The 2nd Workshop on Hot Topics in System Infrastructure

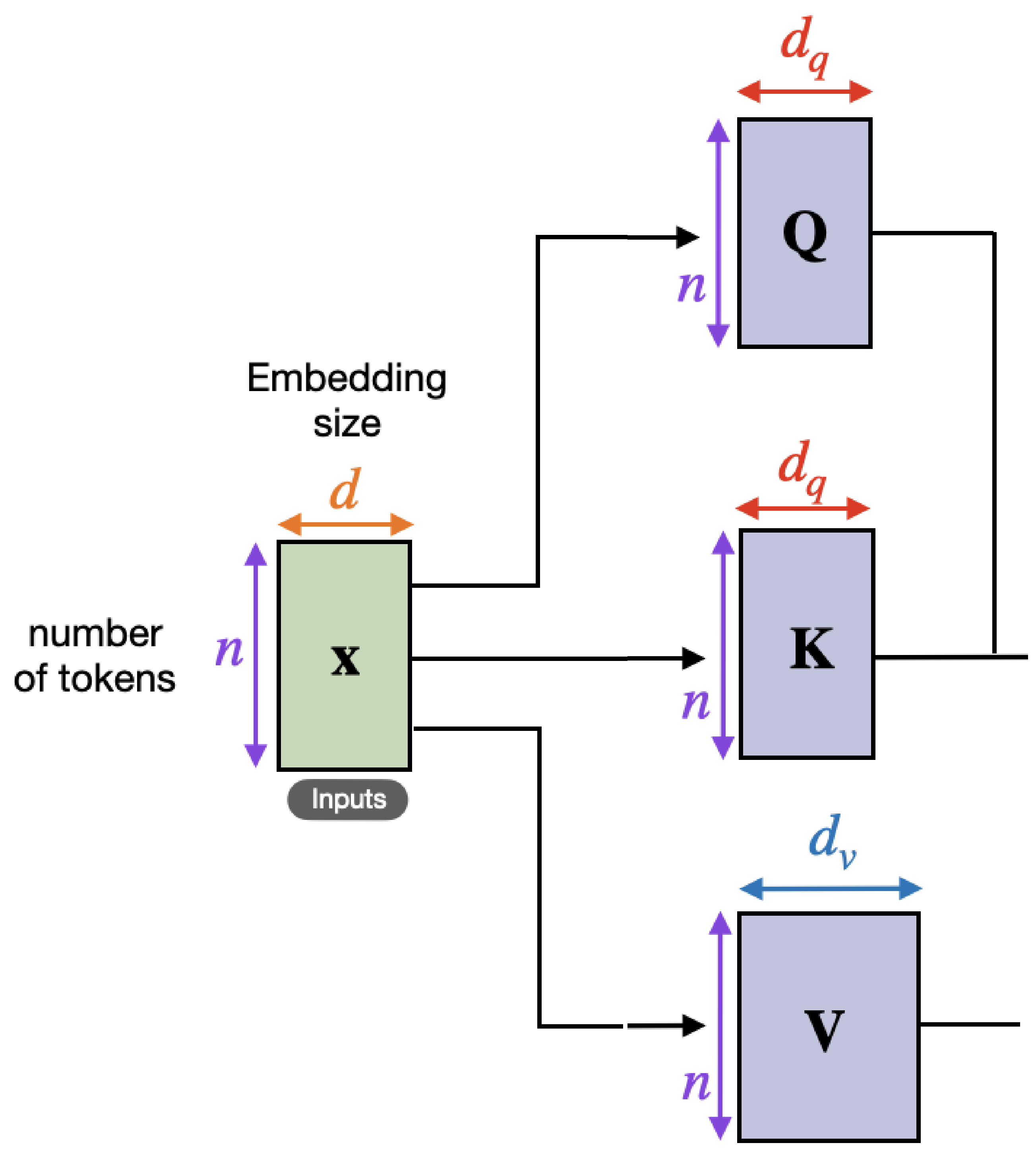
Optimizing Transformer Inference with Selective Distillation: Layerwise Conversion to Linear Attention

Yeonju Ro
The University of Texas at Austin

Yeonju Ro, Zhenyu Zhang, Vijay Chidambaram, Aditya Akella

Motivation

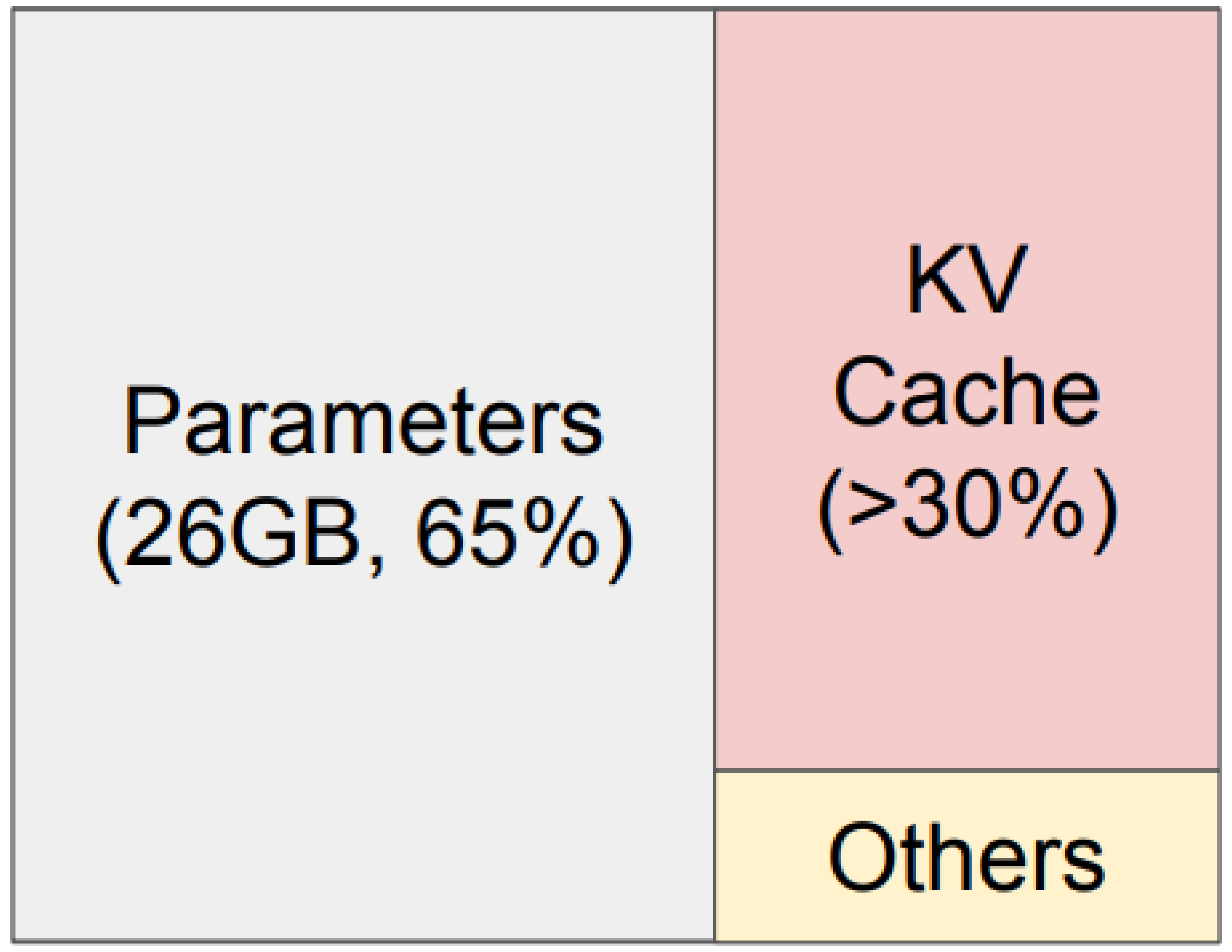
Self-attention is expensive for inference (1) Computation cost



