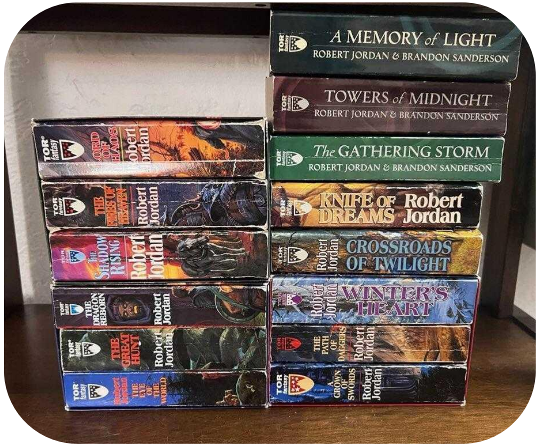
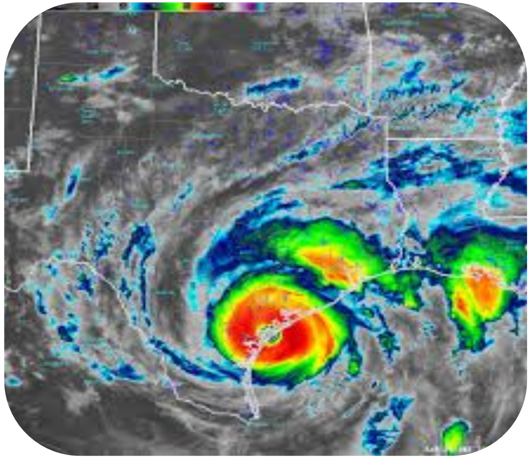
Long context modeling and generalization: two perspectives

Amanda Bertsch

We’d like our models to do tasks that are hard for people



Write a detailed summary of the plot of this book series



Prioritize disaster response using the last 2hr of social media posting

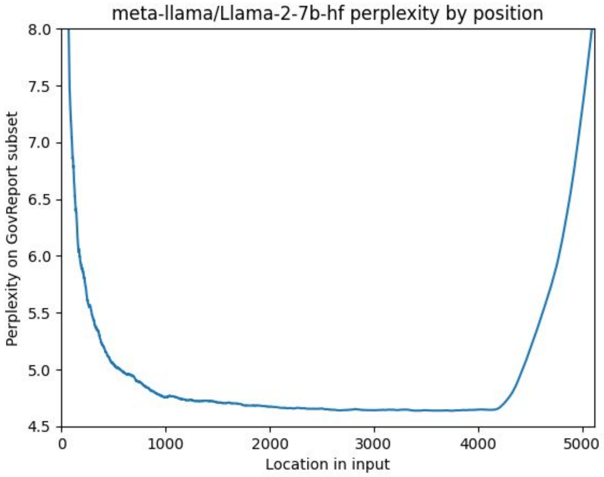
https://www.weather.gov/hgx/hurricaneharvey

NEURAL INFORMATION PROCESSING SYSTEMS

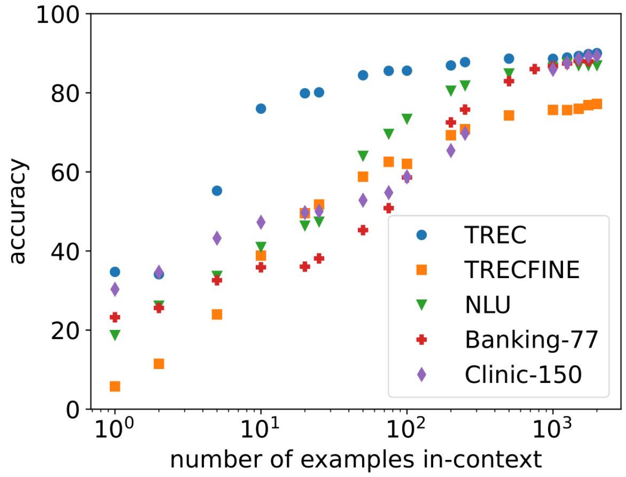
Identify the different research trends in NeurIPS 2022 and 2023

Long context and generalization: two parts

Long-context as (length) generalization

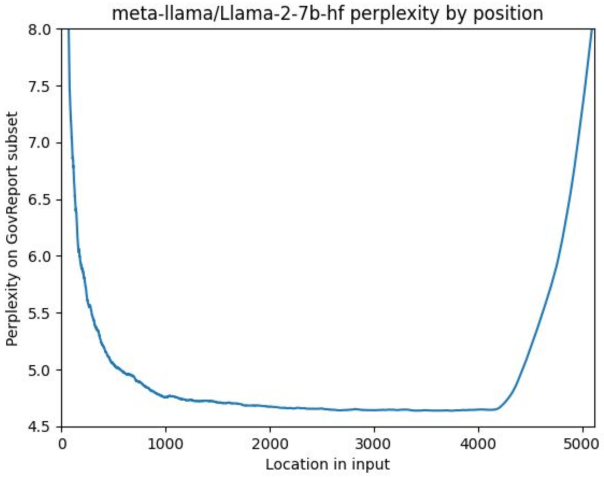


Long-context ICL

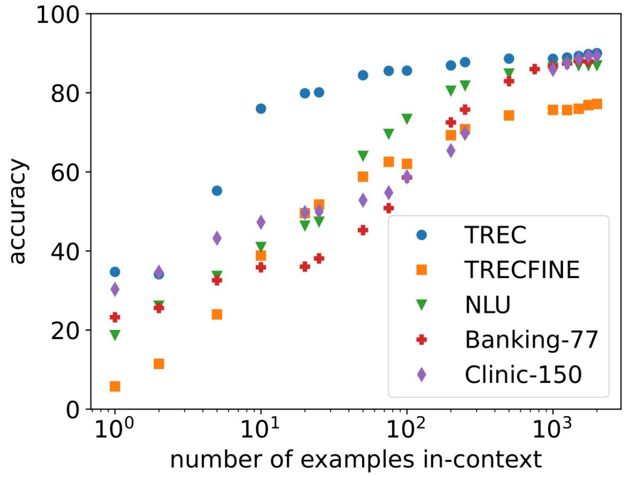


Long context and generalization: two parts

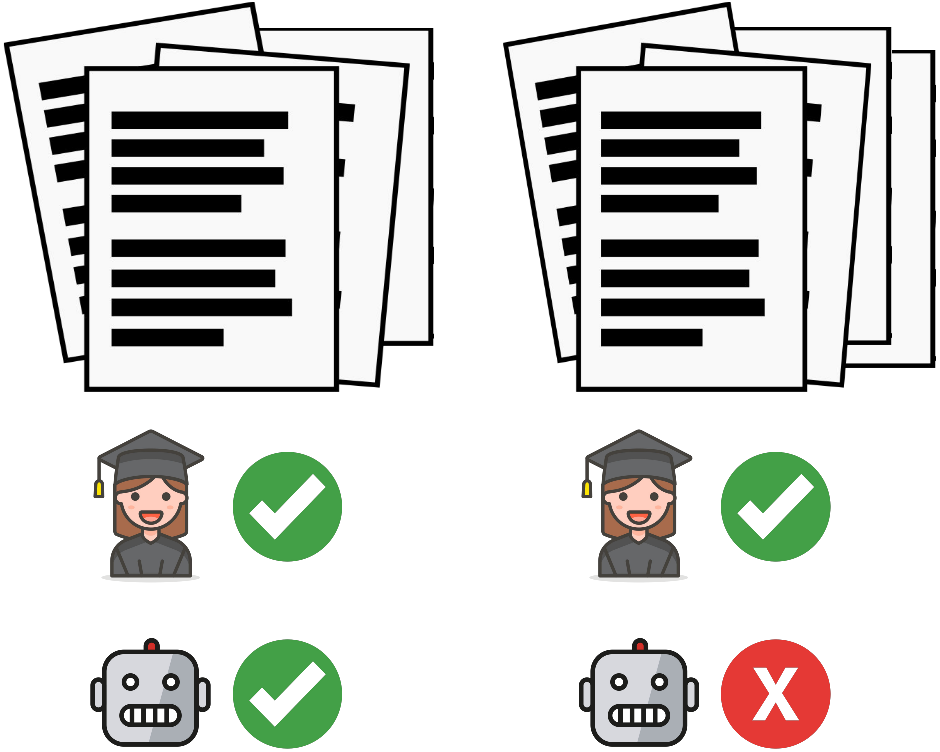
Long-context as (length) generalization

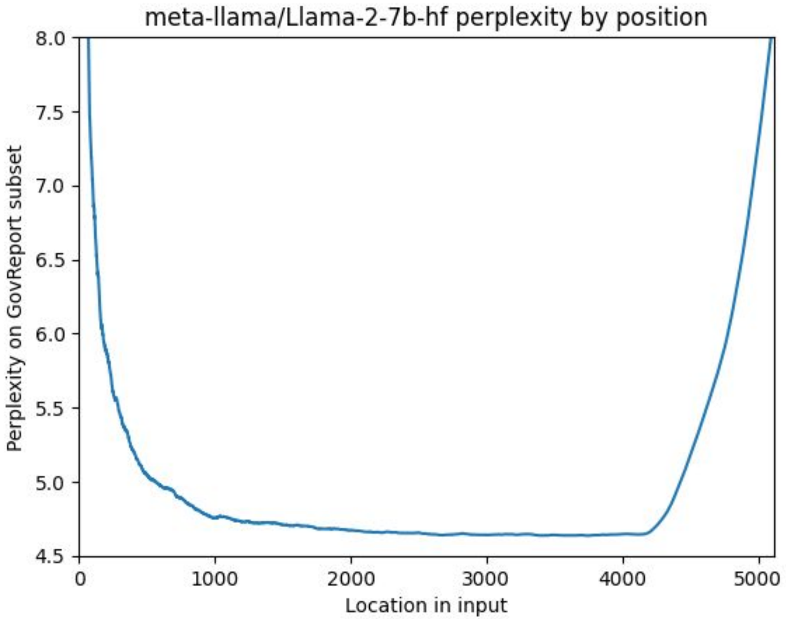


Long-context ICL



Length generalization





Length generalization strategies

Data-side interventions to reduce the difference from pretrained length

Data-side interventions: long context without the long

- Retrieval augmented generation
- Input trimming
- Hierarchical summarization
- “Memories” of prior conversations / episodes / encounters
- Principle learning
- Finetuning

- Tokenization

Generalization is still hard...

