



AILab

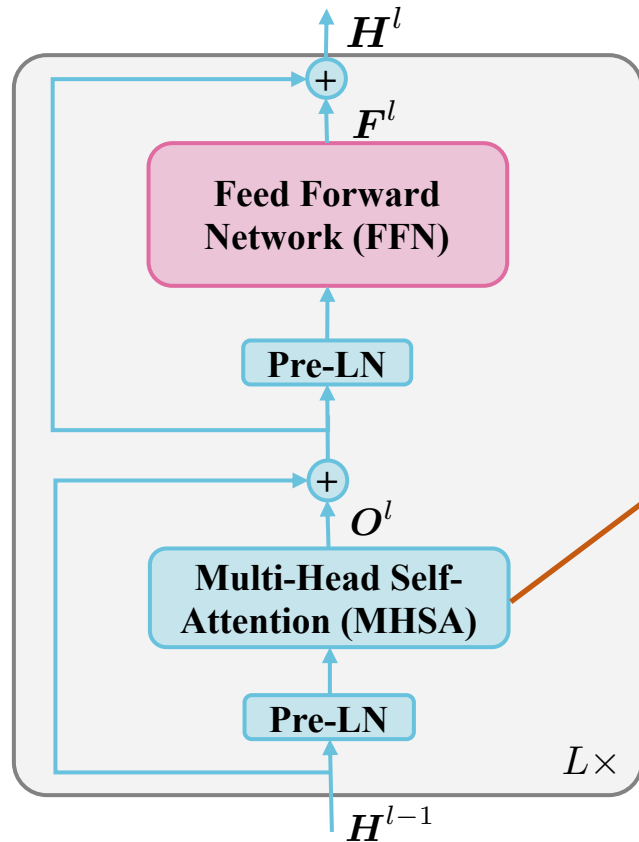


When Attention Sink Emerges in Language Models: An Empirical View

Speaker: Xiangming Gu

What is attention sink?

- Decoder-only Transformer

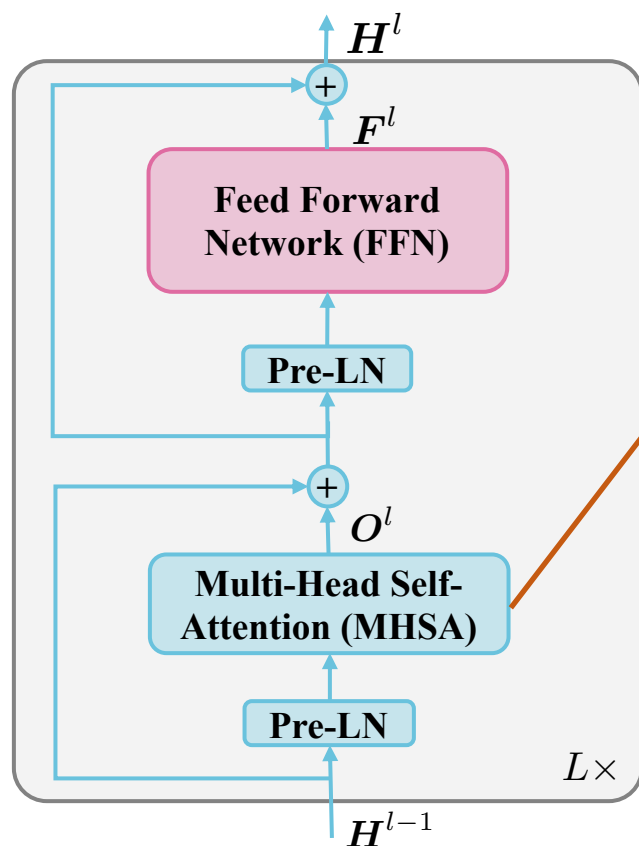


Self-attention is one of the most important part

$$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \underset{\substack{\uparrow \\ \text{queries}}}{Q^{l,h}} \underset{\substack{\uparrow \\ \text{keys}}}{K^{l,h}}^\top + \underset{\substack{\uparrow \\ \text{casual mask}}}{M} \right) \underset{\substack{\uparrow \\ \text{values}}}{V^{l,h}}$$

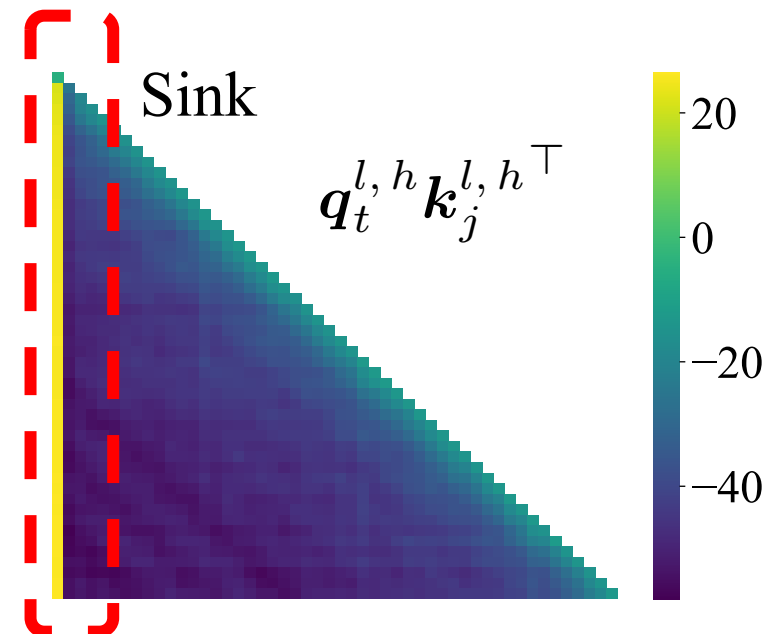
What is attention sink?

- Decoder-only Transformer



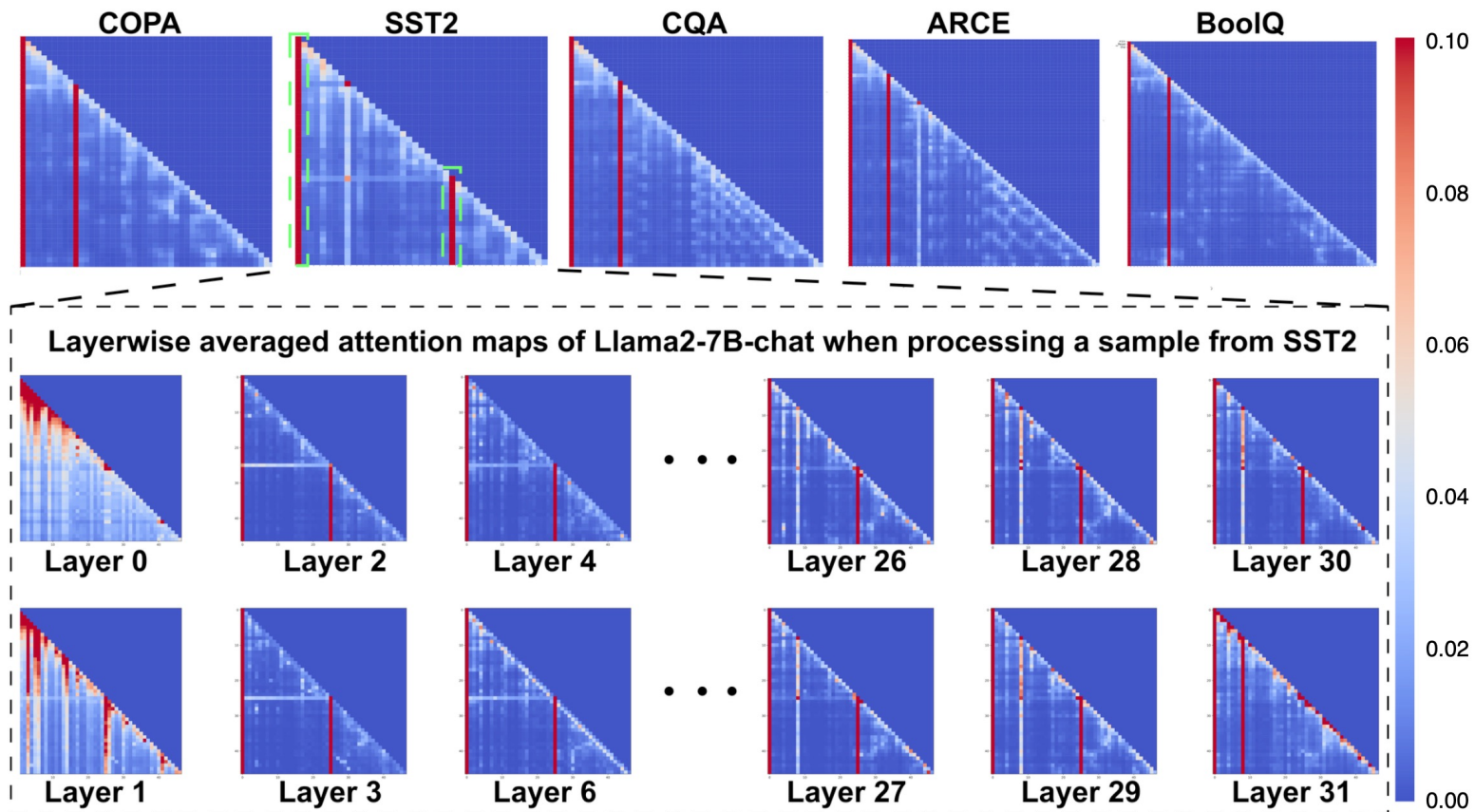
attention sink!

$$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} Q^{l,h} K^{l,h \top} + M \right) V^{l,h}$$



What is attention sink?

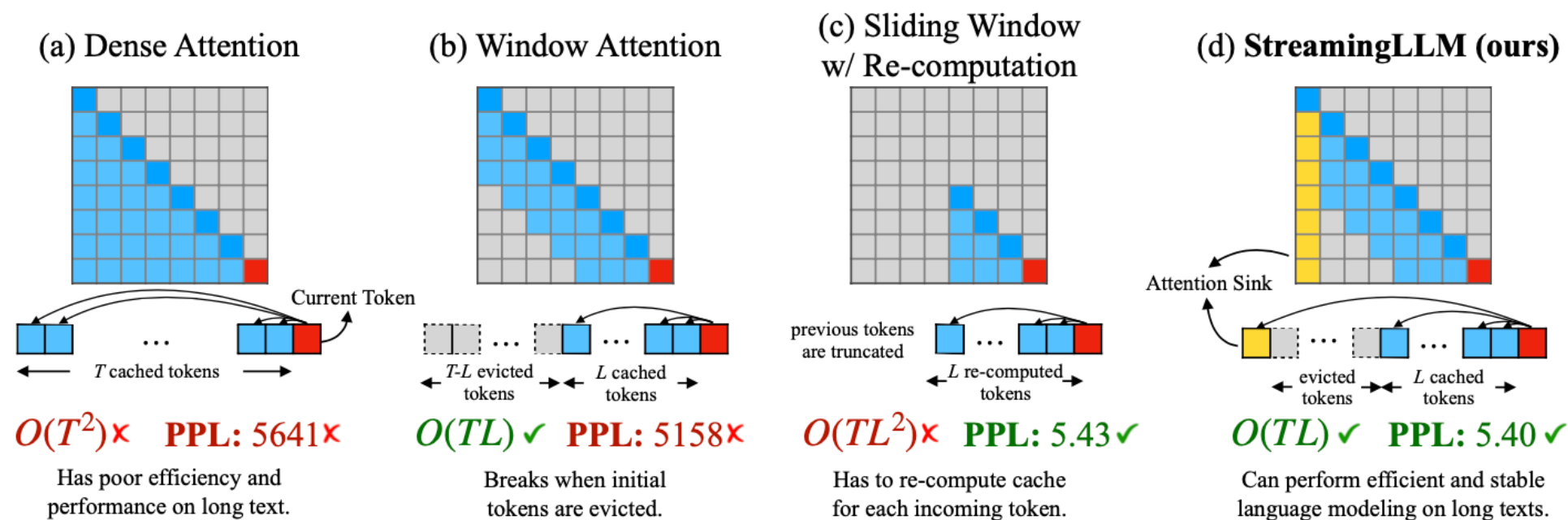
- In some cases, specific tokens may become sink tokens (Yu et al. 2024)



- No fixed positions
- Different LLMs may have different sets for these specific sink tokens

What can we do with attention sink?

