Attention Sink in LLMs and its Applications

Presenter: **Xiangming Gu**

Affiliation: Google Deepmind, National University of Singapore

I am attempting to answer ...

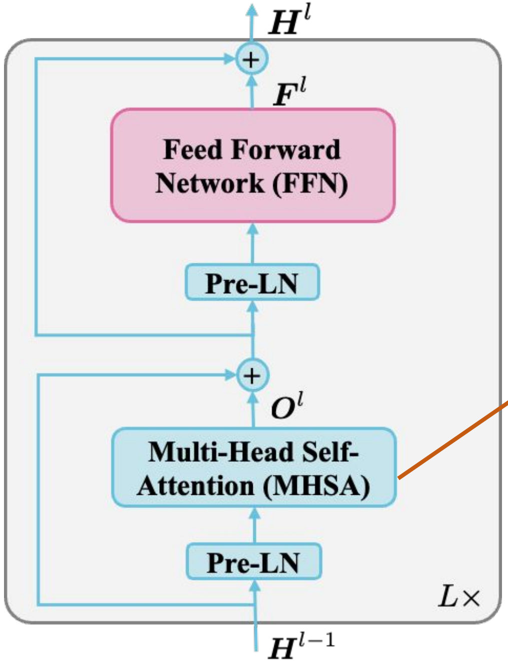
- Mechanism understanding of Attention Sink?
- When Attention Sink Emerges in LLMs?
- Why LLMs need Attention Sink?
- Why GPT-OSS and Qwen3-Next consider Attention Sink in the Model Design?

Covered the following two papers

- When Attention Sink Emerges in Language Models: An Empirical View. ICLR 2025
- Why Do LLMs Attend to the First Token? COLM 2025

What is Attention Sink?

- Decoder-only Transformer



Self-attention is one of the most important parts

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

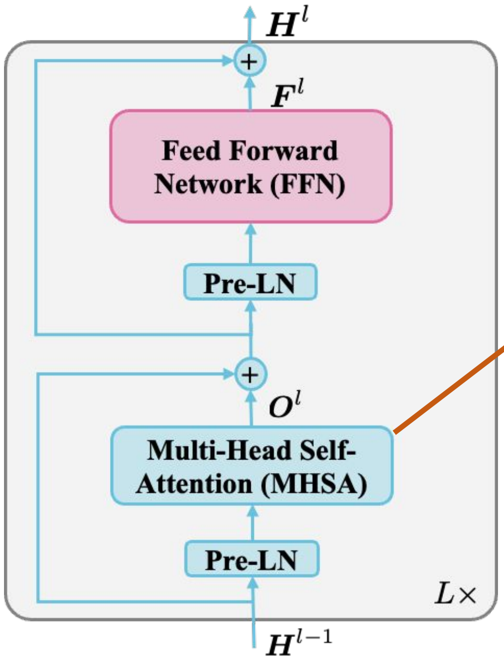
queries

keys

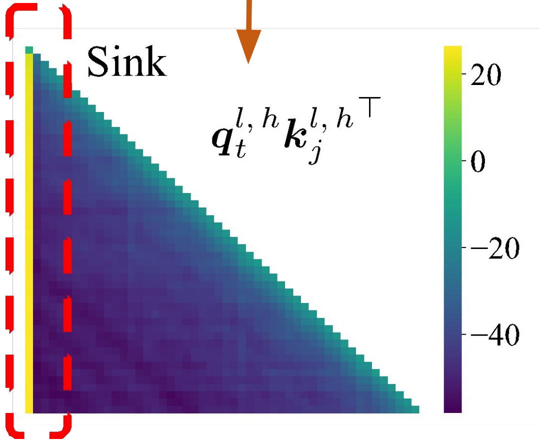
values

Casual mask

What is Attention Sink?



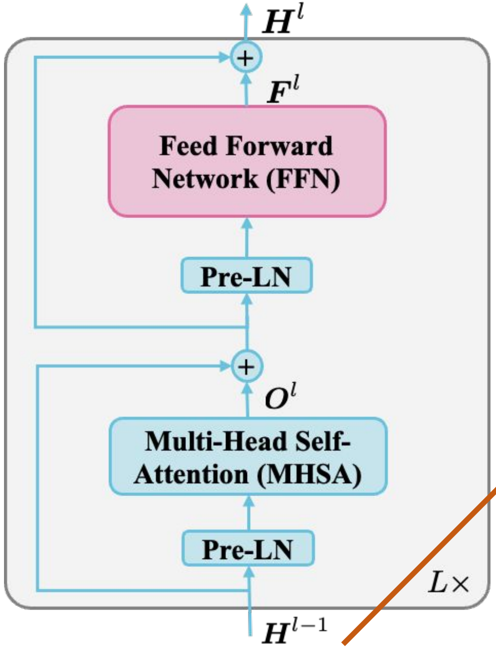
$$\mathrm{Softmax}\left(\frac{1}{\sqrt{d_h}}\boldsymbol{Q}^{l,h}\boldsymbol{K}^{l,h\top}+\boldsymbol{M}\right)\boldsymbol{V}^{l,h}$$

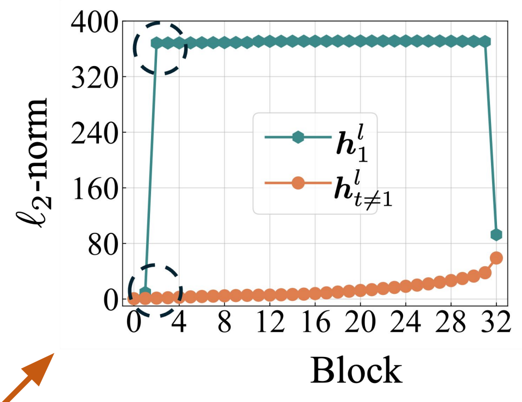


Xiao et al. Efficient Streaming Language Models with Attention Sinks. ICLR 2024

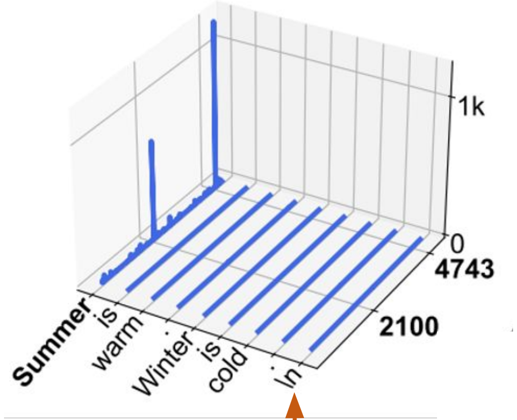
Phenomenons associated to Attention Sink

- Massive Activations





Activations extremely large

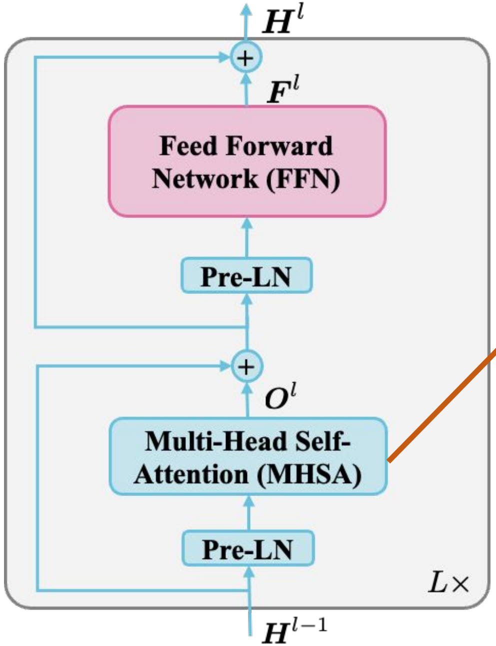


Few dimensions have spikes/outliers

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

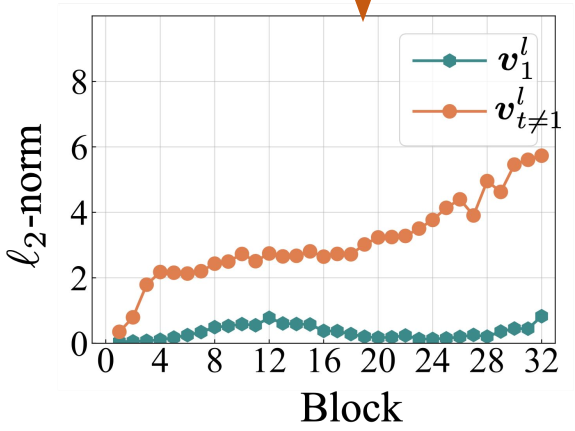
Phenomenons associated to Attention Sink

