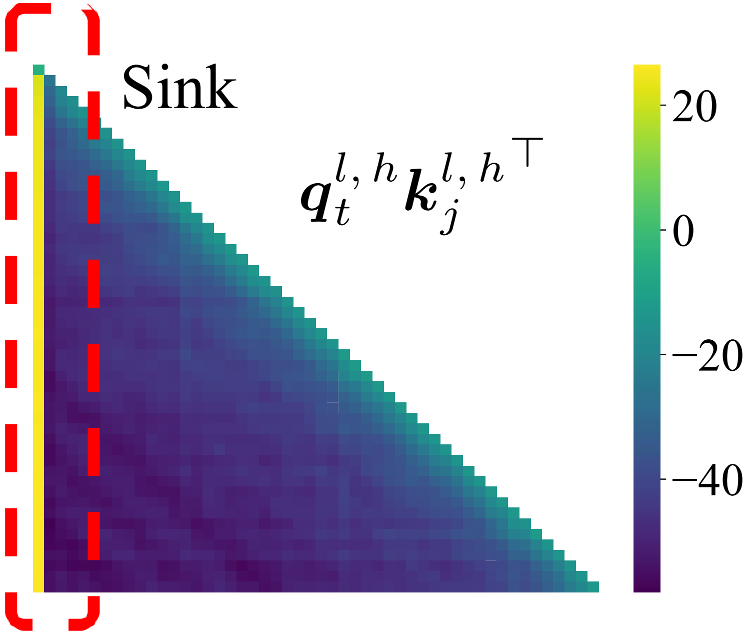
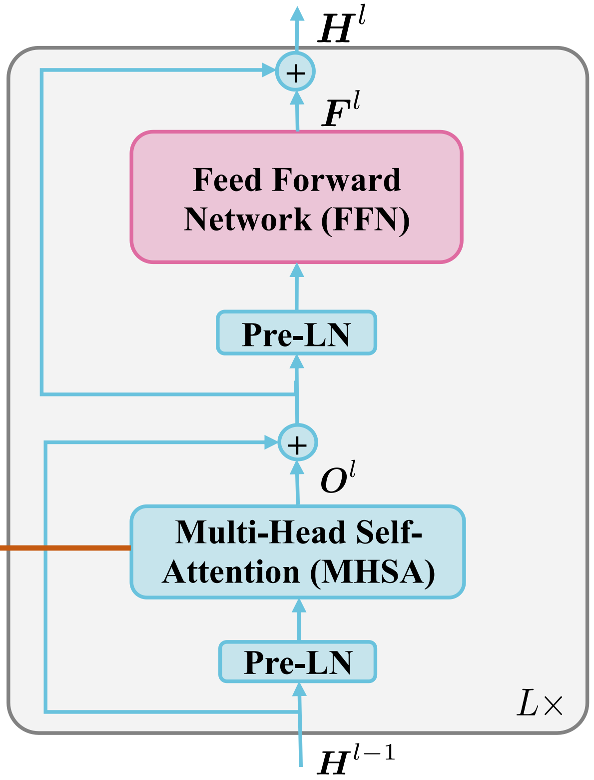
When Attention Sink Emerges in Language Models: An Empirical View

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu,
Fengzhuo Zhang, Cunxiao Du, Ye Wang, Min Lin

What is attention sink?

- Attention sink refers to that Language Models (LMs) assign significant attention to the first token (Xiao et al. 2024)