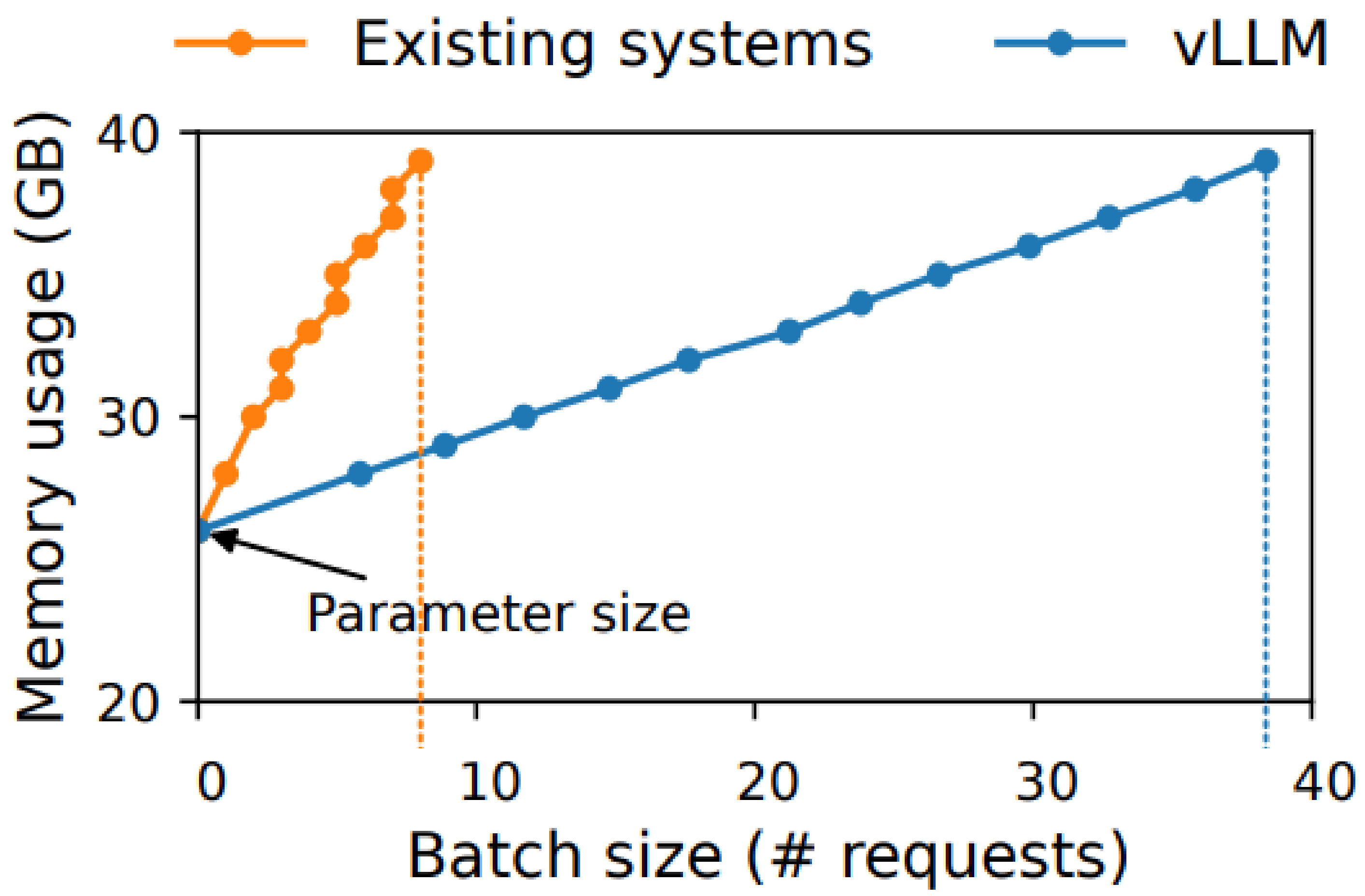
Quadratic to the sequence length

Motivation

Self-attention is expensive for inference (2) KV Cache



NVIDIA A100 40GB

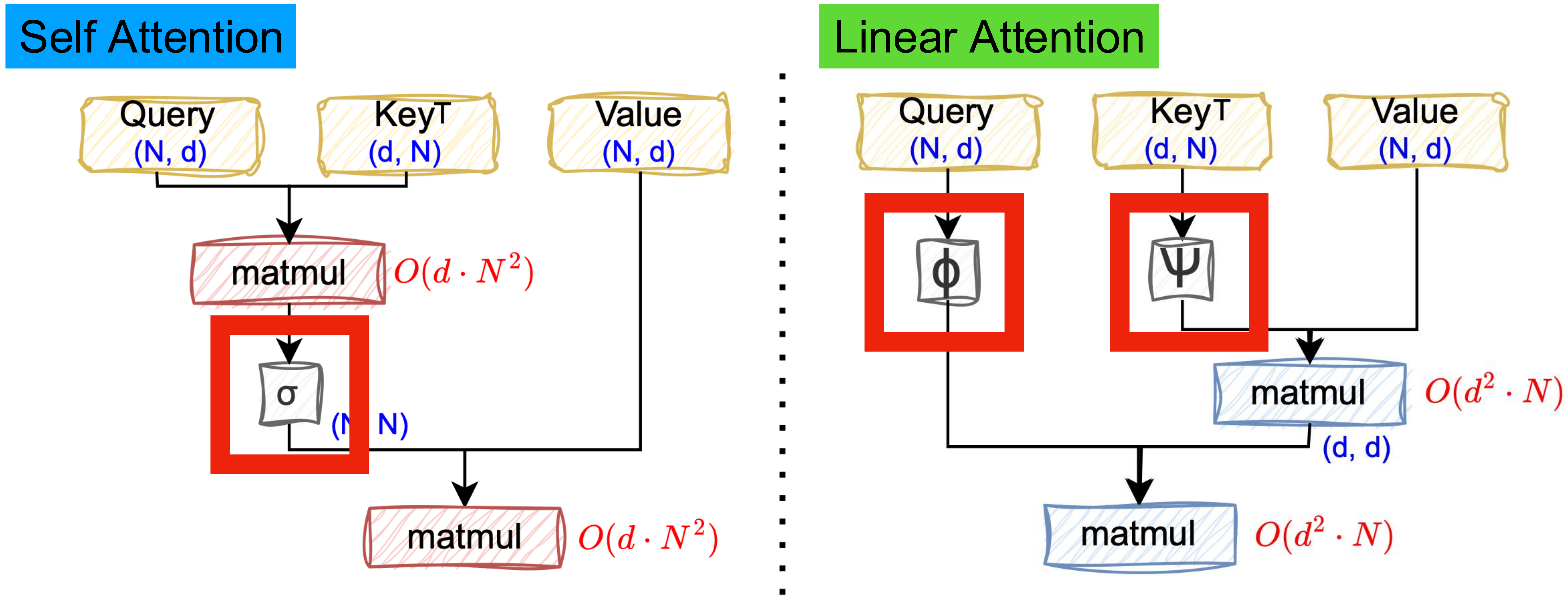


Figures are borrowed from vLLM Paper

Kwon, Woosuk, et al. "Efficient memory management for large language model serving with pagedattention." *Proceedings of the 29th Symposium on Operating Systems Principles*. 2023.