- Value Drains



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

Values extremely small

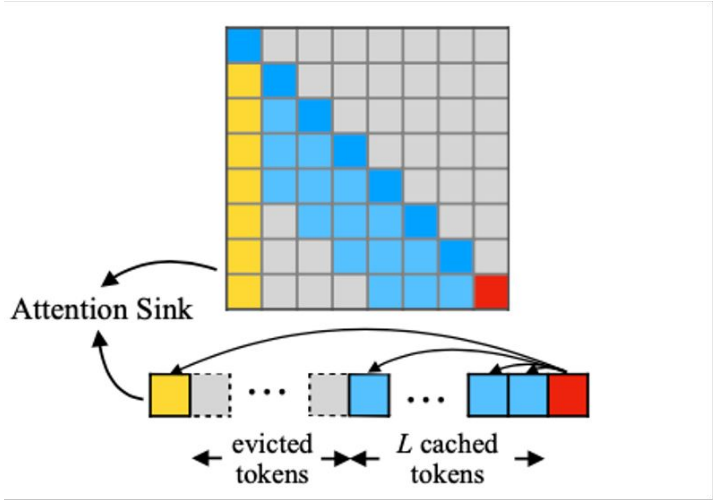


Gu et al. When Attention Sink Emerges in Language Models. ICLR 2025

Post-hoc Applications of Attention Sink

- Long context understanding / generation

- Only computing attention on the first token and recent tokens

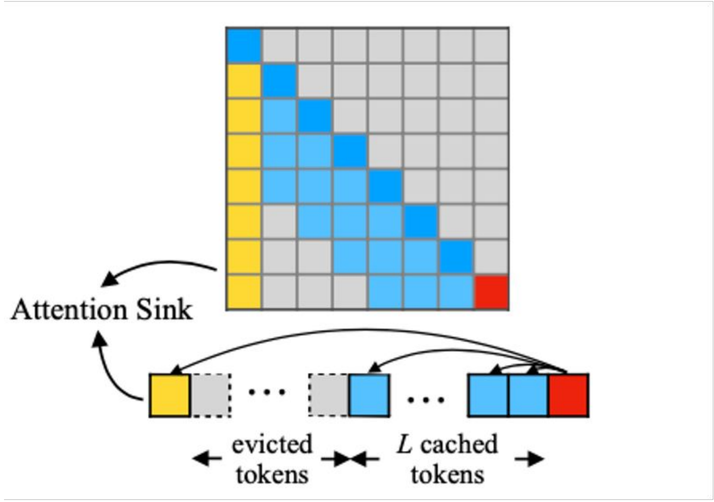


Xiao et al. Efficient Streaming Language Models with Attention Sinks. ICLR 2024

Post-hoc Applications of Attention Sink

- KV cache optimization

- Only retaining KV cache of sink tokens and recent tokens

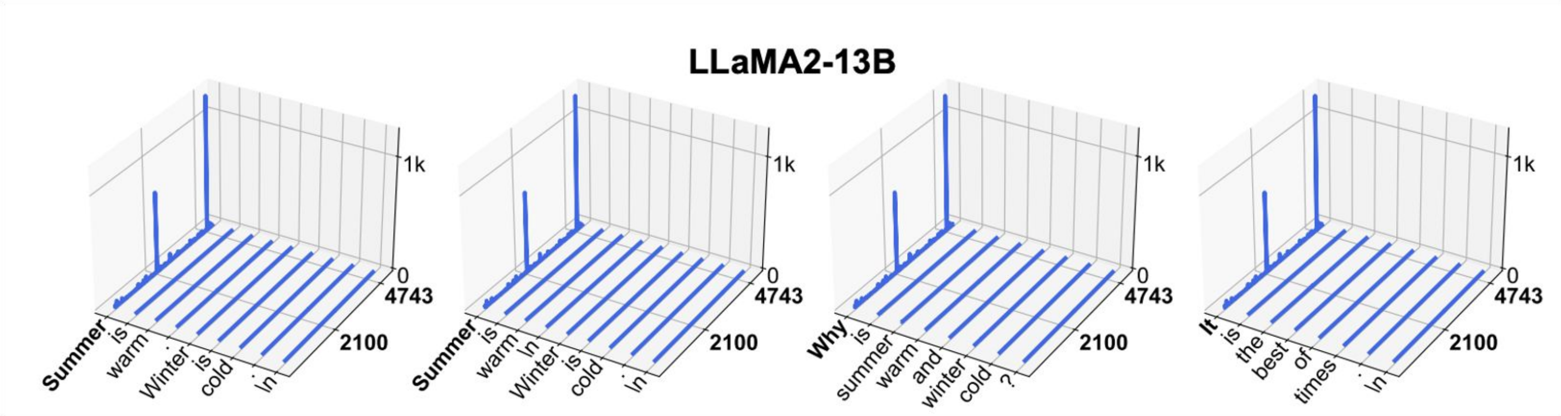


Ge et al. Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. ICLR 2024

Post-hoc Applications of Attention Sink

- Model quantization

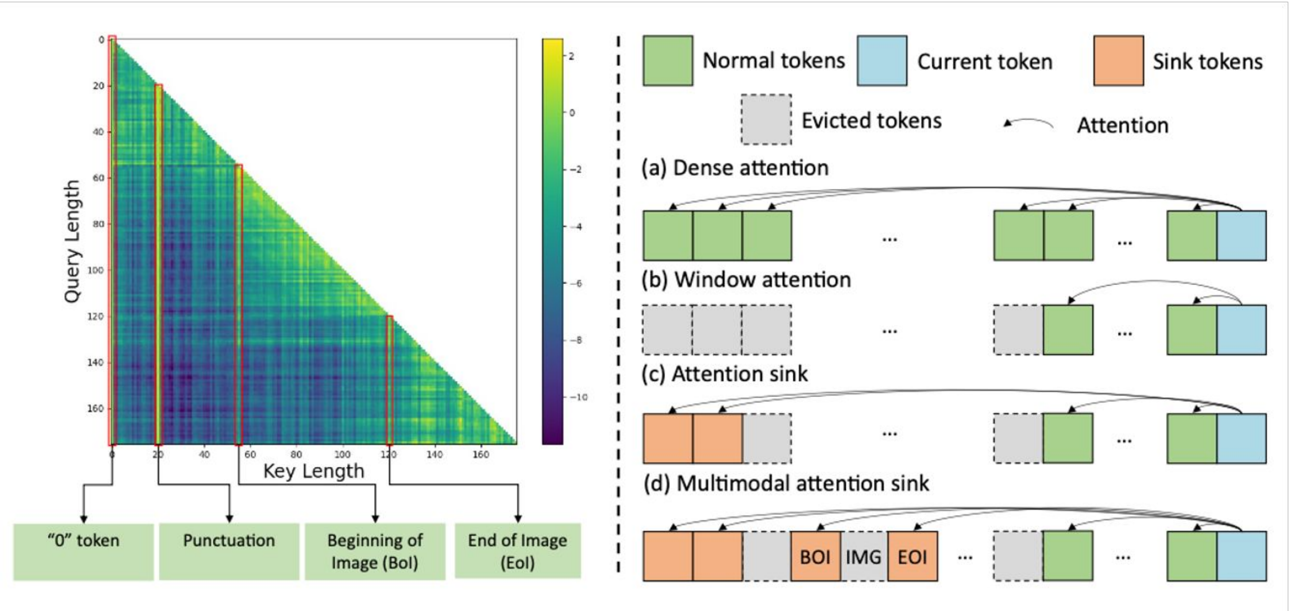
- Preserving the full precision of KV cache of sink token



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

Post-hoc Applications of Attention Sink

- Multimodal language modeling



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

I am attempting to answer ...

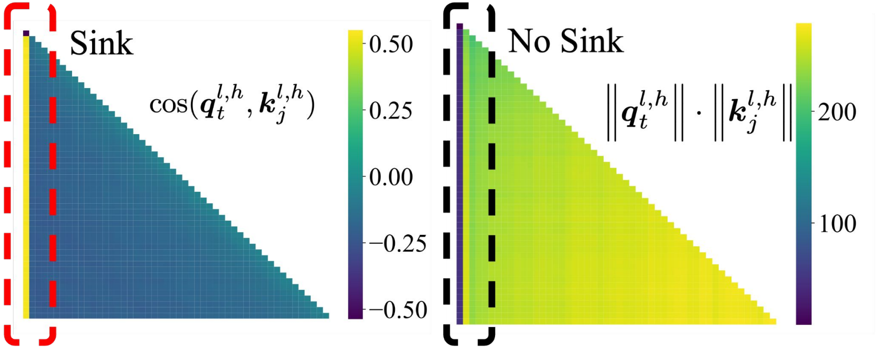
- Mechanism understanding of Attention Sink?
- When Attention Sink Emerges in LLMs?
- Why LLMs need Attention Sink?
- Why GPT-OSS and Qwen3-Next consider Attention Sink in the Model Design?

Mechanism Understanding of Attention Sink

Attention sink is due to the key ***key*** bias of the sink token

$$\boldsymbol{q}_t^{l,h} \boldsymbol{k}_1^{l,h\top} \gg \boldsymbol{q}_t^{l,h} \boldsymbol{k}_{j\neq 1}^{l,h\top}$$

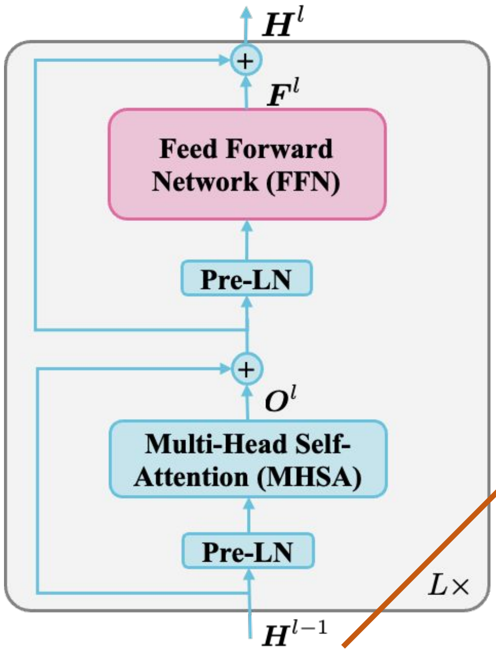
$$\cos(\boldsymbol{q}_t^{l,h}, \boldsymbol{k}_1^{l,h}) \gg \cos(\boldsymbol{q}_t^{l,h}, \boldsymbol{k}_{j\neq 1}^{l,h})$$

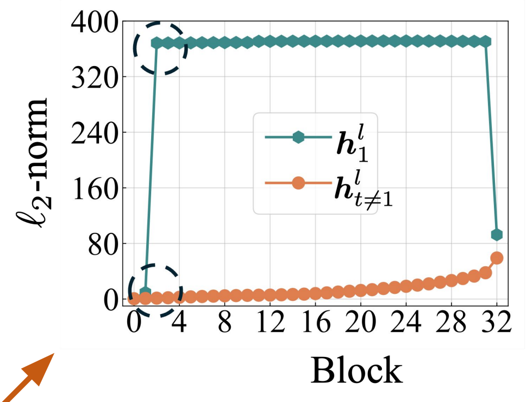


key of the sink token is located in the different manifold, it has small angles with any queries