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

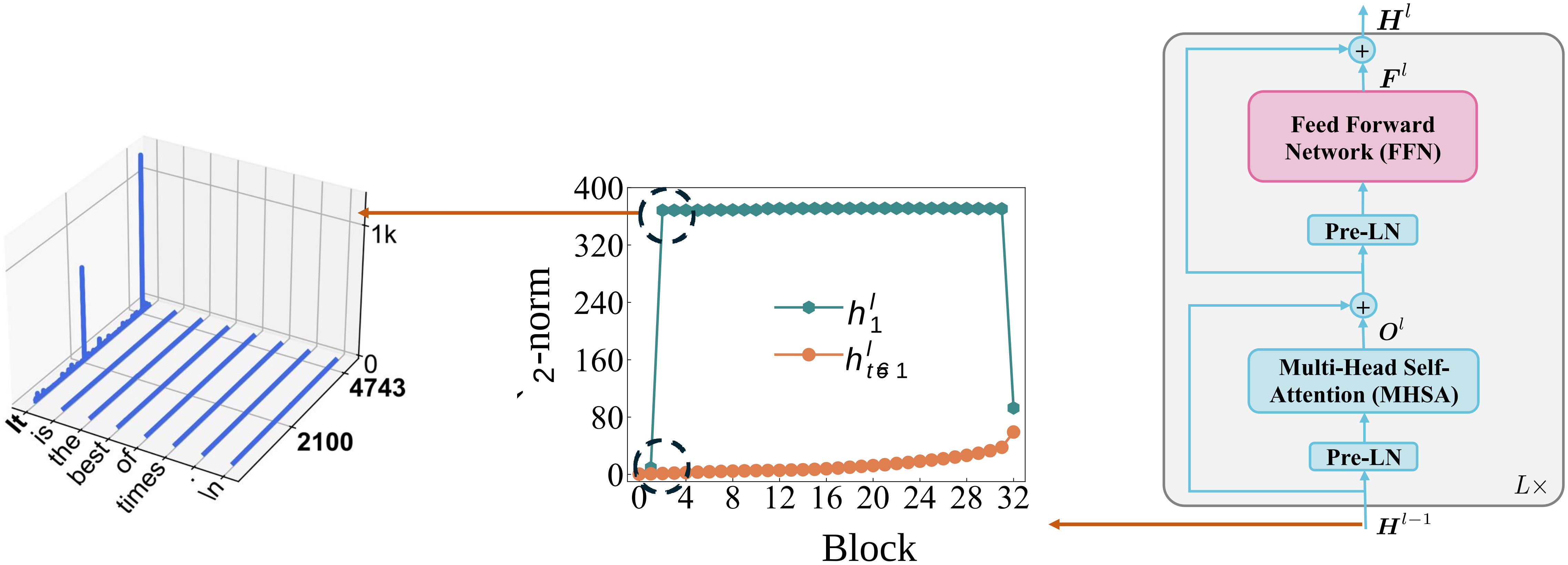
What can we do with attention sink?

- Long context understanding / generation
- KV cache compression
- Model quantization
- etc

Attention sink is important!

Mechanism of attention sink

- 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

Mechanism of attention sink

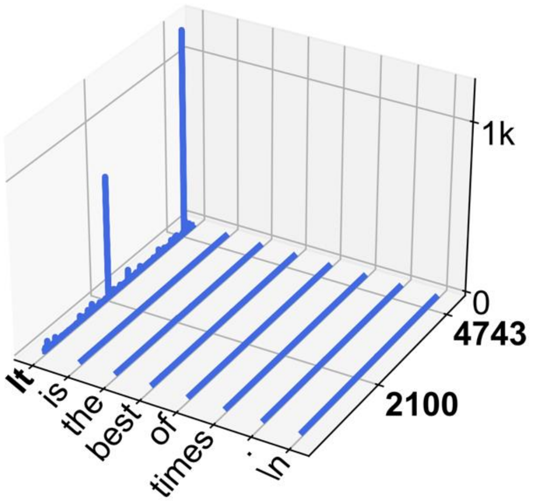
- What is the connection between attention sink and massive activatavations?

