

Large Language Models

Advanced Attention Mechanisms - I

ELL881 · AIL821

Sourish Dasgupta

Assistant Professor, DA-IICT, Gandhinagar

<https://www.daiict.ac.in/faculty/sourish-dasgupta>

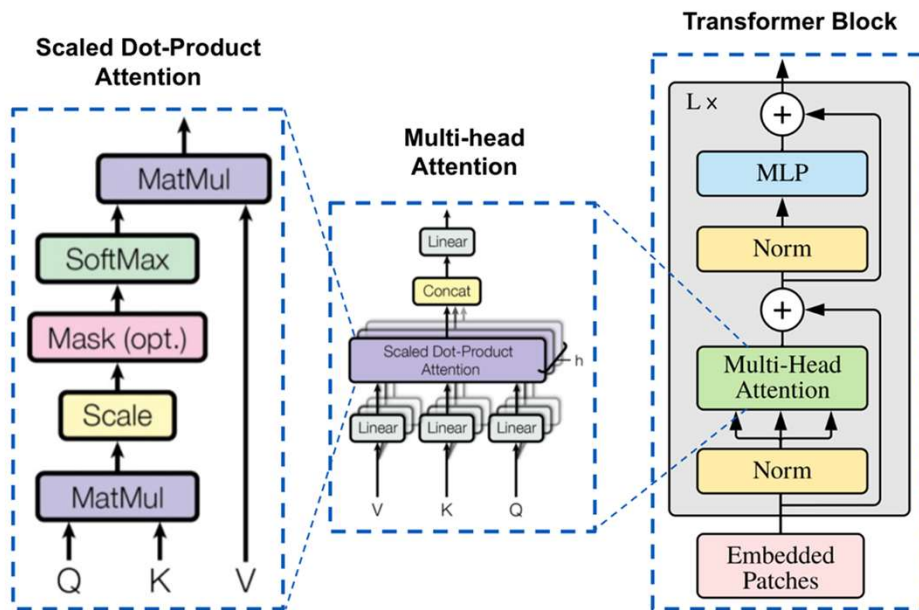


Semester 1, 2024-2025

Year: 2017, NeurIPS

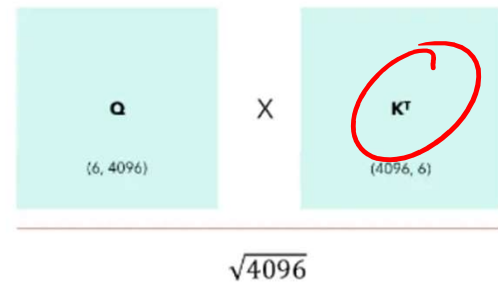


Self Attention



Similarity

Contextual embedding

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


	THE	CAT	IS	ON	A	CHAIR
THE	0.268	0.119	0.134	0.148	0.179	0.152
CAT	0.124	0.278	0.201	0.128	0.154	0.115
IS	0.147	0.132	0.262	0.097	0.218	0.145
ON	0.210	0.128	0.206	0.212	0.119	0.125
A	0.146	0.158	0.152	0.143	0.227	0.174
CHAIR	0.195	0.114	0.203	0.103	0.157	0.229

(6, 6)

Attention (loves. John)

John *loves* *IL* *made* *in* *NY*

v: values.

Handwritten notes include arrows pointing from the words 'loves' and 'IL' to the attention weights in the table above, and various symbols like '%' and 'v' indicating relationships between the words and the attention mechanism.



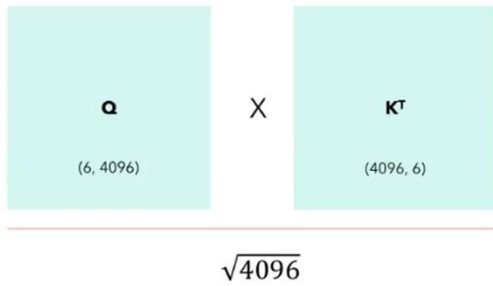
Year: 2017, NeurIPS



Causal (Forward Masked) Attention

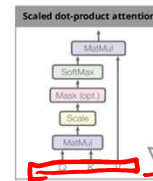
N: Sequence length
d: dimension
Similarity (dot products)
softmax

