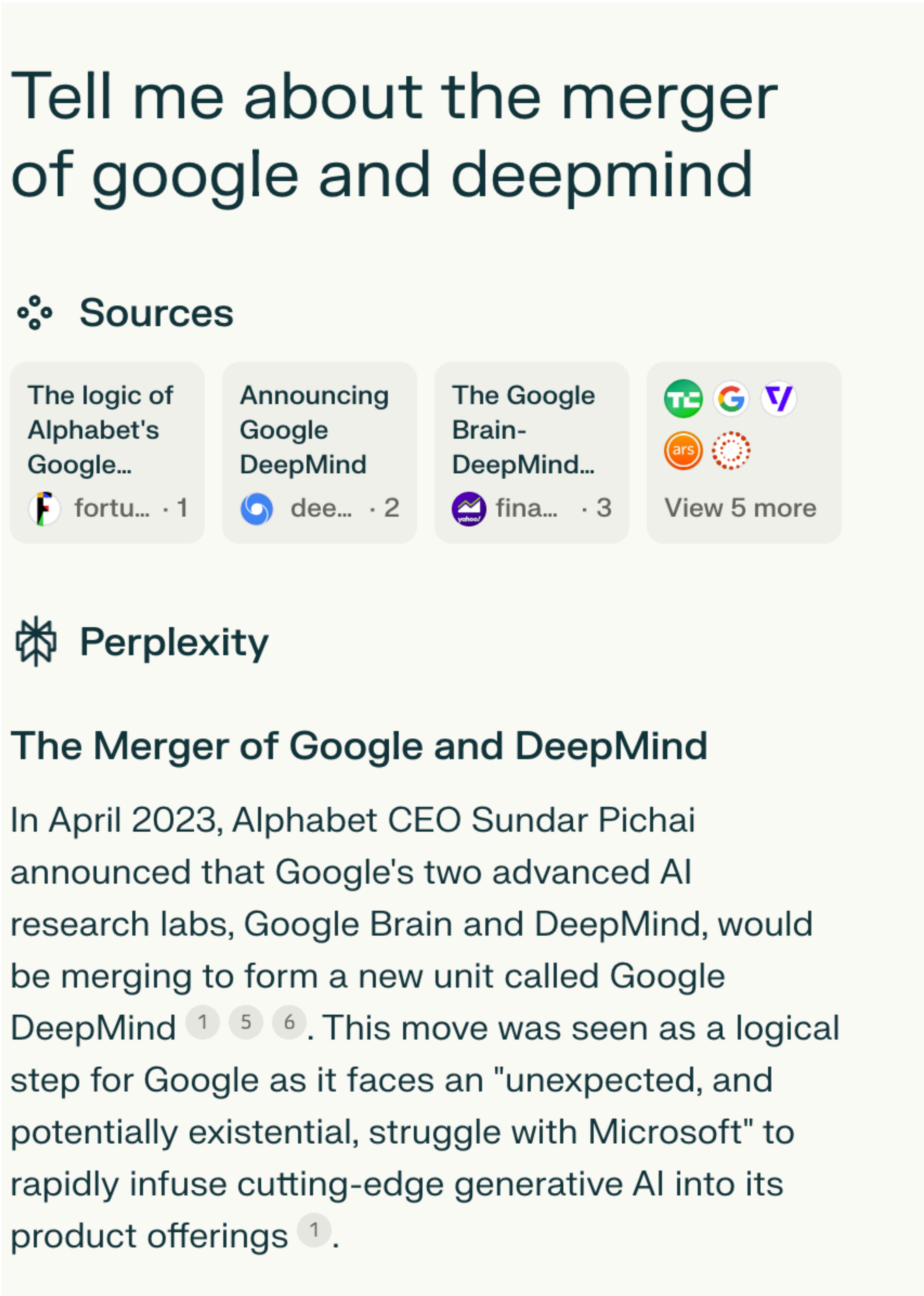
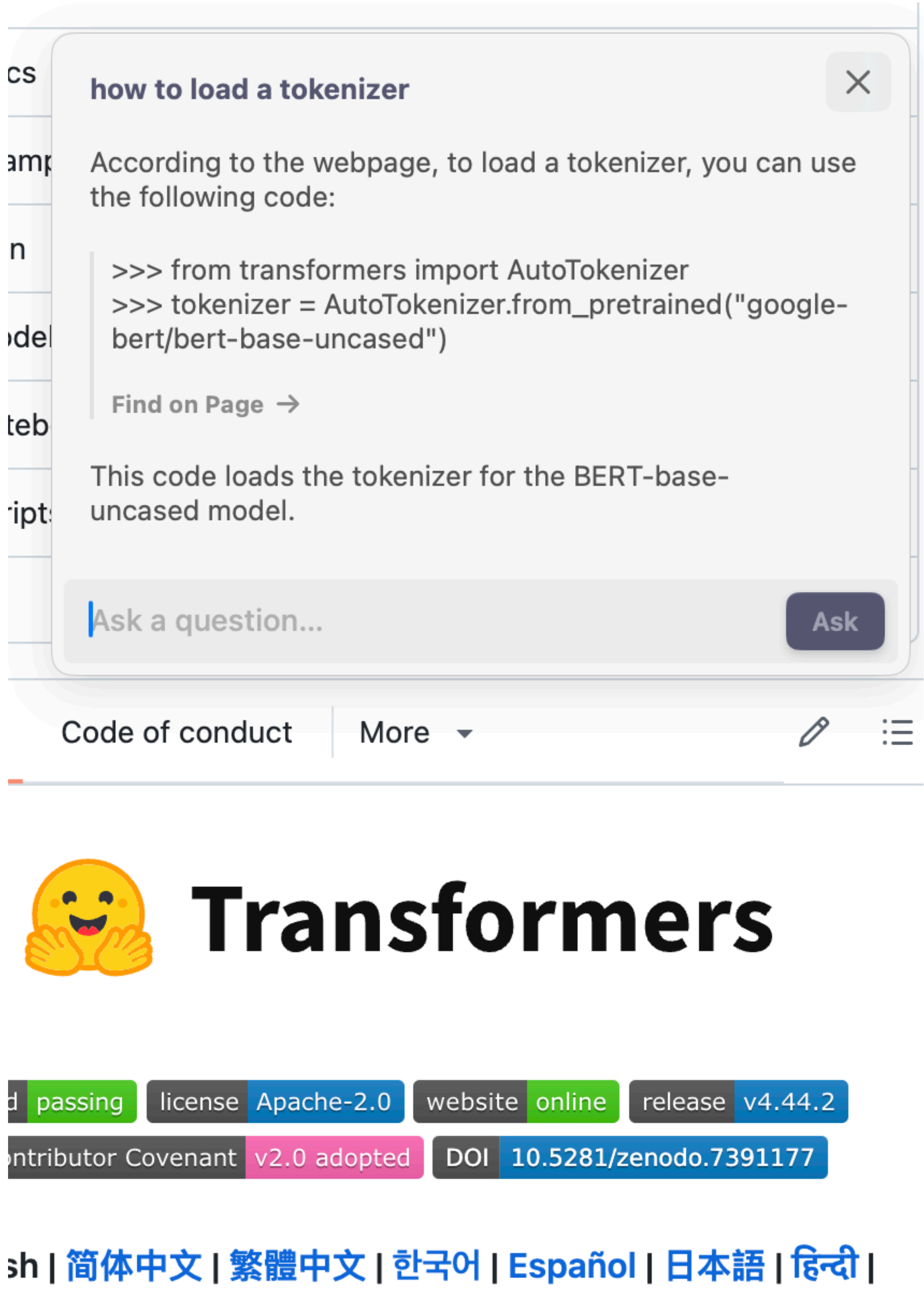
Long-Context Language Models

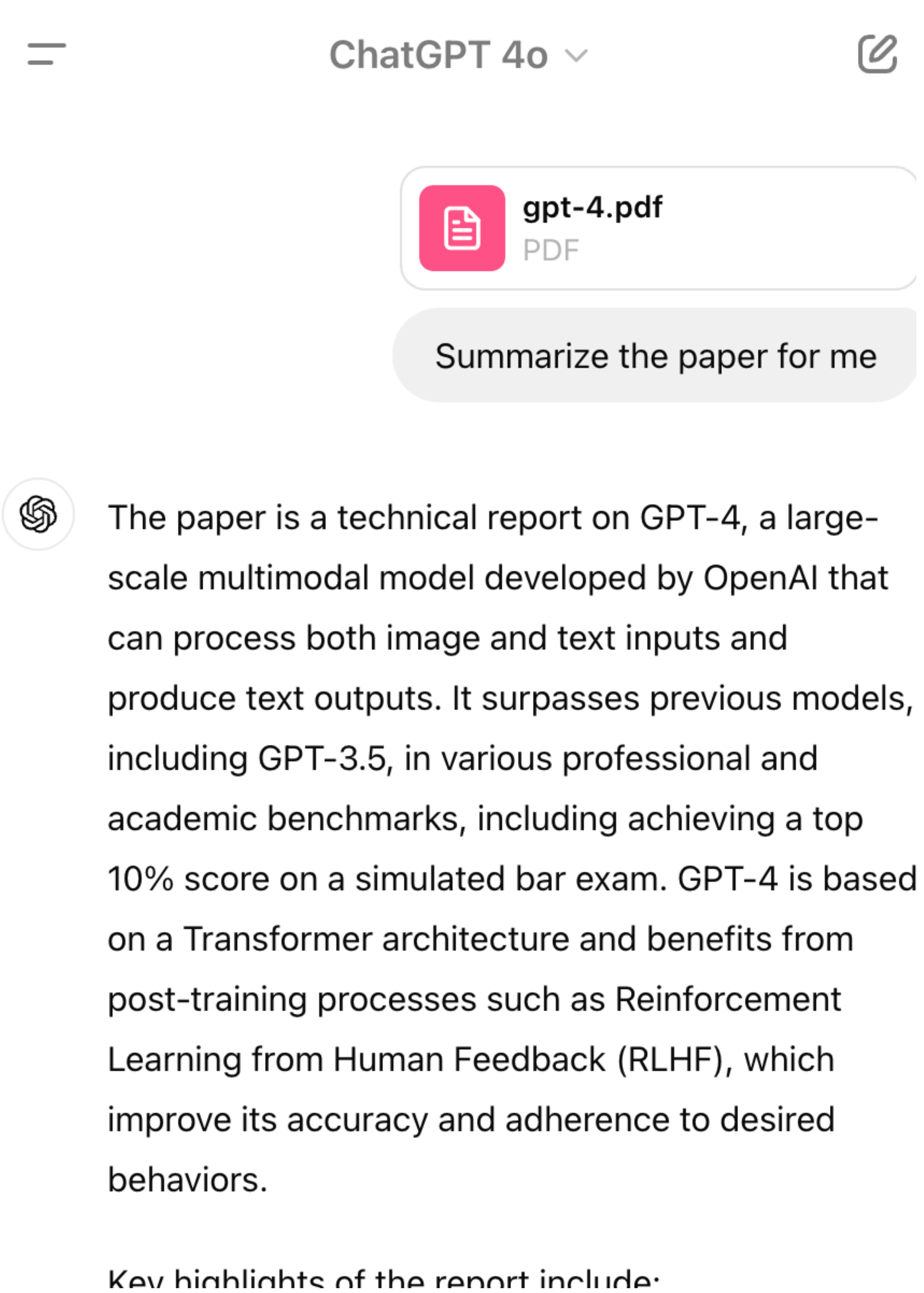
Tianyu Gao
Princeton Language and Intelligence (PLI)
Princeton University

Why do we want long-context models

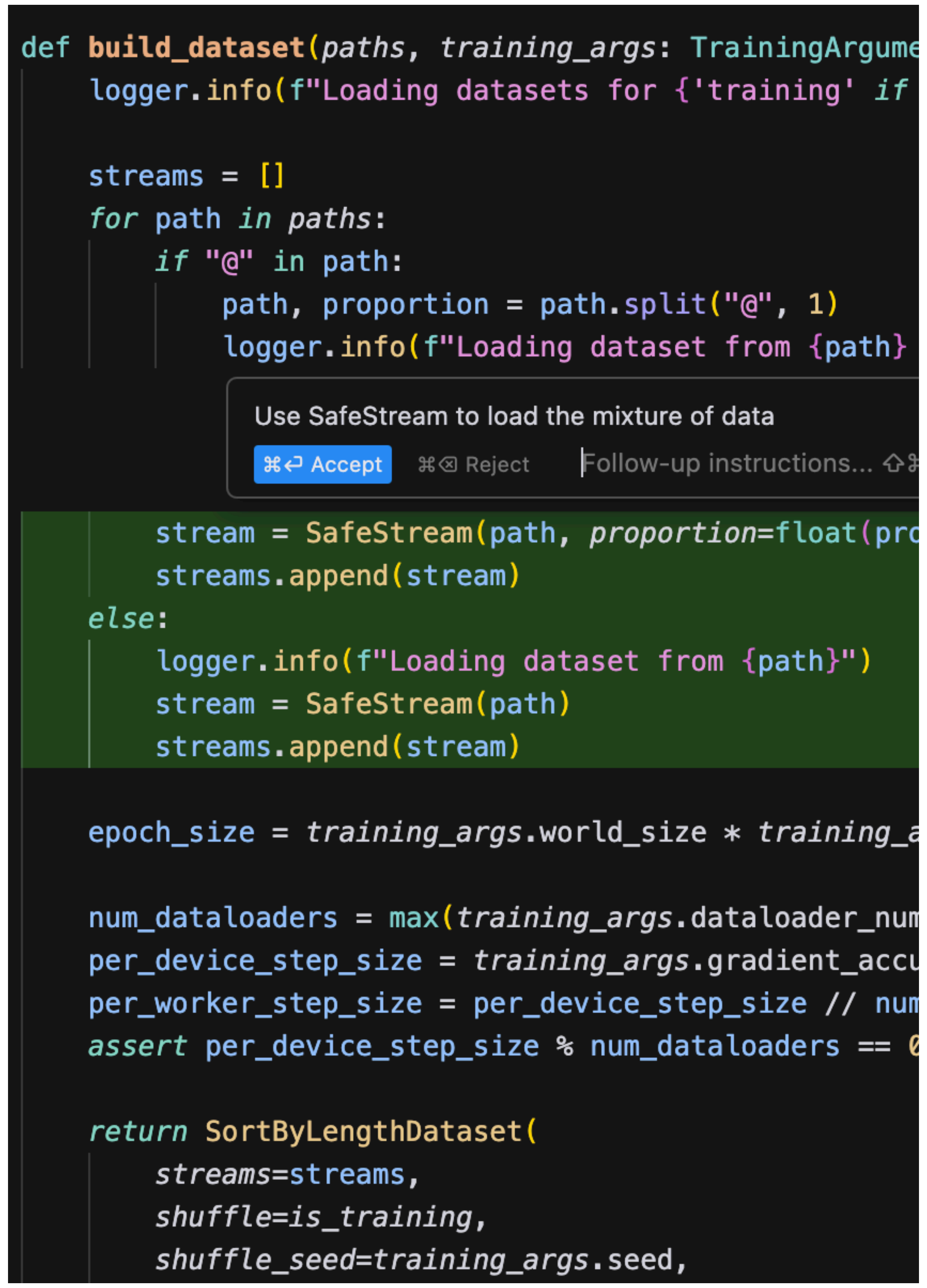


Retrieval-augmented generation (RAG)