Data-side interventions

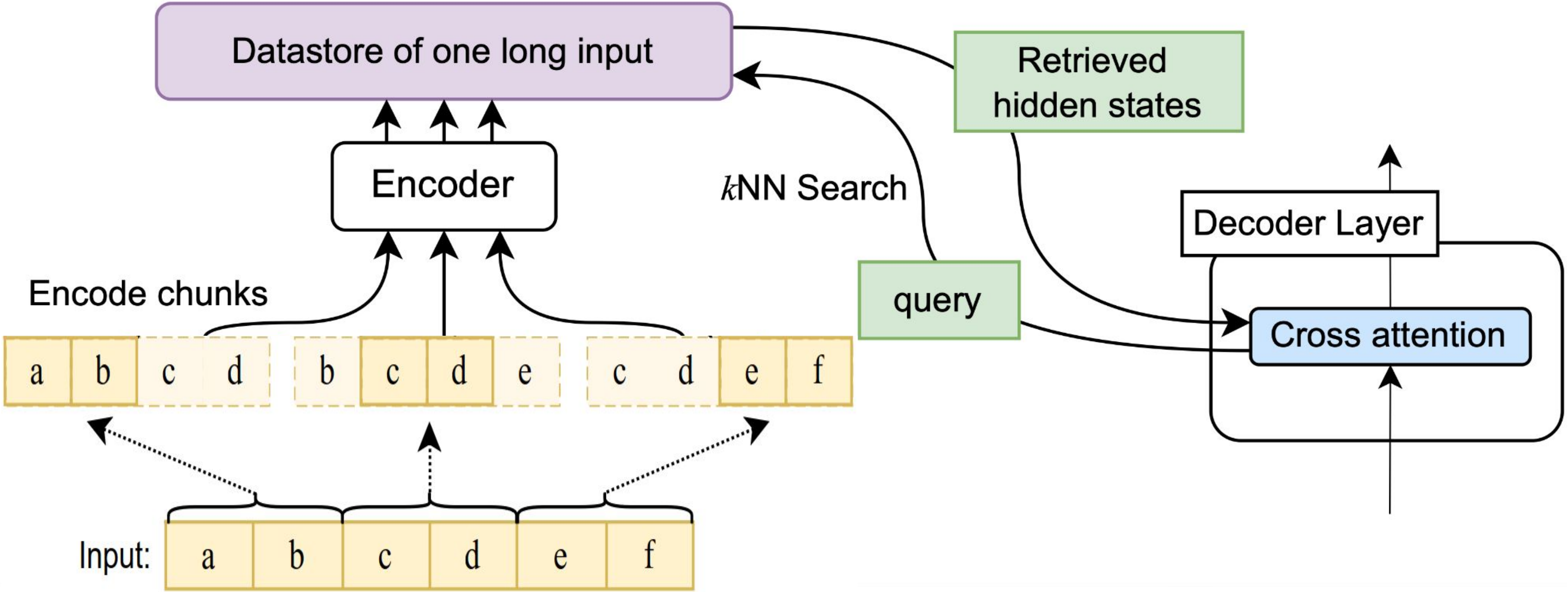
Not possible for every task, imposes additional assumptions

Model-side interventions to reduce the difference from pretrained length

Retrieval at attention time

Key idea: encode everything through the model, then choose a much smaller subset to attend to in order to reduce the difference from the pretraining setting

Unlimiformer encoder-decoder (NeurIPS 2023)



How do we choose the context window? cross-attention

decoder hidden state

encoder hidden state

layer specific
head specific

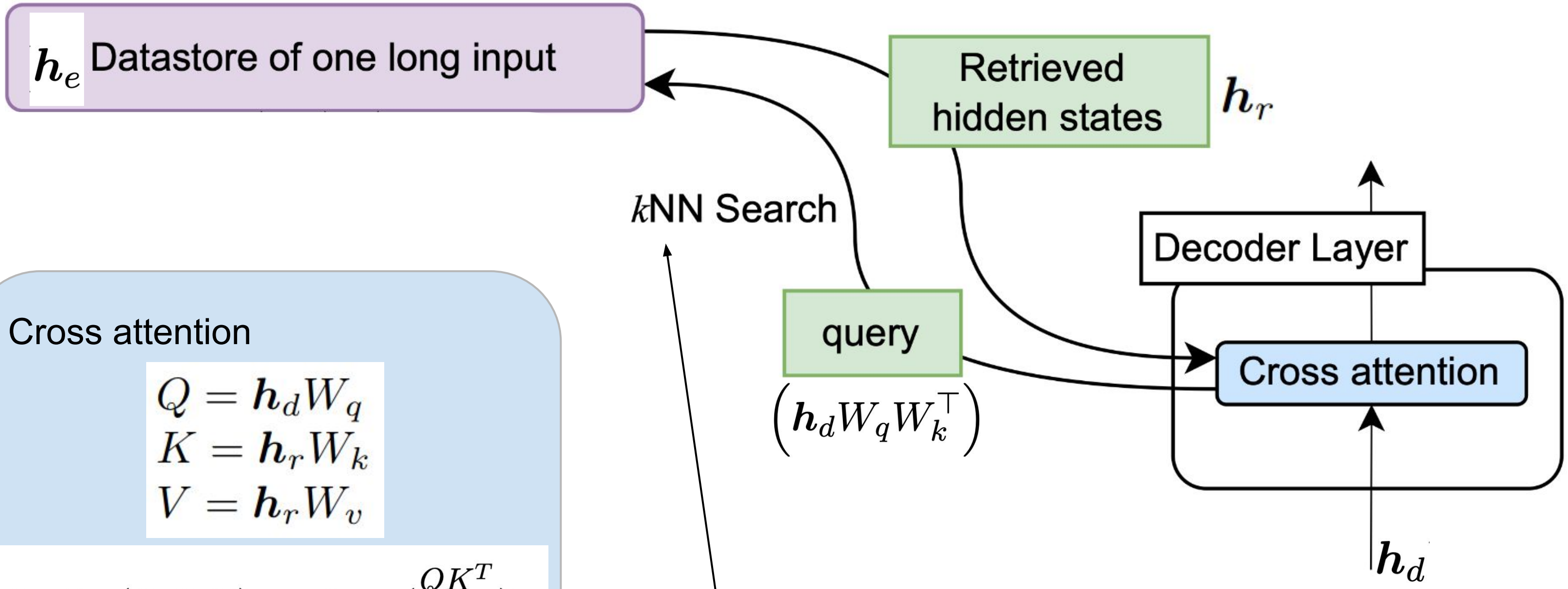
$$QK^T = (\boldsymbol{h}_d \boxed{W_q})\,(\boldsymbol{h}_e \boxed{W_k})^\top$$

Memorizing Transformers (Wu et al. ICLR'2022)
kept two datastores for each <layer,head> pair
<u>Overall datastores</u>: 2 X layers X heads

Project the query differently for every layer/head

We can keep a **single** datastore of the encoded hidden states

How do we choose the context window?



Cross attention

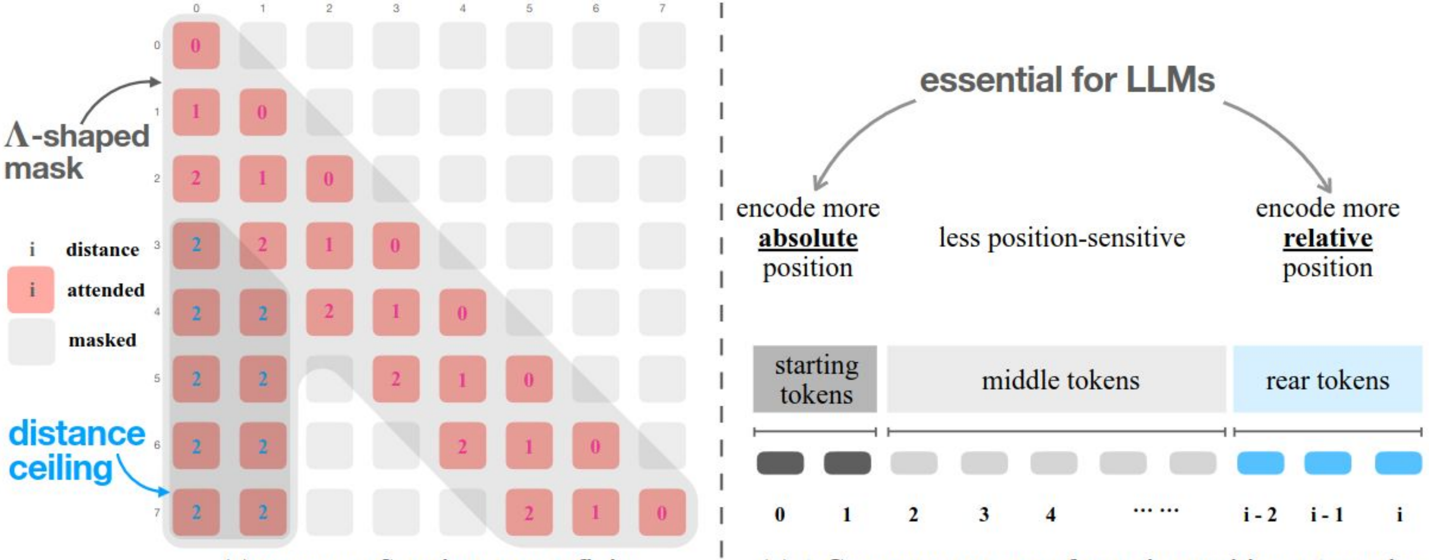
$$Q = \boldsymbol{h}_d W_q$$
$$K = \boldsymbol{h}_r W_k$$
$$V = \boldsymbol{h}_r W_v$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

faiss Public

Approximate, sublinear search

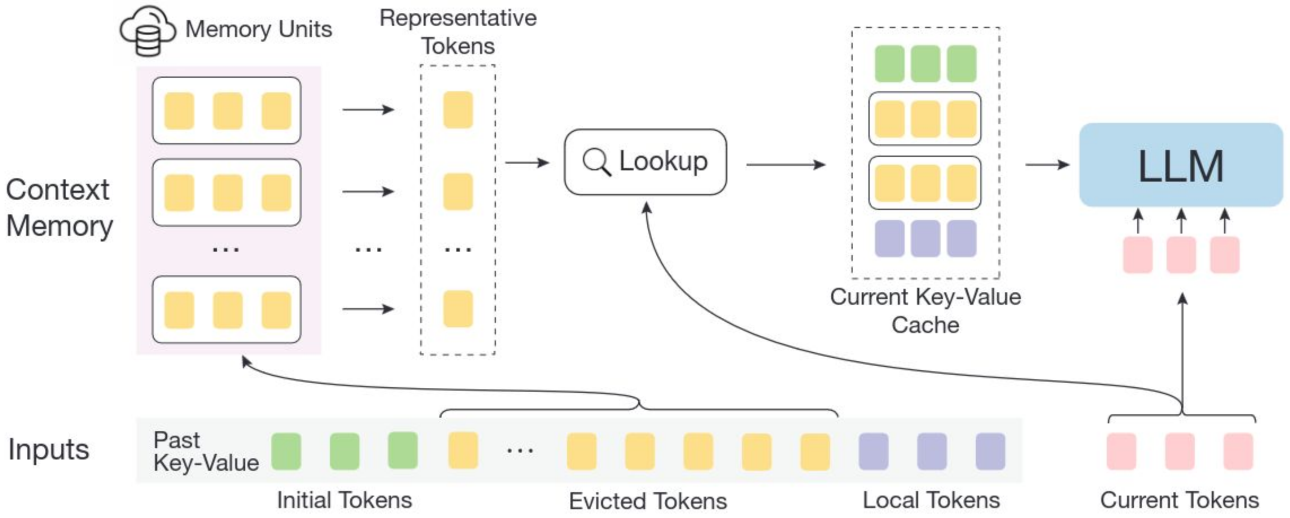
LM-Infinite: Han et al 2024



(a) Proposed Solution: LM-Infinite

(b) A Conceptual Model of Relative Positional Attention

InfLLM: Xiao, Zhang et al 2024



Generalization is still hard...

Data-side interventions

Not possible for every task, imposes additional assumptions

Model-side interventions

Not as effective as full finetuning, decreasingly effective with length

Position embedding modifications for better length generalization

What goes wrong when generalizing to longer input?

Implicit positional information in the network?

Language Modeling with Deep Transformers

Kazuki Irie[1], Albert Zeyer[1,2], Ralf Schlüter[1], Hermann Ney[1,2]

The Impact of Positional Encoding on Length Generalization in Transformers

Amirhossein Kazemnejad[1], Inkit Padhi[2]
Karthikeyan Natesan Ramamurthy[2], Payel Das[2], Siva Reddy[1,3,4]

What goes wrong when generalizing to longer input?

Positional embedding extrapolation?

