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
Emily Xiao[γ] emilyx@cs.cmu.edu
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

[γ] **Carnegie Mellon University** [τ] **Tel Aviv University**

Abstract

As model context lengths continue to increase, the number of demonstrations that can be provided in-context approaches the size of entire training datasets. We study the behavior of in-context learning (ICL) at this extreme scale on multiple datasets and models. We show that, for many datasets with large label spaces, performance continues to increase with thousands of demonstrations. We contrast this with example retrieval and finetuning: example retrieval shows excellent performance at low context lengths but has diminished gains with more demonstrations; finetuning is more data hungry than ICL but can exceed long-context ICL performance with additional data. We use the ICL setting to study several properties of both in-context learning and long-context models. We show that long-context ICL is less sensitive to random input shuffling than short-context ICL, that grouping of same-label examples negatively impacts performance, and that the performance boosts do not arise from cumulative gain from encoding many examples together. We conclude that long-context ICL can be an effective tool, and may not require long-context for encoding the demonstration set at all.[1]

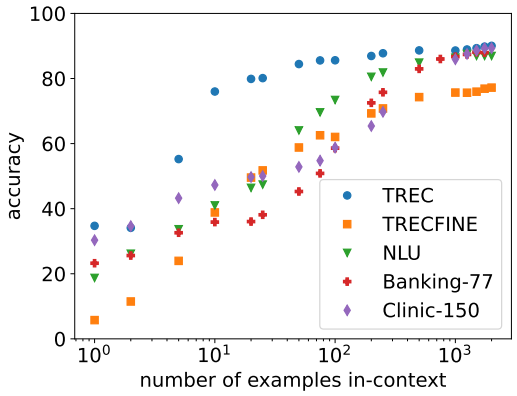


Figure 1: The performance increases with more demonstrations far beyond the context window of the base Llama-2. Results are on Fu et al. (2024)'s long-context finetuned Llama-2-7b model, using a context of up to 80K tokens.

1 Introduction

When a few examples are provided in-context, large language models can perform many tasks with reasonable accuracy. While questions remain about the exact mechanism behind this phenomena (Min et al., 2022b; von Oswald et al., 2023), this paradigm of *in-context learning* (ICL) has seen widespread adoption in both academic and industry applications, thanks to its ease of implementation, relatively small computational cost, and ability to reuse a single model across tasks.

However, most work has focused on models where the maximum number of demonstrations is severely limited by context length. As more and more methods are developed to adapt language models to extreme context lengths (Deepmind (2024); Fu et al. (2024), *inter alia*), in-context learning over large quantities of data becomes a potential alternative to finetuning. The properties of ICL in this regime are not well-understood; and as the cost of inference over many thousands of tokens can be steep, the efficiency and performance tradeoff between many-shot ICL and finetuning on the same data is complex.

We conduct a systemic study of long-context in-context learning. Namely, we consider: a) the performance of prompting the base model naively, b) retrieving examples to use in-context for each test example, c) finetuning the base model (both full and parameter-efficient finetuning), and d) using models trained to adapt to longer contexts. Performance continues to increase past 2000 demonstrations (see Figure 1), approaching and sometimes *exceeding* the performance of models finetuned on thousands of examples from the same dataset (§ 3).

We find that, as the number of demonstrations

*Now at Google DeepMind

[1]Data and code are available at https://github.com/abertsch72/long-context-icl

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 12119–12149

Dataset	Domain	# Labels	Avg demo length	Training set size	Example outputs
TREC	questions	6	22.7	5,452	location / entity
TREC-fine	questions	50	23.7	5,452	abbreviation expansion / location city
NLU	conversational	68	20.7	19,286	takeaway query / iot hue light up
Banking-77	financial	77	27.4	10,003	top up failed / lost or stolen card
Clinic-150	multiple	151	22.3	15,250	rollover 401k / meal suggestion
SAMSum	conversational	n/a	167.9	14,732	John will buy the goat cheese Tracy liked, milk, a couple of grainy rolls, and tissues.

Table 1: The datasets we consider in this work span diverse label spaces and domains. The average demonstration length is the average combined length of input, output, and formatting tokens per demonstration provided in the context window; outliers in the top 1% for demonstration length are discarded and not reflected in these statistics.

in-context increases to extreme values, the behavior of ICL shifts (§ 4). In-context learning becomes less sensitive to example order, and the benefits of retrieval over using a random set of demonstrations diminishes — allowing the use of a single set of demonstrations, encoded once through the model and cached, rather than re-encoding a custom set of demonstrations for each example. We demonstrate that long-context ICL is strongly impacted by grouping examples of the same label. We also find that the effectiveness of long-context ICL is not dependent on long-range attention in the demonstration set– encoding demonstrations with local attention and using global attention only for the test example recovers nearly the same performance (§ 5). Our work furthers the understanding of in-context learning and shows that long-context ICL is a strong alternative to retrieval and finetuning.

2 Experimental setup

We consider 5 classification datasets: TREC (Hovy et al., 2001), TREC-fine (Hovy et al., 2001), NLU (Xingkun Liu & Rieser, 2019), Banking-77 (Casanueva et al., 2020), and Clinic-150 (Larson et al., 2019); and 1 generation dataset: SAMSum (Gliwa et al., 2019). Table 1 contains summary statistics for each dataset, and Appendix G shows additional description for each dataset.