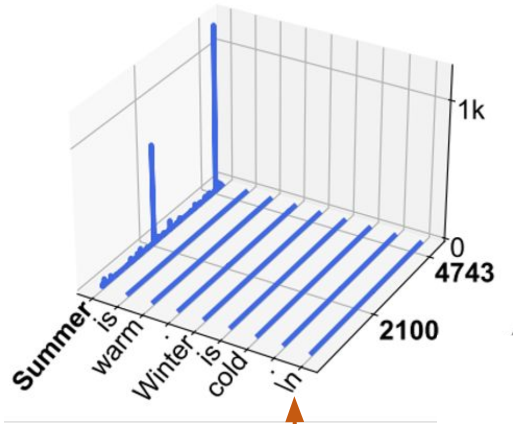
Mechanism Understanding of Attention Sink

- Massive Activations





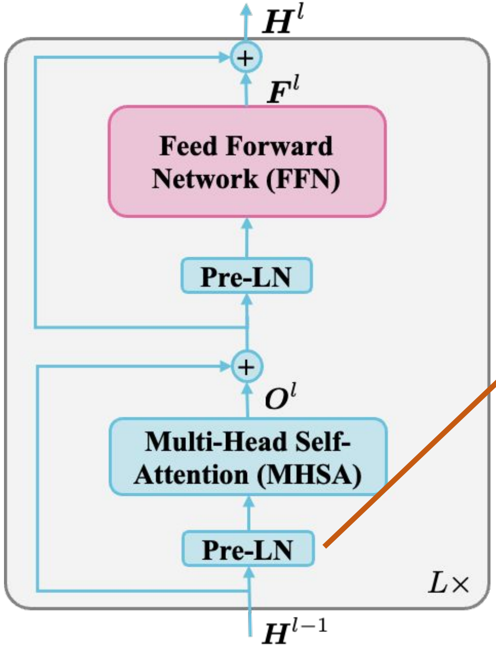
Activations extremely large



Few dimensions have spikes/outliers

Mechanism Understanding of Attention Sink

- Existence of massive activations is to support attention sink



$$\mathrm{LN}(\boldsymbol{h}) = \frac{\boldsymbol{h}}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} \boldsymbol{h}_i^2}} \odot \boldsymbol{g}$$

Layer Norm only retain the spike dimensions (dominate the norm)

$$\boldsymbol{k}_t^{l,h} = \mathrm{LN}(\boldsymbol{h}_t^{l-1})\boldsymbol{W}_K^{l,h}\boldsymbol{R}_{\Theta,\,-t}$$

Linear transformations of spikes

Similar mechanism for small values

Mechanism Understanding of Attention Sink

Why all these phenomenon tend to happen in the first token (not necessary to be BOS)?

- Uniqueness of the first token: self-attention involves no other tokens, all hidden states in the forward path are equivalent to MLP transformations of input embeddings

- LLMs learn to map the input embeddings to massive activations after certain layers, leading to key bias, and then attention sink

Mechanism Understanding of Attention Sink

- Attention sink approximates “no-op”

$$\boldsymbol{v}_i^\dagger = \sum_{j=1}^{i} \boldsymbol{\alpha}_{ij} \boldsymbol{v}_j = \boldsymbol{\alpha}_{i1} \boxed{\boldsymbol{v}_1} + \sum_{j \neq 1}^{i} \boxed{\boldsymbol{\alpha}_{ij}} \boldsymbol{v}_j$$

Small

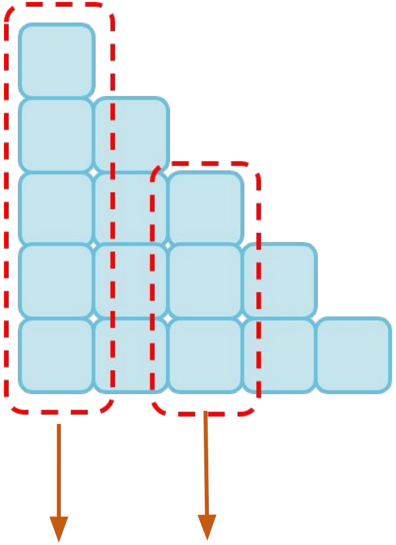
I am attempting to answer …

- Mechanism understanding of Attention Sink?
- When Attention Sink Emerges in LLMs?
- Why LLMs need Attention Sink?
- Why GPT-OSS and Qwen3-Next consider Attention Sink in the Model Design?

A metric to measure Attention Sink

- Motivations: attention scores of the first token dominates

Average



Importance score $\alpha_k^{l,h}$

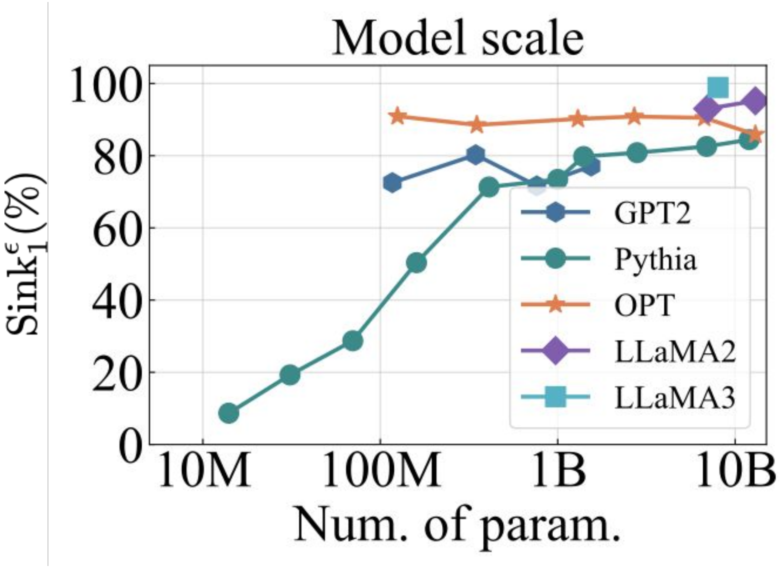
$$\mathrm{Sink}_k^{\epsilon} = \frac{1}{L}\sum_{l=1}^{L}\frac{1}{H}\sum_{h=1}^{H}\mathbb{I}(\alpha_k^{l,h} > \epsilon)$$

Attention sink metric of the whole LM

Within a head, a threshold to decide sink, e.g., 0.3 for 64 tokens

Attention Sink w.r.t. Model Scale / Training Stage

- Attention sink emerges in small LMs, even with 14M params.



- Attention sink already emerges in LM pre-training.

LLM	$\mathrm{Sink}_1^{\epsilon}(\%)$ Base	$\mathrm{Sink}_1^{\epsilon}(\%)$ 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 w.r.t. Different Inputs

- Attention sink emerges with / without BOS (for most LLMs), even with random tokens as input
- Under all the repeated tokens?

LLM	$\text{Sink}_1^{\epsilon}(\%)$ 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

Related to positional embeddings

Attention Sink with Repeated Tokens as Inputs

- For LLMs with NOPE / Relative PE / ALiBi / Rotary

$$\boldsymbol{P} = \boldsymbol{0}$$

Residual streams before Transformer blocks

$$\boldsymbol{h}_t^0 = \boldsymbol{x}\boldsymbol{W}_E + \boldsymbol{P}$$

Then

$$\boldsymbol{h}_1^0 = \boldsymbol{h}_2^0 = \cdots = \boldsymbol{h}_T^0$$

Using induction, we can prove (all have massive activations, distribute the sink)

$$\boldsymbol{h}_1^l = \boldsymbol{h}_2^l = \cdots = \boldsymbol{h}_T^l, \ \forall \ 0 \leq l \leq L$$

Attention Sink with Repeated Tokens as Inputs

- We can even derive the closed form / upper bound attention distributions for NOPE / Relative PE / ALiBi / Rotary (see the paper).

- However, absolute / learnable PE (e.g., GPT2) have no such properties

LLM	$\text{Sink}_1^{\epsilon}(\%)$ 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

When Attention Sink Emerges in LLMs?

- Attention sink appears during LLM pre-training

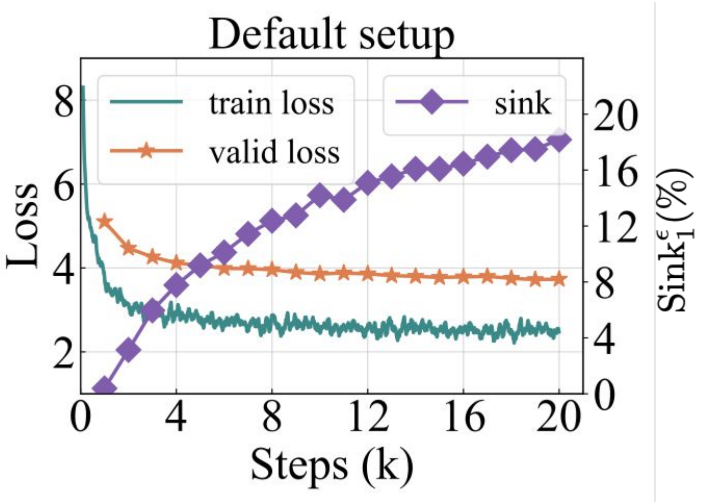
- Attributing attention sink phenomenon to LLM pre-training

