Tensor Product Attention Is All You Need

Yifan Zhang*, Yifeng Liu*, Huizhuo Yuan, Zhen Qin,
Yang Yuan, Quanquan Gu, Andrew C Yao†

https://github.com/tensorgi/TPA *: equal contribution

MLA is RoPE-incompatible

→TPA is compatible to any positional encoding mechanism!

Decoding MLA is compute-intensive

→TPA requires fewer FLOPs!

MLA/GQA has large cache size

→TPA can reduce cache size!

MHA/MQA/GQA are different

→They are all special cases of TPA!

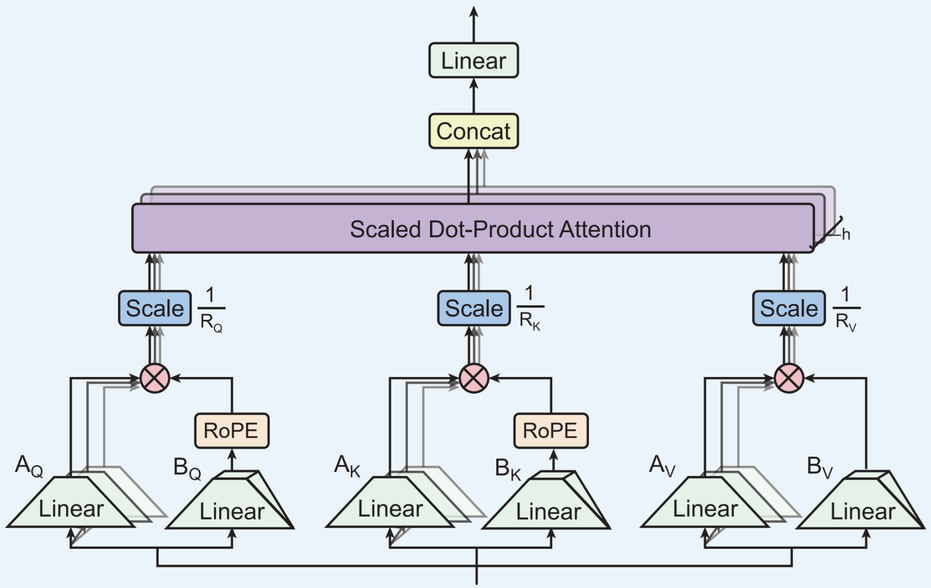
TPA factorizes each $\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t$ into a sum of (contextual) tensor products whose ranks are R_q, R_k, and R_v, respectively and may differ (Here $\mathbf{a}_r^Q(\mathbf{x}_t), \mathbf{a}_r^K(\mathbf{x}_t), \mathbf{a}_r^V(\mathbf{x}_t) \in \mathbb{R}^h$, $\mathbf{b}_r^Q(\mathbf{x}_t), \mathbf{b}_r^K(\mathbf{x}_t), \mathbf{b}_r^V(\mathbf{x}_t) \in \mathbb{R}^{d_h}$):

$$\mathbf{Q}_t = \frac{1}{R_Q}\sum_{r=1}^{R_Q} \mathbf{a}_r^Q(\mathbf{x}_t) \otimes \mathbf{b}_r^Q(\mathbf{x}_t),$$

$$\mathbf{K}_t = \frac{1}{R_K}\sum_{r=1}^{R_K} \mathbf{a}_r^K(\mathbf{x}_t) \otimes \mathbf{b}_r^K(\mathbf{x}_t),$$

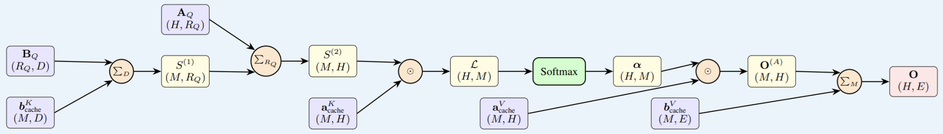
$$\mathbf{V}_t = \frac{1}{R_V}\sum_{r=1}^{R_V} \mathbf{a}_r^V(\mathbf{x}_t) \otimes \mathbf{b}_r^V(\mathbf{x}_t),$$