We compare ICL performance across several long- and short-context models, including variants of Llama-2 with 4k (Touvron et al., 2023), 32k (TogetherAI, 2023), and 80k (Fu et al., 2024) context windows, Mistral-7b-v0.2 (Jiang et al., 2023), and Qwen 2.5-7B (Team, 2024). For more details on models, see Appendix H.

Constrained decoding For each classification dataset, we use *constrained decoding* to only produce valid labels as output. Note that, without constrained decoding, these models may produce invalid labels in the few-shot regimes (see Appendix D). For finetuning, we use a classification head; thus no invalid outputs can be produced.

Evaluation Following prior work (Zhao et al., 2021; Lu et al., 2022; Han et al., 2022; Ratner et al., 2022), we subsample 250 examples from the test set of each dataset. We release the subsampled test set and full prediction outputs for each experiment in the project repository. We evaluate on each classification dataset with accuracy and macro-F1; as the trends for the metrics are very similar, we report accuracy (the more common metric) in the paper. We evaluate on SAMSum with BERTScore (Zhang et al., 2020) and confirm that we see similar trends in ROUGE (Lin, 2004), as measured using the `rouge_scorer` package.[2]

3 Long-context ICL

We consider four common methods for using a large dataset.

3.1 Compared settings

Random sampling ICL We use 10 random shuffles of the training dataset, averaging the results across these shuffles. Across models and across

[2] https://pypi.org/project/rouge-score/

Dataset	Llama2	Llama2-32k	Llama2-80k	Mistral	Qwen2.5
Randomly selected					
TREC	**82.32** / 80.52	**93.12 / 93.12**	**90.04 / 90.04**	**87.28** / 85.00	**94.68** / 94.40
TREC-fine	**61.40 / 61.40**	**75.56** / 75.08	**77.20 / 77.20**	**72.68** / 70.48	**83.40** / 81.24
NLU	**76.88 / 76.88**	**85.04** / 85.00	**87.52** / 86.92	**86.44 / 86.44**	**88.64 / 88.64**
Banking-77	**56.36 / 56.36**	**82.44 / 82.44**	**88.08** / 87.96	**86.76** / 86.68	**88.60** / 87.96
Clinic-150	**60.92 / 60.92**	**84.40 / 84.40**	**89.32 / 89.32**	**90.56 / 90.56**	**93.16** / 92.76
SAMSum	**79.83 / 79.83**	**81.65** / 81.42	**81.17 / 81.17**	**81.78 / 81.78**	**82.01** / 81.86
BM25 Retrieval					
TREC	**90.80** / 85.64	**94.84** / 94.64	**94.28** / 92.68	**90.80 / 90.80**	**95.60** / 95.20
TREC-fine	**78.80 / 78.80**	**83.88** / 81.12	**83.92** / 81.36	**80.80** / 79.60	**88.00 / 88.00**
NLU	**90.00** / 88.40	**89.80 / 89.80**	**89.64** / 89.52	**90.40** / 89.20	**90.40 / 90.40**
Banking-77	**93.20** / 92.40	**94.32 / 94.32**	**94.00** / 92.96	**93.20 / 93.20**	**92.80** / 91.60
Clinic-150	**87.60 / 87.60**	**89.84 / 89.84**	**93.76 / 93.76**	**93.20** / 92.40	**95.20** / 93.20
SAMSum	**79.98 / 79.98**	**81.40** / 81.19	**80.68 / 80.68**	**81.33 / 81.33**	**81.90 / 81.90**

Table 2: For all datasets, performance of ICL continues to increase with additional demonstrations. These results are the best accuracy (left) and accuracy at maximum data (right) for each model on the classification tasks, and the same with BERTScore for SAMSum. Bold indicates the best performance for that model/dataset pair.

varying numbers of demonstrations in-context, we draw the first n examples from each shuffle. In this setting, the encoding of demonstrations can be performed once and cached.

Retrieval ICL A strong alternative for in-context learning is to retrieve a relevant subset of examples as demonstrations for each test set example. Prior work has found that, in some scenarios, retrieval of good examples can make the difference from near-zero to high test accuracy (Levy et al., 2023). We considered two possible retrievers for this setting: BM25 (Robertson & Zaragoza, 2009) and BERTScore-Recall (Gupta et al., 2023). For both, we retrieve the most relevant demonstrations by comparing the test input text to the full demonstration texts. For BM25, we remove stopwords; when doing k-shot prompting, if less than k examples are retrieved by the retriever,[3] we randomly sample additional examples until we reach k.

Finetuning We finetune Llama2-7b with a classification head on varying amounts of data from each dataset with several random seeds, and plot performance at convergence on the same held-out test data. We initialize the classification head from the parameters of the pretrained language modeling head by subsampling the values of the first token of each label; this creates a better-than-random initialization for finetuning. We perform both full finetuning and LoRA (Hu et al., 2022) finetuning using this setup; for more details on the finetuning procedures, see Appendix E.