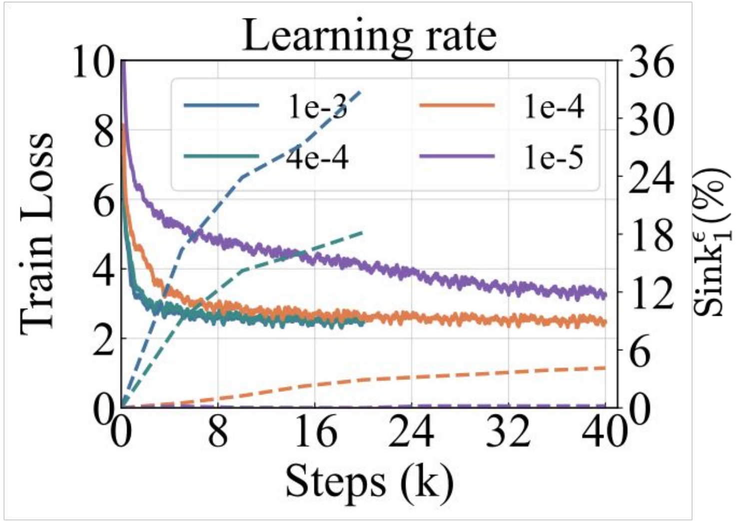
$$\min_{\theta} \mathbb{E}_{\boldsymbol{X} \sim p_{\text{data}}} \left[\mathcal{L}\left(p_{\theta}(\boldsymbol{X}) \right) \right]$$

Optimization Data distribution Loss function Model architecture

Effects of Optimization

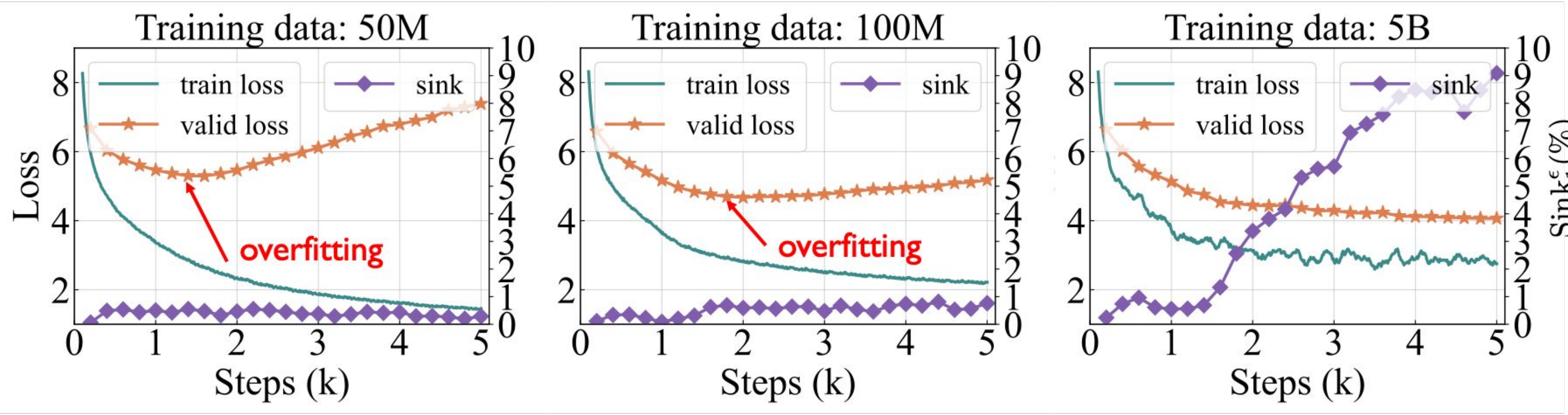
- Attention sink appears during LLM pre-training process (not initialization)
- Large LR encourages attention sink (even under the same LR*steps)





Effects of Data Distribution

- Attention sink emerges when we have enough unique training data amount



Effects of Loss function

- Weight decay encourages attention sink

L2 regularization

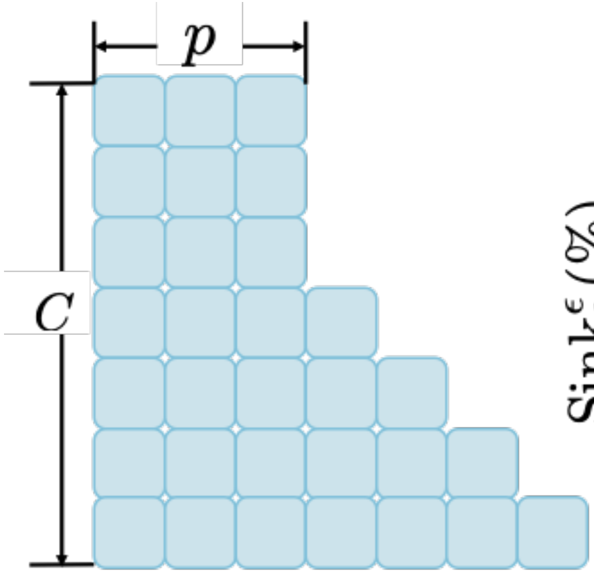
$$\mathcal{L} = \sum_{t=2}^{C} \log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}) + \gamma \, \|\theta\|_2^2$$

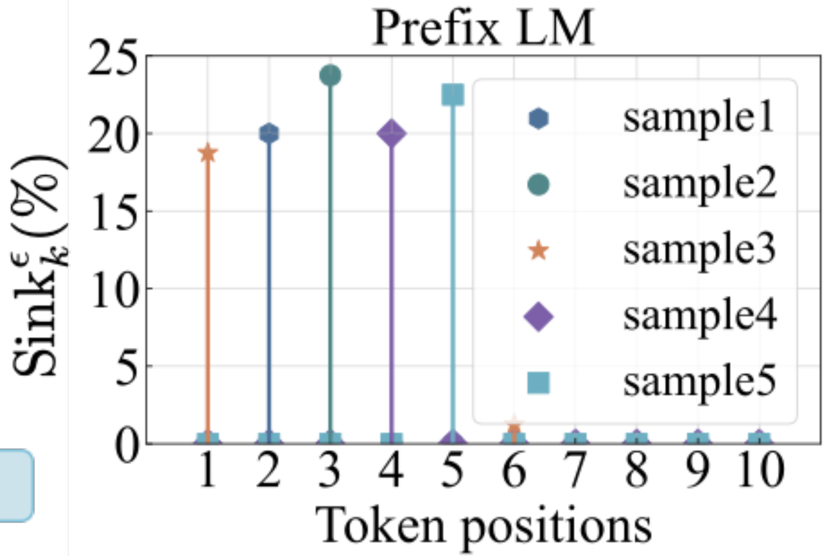
γ	0.0	0.001	0.01	0.1	0.5	1.0	2.0	5.0
$\text{Sink}_1^{\epsilon}(\%)$	15.20	15.39	15.23	18.18	41.08	37.71	6.13	0.01
valid loss	3.72	3.72	3.72	3.73	3.80	3.90	4.23	5.24

Effects of Loss function

- Prefix language modeling: sink token shifts from the first token to other positions within the prefix

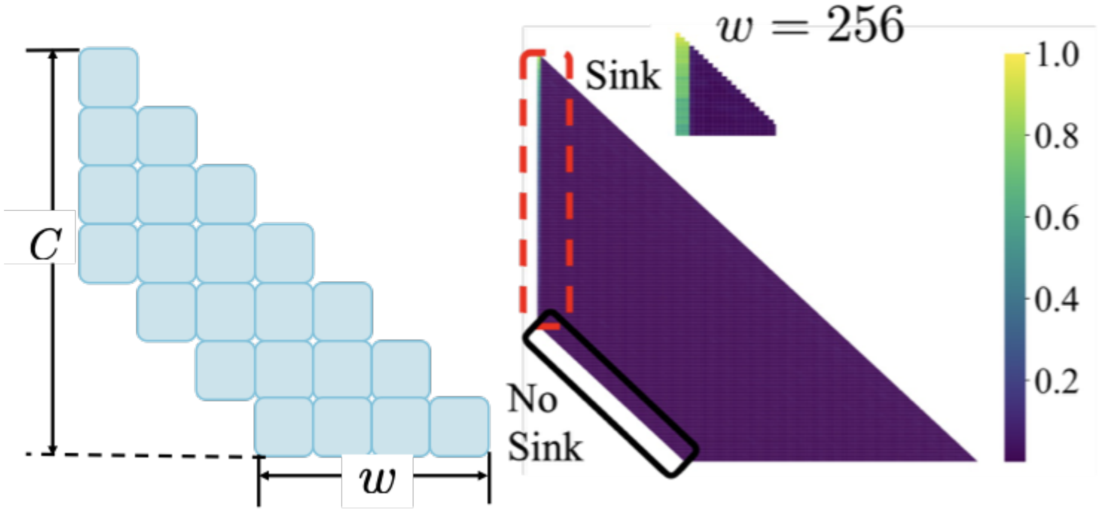
$$\mathcal{L} = \sum_{t=p+1}^{C} \log p_\theta(\boldsymbol{x}_t | \boldsymbol{x}_{p+1:t-1}, \boldsymbol{x}_{1:p})$$





Effects of Loss function

- Shift window attention: attention sink appears on the absolute, not the relative first token
- Small window size mitigates attention sink



Validating sink token has key bias

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

Effects of Model Architecture

The following designs do not affect the emergence of attention sink

- Positional embeddings

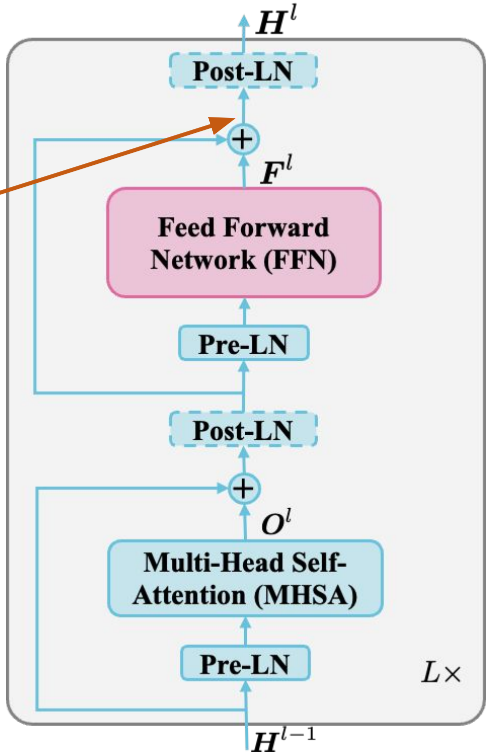
NOPE, learnable PE, absolute PE, relative PE, Rotary, ALIBI

Effects of Model Architecture

The following designs do not affect the emergence of attention sink

- Positional embeddings
- Pre-norm or post-norm

Massive activations
happen before LN



Effects of Model Architecture

The following designs do not affect the emergence of attention sink

- Positional embeddings
- Pre-norm or post-norm
- FFNs with different activation functions
- Number of attention heads, how to combine multiple heads
- ...