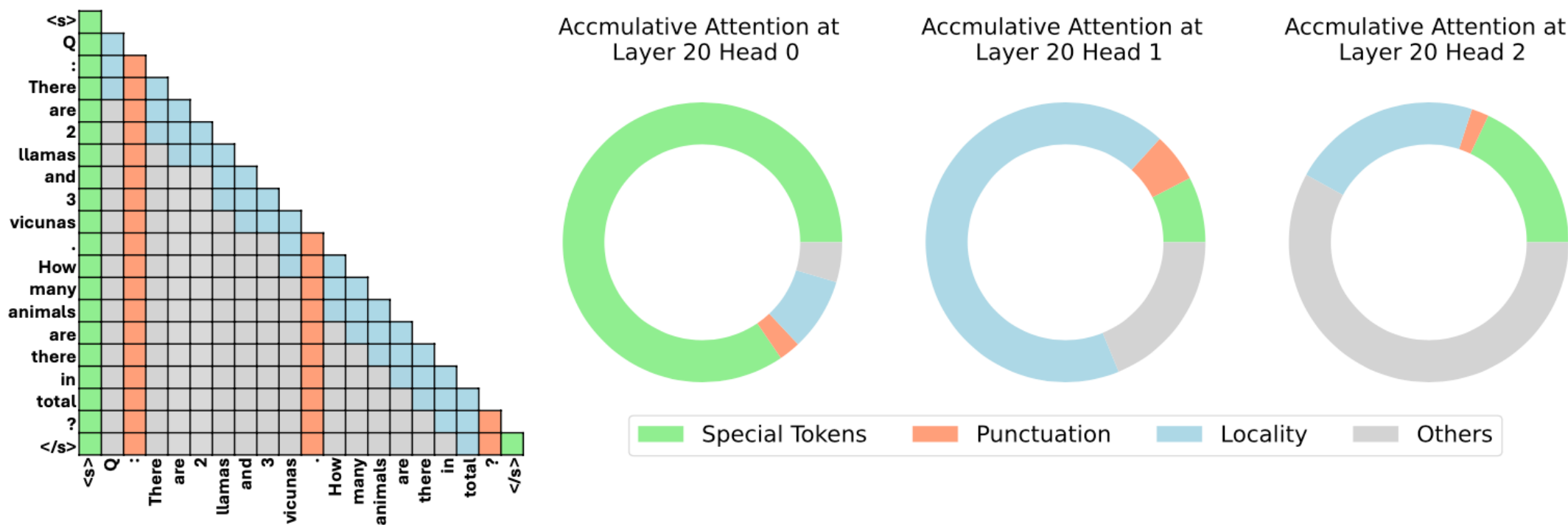
- Long context understanding / generation by only computing the attention on the sink token and recent tokens (Xiao et al. 2024)



Xiao et al. Efficient Streaming Language Models with Attention Sinks. ICLR 2024

What can we do with attention sink?

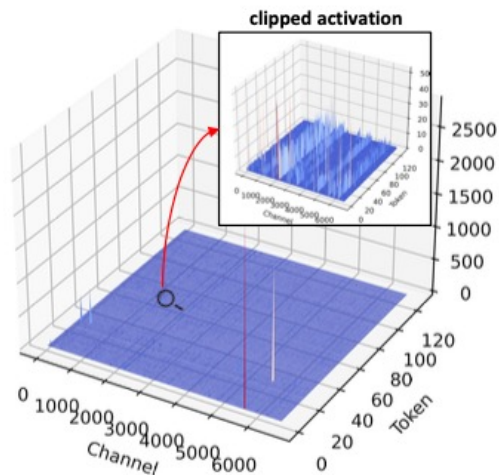
- **KV cache compression** by only constructing the KV cache of special tokens (including sink tokens) and recent tokens (Ge et al. 2024)



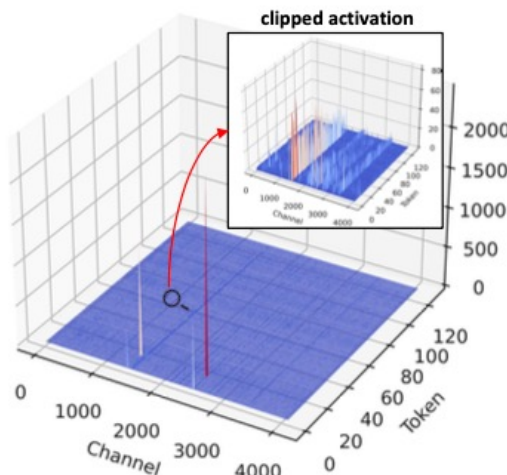
Ge et al. Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. ICLR 2024

What can we do with attention sink?

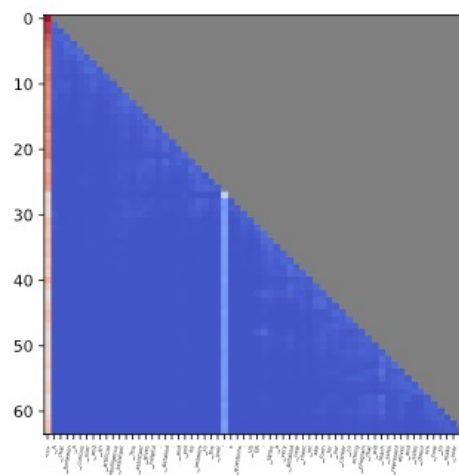
- **Model quantization** by preserving the KV cache of sink tokens with full precision (Liu et al. 2024)



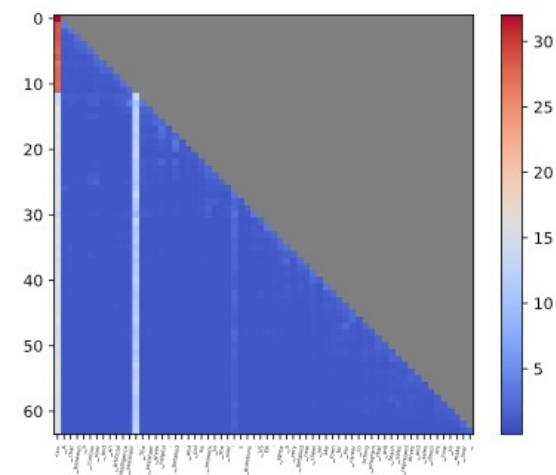
(a) Output activations of LLaMA-30B Layer 24



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



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

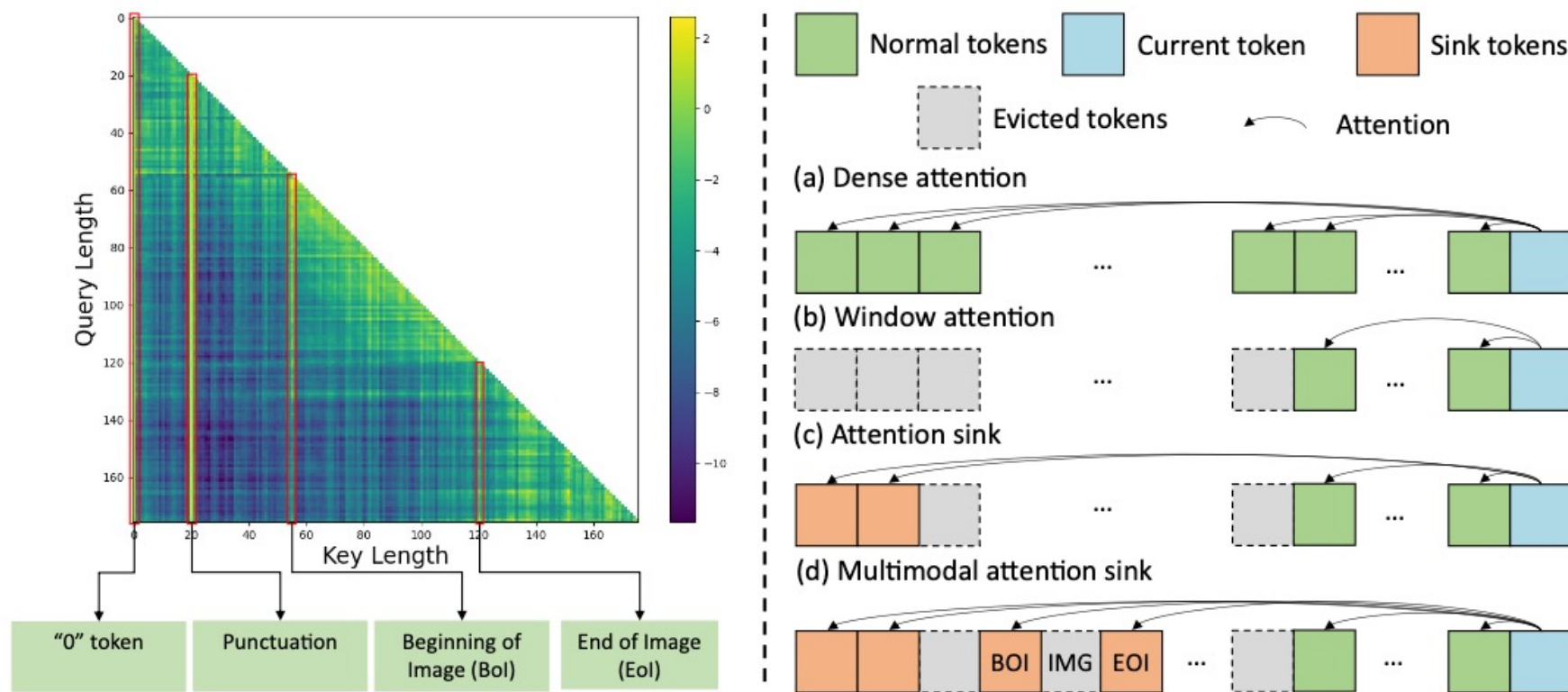


(d) Attention map of LLaMA-2-7B Layer 24

Liu et al. IntactKV: Improving Large Language Model Quantization by Keeping Pivot Tokens Intact. ACL Findings 2024

What can we do with attention sink?

- Multi-model language modeling by considering attention sink (Yang et al. 2024)



Yang et al. SEED-Story: Multimodal Long Story Generation with Large Language Model. Arxiv 2024

Main motivations in this talk

- Attention sink is important due to above applications
- Big questions:

How to understand attention sink?

When attention sink appears in LLMs?

Why LLMs need attention sink?

Main motivations in this talk

- Attention sink is important due to above applications
- Big questions:

How to understand attention sink?

When attention sink appears in LLMs?

Why LLMs need attention sink?