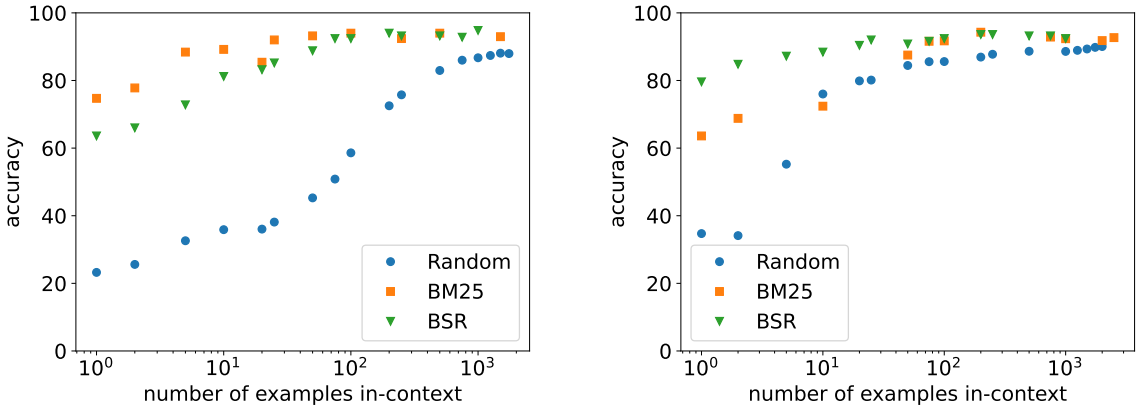
3.2 In-context results

Scaling up ICL to many examples leads to surprisingly strong results Figure 1 and Table 6 show the performance of models in the ICL settings. Scaling up from 10 to 1000 demonstrations results in accuracy gains of up to *50.8* points (and an average of 36.8 points across 5 datasets for Llama2-80k).

Longer context lessens the importance of carefully selecting in-context examples Retrieving relevant examples for each test set example far outperforms using a randomly selected subset in the short-context regime. This is true even if the order of retrieved examples is shuffled (rather than ordered by relevance).[4] However, adding additional examples does continue to slightly improve performance; this is especially surprising for the BM25 retriever because, after all examples with non-trivial lexical overlap are retrieved, remaining

[3]This occurs when there are less than k examples with any (non-stopword) overlap with the test example.

[4]We perform three random shuffles of the BM25 retrieved inputs and test for difference in distribution from the original results; this does not significantly change performance for any dataset (2-sided t-test, $p < 0.05$).



(a) Banking-77

(b) TREC

Figure 2: Comparing three selection methods– random selection, BM25, and BERTScore-Recall (BSR) on two representative datasets. At smaller numbers of demonstrations in-context, BM25 and BSR have differing performance, and the best retriever is dataset-specific; at larger demonstration counts, the two become indistinguishable. Both generally outperform random selection.

examples are randomly selected.

While retrieval continues to outperform random selection, the importance of the selection strategy diminishes with additional examples. On some datasets (e.g. TREC, in Figure 2b), BERTScore-Recall outperforms BM25 for short context ICL; on other datasets (e.g. Banking-77, in Figure 2a), the inverse is true. But on all datasets, the performance difference between the two retrieval methods diminishes with larger k, so that BM25 and BSR have nearly identical performance at long-context ICL. Because of this, we report only BM25 in the remainder of the analysis.

As the performance difference between individual retrievers diminishes, so does the performance difference between retrieval and random selection of demonstrations. On Banking-77, the dataset where retrieval is most beneficial, the performance gain from BM25 retrieval drops from 51.5 points at 1-shot ICL to 4.9 points at 1500-shot ICL. This is compelling because it is more computationally efficient (but less effective) to encode a single random set of demonstrations and cache them, rather than retrieving and re-encoding a custom set of demonstrations for each inference example. In the longest context regime we consider, using a single randomly selected set of examples is feasible; the performance penalty for doing so is never more than 5 points, and as low as 1.8 points (in 2000-shot ICL on TREC).

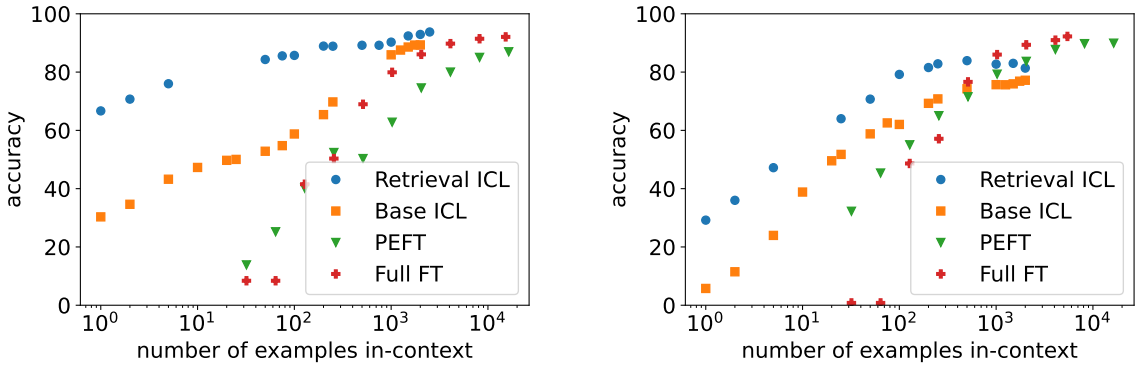
Long-context ICL is also effective for generation. While we primarily focus on classification tasks because of the relative ease of evaluation, we do consider SAMSum, a text summarization task, for our ICL experiments in Table 6. While the number of demonstrations possible in the same context length is much smaller, due to the increased lengths of both inputs and outputs, we observe increased performance with additional demonstrations up to at least 250-shot ICL. Retrieval seems less helpful in this setting, with retrieval sometimes underperforming random selection.