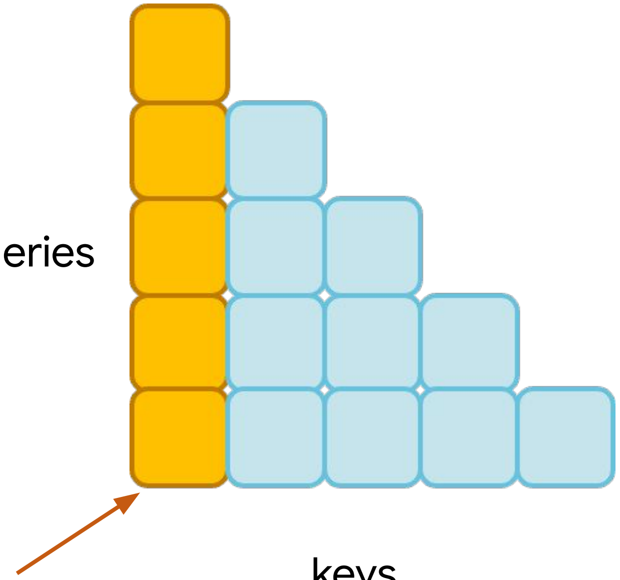
Effects of Model Attention Design

- Standard softmax attention

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

queries keys values

Casual mask

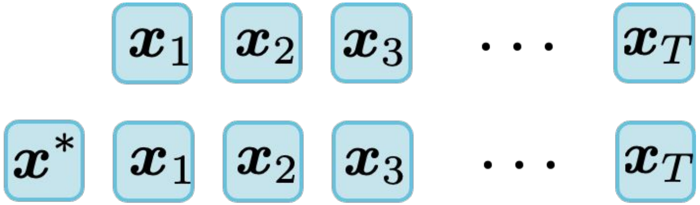


Effects of Model Attention Design

- Softmax attention with a learnable sink token

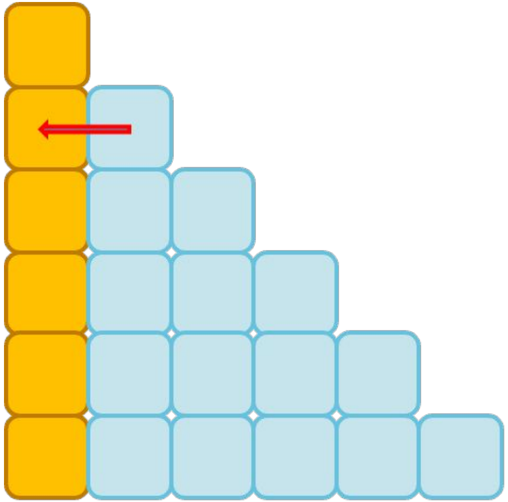
$$\mathrm{Softmax}\left(\frac{1}{\sqrt{d_h}}\begin{bmatrix}\boldsymbol{q}^{*l,h}\\ \boldsymbol{Q}^{l,h}\end{bmatrix}\begin{bmatrix}\boldsymbol{k}^{*l,h^\top} & \boldsymbol{K}^{l,h^\top}\end{bmatrix}+\boldsymbol{M}\right)\begin{bmatrix}\boldsymbol{v}^{*l,h}\\ \boldsymbol{V}^{l,h}\end{bmatrix}$$

QKV for sink token



Learnable sink token

queries



keys

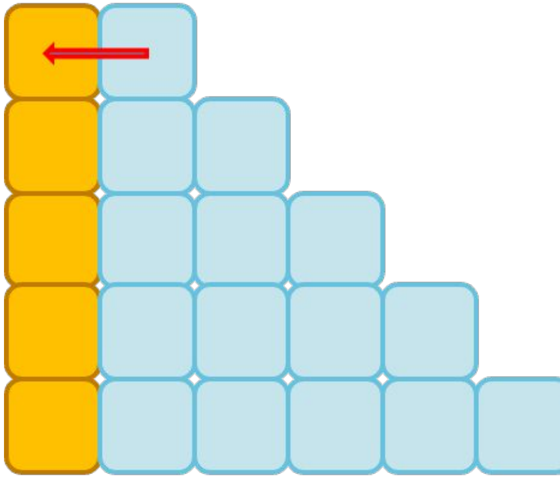
Sink on the learnable sink token

Effects of Model Attention Design

- Softmax attention with learnable KV biases

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

Learnable KV biases

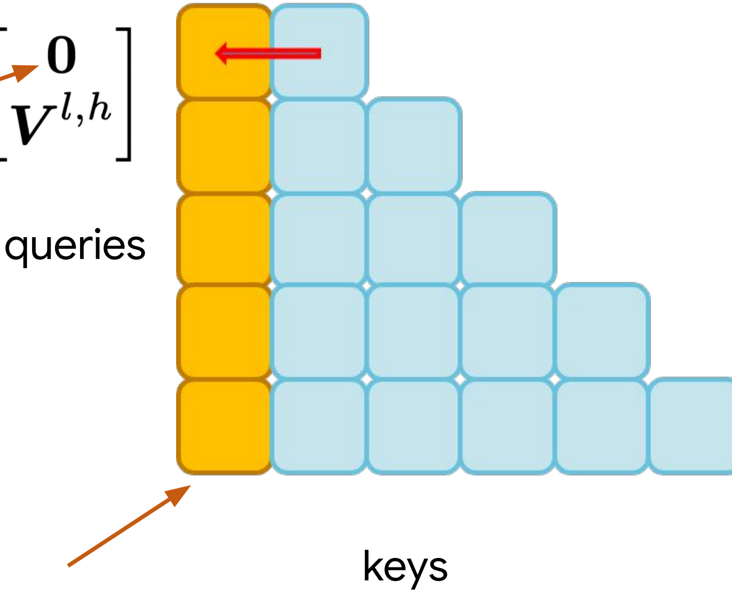


Effects of Model Attention Design

- Softmax attention with learnable K biases

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

Learnable K biases,
zero V biases



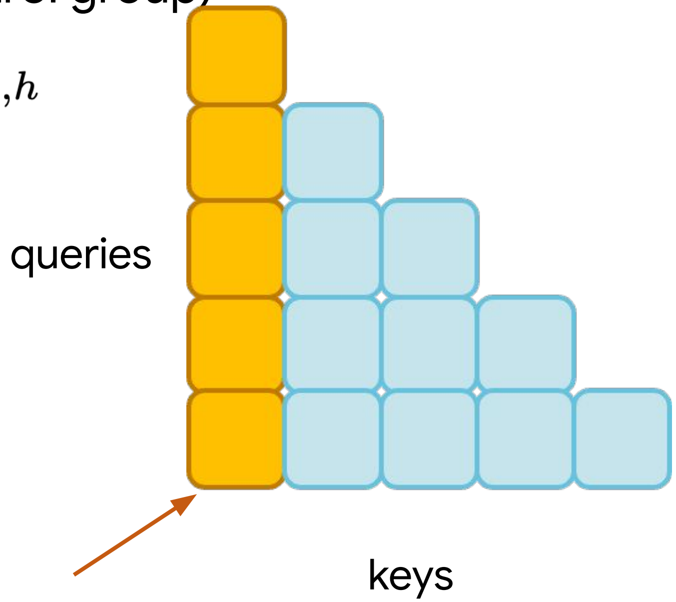
Sink on the learnable
key biases

Effects of Model Attention Design

- Softmax attention with learnable K biases (control group)

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

Learnable V biases



Sink on the first token,
no effects

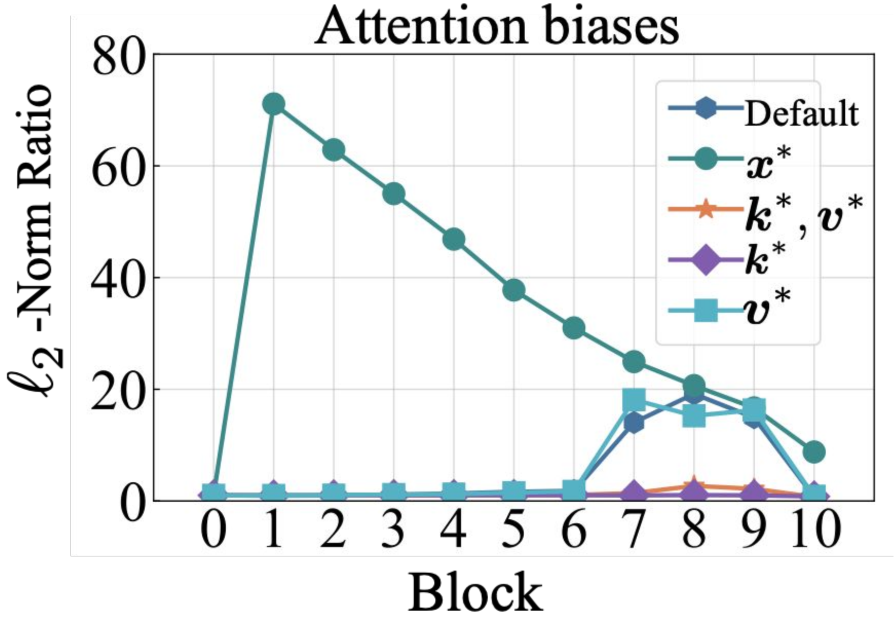
Effects of Attention Biases

- Attention biases can absorb attention sink from the actual first token

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

Effects of Attention Biases

- Key biases can significantly mitigates massive activations, as no need to develop new biases