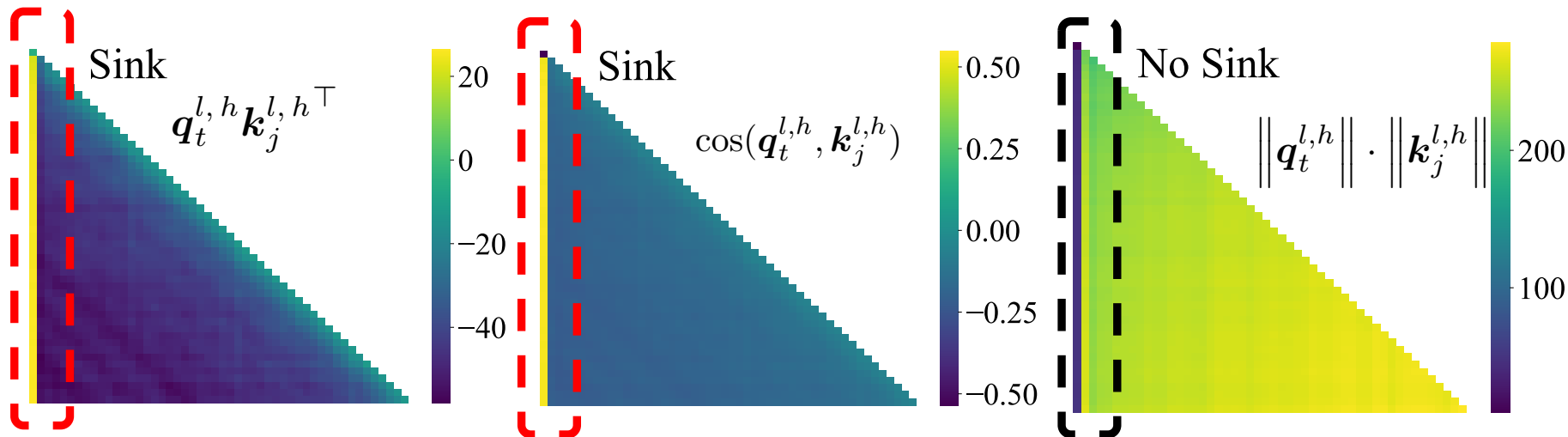
Looking into the internals in LLMs

- We find that QK angle matters for attention sink by decomposing QK

Attention sink

QK angle

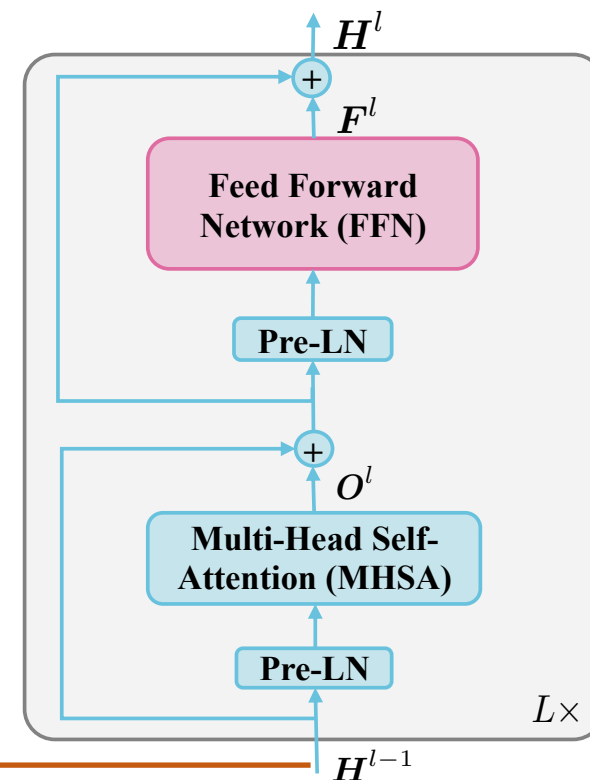
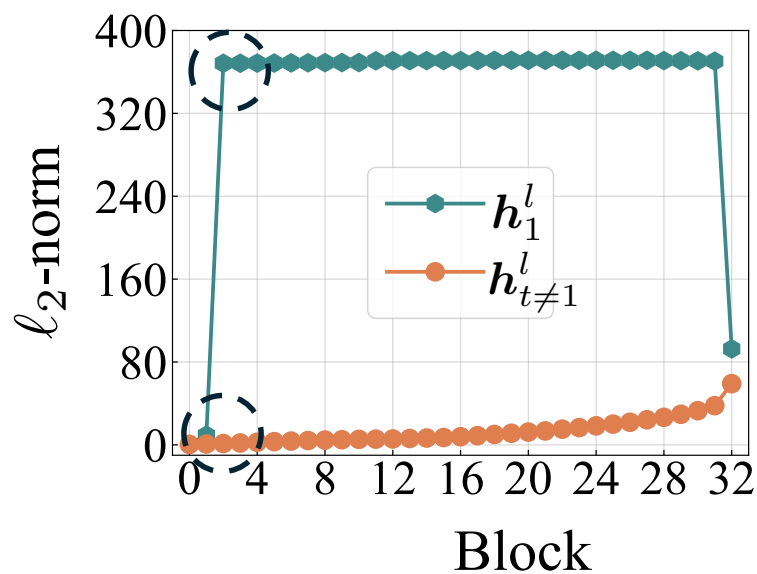
$$\begin{aligned} \mathbf{q}_t^{l,h} \mathbf{k}_1^{l,h\top} &\gg \mathbf{q}_t^{l,h} \mathbf{k}_{j \neq 1}^{l,h\top} \\ \cos(\mathbf{q}_t^{l,h}, \mathbf{k}_1^{l,h}) &\gg \cos(\mathbf{q}_t^{l,h}, \mathbf{k}_{j \neq 1}^{l,h}) \end{aligned}$$



Key of the sink token is distributed in a different manifold

Massive activations in LLMs

- Massive activations in hidden states of sink token: its L2-norm is significantly larger than that of other tokens (Cancedda 2024; Sun et al. 2024)

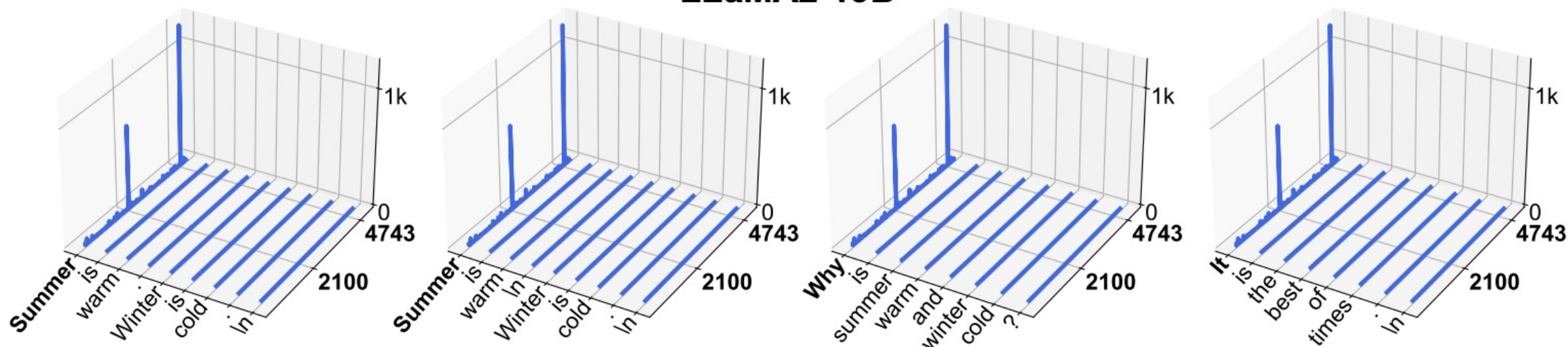


Cancedda, Nicola. Spectral filters, dark signals, and attention sinks. ACL 2024
Sun et al. Massive activations in large language models. COLM 2024

Massive activations in LLMs

- There are spikes in very few dimensions in massive activations

LLaMA2-13B



Sun et al. Massive activations in large language models. COLM 2024

How to connect attention sink?

- Massive activations after LN?

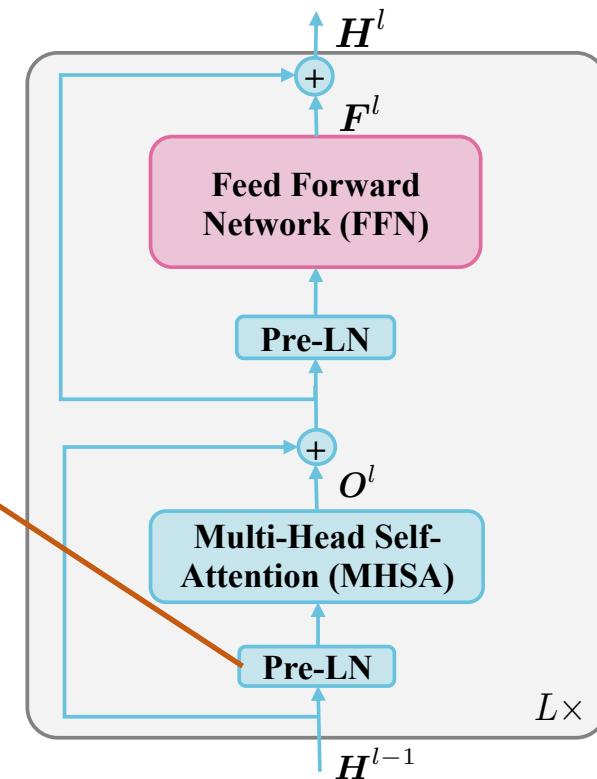
Layer norm retains spikes for specific dimensions and suppress other dimensions

$$\text{LN}(\mathbf{h}) = \frac{\mathbf{h}}{\sqrt{\frac{1}{d} \sum_{i=1}^d h_i^2}} \odot \mathbf{g}$$

Hidden states

Learnable gain parameters

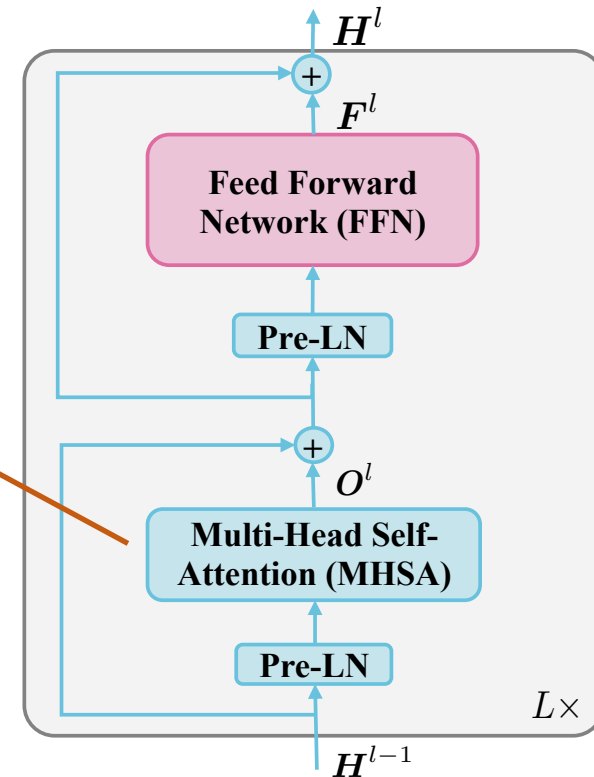
Norm is large



How to connect attention sink?

- The spikes in massive activations construct the manifold for the first key through **linear transformations**:
- Spikes \rightarrow Gain parameters in LN $\rightarrow W_k \rightarrow$ Rotation

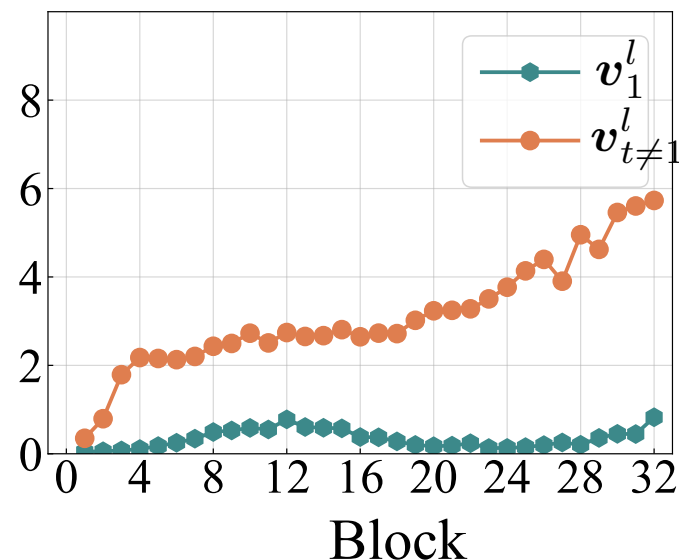
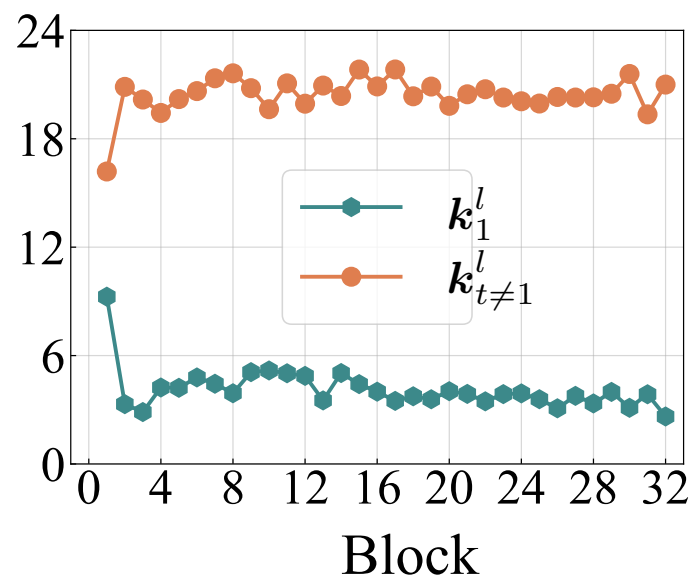
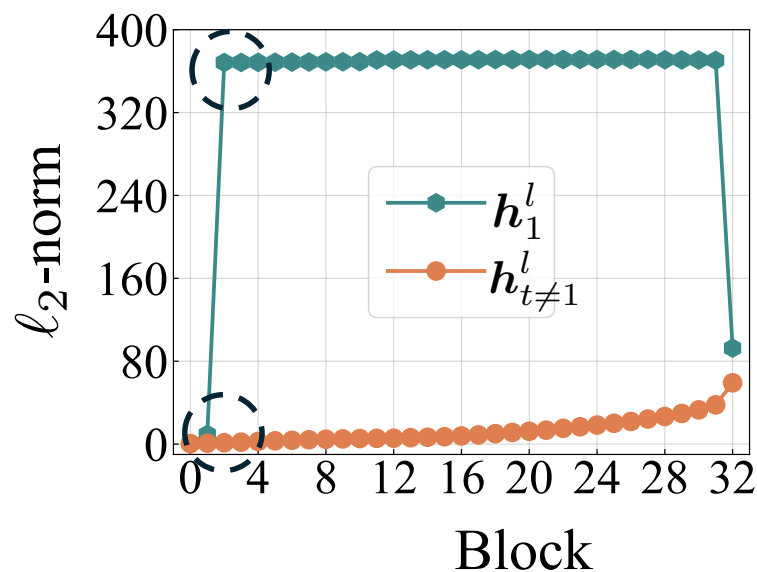
$$k_t^{l,h} = \text{LN}(h_t^{l-1}) W_K^{l,h} R_{\Theta, -t}$$



Other specific properties of attention sink?