3.3 Comparison with finetuning

While we have demonstrated that in-context learning with hundreds or thousands of examples is effective, this amount of data is also appropriate for finetuning a model. Finetuning has higher upfront cost but allows for reduced inference-time cost. We compare in-context learning with full finetuning and the popular parameter-efficient finetuning (PEFT) strategy LoRA (Hu et al., 2022).

Finetuning is (slightly) more data-hungry than ICL When a relatively small set of examples is available, ICL generally outperforms LoRA finetuning on the same model.[5] For most datasets, LoRA finetuning performance never exceeds long-context ICL performance even with additional examples (e.g. Figure 3a); however, for most datasets, full finetuning with drastically more examples than fit in-context yields the highest performance. Gen-

[5]Note that some prior works have showed strong PEFT performance in the few-example setting on different tasks; see Section 6 for more discussion.



(a) Clinic-150

(b) Trecfine

Figure 3: Comparing BM25 retrieval ICL, random selection ICL, and two types of finetuning on two representative datasets. Finetuning sometimes, but not always, exceeds ICL at high numbers of demonstrations. Note that, while retrieval ICL uses the listed number of examples in context, it assumes access to the larger test set to draw examples from (Perez et al., 2021). See Appendix C for results on other datasets.

.

erally, the datasets with larger label spaces show the least strong finetuning performance, likely because these are more open-ended classification problems and require more data to train the classifier; on the dataset with the most labels, Clinic-150, neither LoRA finetuning nor full finetuning ever outperforms ICL at the same number of examples.

In a setting with unlimited training data available, then, finetuning is clearly advantageous over ICL for any model with a fixed maximum context length. Finetuning also offers dramatically reduced inference costs for similar performance; thus, finetuning on 4096 examples may still be preferable to prompting with 1000 if efficiency of inference is a major priority.[6] This is because, even if demonstration encodings can be cached across inference examples, cross-attention to long context is costly.

4 Properties of long-context ICL

We compare the properties of long-context ICL with the known properties of short-context ICL.[7] We primarily use classification, not generation, tasks to study these properties to avoid confounding effects from the difficulty of evaluating generated texts.

Is it best to use the entire context? Prior work suggested that, for some simple tasks, providing additional input can *reduce* performance (Levy et al., 2024). However, we observe monotonically increasing performance on nearly every dataset; in cases where the performance curve begins to flatten, small variation occurs, but no significantly lower performance occurs at higher example counts. While using the full context window is computationally costly and may be unnecessary to achieve high performance on some datasets, it is minimally not harmful to performance.

Sensitivity to example order Many models exhibit strong sensitivity to example order in-context (Lu et al., 2022). We examine this by measuring the percentage of predictions that change when the input is reordered (averaged over 3 shuffles). Figure 4 shows that, while some sensitivity to order persists, this effect weakens substantially with longer context. Across all datasets, the percent of labels flipped by shuffling in 1000-shot ICL is less than *half* the percentage flipped in 10-shot ICL.

Label sorting We also consider an adversarial case for example ordering: we sort the examples so that examples with the same label appear together. At small numbers of examples, this has very little impact; if the average number of examples per class is low, label sorting is similar to a random sort. However, as the number of examples grows, label sorting begins to have a dramatic impact on performance. Figure 5 shows the performance of Llama2-32k on Clinic-150 with and without label sorting. As the number of examples in-context increases, the penalty for input sorting increases as

[6] With the caveat that serving several task-specific models may be more expensive than serving one general-purpose model with customized ICL prompts; the actual best cost/efficiency tradeoff in any downstream setting is of course dependent on the needs of the deploying organization.

[7] We consider using ICL as a testbed for properties of long-context models in Appendix B.

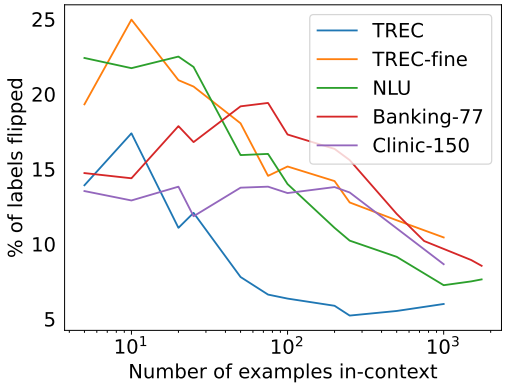


Figure 4: The impact of (randomly) reordering examples in-context decreases with additional demonstrations.

