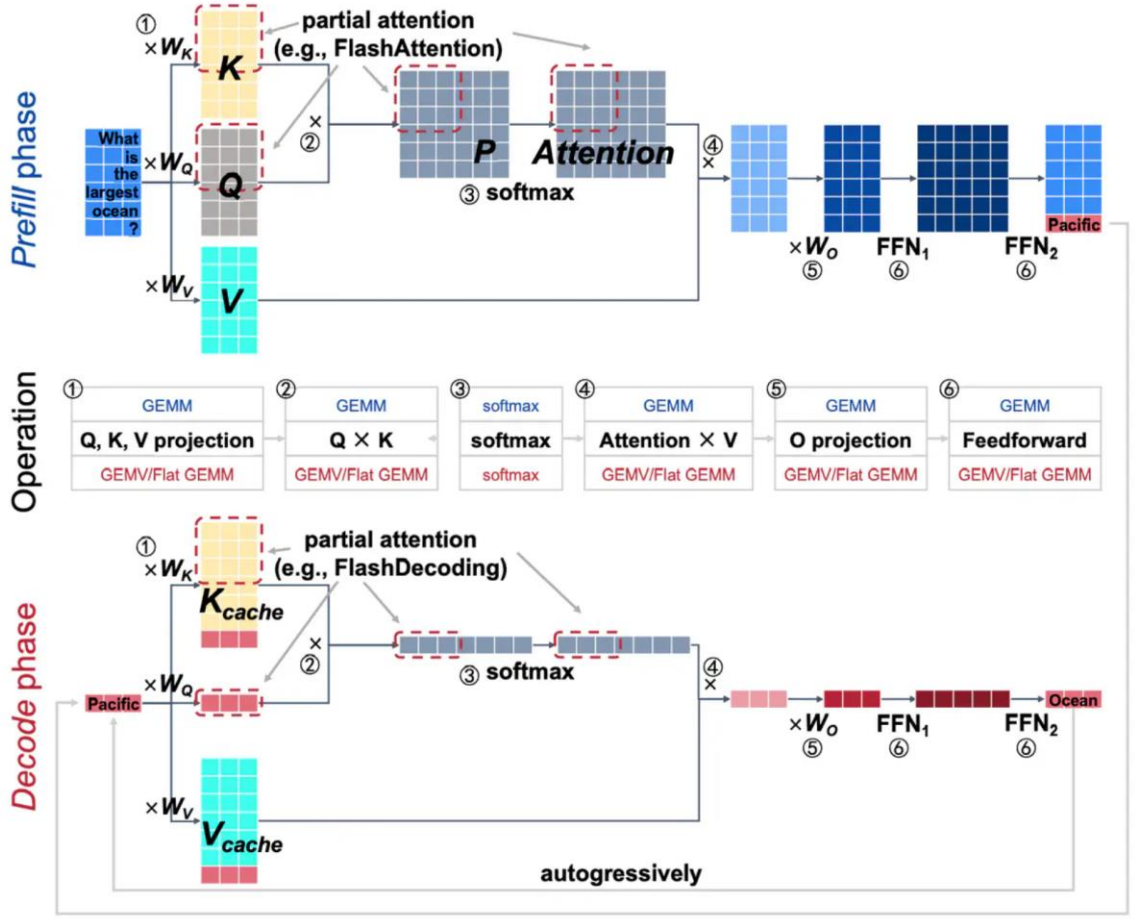


Fig. 4. ARM CPU Flame Graph



Kernel Optimization

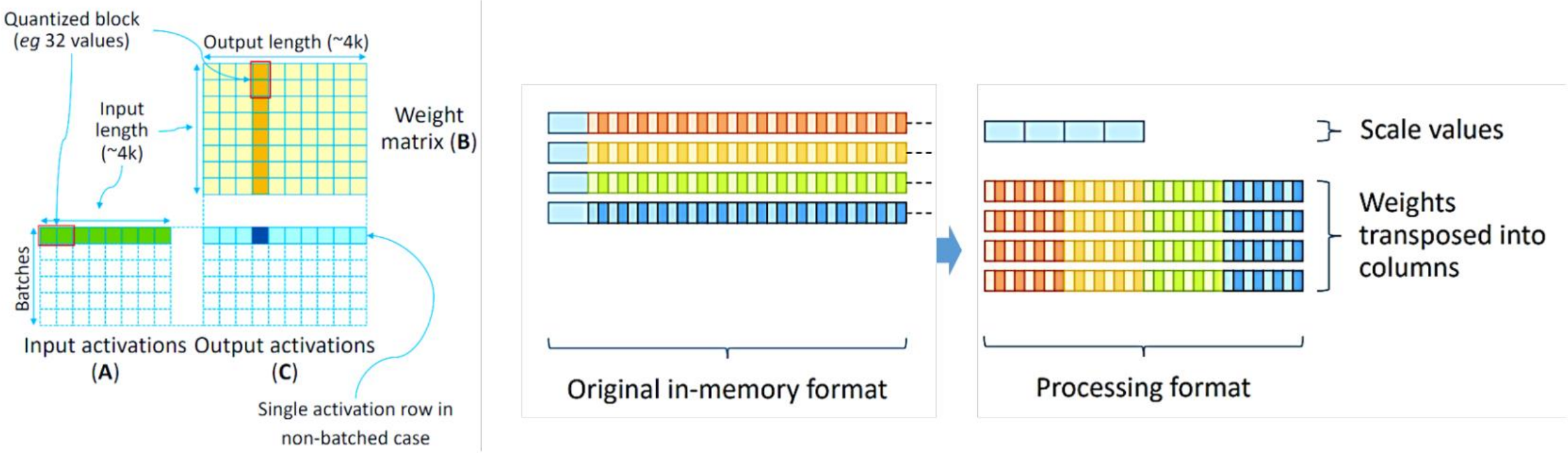


Fig. 5. To avoid pseudo-scalar operations, rearrange parameters in the weight matrix.