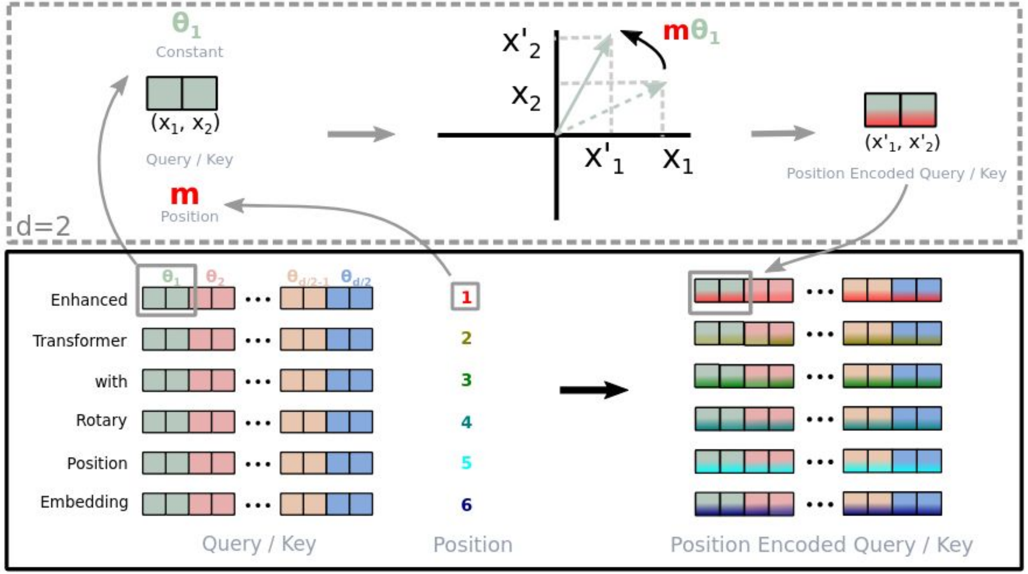
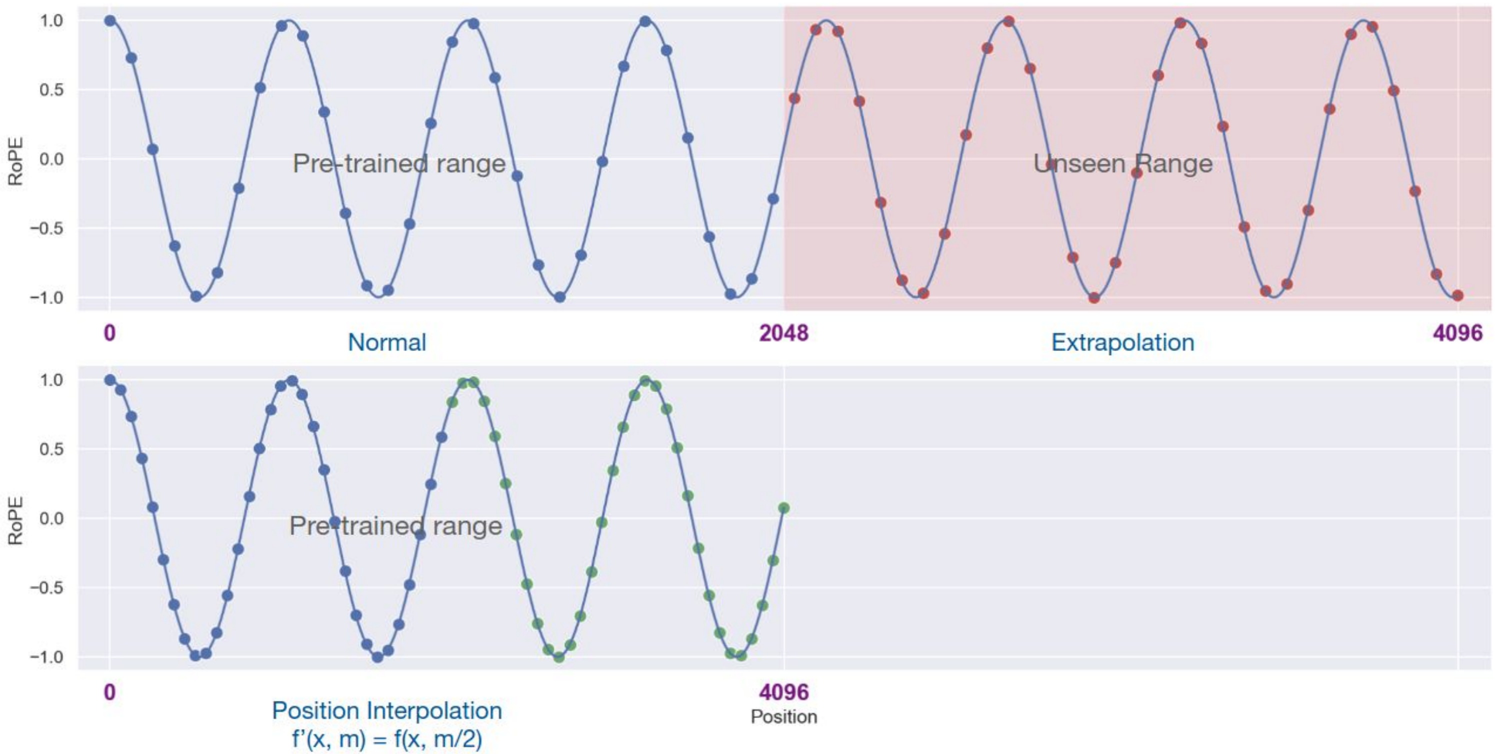


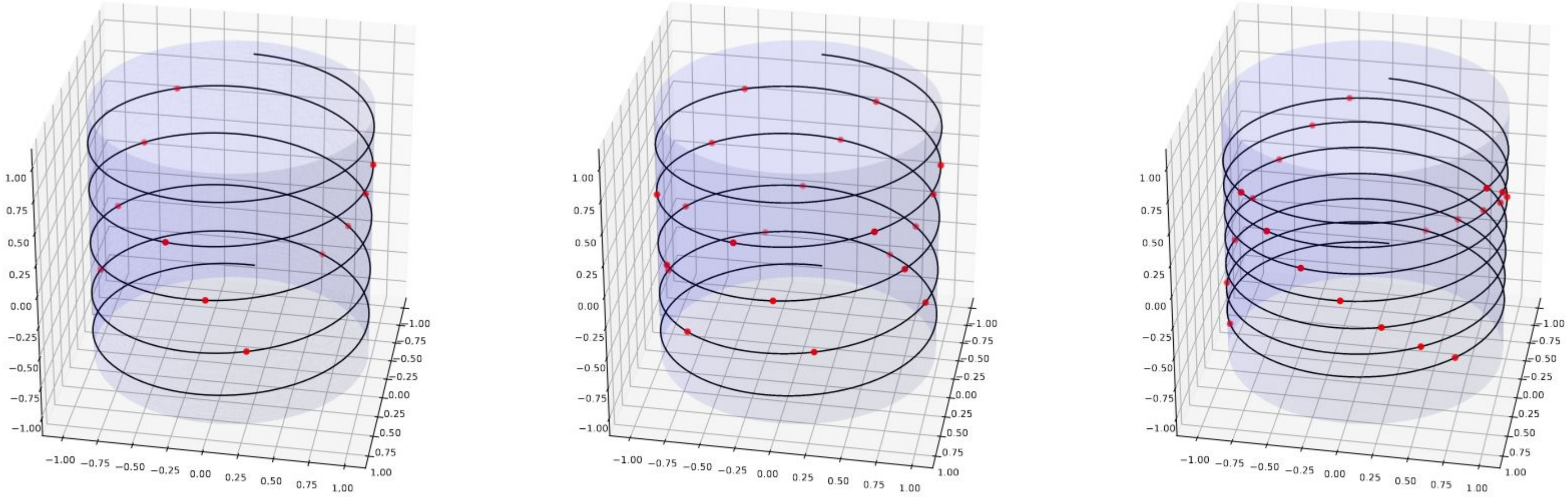
Figure 1: Implementation of Rotary Position Embedding(RoPE).

From RoFormer (Su et al 2021)

Positional interpolation (Chen et al 2023)

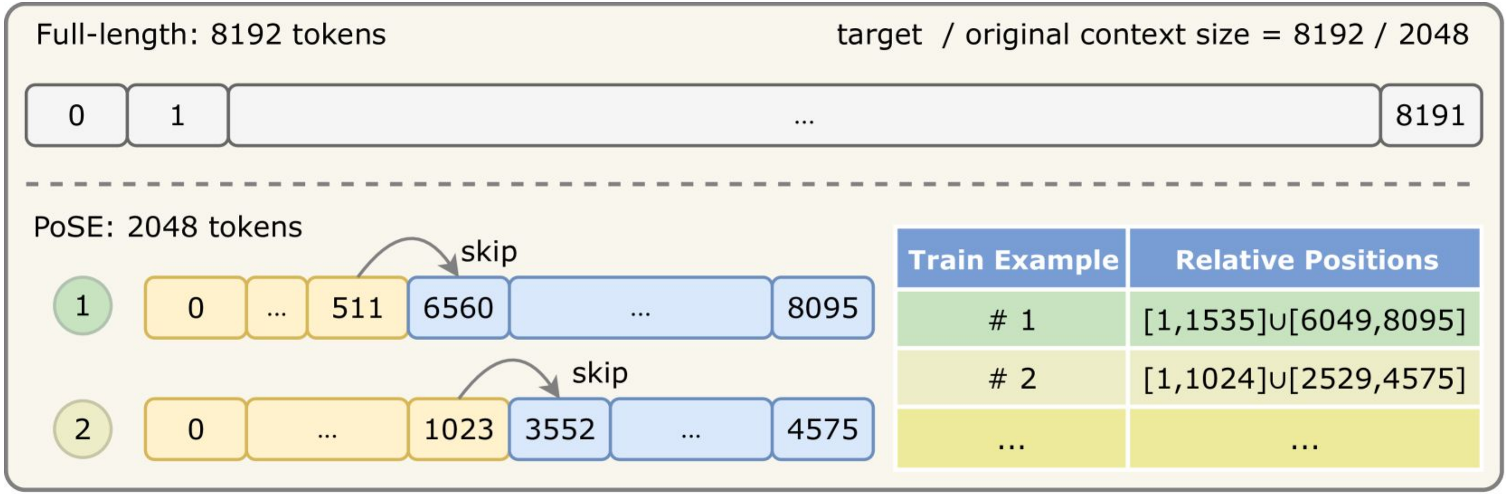


NTK-aware (ABF) scaling (Xiong, Liu et al 2023)



(a) RoPE (b) RoPE+PI (c) RoPE+ABF

POSE (Zhu et al 2024)

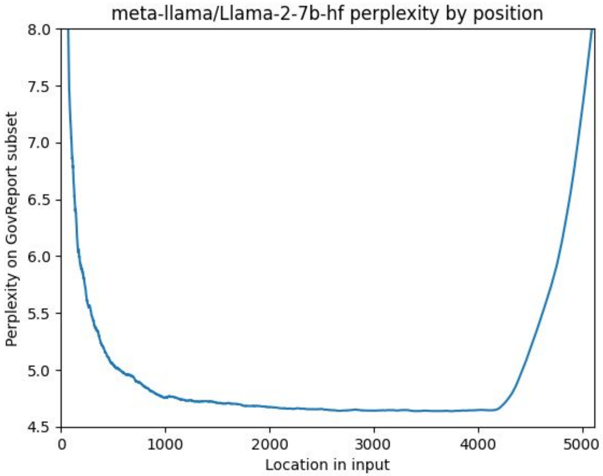


Generalization is still hard...