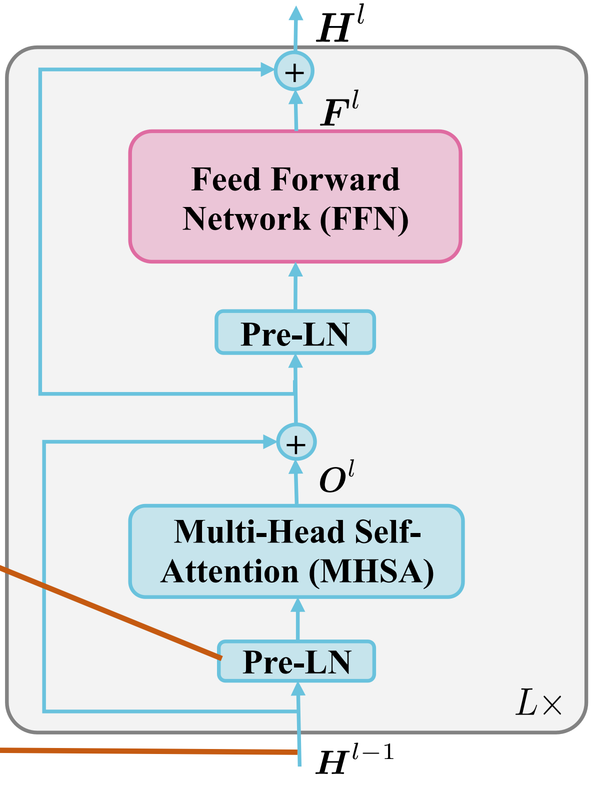
Layer norm retains values for spike dimensions for key of sink token

Key of the sink token is distributed in a different manifold (low-rank)