- Values for the first token are small in L2 norm

$$\mathbf{v}_t^{l,h} = \text{LN}(\mathbf{h}_t^{l-1}) \mathbf{W}_V^{l,h} = \frac{\mathbf{h}}{\sqrt{\frac{1}{d} \sum_{i=1}^d \mathbf{h}_i^2}} \text{diag}(\mathbf{g}) \mathbf{W}_V^{l,h}$$



How LLMs learn the massive activations

- Attention sink always happen in the first token
- Uniqueness of the first token:

The calculation of its hidden states is not involved of self-attention

So LLMs learn to map the first token (the first token can vary, but limited to the size of vocab) to massive activations

What does attention sink head do?

Facts about attention sink:

- Attention on the first token is very large
- V of the first token is very small

Attention sink head is doing nothing?

$$\begin{aligned}\text{Attention output} &= \text{Attention.dot}(V) \\ &= \text{Attention_I} * V_I + \text{Attention_other} * V_other\end{aligned}$$

Small



Main motivations in this talk

- Attention sink is important due to above applications
- Big questions:

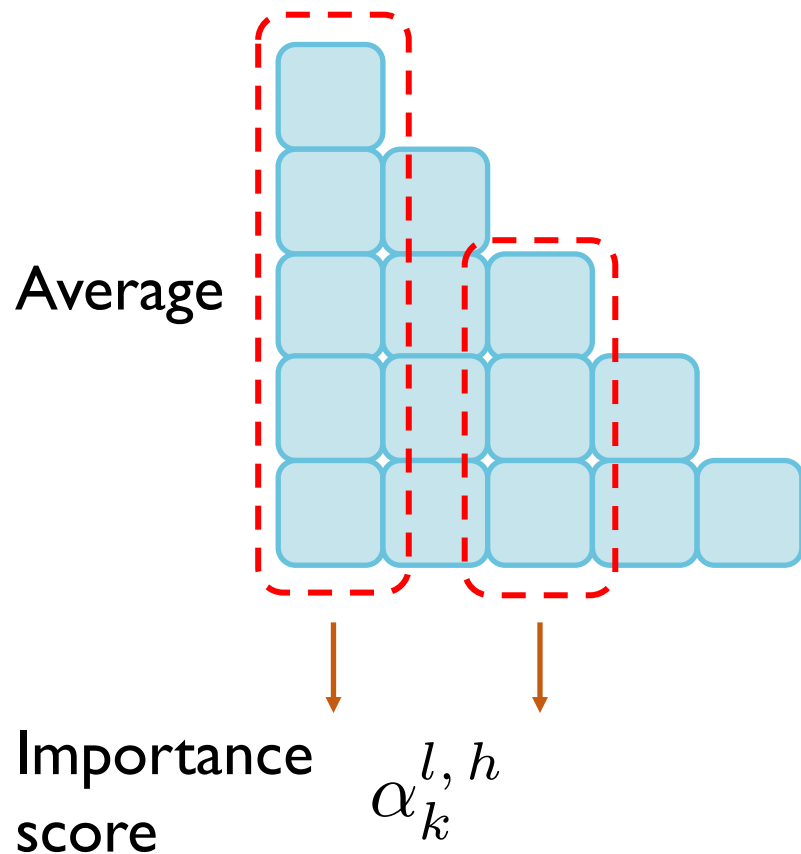
How to understand attention sink?

When attention sink appears in LLMs?

Why LLMs need attention sink?

How to measure attention sink?

- Attention scores of the first token are significantly larger than others



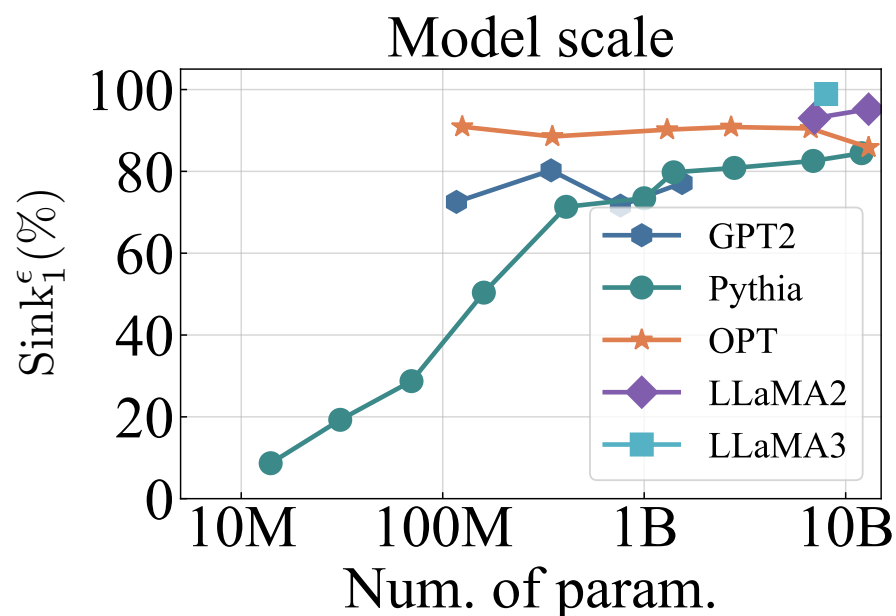
$$\text{Sink}_k^\epsilon = \frac{1}{L} \sum_{l=1}^L \frac{1}{H} \sum_{h=1}^H \mathbb{I}(\alpha_k^{l,h} > \epsilon)$$

Attention sink metric
of the whole LM

Within a head, a threshold
to decide a sink

When attention sink appears using diff. open-sourced LMs?

- Attention sink appears widespread in various LMs, even in LMs with 14M params.



- Attention sink emerges in LM pre-training

LLM	Sink ₁ ^ε (%)	
	Base	Chat
Mistral-7B	97.49	88.34
LLaMA2-7B	92.47	92.88
LLaMA2-13B	91.69	90.94
LLaMA3-8B	99.02	98.85

- Attention sink even appears in Jamba models (mix Mamba and attention)

When attention sink appears in diff. input?

- Attention sink appears with / without BOS (**for most LLMs**), even appears under random tokens
- Under all the repeat token input?

LLM	Sink ₁ ^ε (%)		
	natural	random	repeat
GPT2-XL	77.00	70.29	62.28
Mistral-7B	97.49	75.21	0.00
LLaMA2-7B Base	92.47	90.13	0.00
LLaMA3-8B Base	99.02	91.23	0.00

- Models with NoPE / relative PE / ALiBi / Rotary have same hidden states while models with absolute / learnable PE do not

Impact of positional embeddings under repeated tokens

- For LLMs with NoPE / relative PE / ALiBi / Rotary

$$P = 0$$

Hidden states before transformer blocks:

$$h_t^0 = xW_E + P$$

Then

$$h_1^0 = h_2^0 = \dots = h_T^0$$

Using mathematical induction, we can prove

$$h_1^l = h_2^l = \dots = h_T^l, \quad \forall \quad 0 \leq l \leq L$$

Impact of positional embeddings under repeated tokens

- Closed form/upper bound for NoPE / relative PE / ALiBi / Rotary

Proposition 1. *For LMs with NoPE, the attention scores for t repeated tokens are t^{-1} uniformly, i.e., there is no attention sink.*

Proof. We have that

$$\mathbf{A}_{ti}^{l,h} = \frac{e^{\langle \mathbf{q}_t^{l,h}, \mathbf{k}_i^{l,h} \rangle}}{\sum_{j=1}^t e^{\langle \mathbf{q}_t^{l,h}, \mathbf{k}_j^{l,h} \rangle}} = \frac{e^{\mathbf{q}_t^{l,h} \mathbf{k}_i^{l,h \top}}}{\sum_{j=1}^t e^{\mathbf{q}_t^{l,h} \mathbf{k}_j^{l,h \top}}} = \frac{e^{\mathbf{q}^{l,h} \mathbf{k}^{l,h \top}}}{t e^{\mathbf{q}^{l,h} \mathbf{k}^{l,h \top}}} = \frac{1}{t}. \quad (18)$$

Therefore, the attention scores follow a uniform distribution over all previous tokens. \square

Proposition 2. *For LMs with relative PE, there is no attention sink for t repeated tokens.*

Proof. For LMs with relative PE, the dot product between each query and key is

$$\langle \mathbf{q}_t^{l,h}, \mathbf{k}_i^{l,h} \rangle = \mathbf{q}_t^{l,h} \mathbf{k}_i^{l,h \top} + g_{\text{rel}}(t-i) = \mathbf{q}^{l,h} \mathbf{k}^{l,h \top} + g_{\text{rel}}(t-i), \quad (19)$$

then we have the attention scores

$$\mathbf{A}_{t,i}^{l,h} = \frac{e^{\langle \mathbf{q}_t^{l,h}, \mathbf{k}_i^{l,h} \rangle}}{\sum_{j=1}^t e^{\langle \mathbf{q}_t^{l,h}, \mathbf{k}_j^{l,h} \rangle}} = \frac{e^{\mathbf{q}^{l,h} \mathbf{k}^{l,h \top} + g_{\text{rel}}(t-i)}}{\sum_{j=1}^t e^{\mathbf{q}^{l,h} \mathbf{k}^{l,h \top} + g_{\text{rel}}(t-j)}} = \frac{e^{g_{\text{rel}}(t-i)}}{\sum_{j=1}^t e^{g_{\text{rel}}(t-j)}}. \quad (20)$$

Impact of positional embeddings under repeated tokens

- Closed form/upper bound for NoPE / relative PE / ALiBi / Rotary

Proposition 3. *For LMs with ALiBi, there is no attention sink for t repeated tokens.*

Proof. For LMs with ALiBi, similar to relative PE, the dot product between each query and key is

$$\langle \mathbf{q}_t^{l,h}, \mathbf{k}_i^{l,h} \rangle = \mathbf{q}_t^{l,h} \mathbf{k}_i^{l,h \top} + g_{\text{alibi}}^h(t - i) = \mathbf{q}_t^{l,h} \mathbf{k}^{l,h \top} + g_{\text{alibi}}^h(t - i), \quad (21)$$

then we have the attention scores

$$\mathbf{A}_{t,i}^{l,h} = \frac{e^{\langle \mathbf{q}_t^{l,h}, \mathbf{k}_i^{l,h} \rangle}}{\sum_{j=1}^t e^{\langle \mathbf{q}_t^{l,h}, \mathbf{k}_j^{l,h} \rangle}} = \frac{e^{\mathbf{q}_t^{l,h} \mathbf{k}^{l,h \top} + g_{\text{alibi}}^h(t-i)}}{\sum_{j=1}^t e^{\mathbf{q}_t^{l,h} \mathbf{k}^{l,h \top} + g_{\text{alibi}}^h(t-j)}} = \frac{e^{g_{\text{alibi}}^h(t-i)}}{\sum_{j=1}^t e^{g_{\text{alibi}}^h(t-j)}}. \quad (22)$$

Here $g_{\text{alibi}}^h(t - i)$ is monotonic decreasing function of $t - i$, so there is no attention sink on the first token. \square

Impact of positional embeddings under repeated tokens

- Closed form/upper bound for NoPE / relative PE / ALiBi / Rotary

Proof. For LMs with Rotary, the dot product between each query and key is

$$\langle \mathbf{q}_t^{l,h}, \mathbf{k}_i^{l,h} \rangle = \mathbf{q}_t^{l,h} \mathbf{R}_{\Theta, i-t} \mathbf{k}_i^{l,h \top} \quad (23)$$

$$= \mathbf{q}_t^{l,h} \mathbf{R}_{\Theta, i-t} \mathbf{k}^{l,h \top} \quad (24)$$

$$= \|\mathbf{q}_t^{l,h}\| \|\mathbf{k}^{l,h} \mathbf{R}_{\Theta, i-t}\| \cos \left(\frac{\mathbf{q}_t^{l,h} \mathbf{R}_{\Theta, i-t} \mathbf{k}^{l,h \top}}{\|\mathbf{q}_t^{l,h}\| \|\mathbf{k}^{l,h} \mathbf{R}_{\Theta, i-t}\|} \right) \quad (25)$$

$$= \|\mathbf{q}_t^{l,h}\| \|\mathbf{k}^{l,h}\| \cos(\beta_{t-i}), \quad (26)$$

where β_{j-t} is the angle between the rotated query and the rotated key. Then the attention scores are

$$\mathbf{A}_{t,i}^{l,h} = \frac{e^{\langle \mathbf{q}_t^{l,h}, \mathbf{k}_i^{l,h} \rangle}}{\sum_{j=1}^t e^{\langle \mathbf{q}_t^{l,h}, \mathbf{k}_j^{l,h} \rangle}} = \frac{e^{\mathbf{q}_t^{l,h} \mathbf{R}_{\Theta, j-i} \mathbf{k}^{l,h \top}}}{\sum_{j=1}^t e^{\mathbf{q}_t^{l,h} \mathbf{R}_{\Theta, j-i} \mathbf{k}^{l,h \top}}} = \frac{e^{\|\mathbf{q}_t^{l,h}\| \|\mathbf{k}^{l,h}\| \cos(\beta_{t-i})}}{\sum_{j=1}^t e^{\|\mathbf{q}_t^{l,h}\| \|\mathbf{k}^{l,h}\| \cos(\beta_{t-j})}}. \quad (27)$$

