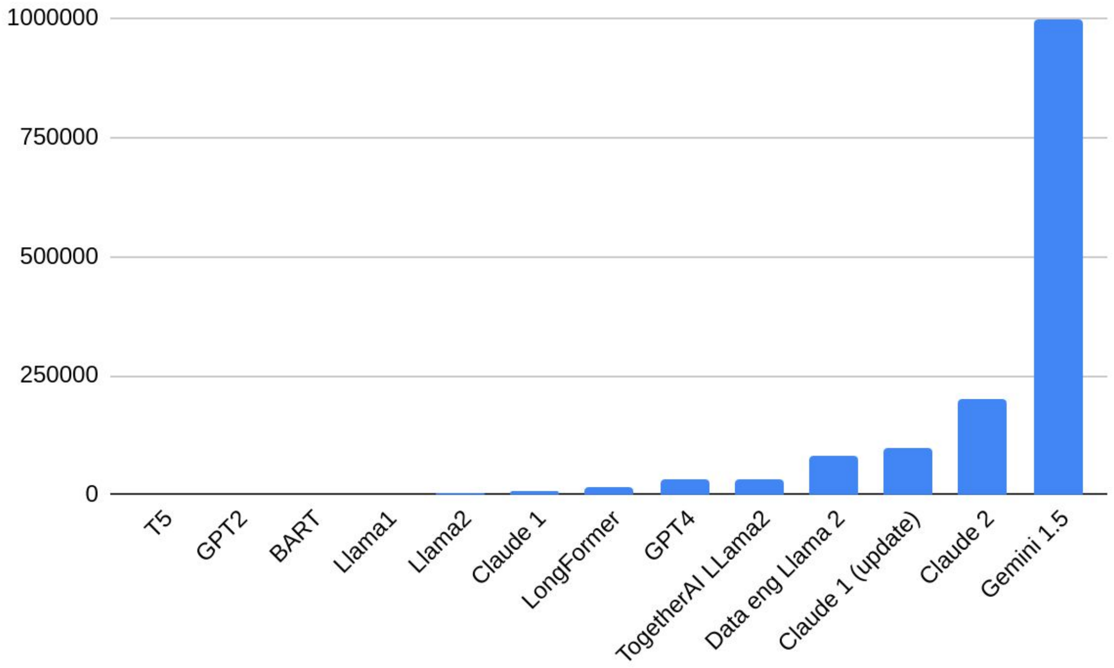
In-Context Learning with Long-Context Models: An In-Depth Exploration

Amanda Bertsch Maor Ivgi Uri Alon
Jonathan Berant Matt Gormley Graham Neubig

Models with *very* long context length abound



What do we do with 10k-1000k context?

- Fit books in the context window
- Fit a language grammar for translation
- Fit a training dataset?

Traditional ICL requires selecting a small subset of data

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

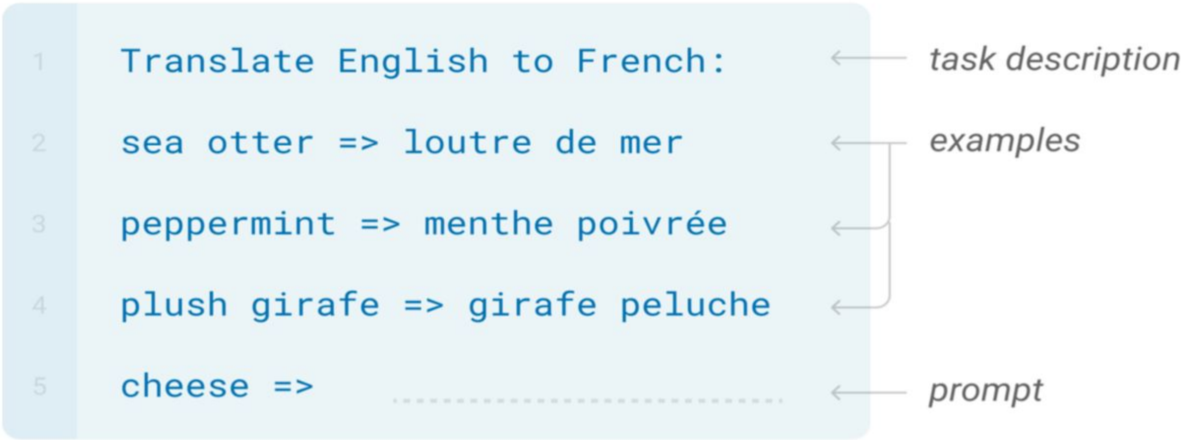


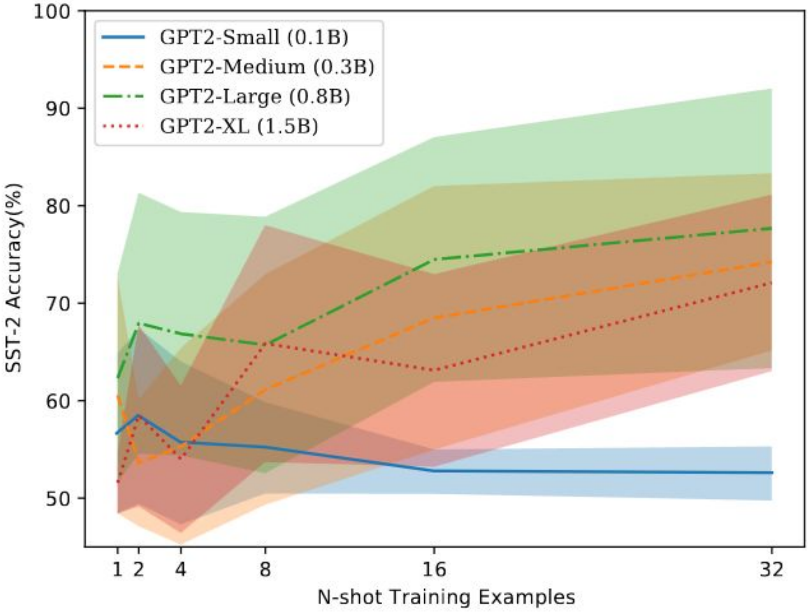
Figure credit: https://thegradient.pub/in-context-learning-in-context/

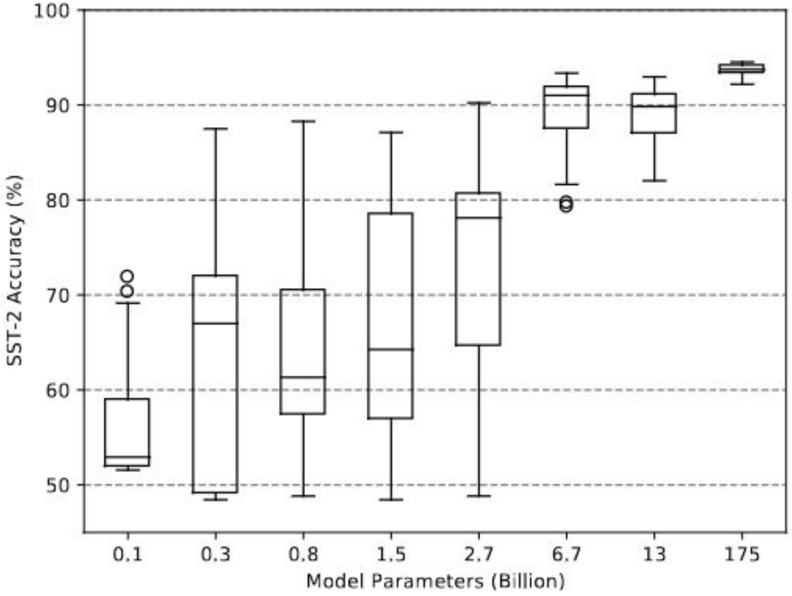
We’re approaching the scale where full datasets could be used as demonstrations…

…what does ICL look like at these extremes?

