Your Next-Token Prediction and Transformers Are **Biased** for Long-Context Modeling

Yifei Wang
MIT

Joint work with:
Lizhe Fang, Xinyi Wu, Zhaoyang Liu, Chenheng Zhang, Jinyang Gao, Bolin Ding,
Yisen Wang, Stefanie Jegelka, Ali Jadbabaie

LLMs ≈ Next-Token Prediction + Transformers

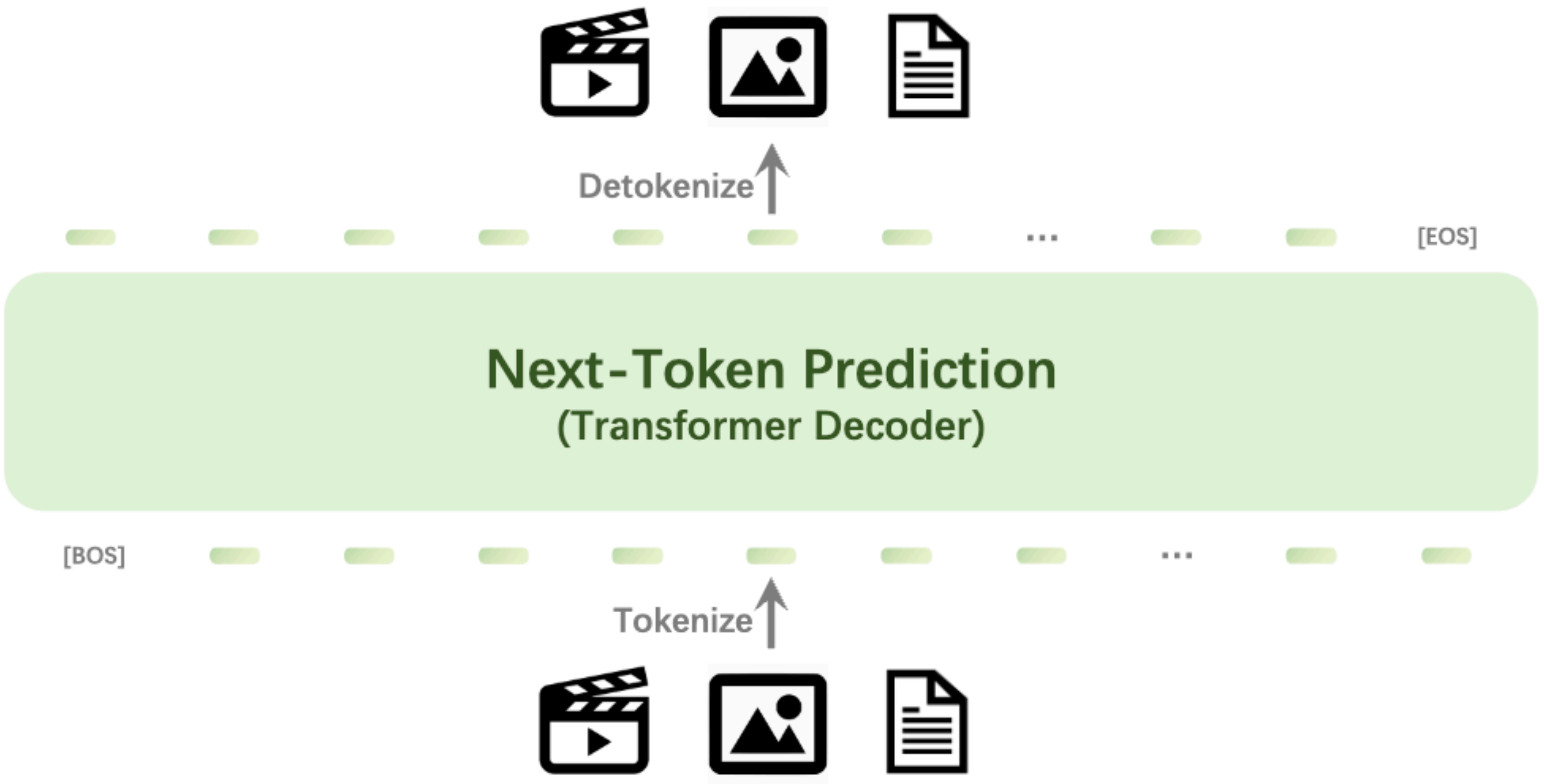


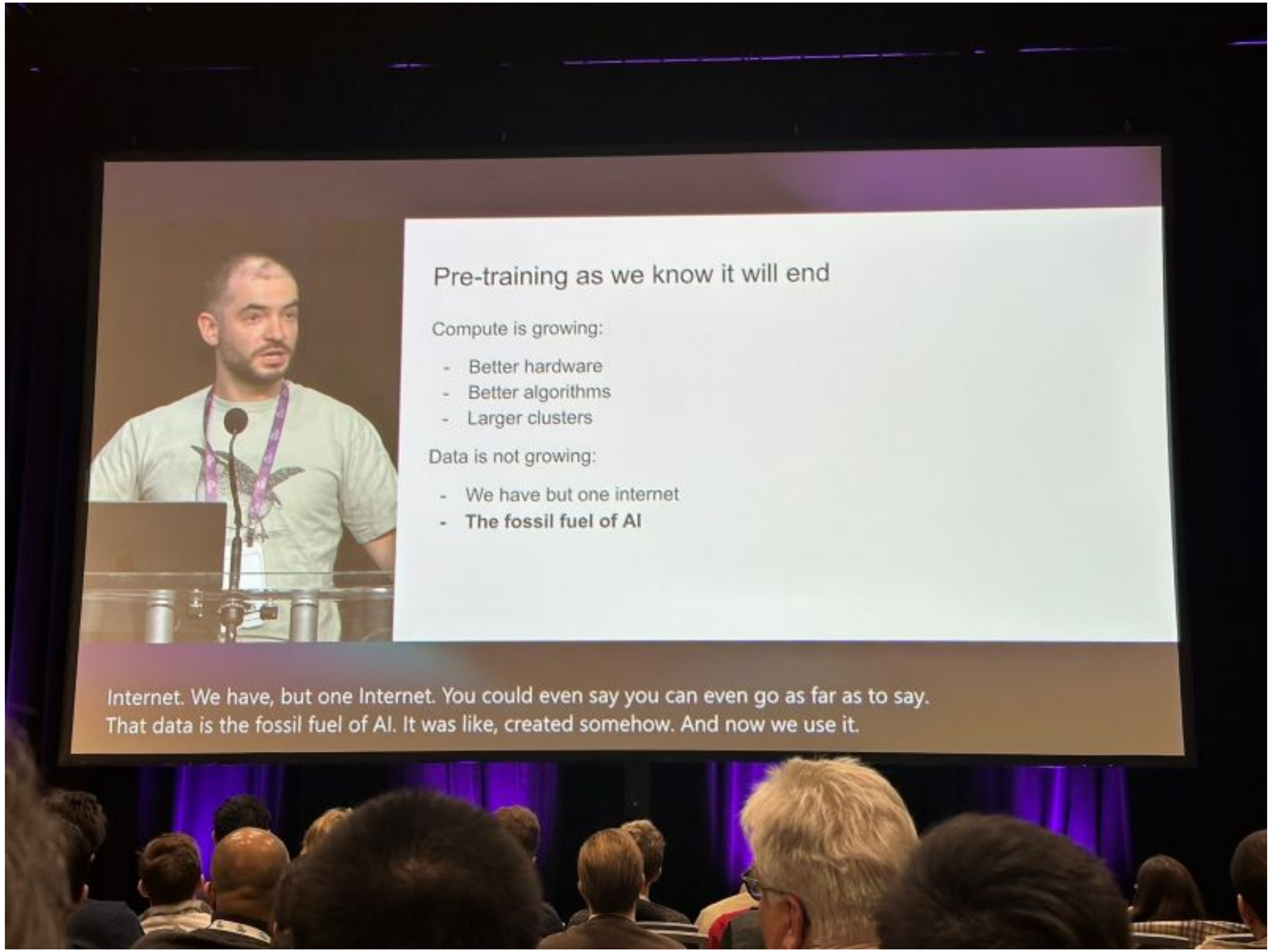
Fig from Emu3: Next-Token Prediction is All You Need

A very powerful paradigm that gets us so far

A great example of "minimum innovation, maximum results" (Ilya Sutskever, c.f. Pieter Abbeel)