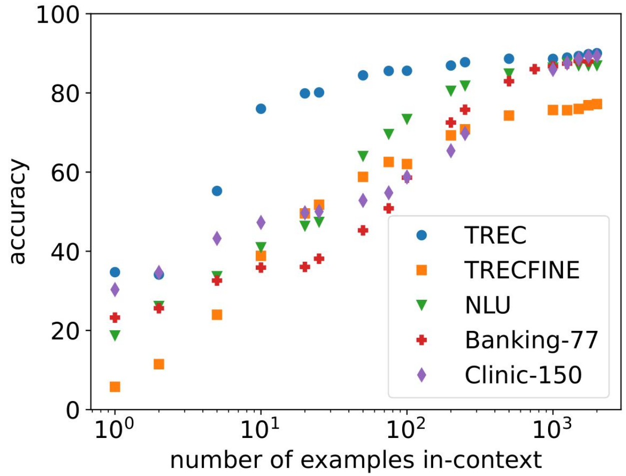
Data-side interventions	Not possible for every task, imposes additional assumptions
Model-side interventions	Not as effective as full finetuning, decreasingly effective with length
Position embedding modifications	Have to fully finetune, still struggles to extrapolate longer

Long context and generalization: two parts

Long context **as** (length) generalization



Long-context ICL



Perspective 2: Long-context ICL

In-Context Learning with Long-Context Models: An In-Depth Exploration

Amanda Bertsch[γ]
abertsch@cs.cmu.edu

Maor Ivgi[τ]
maor.ivgi@cs.tau.ac.il

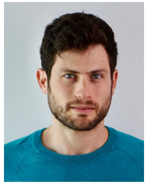
Uri Alon[γ*]
urialon@cs.cmu.edu

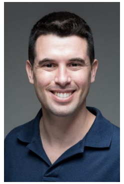
Jonathan Berant[τ]
joberant@cs.tau.ac.il

Matthew R. Gormley[γ]
mgormley@cs.cmu.edu

Graham Neubig[γ]
gneubig@cs.cmu.edu

Joint work with:


Maor Ivgi


Uri Alon


Jonathan Berant


Matt Gormley


Graham Neubig

Traditional ICL is sensitive

To example selection:

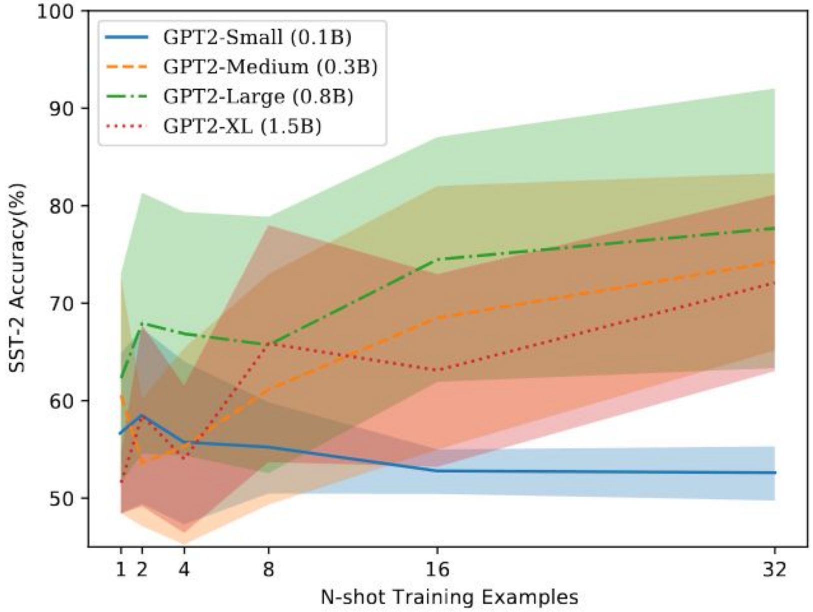
Method	NQ	WQ	TriviaQA*
RAG (Open-Domain)	44.5	45.5	68.0
T5+SSM (Closed-Book)	36.6	44.7	60.5
T5 (Closed-Book)	34.5	37.4	50.1
GPT-3 (64 examples)	29.9	41.5	-
Ours			
Random	28.6 ± 0.3	41.0 ± 0.5	59.2 ± 0.4
kNN$_{\text{roberta}}$	24.0	23.9	26.2
KATE$_{\text{roberta}}$	40.0	47.7	57.5
KATE$_{\text{nli}}$	40.8	**50.6**	60.9
KATE$_{\text{nli+sts-b}}$	**41.6**	50.2	**62.4**

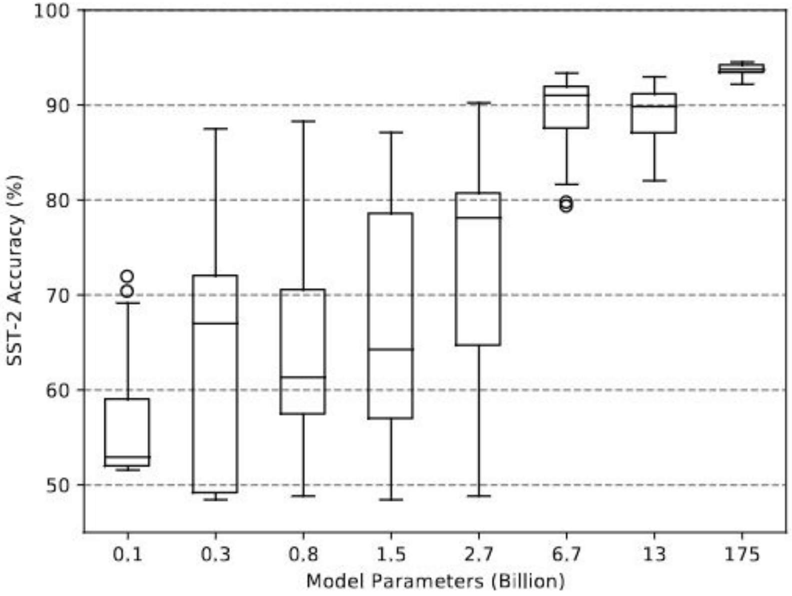
Table 7: QA results on various datasets. (*) On TriviaQA, we used 10 examples. On NQ and WQ, we used 64 examples.

from What Makes Good In-Context Examples for GPT-3? (Liu et al 2021)

Traditional ICL is sensitive

To example order:

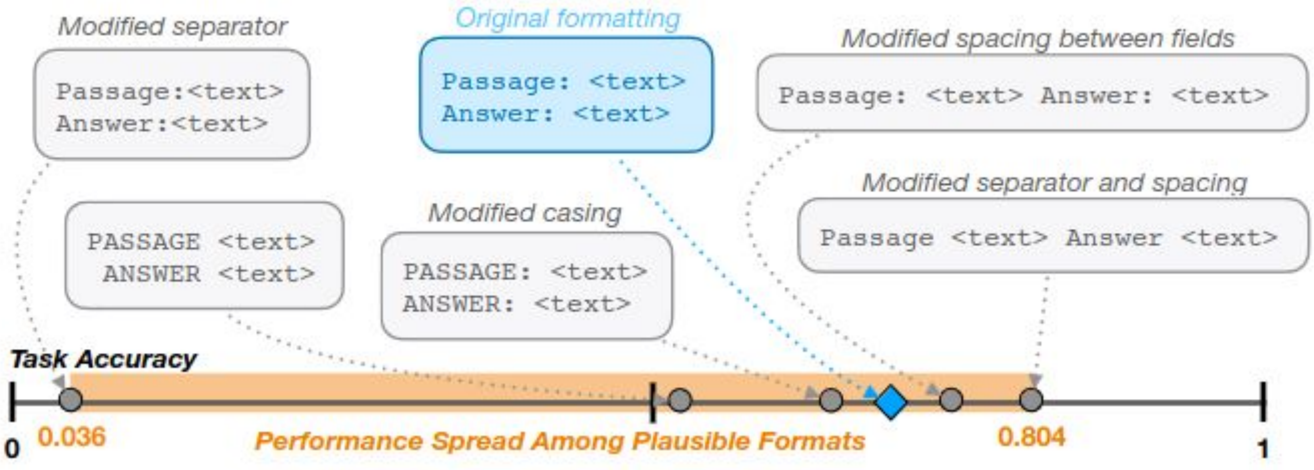




from Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity

Traditional ICL is sensitive

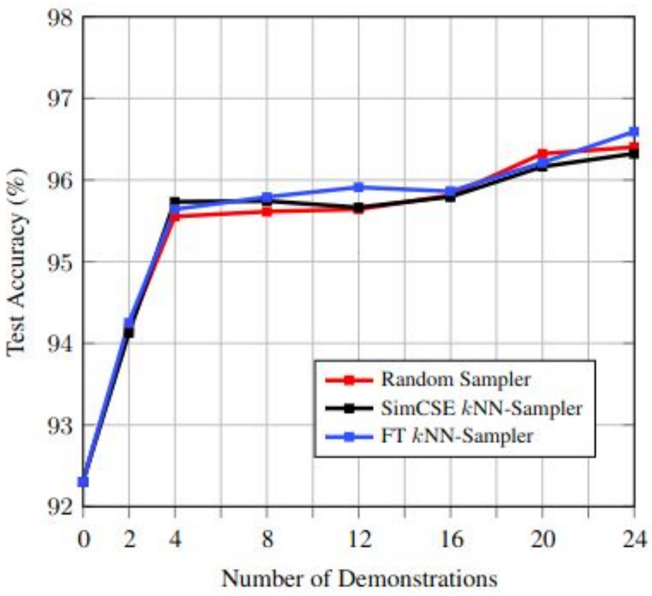
To instruction format:



from QUANTIFYING LANGUAGE MODELS' SENSITIVITY TO SPURIOUS FEATURES IN PROMPT DESIGN or: How I learned to start worrying about prompt formatting

Traditional ICL: more demonstrations is better?