Traditional ICL is also very sensitive

To example order:

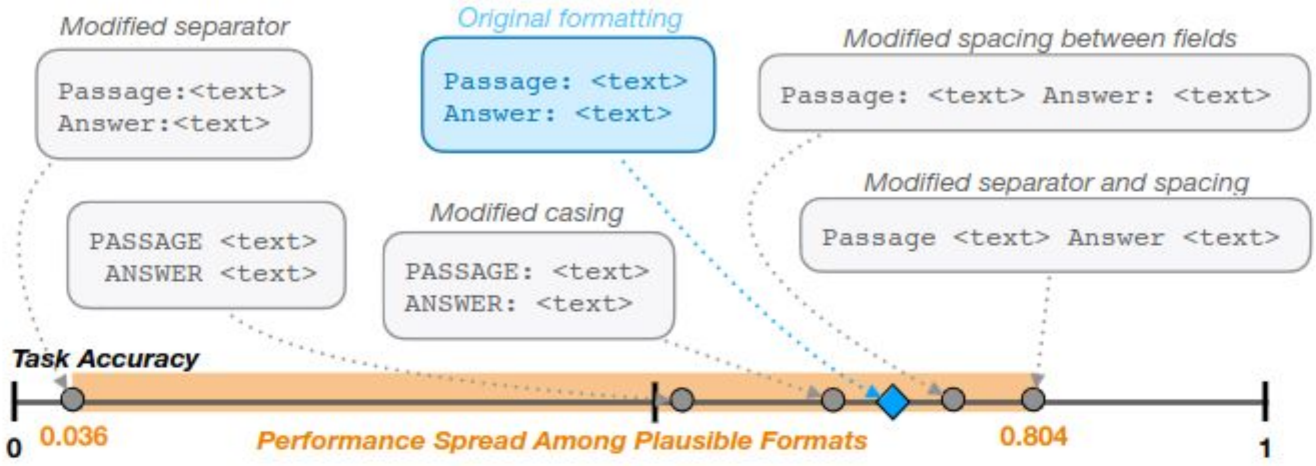




from Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity

Traditional ICL is also very sensitive

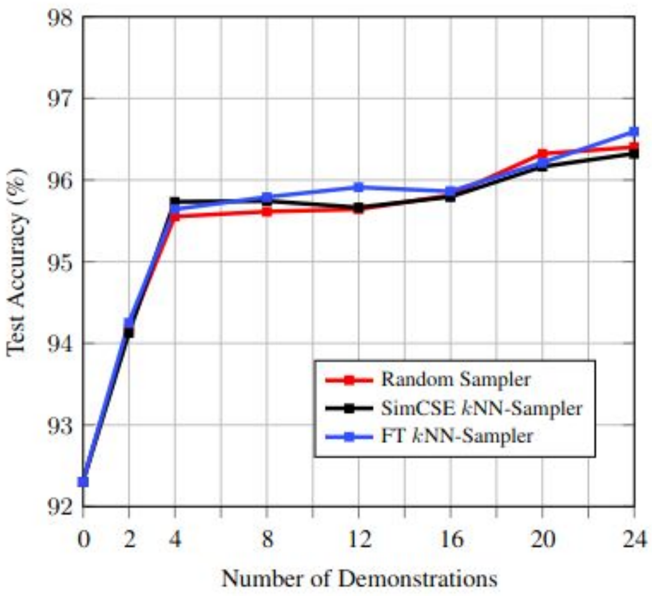
To instruction format:



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

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

> Preliminaries

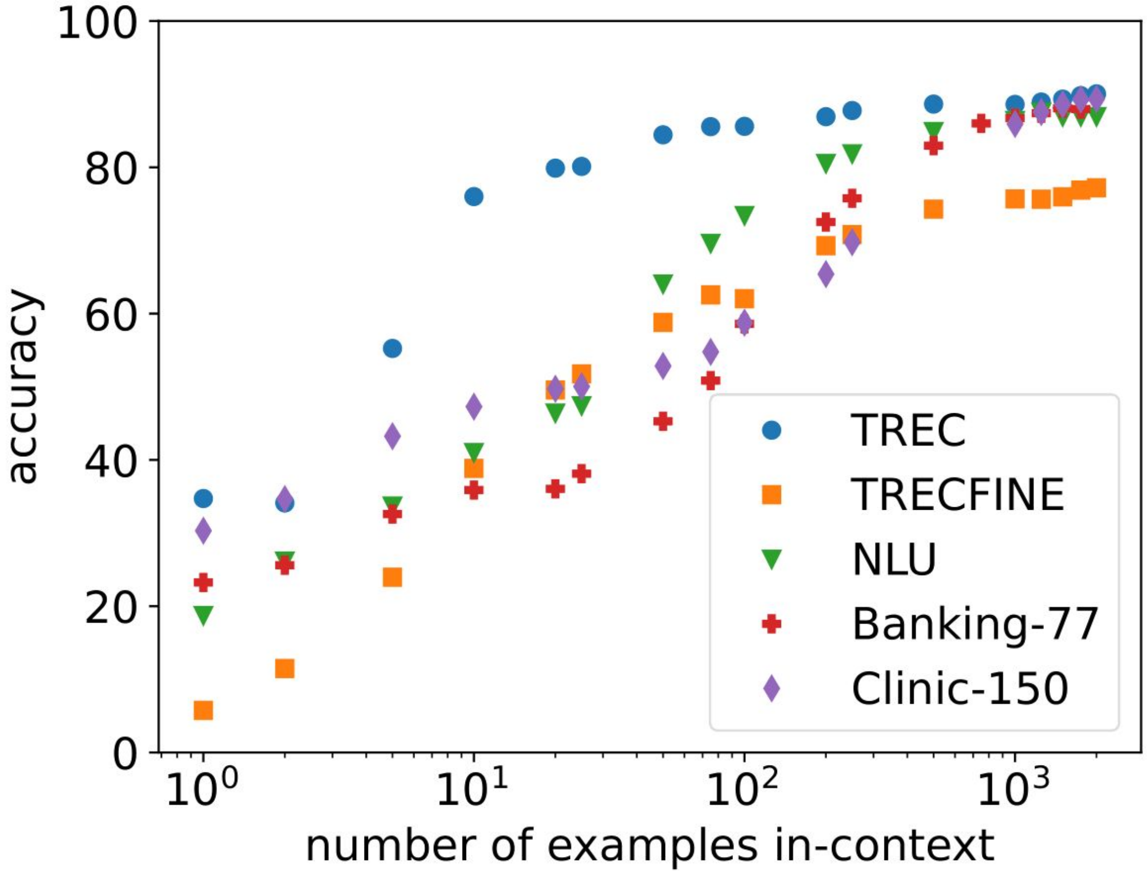
> Comparison points: performance and efficiency

> Properties of long-context ICL

> Why does long-context ICL work?

> Using ICL to benchmark long-context models

Adding more demonstrations continues to increase performance!



Preliminaries: models and data

Modeling

Llama2-7b family

- Original model: 4096 context
- TogetherAI model: 32k context
- Fu et al 2024: 80k context

Similar trends on Mistral v0.2 (32k context)

Data

TREC: 6-way question classification

TREC-fine: 50-way question classification

NLU: 68-way intent classification for conversational assistant commands

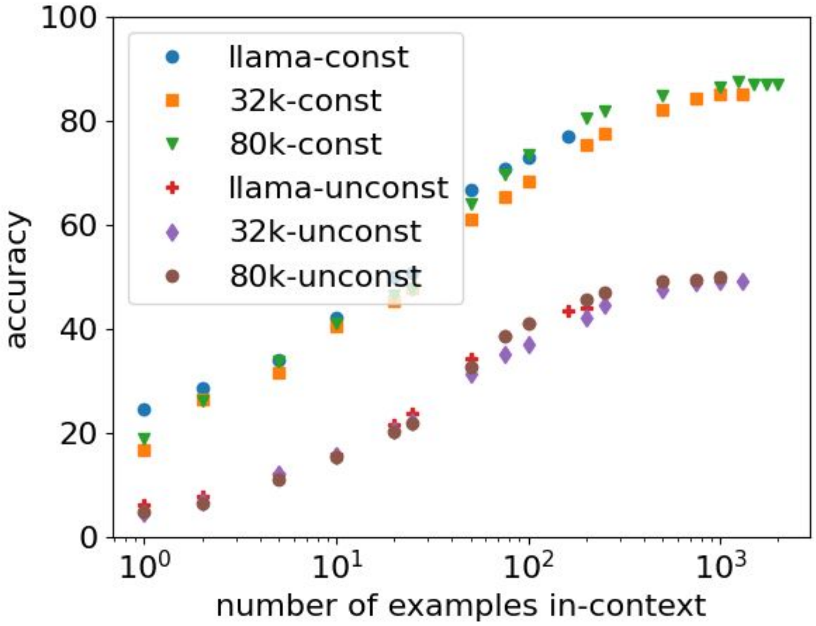
Banking77: 77-way intent classification for financial domain