Suppose the norm of multiplication for query and key $\|\mathbf{q}_t^{l,h}\| \|\mathbf{k}^{l,h}\| = \xi$. Considering $-1 \leq \cos(\beta_{t-j}) \leq 1$, then we have

$$\mathbf{A}_{t,i}^{l,h} = \frac{e^{\xi \cos(\beta_{t-i})}}{\sum_{j=1}^t e^{\xi \cos(\beta_{t-j})}} = \frac{1}{1 + \frac{\sum_{j \neq i} e^{\xi \cos(\beta_{t-j})}}{e^{\xi \cos(\beta_{t-i})}}} \leq \frac{e^{2\xi}}{e^{2\xi} + (t-1)} \quad (28)$$

Then the attention scores for each token are upper-bounded and decrease to 0 as t grows. \square

Attributing attention sink to LM pre-training

- LM pre-training objective $\min_{\theta} \mathbb{E}_{\mathbf{X} \sim p_{\text{data}}} [\mathcal{L}(p_{\theta}(\mathbf{X}))]$
- Experiments on LLaMA2-style models

Optimization

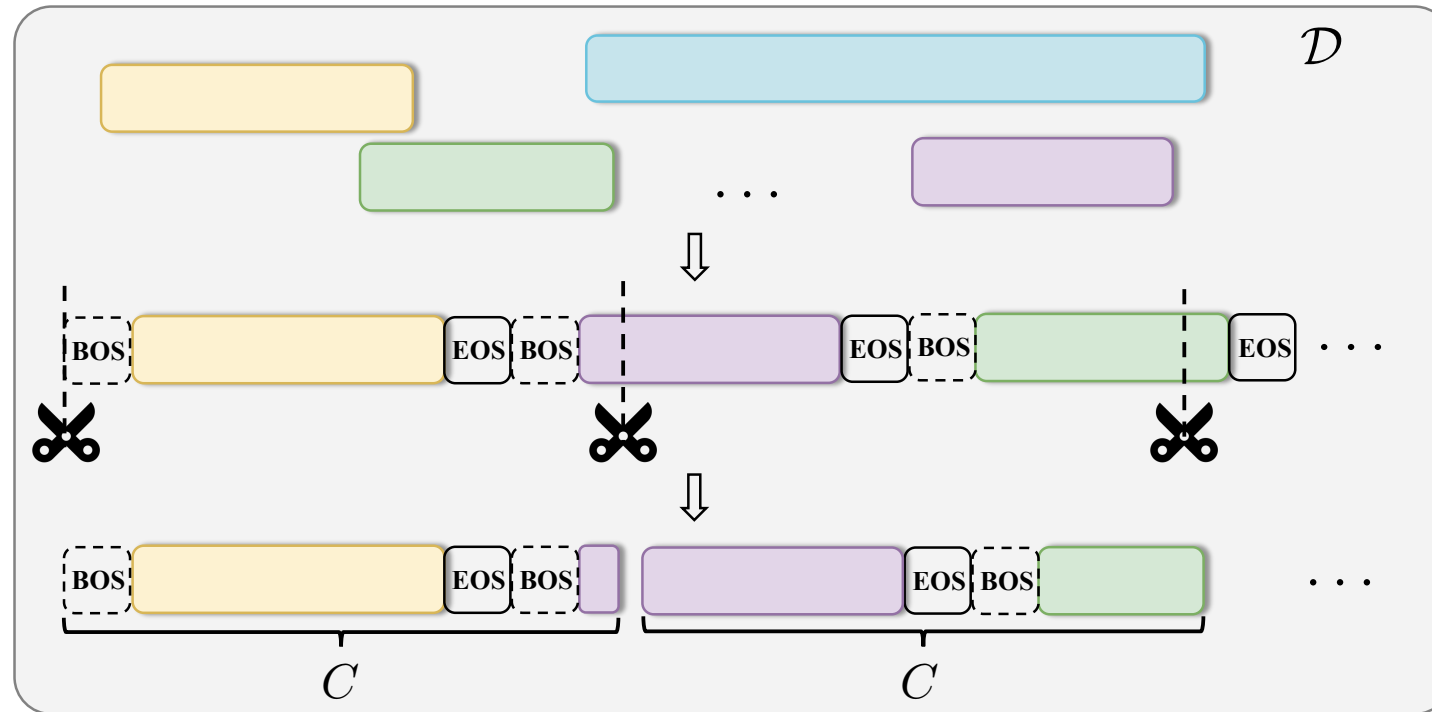
Data distribution

Loss function

Model architecture

A quick preliminary on LM pre-training

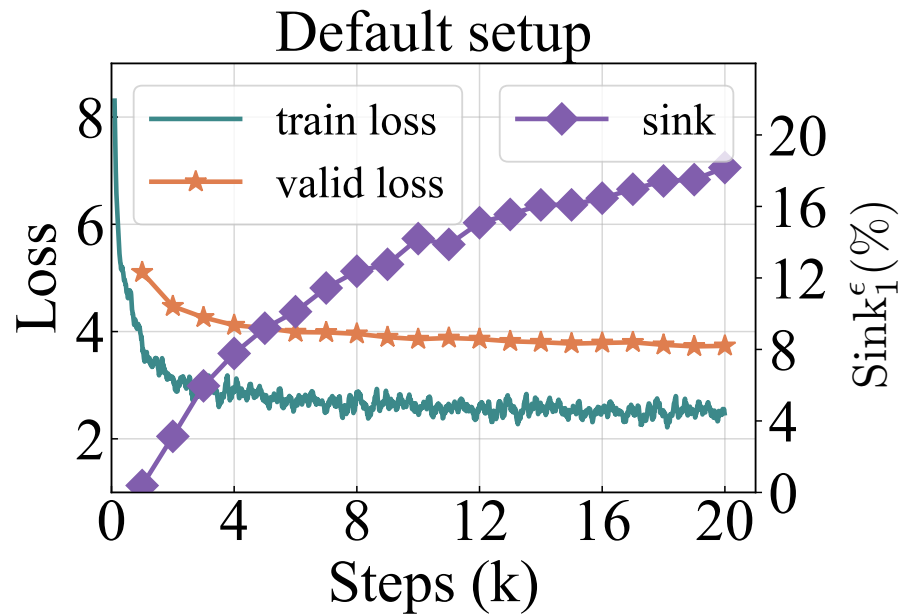
- Data packing strategy in LM pre-training



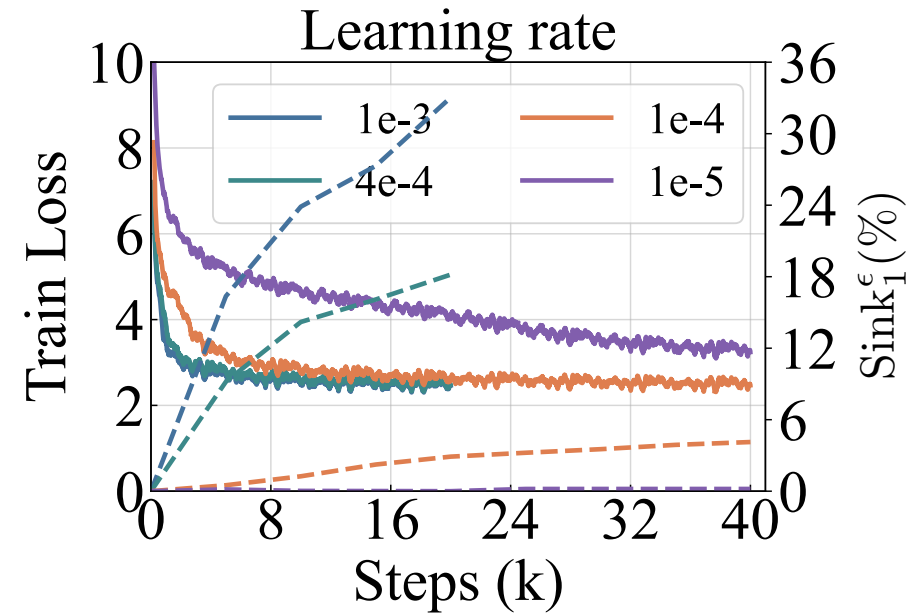
- In this case, adding BOS is optional, then $\text{BOS} = \text{EOS}$
- There could be any token in the first position of context window

Effects of optimization on attention sink

- Training steps



- Learning rate



Effects of optimization on attention sink

- Small learning rates not only slow down the emergence, but also mitigate attention sink

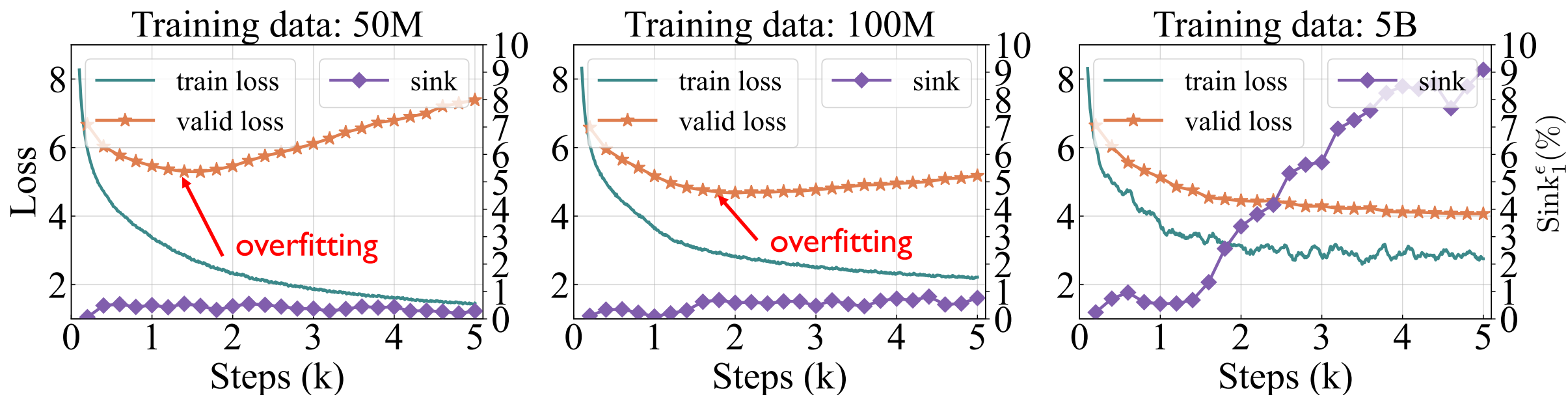
learning rate	training steps (k)	$\text{Sink}_1^\epsilon(\%)$	valid loss
8e-4	10	23.44	3.79
8e-4	20	32.23	3.70
4e-4	20	18.18	3.73
2e-4	20	11.21	3.78
2e-4	40	16.81	3.68
1e-4	20	2.90	3.92
1e-4	80	6.29	3.67

We keep the training steps x learning rate the same

Effects of data distribution on attention sink

- Unique training data amount

Attention sink emerges after LMs are trained on **sufficient unique training data**, not really related to **overfitting**



Effects of data distribution on attention sink

- Fix a token in the specific position of context window
- The fixed token will become the sink token

Fixed position	1	2	3
Sink_1^ϵ (%)	74.11	0.00	0.00
Sink_2^ϵ (%)	0.00	69.03	0.00
Sink_3^ϵ (%)	0.00	0.00	69.64
Sink_4^ϵ (%)	0.01	0.01	0.00

Effects of loss function on attention sink

- Auto-regressive loss

$$\mathcal{L} = \sum_{t=2}^C \log p_{\theta}(\mathbf{x}_t | \mathbf{x}_{<t})$$

- Weight decay

$$\mathcal{L} = \sum_{t=2}^C \log p_{\theta}(\mathbf{x}_t | \mathbf{x}_{<t}) + \gamma \|\theta\|_2^2$$