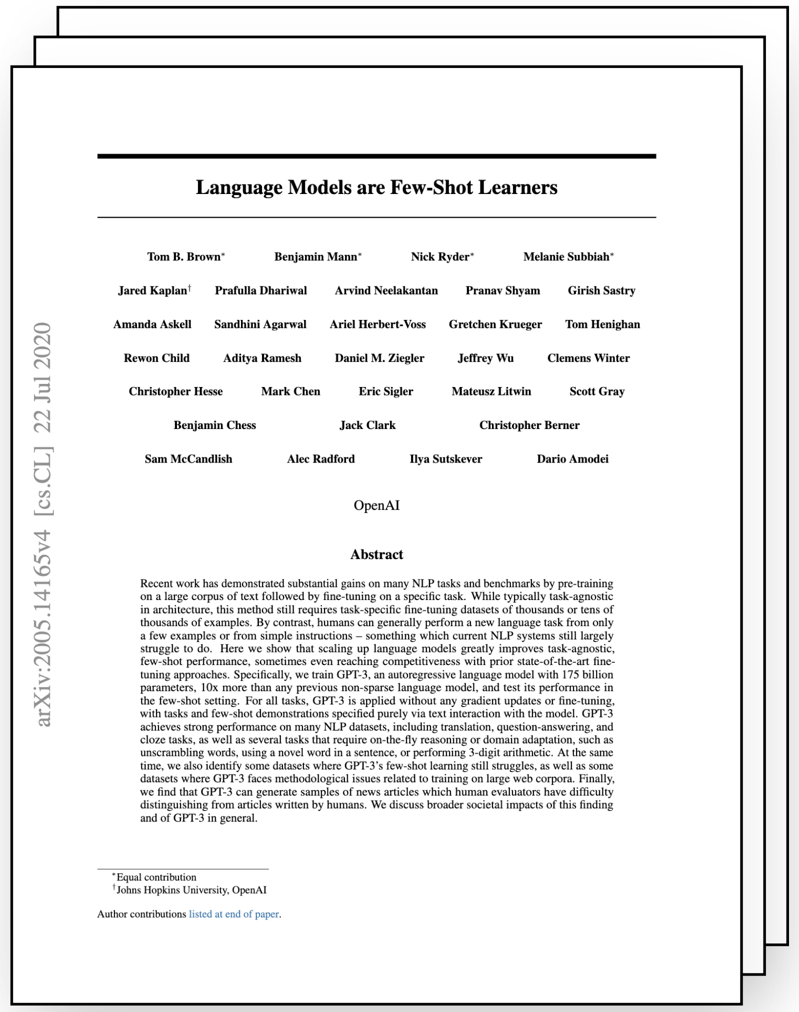


Long document QA / summarization



Repo-level code understanding / Agent

Why do we want long-context models

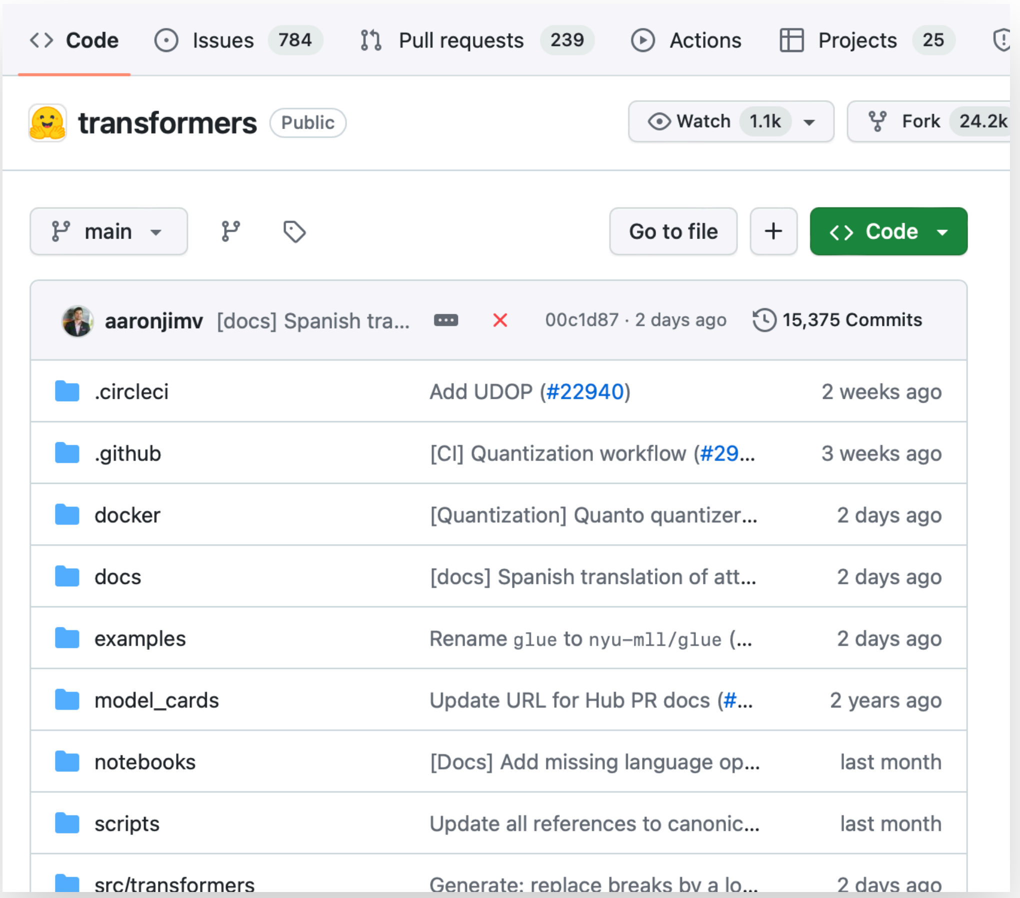


The GPT-3 paper

(~**75K** tokens)

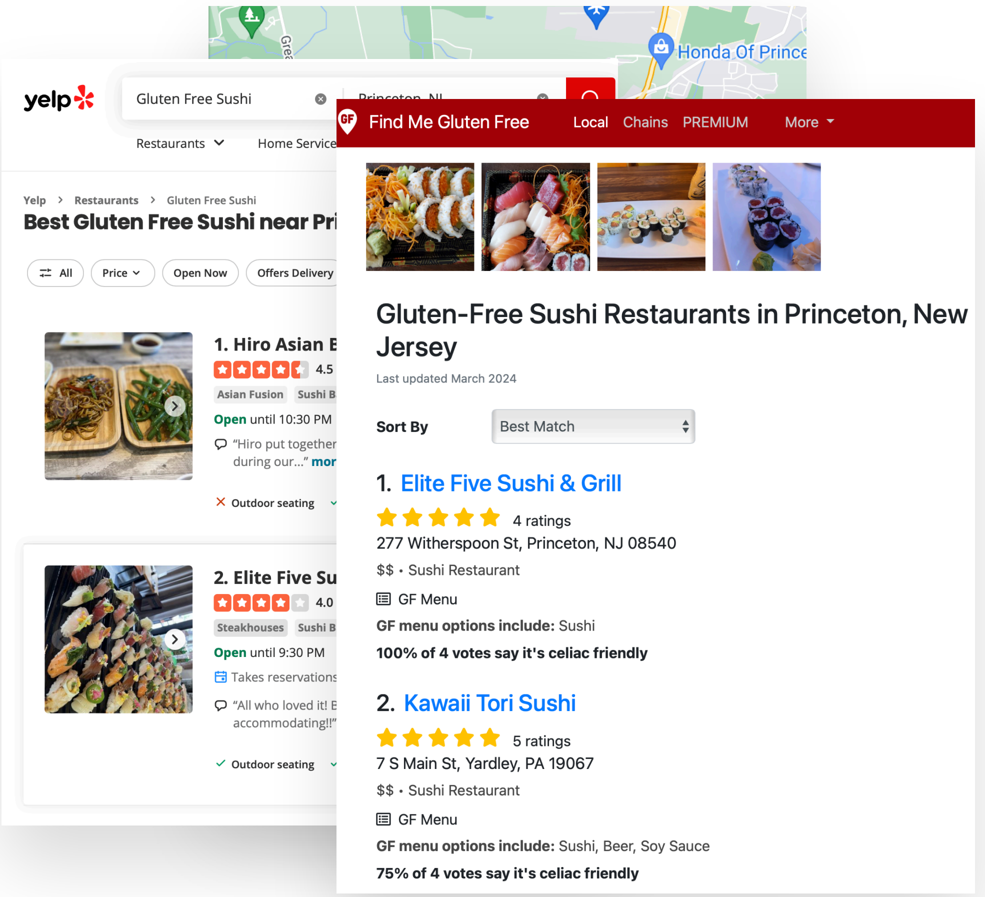
The Dune series

(~**1M** words)



The Transformers package

(~**10M** tokens)



100 web pages

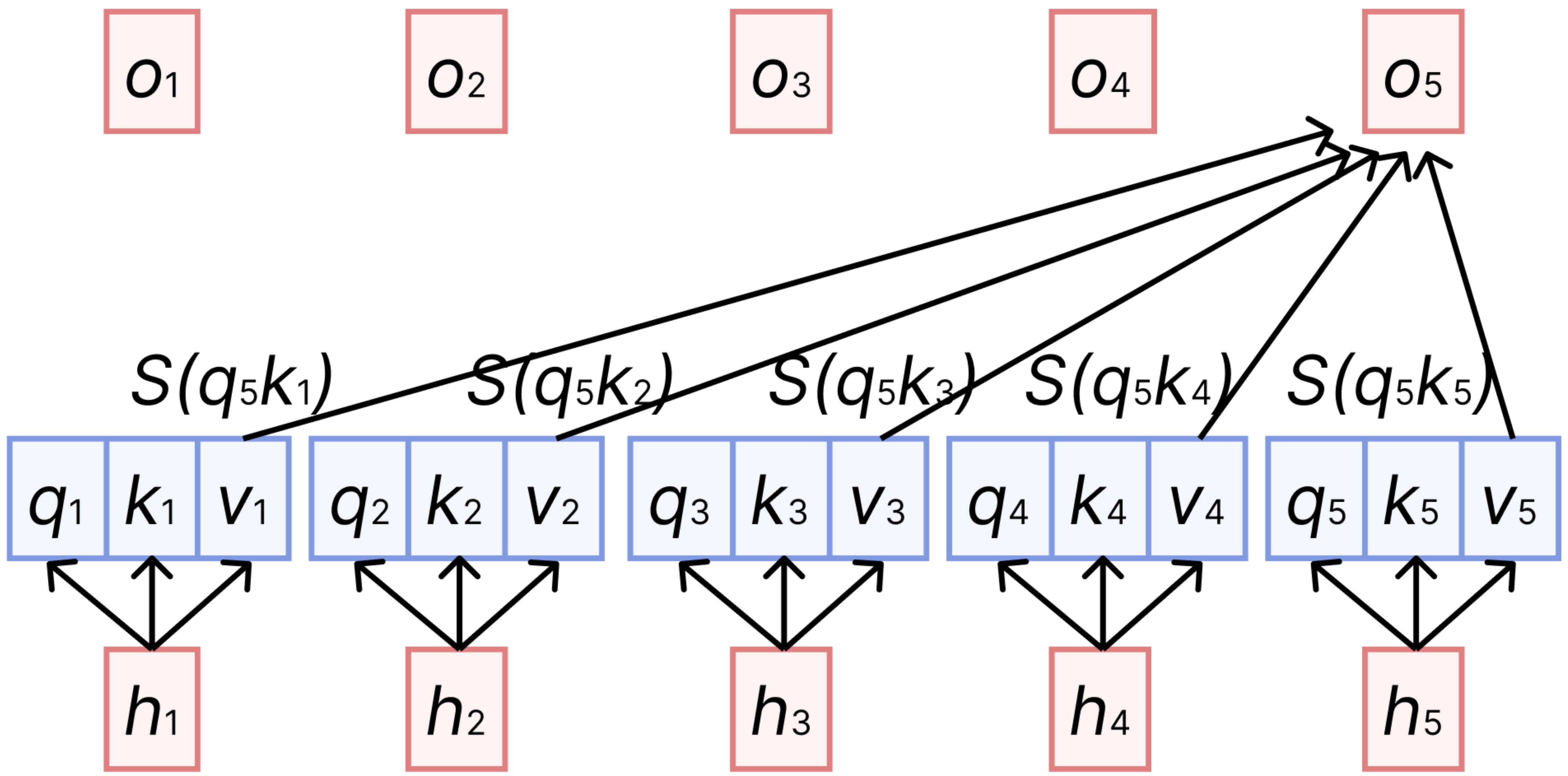
(~**100K** tokens)

Meta Llama-3 (3.1): 8K / 128K   GPT-4o: 128K   Claude 3.5: 200K   Gemini: 2M

Long-context hurdle #1: Transformers

Transformers are costly (both computation and memory)