Clinic150: 151-way intent classification, cross-domain

Preliminaries: ICL settings

Evaluation

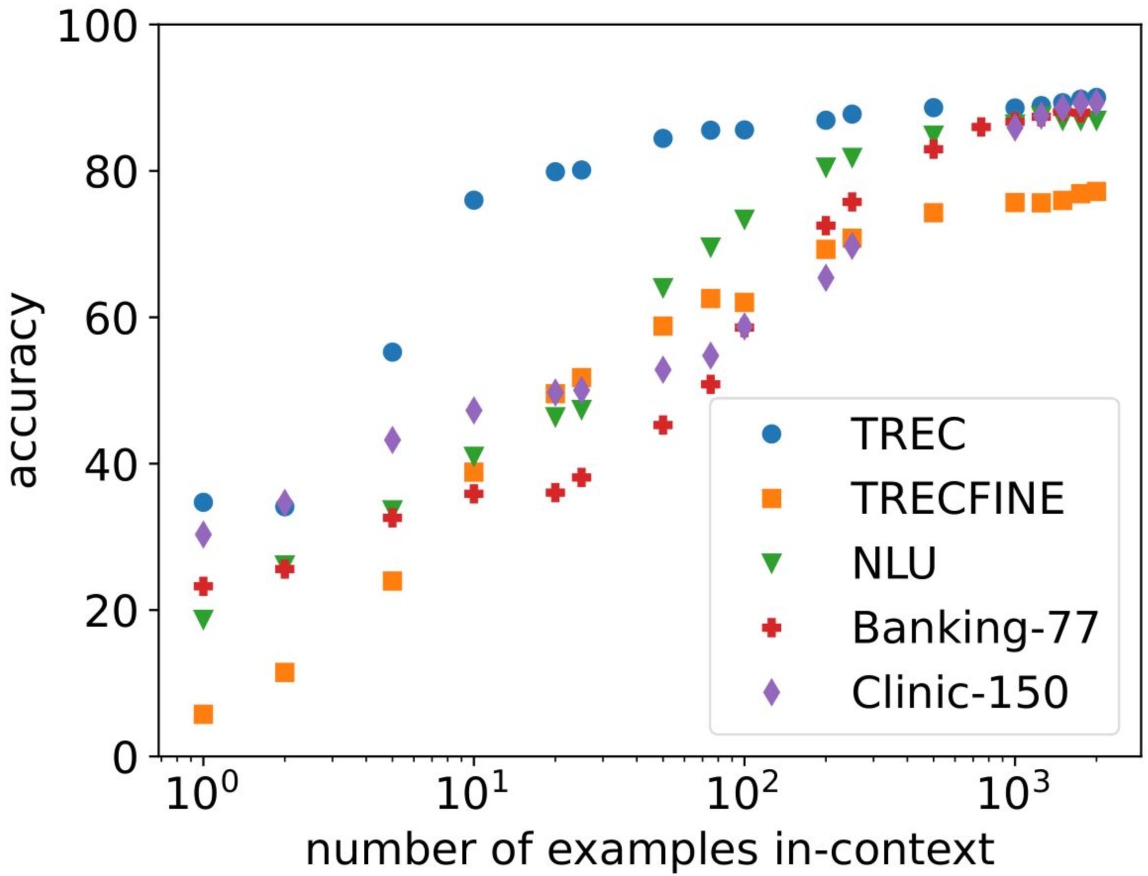
- Constrained decoding
- Average over 10 seeds
- Similar trends with f1



Model: Llama2-80k

Data:

- TREC: 6-way question classification
- TREC-fine: 50-way question classification
- NLU: 68-way intent classification for conversational assistant commands
- Banking77: 77-way intent classification for financial domain
- Clinic150: 151-way intent classification, cross-domain



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

> retrieval ICL

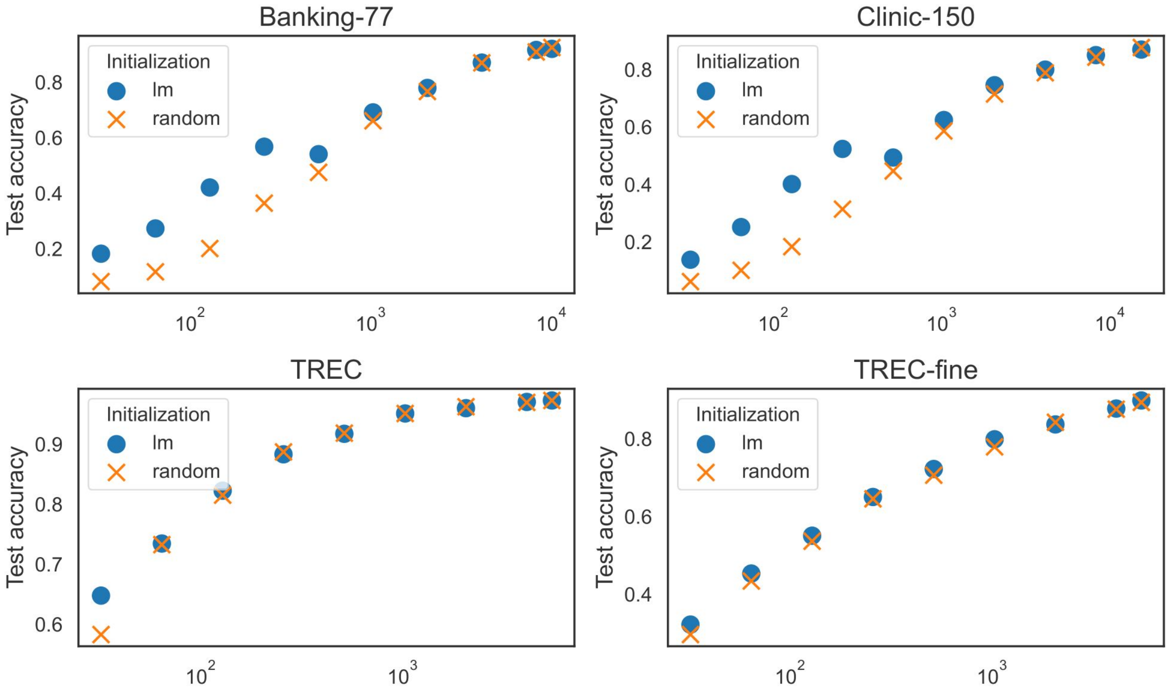
BM25 retriever; if we get <n results, we'll sample randomly to fill in the rest