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



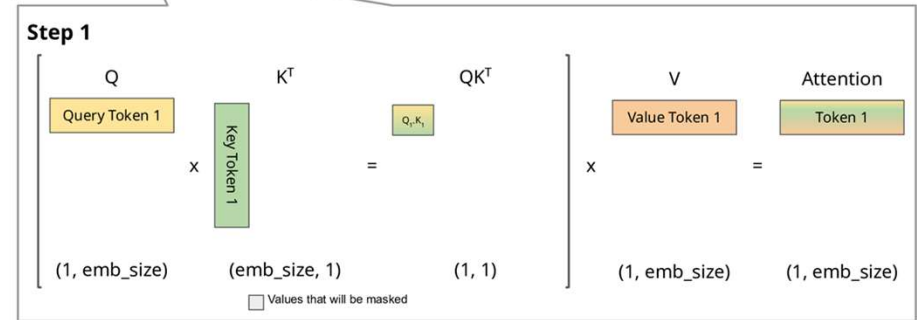
	THE	CAT	IS	ON	A	CHAIR
THE	0.268	-∞	-∞	-∞	-∞	-∞
CAT	0.124	0.278	-∞	-∞	-∞	-∞
IS	0.147	0.132	0.262	-∞	-∞	-∞
ON	0.210	0.128	0.206	0.212	-∞	-∞
A	0.146	0.158	0.152	0.143	0.227	-∞
CHAIR	0.195	0.114	0.203	0.103	0.157	0.229

(6, 6)



Time: $O(N^2*d) + O(N^2) + O(N^2*d) = O(N^2*d)$

Space: $O(3*N*d) + O(N^2) + O(N*d) = O(N^2 + N*d)$
 $\approx (3*N*d + N^2) \times \text{Size of a float}$



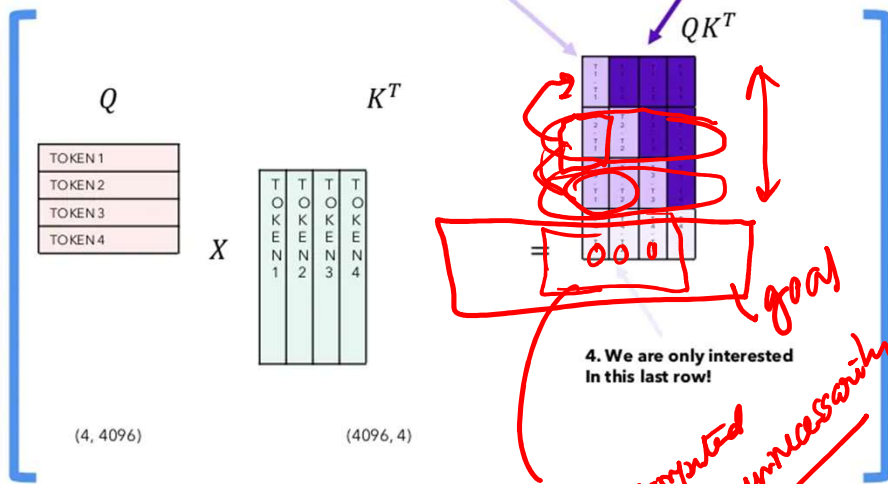
Zoom-in! (simplified without Scale and Softmax)



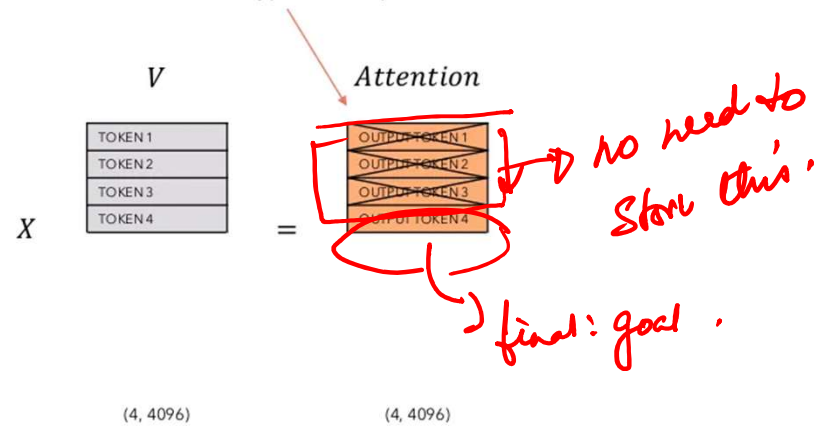
Why do we need to do better?

1. We already computed these dot products in the previous steps. **Can we cache them?**

2. Since the model is causal, **we don't care about the attention of a token with its successors**, but only with the tokens before it.