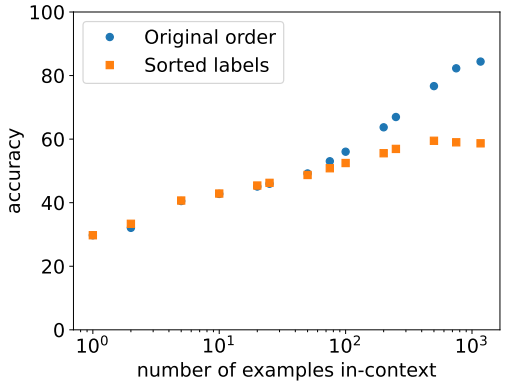


Figure 5: By contrast, sorting examples by label has an increasingly negative impact on performance in longer context regimes. Results on Llama2-32k with Clinic-150.

well; at 1169-shot ICL, label sorting decreases accuracy by 25.7 percentage points. This suggests that contextualization of examples with *different* labels is important to performance, and that this contextualization only occurs effectively over relatively short distances in the context window.

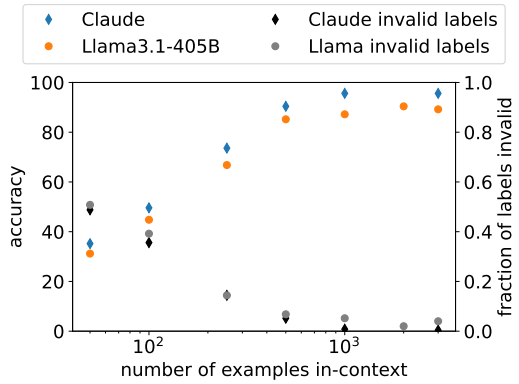


Figure 6: Performance of two frontier models on Clinic-150. Performance increases with the number of demonstrations at first, but saturates relatively early; the number of invalid labels produced continues to decline with increased demonstrations, even when accuracy plateaus.

Effectiveness for frontier models We focus on 7B/8B models because of the feasibility of in-depth analysis; however, it is also useful to consider the performance of the current strongest long-context models. We evaluate Claude Sonnet 3.5 (Anthropic, 2024) and Llama 3.1 405B (Dubey et al., 2024), and using 50 to 3000 demonstrations from Clinic-150. Due to computational cost and (in the case of Claude) the limitations of API access, we do not apply constrained decoding in these runs and use only a single run over the test set instead of 10 runs at each example count. Figure 6 shows these results. While performance increases past the typical fewshot range, at least some of this benefit comes from improved knowledge of the label space (i.e. less invalid labels generated) and performance on this task saturates quite quickly. Thus, while long-context ICL seems to be a promising direction even for frontier models, there may be diminishing returns on relatively simple tasks such as intent classification at even the 1,000-example scale.

Tasks where long context does not help Concurrently to our work, Li et al. (2024) identify a set of tasks where long context is not uniformly helpful. However, we observe that the tasks that show this trend either have near-0 performance at short demonstration lengths or also display an inverse performance trend on the short context scale (e.g. for TacRED (Zhang et al., 2017), we observe that performance decreases from 1 to 10 total demonstrations; on Discovery (Sileo et al., 2019), we observe that performance is near-zero at all demonstration lengths and decreases from 5 to 10 total demonstrations). While these are important failure modes of language models, we restrict our analysis to tasks without these confounding issues. In Banking-77, the one dataset that our works share, both papers observe similar trends of improved performance with additional context.

5 Why does long-context ICL help?

The most notable differences between long-context and short-context ICL are the additional number of demonstrations and the average number of demonstrations that each demonstration is contextualized

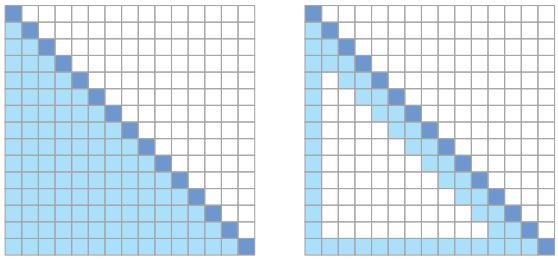
with respect to– that is, the same demonstration may have a different impact on the prediction if it is the only demonstration in-context versus if it is the 90th demonstration to appear in-context, because the demonstration will be better-contextualized.

We hypothesize that long-context ICL is primarily beneficial because of *retrieval in-context*– the idea that attention is used to select examples, rather than the model aggregating a complex decision boundary by better contextualizing across examples. This means that the primary benefit of long-context ICL is the number of demonstrations, *not* the quality of contextualization that each individual demonstration receives. If this is the case, then there should exist some sparse attention pattern that severely restricts long-range attention *between* demonstrations, but has limited or no impact on ICL performance.[8] In all cases, the current test example can attend back to all demonstrations, in the same style as Acharya et al. (2024).

Block-sparse attention patterns We test several previously proposed sparse attention patterns and observe that two things are necessary to enable block attention: attending to an *attention sink* (Xiao et al., 2024) block that is always first in the context window; and attending to two prior local blocks (Guo et al., 2024). We ablate over these decisions in Appendix F.

The pattern we identify, a small variation on Star Attention (Acharya et al., 2024), is visualized in Figure 7. This attention pattern sparsifies attention between demonstrations substantially, removing almost all long-range connectivity between demonstrations in-context, without substantial impact on performance. This suggests that long-context ICL does not, in the encoding of the demonstration set, require long-range attention.