> LoRA finetuning

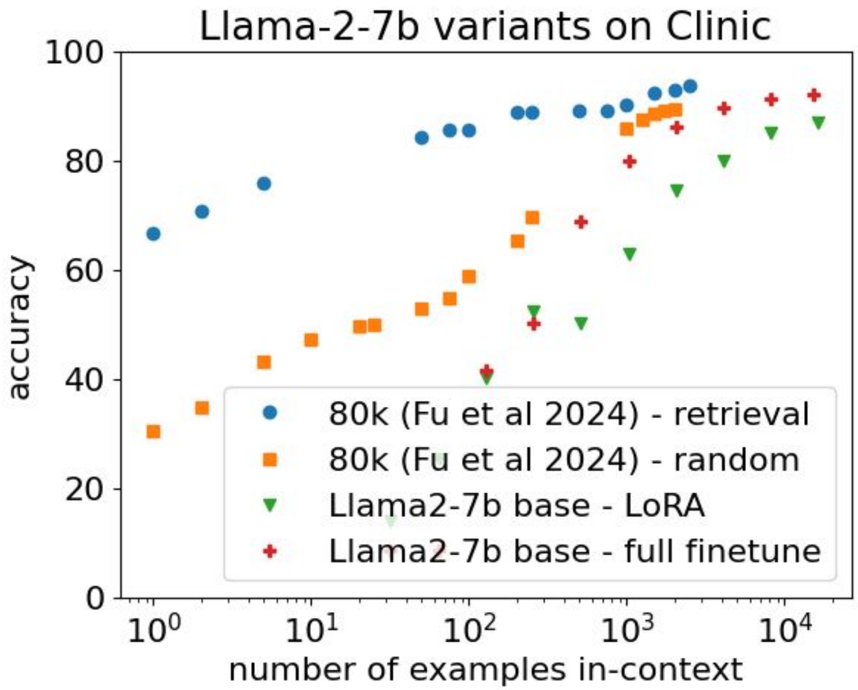
> full finetuning

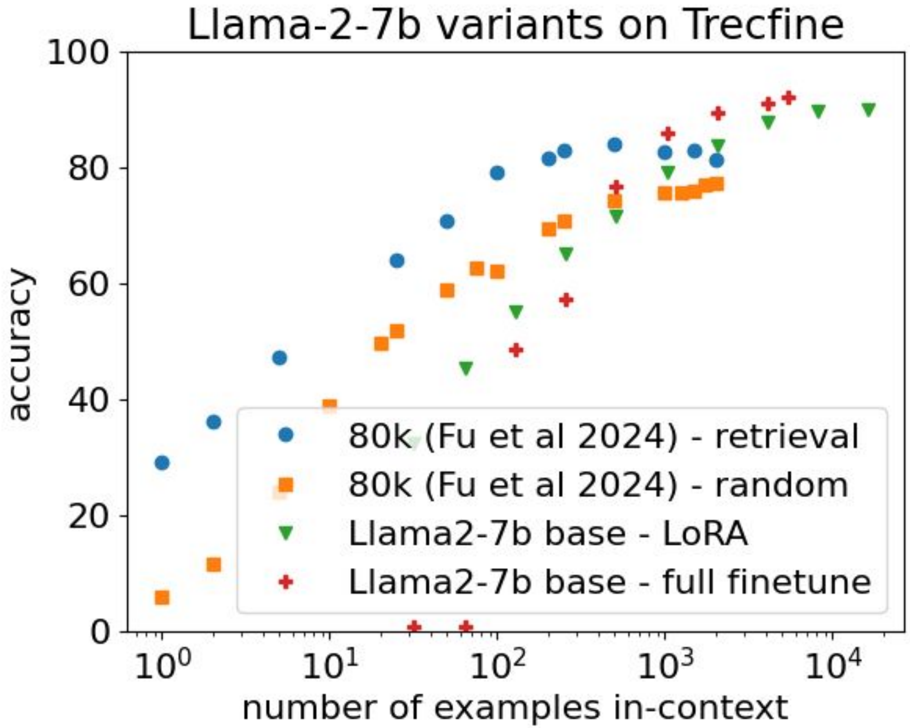
Classification head initialized with representation of each label's first token