Emergence of RNN-based Language Models

Linear attention linearizes the computation cost of Attention



- Linear attention approximates softmax function with projection kernels (e.g., $\phi(), \psi()$)
- With use of associative property of matrix multiplication, computation cost is reduced to $O(n \cdot d^2)$

Computation cost —> $)(n \cdot d^2)$

Figures are borrowed from Linformer Paper

Wang, Sinong, et al. "Linformer: Self-attention with linear complexity." arXiv preprint arXiv:2006.04768 (2020).

Emergence of RNN-based Language Models

Linear attention can be computed in recurrent form

$$
\begin{aligned}
s_0 &= 0, \\
z_0 &= 0, \\
s_i &= s_{i-1} + \phi\left(x_i W_K\right)\left(x_i W_V\right)^T, \\
z_i &= z_{i-1} + \phi\left(x_i W_K\right), \\
y_i &= f_l\left(\frac{\boxed{\phi\left(x_i W_Q\right)^T s_i}}{\phi\left(x_i W_Q\right)^T z_i} \boxed{+ x_i}\right).
\end{aligned}
$$

By computing output in a recurrent form,
Linear attention does not need to keep KV Cache anymore.

Tradeoff between Self-Attn and Linear-Attn

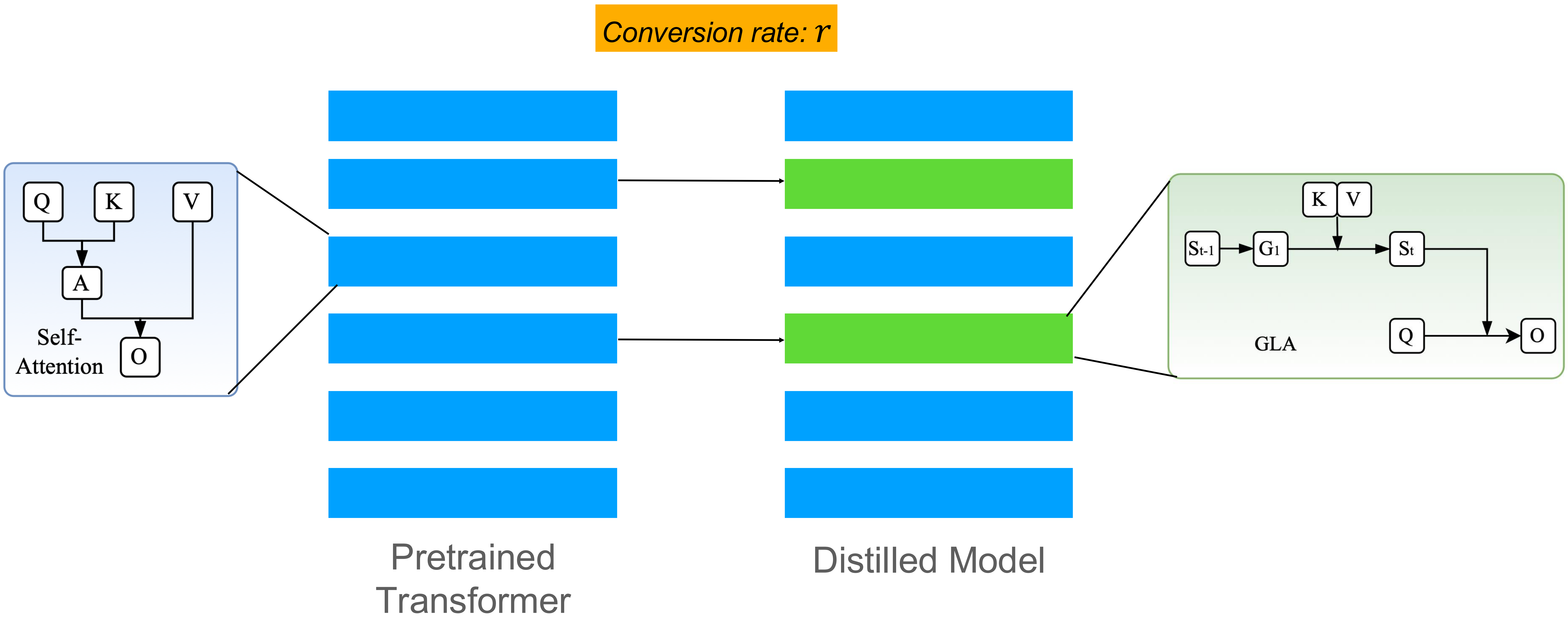
	Training Complexity	**Inference Complexity**	**Model Performance (Accuracy)**
Self Attention	Easy to Parallelize (e.g., TP, PP, DP)	Quadratic	Great
Linear Attention	Hard to Parallelize	Linear	Worse
Mamba or Gated Linear Attention	Somewhere in the middle (e.g., Parallel-scan)	Linear	Better than Linear Attn, Worse than Self Attn.

How to achieve benefits of both?

Train in transformer architecture, and convert it to a linear model for inference

How to achieve benefits of both?

Selectively Distilling Self Attention to Linear Attention



How to achieve benefits of both?

Distilling Self Attention to Linear Attention

- We define our kernel functions $\phi(q)$ and $\psi(k)$ with Multi-layer Perceptron (MLP) and GELU activation function.
- As a loss function, we minimize the l_2 distance to the output projection of the original attention layer.

$$\phi(q) = \left|\sigma_\phi\left(\sigma_\phi(qW_{\phi,1})W_{\phi,2}\right)\right|$$
$$\psi(k) = \left|\sigma_\psi\left(\sigma_\psi(kW_{\psi,1})W_{\psi,2}\right)W_{\psi,3}\right|$$

$$\mathcal{L} = ||\text{attn\_out} - \text{out}|| + \lambda_1 \cdot ||q - \phi(q)|| + \lambda_2 \cdot ||k - \psi(k)||$$

Selective Conversion of Layers

How many layers to convert?

- Computations in transformer and linear model are determined by the sequence length, the latency can be easily interpolated by performing a single inference run at each end of the conversion spectrum.
- Given latency budget lat_b, we measure the latency lat_0 using a base transformer, and lat_1 using a fully linear model.
- Maximum conversion rate r for a given latency budget can be obtained as follows.

$$r = \arg\max \{ r \mid \mathrm{lat}_b \geq |r \cdot \mathrm{lat}_0 + (1 - r) \cdot \mathrm{lat}_1| \}$$

Which layers to convert?

Selection criteria

- Layer position: Early/late layers benefit from sequential info (e.g., linear attention); middle layers rely on self-attention for capturing fine-grained token relationships.
- Attention distribution (by entropy): Analyzing attention distribution helps identify which layers have a more focused and meaningful distribution, which may benefit from self-attention.
- Task accuracy: Per-layer sensitivity to the downstream task.