<u>Many post-training methods are simple variants of next-token prediction</u>

Overaching Challenges



Next-token prediction may be having a diminishing return

which, to be fair, makes sense because LLMs are already fairly good

<u>Can we bend the curve on remaining challenging tasks?</u>

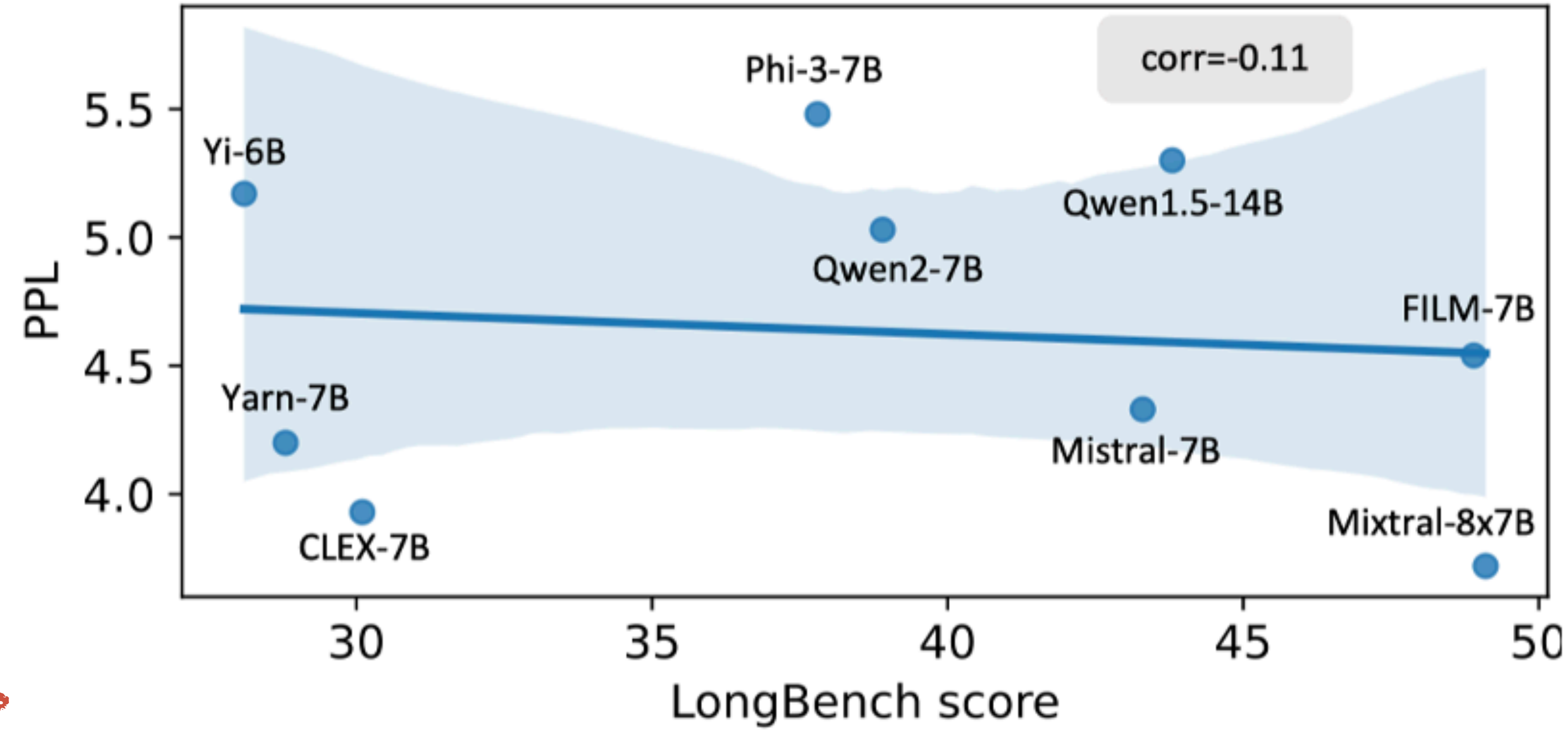
This Talk

- What is Wrong with Next-token Prediction (LongPPL, ICLR'25)

- What is Wrong with Transformers (Emergence of Postion Bias, ICML'25)

One remaining challenge: long-context understanding

One potential reason: no (significant) correlation between perplexity (NTP) and long-context performance



Our intuition:
1. PPL is good enough
2. We could fix it!

Human-designed benchmarks:
eg RULER, LongEval, LongBench

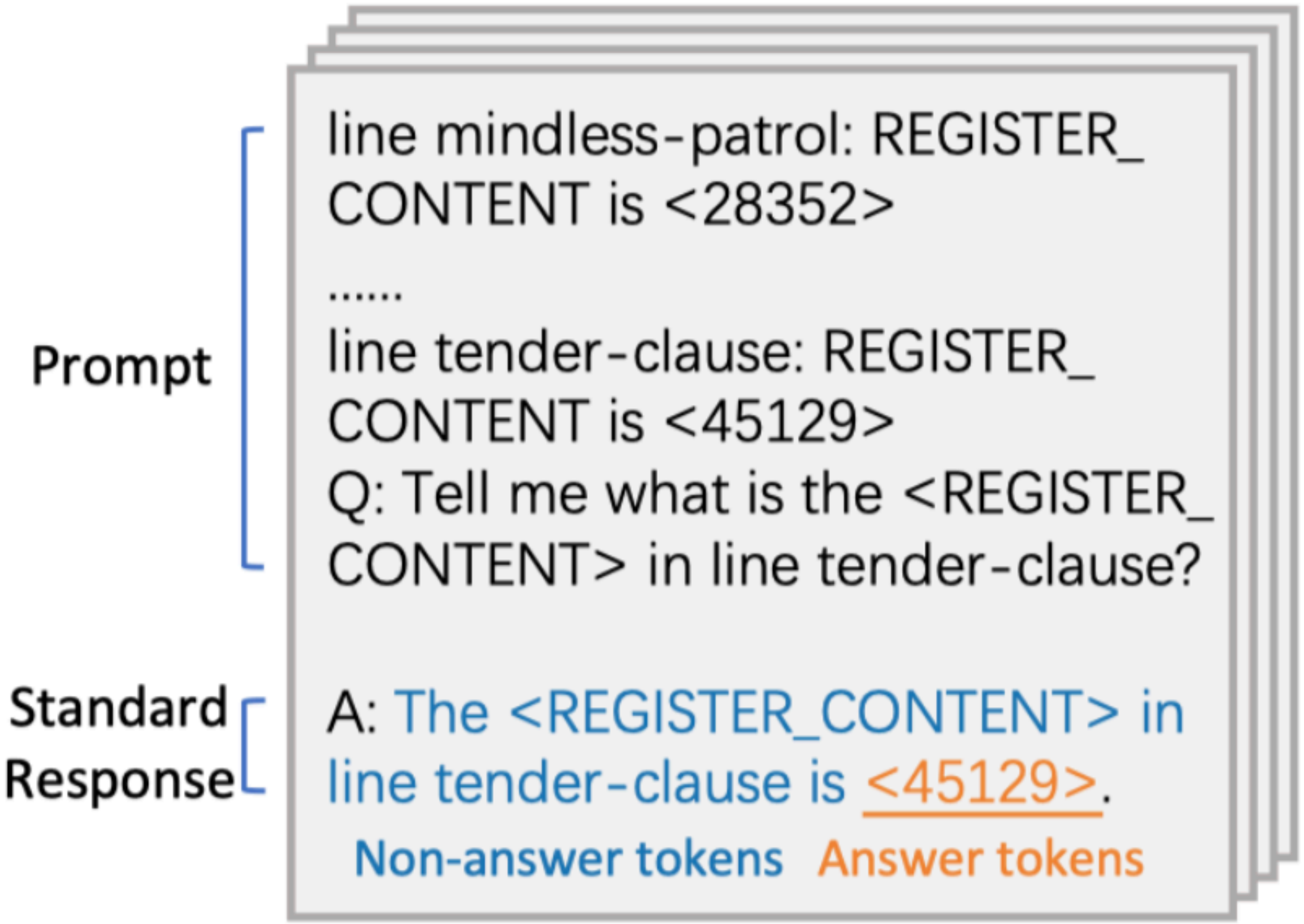
Implications:

1. Perplexity is not a reliable metric for evaluation
2. NTP is not an efficient metric to optimize for

More data & human curation

"Human, All Too human"

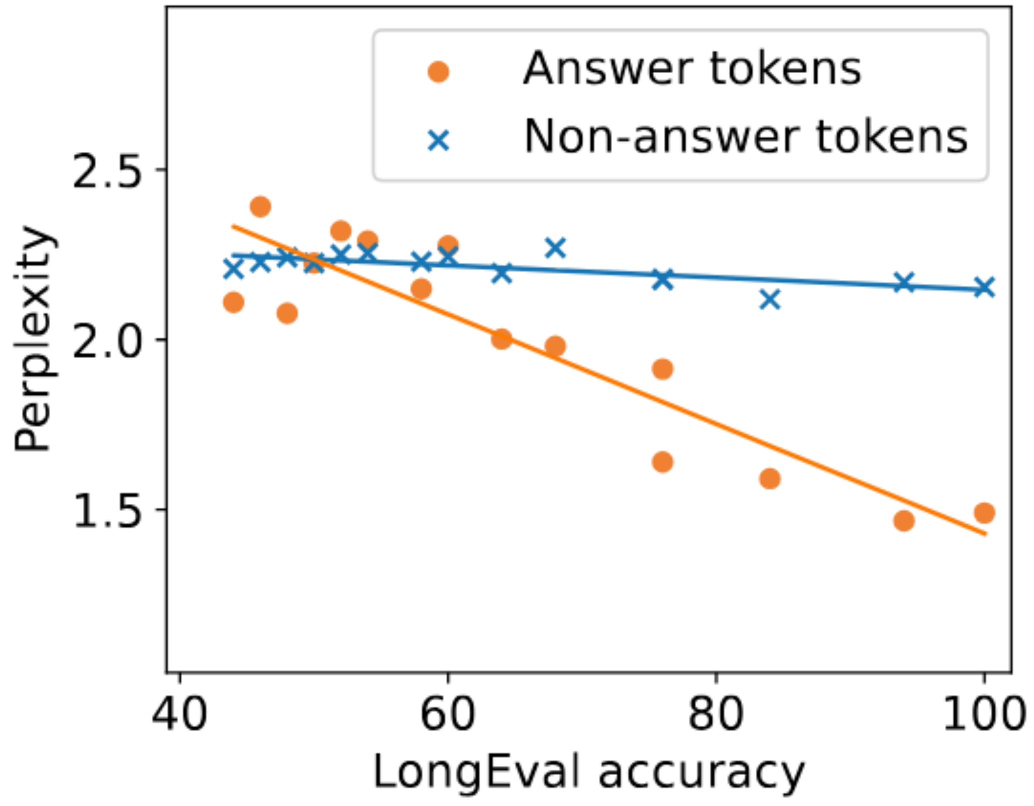
When we curate data/benchmarks, what are we curating?