Finetuning: classification head init

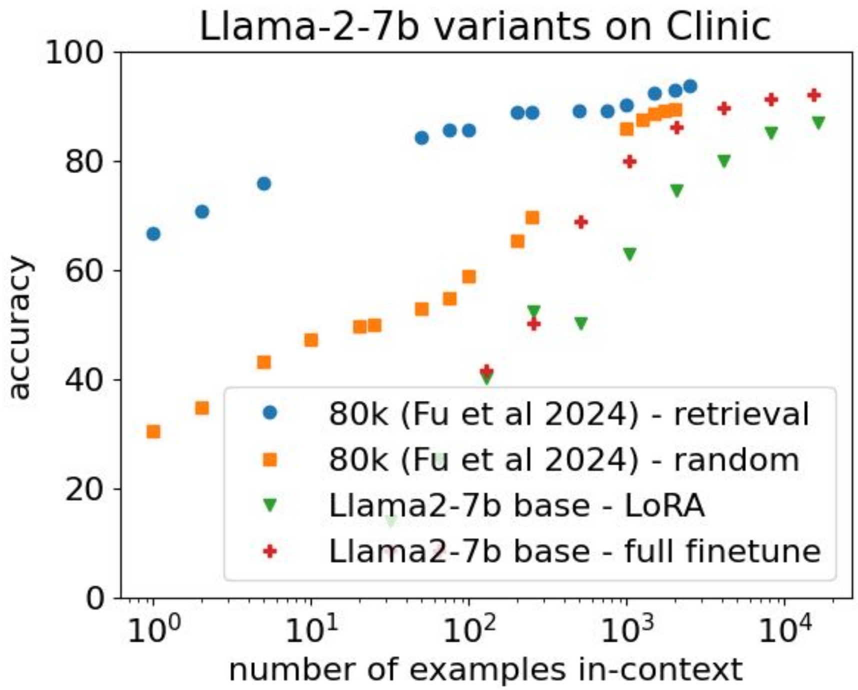


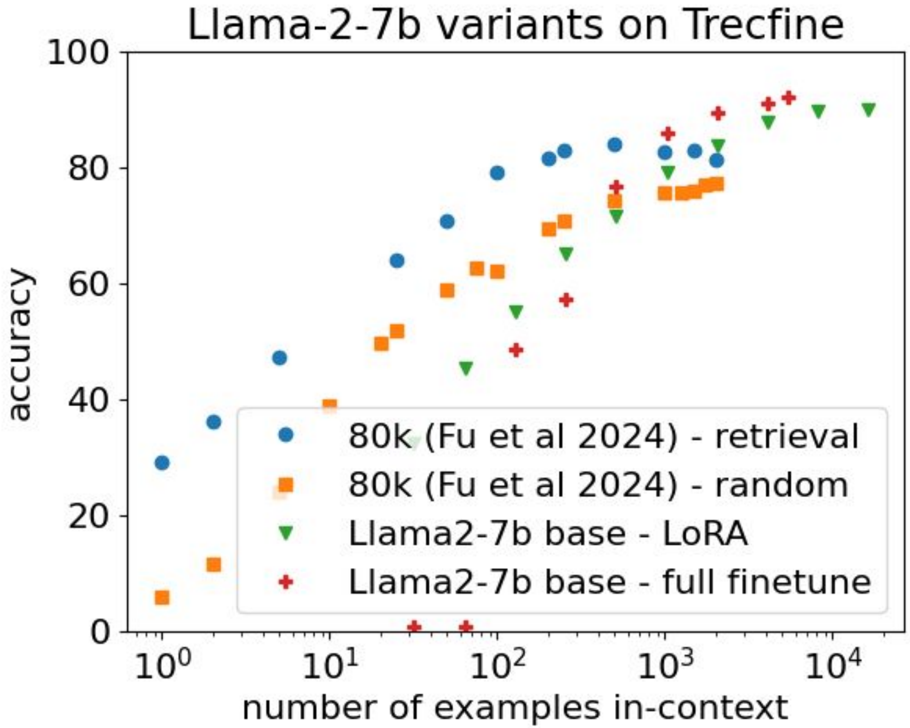
Comparison: results





Long-context ICL benefits less from retrieval

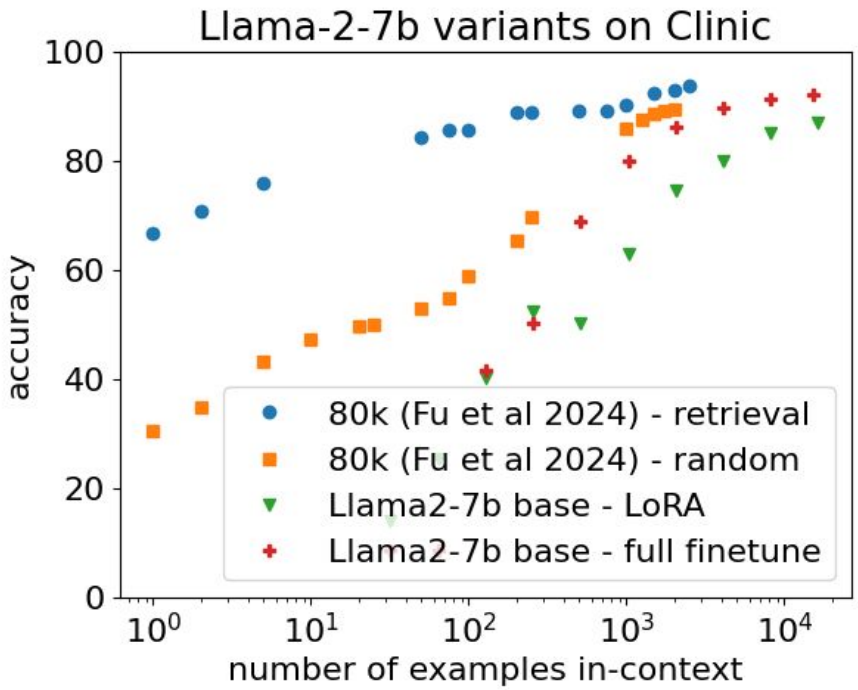


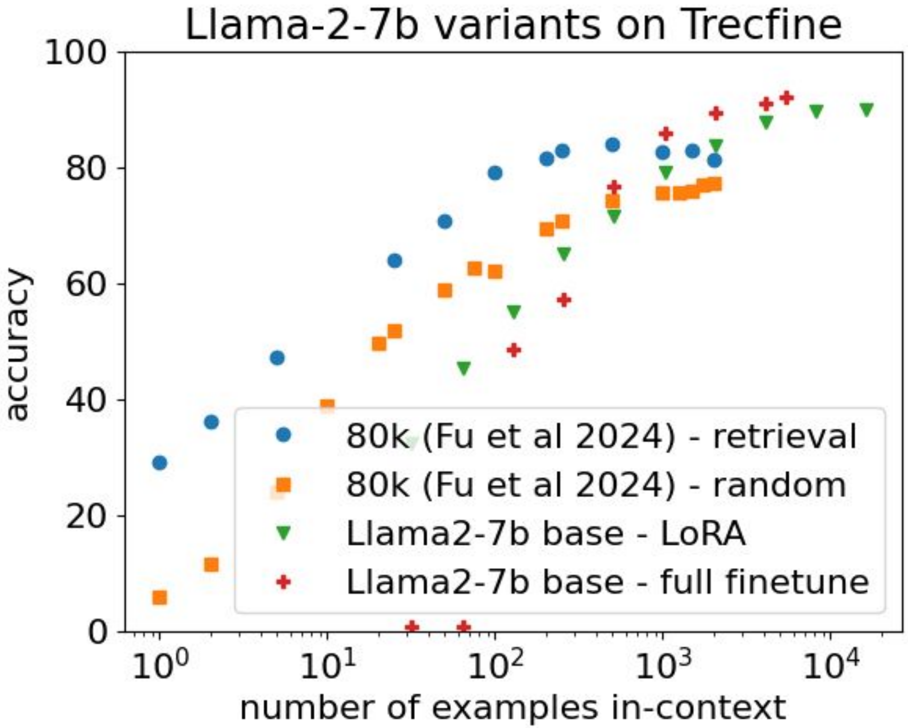


Long-context ICL benefits less from retrieval

- Requires a larger dataset to perform selection from
- Can introduce additional param

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





Efficiency comparisons

2,000 demonstrations, each ~30 tokens long

	Training VRAM requirement	Inference VRAM requirement	Inference speed
LoRA finetuning	76GB	18GB	Fast
Full finetuning	256GB	18GB	Fast
Long-context ICL	None	78GB	Slow
Retrieval ICL (requires >2000 examples)	None	>18GB, <78GB	Medium to slow; depends on retrieval method

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

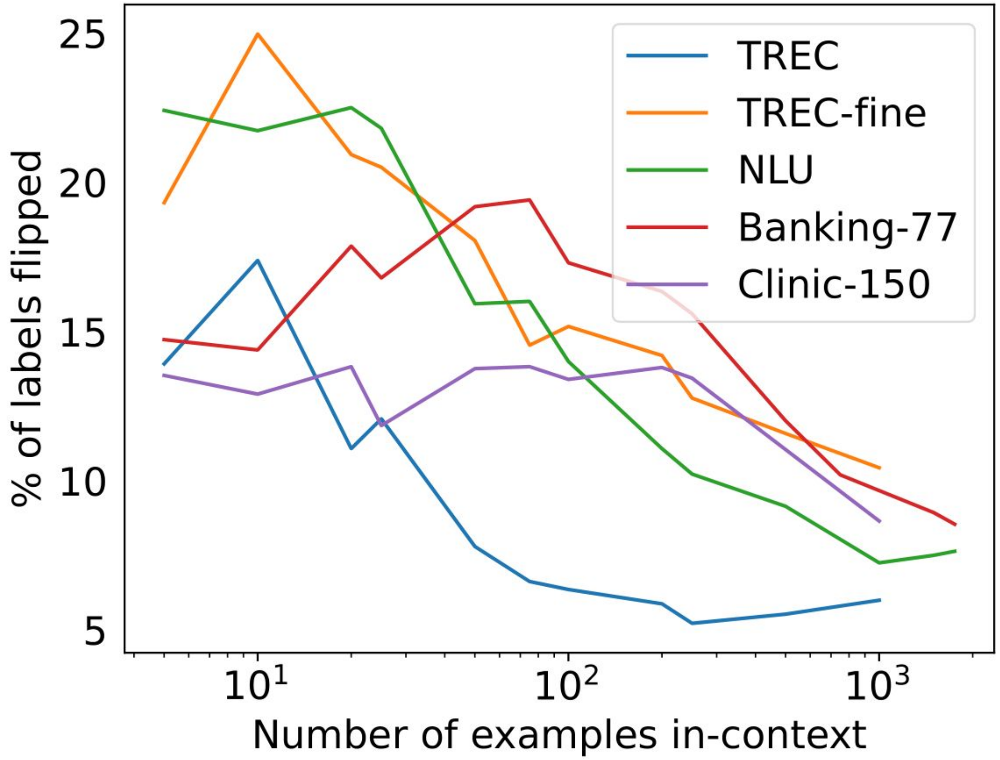
Traditional ICL shows some undesirable sensitivities

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

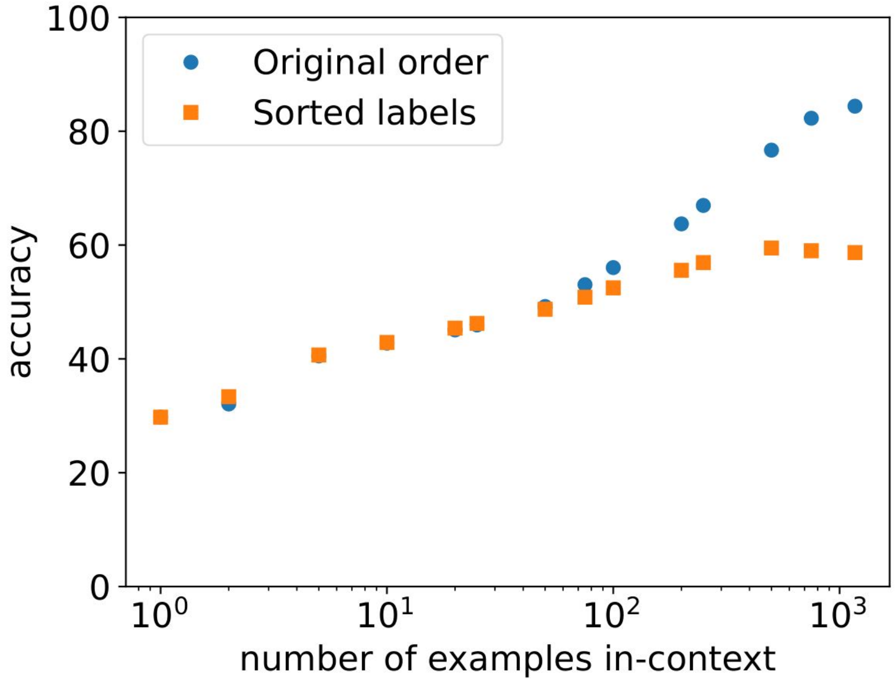
Long-context ICL is less sensitive to randomized example orders...

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?