Effects of Attention Biases

- Value bias needs to be close to zero

$\boldsymbol{v}^{*l,h}$	$\mathbf{0}$	$\boldsymbol{v}'$	$5\boldsymbol{v}'$	$20\boldsymbol{v}'$	$\boldsymbol{v}''$	$5\boldsymbol{v}''$	$20\boldsymbol{v}''$
$\mathrm{Sink}_*^{\epsilon}(\%)$	73.34	70.03	44.43	1.51	69.74	27.99	0.00
$\mathrm{Sink}_1^{\epsilon}(\%)$	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

$\boldsymbol{v}' = [1, 0, 0, .., 0]$

$\boldsymbol{v}'' = [1, 1, 1, .., 1]/\sqrt{d_h}$

Effects of Attention Biases

- Key bias is low-rank

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

Comparing different Attention Biases

- Learnable key biases, zero value biases

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

- Softmax off-by-one

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

- Learnable attention score biases (single number for each head, layer)

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

$$\boldsymbol{b}^{*l,h} = b^{*l,h}[1, 1, 1, .., 1]$$

Comparing different Attention Biases

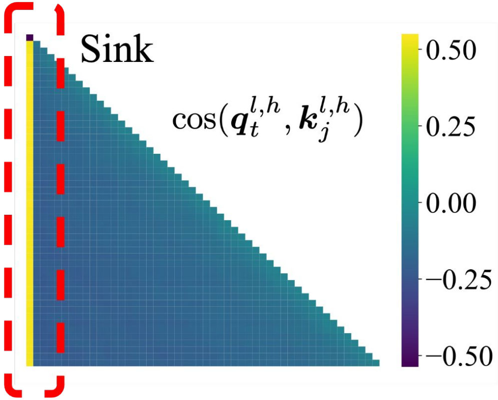
- Softmax off-by-one: with any query, the cosine similarity is zero

$$\mathrm{Softmax}\left(\frac{1}{\sqrt{d_h}}\begin{bmatrix}\mathbf{0}^{*l,h\top} & \boldsymbol{Q}^{l,h}\boldsymbol{K}^{l,h\top}\end{bmatrix} + \boldsymbol{M}\right)\begin{bmatrix}\mathbf{0} \\ \boldsymbol{V}^{l,h}\end{bmatrix}$$

- Original format:

$$(\mathrm{softmax}_1(x))_i = \frac{\exp(x_i)}{1 + \sum_j \exp(x_j)}$$

- Zero may already be enough



Effects of Attention Biases

The learnable key bias and **zero** value bias experiments show that:

- Large attention score does not mean important in semantic
- Sink token save extra attention, adjusts the dependence among tokens

But why LLMs need such a mechanism?

Effects of Normalization in Softmax Attention

Whether this is due to the normalization in Softmax attention?

$$\boldsymbol{v}_i^{\dagger} = \sum_{j=1}^{i} \frac{\alpha \mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_j))}{\sum_{j'=1}^{i} \mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_{j'}))} \boldsymbol{v}_j = \sum_{j=1}^{i} \frac{\mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_j))}{\sum_{j'=1}^{i} \mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_{j'}))} \boldsymbol{h}_j(\alpha \boldsymbol{W}_V),$$

$$\boldsymbol{o}_i' = \mathrm{Concat}_{h=1}^{H}(\boldsymbol{v}_i'^{h}) \boldsymbol{W}_O.$$

Scaling the normalization $\boldsymbol{Z}_i \rightarrow \boldsymbol{Z}_i/\alpha$, equivalent to scaling weight matrices, and then scaling the LR, mitigates attention sink

$$\begin{aligned} \boldsymbol{W}_O^{s+1} &= \boldsymbol{W}_O^s - \eta \nabla_{\boldsymbol{W}_O^s} \mathcal{L}(\alpha \boldsymbol{W}_O^s) \\ &= \boldsymbol{W}_O^s - \alpha\eta \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W})|_{\boldsymbol{W}=\alpha \boldsymbol{W}_O^s}, \end{aligned}$$

$$\begin{aligned} \hat{\boldsymbol{W}}_O^{s+1} &= \hat{\boldsymbol{W}}_O^s - \eta' \nabla_{\hat{\boldsymbol{W}}_O^s} \mathcal{L}(\hat{\boldsymbol{W}}_O^s) \\ &= \alpha \boldsymbol{W}_O^s - \eta' \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W})|_{\boldsymbol{W}=\alpha \boldsymbol{W}_O^s}, \end{aligned}$$

Effects of Normalization in Softmax Attention

Power of sum to one: may mitigate attention sink but does not prevent, sensitive to LR, large LR may incentivize attention sink

$$\boldsymbol{v}_i^\dagger = \frac{\sum_{j=1}^{i} \mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_j))\boldsymbol{v}_j}{\left(\sum_{j'=1}^{i} \mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_{j'}))^p\right)^{\frac{1}{p}}}$$

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

Effects of Normalization in Softmax Attention

- Removing the normalization in Softmax attention

Using sigmoid attention (exponential kernel in Softmax tends to explode)

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

Or ELU plus one attention

No normalization -> No attention sink; add back - > attention sink

Effects of Normalization in Softmax Attention

Other attention variants

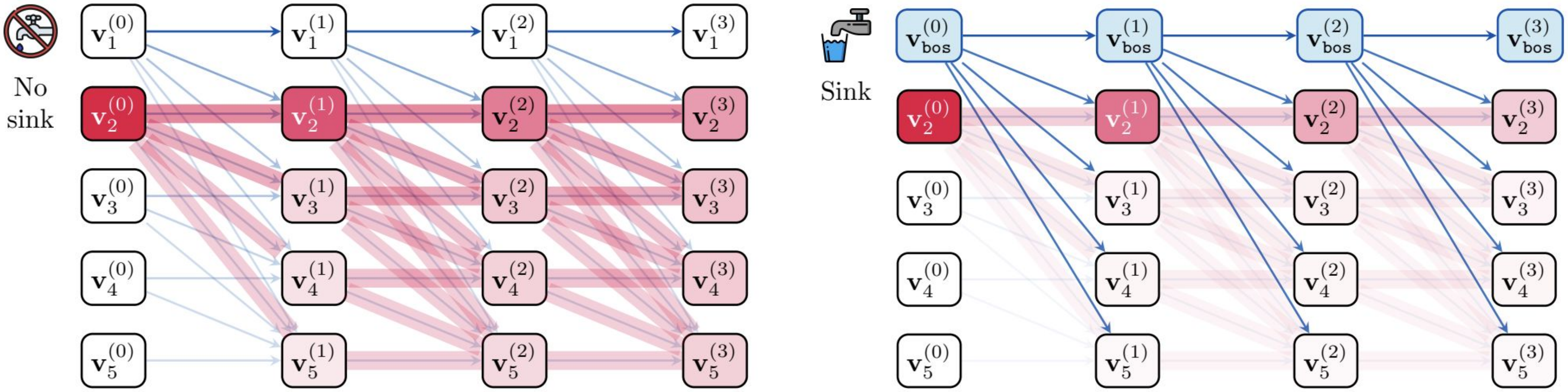
$\mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_j))$	$\boldsymbol{Z}_i$	$\mathrm{Sink}_1^{\epsilon}(\%)$	valid loss
$\exp(\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}})$	$\sum_{j'=1}^{i} \exp(\frac{\boldsymbol{q}_i \boldsymbol{k}_{j'}^\top}{\sqrt{d_h}})$	18.18	3.73
$\mathrm{sigmoid}(\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}})$	1	0.44*	3.70
$\mathrm{sigmoid}(\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}})$	$\sum_{j'=1}^{i} \mathrm{sigmoid}(\frac{\boldsymbol{q}_i \boldsymbol{k}_{j'}^\top}{\sqrt{d_h}})$	30.24	3.74
$\mathrm{elu}(\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}}) + 1$	1	0.80*	3.69
$\mathrm{elu}(\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}}) + 1$	$\sum_{j'=1}^{i} \mathrm{elu}(\frac{\boldsymbol{q}_i \boldsymbol{k}_{j'}^\top}{\sqrt{d_h}}) + 1$	-	-
$\frac{(\mathrm{elu}(\boldsymbol{q}_i)+1)(\mathrm{elu}(\boldsymbol{k}_j)+1)^\top}{\sqrt{d_h}}$	$\sum_{j'=1}^{i} \frac{(\mathrm{elu}(\boldsymbol{q}_i)+1)(\mathrm{elu}(\boldsymbol{k}_{j'})+1)^\top}{\sqrt{d_h}}$	53.65*	4.19
$\frac{(\mathrm{elu}(\boldsymbol{q}_i)+1)(\mathrm{elu}(\boldsymbol{k}_j)+1)^\top}{\sqrt{d_h}}$	1	-	-
$\boldsymbol{q}_i \boldsymbol{k}_j^\top$	$\max\left(\left\lvert\sum^{i} \ldots \boldsymbol{q}_i \boldsymbol{k}_{j'}^\top\right\rvert, 1\right)$		

I am attempting to answer ...

- Mechanism understanding of Attention Sink?
- When Attention Sink Emerges in LLMs?
- Why LLMs need Attention Sink?
- Why GPT-OSS and Qwen3-Next consider Attention Sink in the Model Design?