L2 regularization



Larger weight decay encourages attention sink

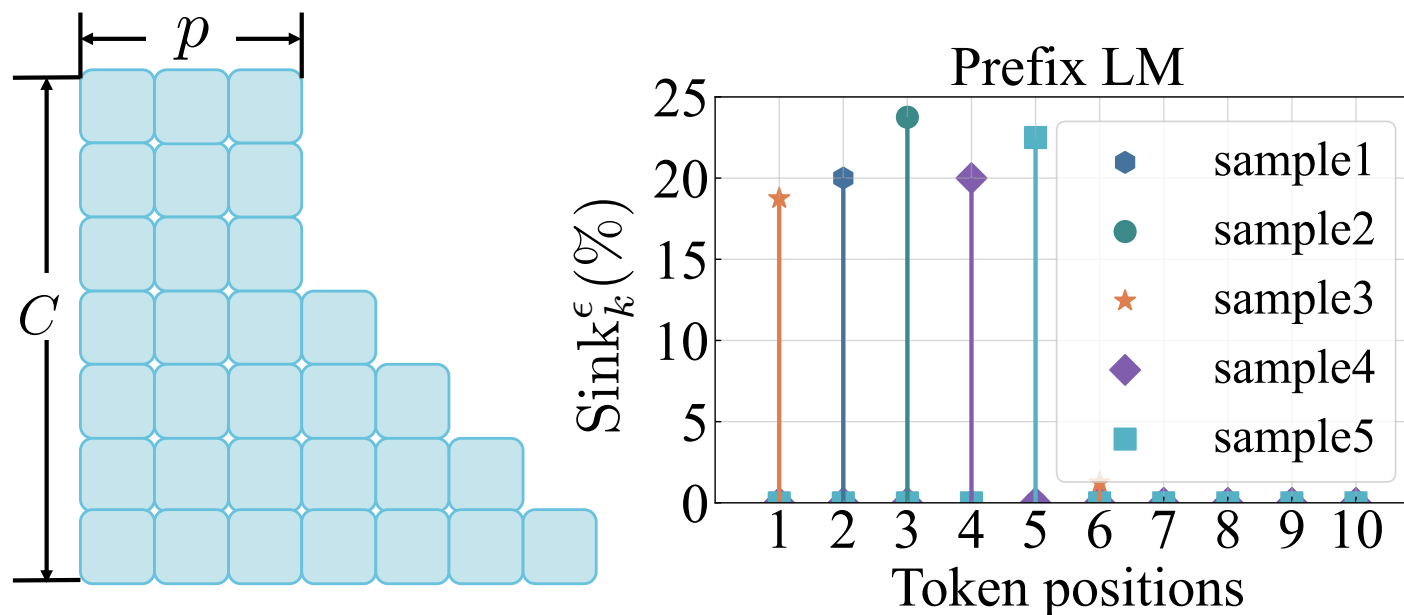
γ	0.0	0.001	0.01	0.1	0.5	1.0	2.0	5.0
Sink ₁ ^ε (%)	15.20	15.39	15.23	18.18	41.08	37.71	6.13	0.01
valid loss	3.72	3.72	3.72	3.73	3.80	3.90	4.23	5.24

Effects of loss function on attention sink

- Prefix language modeling $\mathcal{L} = \sum_{t=p+1}^C \log p_{\theta}(\mathbf{x}_t | \mathbf{x}_{p+1:t-1}, \mathbf{x}_{1:p})$

More prefix tokens

Sink token shifts from the first position to other positions within the prefix



Effects of loss function on attention sink

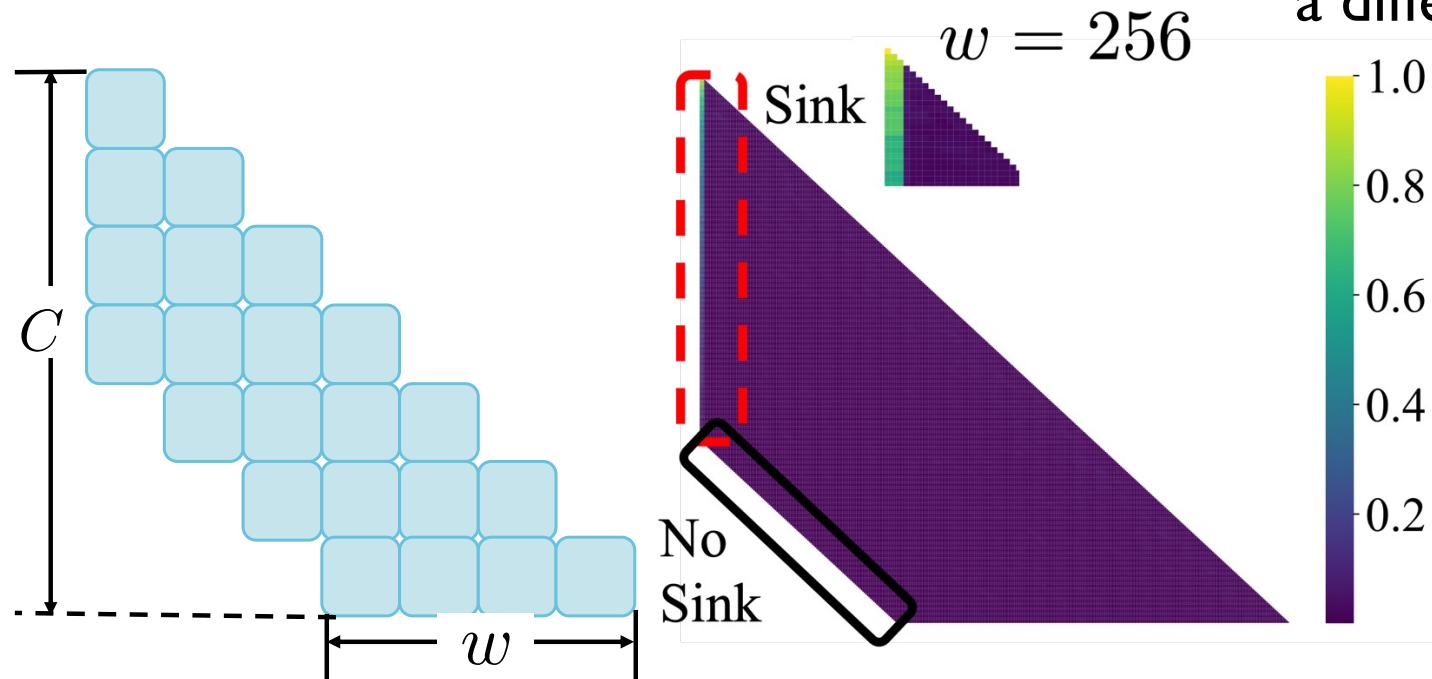
- Shifted window attention

$$\mathcal{L} = \sum_{t=2}^C \log p_{\theta}(\mathbf{x}_t | \mathbf{x}_{t-w:t-1})$$

Attention sink appears on the **absolute, not the relative** first token

Small window size mitigates attention sink

Key of the sink token is trained to be distributed in a different manifold



Effects of model architecture on attention sink

The following designs do not affect the emergence of attention sink

- Positional embeddings

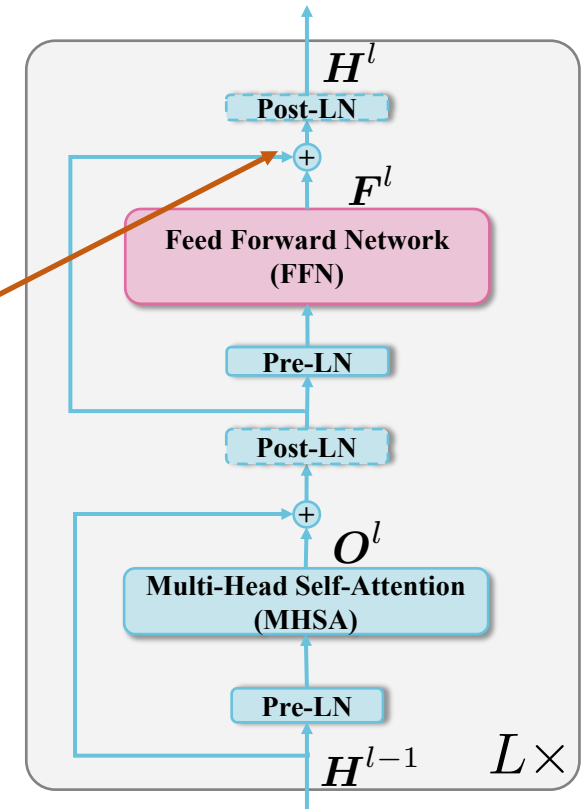
NOPE, learnable PEs, absolute PEs, relative PEs, rotary, alibi

Effects of model architecture on attention sink

The following designs do not affect the emergence of attention sink

- Positional embeddings
- Pre-norm and post-norm transformer block structure

Massive activations
happen before LNs



Effects of model architecture on attention sink

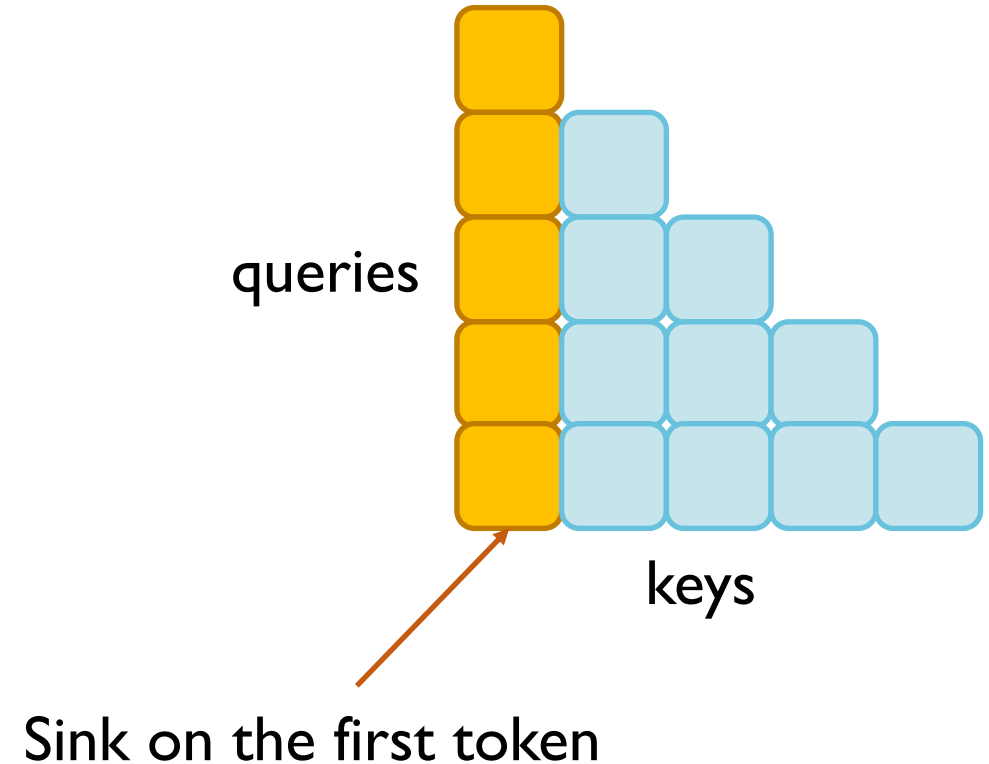
The following designs do not affect the emergence of attention sink

- Positional embeddings: including no positional embedding
- Pre-norm and post-norm transformer block structure
- Feed forward networks (FFNs) with different activation functions
- Number of attention heads, how to combine multiple heads

Effects of model architecture on attention sink

Standard softmax attention in h -th head l -th block

$$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \underset{\substack{\uparrow \\ \text{queries}}}{Q^{l,h}} \underset{\substack{\uparrow \\ \text{keys}}}{K^{l,h}}^\top + \underset{\substack{\uparrow \\ \text{casual mask}}}{M} \right) \underset{\substack{\uparrow \\ \text{values}}}{V^{l,h}}$$



Effects of model architecture on attention sink

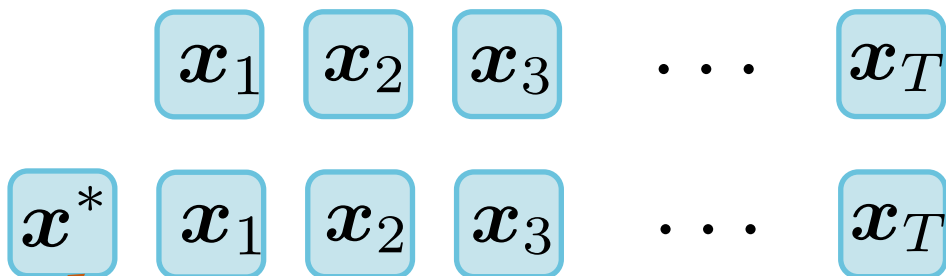
Softmax attention with a learnable sink token (Xiao et al. 2024)

$$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \begin{bmatrix} q^{*l,h} \\ Q^{l,h} \end{bmatrix} \begin{bmatrix} k^{*l,h\top} & K^{l,h\top} \end{bmatrix} + M \right) \begin{bmatrix} v^{*l,h} \\ V^{l,h} \end{bmatrix}$$