3. We don't care about these, as we want to predict the next token and we already predicted the previous ones.

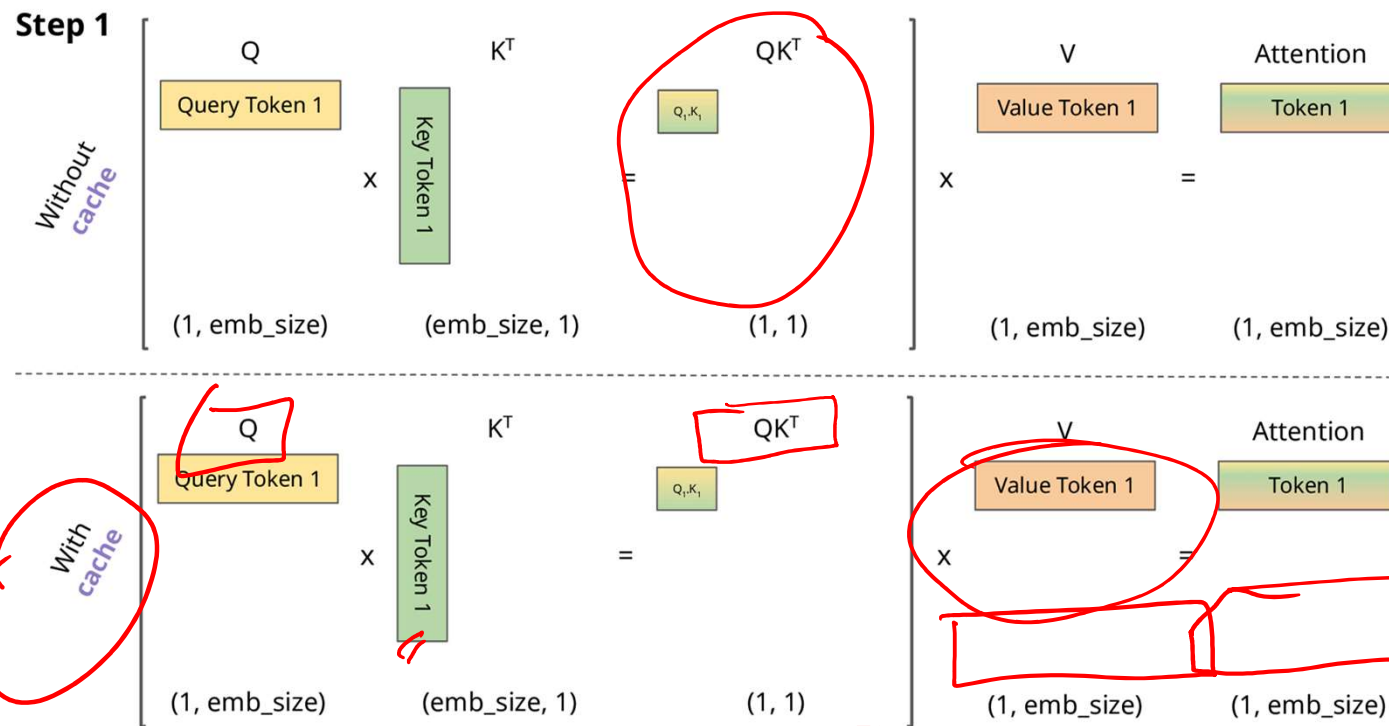


$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Inference
T = 4



KV Cache based (Forward Masked) Attention



KV Cache Storage = $O(N*d)$

$(2*N*d) \times \text{Size of a float}$

vs.

$(3*N*d + N^2) \times \text{Size of a float}$



KV Cache!