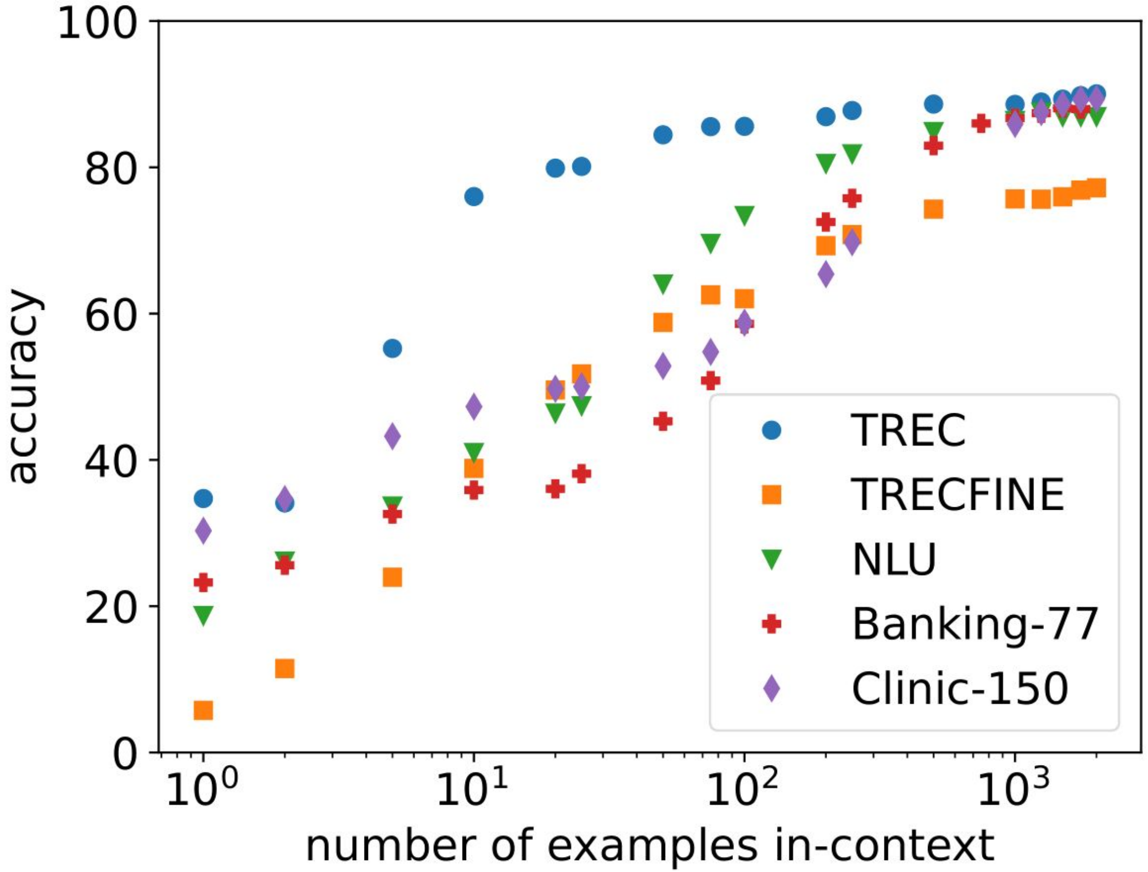
Well…. *sometimes*



SST-2: 2-label sentiment classification

from Text Classification via Large Language Models

Adding more demonstrations continues to increase performance!



Long-context ICL differs from short-context ICL in many ways!

> Comparison points: performance and efficiency

> Properties of long-context ICL

> Why does long-context ICL work?

Comparison: given a big enough dataset, how could we approach the task?

> retrieval ICL

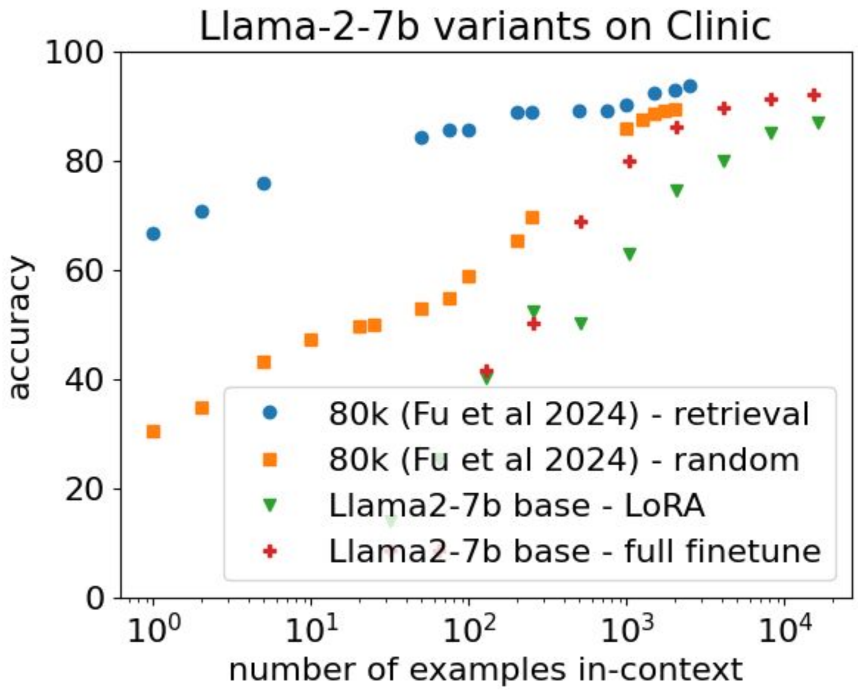
BM25 retriever; if we get <n results, we'll sample randomly to fill in the rest

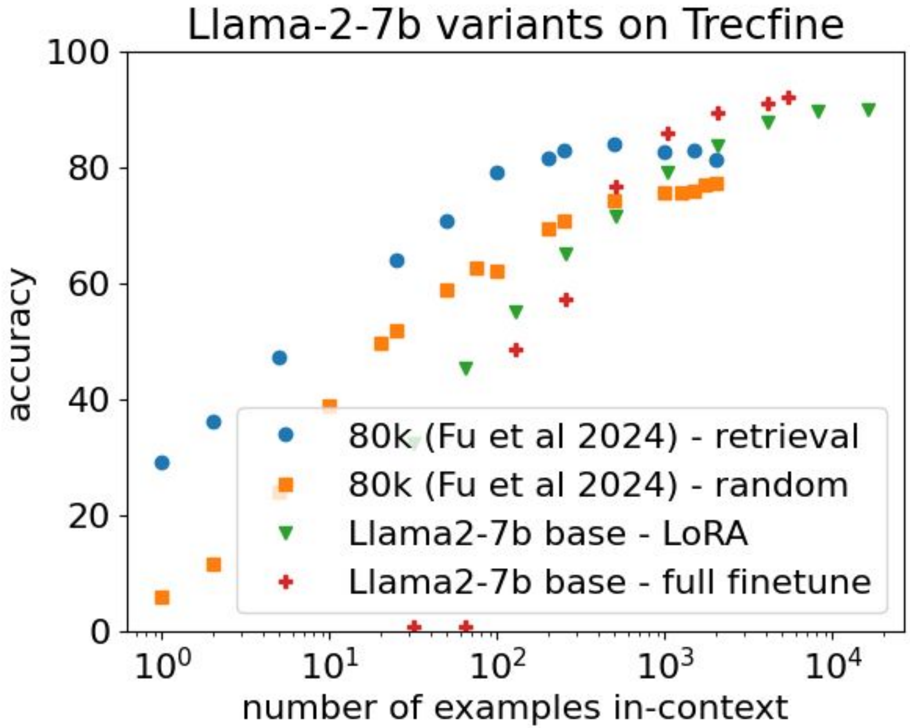
> LoRA finetuning

> full finetuning

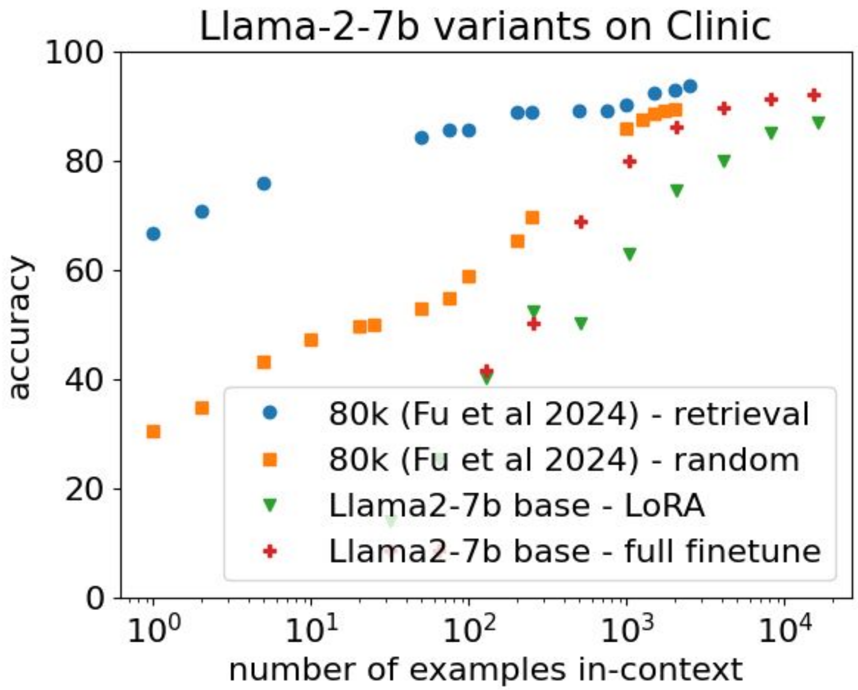
Classification head initialized with representation of each label's first token