Evaluation

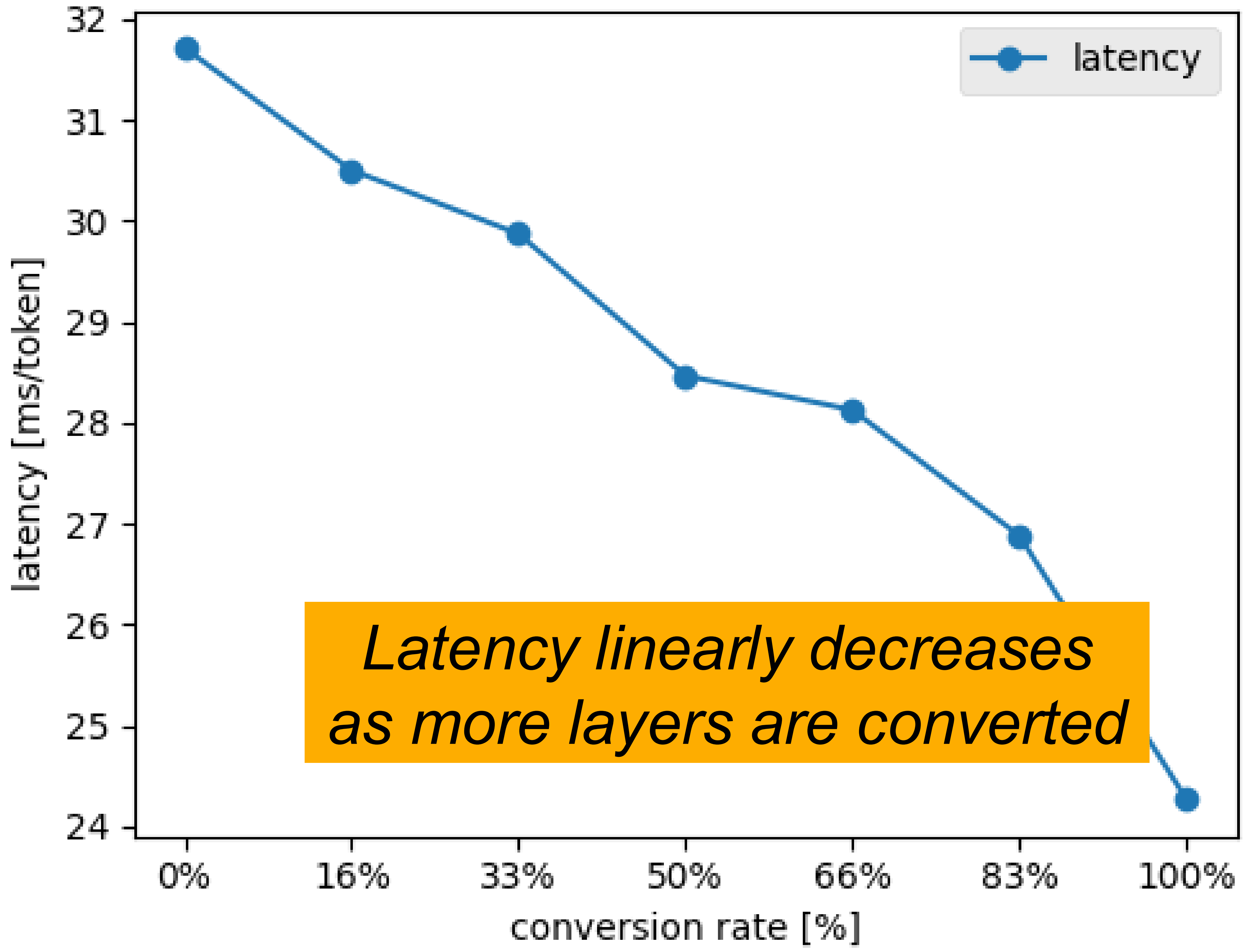
Experimental Setup

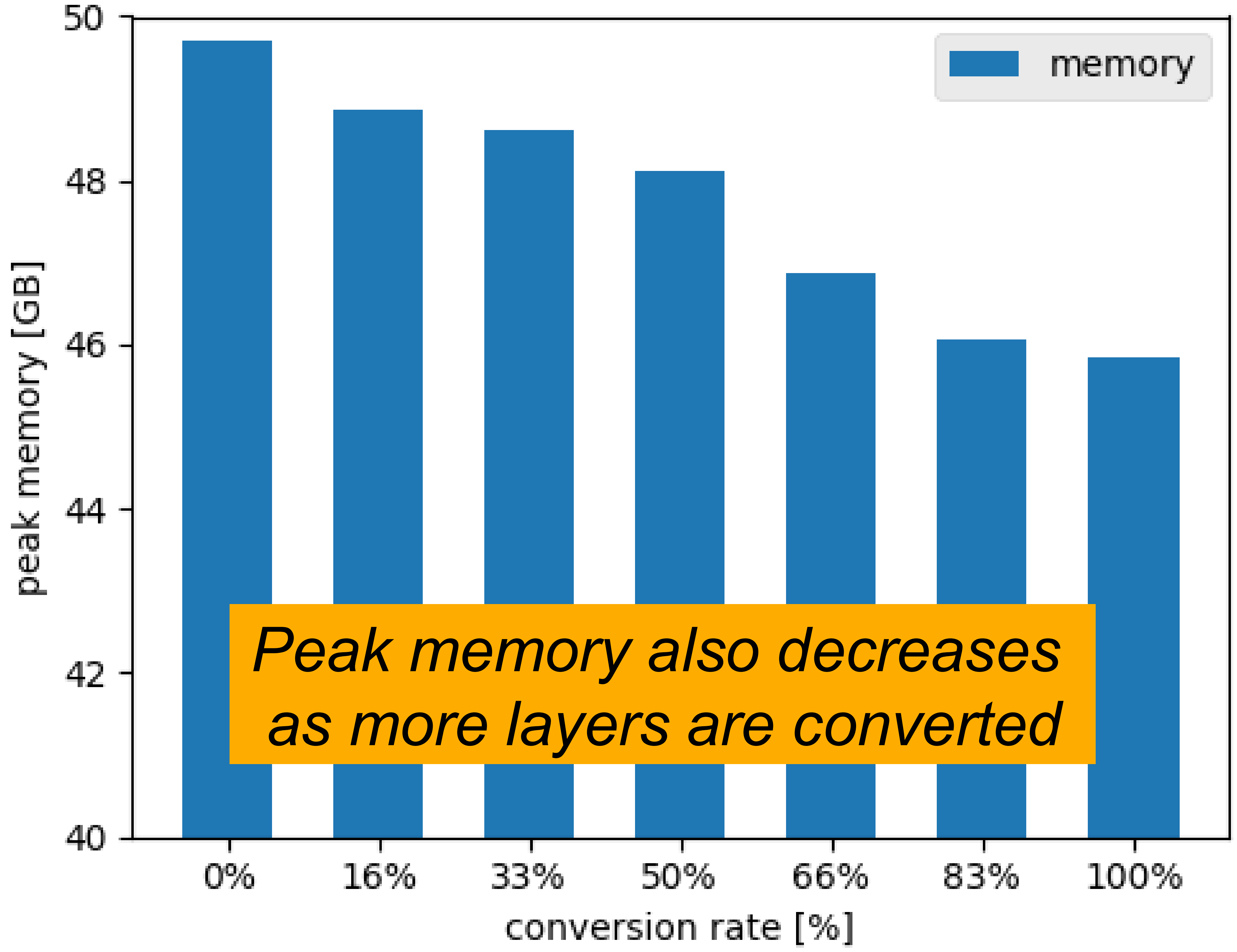
- Used a single A100 GPU with 80 GB of memory.
- Base transformer model: Llama2-7b
- Distilled linear model size: 7b
- Distillation used 512 sequences from C4 training dataset.
- Evaluation task: Text summarization task on two different datasets

  (CNN • Dailymail/ Xsum), used ROUGE-1 score as a metric which measures the overlap of unigrams between a generated text and a ground-truth text.