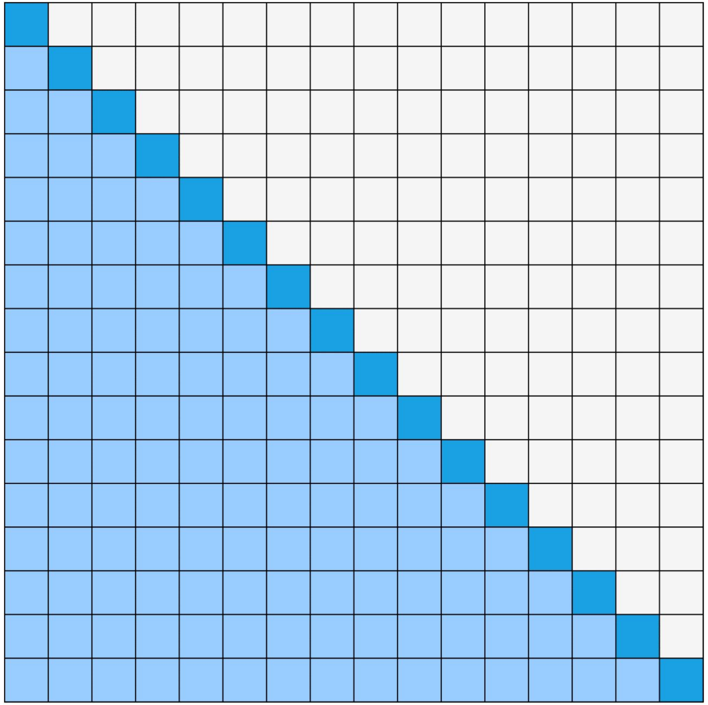
- Local context of all the same label is harmful to performance

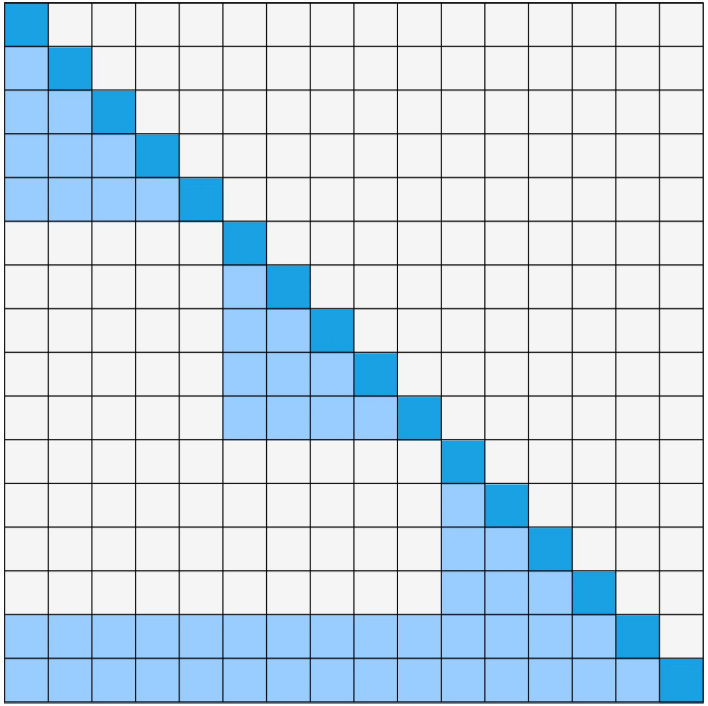
What makes long-context ICL work?

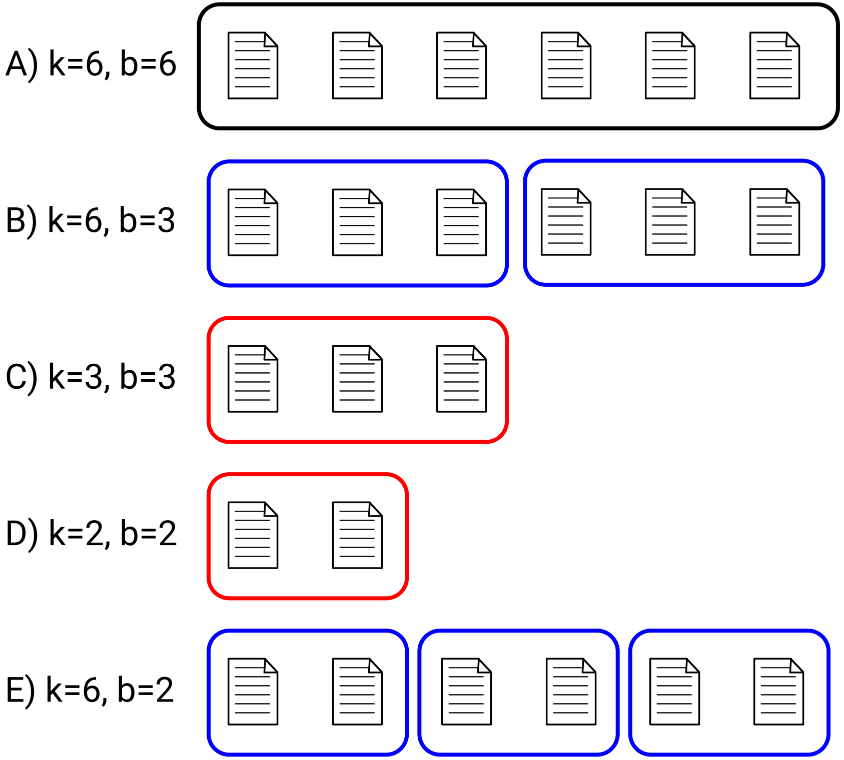
Is it:

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

Block attention

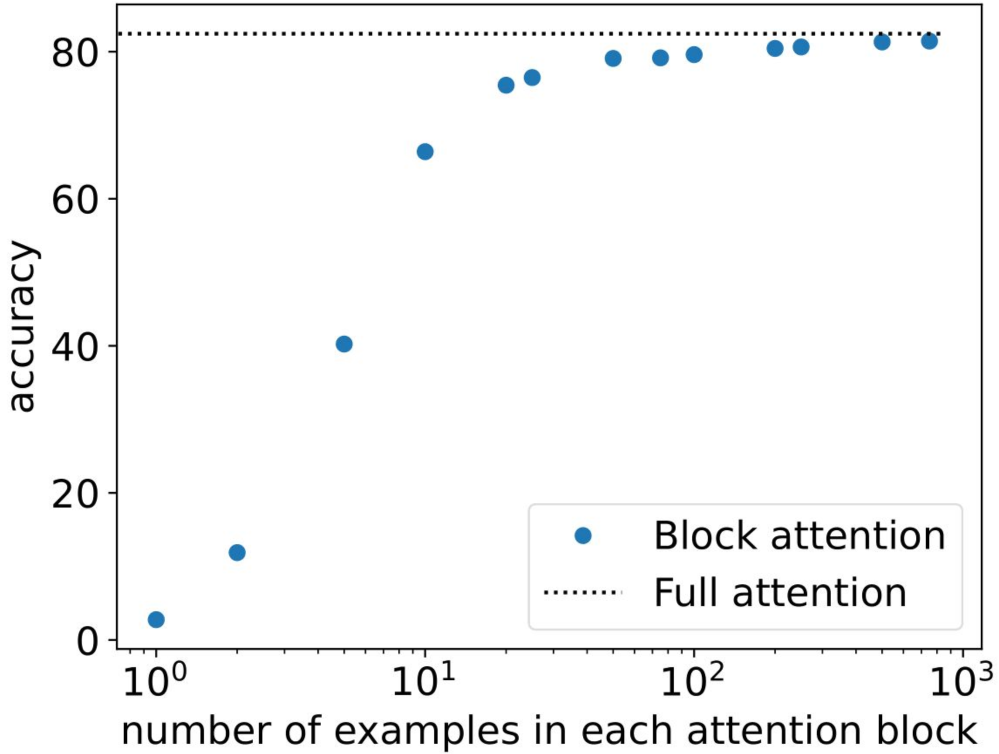





A) k=6, b=6
B) k=6, b=3
C) k=3, b=3
D) k=2, b=2
E) k=6, b=2

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



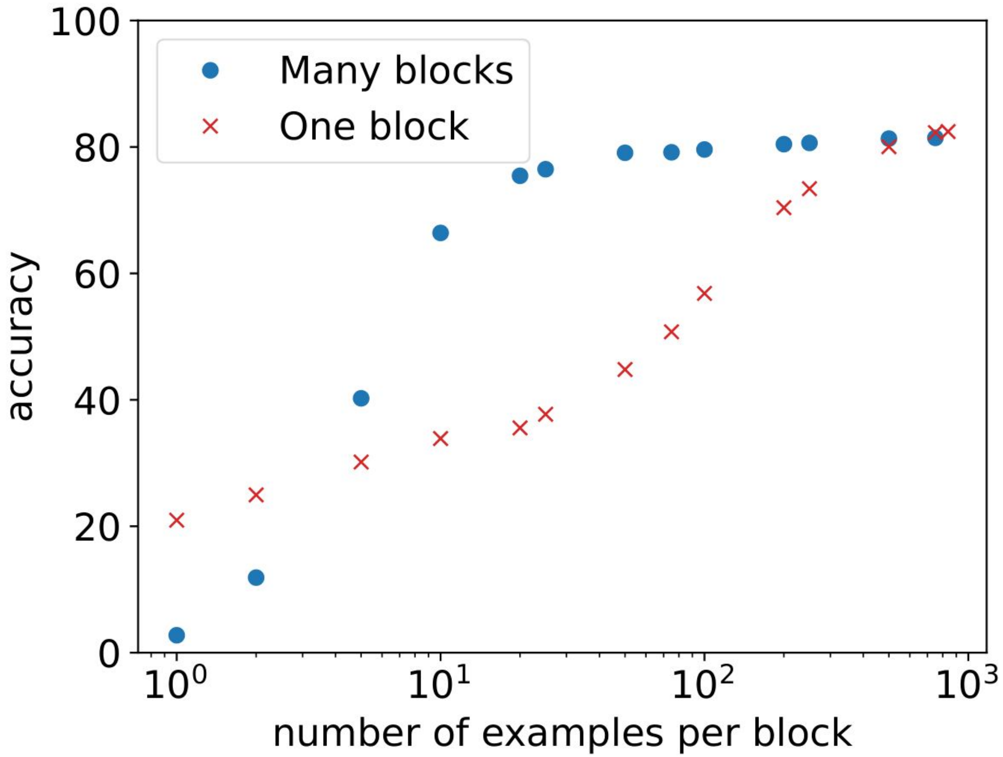
Block attention with one vs many blocks

In short contexts:

- One block outperforms many

In longer contexts:

- Many blocks outperform one



A and B have similar performance; the model does not benefit from the use of long-range cross-attention when encoding the demonstration sets

B outperforms C; the limiting factor in performance is the number of demonstrations, not the quality of contextualization

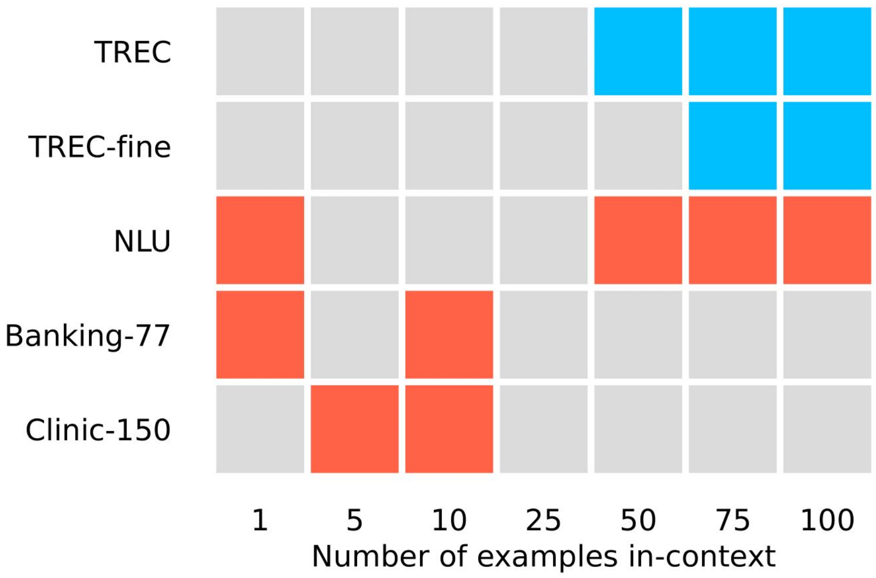
D outperforms E; the limiting factor in performance is the quality of contextualization, so adding more demonstrations with the same amount contextualization does not help

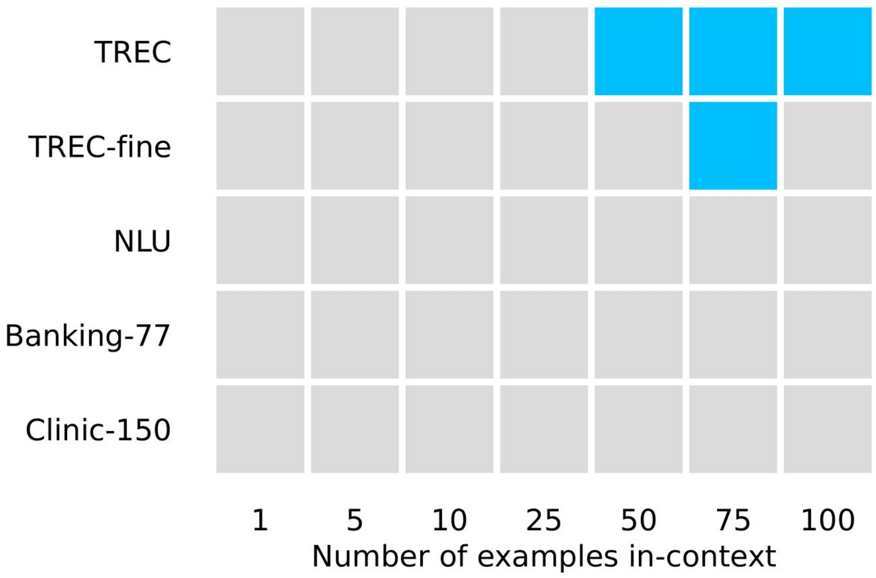
Benchmarking long context models with ICL

Long ICL is **not** a great test of whether models use long-context dependencies :(

Are there other things we can test for, though?

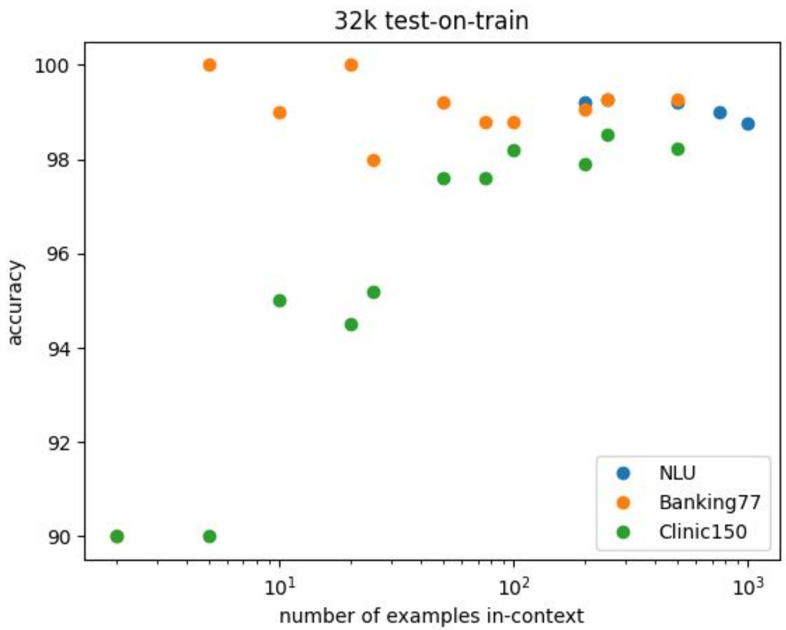
Testing for short-context regression with long-context models?