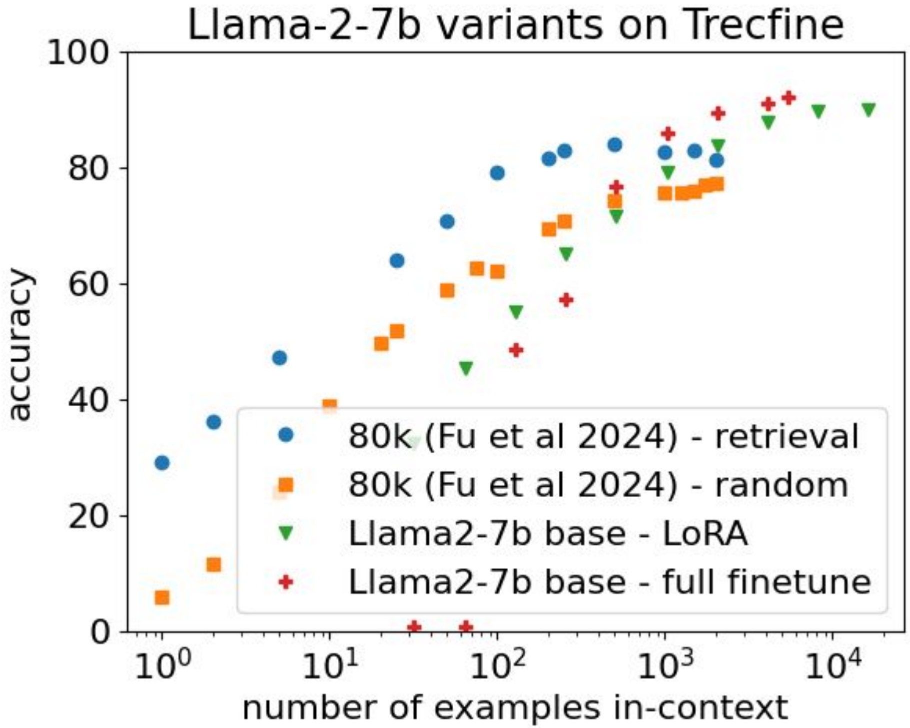
Comparison: results



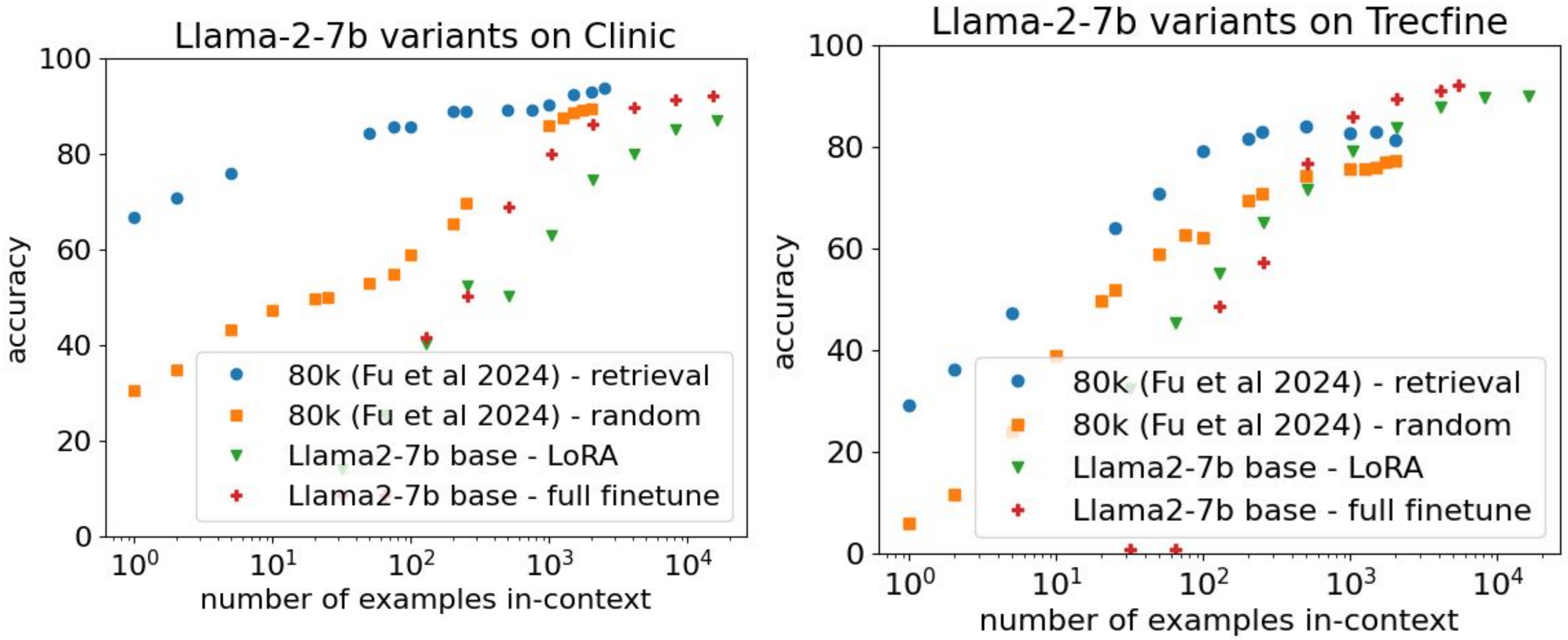


Long-context ICL benefits less from retrieval





Long-context ICL is often competitive with (or better than!) LoRA and full finetuning at the same dataset size



Properties: does long-context ICL exhibit the same sensitivities as short-context ICL?

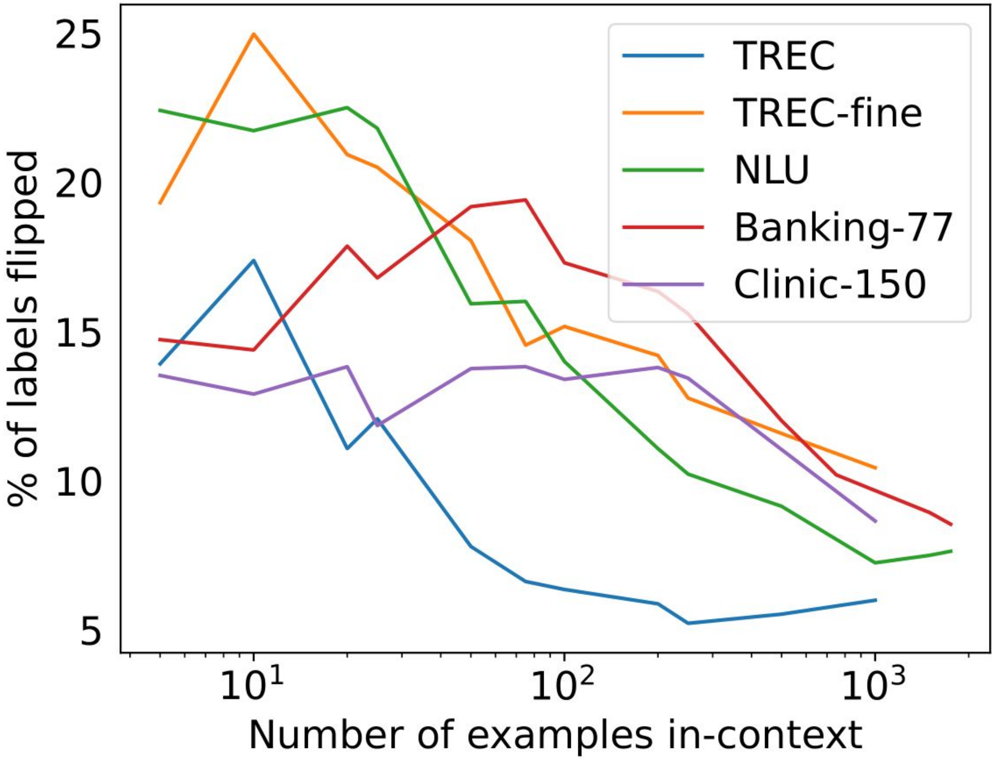
Traditional ICL shows some undesirable sensitivities

We’ve already seen a decreased sensitivity to data selection strategy…

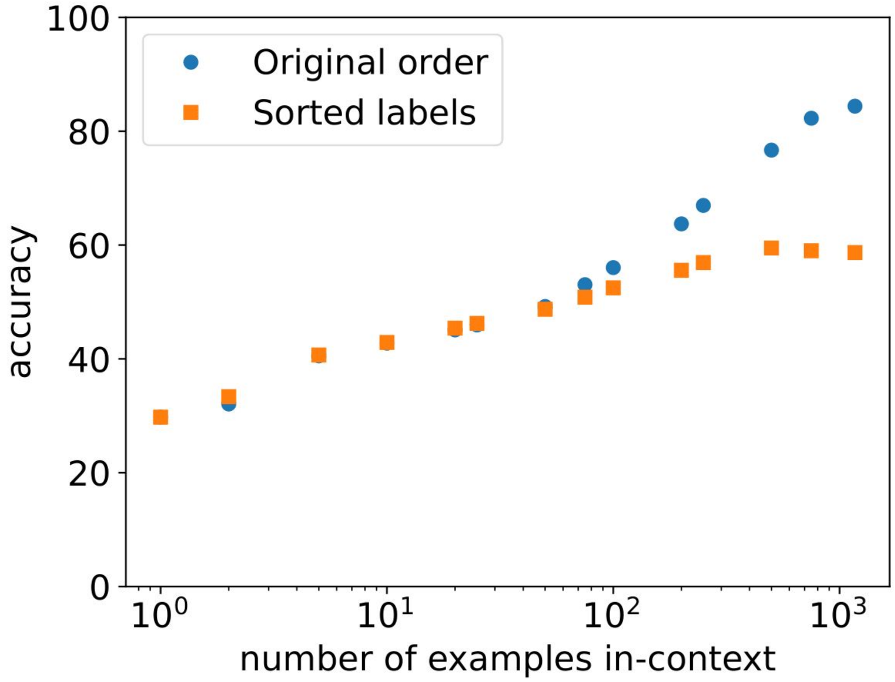
Long-context ICL is less sensitive to randomized example orders…

How do we measure this?

- Given a set of examples, shuffle 3 times
- Measure the % of predictions that changed when data was shuffled
- Average this over the 3 runs



...but more sensitive to sorting demonstrations by label



Clinic 150, Llama-32k

Why?

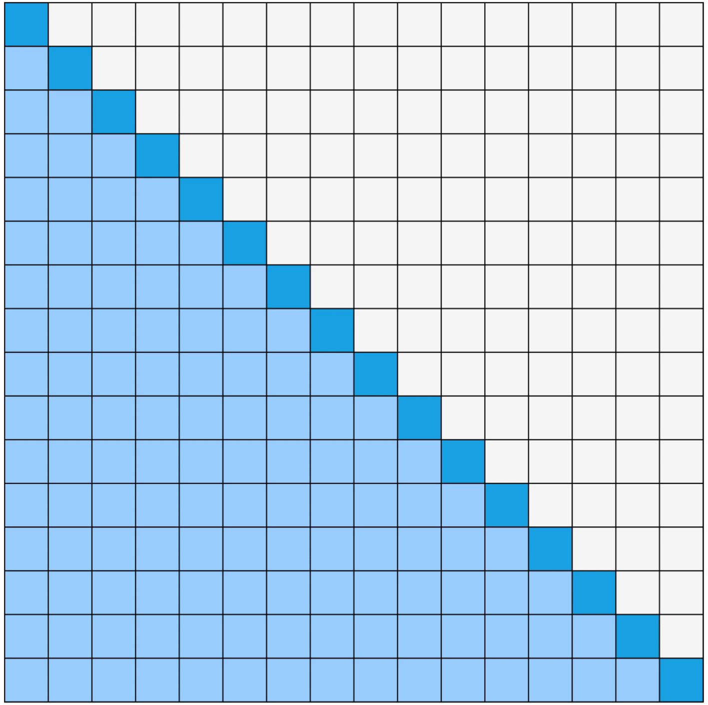
- Local context of all the same label is harmful to performance

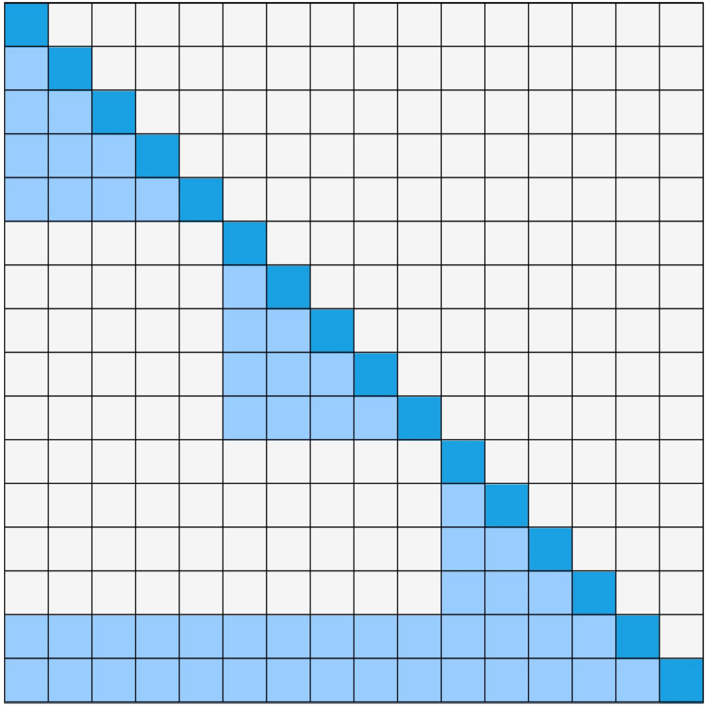
What makes long-context ICL work?

Is it:

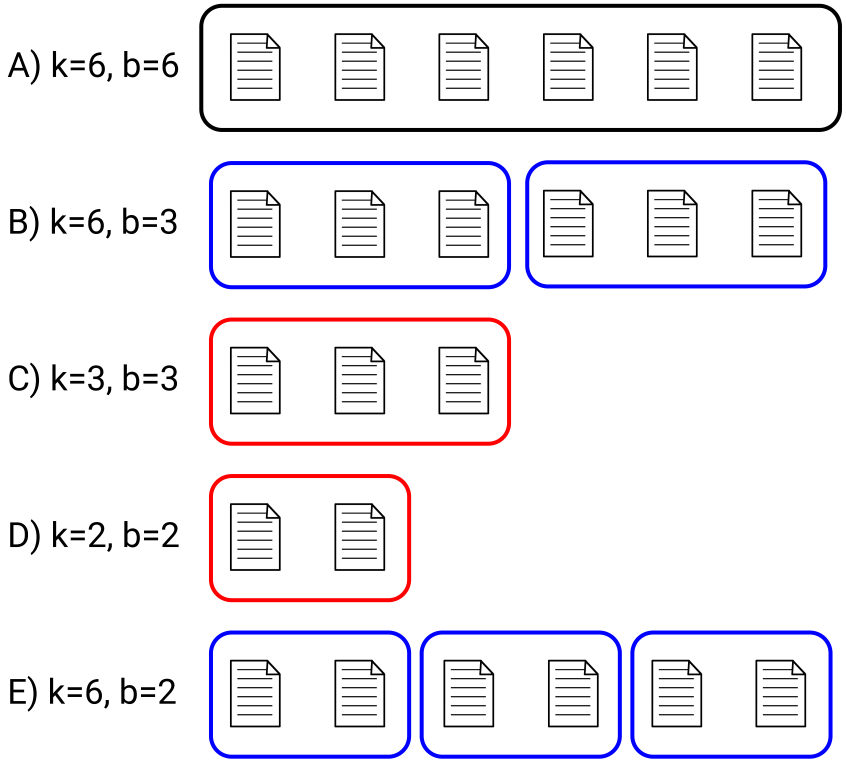
- The much larger number of examples?
- The much better contextualization of examples?
- Something else?