System Performance

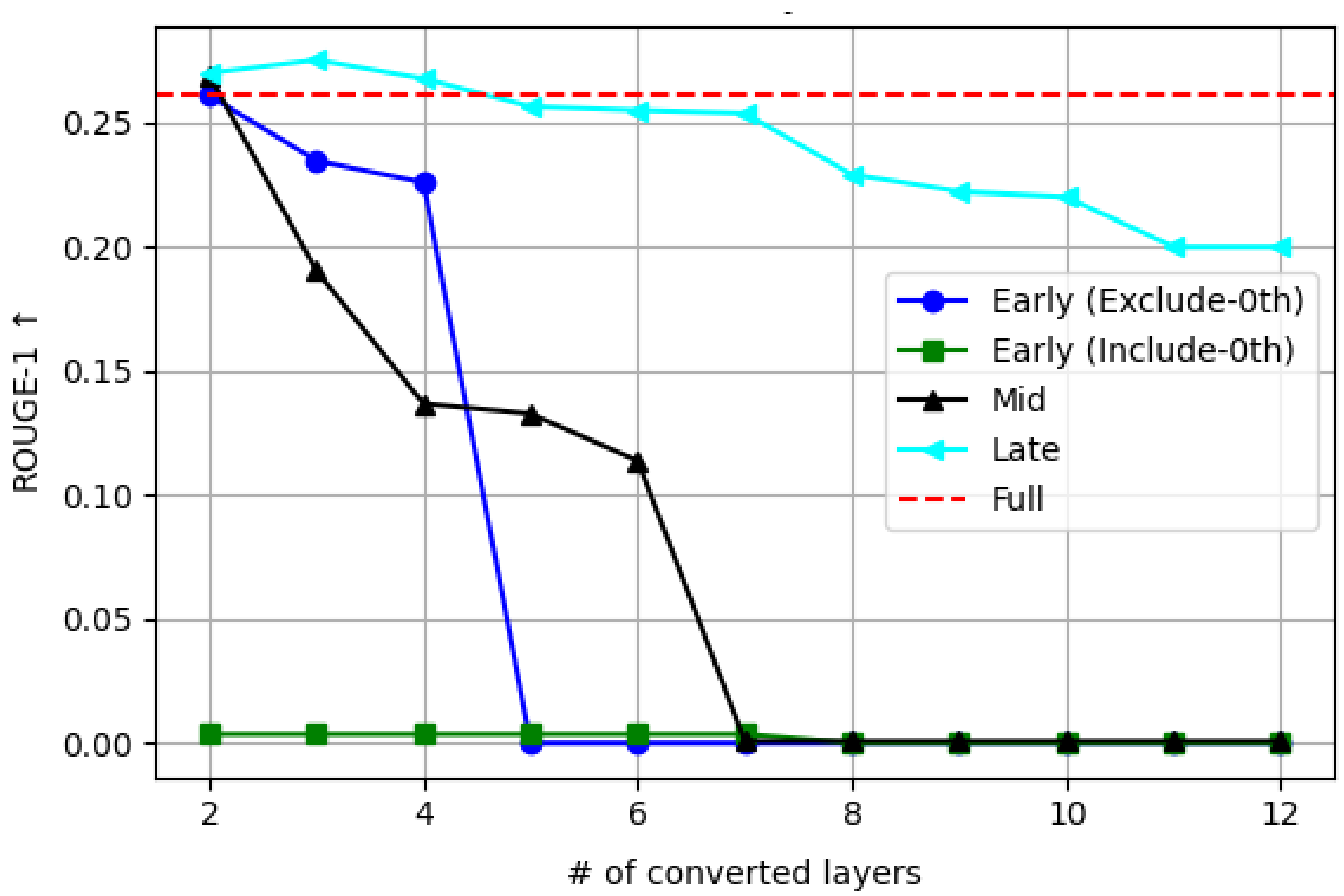
Inference latency and Peak memory



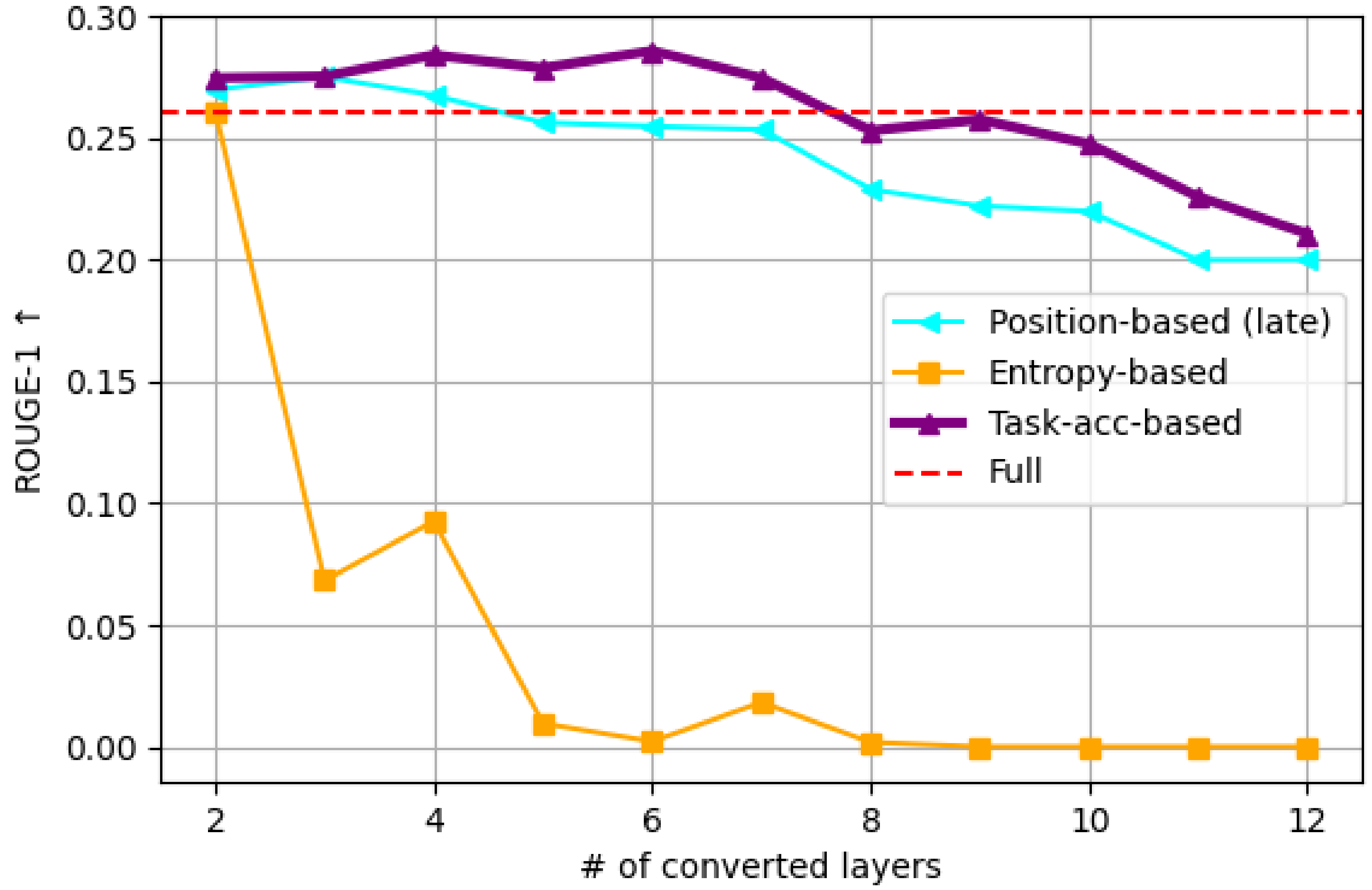


Selection Criteria

For given selection rate, which layer should we convert?