With cache

GOAL

Source: <https://medium.com/@joalages/kv-caching-explained-276520203249>

□ Values that will be masked ■ Values that will be taken from cache



Sourish Dasgupta

LLMs: Advanced Attention Mechanisms

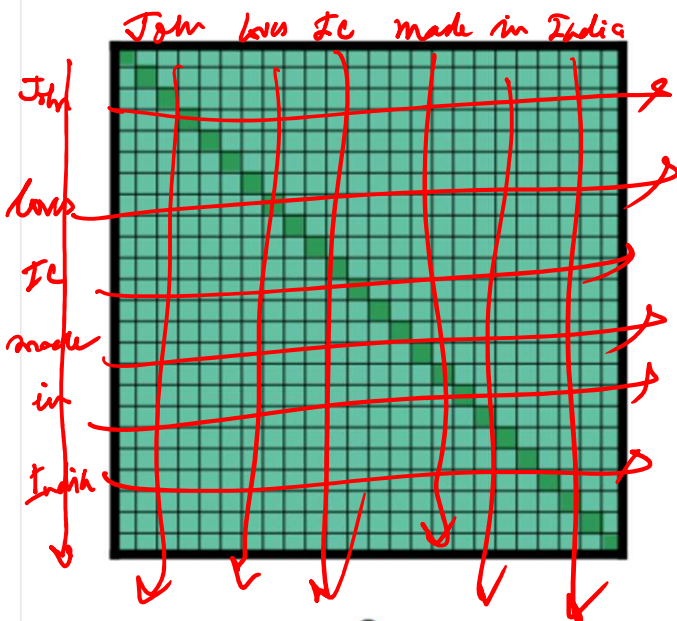
Year: 2020, Arxiv



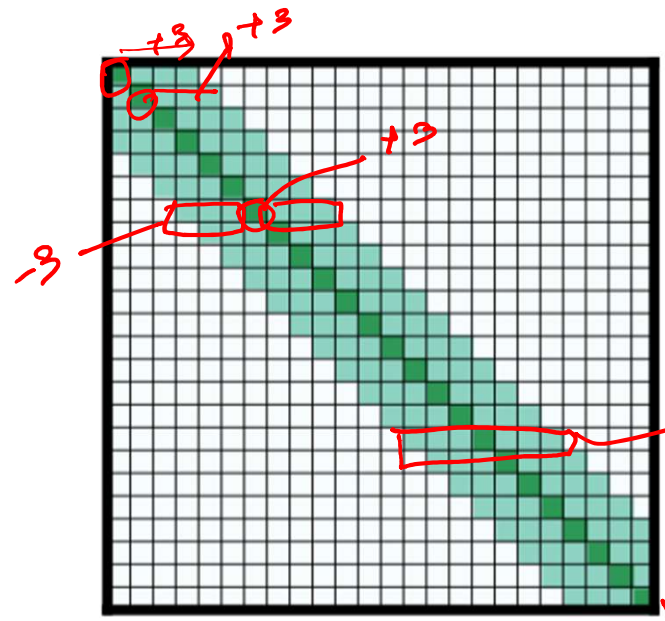
Sliding Window Attention

Any LM (not just causal)

Window size = 29 (4, 5, 6)



(a) Full n^2 attention



(b) Sliding window attention

Context window

w = 7

apply to causal LM.

get information from the future. → attention

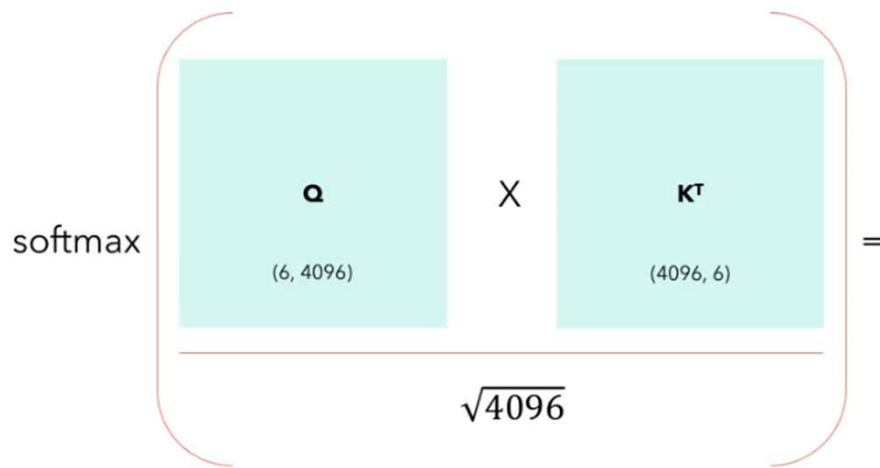


Year: 2020, Arxiv



Sliding Window Attention

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



	THE	CAT	IS	ON	A	CHAIR
THE	1.0	0	0	0	0	0
CAT	0.461	0.538	0	0	0	0
IS	0.3219	0.317	0.361	0	0	0
ON	0	0.316	0.341	0.343	0	0
A	0	0	0.326	0.323	0.351	0
CHAIR	0	0	0	0.313	0.331	0.356

$w = 3$



What happens to the KV Cache?

The motivation

Since our sliding window size is 4, we only want the dot-product of the current token with the previous 4 (including the token itself)

We don't care about these either, as we want the output token to only depend on the previous 4 tokens!

$w=4$

QK^T

Q (1, 4096)

K^T (4096, 8)

X

Cache

$X \rightarrow 4$

do not take every one!

V (8, 4096)

Attention (1, 4096)

OUTPUT TOKEN 8

$(N \times w) \times d$

$(N \times w) \times d$

Time: $O(N^2 \times d) + O(N^2) + O(N^2 \times d) = O(N \times w \times d)$

Space: $O(3 \times N \times d) + O(N^2) + O(N \times d) = O(N \times w + N \times d)$

$(2 \times w \times d) \times \text{Size of a float}$

VS.