Multi-head attention: the representation of every word is a *weighted sum* of *all* previous words



To encode a context of n words
$\mathcal{O}(n^2)$ compute complexity

To predict the next word
$\mathcal{O}(n)$ memory complexity

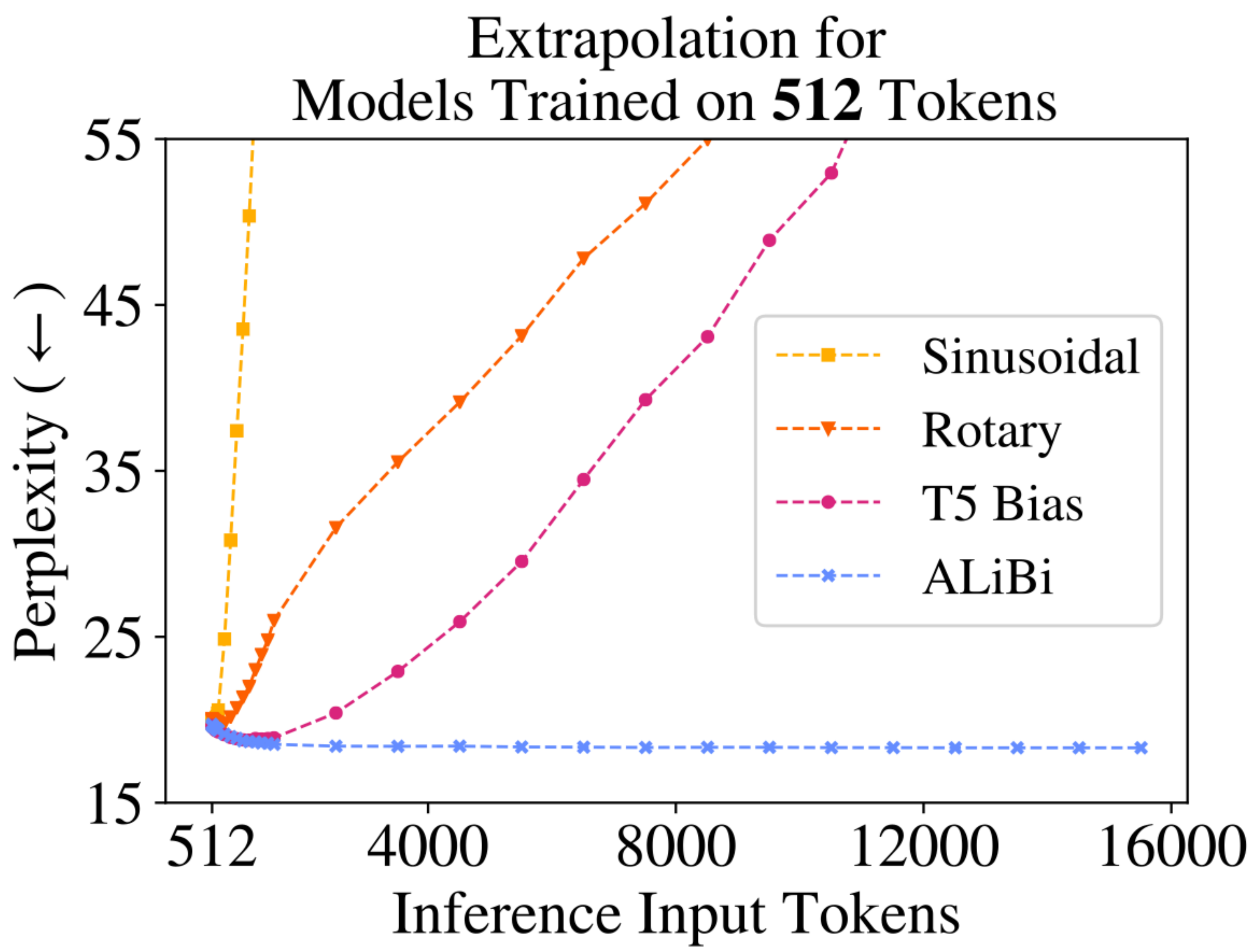
A **1M** token context would cost **164GB** memory!
(Llama-70B, FP16)

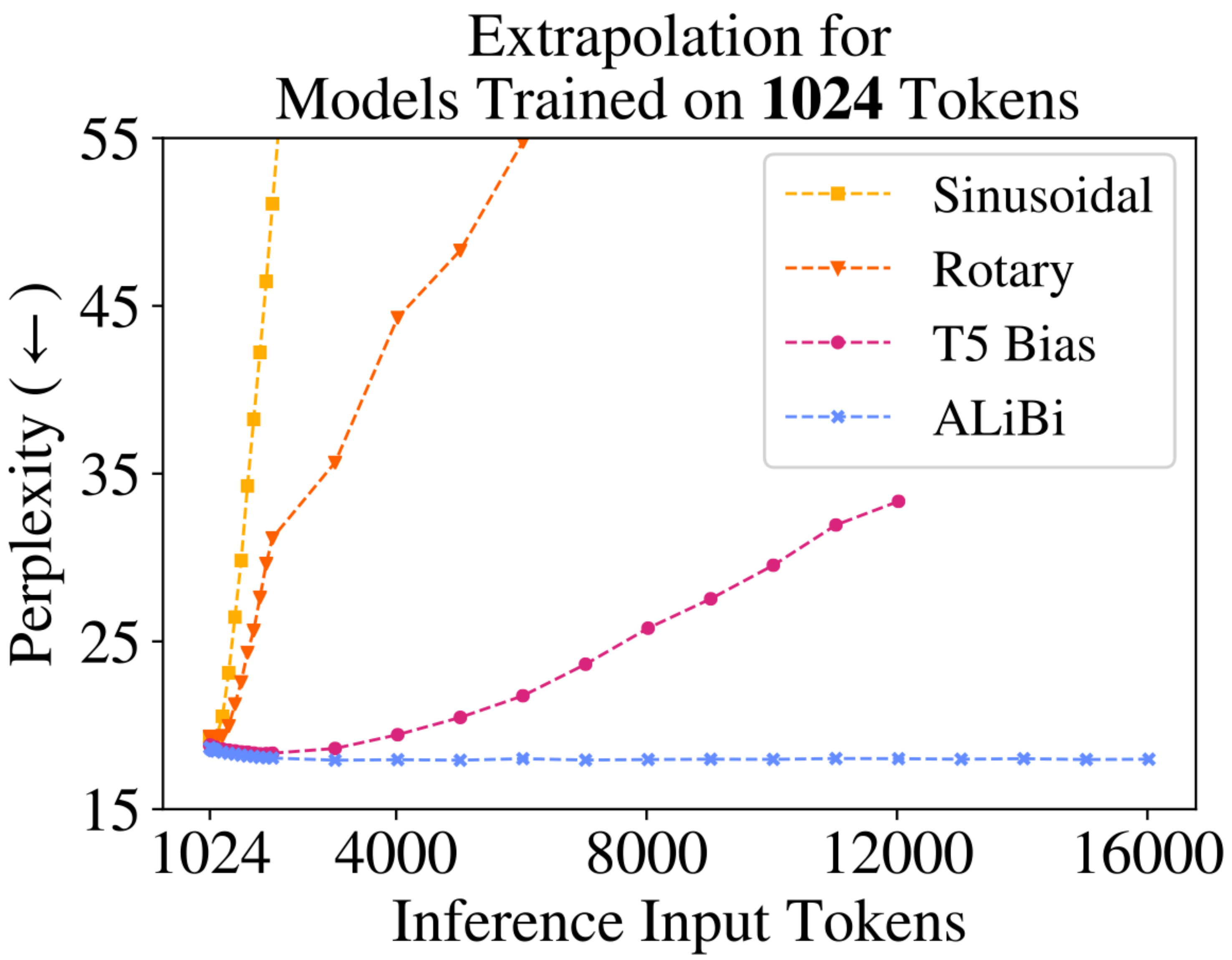
Vaswani et al., 2017. Attention is all you need.

Long-context hurdle #2: Positional encodings

Popular positional encodings are not generalizable

The most popular positional encoding method, RoPE (Su et al., 2021), cannot generalize beyond the training length.





Su et al., 2021. RoFormer: Enhanced Transformer with Rotary Position Embedding.
Press et al., 2021. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation.

Long-context hurdle #3: Evaluation

How can we create tasks that require reading >100K tokens?

The most popular evaluation tasks (MMLU, HellaSwag, GSM8K) use **<500 tokens**

Chat-based evaluations (AlpacaEval, ArenaHard, MT-Bench) all have **<1K tokens**

Developers instead rely on either **perplexity** or **simple synthetic tasks**

- … but can we trust those to reflect models' true long-context performance?

Lin et al., 2024. WildBench: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild.

Long-context hurdle #4: Data scarcity

High-quality long-context data are hard to find

Average length of domains from **RedPajama** (a open-source pre-training data collection)

- Wikipedia: 0.5K tokens
- C4 (webpages): 0.5K tokens
- Arxiv: 20K tokens
- Books: 147K tokens

Average length of **instruction-tuning/chat** data: <1K tokens

How can we train a model that can continually generalize to longer length?

Together. 2023. RedPajama, a project to create leading open-source models, starts by reproducing LLaMA training dataset of over 1.2 trillion tokens.

Today's lecture

- **Positional encoding**: the promise of train shorter and test longer?
- **Evaluation**: how much can we trust intuitive synthetic evaluations?
- **Training**: how to effectively train a long-context language model?

How do language models encode positions?

Why do we need position information?

“Mike bit the dog” vs. “the dog bit Mike”

Can LMs learn to encode position information implicitly?

In recurrent neural networks (RNNs), yes

- Models can learn an decaying effect on the past cells

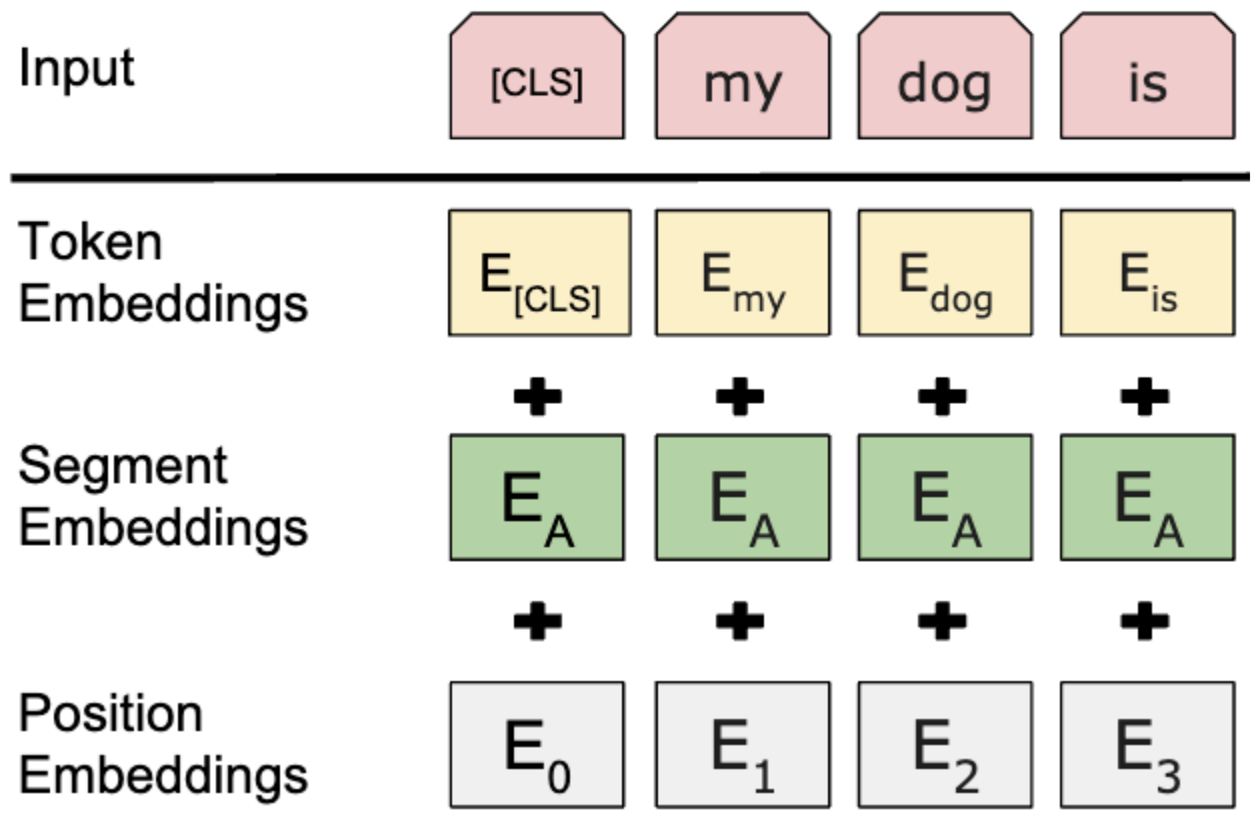
In Transformers, no

- Because every position attends to all previous positions equally (no way to tell the difference)
- … but, if you have **multiple layers**, you can still learn it (Kazemnejad et al., 2023)!

Kazemnejad et al., 2023. The Impact of Positional Encoding on Length Generalization in Transformers.