Sources: "Large Language Models on CPUs", by Dibakar Gope, David Mansell, Ian Bratt, PCCC23

Results

Our experimental setup utilized an ARM CPU (Yitian 710) supplied by the event organizers, which boasted a 8-core CPU, 30GB of memory, and 50GB of disk space. Meanwhile, in compliance with the competition's specifications, we validated our optimization method using the Qwen-1.8B [4].

TABLE V
LLM PREFILL THROUGHPUT OUTPERFORMS PYTORCH-BASED SOLUTION BY UP TO 4.38X UNDER SHORT5-Q8_0+RW.

Model	PIQA	RSS	VMS	Prefill	Decode	Score
Short2-Q4_0	0.7084	1180	1276	121.69	49.65	60.05
Short2-Q8_0	**0.7255**	1970	2064	140.93	52.01	67.86
Short4-Q4_0	0.6823	**1132**	**1227**	134.34	52.66	47.29
Short4-Q8_0	0.7160	1835	1930	163.67	55.47	64.62
Short5-Q8_0	0.7187	1775	1849	175.55	56.18	66.74
Short5-Q8_0+RW	0.7188	2756	2871	**380.61**	**64.05**	**70.66**

$$Score = Acc \cdot 0.30 + RSS \cdot 0.15 + VMS \cdot 0.15 + Pre \cdot 0.15 + Dec \cdot 0.25 \quad (6)$$

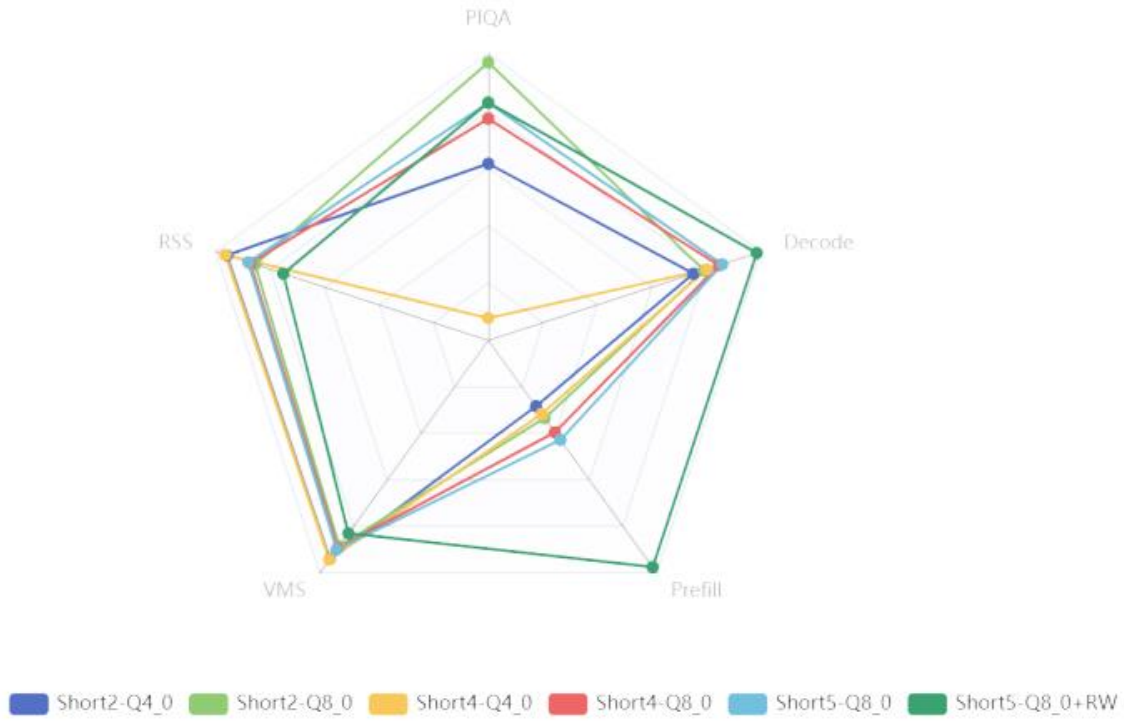


Fig. 6. Performance of different model sparsification and quantization methods. The increase in memory space resulting from the rearrangement of the weight matrix is traded for enhanced prefill throughput, denoted as **+RW**

Conclusion And Future Work

- CPUs serve as a practical platform for running LLMs inference.
- We demonstrated the performance advantage over the open-source sol ution on ARM CPUs.
- Future work to optimize batched inference on ARM CPUs.