$(2 \times N \times d) \times \text{Size of a float}$

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

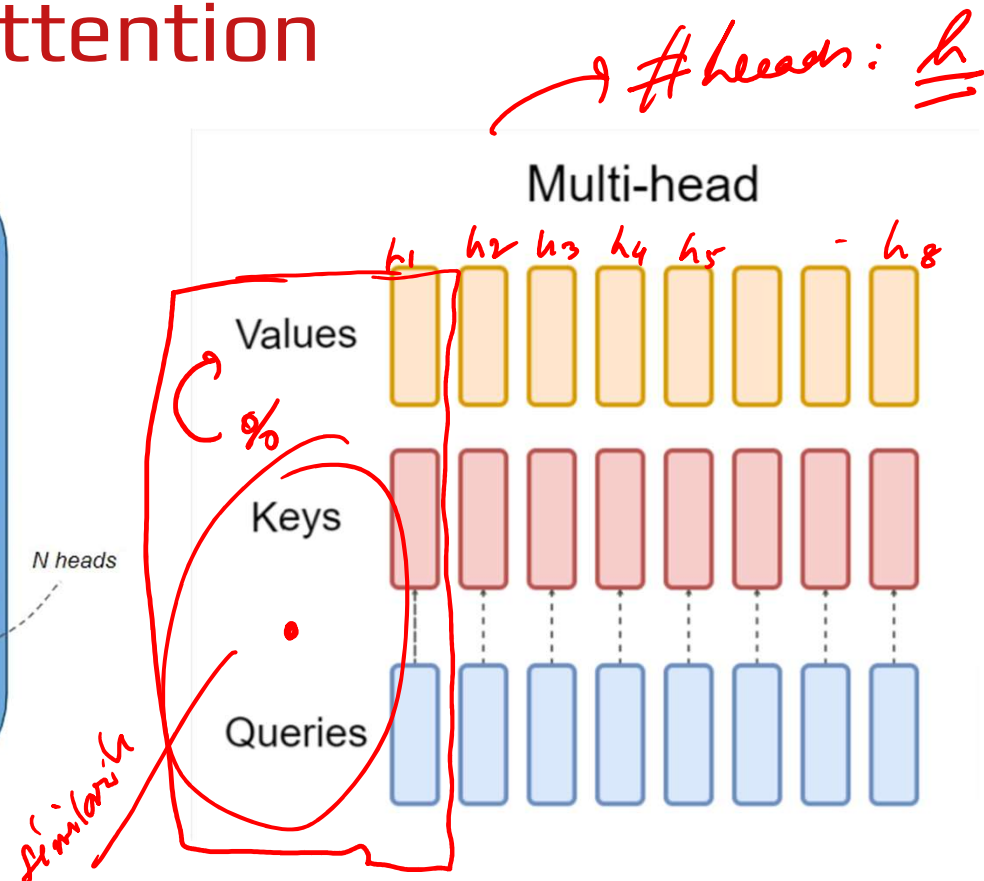
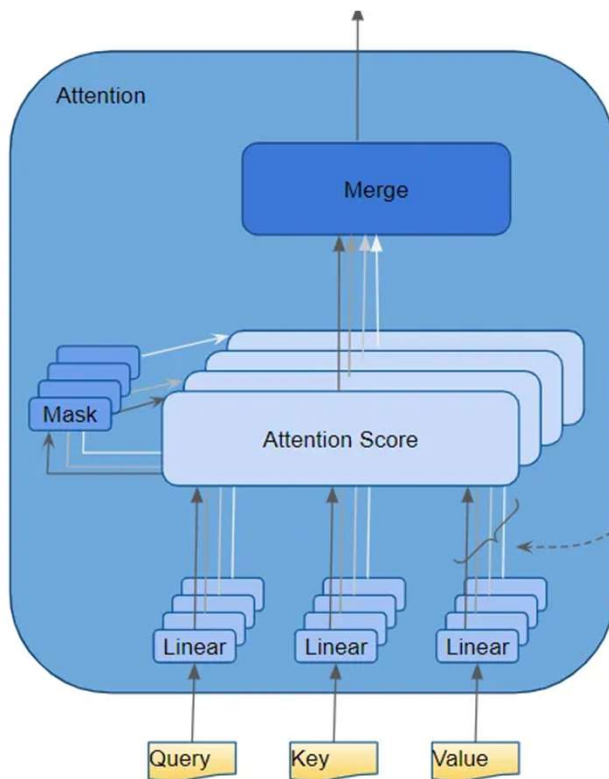
Inference
T = 4