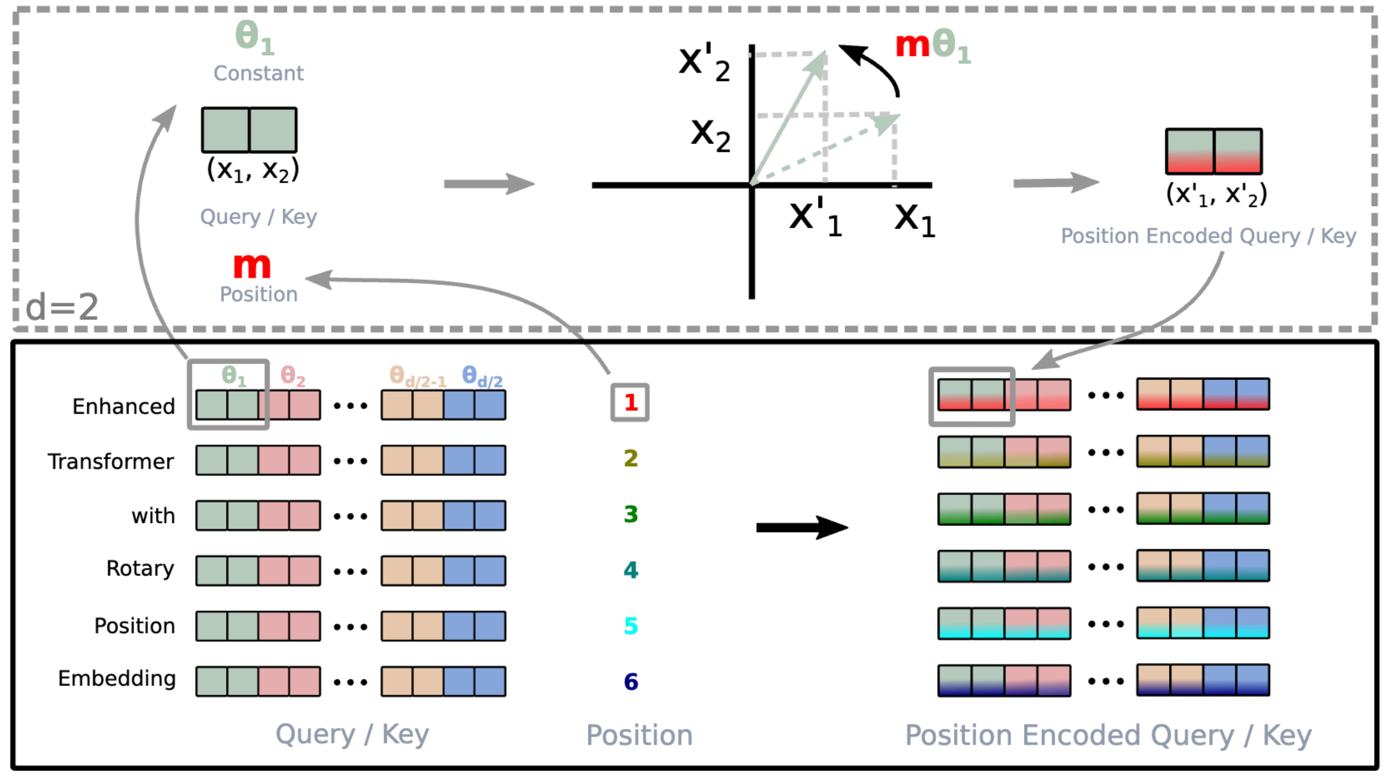
How do language models encode positions?

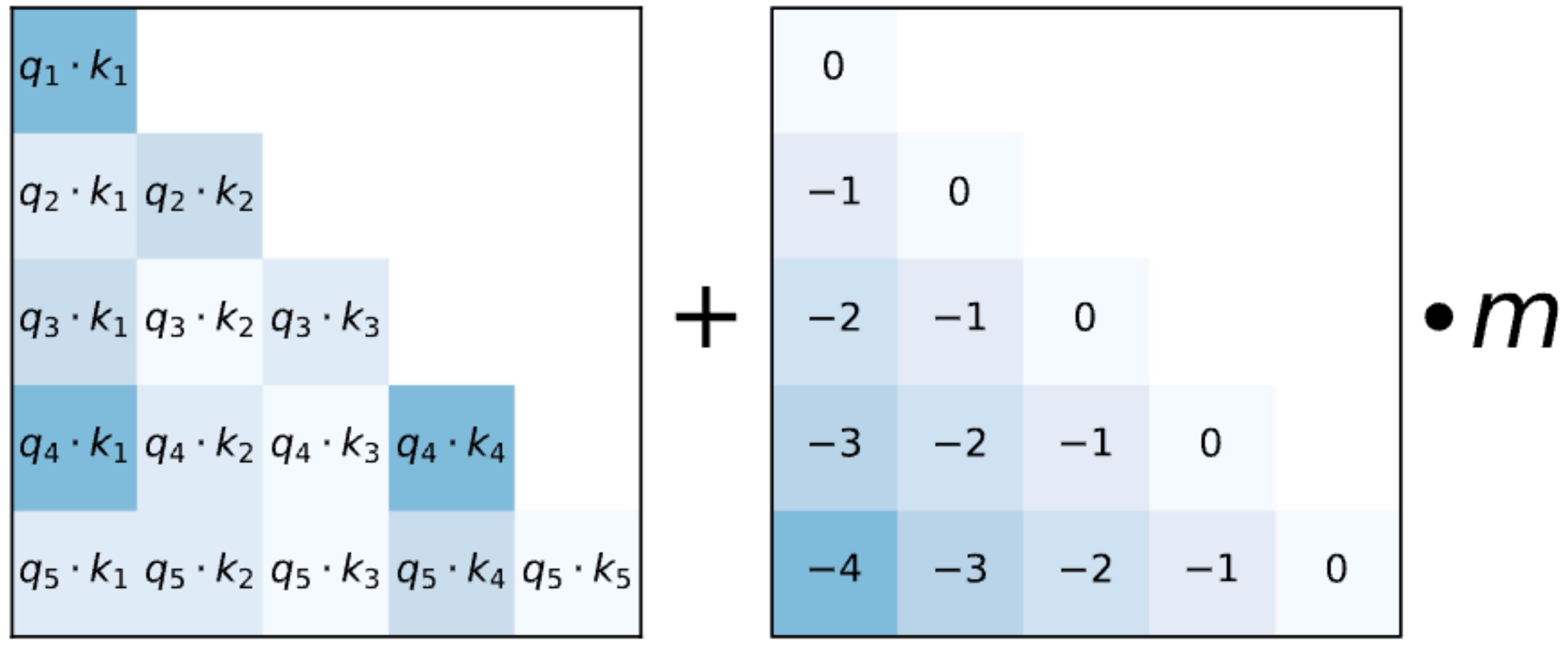
We need some explicit ways of encoding positions



Absolute position embeddings

(learnable / fixed)





Relative position embeddings

(learnable / **fixed**)

*holds the promise of **length generalization***

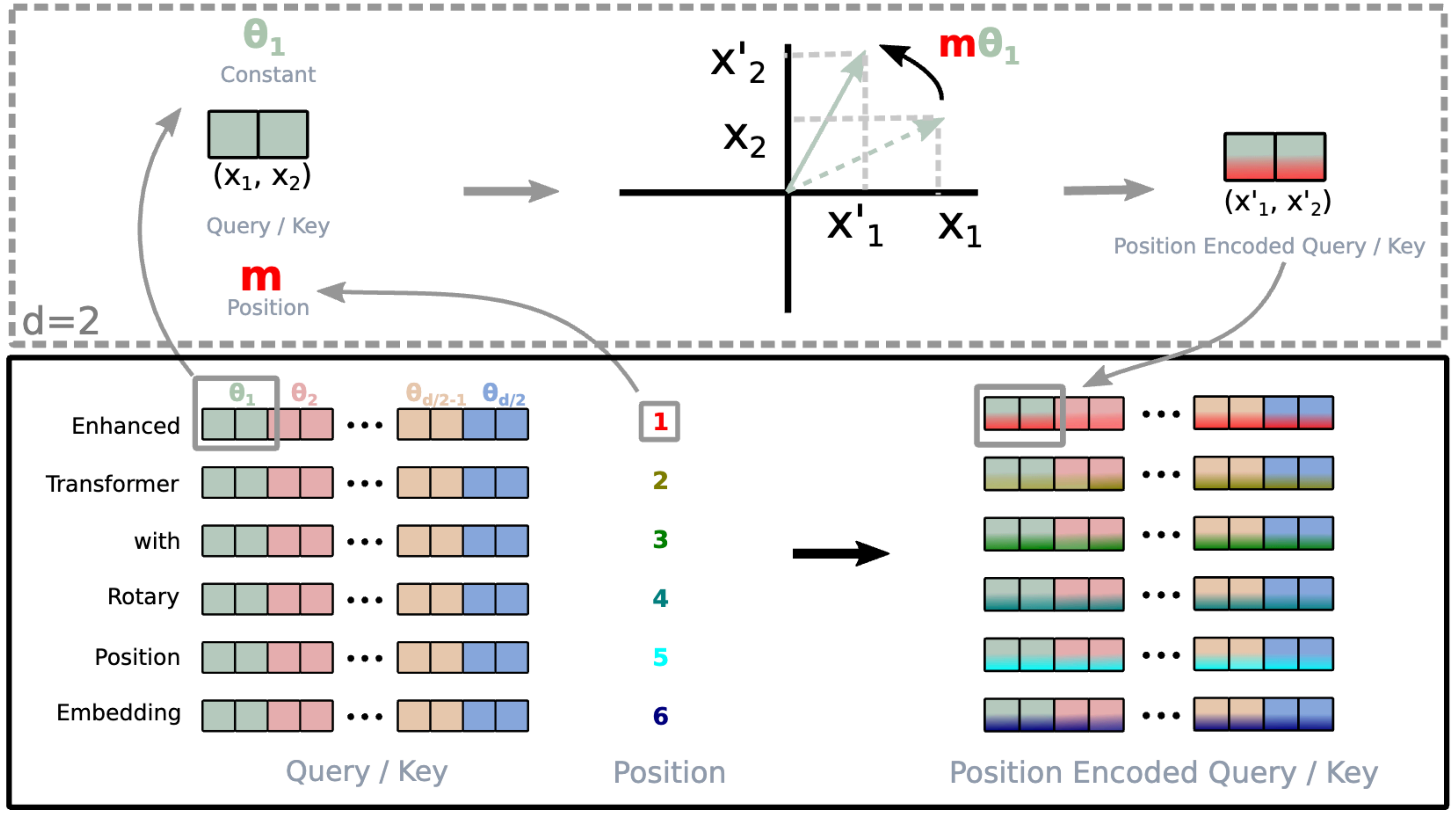
Devlin et al., 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
Su et al., 2021. RoFormer: Enhanced Transformer with Rotary Position Embedding.
Press et al., 2021. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation.

Rotary position embedding (RoPE)

Rotate the **query** / **key** vectors in attention **based on their positions**

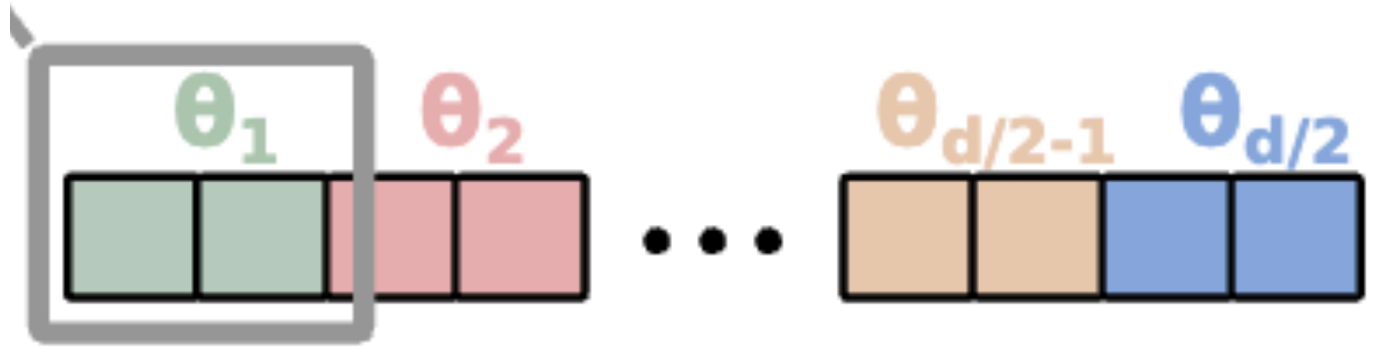
$$R\mathbf{v} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}.$$

Group the values in the vector by two and rotate different groups **by different frequencies**



Su et al., 2021. RoFormer: Enhanced Transformer with Rotary Position Embedding.

Rotary position embedding (RoPE)



$$\theta_i = 10000^{-2(i-1)/d}, i \in [1, 2, ..., d/2]$$

Frequency base

$$\mathbf{q}_{m\theta_i}^T \mathbf{k}_{n\theta_i} = \mathbf{q}^T \mathbf{k} \cos\left((m-n)\theta_i\right)$$

Encode relative position information!