QKV for sink token

queries

keys



Learnable sink token

Sink on the learnable sink token

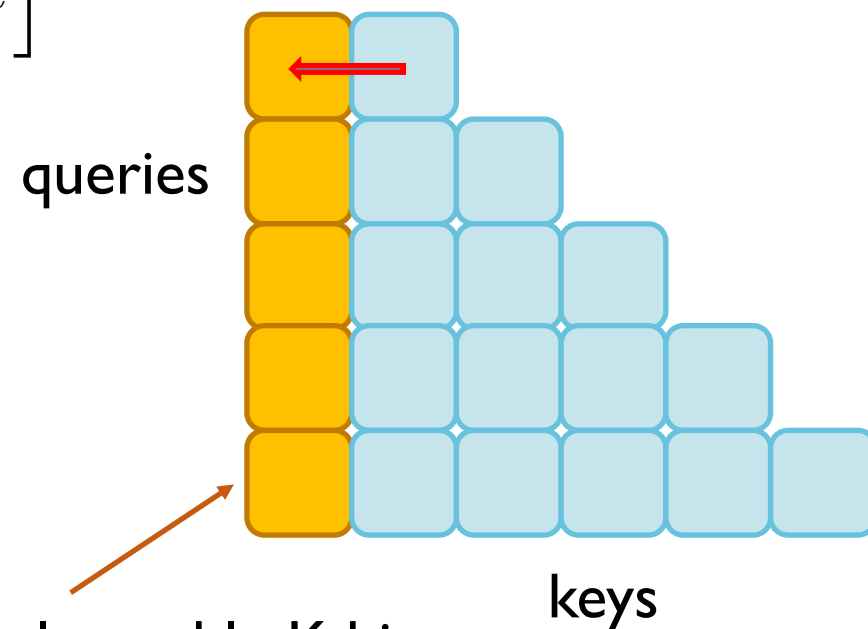
Xiao et al. Efficient Streaming Language Models with Attention Sinks. ICLR 2024

Effects of model architecture on attention sink

Softmax attention with **learnable KV biases** (Sun et al. 2024)

$$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \mathbf{Q}^{l,h} \begin{bmatrix} \mathbf{k}^{*l,h\top} & \mathbf{K}^{l,h\top} \end{bmatrix} + \mathbf{M} \right) \begin{bmatrix} \mathbf{v}^{*l,h} \\ \mathbf{V}^{l,h} \end{bmatrix}$$

Learnable KV biases



Effects of model architecture on attention sink

Softmax attention with learnable K biases

$$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \mathbf{Q}^{l,h} \left[\mathbf{k}^{*l,h\top} \quad \mathbf{K}^{l,h\top} \right] + \mathbf{M} \right) \begin{bmatrix} \mathbf{0} \\ \mathbf{V}^{l,h} \end{bmatrix}$$

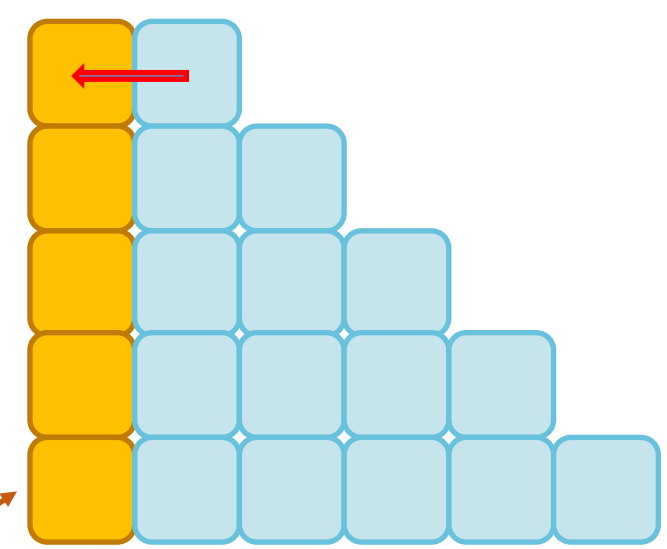
Learnable K biases

V biases are all zeros

queries

keys

Sink on the learnable K bias

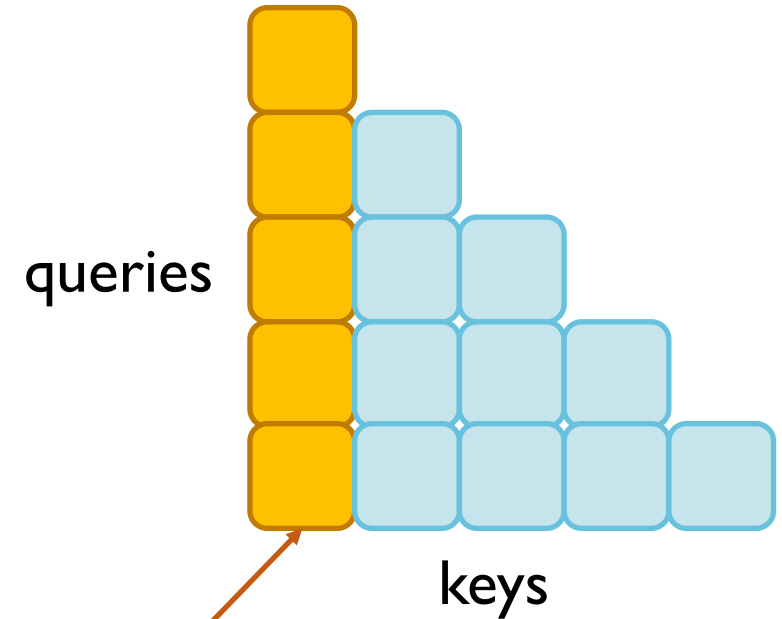


Effects of model architecture on attention sink

Softmax attention with **learnable V biases** (control group)

$$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \mathbf{Q}^{l,h} \mathbf{K}^{l,h\top} + \mathbf{M} \right) \mathbf{V}^{l,h} + \mathbf{v}^{*l,h}$$

Learnable V biases



Sink on the first token,
no effects

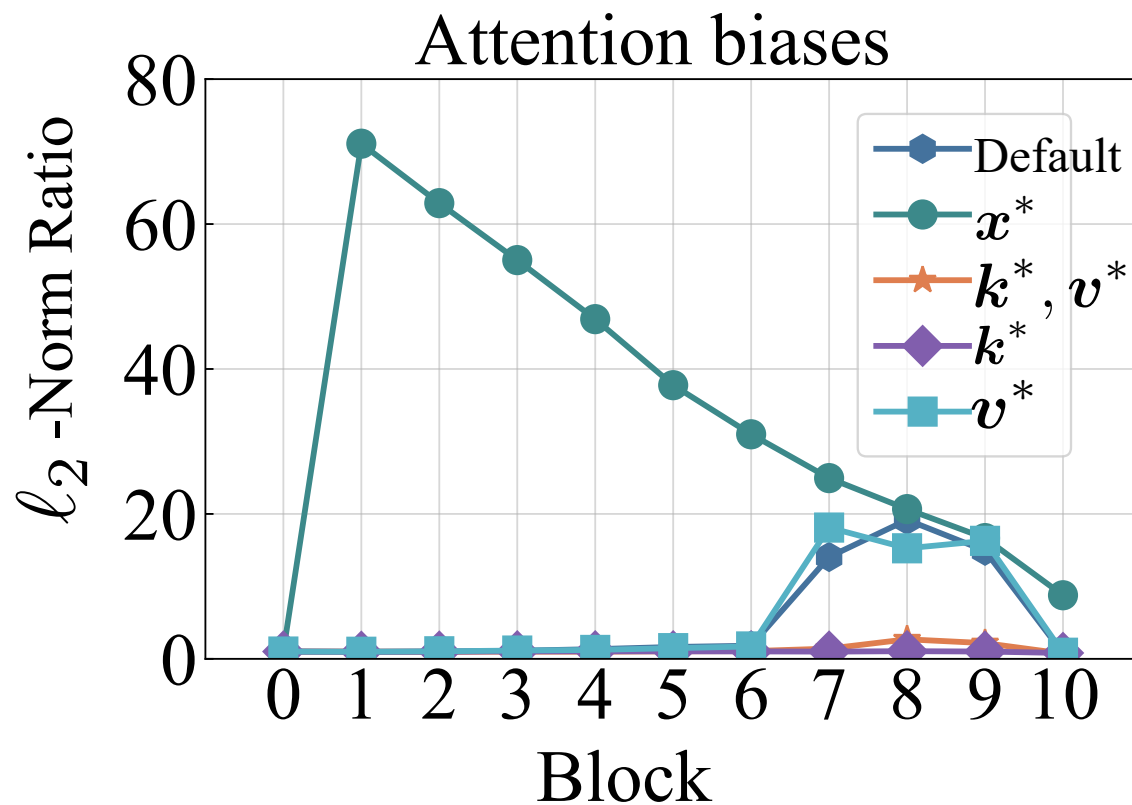
Effects of model architecture on attention sink

Effects of different learnable biases on attention sink

Attention in each head	$\text{Sink}_*^{\epsilon}(\%)$	$\text{Sink}_1^{\epsilon}(\%)$	valid loss
$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \mathbf{Q}^{l,h} \mathbf{K}^{l,h \top} + \mathbf{M} \right) \mathbf{V}^{l,h}$	-	18.18	3.73
$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \begin{bmatrix} \mathbf{q}^{*l,h} \\ \mathbf{Q}^{l,h} \end{bmatrix} \begin{bmatrix} \mathbf{k}^{*l,h \top} & \mathbf{K}^{l,h \top} \end{bmatrix} + \mathbf{M} \right) \begin{bmatrix} \mathbf{v}^{*l,h} \\ \mathbf{V}^{l,h} \end{bmatrix}$	74.12	0.00	3.72
$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \mathbf{Q}^{l,h} \begin{bmatrix} \mathbf{k}^{*l,h \top} & \mathbf{K}^{l,h \top} \end{bmatrix} + \mathbf{M} \right) \begin{bmatrix} \mathbf{v}^{*l,h} \\ \mathbf{V}^{l,h} \end{bmatrix}$	72.76	0.04	3.72
$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \mathbf{Q}^{l,h} \begin{bmatrix} \mathbf{k}^{*l,h \top} & \mathbf{K}^{l,h \top} \end{bmatrix} + \mathbf{M} \right) \begin{bmatrix} \mathbf{0} \\ \mathbf{V}^{l,h} \end{bmatrix}$	73.34	0.00	3.72
$\text{Softmax} \left(\frac{1}{\sqrt{d_h}} \mathbf{Q}^{l,h} \mathbf{K}^{l,h \top} + \mathbf{M} \right) \mathbf{V}^{l,h} + \mathbf{v}^{*l,h}$	-	17.53	3.73

Effects of model architecture on attention sink

Effects of different learnable biases on massive activations



Learnable token encourages massive activations

K biases eliminate massive activations
(KV biases also, but less): do not need!

Effects of model architecture on attention sink

Learnable K bias + fixed V bias

$\mathbf{v}^{*l,h}$	$\mathbf{0}$	\mathbf{v}'	$5\mathbf{v}'$	$20\mathbf{v}'$	\mathbf{v}''	$5\mathbf{v}''$	$20\mathbf{v}''$
$\text{Sink}_*^\epsilon(\%)$	73.34	70.03	44.43	1.51	69.74	27.99	0.00
$\text{Sink}_1^\epsilon(\%)$	0.00	0.06	3.71	25.88	2.15	5.93	11.21
valid loss	3.72	3.72	3.72	3.71	3.72	3.72	3.73

$$\mathbf{v}' = [1, 0, 0, \dots, 0]$$

$$\mathbf{v}'' = [1, 1, 1, \dots, 1] / \sqrt{d_h}$$

Effects of model architecture on attention sink

Learnable K bias with lower learnable dimensions + zero V bias

d_a	1	2	4	8	16	32	64
$\text{Sink}_*^\epsilon(\%)$	32.18	30.88	30.94	31.39	23.30	51.23	69.19
$\text{Sink}_1^\epsilon(\%)$	4.74	4.96	4.39	4.54	2.19	1.94	0.04
valid loss	3.73	3.72	3.72	3.73	3.73	3.73	3.72

K for sink token locates in a low-dimensional manifold

Similar to the manifold constructed by low-dimensional spikes

Effects of model architecture on attention sink

- Large attention score \neq important in semantic
- Sink token saves extra attention, adjusts the dependence among other tokens

Why need such a mechanism?

Is it because attention score added up to one?