Going back to year: 2017, NeurIPS



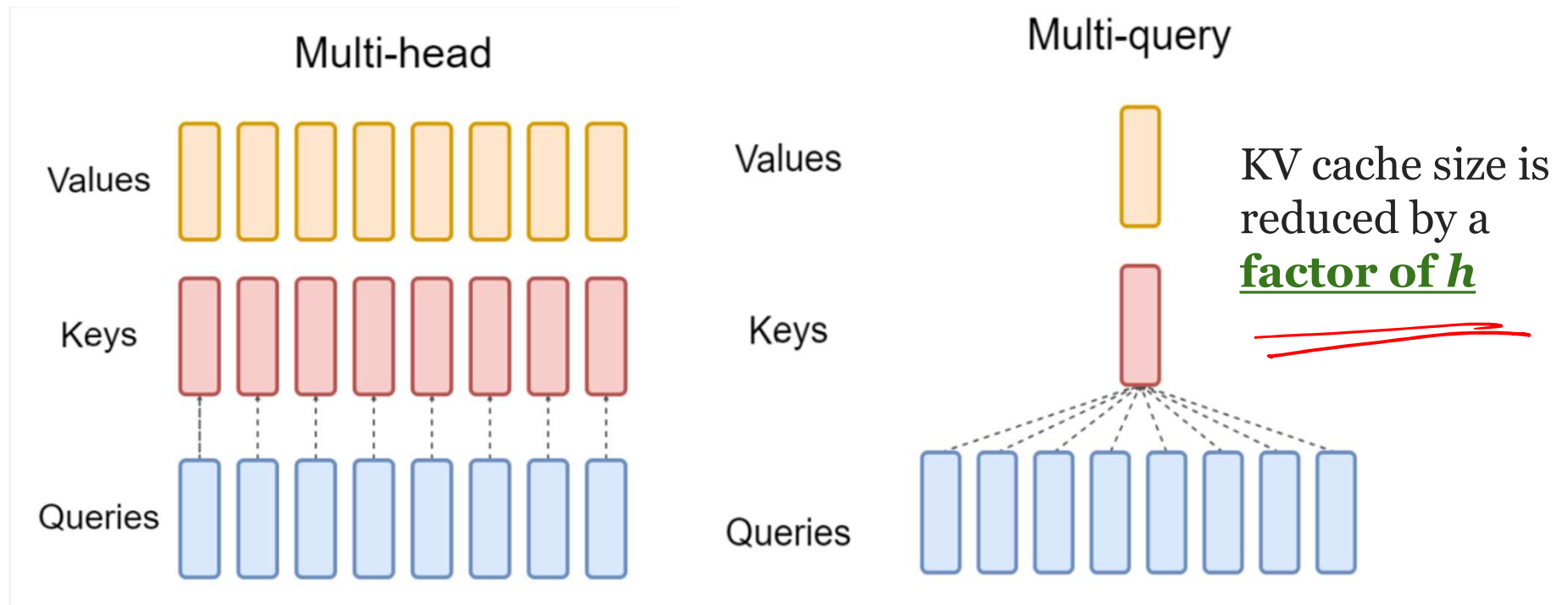
Multi-Head Self Attention



Year: 2019, arxiv



Multi-Query Attention (MQA)



Do we lose out on something?

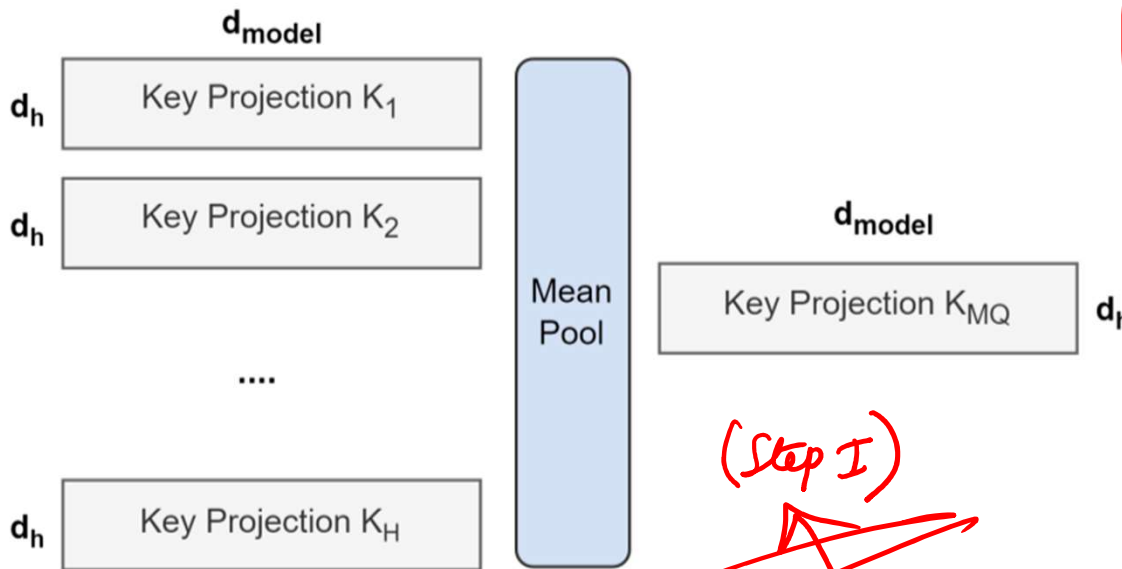
- Decline in performance quality
- Training instability