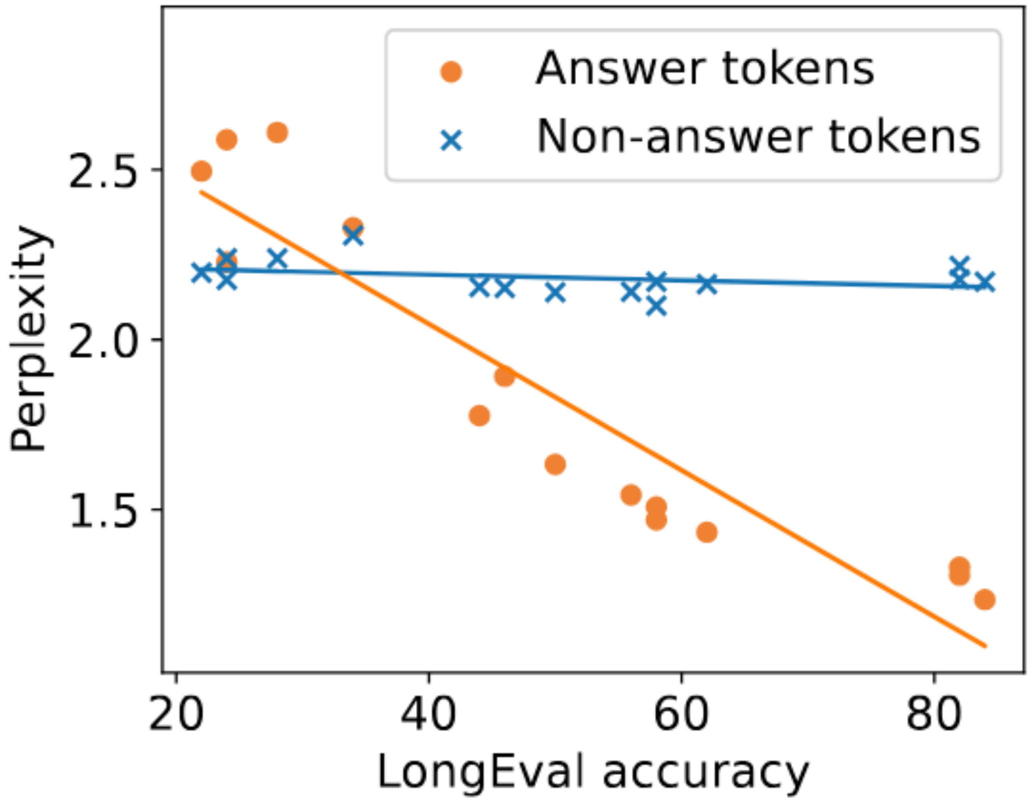
An example from LongEval

LongEval is saying: **Some tokens are more critical than others and we should focus on those**

Perplexity on "key tokens"



(b) PPL vs LongEval (Yi-6B)



(c) PPL vs LongEval (CLEX-7B)

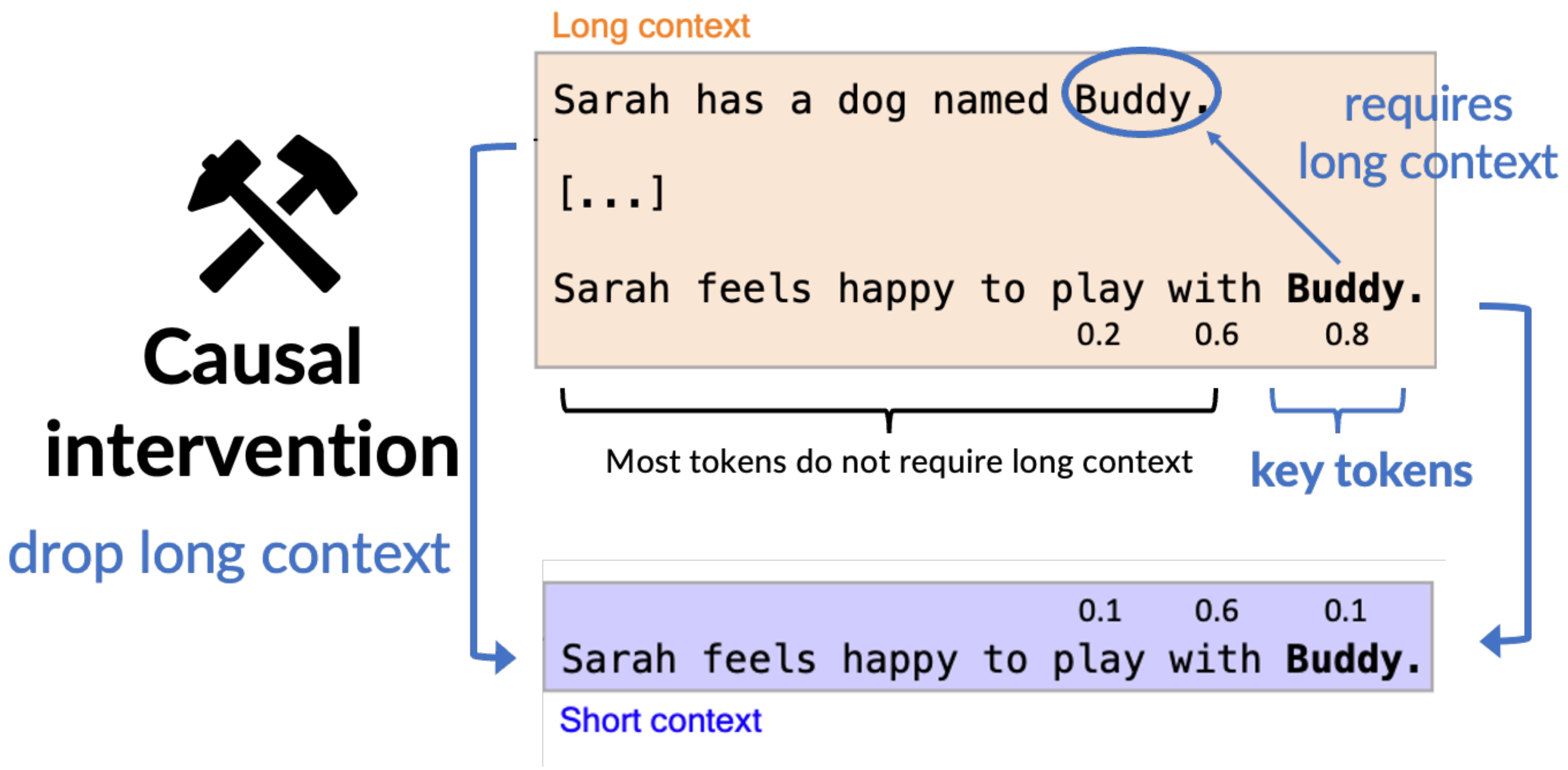
We recover Perplexity's correlation when evaluated only on answer tokens

Implications:

1. Perplexity / next-token prediction is **not** the problem
2. Why we really need humans -> **select the tokens** that we really care about for a task

The only technical question left

- How to identify key tokens from natural data without humans? An SSL problem!
- Lesson: find tokens reflecting the model's ability on long context



Log Probability Gain (LPG)