$i = 1, \theta_i = 1$

More local information

$\vdots$

$i = \frac{d}{2}, \theta_i \approx 1/\text{base}$

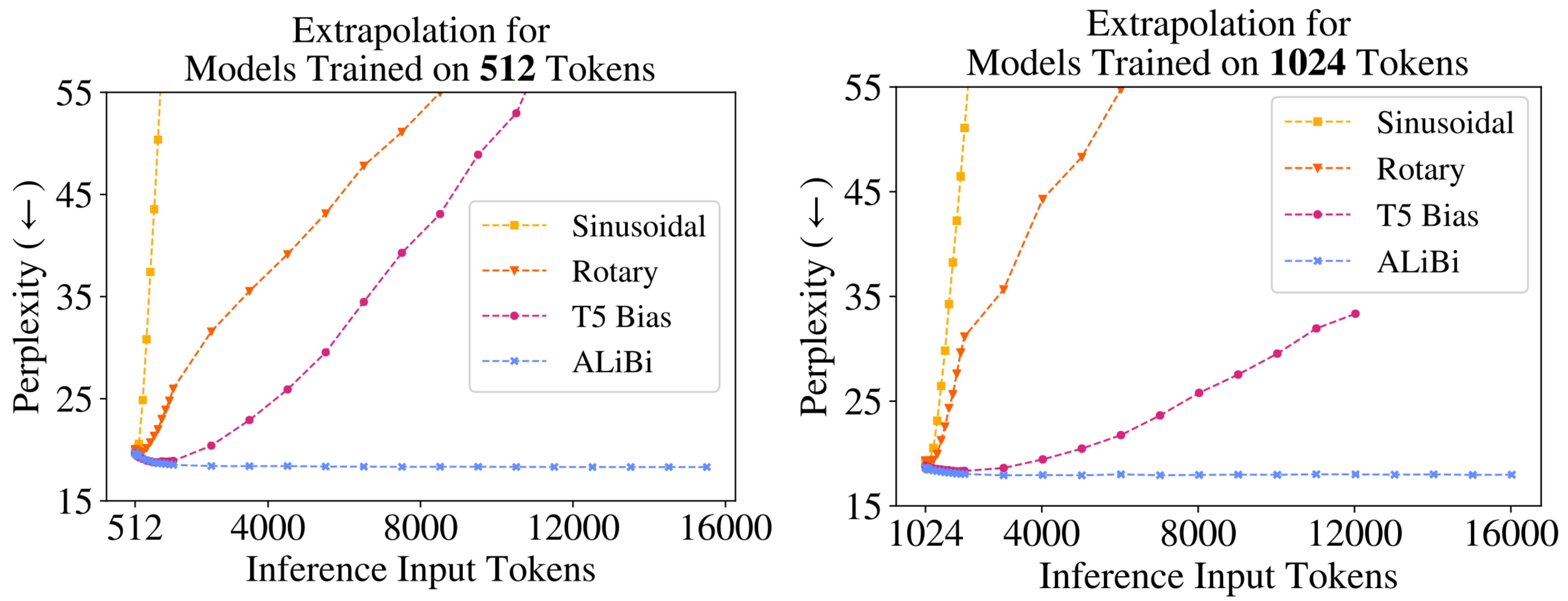
Long-term decay (further -> smaller cosine)

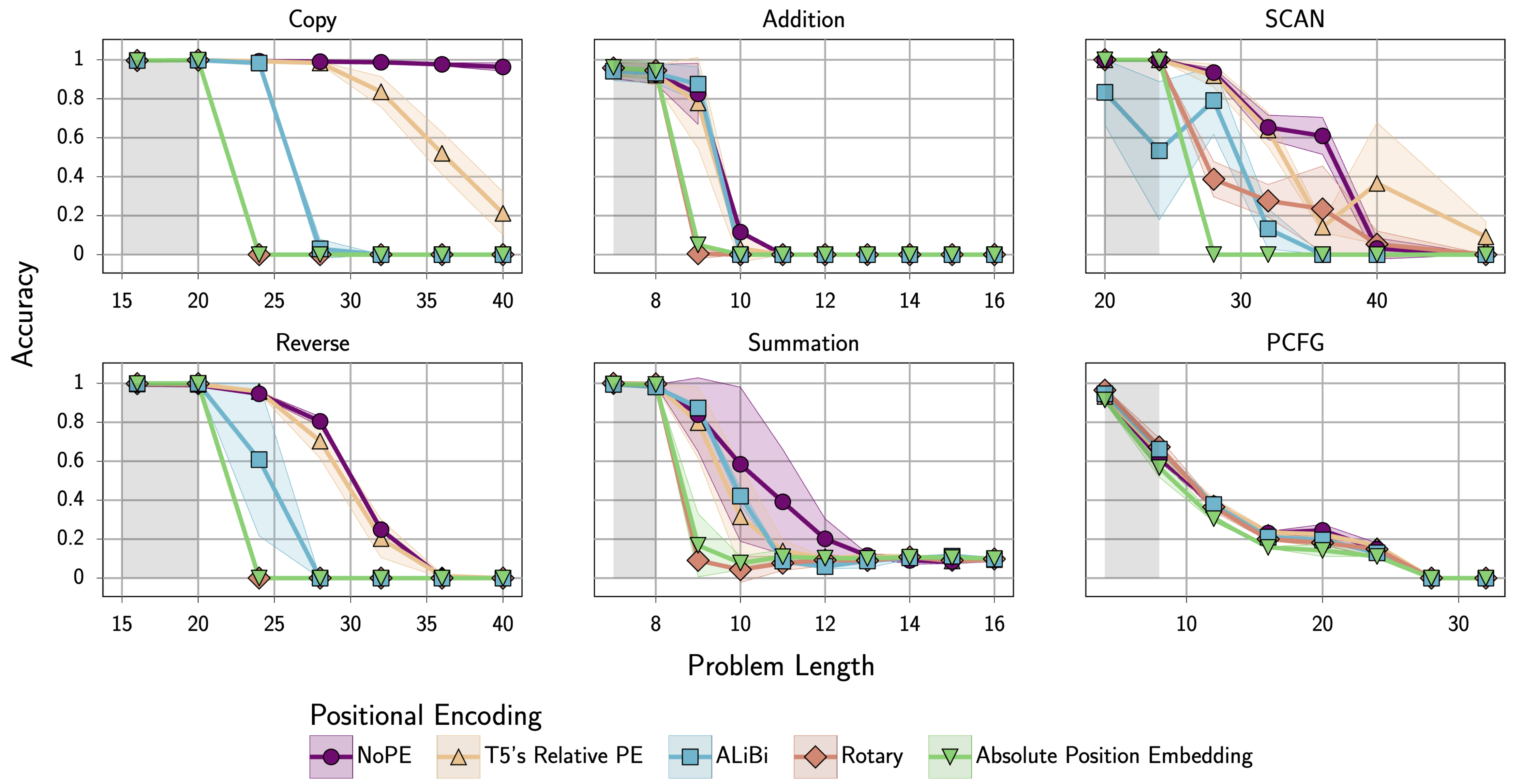
Su et al., 2021. RoFormer: Enhanced Transformer with Rotary Position Embedding.

Rotary position embedding (RoPE)

How well does RoPE generalize beyond the training lengths

Not so good …

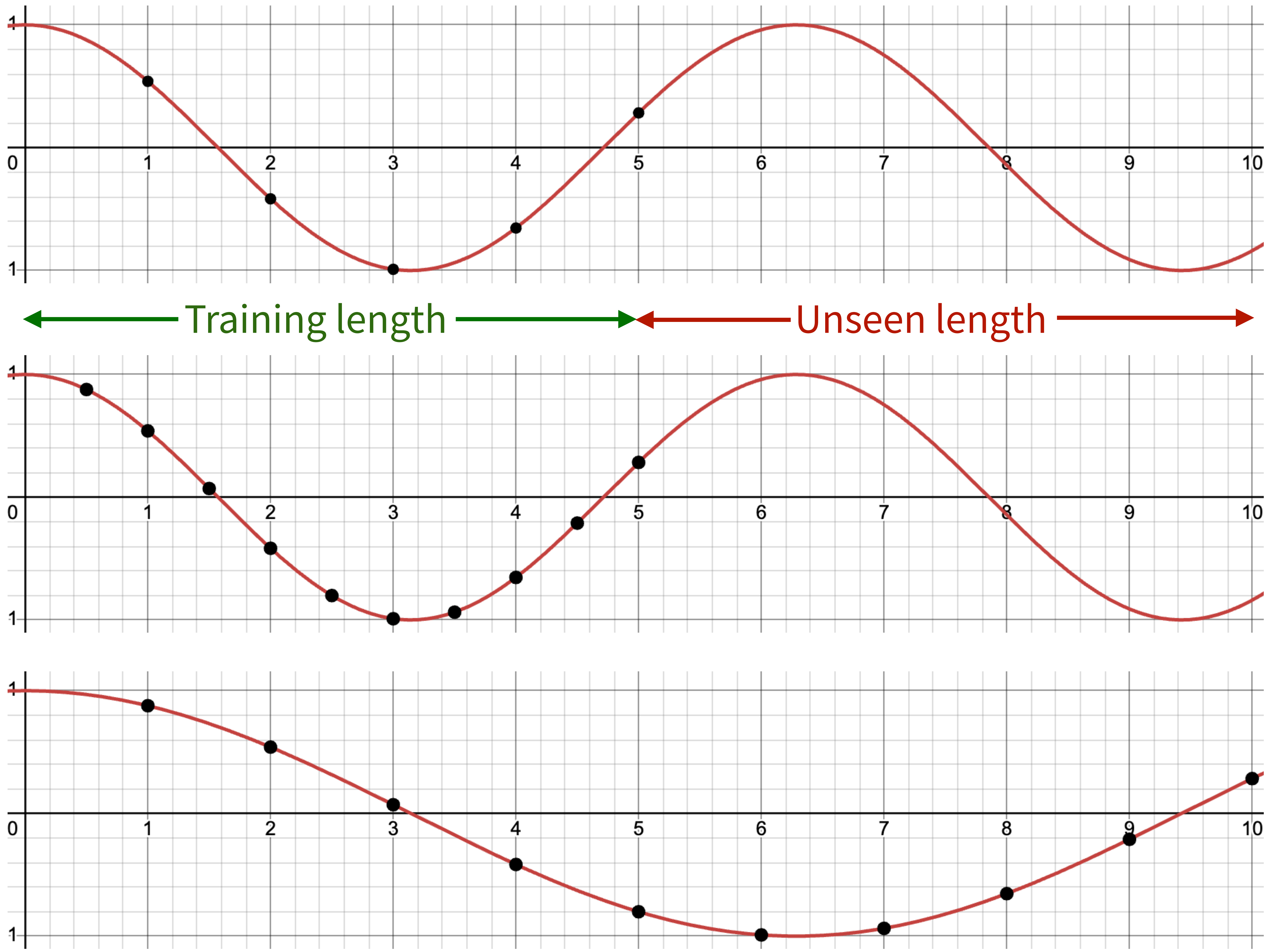




Press et al., 2021. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation.
Kazemnejad et al., 2023. The Impact of Positional Encoding on Length Generalization in Transformers.

Position extrapolation

Can we use some "tricks" to let the LM generalize to longer lengths? Yes!



Training length: 5

Testing length: 10

Position interpolation (PI)

$$m' = m/2$$

Change the frequency base ← better!

$$\theta_i = \boxed{10000}^{-2(i-1)/d}, i \in [1, 2, ..., d/2]$$

Chen et al., 2023. Extending Context Window of Large Language Models via Positional Interpolation

Lu et al., 2024. A controlled study on long context extension and generalization in llms.

https://www.reddit.com/r/LocalLLaMA/comments/14mrgpr/dynamically_scaled_rope_further_increases/

Position extrapolation

What happens when we change the frequency base?

d/2=512

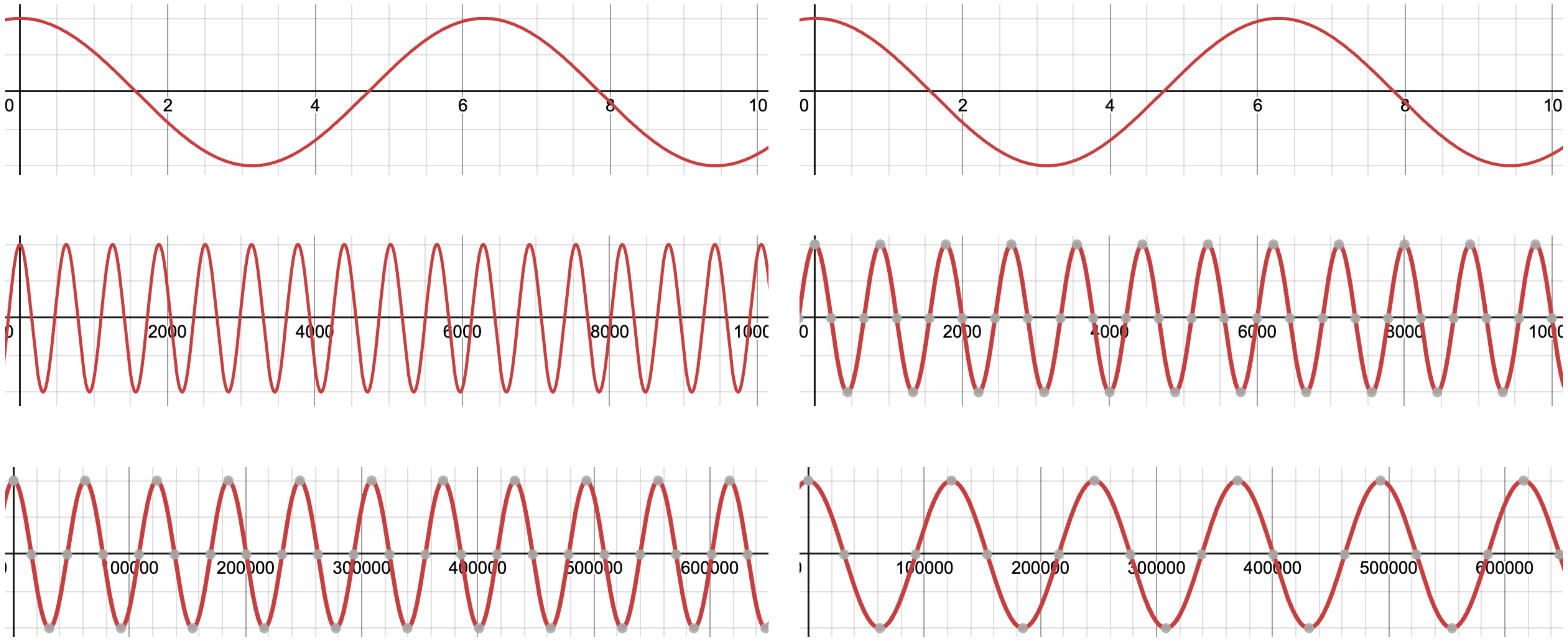
base=10000

base=20000

i=0

i=256

i=511



Position extrapolation

Can we use some “tricks” to let the LM generalize to longer lengths?

Yes (**with a little bit fine-tuning**)!

Attention Mechanisms		Model	PPL	Needle	Mshots	LongB	RULER
Exact Attention	Frozen	NTK-F	14.52	18.8	64.5	25.54	0.72
	Fine-Tuned	PI	5.85	42.1	**75.5**	33.48	57.66
		YaRN	5.85	46.7	75.0	33.45	36.95
		CLEX	5.82	71.1	74.0	33.48	52.17
		NTK-32K	**5.79**	**83.7**	71.0	**35.32**	59.42
		NTK-64K	5.93	69.1	73.0	34.30	**60.03**

Change frequency bases

What frequency bases to choose is more of an empirical question

Lu et al., 2024. A controlled study on long context extension and generalization in llms.