Early vs Mid vs **Late**



Position-based

vs

Entropy-based

vs

Task-acc-based

Conclusion

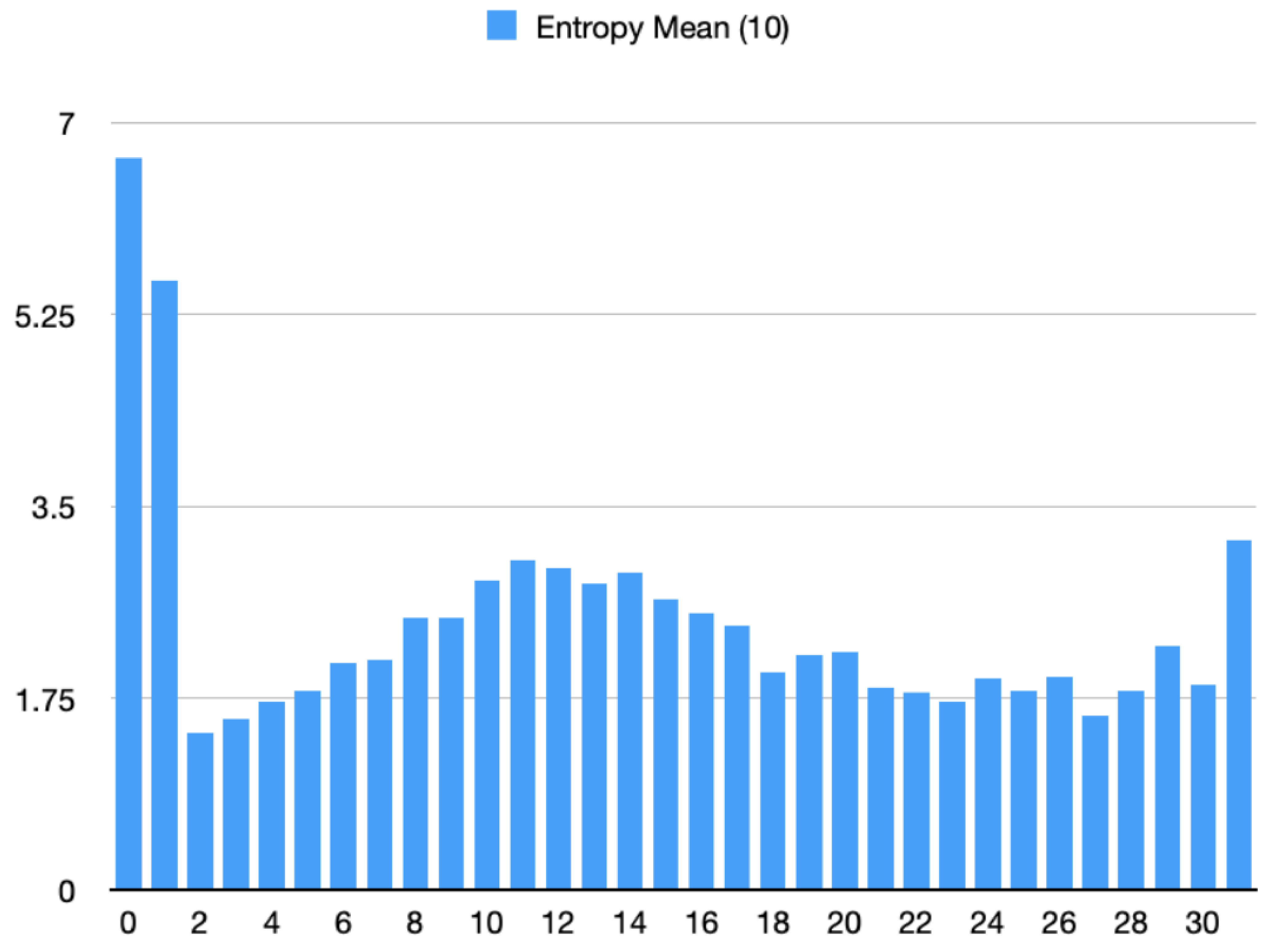
And future work

- In this work, we proposed a distillation framework that selectively distills transformer layers to gated linear attention layers for efficient inference.
- Our framework allows balancing the accuracy of distilled model and inference efficiency, for a given latency budget.
- Future work includes supporting more objective (e.g., accuracy, throughput) for elasticity.

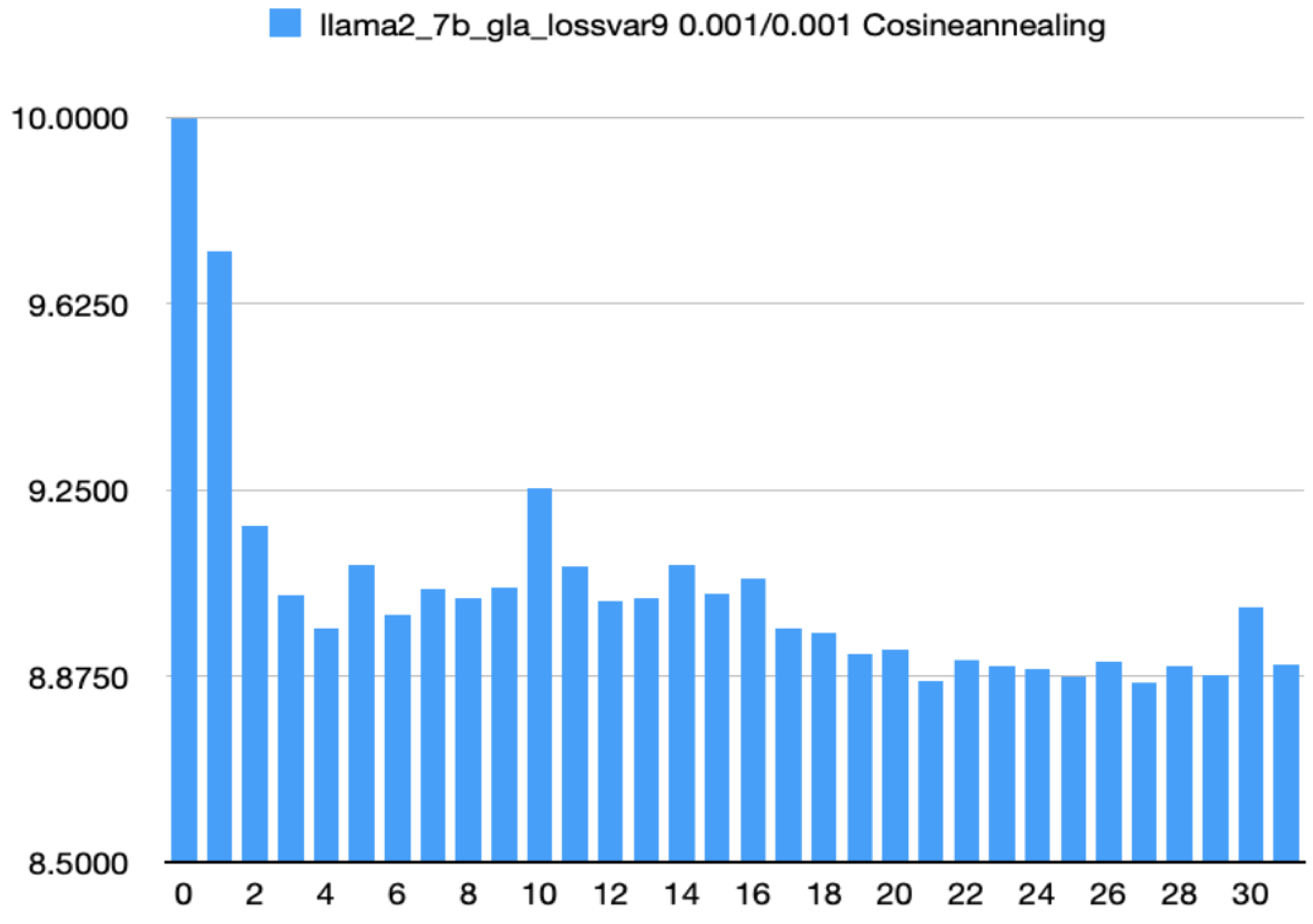
Q&A

Thank you!