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

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

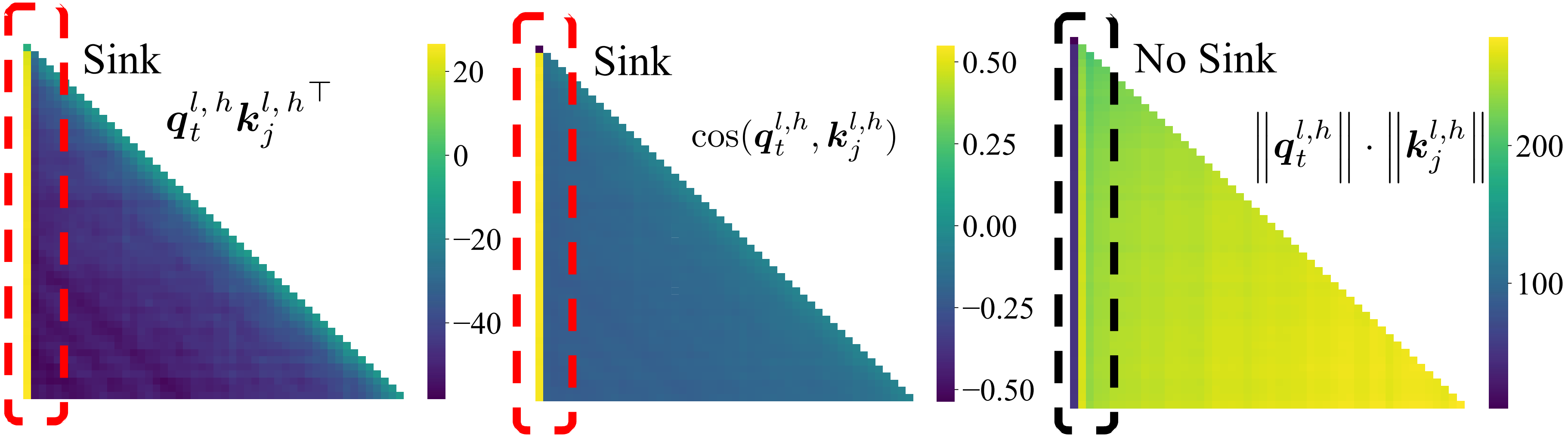




Mechanism of attention sink

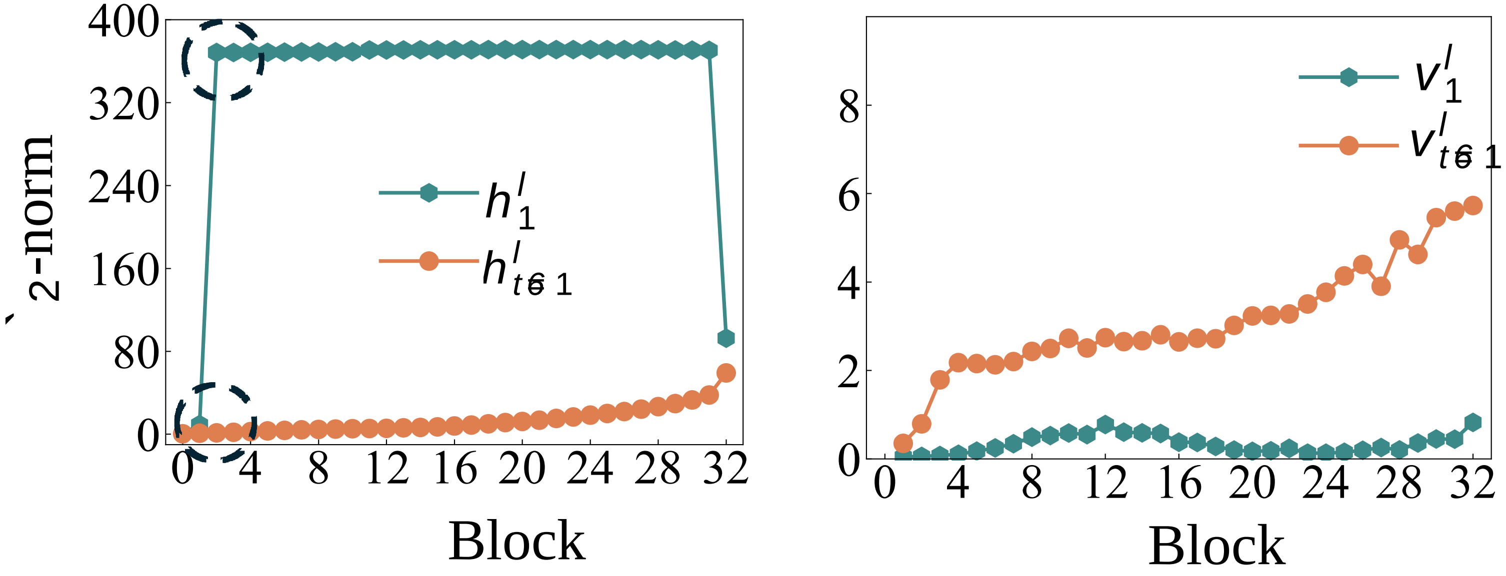
- What's the result of "first key is distributed in a different manifold"

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



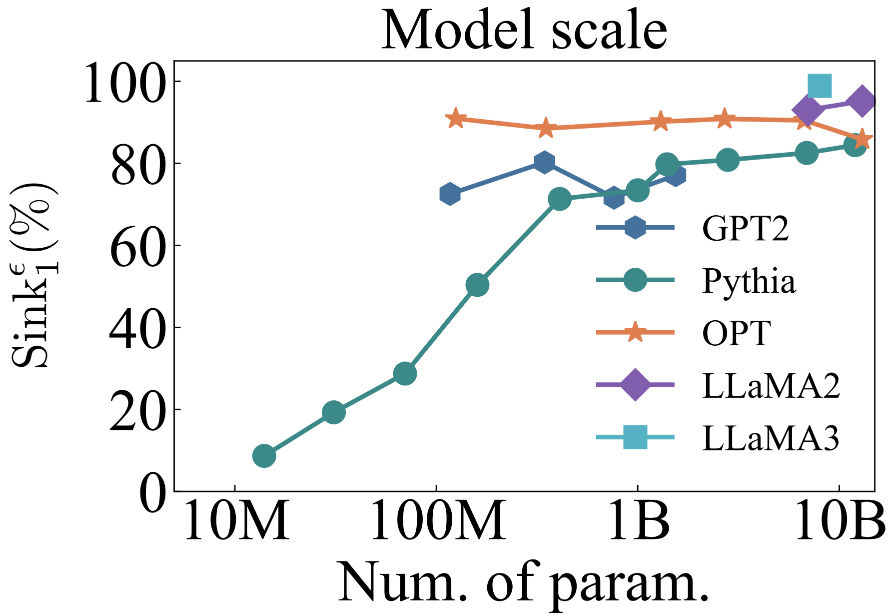
Mechanism of attention sink

- Another property of sink token: small values



Attention sink widespread appears in pre-trained LMs

- Attention sink appears widespread in various LMs, even in LMs with14M params.