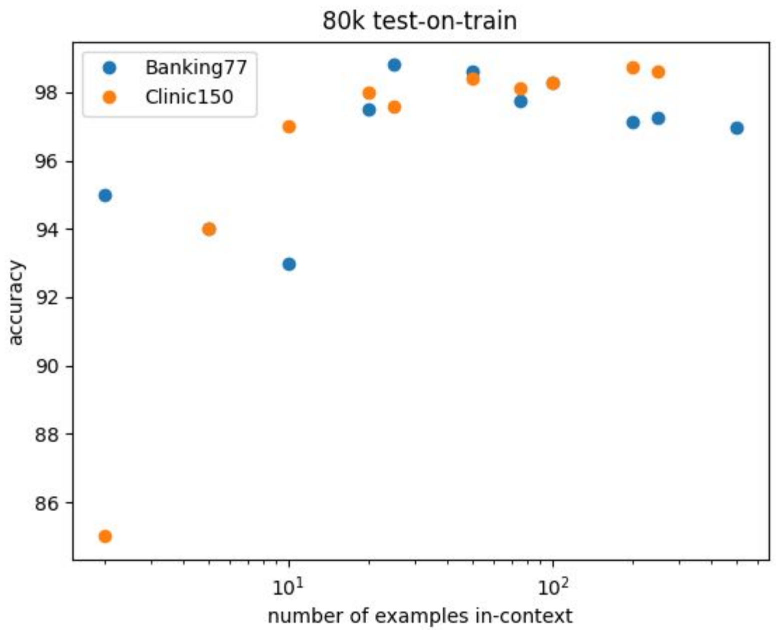




Needle in a needlestack test

Models should be able to copy from input





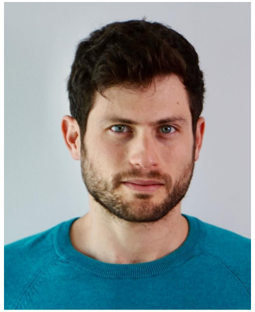
Where do we go from here?

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

Thank you! questions?

Joint work with:



Maor Ivgi

Uri Alon

Jonathan Berant

Graham Neubig

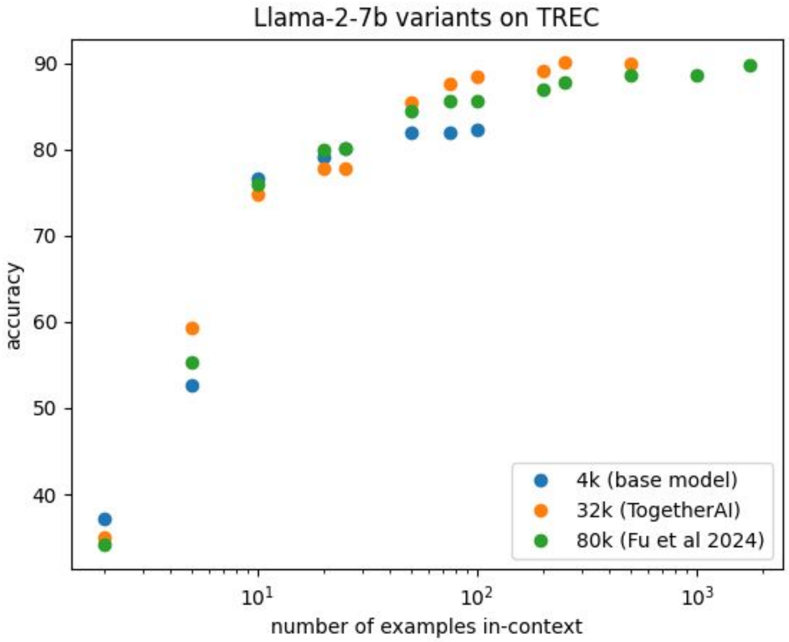
Matt Gormley

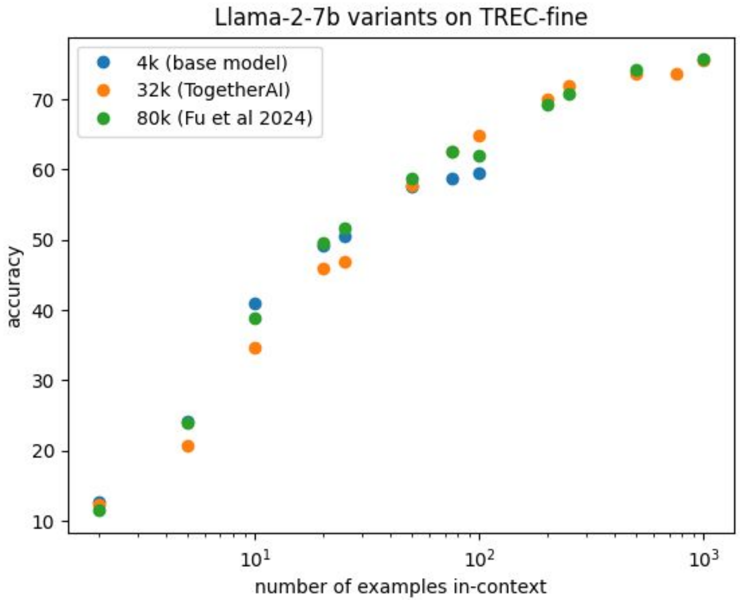
Contact me: abertsch@cs.cmu.edu @abertsch72

Extras

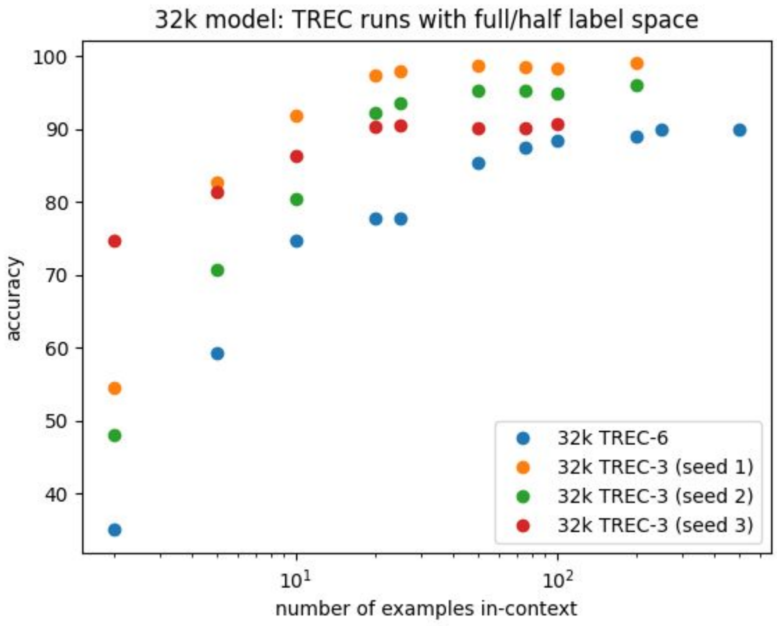
Properties of ICL: saturation point

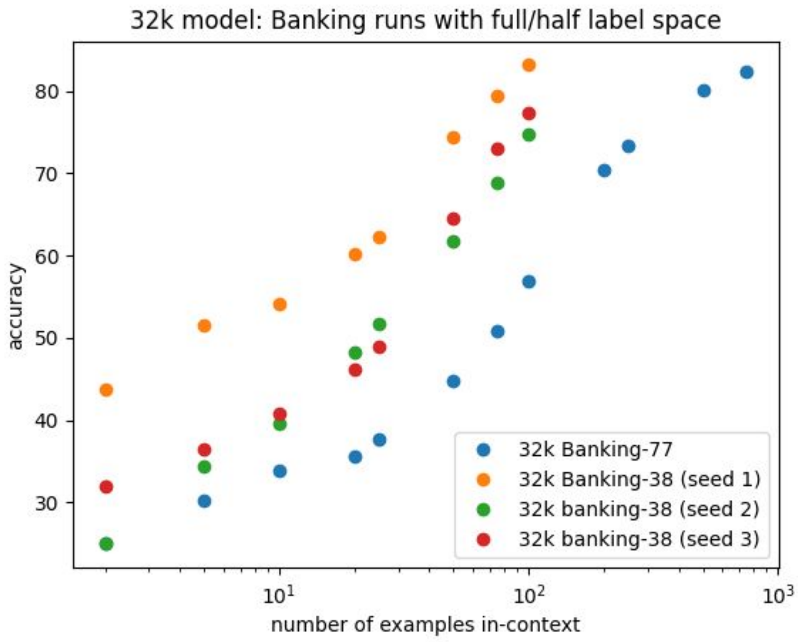
Does label space impact saturation point?





Does label space impact saturation point? Not really...





Not considered here: RLHFing

Why are we using the base and not the chat model?

- Simple answer: base model is slightly better

- Interesting question that we don't answer: are chat models *more* sensitive to prompt formatting than base models?

Not a huge concern: fp16

In our initial tests: nearly the same performance

We *don't* try further quantization, however