Entropy and Perplexity



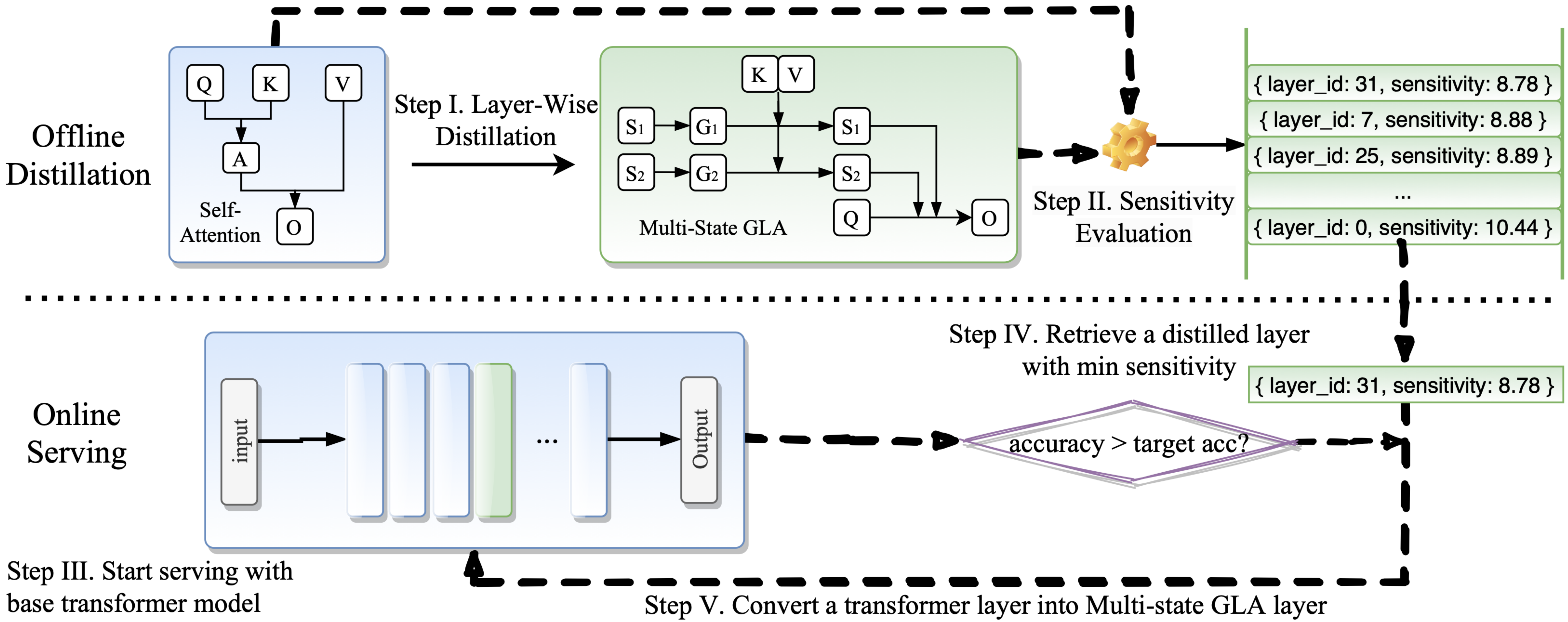
entropy mean



wikitext ppl

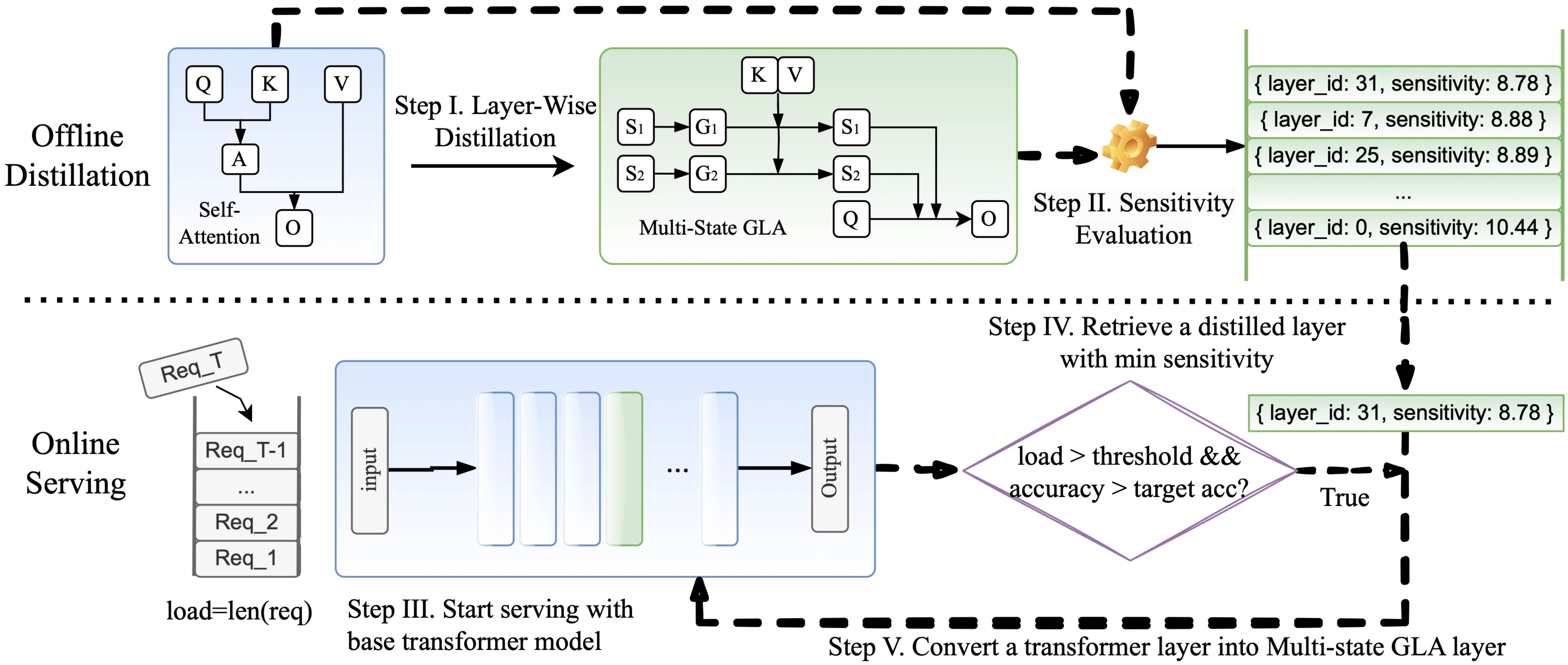
Extension: Adaptive Conversion of Layers