This pattern can also be used to disentangle two previously conflated factors: k, the total number of demonstrations that the test example can attend back to (e.g. k-shot prompting); and b, the number of examples per block in the blockwise attention pattern.[9] block of examples that are contextualized together. We can fix the total number of examples in context, k, and vary b by changing the attention mask; and we can fix b with a custom attention mask and vary k by adding additional blocks of examples to the context window.



(a) Causal attention (b) Blockwise attention

Figure 7: Causal attention versus the blockwise pattern we apply in the following experiments. Here, each square represents a *block* of examples, and the last square represents the test example. This pattern represents attending to the first block (the attention sink) and two local blocks.

Restricting contextualization on a fixed example set We fix a number of examples per-block. If the block size is equal to the number of examples in-context ($b = k$), this is equivalent to normal (full) attention; if the block size is $b = 1$, each example can attend only to itself and its sink/local block. Figure 8 shows results on Banking-77 as a representative example. The performance of block attention quickly approaches the performance of full attention; 95% of the performance of full attention is recovered by a block of 50 examples in the case of Banking-77 (and in a similar range for all datasets studied). This suggests that the ICL performance is not strongly benefiting from encoding long-range (or even medium-range) dependencies across the demonstration set. However, some short-range dependencies between examples are clearly necessary for ICL performance, as using block sizes $b < 10$ results in near-zero performance and performance increases slightly with increased size from there.

Increasing number of examples with a fixed contextualization quality We fix the block size and compare attending to a single block ($k = b$) to attending to many blocks ($k >> b$) in Figure 9. This comparison measures the effect of adding additional examples without introducing substantial improvements to the contextualization of each individual example.

When the context is extremely local (e.g. for banking, $b < 10$ examples), attending across many locally encoded blocks is *worse* than attending to a single block of examples. We hypothesize that

[8]Note that this is distinct from methods that overload the same positions with multiple embeddings in order to process longer contexts (e.g. (Ratner et al., 2022)); here, we are not modifying any positional information, only restricting attention between demonstrations to a local context block.

[9]Note that each block attends to two blocks of local context ($2b$ examples) and a sink block (b examples), so that the maximum number of demonstrations that the *last* demonstration in the last block can attend to is actually $4b - 1$.

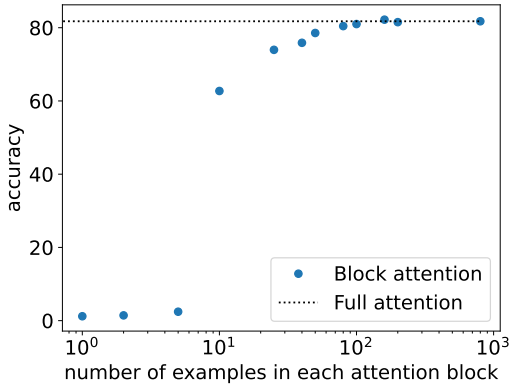


Figure 8: Comparing block attention to full attention over the same set of examples with Llama2-32k on Banking-77 (i.e., fixing k and varying b along the x axis). Block attention approaches full attention (the black skyline) with relatively small block size.

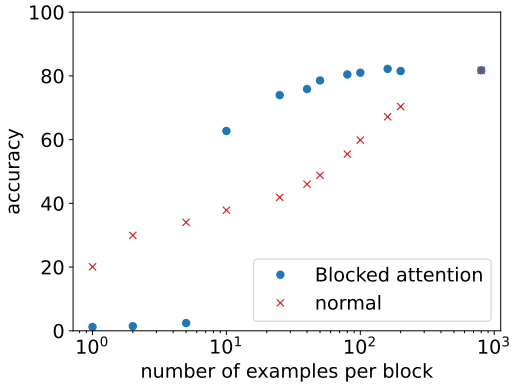


Figure 9: Comparing a single block to many blocks with a fixed block size using Llama2-32k on Banking-77 (i.e., fixing b and varying k). When the block size is very small, poor contextualization appears to be the limiting factor on performance (so adding more examples with the same block size does not help); when the block size is larger, the total number of examples appears to be the limiting factor on performance (so adding more examples with the same block size is helpful).

this is due to inadequate contextualization of each example leading to less informative embeddings. However, after some minimal average quality of contextualization is achieved (in Figure 9, around $b = 10$), adding more blocks of examples with the same local encoding dramatically increases performance. This supports the hypothesis that the primary performance improvement from long-context modeling is due to retrieving from more relevant examples in-context, rather than learning a better task boundary; this is supported as well by the retrieval results in Table 6, where retrieval performance at short contexts is near (though never exceeding) very-long-context ICL performance.

6 Related Work

Augmenting decoder-only models with long context Many methods for extending the context of language models have been introduced in the last few years. One series of work has focused on positional embedding extrapolation strategies (Peng et al., 2023; Rozière et al., 2024; Chen et al., 2023; Liu et al., 2023; Zhu et al., 2024; Xiao et al., 2024; Han et al., 2024). When extrapolating past pretraining length, models also generally benefit from additional finetuning on long-context data (Xiong et al., 2023). Other methods include adding retrieval-based attention (Bertsch et al., 2023; Tworkowski et al., 2023; Yen et al., 2024) or hierarchical merging of information (Song et al., 2024; YU et al., 2023). The two long-context Llama variants we consider in this work are both examples of finetuning for length extrapolation.