Year: 2023; ICLR

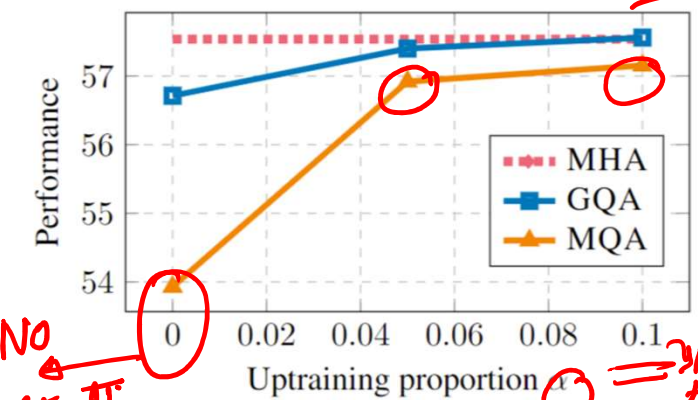


Uptraining: Converting MHA to MQA



(Step I)

gettin drained } k epochs -
 " further "
 [Step 2] - pre-train
 $k' \rightarrow k$



NO step II.
 STOP the train @ k' epoch.
 $\alpha = \frac{k-k'}{k}$



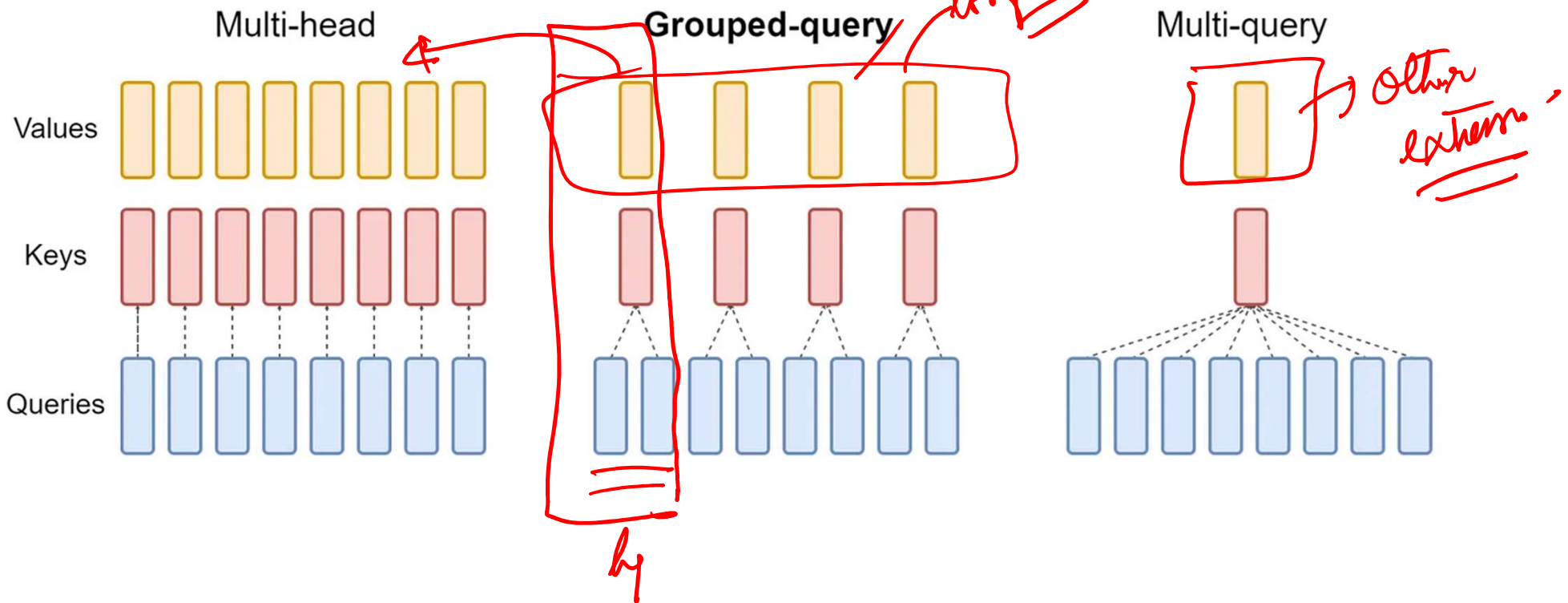
What can still go wrong?