Problem solved?

It seems that with the **position extrapolation** trick + **a little bit fine-tuning**,

we can get long-context language models!

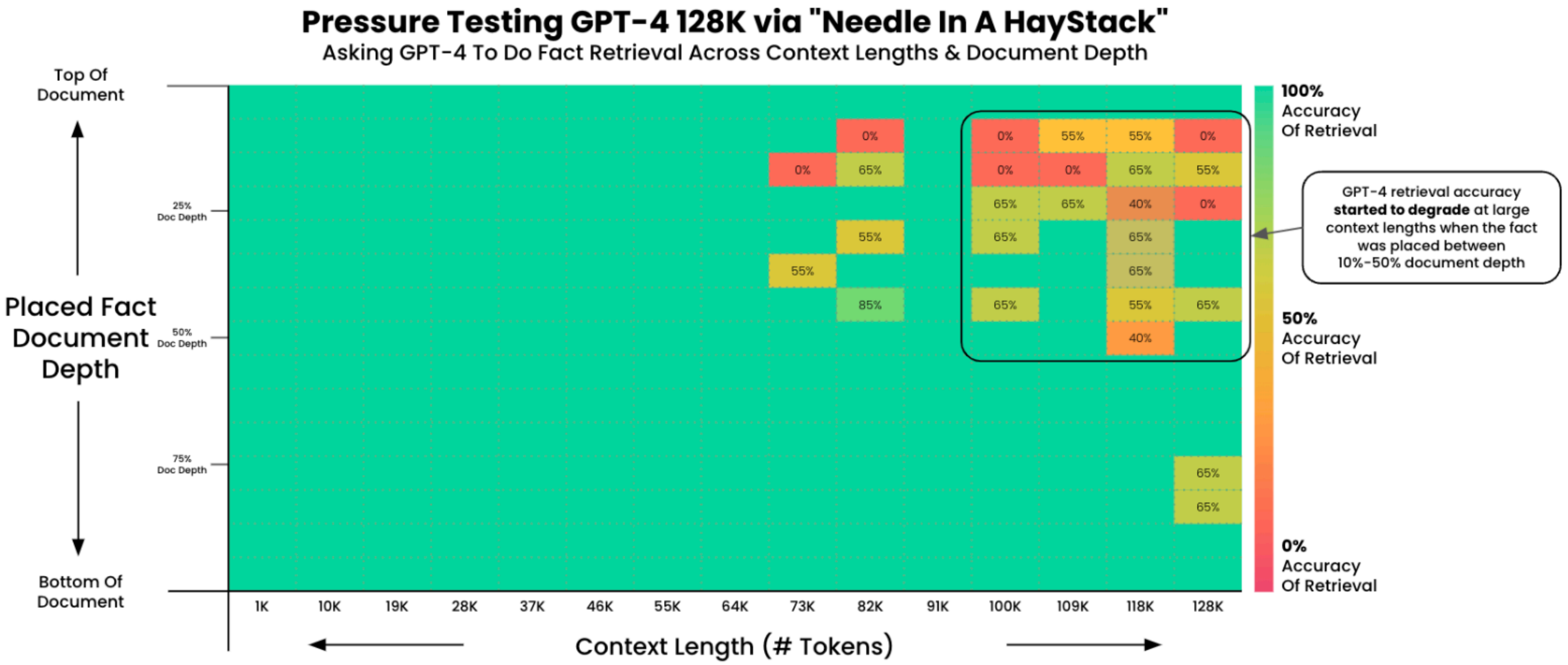
But how did we reach this conclusion?

- **Perplexity**: it is fine as long as the perplexity does not "explode" with longer-than-training length.
- **Synthetic tasks (needle-in-a-haystack, NIAH)**: whether the model can find the "needle" (a piece of information) from the haystack (irrelevant contexts).
- **Long QA/summarization datasets**: as long as the performance does not drop with longer evaluation lengths.

Can we trust those evaluations?

Existing benchmarks

Synthetic



(essays) ......
One of the special magic numbers for long-context is: 12345.
One of the special magic numbers for large-model is: 54321.
......
What is the special magic number for long-context mentioned in the provided text?
Answer: 12345

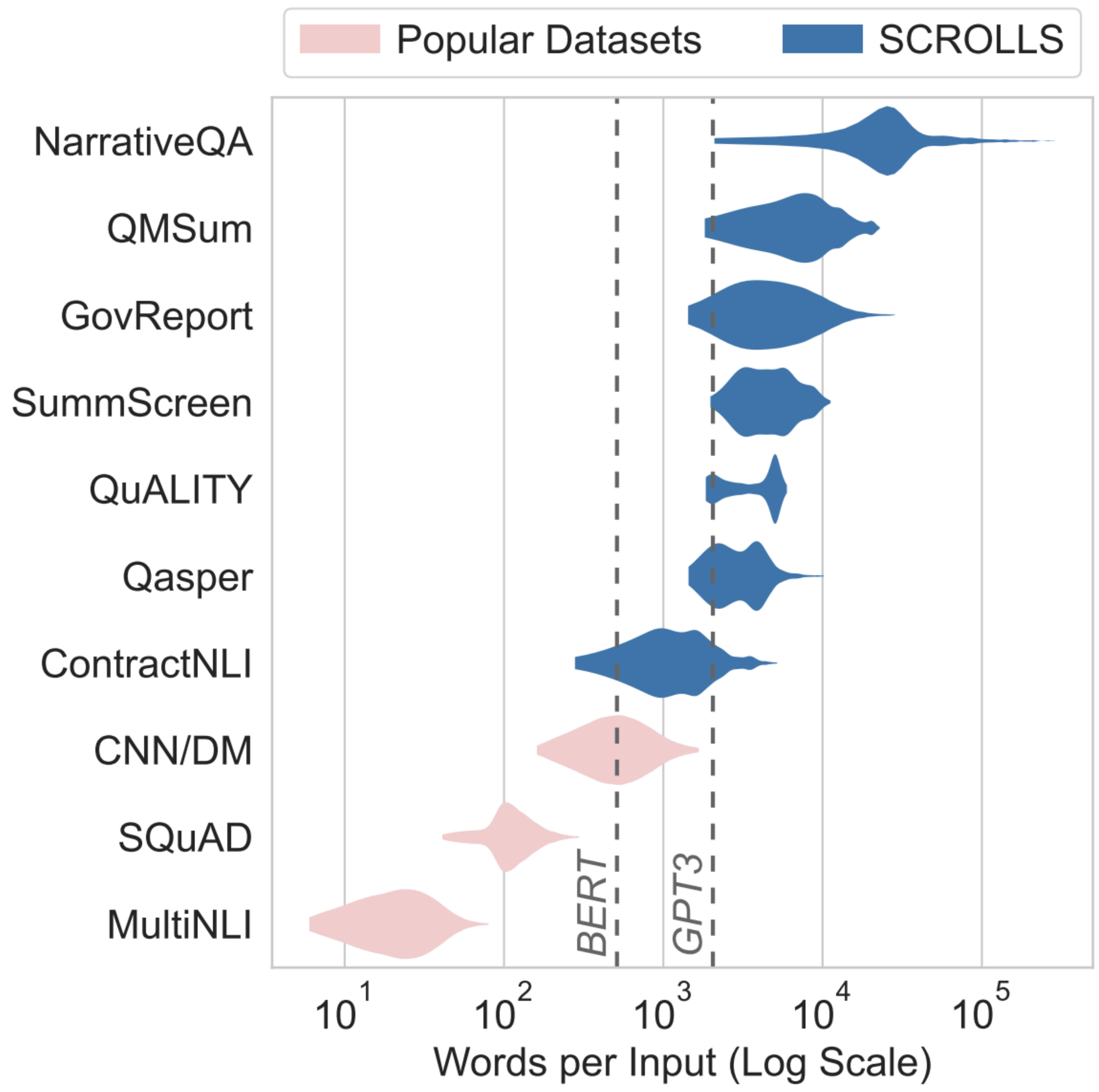
Needle in a haystack / RULER: unrelated context (Paul Graham essays) + a needle

- ✅ Easy to control (how long the “haystack” is and where the “needle” is); easy to evaluate
- ❌ The retrieving is super easy (irrelevant context; simple keys)
- ❌ Unclear whether they reflect real applications

Hsieh et al., 2024. RULER: What's the Real Context Size of Your Long-Context Language Models?

Existing benchmarks

“Long” QA / summarization



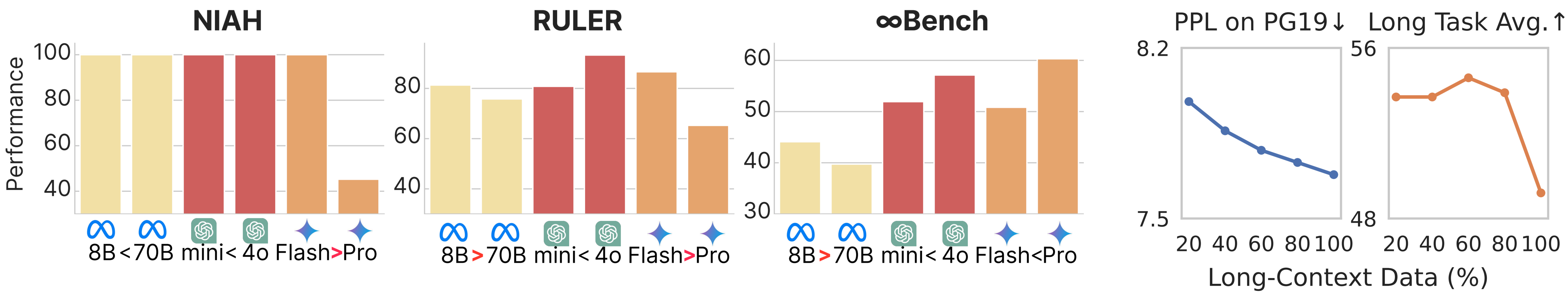
Datasets used by SCROLLS/ZeroSCROLLS

ZeroSCROLLS / L-Eval / BAMBOO / LongBench / LooGLE / InfiniteBench

- ✅ Include real datasets (mostly QA and summarization on books, papers, or meeting transcripts)
- ❌ No control over the length; most datasets used are very short (< 10K)
- ❌ No control over where the critical information is
- ❌ Broken evaluation

Shaham et al., 2022. SCROLLS: Standardized CompaRison Over Long Language Sequences

Existing benchmarks



Existing benchmarks show unintuitive trends

Perplexity does not correlate downstream tasks

Yen et al., 2024. HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly.
Gao et al., 2024. How to Train Long-Context Language Models (Effectively).

Model	Syn.	PPL	∞ QA	∞ All	ZeroSCROLLS NQA	ZeroSCROLLS QS	ZeroSCROLLS QL	ZeroSCROLLS SQ	ZeroSCROLLS All	RAG	ICL
Gemini-1.5	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
GPT-4	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Claude-3.5	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Llama-3.1	✓	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗
Phi-3	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗
Jamba-1.5	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Qwen2	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Command R	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Xiong et al.	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗
Chen et al.[♭]	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Peng et al.[♭]	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Fu et al.[♭]	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗

As a result, developers don't agree on what evaluations to use!

HELMET

- **HELMET is a new long-context benchmark that is diverse and reliable**
  - Covering a wide range of applications
    - Synthetic recall
    - Retrieval-augmented generation (RAG)
    - Passage re-ranking
    - Generation with citations
    - Long-document QA
    - Summarization
    - Many-shot in-context learning
  - Fixing many other issues from existing benchmarks (e.g., unreliable evaluation metrics)

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, Danqi Chen. 2024.
HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly.

HELMET

- **HELMET is a new long-context benchmark that is diverse and reliable**
  - Covering a wide range of applications
    - Synthetic recall
    - Retrieval-augmented generation (RAG)
    - Passage re-ranking
    - Generation with citations
    - Long-document QA
    - Summarization
    - Many-shot in-context learning
  - Fixing many other issues from existing benchmarks (e.g., unreliable evaluation metrics)

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, Danqi Chen. 2024.
HELMET: How to Evaluate Long-Context Language Models Effectively and Thoroughly.

HELMET tasks

Tasks

- **Synthetic recall**
- Retrieval-augmented generation (RAG)
- Passage re-ranking
- Generation with citations
- Long-document QA
- Summarization
- Many-shot in-context learning

Extract the value corresponding to the specified key in the JSON object below.
JSON data:
{"9f4a92b9-5f69-4725-ba1e-403f08dea695":
"703a7ce5-f17f-4e6d-b895-5836ba5ec71c",
…
"f4eb1c53-af0a-4dc4-a3a5-c2d50851a178":
“d733b0d2-6af3-44e1-8592-e5637fdb76fb"}

Key: "9f4a92b9-5f69-4725-ba1e-403f08dea695"
Corresponding value: *703a7ce5-f17f-4e6d-b895-5836ba5ec71c*

Metric: substring exact match

Liu et al., 2023. Lost in the Middle: How Language Models Use Long Contexts.

HELMET tasks

Tasks

- Synthetic recall
- **Retrieval-augmented generation (RAG)**
- Passage re-ranking
- Generation with citations
- Long-document QA
- Summarization
- Many-shot in-context learning

Write a high-quality answer for the given question using only the provided search results (some of which might be irrelevant).

Document [1](Title: Asian Americans in science and technology) Prize in physics …

Document [2](Title: List of Nobel laureates in Physics) The first Nobel Prize …
…

Question: who got the first nobel prize in physics
Answer: *Wilhelm Conrad Röntgen*

Metric: substring exact match

Liu et al., 2023. Lost in the Middle: How Language Models Use Long Contexts.