- Sink metric refers to the number of attention sink heads in the whole LM

- Attention sink emerges in LM pre-training

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

Attributing attention sink to LM pre-training

- LM pre-training objective $$\min_{\theta} \mathbb{E}_{\boldsymbol{X} \sim p_{\text{data}}} \left[\mathcal{L} \left(p_{\theta}(\boldsymbol{X}) \right) \right]$$
- Experiments on LLaMA2-style models: Attributing model behavior to training recipes

Optimization

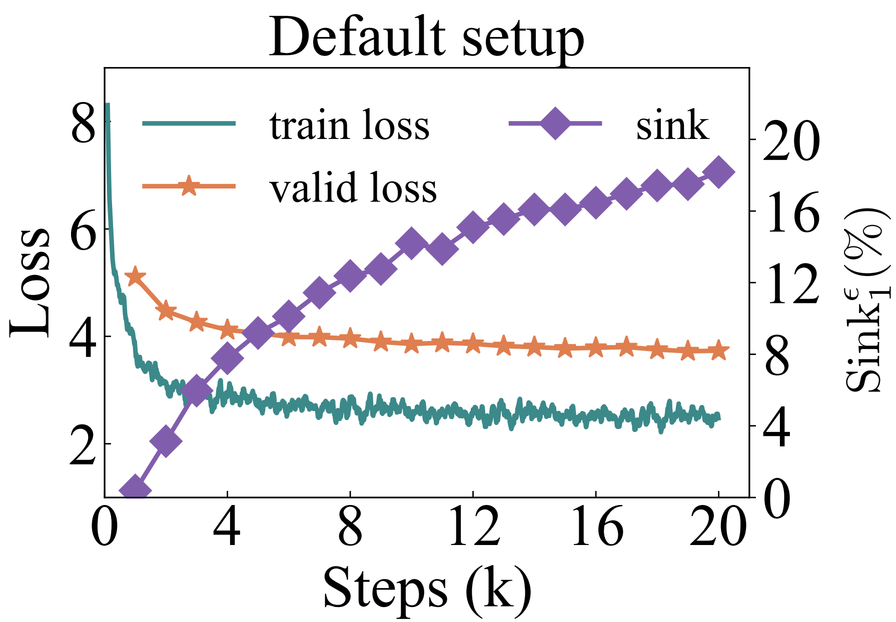
Data distribution

Loss function

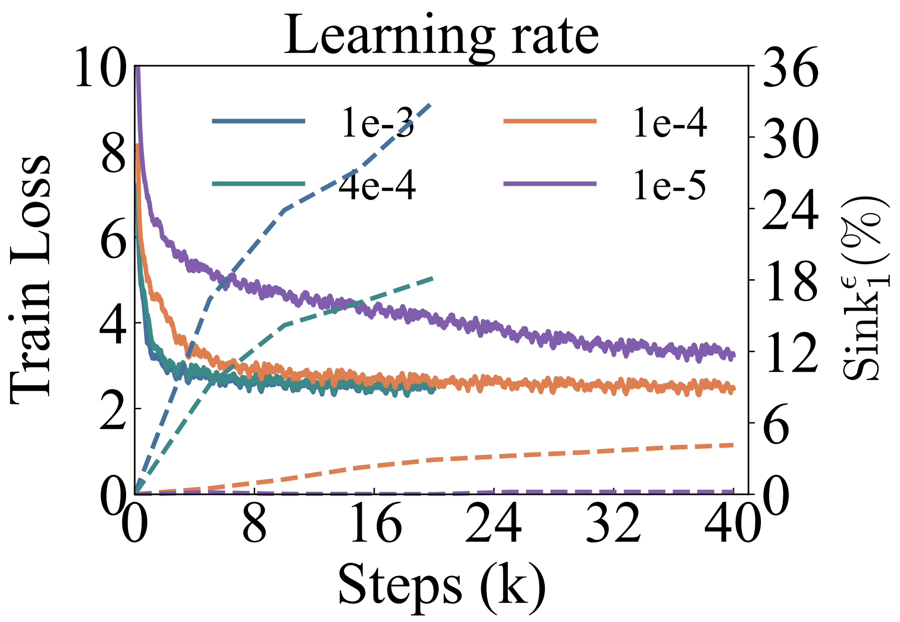
Model architecture

Effects of optimization on attention sink

- Training steps



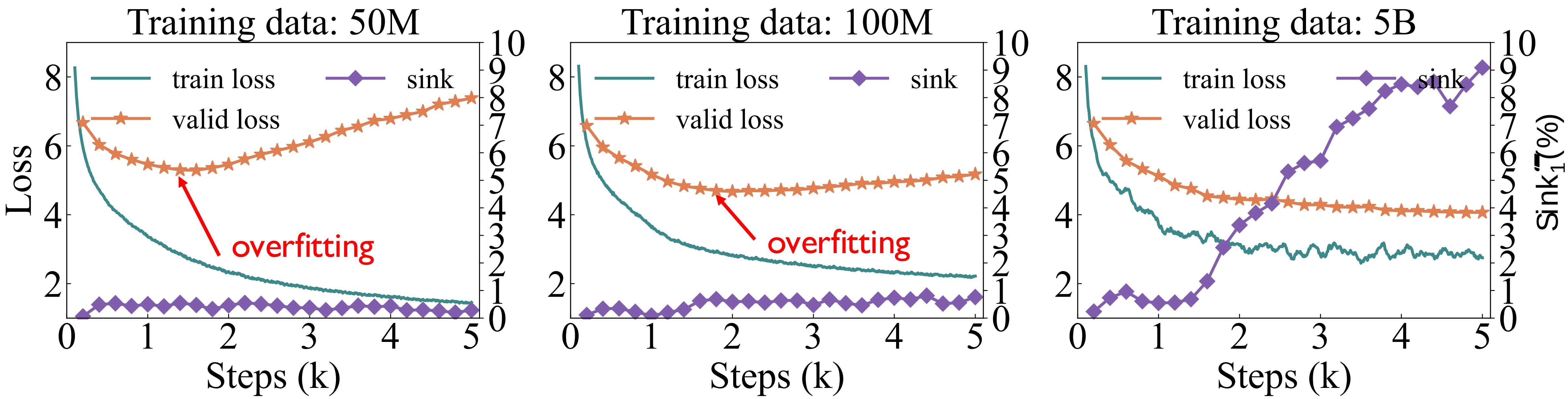