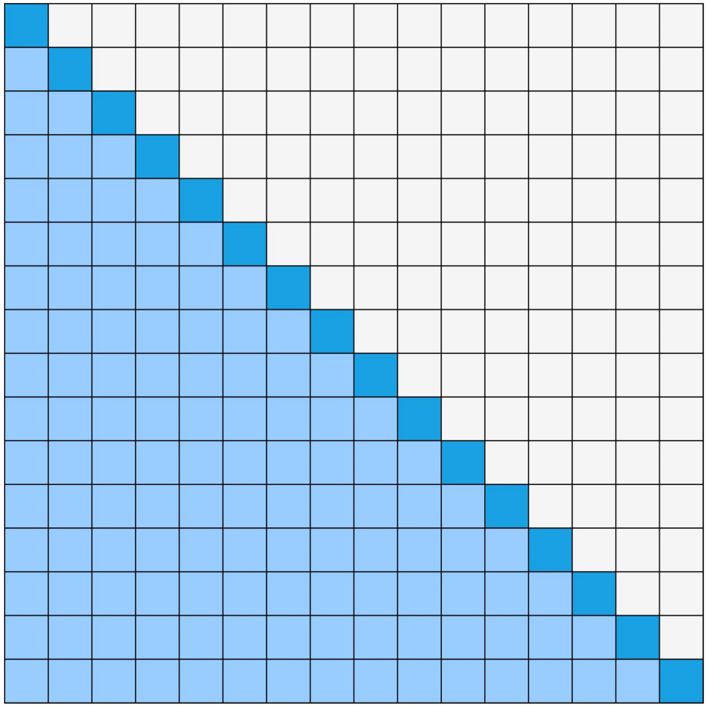
Does long-context ICL need long-context attention?