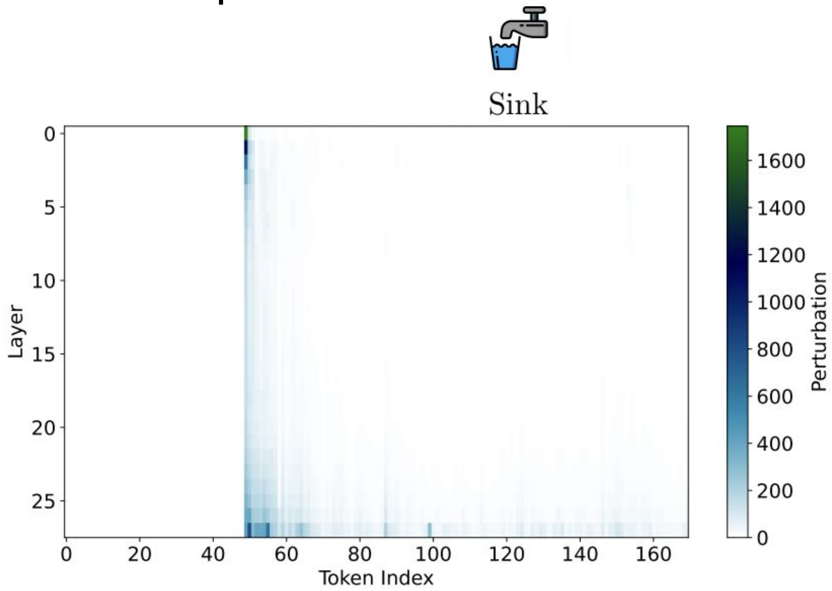
LLMs need attention sink to prevent over-mixing

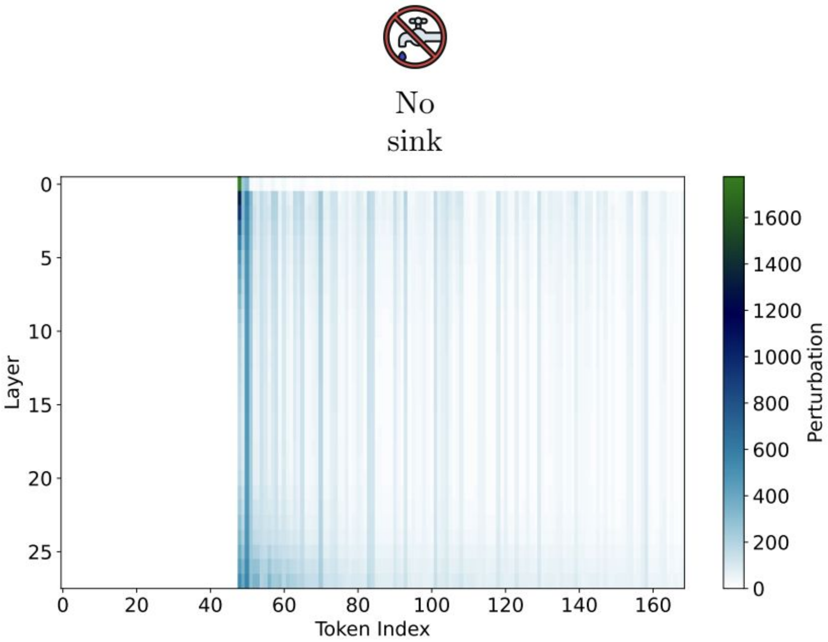
- Attention blocks try to mix representations
- Attention sink serves as a mechanism to prevent over-mixing (see the paper for theory, longer context needs stronger mechanism)



LLMs need attention sink to prevent over-mixing

With attention sink, perturbation on one token (“greatest”->”best”) won’t change token representations a lot



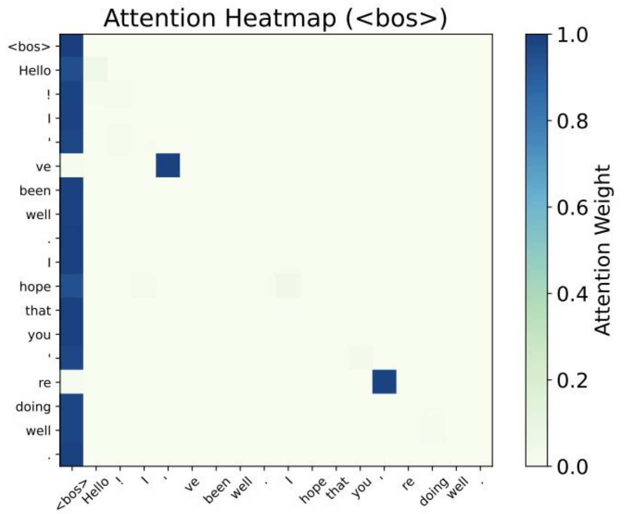


Attention sink implements “no-op”

- Attention sink approximates “no-op”: either sharply to attend one important token or attend to the first token
- From the representation mixing perspective, LLMs need “no-op” to prevent over-mixing

$$\boldsymbol{v}_i^\dagger = \sum_{j=1}^{i} \boldsymbol{\alpha}_{ij}\boldsymbol{v}_j = \boldsymbol{\alpha}_{i1}\boxed{\boldsymbol{v}_1} + \sum_{j\neq 1}^{i}\boxed{\boldsymbol{\alpha}_{ij}}\boldsymbol{v}_j$$

Small

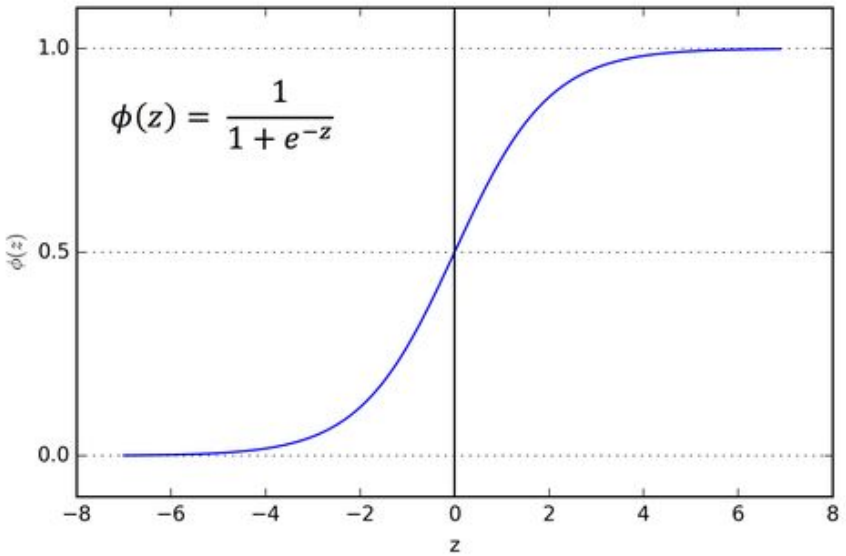


Interpreting attention variants using “no-op”

Sigmoid attention allows approximate

zero attention

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



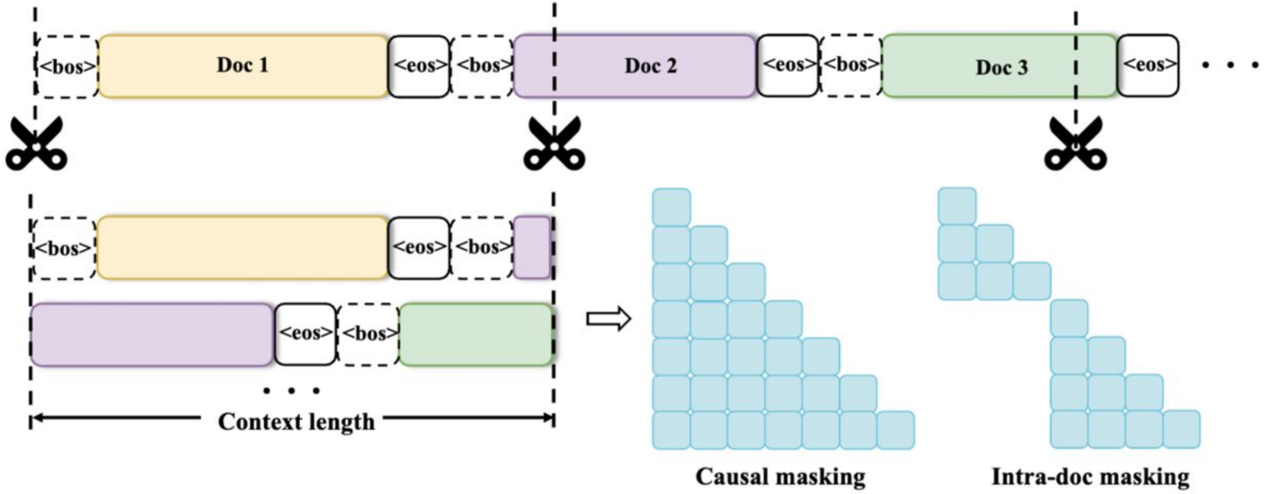
Interpreting attention variants using "no-op"

The following linear attention could have all zero attention scores

$\mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_j))$	$\boldsymbol{Z}_i$	$\mathrm{Sink}_1^{\epsilon}(\%)$	valid loss
$\exp(\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}})$	$\sum_{j'=1}^{i} \exp(\frac{\boldsymbol{q}_i \boldsymbol{k}_{j'}^\top}{\sqrt{d_h}})$	18.18	3.73
$\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}}$	$\max\left(\left\lvert\sum_{j'=1}^{i} \frac{\boldsymbol{q}_i \boldsymbol{k}_{j'}^\top}{\sqrt{d_h}}\right\rvert, 1\right)$	-	-
$\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}}$	1	0.00*	3.99
$\frac{\mathrm{mlp}(\boldsymbol{q}_i)\mathrm{mlp}(\boldsymbol{k}_j)^\top}{\sqrt{d_h}}$	$\max\left(\left\lvert\sum_{j'=1}^{i} \frac{\mathrm{mlp}(\boldsymbol{q}_i)\mathrm{mlp}(\boldsymbol{k}_{j'})^\top}{\sqrt{d_h}}\right\rvert, 1\right)$	0.19*	3.85
$\frac{\mathrm{mlp}(\boldsymbol{q}_i)\mathrm{mlp}(\boldsymbol{k}_j)^\top}{\sqrt{d_h}}$	1	0.74*	3.91