Separately, methods for longer context for ICL have also been proposed. Parallel context windows (Ratner et al., 2022) and structured prompting (Hao et al., 2022) propose methods of re-using the same positional embeddings multiple times to encode more demonstrations; this is quite effective for small numbers of overlaps, albeit with diminishing returns as the number of overlapping windows increases. Li et al. (2023c) propose a new efficient attention mechanism and motivate its use using long-context ICL on a model tuned with many-shot instruction finetuning. Cho et al. (2023) propose a hybrid of ICL and linear prompting which improves beyond few-shot ICL performance.

Several works have also critiqued the efficacy of long context models. Liu et al. (2024) demonstrate that some long-context models fail to effectively use the middle of the context window; the models we use were released after this work and have generally high scores for middle-of-context retrieval. Li et al. (2023a) suggest that some long-context models are only effective at utilizing inputs that are shorter than their context window's intended supported length; we do not observe this effect strongly, though we do see saturating performance slightly before the maximum number of examples in-context for some models. Li et al. (2023b) show that many models fail at tasks that require reasoning over long dependency lengths; this is unlikely to be an issue in our setting.

Properties of in-context learning Milios et al. (2023) study ICL for many-class classification with models up to 4k context length and find that, when

retrieving demonstrations, 7b models show early performance saturation on many tasks. Our results suggest that this failure to use longer context effectively is not an inherent property of 7b models, but instead a type of shallow heuristic used by some particular models when the demonstrations are of sufficiently high quality.

Xu et al. (2023) study the impacts of ground-truth label, input distribution, and explanations on ICL performance; Bölücü et al. (2023) study the impact of example selection in a specific domain. Lin & Lee (2024) argue that ICL occurs in two modes: learning tasks and retrieving tasks, and that retrieval of similar-but-not-quite-correct tasks can explain "early ascent" behaviors where ICL performance peaks once in a fewshot regime and then performance improves again with a much higher number of examples. Similarly, Pan et al. (2023) argue for a distinction between task recognition and task learning, and suggest that task learning continues to benefit from additional examples at scale. von Oswald et al. (2023) suggest in-context learning can be viewed as gradient descent, although Deutch et al. (2024) argue against this interpretation. Hendel et al. (2023) view ICL as compressing the demonstrations into a "task vector" that maps from inputs to outputs.

Concurrently to our work, Agarwal et al. (2024) study many-shot prompting of Gemini 1.5 and show improvements from the fewshot setting across both classification and generation tasks. Our work differs in its evaluation of multiple open-source models, our comparison to finetuning the same base model, and our use of ICL as a testbed for analysis of long context behaviors.

Comparing in-context learning and finetuning Min et al. (2022a) show that models trained on fewshot learning can generalize to perform few-shot learning on new tasks; in some cases, this can outperform finetuning directly on the new task. Mosbach et al. (2023) compare finetuning to ICL more directly; they find that finetuning generally outperforms ICL with the same number of examples both in-domain and out-of-domain, when comparing 16-example ICL to finetuning on the same 16 examples. Their setting differs from ours in their choice of model (OPT) and the amount of data considered (16 for ICL, 16 or 128 for finetuning). Liu et al. (2022) find that PEFT generally outperforms ICL in their setting, where they finetune an encoder-decoder model with a language modeling objective using their T-few method and 20-70 samples. Asai et al. (2023) compare finetuning and ICL for mT5 on cross-lingual transfer and find that ICL outperforms finetuning in some, but not all, of the tasks studied. To the best of our knowledge, no prior work has considered the relative performance of finetuning and ICL in the many-shot regime with hundreds or thousands of examples in-context.

7 Conclusion

We have demonstrated that ICL with large demonstration sets can be surprisingly effective, and shed light on a few surprising properties in its behavior. Namely, long-context ICL exhibits a reduced dependence on example selection, relatively stable performance with respect to example order, and performance often approaching or exceeding parameter-efficient finetuning on the same data, all properties that make this an appealing option for a variety of tasks. We have also shown that long-context ICL's effectiveness is largely due to retrieval from the long context during prediction, rather than cross-attention within the large demonstration set during encoding.

Our work also highlights that our understanding of ICL remains incomplete. Though much work has studied the potential mechanisms behind ICL, these works have largely focused on simple tasks with small (< 10 examples) demonstration sets; as our work demonstrates that ICL's properties shift in the long context regime, more work is necessary to validate hypotheses about ICL at larger scales.

While prior work has focused on two strategies for performing inference on a new task– either finetuning on task-specific data or selecting a subset of that data to use in-context– our results points to a potential third paradigm: adapting the *model* to fit as much of that data in-context as possible, caching and reusing the encoding of the long demonstration set. While finetuning with full datasets is still a powerful option if the data vastly exceeds the context length, our results suggest that long-context ICL is an effective alternative. ICL trades finetuning-time cost for increased inference-time compute, and increasing the amount of inference-time compute by using more examples in-context is an effective strategy to improve performance. As the effectiveness and efficiency of long context models continue to increase, we believe long-context ICL will be a powerful tool for many tasks.