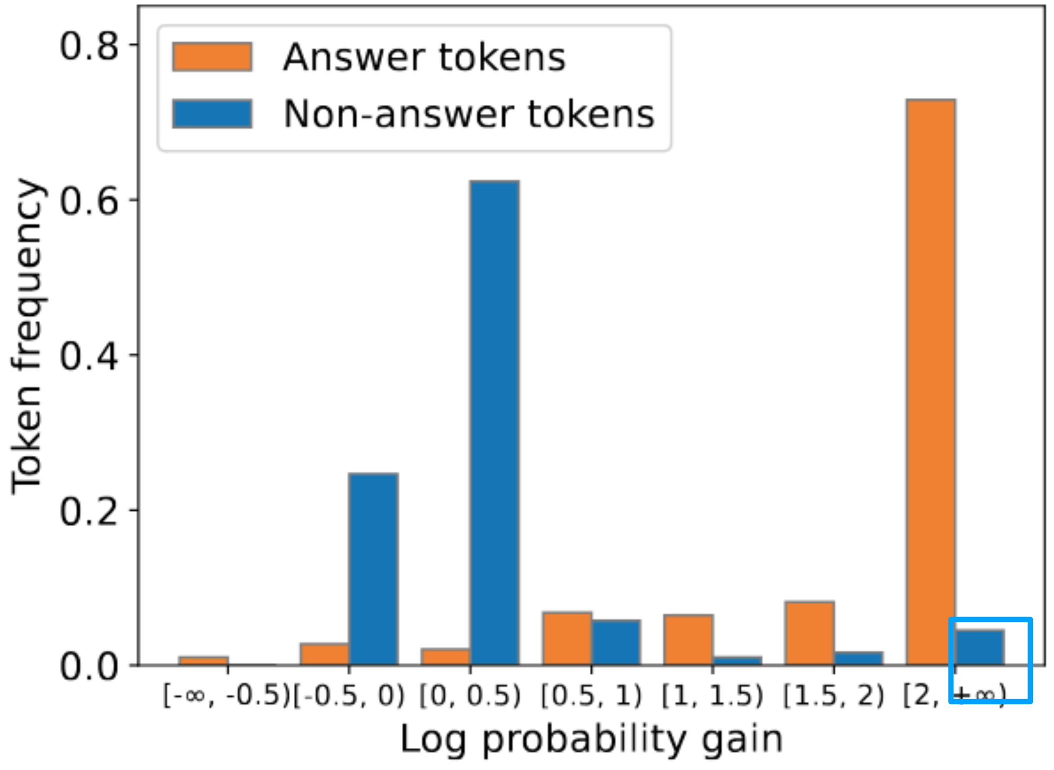
$$r(x_i) = \frac{P_\theta(x_i | l_i)}{P_\theta(x_i | s_i)}$$

By selecting a good threshold, LPG can identify key tokens with **85.6% acc** on LongEval

A little more technical nauance (optional)

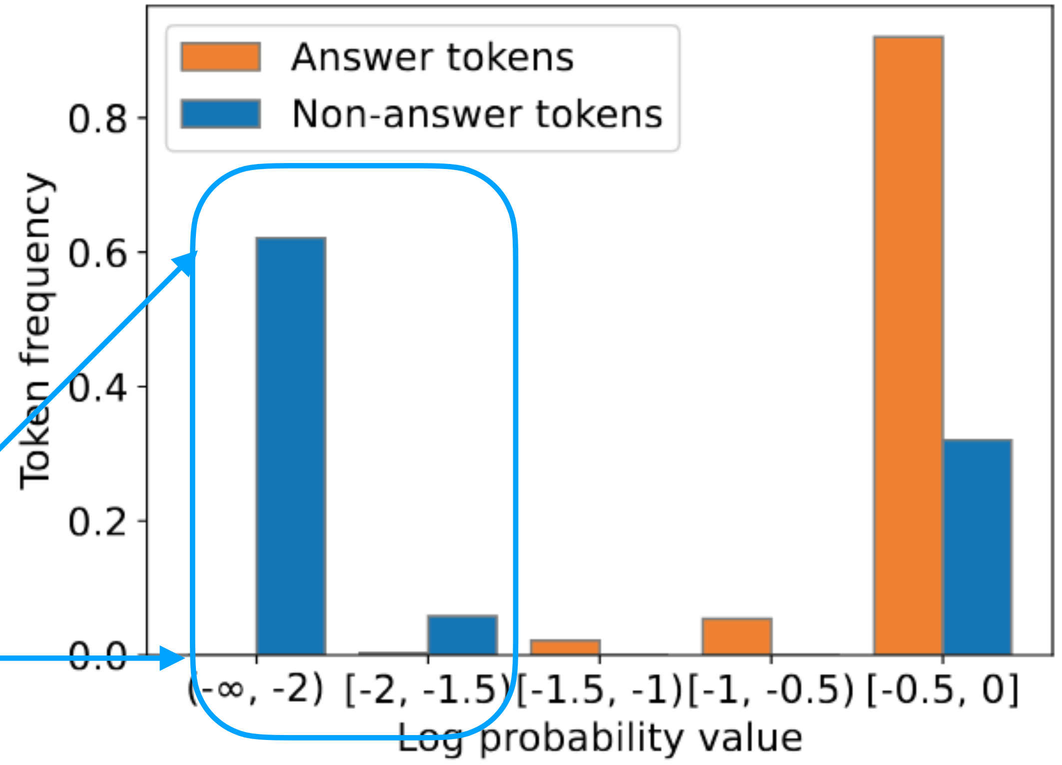
- What about the remaining 14.4%?

Some non-answer tokens also have high LPG



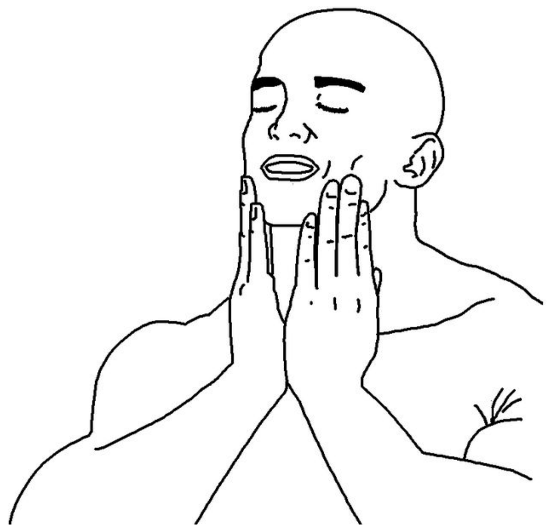
(a) LPG of tokens on LongEval.

These tokens often have low log prob. values (LPV) - hard to fit



(b) LPV of tokens on LongEval with large LPG

Combining LPV and LPG critiria, we can predict key tokens with **98.2%** accuracy!



From PPL to LongPPL

- PPL: calculated on a uniform avg of all tokens
- Long-context Perplexity (LongPPL): calculating perplexity on **filtered key tokens**

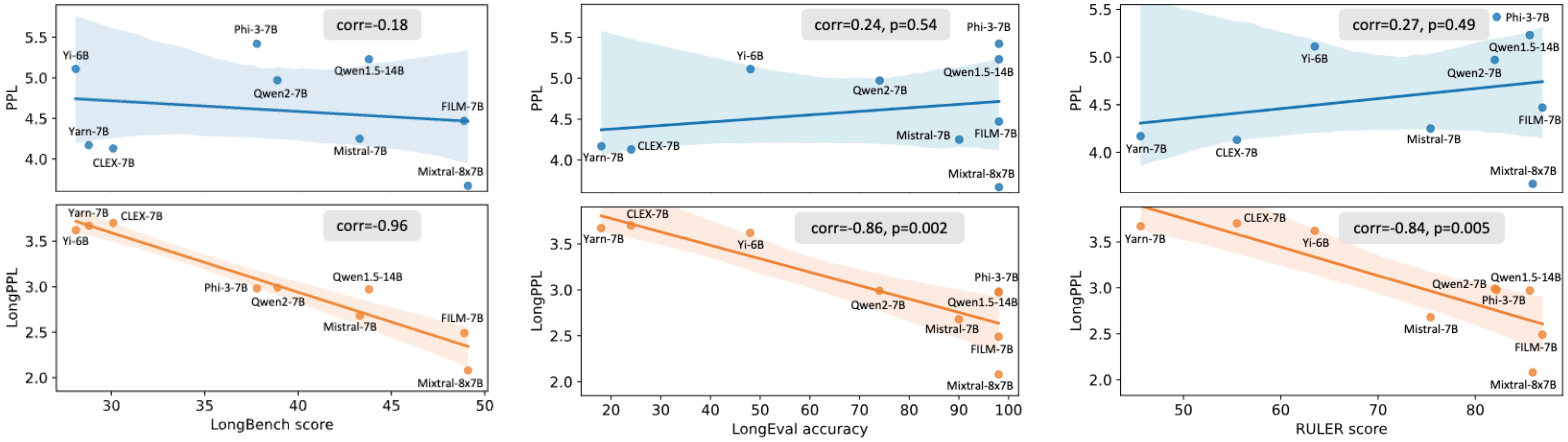
$$\mathrm{PPL}_\theta(\boldsymbol{x}) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} \log P_\theta(x_i|\boldsymbol{x}_{<i})\right)$$

$$\mathrm{LongPPL}(\boldsymbol{x};\theta,\theta_0) = \exp\left(\sum_{i=1}^{n} -\hat{I}(x_i;\theta_0)\log P_\theta(x_i|\boldsymbol{x}_{<i})\right),$$

$$\text{where } I(x_i;\theta_0) = \begin{cases} 1, & \text{if } \mathrm{LSD}_{\theta_0}(x_i) > \alpha \text{ and } \mathrm{LCL}_{\theta_0}(x) > \beta; \\ 0, & \text{otherwise.} \end{cases}$$