HELMET tasks

Tasks

- Synthetic recall
- Retrieval-augmented generation (RAG)
- **Passage re-ranking**
- Generation with citations
- Long-document QA
- Summarization
- Many-shot in-context learning

… Rank each document based on their relevance to the question in descending order …
Ranking: ID3 > ID1 > ID2

[ID: 4842895] Document: RSA Security is a United States-based …

[ID: 1929910] Document: Definition - What does RSA Encryption mean? RSA encryption …
…

Query: rsa definition key
Ranking: *1929910* > *4842895* > *9384722* > …

Metric: nDCG@10

HELMET tasks

Tasks

- Synthetic recall
- Retrieval-augmented generation (RAG)
- Passage re-ranking
- **Generation with citations**
- Long-document QA
- Summarization
- Many-shot in-context learning

Instruction: Write an accurate, engaging, and concise answer for the given question using only the provided search results and cite them properly …
For example: …

Document [1](Title: American Decolonization) …
Document [2](Title: Decolonization) ...
Document [3](Title: American Revolution) ...

Question: When did US break away from England?
Answer: *The United States took the first step towards gaining independence ... [1][2]. The Treaty of Paris was later signed ... [3].*

Metric: fluency, correctness, citation quality

Gao et al., 2023. Enabling Large Language Models to Generate Text with Citations.

HELMET tasks

Tasks

- Synthetic recall
- Retrieval-augmented generation (RAG)
- Passage re-ranking
- Generation with citations
- **Long-document QA**
- **Summarization**
- Many-shot in-context learning

You are given a story, which can be either a novel or a movie script, and a question. Answer the question as concisely as you can, using a single phrase if possible.

For example: …

Now, use the following story to answer the question:

{BOOK}

Question: Where was Almasy detained?
Answer: *El Tag.*

Metric: ~~F1 / ROUGE~~ model-based evaluation

HELMET tasks

Tasks

- Synthetic recall
- Retrieval-augmented generation (RAG)
- Passage re-ranking
- Generation with citations
- Long-document QA
- Summarization
- **Many-shot in-context learning**

power off
label: 40

please play this playback on audiobook
label: 53

what's the easiest and quickest way to cook a turkey
label: 47

…

cancel the meeting next week Thursday at two pm
label: *48*

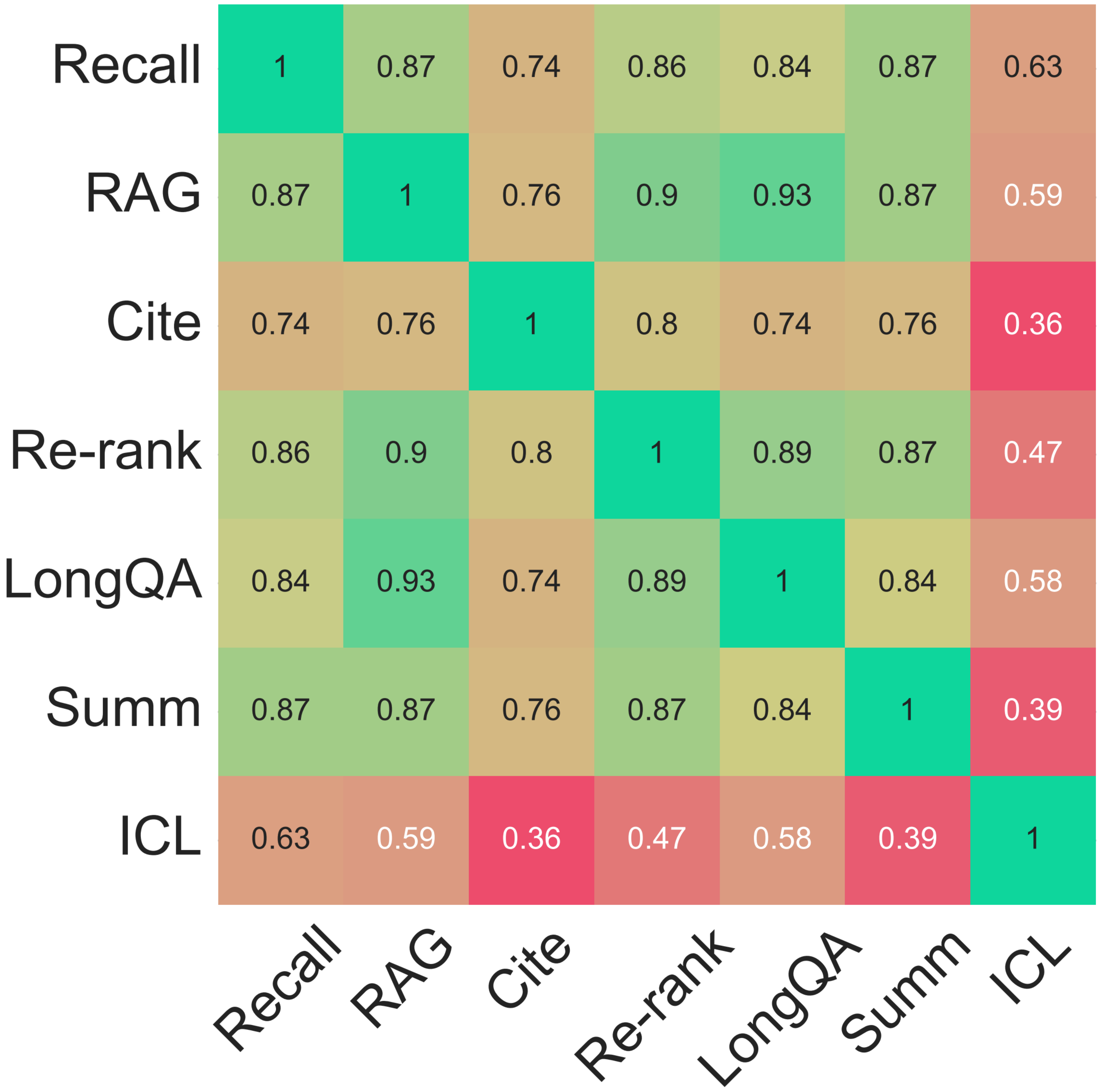
Metric: accuracy

Why do we need so many tasks?

Each task reflects distinct capabilities and (real-world) applications

- Synthetic recall: recall
- Retrieval-augmented generation (RAG): recall, robustness to noise
- Passage re-ranking: recall, long-context reasoning, long instruction following, robustness to noise
- Generation with citations: recall, long instruction following, robustness to noise
- Long-document QA: recall
- Summarization: long-context reasoning
- Many-shot in-context learning: recall, learning from context

Why do we need so many tasks?



Because they reflect distinct long-context abilities and do not correlate with each other.

(Spearman correlation on 30 long-context models' performance)

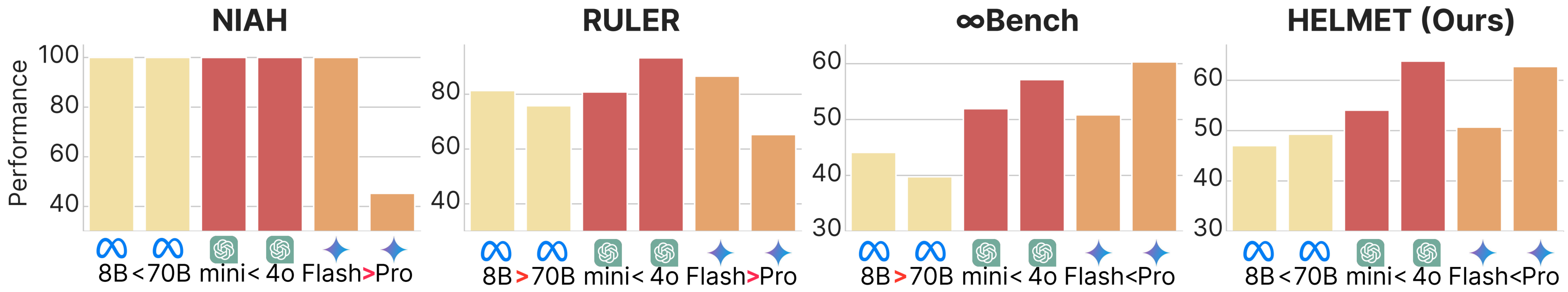
Why do we need so many tasks?

More complex

	NIAH					RAG					Re-rank					Cite				
	8k	16k	32k	64k	128k	8k	16k	32k	64k	128k	8k	16k	32k	64k	128k	8k	16k	32k	64k	128k
GPT-4o-08	100.0	100.0	100.0	100.0	100.0	73.4	73.8	72.4	71.1	70.8	75.6	73.1	67.4	59.5	47.9	45.8	47.1	46.4	45.7	45.3
Gemini-1.5-Pro	97.9	97.0	98.0	54.6	45.3	73.0	72.9	71.6	71.9	70.9	75.8	73.2	71.7	65.9	58.6	47.1	43.0	44.7	45.1	42.5
Llama-3.1-8B-Inst	100.0	100.0	100.0	100.0	100.0	69.1	67.9	64.8	64.6	59.0	58.7	45.9	42.0	31.9	15.0	35.4	26.9	12.6	12.8	3.4
Llama-3.1-70B-Inst	100.0	100.0	100.0	100.0	100.0	73.0	72.2	71.5	70.3	55.8	73.3	69.7	58.4	40.0	19.4	44.5	42.1	39.5	30.9	7.6
Jamba-1.5-Mini	100.0	100.0	100.0	100.0	100.0	66.2	65.0	64.0	63.4	56.6	53.5	43.0	35.6	23.2	14.6	15.4	10.0	5.7	3.1	2.5
Qwen2-7B-Inst	100.0	100.0	100.0	97.0	18.0	62.6	57.8	56.6	45.2	28.3	44.5	30.3	13.3	0.6	0.0	18.2	5.0	2.3	2.6	2.1
Qwen2-57B-Inst	100.0	100.0	100.0	90.0	37.0	65.3	64.6	63.5	55.3	11.6	44.3	37.7	17.1	5.2	0.0	35.5	11.0	4.0	3.4	0.9

Because introducing more challenging tasks better reveal the performance gap

HELMET reveals more consistent trends



	Type of tasks							Benchmark features		
	Cite	RAG	Re-rank	Long-QA	Summ	ICL	Synthetic Recall	Robust Eval.	$L \geq 128k$	Controllable L
ZeroSCROLLS	✗	✗	✗	✓	✓	✗	✗	✗	✗†	✗
LongBench	✗	✓	✗	✓	✓	✓	✓	✗	✗†	✗
L-Eval	✗	✓	✗	✓	✓	✗	✗	✓‡	✗†	✗
RULER	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
∞Bench	✗	✗	✗	✓	✓	✗	✓	✗	✓	✓
HELMET (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

So, how did position extrapolation perform?

	Recall					RAG				
	8k	16k	32k	64k	128k	8k	16k	32k	64k	128k
GPT-4	99.5	93.5	93.1	88.6	72.8	75.3	73.6	70.9	68.1	65.0
GPT-4o-05	94.7	93.4	91.2	87.9	81.6	74.1	73.1	71.8	71.1	71.0
GPT-4o-08	99.8	99.4	97.9	97.0	97.0	73.4	73.8	72.4	71.1	70.8
GPT-4o-mini	100.0	99.8	99.1	92.0	83.6	72.6	71.0	69.6	68.3	66.7
Claude-3.5-sonnet	99.9	97.2	96.2	95.2	93.3	60.4	52.8	51.1	39.8	41.1
Gemini-1.5-Flash	93.5	93.6	93.2	92.5	87.8	71.6	69.9	69.6	68.6	67.6
Gemini-1.5-Pro	81.3	83.6	86.9	87.1	84.1	73.0	72.9	71.6	71.9	70.9
Llama-3-8B-θ	93.9	85.5	50.9	0.0	0.0	60.1	61.0	56.5	7.4	1.0
Llama-3-8B-θ	98.8	93.6	74.5	0.0	0.0	66.4	63.0	60.2	3.8	0.2
Llama-3-70B-θ	91.3	86.0	63.7	0.0	0.0	65.7	64.6	61.4	2.6	0.0
Llama-3-70B-θ	99.7	98.2	74.7	0.0	0.0	69.9	69.8	66.2	1.1	0.0
Llama-3.1-8B	91.1	93.3	81.9	78.7	69.6	64.7	64.8	62.4	60.5	53.7
Llama-3.1-8B	99.4	99.6	97.2	98.3	91.1	69.1	67.9	64.8	64.6	59.0
Llama-3.1-70B	89.4	90.0	81.1	79.3	71.6	67.5	67.2	66.1	64.6	55.9
Llama-3.1-70B	99.9	99.8	98.0	87.4	84.4	73.0	72.2	71.5	70.3	55.8
Llama-3.2-1B	56.3	45.4	29.4	31.3	17.3	43.7	41.2	39.0	37.0	35.1
Llama-3.2-1B	72.5	60.5	42.0	42.4	28.8	48.8	45.3	42.3	40.1	34.4
Llama-3.2-3B	81.9	77.7	70.4	57.6	49.6	62.2	61.6	59.1	57.5	51.4
Llama-3.2-3B	97.3	88.9	76.3	57.1	47.5	64.3	64.6	62.1	60.2	57.0
Llama-7B-80k	69.5	57.4	43.2	26.9	16.2	55.0	52.6	51.8	49.1	39.0
Yarn-Llama-2-7B-128k	39.5	25.3	13.4	12.1	5.2	54.5	50.8	49.8	41.4	26.0

Not good enough!