- Learning rate
- Small learning rates mitigate attention sink



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 loss function on attention sink

- Auto-regressive loss

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

- Weight decay

L2 regularization

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

Larger weight decay encourages attention sink

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 model architecture on attention sink

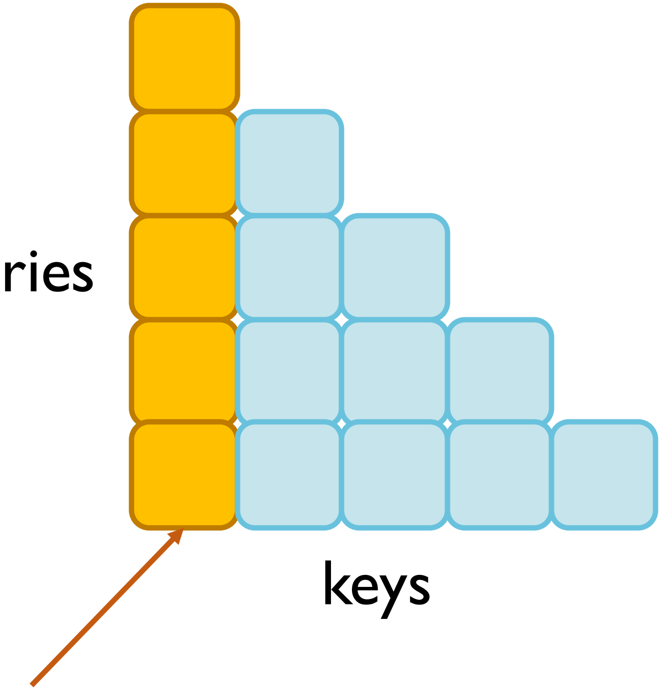
The following designs do not affect the emergence of attention sink

- Positional embeddings: including no positional embedding
- Pre-norm and post-norm design
- 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