From PPL to LongPPL

- LongPPL (on natural data) correlates highly with long-context benchmarks



- LongPPL is insensitive to the evaluator model (Llama-3.1-8B model suffices)
- Compared to benchmark eval like RULER, LongPPL gives **real-world, efficient & adaptive** estimate for long-context performance **on the fly**

Now easy to use with ``pip install longppl`` (see https://github.com/PKU-ML/LongPPL)

From PPL to LongPPL

- Others are also reporting LongPPL — **better reveals the gains of your method**
- You may think of it as the perplexity on **"hard tokens"**

Table 2: **Perplexity on PG19 Long QA [He et al., 2025]. Our simple Δ correction results in a significant drop in both PPL and Long PPL.**

Method	Long PPL ↓	PPL ↓
Flash Attention 2	5.11 (-)	3.33 (-)
Streaming LLM	7.02 (+1.91)	3.54 (+0.21)
Streaming LLM + Δ	**5.96 (+0.85)**	**3.41 (+0.08)**
HiP Attention	6.29 (+1.18)	3.48 (+0.15)
HiP Attention + Δ	**5.45 (+0.34)**	**3.37 (+0.04)**

From Delta Attention: Fast and Accurate Sparse Attention Inference by Delta Correction. 2025

Training: from CE to LongCE

- Long-context Cross Entropy (LongCE) emphasizes key tokens **softly (no reference)**

$$\mathrm{CE}(x;\theta) = -\frac{1}{n}\sum_{i=1}^{n} \log P_\theta(x_i|\boldsymbol{x}_{<i}).$$

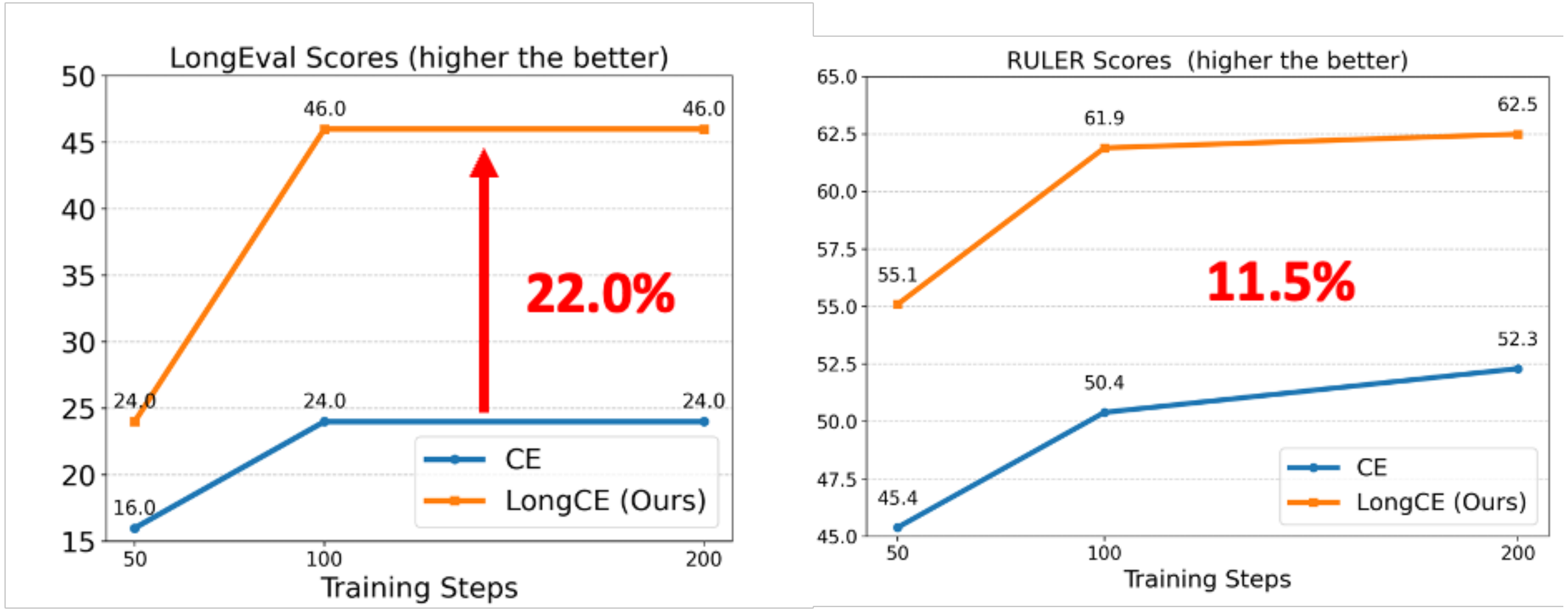
$$\mathrm{LongCE}(x;\theta) = -\frac{1}{n}\sum_{i=1}^{n} I_{\mathrm{soft}}(x_i;\theta) \log P_\theta(x_i|\boldsymbol{x}_{<i}).$$

$$I_{\mathrm{soft}}(x_i;\theta) = \min\left(\exp\left(\mathrm{LSD}_\theta(x_i)\right),\ \gamma\right) = \min\left(\frac{P_\theta(x_i|\boldsymbol{l}_i)}{P_\theta(x_i|\boldsymbol{s}_i)},\ \gamma\right)$$

- **LongCE as an Expectation-Maximization Process:**
  - E step: contrastive estimate of token importance (latent var, unknown)
  - M step: training models to maximize importance-weighted prediction
  - In this way, LongCE bootstraps its own long-context prediction & estimate
- **LongCE resembles RL training with fine-grained self-rewards**
  - No need for external rewards; more efficient than RL w/o online sampling

Training: from CE to LongCE

- LongCE improves benchmark scores up to **22%** over vanilla CE



<u>Effective across different LLMs, such as Mistral-7B, LLama2-7B, Llama2-13B!</u>

Training: from CE to LongCE

- Start to be adopted in recent works
- Found to be very effective for **efficient LLMs** (even better than our experiments)

- Our guess: Efficient LLMs (due to limited capability) require stronger training signals to focus on key information

From RWKV-X: A Linear Complexity Hybrid Language Model

shows a steep drop in performance—falling to 67.0 on S-NIAH-2 and 62.6 on S-NIAH-3. In contrast, the full model with LongCE maintains high accuracy at 99.8 and 95.6, respectively. These results demonstrate that LongCE plays a crucial role in helping the model focus on semantically important tokens over extended contexts, thereby preserving performance as sequence length increases.

Overall, LongCE significantly enhances the long-context generalization ability of RWKV-X, especially in tasks where key information is sparsely distributed across the input.

Table 4: Ablation Study on LongCE Loss using the S-NIAH Benchmark (Higher is Better).

Model	Task	1K	2K	4K	8K
RWKV-X-3.6B	S-NIAH-1	100.0	100.0	100.0	100.0
w/o LongCE	S-NIAH-1	100.0	100.0	100.0	100.0
RWKV-X-3.6B	S-NIAH-2	100.0	100.0	100.0	**99.8**
w/o LongCE	S-NIAH-2	100.0	100.0	98.4	67.0
RWKV-X-3.6B	S-NIAH-3	100.0	100.0	99.8	**95.6**
w/o LongCE	S-NIAH-3	100.0	100.0	98.4	62.6

Beyond Long Context

- Next-token prediction is biased because tokens are not generated equally
  - this bias is more evident on challenging tasks, where we care about generating certain steps/tokens correctly
- Besides RL/CoT, **reweighted/focused next-token prediction** as a general way out
  - Improve signal-noise ratio efficiently on challenging tasks

- Contrastive estimate is a general methodology for identifying task-specific tokens

$$r(x_i) = \frac{P_\theta(x_i|l_i)}{P_\theta(x_i|s_i)}$$

Target model; ideal performance

Base model; original performance

For example, in knowledge distillation, the token relevance is calculated by contrasting teacher vs student

Lizhe Fang*, YW*. et al. What is Wrong with Perplexity for Long-context Language Modeling? ICLR 2024.