TPA Architecture

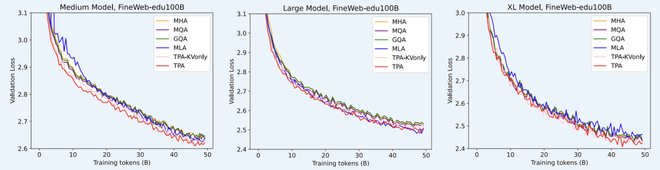


Method	KV Cache	# Parameters	# Query Heads	# KV Heads
MHA	$2hd_h$	$4d_{\text{model}}^2$	h	h
MQA	$2d_h$	$(2+2/h)d_{\text{model}}^2$	h	1
GQA	$2gd_h$	$(2+2g/h)d_{\text{model}}^2$	h	g
MLA	$d_c + d_h^R$	$d'_c(d_{\text{model}} + hd_h + hd_h^R)$ $+d_{\text{model}}d_h^R + d_c(d_{\text{model}} + 2hd_h)$	h	h
TPA	$(R_K + R_V)(h + d_h)$	$d_{\text{model}}(R_Q + R_K + R_V)(h + d_h) + d_{\text{model}}\, hd_h$	h	h
TPA (KVonly)	$(R_K + R_V)(h + d_h)$	$d_{\text{model}}(R_K + R_V)(h + d_h) + 2d_{\text{model}}\, hd_h$	h	h
TPA (Non-contextual A)	$(R_K + R_V)d_h$	$(R_Q + R_K + R_V)(d_{\text{model}}d_h + h) + d_{\text{model}}\, hd_h$	h	h
TPA (Non-contextual B)	$(R_K + R_V)h$	$(R_Q + R_K + R_V)(d_{\text{model}}h + d_h) + d_{\text{model}}\, hd_h$	h	h

FlashTPA



Experiments

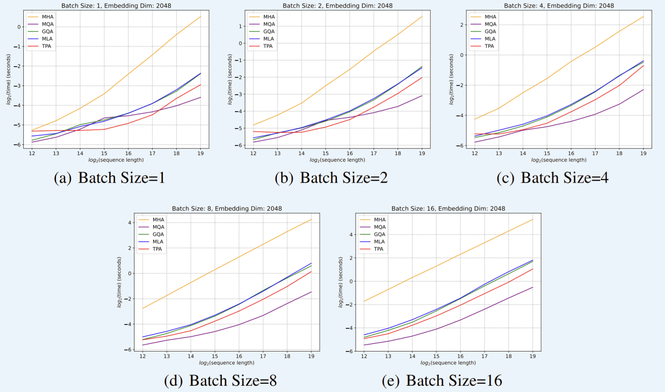


(a) Medium models (353M) (b) Large models (773M) (c) XL models (1.5B)

Figure 4: The validation loss of medium-size (353M), large-size (773M) as well as XL-size (1.5B) models, with different attention mechanisms on the FineWeb-Edu 100B dataset.

Table 12: Evaluation results of XL models (1.5B) with different attention mechanisms, pre-trained on the FineWeb-Edu 100B dataset (0-shot with lm-evaluation-harness). The best scores in each column are **bolded**. Abbreviations: HellaSw. = HellaSwag, W.G. = WinoGrande. If not specified, TPA and TPA-KVonly models use $R_K = R_V = 2$.

Method	ARC-E	ARC-C	BoolQ	HellaSw.	OBQA	PIQA	W.G.	MMLU	SciQ	Avg.
MHA	64.81	35.41	61.90	54.32	37.20	72.74	55.80	25.44	**82.80**	54.49
MQA	64.10	36.01	62.26	54.38	39.00	72.58	56.43	23.70	81.90	54.48
GQA	63.68	35.92	60.46	54.17	38.40	**73.56**	56.27	24.77	81.70	54.33
MLA	64.14	35.92	60.12	53.60	39.20	72.25	55.17	24.71	81.60	54.08
TPA-KVonly	65.61	36.77	**63.02**	54.17	37.00	73.34	54.62	25.02	81.60	54.57
TPA-KVonly ($R_{K,V} = 4$)	64.52	**37.03**	63.27	**54.89**	39.80	72.91	56.51	24.74	81.60	**55.03**
TPA-KVonly ($R_{K,V} = 6$)	65.78	35.92	61.71	54.86	38.60	72.69	**57.93**	**25.59**	82.20	**55.03**
TPA	**66.71**	36.52	61.38	54.03	**40.40**	72.52	56.83	24.49	82.20	55.01



(a) Batch Size=1 (b) Batch Size=2 (c) Batch Size=4

(d) Batch Size=8 (e) Batch Size=16

Figure 5: Decoding time comparison of different attention mechanisms with an embedding dimension of 2048 and $d_h = 64$. The y-axis represents $\log_2$(time) in seconds, and the x-axis represents $\log_2$(sequence length). Each subfigure corresponds to a different batch size.