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



Effects of model architecture on attention sink

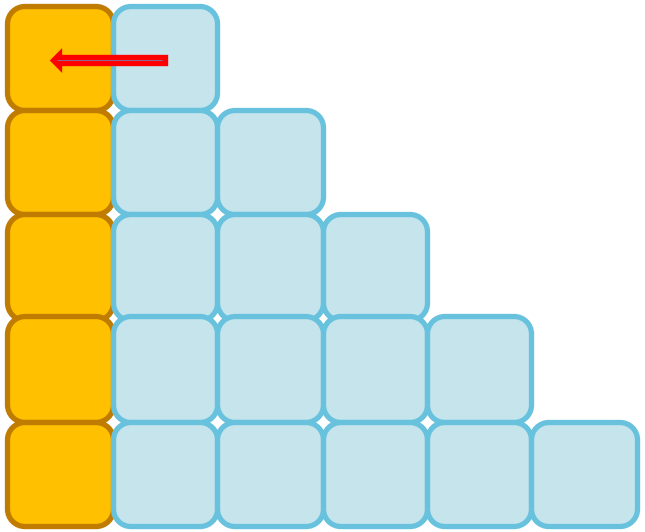
Softmax attention with learnable K biases

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

V biases are all zeros

Learnable K biases

queries



keys

Sink on the learnable K bias

Effects of model architecture on attention sink

- LM with K biases has no massive activations!

- 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

$$\mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_j)) = \exp(\frac{\boldsymbol{q}_i \boldsymbol{k}_j^\top}{\sqrt{d_h}})$$

softmax

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

$$\boldsymbol{Z}_i = \sum_{j'=1}^{i} \mathrm{sim}(\varphi(\boldsymbol{q}_i), \varphi(\boldsymbol{k}_j))$$

normalization term

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

Effects of model architecture on attention sink

- Relax tokens' inner dependence by removing normalization

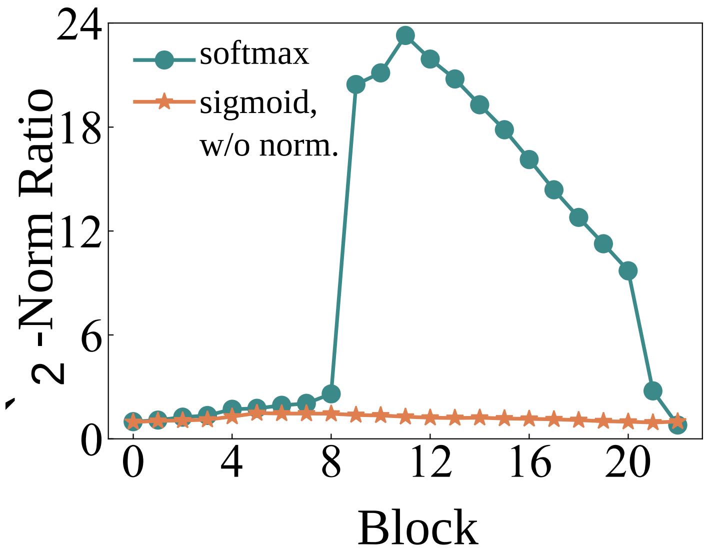
Sigmoid attention:

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

ELU plus one attention:

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

Scale up to 1B LLMs



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

$$\boldsymbol{v}_i^{\dagger} = \sum_{j=1}^{i} \frac{\mathrm{mlp}(\boldsymbol{q}_i)\mathrm{mlp}(\boldsymbol{k}_j)^\top}{\sqrt{d_h}} \boldsymbol{v}_j$$

-> No attention sink, no massive activations

Add a normalization

$$\boldsymbol{Z}_i = \max\left(\left|\sum_{j'=1}^{i} \frac{\mathrm{mlp}(\boldsymbol{q}_i)\mathrm{mlp}(\boldsymbol{k}_{j'})^\top}{\sqrt{d_h}}\right|, 1\right)$$

-> No attention sink, no massive activations

Takeaway

- Attention sink is a widespread phenomena across models and input
- Attention sink emerges during the LM pre-training
- Attention sink acts as key biases, storing extra attention and non-informative
- Softmax plays an important role in the emergence of attention sink

Please check our paper to see more interesting results!