This Talk

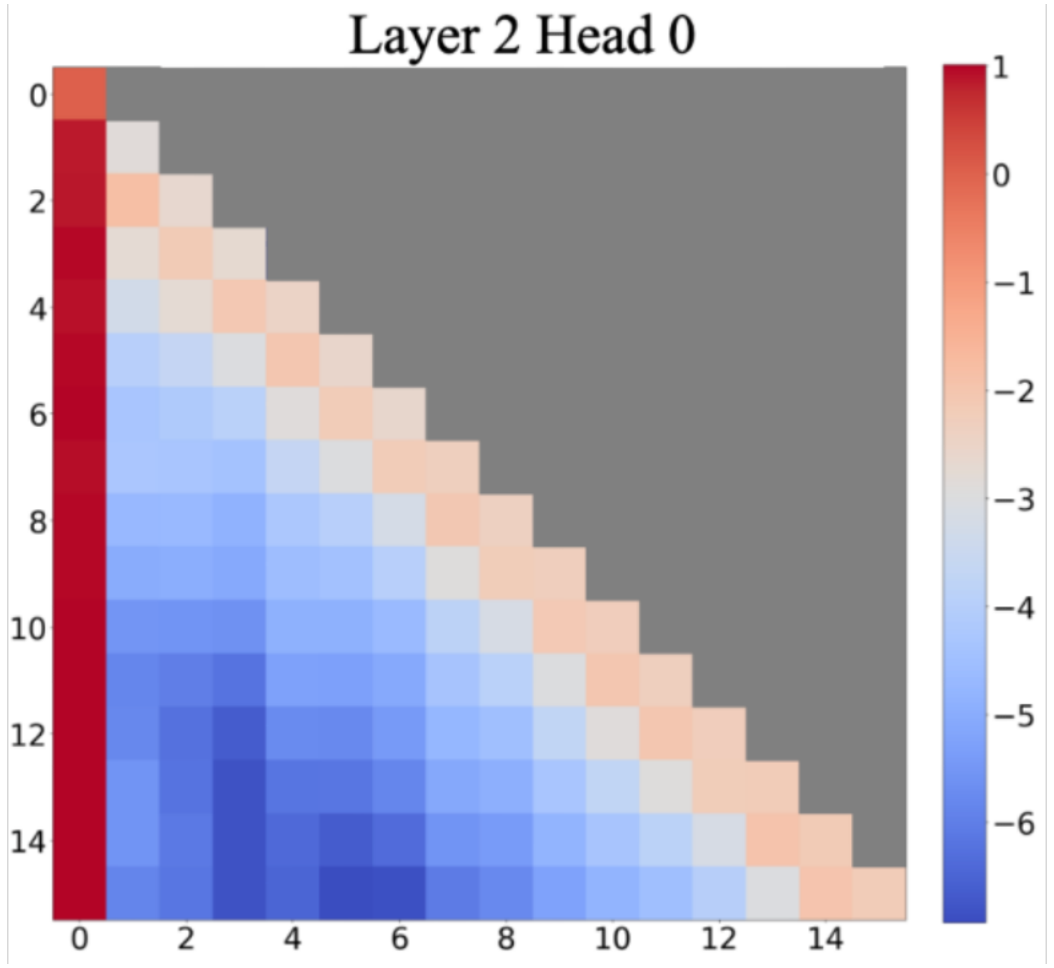
- What is Wrong with Next-token Prediction (LongPPL, ICLR'25)

- What is Wrong with Transformers (Emergence of Postion Bias, ICML'25)

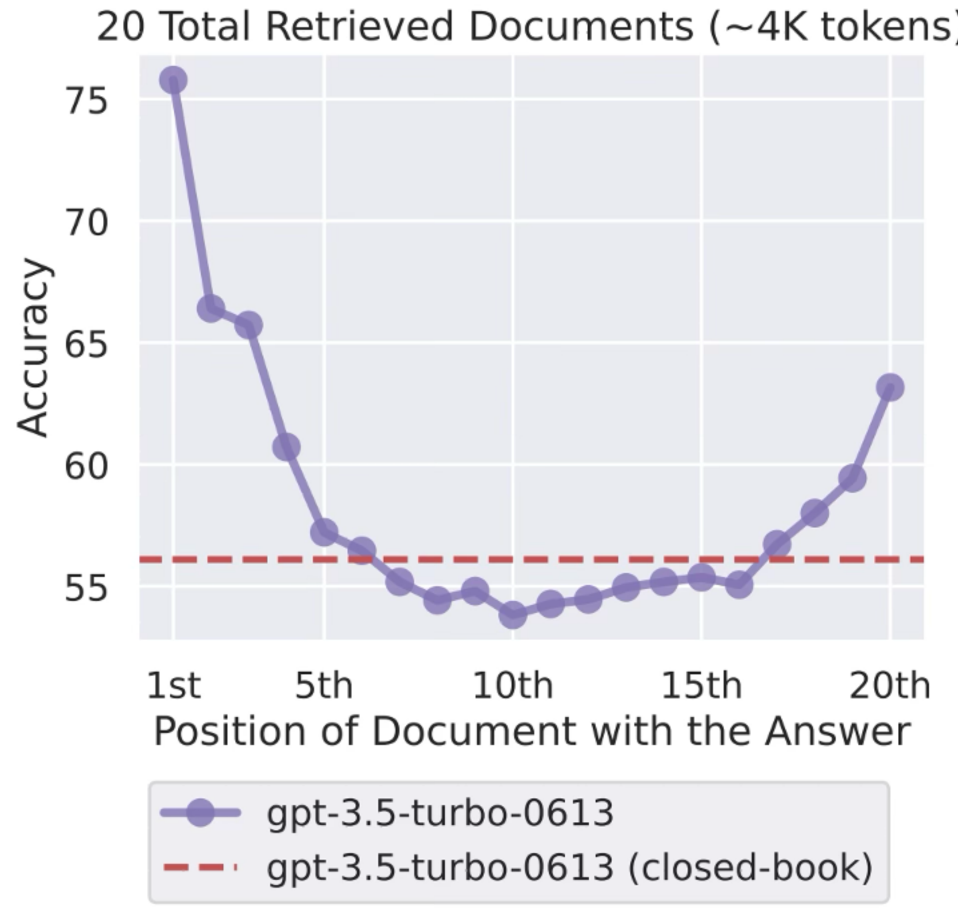
LLMs are over-sensitive to context, esp if it's long

- Sensitive to prompts
- Sensitive to ordering of in-context examples
- Sensitive to needle positions in the haystack

"attention sink"

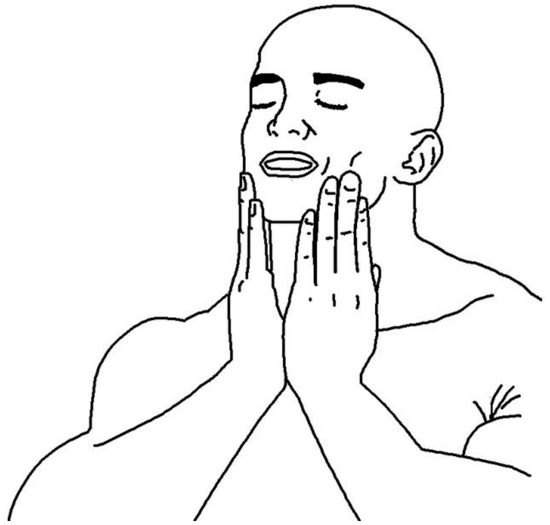


"lost in the middle"



Challenge: We don't really understand how it arises

- **Mysteries:**
  - What is the essential cause? Is it a good/bad/natural thing?
  - Is it limited to Transformers or applicable to other architectures as well?
- **Potential benefits if we can understand:**
  - A guideline on designing Transformer variants to alleviating bias
  - A guideline on designing PEs



- **Speculations/Obstacles:**
  - Mixed effects of data, model, position encodings

First step: contrlled study on position bias

- Data: independent query-key pairs (no data bias)
- PEs: NoPE (pure causal mask), Decay Mask (Alibi), RoPE