Effects of model architecture on attention sink

Attention output

$$\mathbf{v}_i^\dagger = \sum_{j=1}^i \frac{\text{sim}(\varphi(\mathbf{q}_i), \varphi(\mathbf{k}_j))}{\sum_{j'=1}^i \text{sim}(\varphi(\mathbf{q}_i), \varphi(\mathbf{k}_{j'}))} \mathbf{v}_j$$

$$\text{sim}(\varphi(\mathbf{q}_i), \varphi(\mathbf{k}_j)) = \exp\left(\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_h}}\right)$$

softmax

$$\mathbf{Z}_i = \sum_{j'=1}^i \text{sim}(\varphi(\mathbf{q}_i), \varphi(\mathbf{k}_{j'}))$$

normalization term

Perhaps normalization matters, as it forces the attention scores sum to one?

Effects of model architecture on attention sink

- Scale the normalization term

$$\mathbf{Z}_i \rightarrow \mathbf{Z}_i / \alpha$$

- Power of attention scores sum up to one

$$\mathbf{v}_i^\dagger = \frac{\sum_{j=1}^i \text{sim}(\varphi(\mathbf{q}_i), \varphi(\mathbf{k}_j)) \mathbf{v}_j}{\left(\sum_{j'=1}^i \text{sim}(\varphi(\mathbf{q}_i), \varphi(\mathbf{k}_{j'})) \right)^{\frac{1}{p}}}$$
$$\mathbf{v}_i^\dagger = \sum_{j=1}^i \left(\frac{\exp(\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_h}/p})}{\sum_{j'=1}^i \exp(\frac{\mathbf{q}_i \mathbf{k}_{j'}^\top}{\sqrt{d_h}/p})} \right)^{\frac{1}{p}} \mathbf{v}_j$$

softmax

- May mitigate attention sink, but not prevent the emergence

Effects of model architecture on attention sink

- Relax tokens' inner dependence by removing normalization

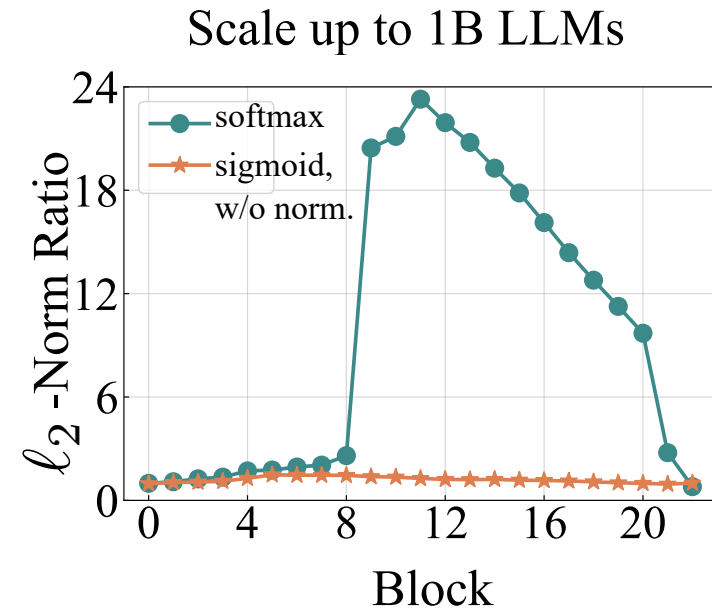
Sigmoid attention:

$$\mathbf{v}_i^\dagger = \sum_{j=1}^i \text{sigmoid}\left(\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_h}}\right) \mathbf{v}_j$$

ELU plus one attention:

$$\mathbf{v}_i^\dagger = \sum_{j=1}^i \left(\text{elu}\left(\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d_h}}\right) + 1 \right) \mathbf{v}_j$$

No normalization -> No attention sink, no massive activations!
Added back normalization -> Attention sink, massive activations!



Effects of **model architecture** on attention sink

- Relax tokens' inner dependence by allowing negative attention scores

Linear attention, with a mlp kernel

$$\mathbf{v}_i^\dagger = \sum_{j=1}^i \frac{\text{mlp}(\mathbf{q}_i)\text{mlp}(\mathbf{k}_j)^\top}{\sqrt{d_h}} \mathbf{v}_j \rightarrow \text{No attention sink, no massive activations}$$

Add a normalization

$$\mathbf{Z}_i = \max \left(\left| \sum_{j'=1}^i \frac{\text{mlp}(\mathbf{q}_i)\text{mlp}(\mathbf{k}_{j'})^\top}{\sqrt{d_h}} \right|, 1 \right) \rightarrow \text{No attention sink, no massive activations}$$

Effects of model architecture on attention sink

- Alternative LM architecture which have no attention sink

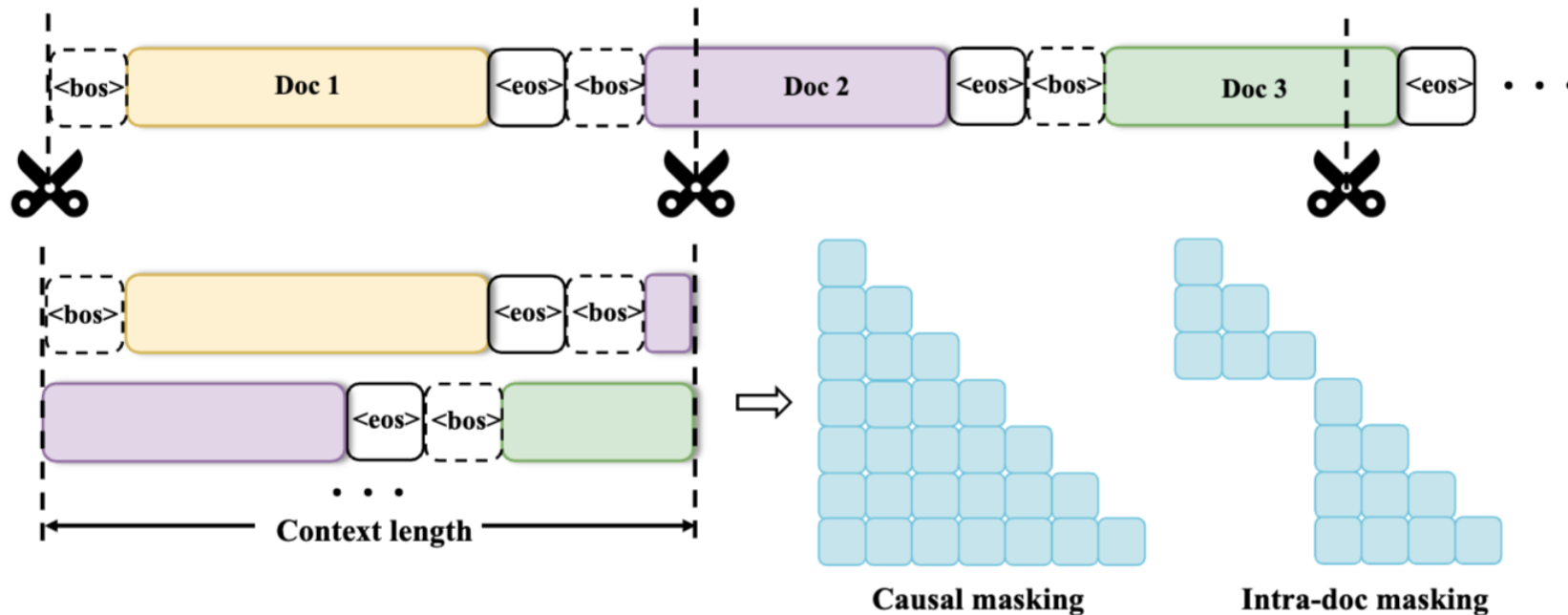
Softpick (Zuhri et al. 2025)

$$\text{Softpick}(\mathbf{x})_i = \frac{\text{ReLU}(e^{x_i - m} - e^{-m})}{\sum_{j=1}^N |e^{x_j - m} - e^{-m}| + \epsilon}$$

Zuhri et al. Softpick: No Attention Sink, No Massive Activations with Rectified Softmax. 2025

Effects of data packing on attention sink (extension)

- In (Barbero et al. 2025, I am the 3rd author), I checked how data packing affects attention sink and its relationship with <BOS>



Barbero et al. Why do LLMs attend to the first token?. 2025

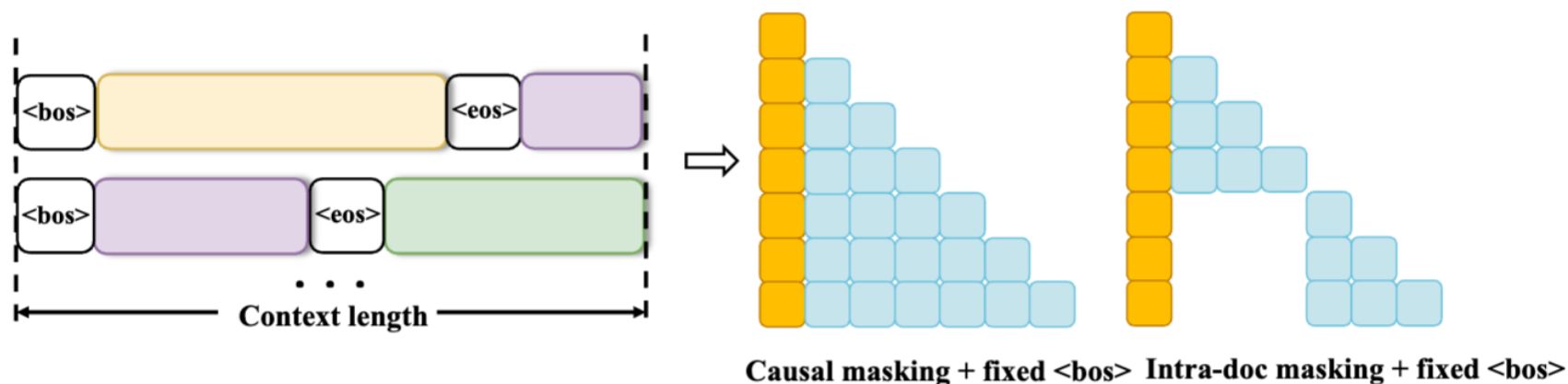
Effects of data packing on attention sink (extension)

- Causal masking vs intra-doc masking

Attention Masking	$\langle \text{bos} \rangle$	$\langle \text{eos} \rangle$	Inference	Sink Metric (%)	Valid loss
Causal	No	Yes	$\langle \text{bos} \rangle^* + \text{text}$	65.10	2.69
Causal	No	Yes	text	65.15	2.70
Causal+fixed $\langle \text{bos} \rangle$	Yes	Yes	$\langle \text{bos} \rangle + \text{text}$	90.84	2.69
Causal+fixed $\langle \text{bos} \rangle$	Yes	Yes	text	<u>0.05</u>	<u>7.56</u>
Intra-doc	No	Yes	text	28.23	2.67
Intra-doc	Yes	Yes	$\langle \text{bos} \rangle + \text{text}$	83.33	2.67
Intra-doc	Yes	Yes	text	50.24	2.68
Intra-doc + fixed $\langle \text{bos} \rangle$	Yes	Yes	$\langle \text{bos} \rangle + \text{text}$	90.56	2.67
Intra-doc + fixed $\langle \text{bos} \rangle$	Yes	Yes	text	<u>0.00</u>	<u>7.78</u>

Effects of data packing on attention sink (extension)

- Fix a <BOS> in the first position of context window



Effects of data packing on attention sink (extension)

- Training with fix <BOS> makes LLMs very sensitive to it during the inference

Attention Masking	<bos>	<eos>	Inference	Sink Metric (%)	Valid loss
Causal	No	Yes	<bos> * + text	65.10	2.69
Causal	No	Yes	text	65.15	2.70
Causal+fixed <bos>	Yes	Yes	<bos> + text	90.84	2.69
Causal+fixed <bos>	Yes	Yes	text	0.05	7.56
Intra-doc	No	Yes	text	28.23	2.67
Intra-doc	Yes	Yes	<bos> + text	83.33	2.67
Intra-doc	Yes	Yes	text	50.24	2.68
Intra-doc + fixed <bos>	Yes	Yes	<bos> + text	90.56	2.67
Intra-doc + fixed <bos>	Yes	Yes	text	0.00	7.78

Main motivations in this talk

- Attention sink is important due to above applications
- Big questions:

How to understand attention sink?

When attention sink appears in LLMs?

Why LLMs need attention sink?

Why LLMs need attention sink

- Without changing attention operation, the always used LM pre-training recipes always result in attention sink
- Why LLMs need attention sink in the above scenario?

Why LLMs need attention sink

- The aim of attention is to mix representations of tokens
- If all heads are doing token mixing, the token representations will be over-squashed, leads to representation collapse
- Attention sink head is a no-op, preventing over-mixing

Barbero et al. Why do LLMs attend to the first token?. 2025



Thank you for listening.