How can we get long-context models

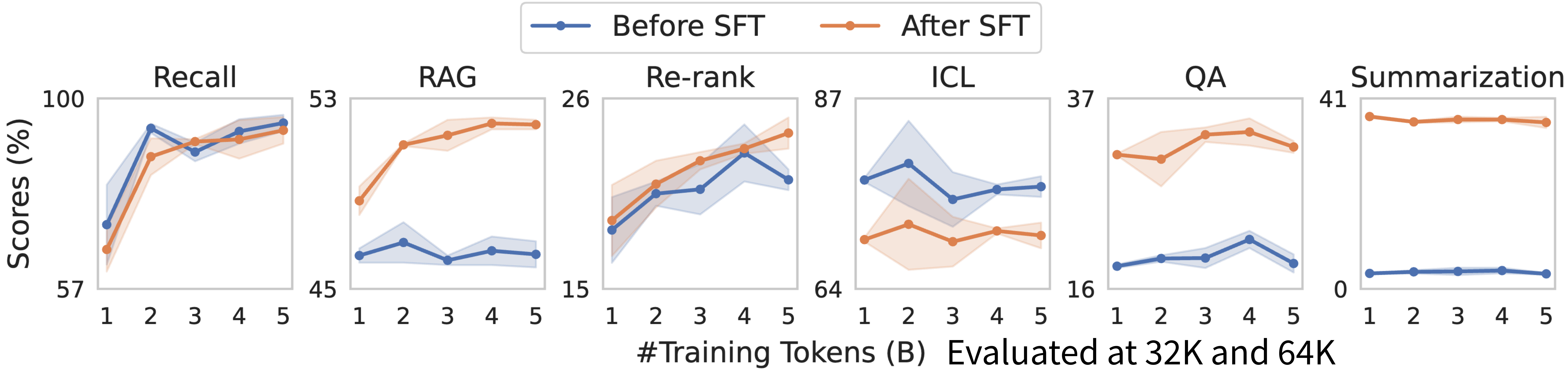
- **Position extrapolation**
  - It often only avoids “perplexity explosion”, but fails to do real long-context tasks
- **Training from scratch**
  - Extremely expensive
- **Continuing pre-training an LM for long contexts**
  - Relatively cheap
  - Much more effective than training-free methods
- **Alternative/hybrid architectures**

ProLong: Princeton Long-Context Language Model

- **Setting:** continued training + SFT a short-context LM for long-context use
- **Research questions in developing long-context models**
  - How to evaluate?
  - What data do we train on?
  - How long do we train the model on/for?
  - How to do supervised fine-tuning (SFT)?

Tianyu Gao*, Alexander Wettig*, Howard Yen, Danqi Chen. How to Train Long-Context Language Models (Effectively).

Q1: How to evaluate?



Evaluated at 32K and 64K

(1) SFT first, then evaluate

- SFT did not amplify the variance
- SFT enables evaluations on QA/summarization tasks
- On RAG/re-ranking, SFT shows clearer signals (and different trends from base models)

Pre-training data: Fu et al., 2024. Data Engineering for Scaling Language Models to 128K Context.
SFT data: UltraChat. LR = 1e-5, rope theta = 8M. Averaged over two seeds. Base model: Llama-3-8B-base.

Q1: How to evaluate?

	HSwag	MMLU	ARC-c	WG	GSM8K
Llama-3-8B	82.1	66.5	59.4	77.1	44.7
+ PE	81.5	64.7	58.1	75.5	40.1
+ SlimPajama	81.0	63.1	57.8	75.1	40.6

PE: position extrapolation.
SlimPajama: training on Fu et al., 2024 for 5B.

(2) Also evaluate on short-context task performance

- Naive position extrapolation (PE) or fine-tuning on arbitrary data hurts short-context performance.

Q2: What data do we train on?

Long data + short data

Data	#Long tokens
Code Repos	**98.8B**
SP/Books	**33.2B**
SP/CC	15.3B
SP/Arxiv	5.2B
SP/GitHub	2.8B
SP/Wiki	0.1B
SP/StackEx	<0.1B
SP/C4	<0.1B

#Tokens from documents > 64K.
SP: SlimPajama.

What long data sources can we use?

- Code repositories and books are the most abundant sources
- Mixing them leads to the best performance

Long Data (60%)	Long-Context							Short-Context
	Recall	RAG	Re-rank	ICL	QA	Summ.	Avg.	Avg.
CommonCrawl	84.1	53.3	28.1	67.5	35.2	37.0	50.9	66.5
Books	94.9	53.9	**30.7**	72.2	33.2	37.7	53.8	65.5
Code Repos	**99.2**	53.8	29.0	61.2	34.7	36.2	52.3	65.9
Books/Repos 1:1	96.0	**54.9**	29.4	**73.9**	**35.7**	**37.9**	**54.6**	65.5

Evaluated at 32K and 64K

Q2: What data do we train on?

Long data + short data

Table 5: Our ShortMix.

Components	%
FineWeb	25
FineWeb-Edu	25
Wikipedia	10
Tulu-v2	10
StackExchange	10
ArXiv	10
OpenWebMath	10

What short data sources should we use?

- Does it matter what short data we use? Yes!
- A good pre-training data source is not necessarily good short data here!

Short Data (40%)	Long-Context	Short-Context					
	Avg.	HellaS.	MMLU	ARC-c	WG	GSM8K	Avg.
Original model (Llama-3-8B)	-	82.1	66.5	59.4	77.1	44.7	66.0
SlimPajama	52.9	81.2	63.0	58.5	76.2	41.9	64.2
FineWeb-Edu	53.0	81.0	62.6	57.7	74.4	39.4	63.0
DCLM-Baseline	52.0	82.0	65.6	59.6	77.4	39.4	64.8
ProLong ShortMix	**54.6**	81.6	65.3	58.0	76.2	46.6	**65.5**

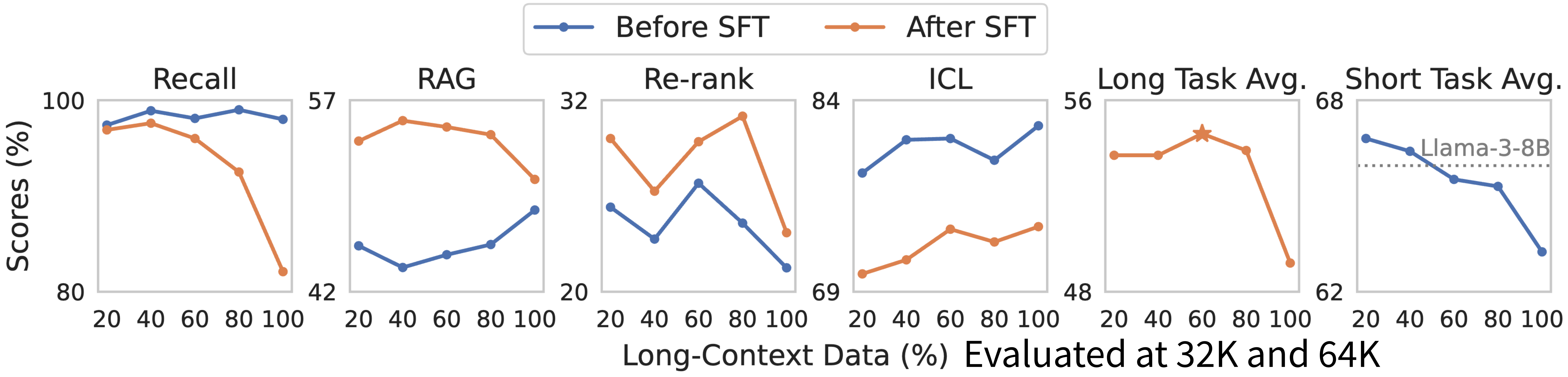
Evaluated at 32K and 64K

Q2: What data do we train on?

Long data + short data

How much long and how much short?

- Only training on long data hurts long-context performance (especially after SFT)!
- Training on 60% long + 40% short is the best



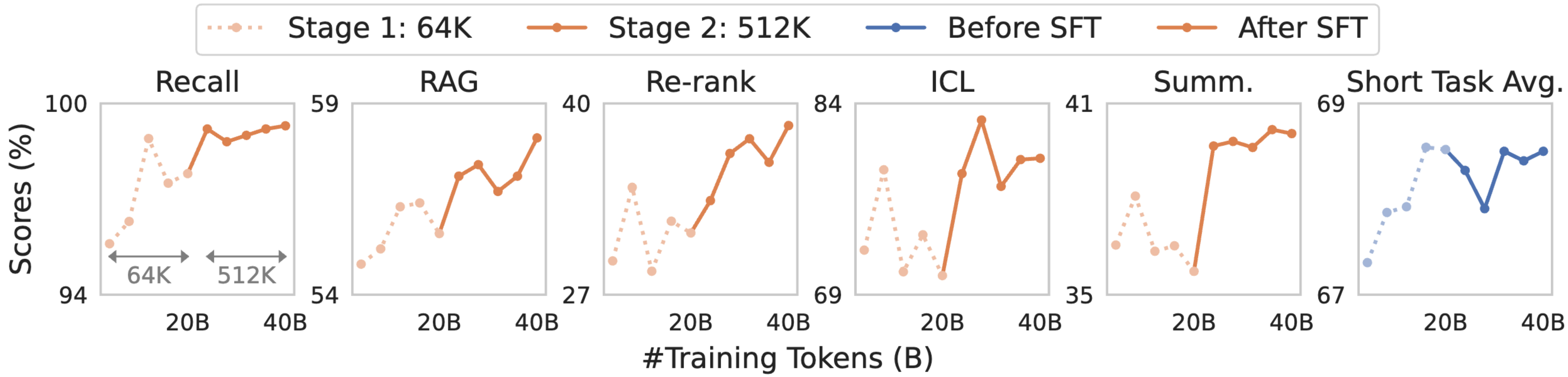
Evaluated at 32K and 64K

Setting: train with long and short data mix. For long mix, 50% code and 50% books. LR = 1e-5 for 5B.
Base model: Llama-3-8B-base.

Q3: How long do we train the model on/for?

Does further scaling data sizes help?

- Yes! ProLong is trained on 64K data for 20B and 512K data for 20B
- More training continues to improve the model performance



Q3: How long do we train the model on/for?

What context lengths should the model be trained on?

- Training on a length longer than the test length significant improves the result.

Max Seq. Length	Recall	RAG	Re-rank	ICL
ProLong 64K training (20B)	96.5	52.7	22.8	70.6
+4B 64K training	95.0	56.4	28.0	78.8
+4B 512K training	**98.5**	**56.9**	**32.9**	**79.2**

All evaluated at 64K.

Q4: How to do supervised fine-tuning (SFT)

Do we need long-context instruction data?

- UltraChat (our SFT data) only has an average length of **1.2K tokens**
- Previous work (including Llama-3.1) all suggests the use of **synthetically generated long instruction data**
- An example (synthetic QA data)
  - Sample a long document from the pre-training corpus
  - Sample a short paragraph from the document
  - Prompt an existing LM (can be short-context) to generate a QA pair
  - Concatenate the original document and the QA pair

Llama-3 team. 2024. The Llama 3 Herd of Models.

Q4: How to do supervised fine-tuning (SFT)

Table 8: Effect of different ratios of synthetic SFT data (mixed with UltraChat). We report the 32K-and-64K-averaged performance except tasks marked with †, which are evaluated at 512K for stress testing. The number of percentage is based on #tokens, not #samples.

% Synthetic Data	JsonKV†	RAG	Re-rank	ICL	QA†	Summ.†	Avg.
0%	65.7	58.1	38.5	80.3	49.7	42.1	**55.7**
1%	61.5	57.0	38.3	80.8	45.3	41.5	54.1
3%	62.0	56.4	37.9	80.6	44.8	39.5	53.5
10%	70.3	55.5	36.1	80.6	41.7	39.4	53.9
50%	45.8	48.8	18.8	70.5	42.3	33.3	43.3

Do we need long-context instruction data?

- We generated synthetic QA, summarization, and RAG instruction data
- In our experiment, mixing even 1% of long instruction data hurts the performance
- Only training on short SFT data is sufficient!

Q*: So, do we still need position extrapolation?

Yes (even with a lot of training)!

… and it should be carefully ablated

Table 18: Ablation study on RoPE frequency base at a maximum training length of 64K. Dynamic NTK (emozilla, 2023) roughly suggests to use 4m as the frequency base.

RoPE Base ($\times 10^6$)	Long-Context							Short-Context
	Recall	RAG	Re-rank	ICL	QA	Summ.	Avg.	Avg.
0.5	25.8	37.0	4.4	73.8	17.5	16.3	29.1	65.0
4.0	81.3	47.8	18.2	76.5	31.8	36.3	48.7	65.3
8.0	96.0	54.9	29.4	73.9	35.7	37.9	**54.6**	**65.5**

Table 19: Ablation study on RoPE frequency base at a maximum training length of 512K. Dynamic NTK (emozilla, 2023) roughly suggests to use 64×10^6 as the frequency base.

RoPE Base ($\times 10^6$)	Long-Context							Short-Context
	Recall	RAG	Re-rank	ICL	QA	Summ.	Avg.	Avg.
64	98.8	57.8	30.4	82.2	38.2	38.3	57.6	68.3
128	98.8	57.4	30.7	80.0	40.4	38.8	57.7	**68.6**
256	98.8	56.8	33.8	79.8	37.9	39.7	**57.8**	68.4

ProLong performance

With careful ablations on data, scaling, and SFT

- ProLong is the strongest long-context LM under 10B.
- Better than Llama-3.1 with only 5% of its long-context training budget.

Model	Max Len.	Recall	RAG	ICL	Re-rank	QA	Summ.	Avg.
ProLong (8B)	512K	**99.4**	**66.0**	81.1	**33.2**	**40.8**	40.5	**60.2**
MegaBeam-Mistral (7B)	512K	**99.4**	58.1	**82.1**	22.1	33.7	43.6	56.5
Meta-Llama-3.1 (8B)	128K	98.7	62.8	79.7	26.6	40.4	**46.1**	59.0
Qwen2 (7B)	128K	34.4	43.4	54.8	4.6	23.3	38.5	33.2
Phi-3-small (7B)	128K	74.8	60.6	82.0	18.5	34.1	42.4	52.1
Mistral-Nemo (12B)	128K	24.9	48.1	82.0	4.7	37.7	37.0	39.1
Jamba-1.5-Mini (12B/52B)	256K	87.7	61.3	88.4	25.9	42.0	38.6	57.3
Meta-Llama-3.1 (70B)	128K	98.5	65.9	80.0	39.4	47.2	51.1	63.7
Claude-3.5-Sonnet	200K	99.4	44.0	79.3	19.9	38.1	49.2	55.0
Gemini-1.5-Pro	2M	94.2	71.4	78.9	65.3	44.4	56.2	68.4
GPT-4o	128K	99.9	71.5	86.7	59.6	47.0	55.7	70.1

Evaluated at 32K, 64K, 128K (averaged).

Conclusion

How to reach long-context language models?

- **Position extrapolation**: tricks to avoid language models from OOD collapsing
- **Evaluation**: diverse, application-centric evaluation is needed
- **Training**: careful data engineering is important; scaling still matters

Contact: tianyug@princeton.edu