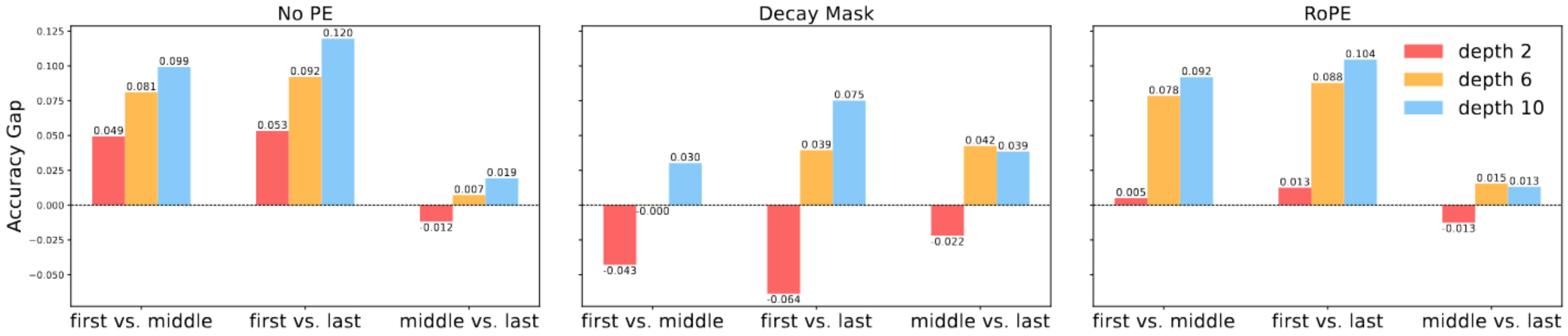
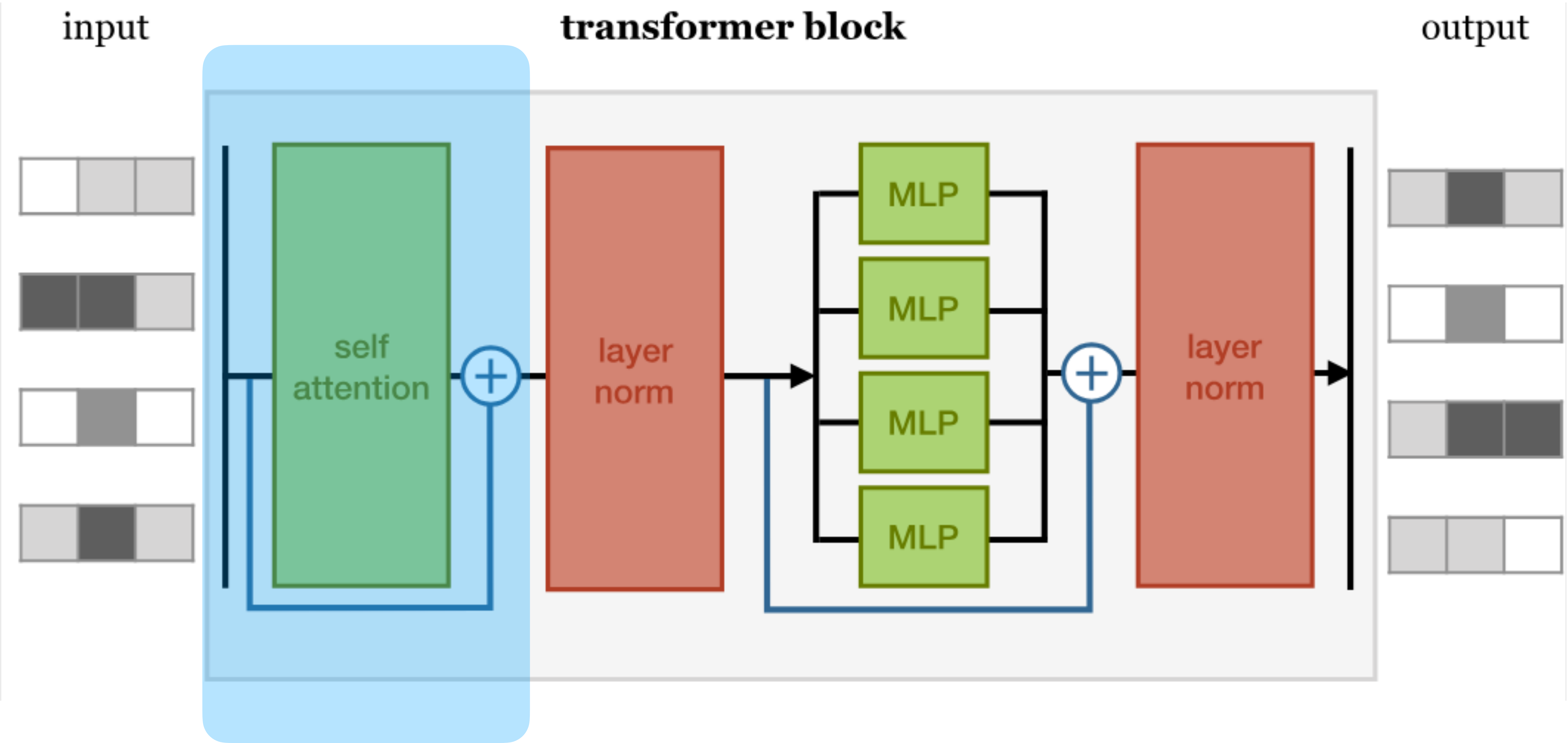


Figure 2: Position bias arising solely from the architectural design of the attention mechanism, with **no positional bias in the training data**. a vs. b denotes the gap for the case $[\boldsymbol{a}, b] - [\boldsymbol{b}, a]$, where bar

Observations:
1. Model is biased, with or without PE
2. Deeper model is more biased
3. Attention sink is more evident than recency bias

Why Position bias Emerge even w/o data bias?

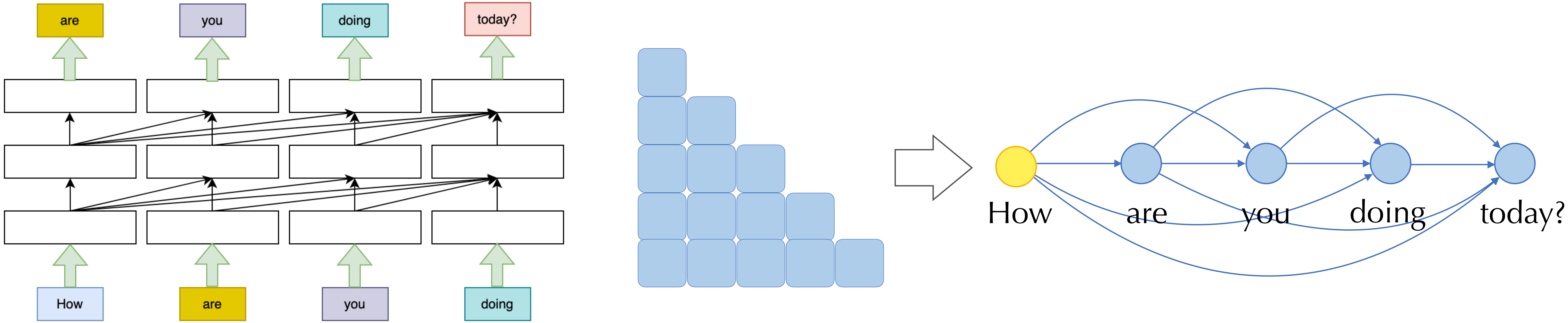


Only attention mixes tokens!

Attention is the **core cause** of position bias

Attention as a graph

- Attention induces an (adaptive) directed (computation) graph among tokens



Observations:
1. First token is a central node (it can reach all nodes)
2. This is a very imbalanced graph (node degrees decrease significantly)
3. Imbalance accumulates as models become deeper

When model goes deeper, beginning tokens **gains more "overall" weights in Transformers**

Attention as a graph

- Formaluation of multi-layer attention effect on context aggregation

$$X_{i,:}^{(t+1)} = \sum_{j=1}^{N} (A^{(t)} \cdots A^{(0)})_{ij} \cdot X_{j,:}^{(0)} W_V^{(0)} \cdots W_V^{(t)}$$

Context aggregator

$\mathbb{P}^{(t)}(z_i = j \mid X^{(0)})$

$f^{(t)}(X^{(0)}_{z_i,:})$

Context selector

The overall contribution of each context token

Attention sink, derived - causal mask / NoPE

Theorem For each token i,

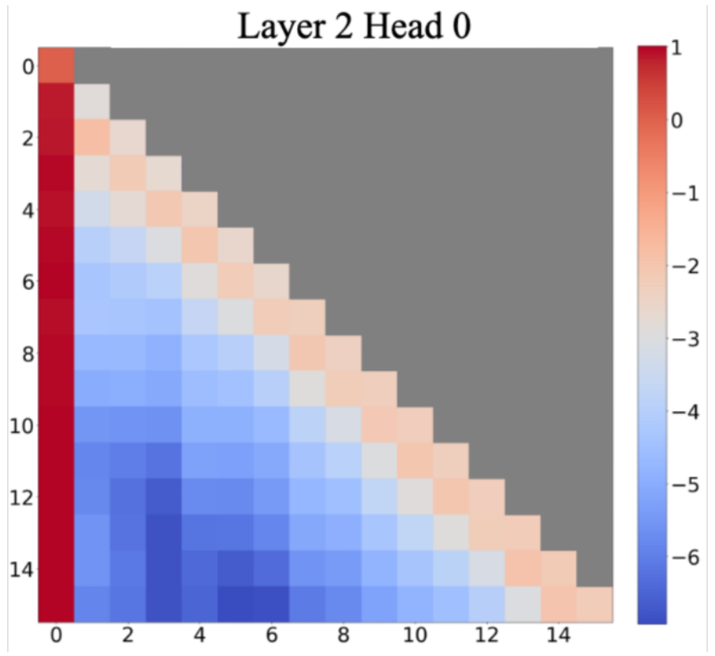
$$\lim_{t\to\infty} \mathbb{P}^{(t)}(z_i = 1 | X^{(0)}) = 1$$

The impact of tokens j > 1 exponentially decay with attention depth

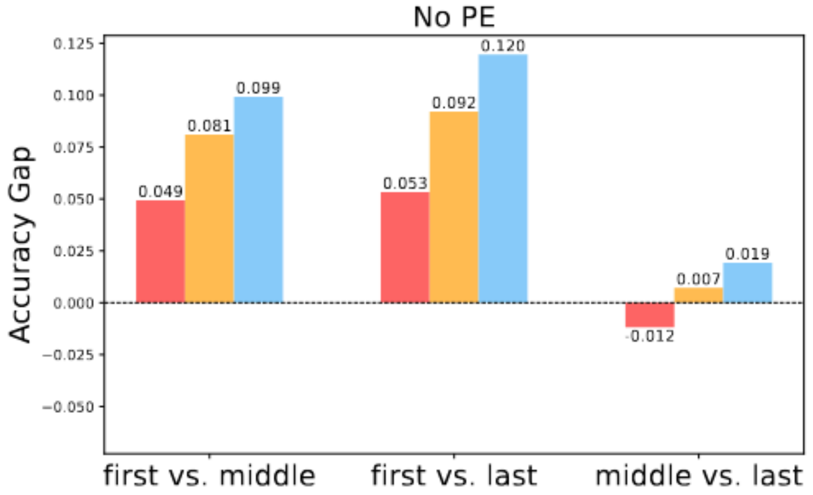
$$\mathbb{P}^{(t)}(z_i = j | X^{(0)}) \leq C(1 - (j-1)\epsilon)^t$$