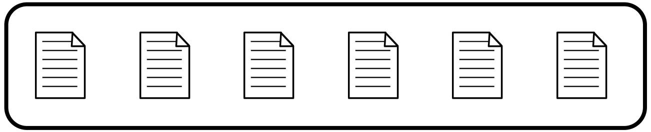
Block attention





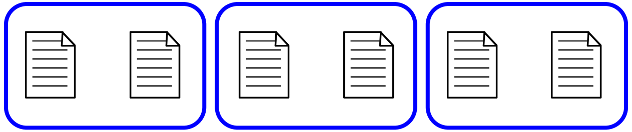
Block attention

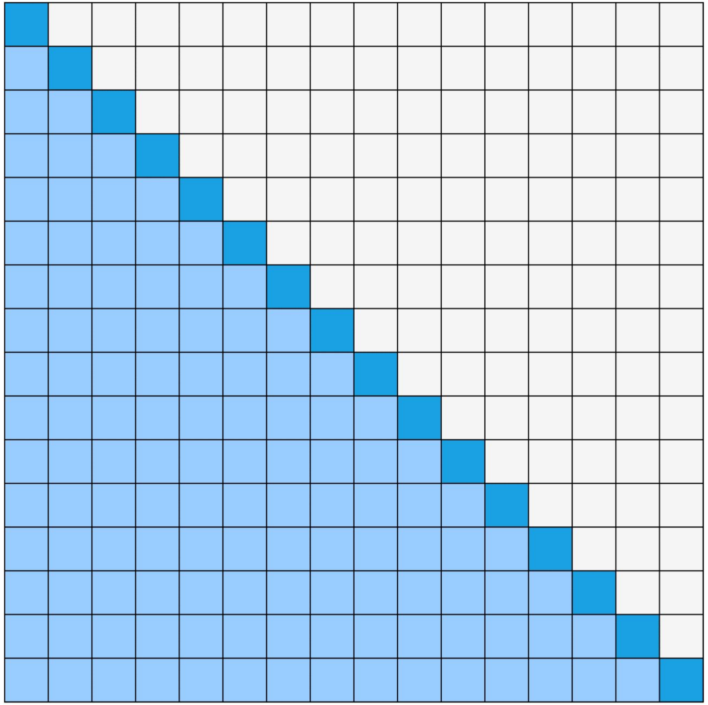
A) k=6, b=6

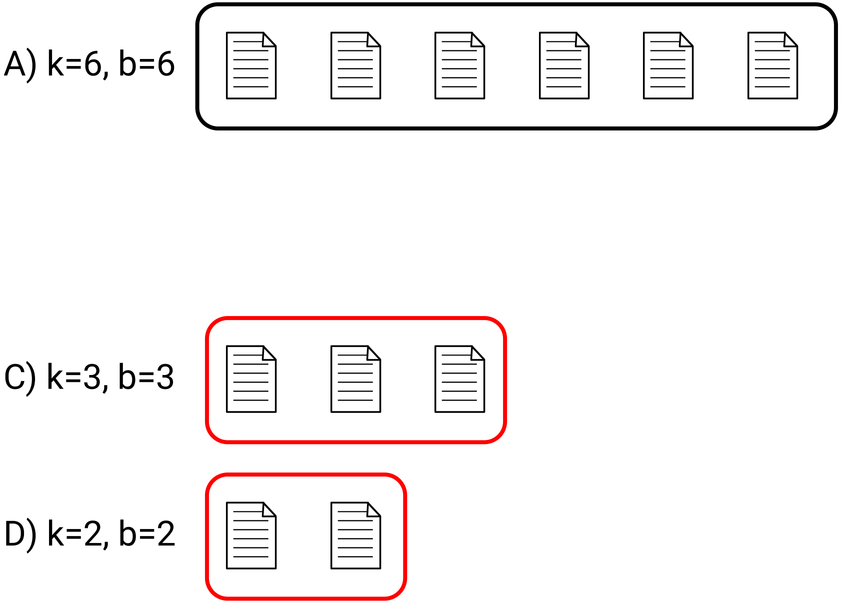


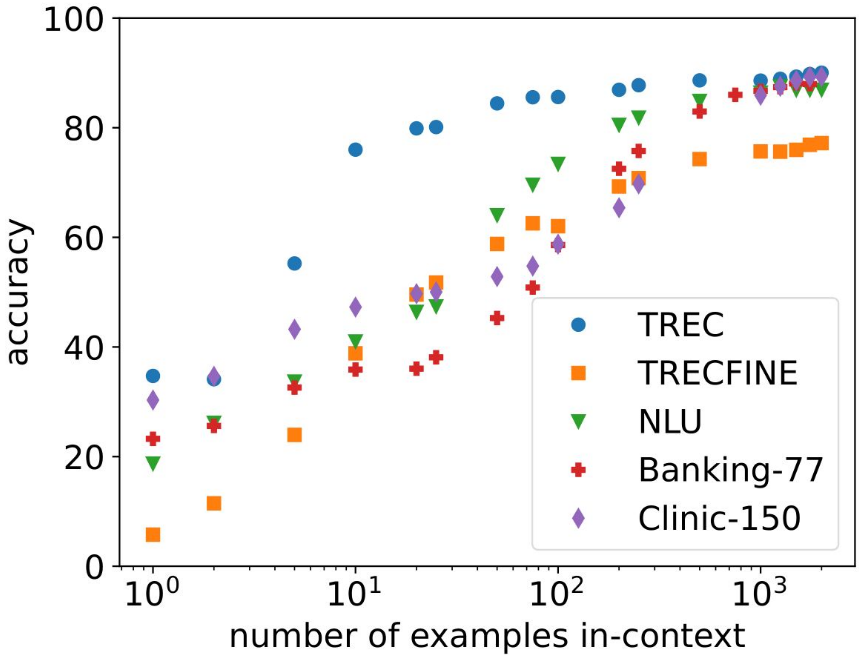
C) k=3, b=3

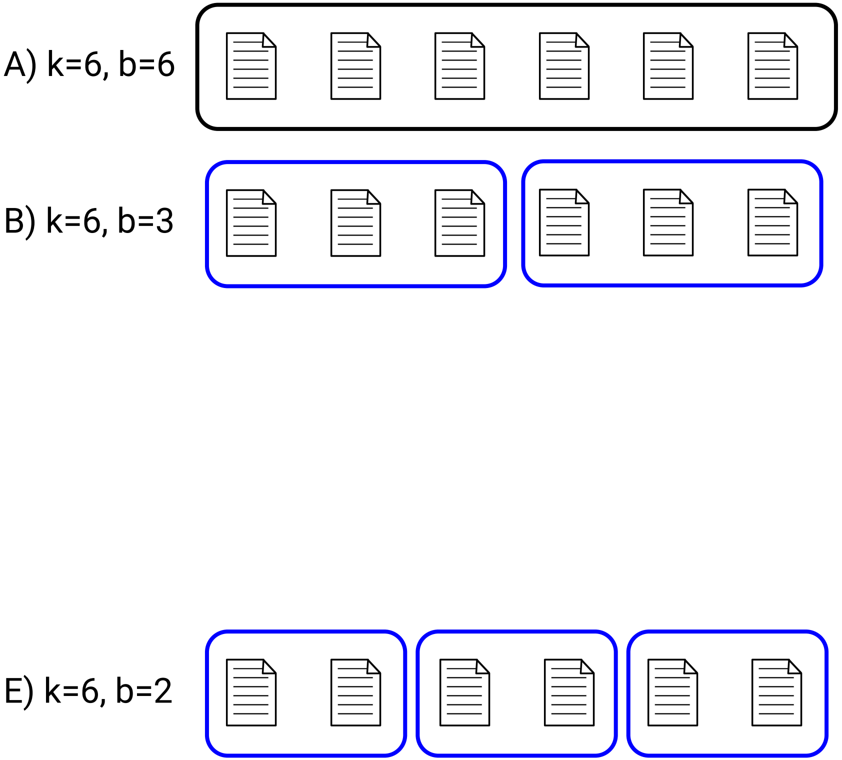
E) k=6, b=2

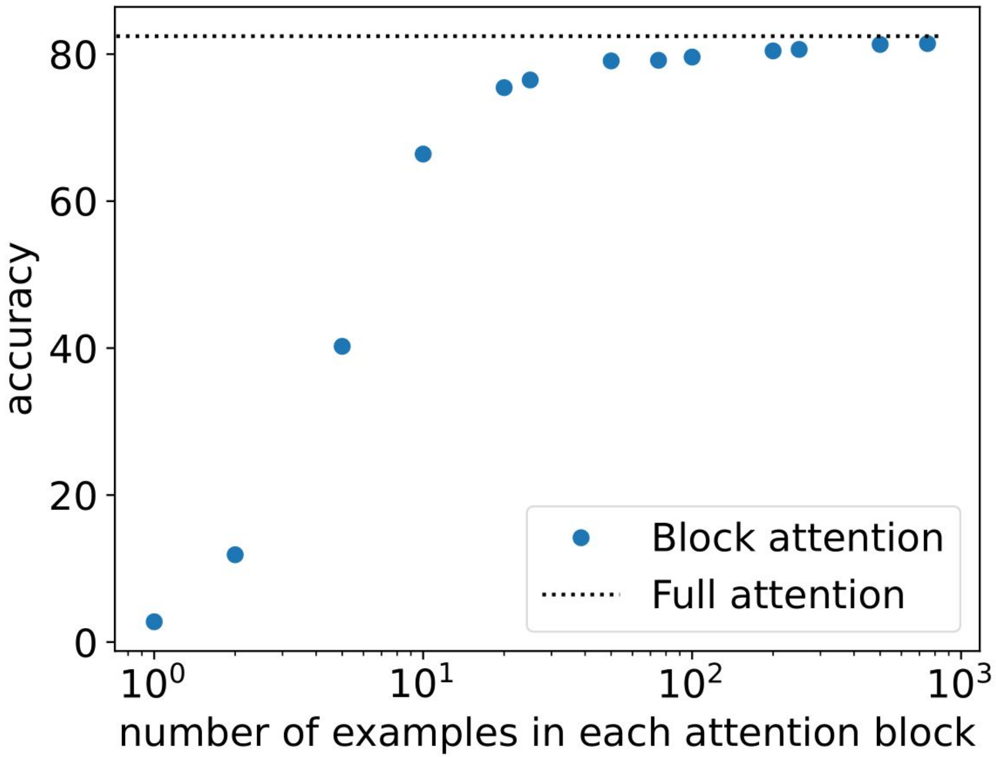





A) k=6, b=6
C) k=3, b=3
D) k=2, b=2

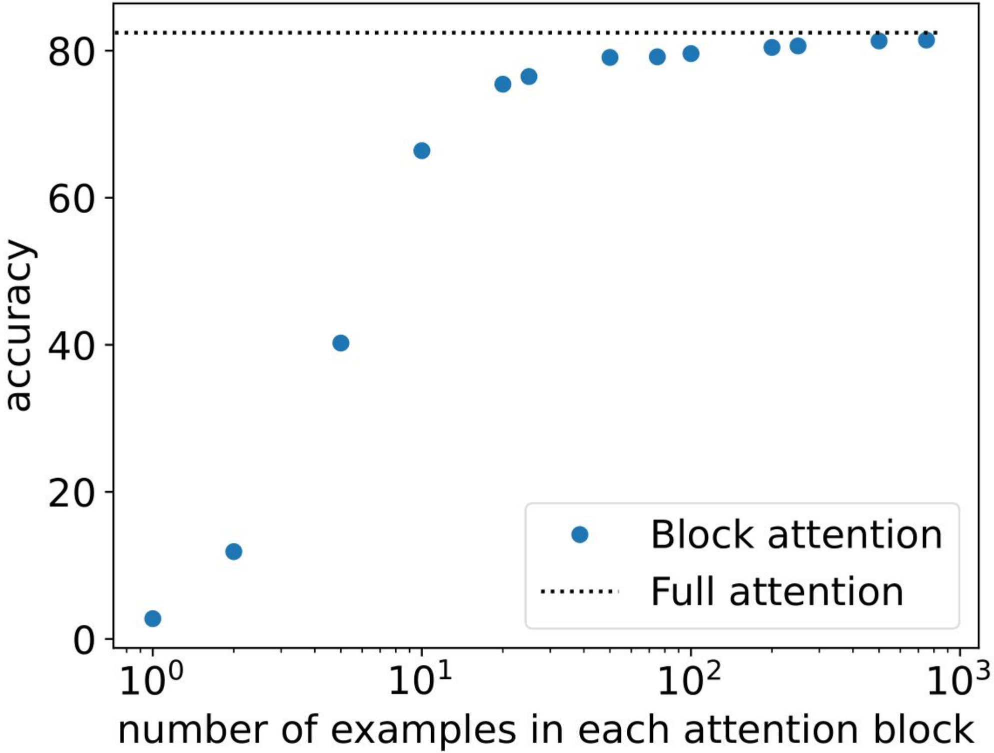

accuracy
number of examples in-context
TREC
TRECFINE
NLU
Banking-77
Clinic-150


A) k=6, b=6
B) k=6, b=3
E) k=6, b=2

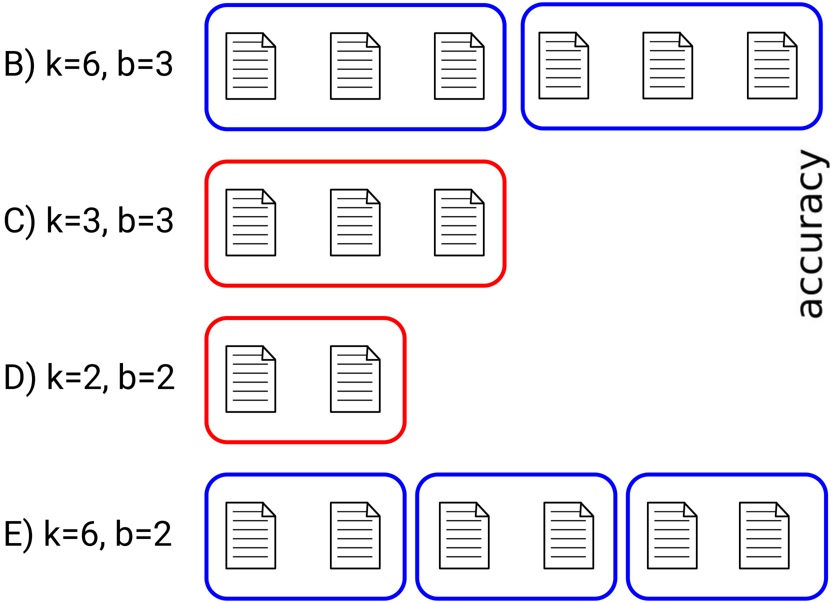

accuracy
80
60
40
20
0
10^0
10^1
10^2
10^3
number of examples in each attention block
Block attention
Full attention

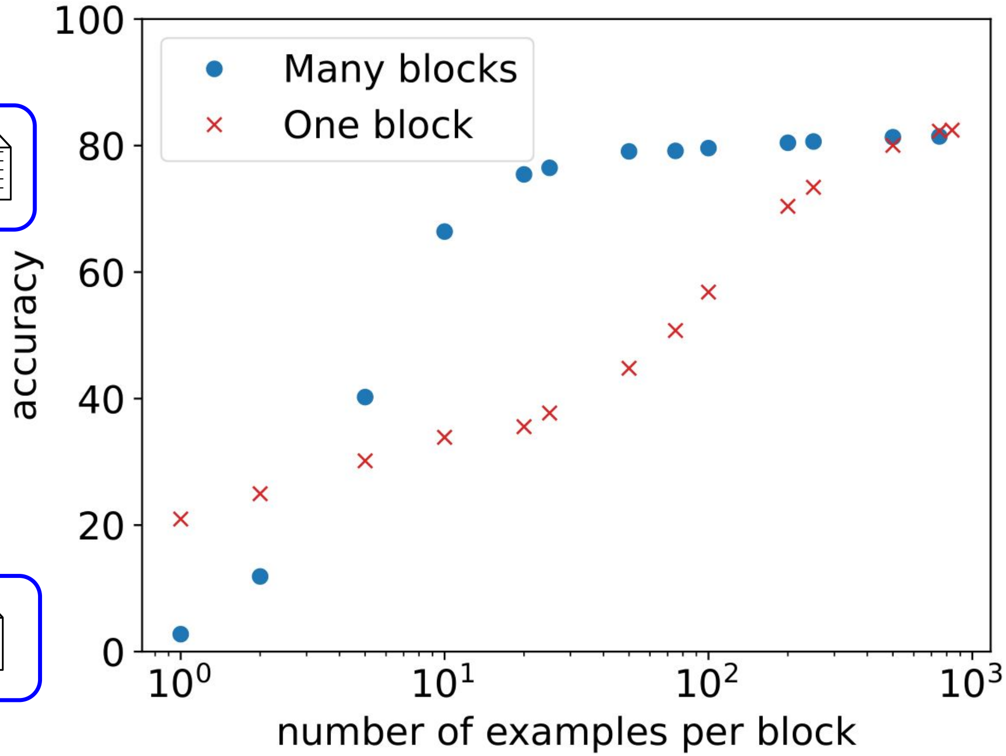
Block attention quickly nears full attention performance

- Block sizes of b=50-100 recover nearly full attention performance at k=1000
- Why a little less?
  - Remember the start of each block lacks good contexualization



Block attention with one vs many blocks





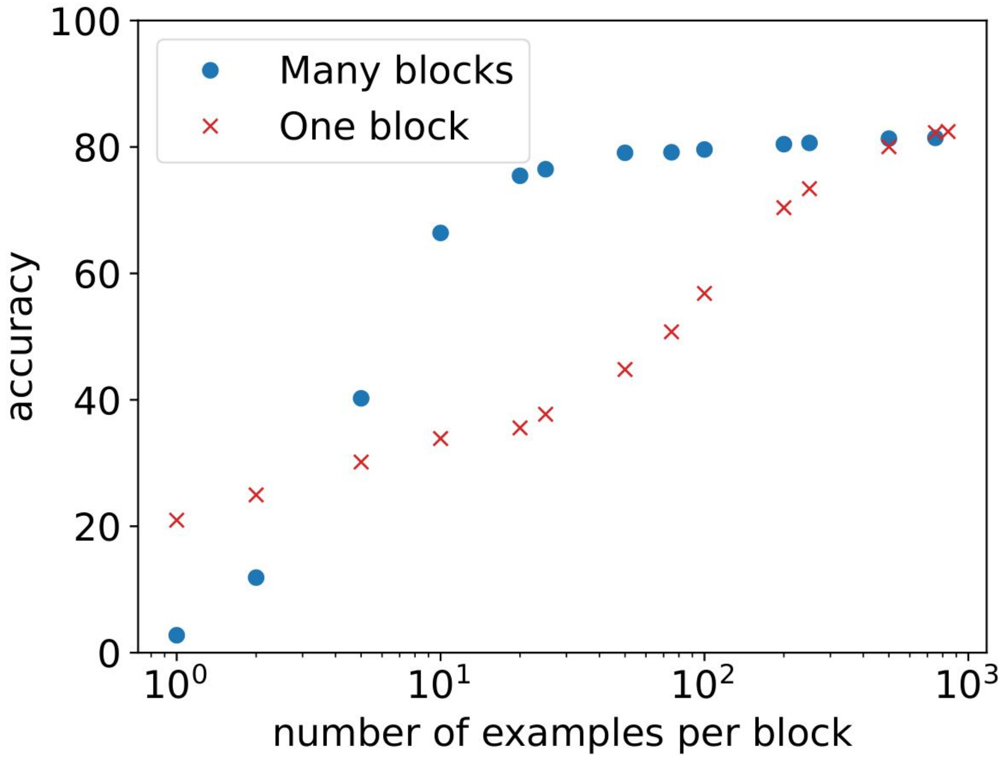
Block attention with one vs many blocks

In short contexts:

- One block outperforms many

In longer contexts:

- Many blocks outperform one



What does this mean?

Long-context ICL is:

- ✅ less sensitive to demonstration selection and ordering
- ✅ able to take advantage of cached demonstration encodings
- ✅ strongly competitive with finetuning
- ✅ effective even with only local attention for demonstration set
- ❌ a panacea
- ❌ always the best compute-performance tradeoff

What does it all mean?

Long context modeling

- Can often be framed as a generalization problem
- Allows models to do interesting (different?) things in-context

Thank you! questions?

Contact me: abertsch@cs.cmu.edu @abertsch72