When Attention Sink Attaches to <BOS>

Data packing (fixed <BOS> in the first position will have similar behavior as Gemma)

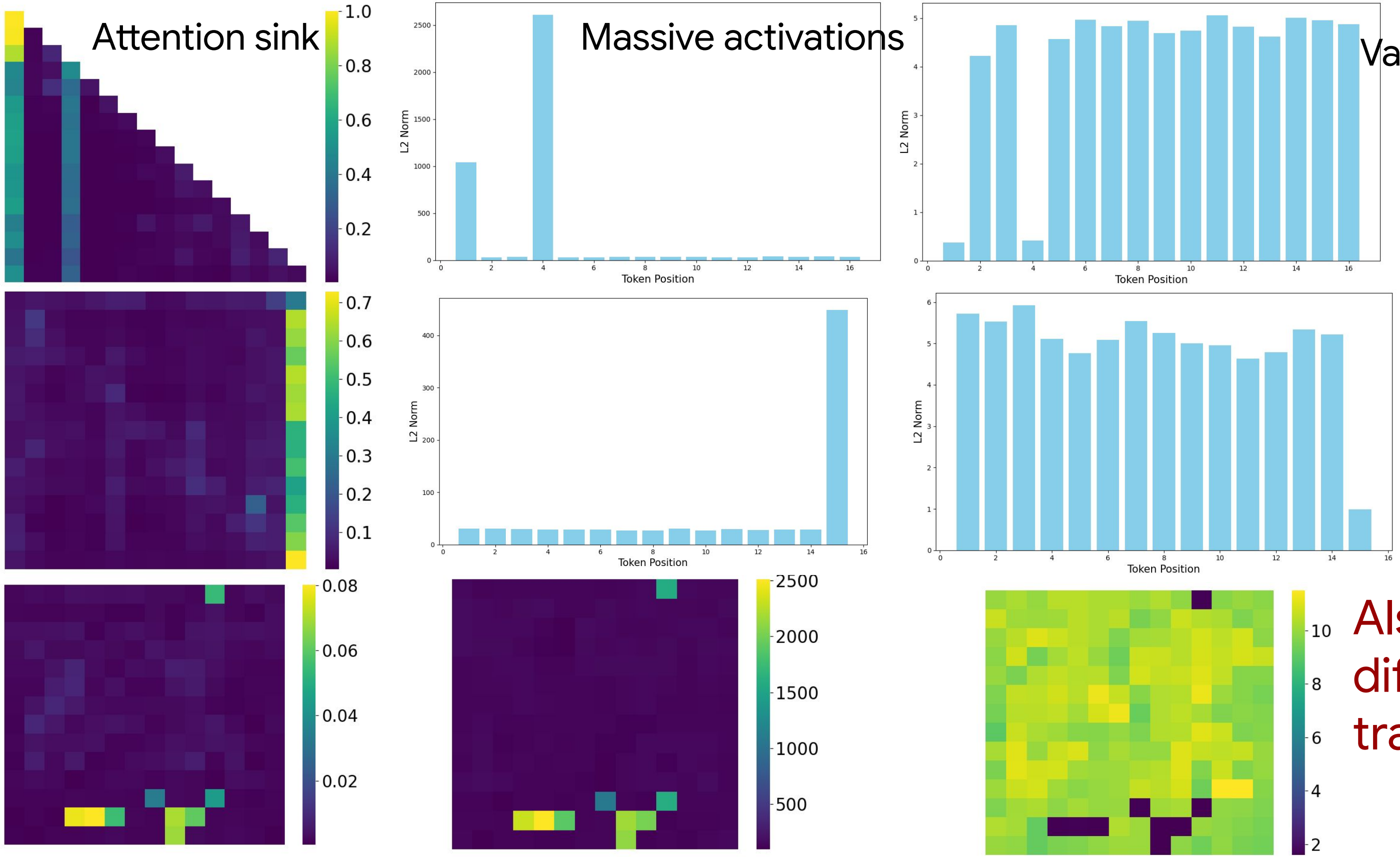


Attention sink / “No-op” widely exists in Transformer family

LLaMA

BERT

ViT



Also appear i
diffusion
transformers

I am attempting to answer ...

- Mechanism understanding of Attention Sink?
- When Attention Sink Emerges in LLMs?
- Why LLMs need Attention Sink?
- Why GPT-OSS and Qwen3-Next consider Attention Sink in the Model Design?

GPT-OSS adopts Attention Biases

- Learnable key biases, zero value biases

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

- Softmax off-by-one

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

- Learnable attention score biases (single number for each head, layer)

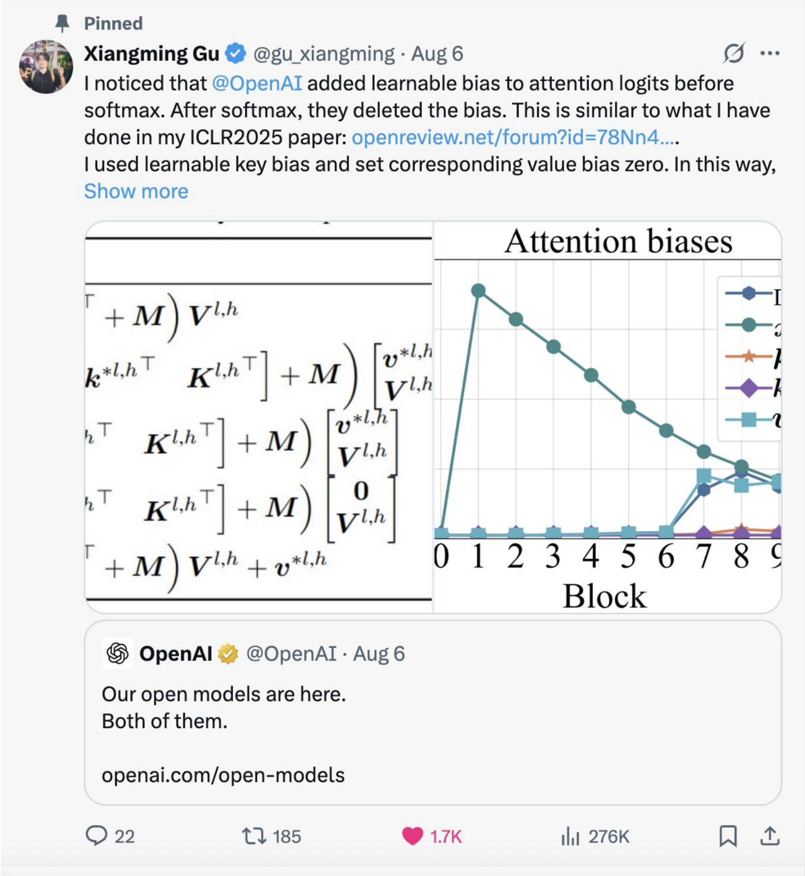
$$\text{Softmax}\left(\frac{1}{\sqrt{d_h}}\begin{bmatrix}\boldsymbol{b}^{*l,h\top} & \boldsymbol{Q}^{l,h}\boldsymbol{K}^{l,h\top}\end{bmatrix} + \boldsymbol{M}\right)\begin{bmatrix}\mathbf{0} \\ \boldsymbol{V}^{l,h}\end{bmatrix}$$

$$\boldsymbol{b}^{*l,h} = b^{*l,h}[1, 1, 1, .., 1]$$

GPT-OSS adopts Attention Biases

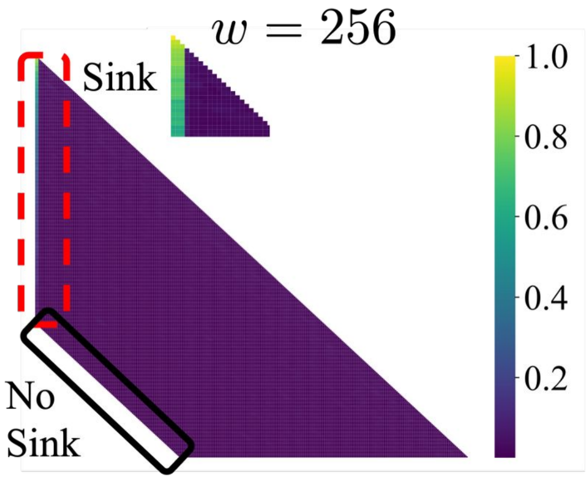
The first token does not to develop strong attention sink, thus mitigating massive activations/outliers

Benefits 1: facilitate quantization, pre-training stability



GPT-OSS adopts Attention Biases

- Attention sink only happens in ***absolute*** first token, not ***relative*** first token



- Tokens beyond window size have no sinks to attend, possible over-mixing

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

- Facilitate long context, especially in LLMs with alternative shifted window / full attention

Qwen3-Next adopts Gated Attention

$$\mathrm{Sigmoid}(\boldsymbol{G}^{l,h}) \odot \left[\mathrm{Softmax}\left(\frac{1}{\sqrt{d_h}}\boldsymbol{Q}^{l,h}{\boldsymbol{K}^{l,h}}^{\top} + \boldsymbol{M}\right)\boldsymbol{V}^{l,h}\right]$$

Transformations of inputs

Sigmoid gate allows "no-op", no need to only rely on attention sink for "no-op" → No attention sink, massive activations, better long context, pre-training stability

Zihan et al. Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free. NeurIPS 2025 Best Paper Award

Thank you for listening!