Generalizable to sliding window and prefix Transformers (skipped)

Layer-wise attention sink



Context-level "attention sink"

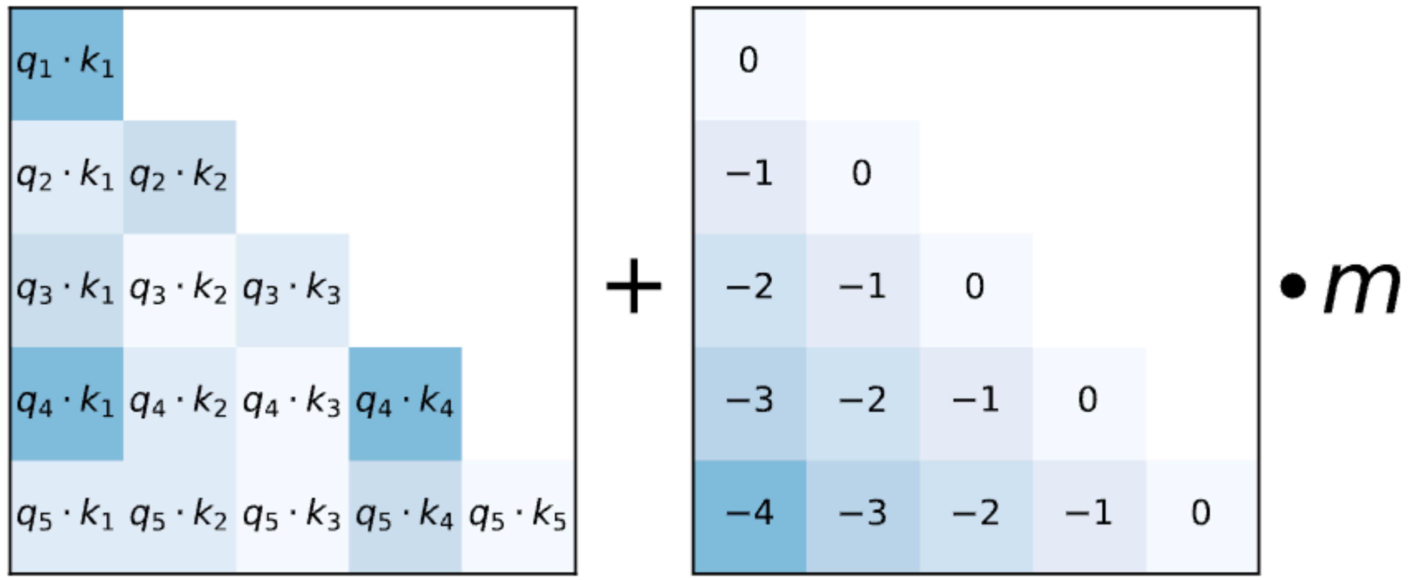


Influence of Position Encoding?

- Most attention PEs encode recency bias to attention weights (eg AliBi)

$$A^{(t)}_{\text{decay}} = \text{softmax}_{\mathcal{G}}(X^{(t)}W_Q^{(t)}(X^{(t)}W_K^{(t)})^\top + D)\,.$$

$$D_{ij} = \begin{cases} -(i-j)m & \text{if } j \leq i \\ -\infty & \text{otherwise} \end{cases}$$



essentially it **rewires** the graph

Layer-wise influence of PEs

Theorem

Decay mask

$$C_{\min} e^{-(i-j)m} \le (A^{(t)}_{\text{decay}})_{ij} \le C_{\max} e^{-(i-j)m}$$

RoPE (d=2) [need some regularity conditions on original angles and sequence

$$C_{\min} e^{-c(i-j)^2\theta_1^2} \le (A^{(t)}_{\text{RoPE}})_{ij} \le C_{\max} e^{-c'(i-j)^2\theta_1^2}$$

Observation:

Since θ_1 is typically chosen to be small, **the decay effect induced by RoPE should be significantly smaller compared to that of the decay mask**.

Lost in the middle, derived

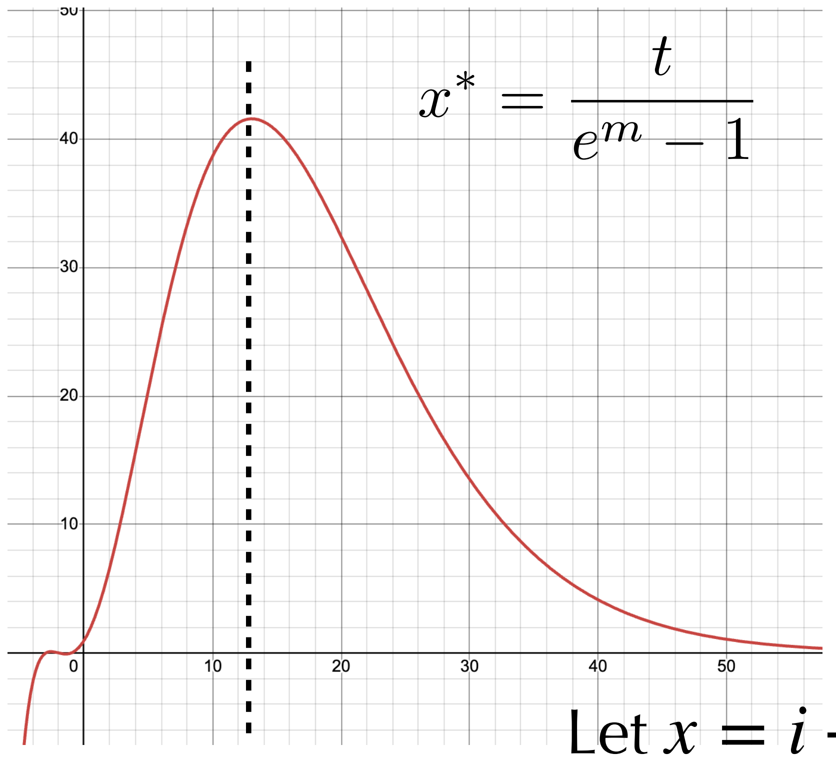
Theorem (combined effect of causal mask and PE at depth)

Decay mask

$$\mathbb{P}^{(t)}_{\text{decay}}(z_i = j | X^{(0)}) = \Theta\left(\binom{t+i-j}{i-j} e^{-(i-j)m}\right)$$

RoPE (d=2)

$$\mathbb{P}^{(t)}_{\text{RoPE}}(z_i = j | X^{(0)}) = \Theta\left(\binom{t+i-j}{i-j} e^{-c(i-j)^2\theta_1^2}\right)$$



$$x^* = \frac{t}{e^m - 1}$$

Let $x = i - j$

- Deeper models become more biased toward initial tokens.
- Increasing the decay strength m or base rotational angle amplifies the long-term decay effect and causes tokens to focus more on nearby tokens.

Summary

Empirical Observations on Position Bias	Our Results
Positional information induced by the causal mask (Barbero et al., 2024a; Kazemnejad et al., 2023; Wang et al., 2024)	Theorem 4.1, Section 5.2
Decay effects induced by relative PEs (Su et al., 2023)	Lemma 4.4-4.6, Section 5.1
Interplay between the causal mask and relative PEs (Wang et al., 2024)	Theorem 4.5-4.7, Section 5.1
Attention sinks (Gu et al., 2025; Xiao et al., 2024)	Theorem 4.1-4.3, Appendix K.2
The "lost-in-the-middle" phenomenon (Liu et al., 2024)	Section 5.2

- Position bias is essentially caused by the graph structure of attention
  - NoPE (causal mask) has its own position bias
  - PEs can make models more sensitive to data bias
- A good PE / Transformer variant should be able to derive a balanced graph

Xinyi Wu, YW, Stefanie Jegelka, and Ali Jadbabaie. On the Emergence of Position Bias in Transformers. ICML 2025.

Final thoughts

- Next-token prediction and Transformers might be good (enough) for curve fitting

- But if we want more than distr. matching (capability, robustness), we need to
  - Redistribute the token/sequence rewards for efficient training
  - Debiase Architectures

- Thinking LLMs as "Large Context Model" helps
  - A unified perspective of data molidaties, and understanding/reasoning tasks
  - It's all about contextualized prediction/representation

YW*, Yuyang Wu*, Zeming Wei, Stefanie Jegelka, and Yisen Wang. A Theoretical Understanding of Self-Correction through In-context Alignment. NeurIPS 2024.