Accuracy-feedback-loop to meet Accuracy SLO



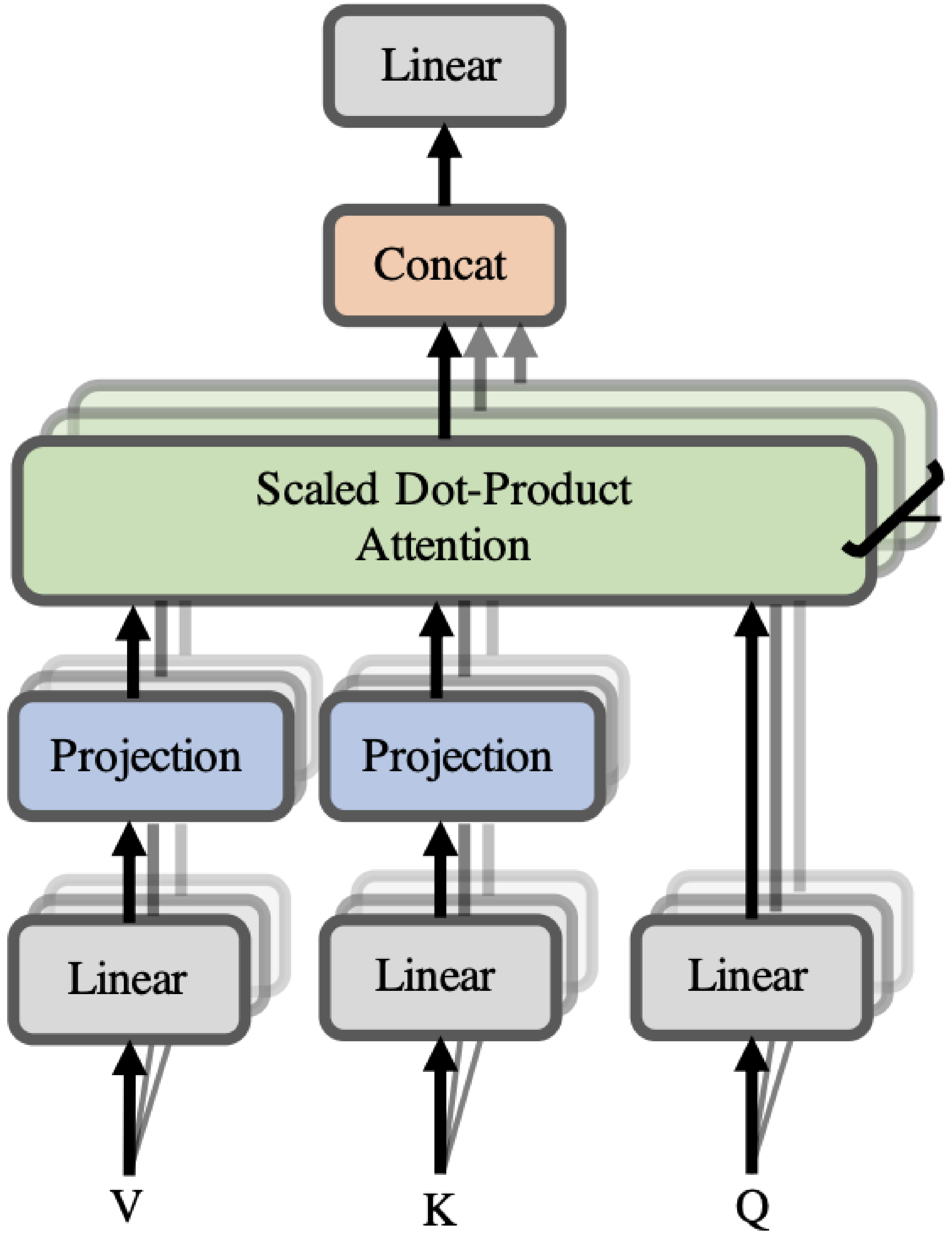
Extension: Adaptive Conversion of Layers

For elastic inference to support dynamic load and meet accuracy SLO



Emergence of RNN-based Language Models

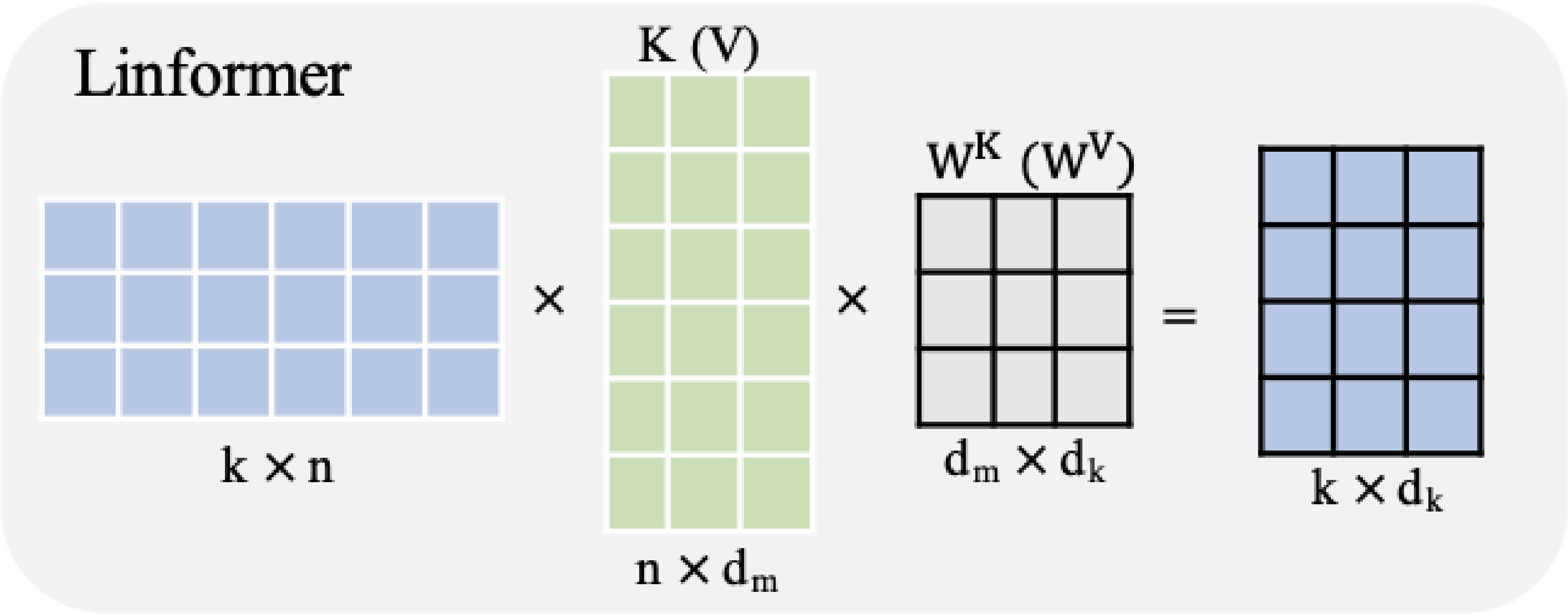
Linear attention linearizes the computation cost of Attention



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{Q \cdot K^{\intercal}}{\sqrt{d_k}}\right) \cdot V$$

$$\text{LinearAtttention}(Q, K, V) = \frac{\phi(Q_i)\left(\sum_{j=1}^{N} \phi(K_j)^{\intercal} V_j\right)}{\phi(Q_i) \sum_{j=1}^{N} \phi(K_j)}$$

Can reuse these for every query
—> Memory cost becomes O(N)



Computation cost —> $O(n \cdot d^2)$