- Decline in performance quality
- ~~Training instability~~





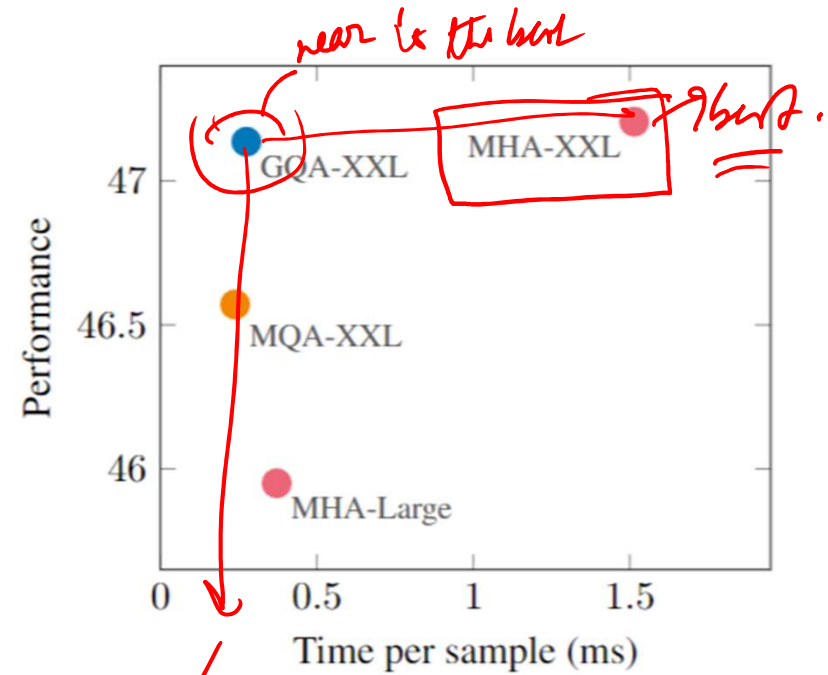
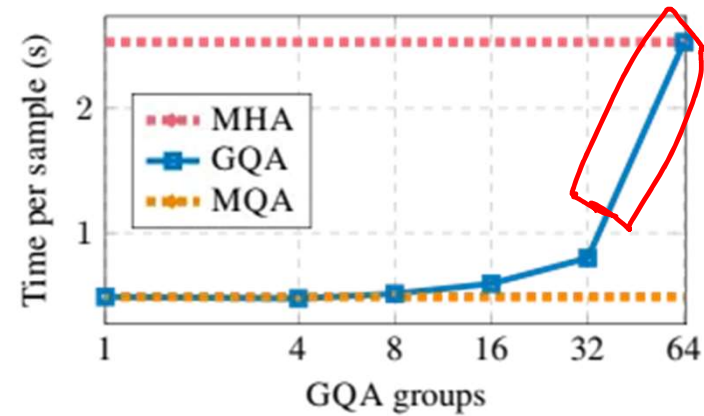
Grouped Query Attention



What did we gain?

inference Time

Model	T_{infer}	Average	CNN	arXiv	PubMed	MediaSum	MultiNews	WMT	TriviaQA
	s		R_1	R_1	R_1	R_1	R_1	BLEU	F1
MHA-Large	0.37	46.0	42.9	44.6	46.2	35.5	46.6	27.7	78.2
MHA-XXL	1.51	47.2	43.8	45.6	47.5	36.4	46.9	28.4	81.9
MQA-XXL	0.24	46.6	43.0	45.0	46.9	36.1	46.5	28.5	81.3
GQA-8-XXL	0.28	47.1	43.5	45.4	47.7	36.3	47.2	28.4	81.6



best performance



So are we all set? Key

- GQA/MQA Aim: *To reduce the need for storing a large amount of KV cache*
- LLM server can handle more requests, larger batch sizes and increased throughput
- *Cannot significantly reduce the computational load*
- *Quality degradation remains*

BUT
~~***~~