8 Limitations

Our work focuses on open-source models (and predominately on the Llama-2 family); more work is necessary to establish whether this trend holds across model families, although we are encouraged by the strong results Agarwal et al. (2024) observed for many-shot ICL on Gemini 1.5. Additionally, we do not consider other non-LoRA PEFT methodologies; it is possible that some of these methodologies may outperform in-context learning. Finally, we focus primarily on classification tasks, and care should be taken in generalizing to new tasks.

9 Broader impacts

Any work that studies general capabilities of language models can be used for both positive and negative applications downstream. In-context learning work is perhaps particularly vulnerable to this dual use, in part because it is a relatively accessible method, requiring far less compute than finetuning or pretraining models.

Independent of our work, Anil et al. (2024) observed that many-shot prompting can be used to jailbreak some models. We do not study or suggest the use of long-context ICL for jailbreaking in our work, but this is a potentially harmful use case of this paradigm. Outside of this risk, we do not foresee any additional harms introduced by long-context ICL methods.

We hope that our work opens up additional possibilities for people who are compute-constrained to customize models to their use cases or tasks. We also aim to carefully explore the boundary between where finetuning and long-context ICL is appropriate, to enable practitioners to make informed choices on when to apply each method.

References

Shantanu Acharya, Fei Jia, and Boris Ginsburg. Star attention: Efficient llm inference over long sequences, 2024. URL https://arxiv.org/abs/2411.17116.

Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. Many-shot in-context learning, 2024.

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, Francesco Mosconi, Rajashree Agrawal, Rylan Schaeffer, Naomi Bashkansky, Samuel Svenningsen, Mike Lambert, Ansh Radhakrishnan, Carson Denison, Evan J. Hubinger, Yuntao Bai, Trenton Bricken, Timothy Maxwell, Nicholas Schiefer, Jamie Sully, and Alex Hernandez. Many-shot jailbreaking, 2024.

Anthropic. Introducing claude 3.5 sonnet, 2024.

Akari Asai, Sneha Kudugunta, Xinyan Velocity Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. Buffet: Benchmarking large language models for few-shot cross-lingual transfer, 2023.

Amanda Bertsch, Uri Alon, Graham Neubig, and Matthew Gormley. Unlimiformer: Long-range transformers with unlimited length input. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2023.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Gregory Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Necva Bölücü, Maciej Rybinski, and Stephen Wan. impact of sample selection on in-context learning for entity extraction from scientific writing. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.338.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pp. 38–45, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.nlp4convai-1.5.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation, 2023.

Hyunsoo Cho, Hyuhng Joon Kim, Junyeob Kim, Sang-Woo Lee, Sang goo Lee, Kang Min Yoo, and Taeuk Kim. Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners, 2023.

Google Deepmind. Our next-generation model: Gemini 1.5, 2024.

Gilad Deutch, Nadav Magar, Tomer Bar Natan, and Guy Dar. In-context learning and gradient descent revisited, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela

Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick

Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. Data engineering for scaling language models to 128k context, 2024.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Association for Computational Linguistics, 2019. doi: 10.18653/v1/d19-5409. URL http://dx.doi.org/10.18653/v1/D19-5409.

Junxian Guo, Haotian Tang, Shang Yang, Zhekai Zhang, Zhijian Liu, and Song Han. Block Sparse Attention. https://github.com/mit-han-lab/Block-Sparse-Attention, 2024.

Shivanshu Gupta, Matt Gardner, and Sameer Singh. Coverage-based example selection for in-context learning, 2023. URL https://arxiv.org/abs/2305.14907.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. Lm-infinite: Zero-shot extreme length generalization for large language models, 2024.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. Prototypical calibration for few-shot learning of language models, 2022.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. Structured prompting: Scaling in-context learning to 1, 000 examples. *ArXiv preprint*, abs/2212.06713, 2022.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.624.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Maor Ivgi, Uri Shaham, and Jonathan Berant. Efficient long-text understanding with short-text models. *Transactions of the Association for Computational Linguistics*, 2023. doi: 10.1162/tacl_a_00547.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *ArXiv preprint*, abs/2312.03732, 2023.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1131.

Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.78.

Mosh Levy, Alon Jacoby, and Yoav Goldberg. Same task, more tokens: the impact of input length on the reasoning performance of large language models, 2024.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. How long can context length of open-source LLMs truly promise? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023a.

Jiaqi Li, Mengmeng Wang, Zilong Zheng, and Muhan Zhang. Loogle: Can long-context language models understand long contexts?, 2023b.

Mukai Li, Shansan Gong, Jiangtao Feng, Yiheng Xu, Jun Zhang, Zhiyong Wu, and Lingpeng Kong. In-context learning with many demonstration examples, 2023c.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning, 2024.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Ziqian Lin and Kangwook Lee. Dual operating modes of in-context learning, 2024.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context, 2023.

Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00638.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8086–8098, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.556.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Aristides Milios, Siva Reddy, and Dzmitry Bahdanau. In-context learning for text classification with many labels. In Dieuwke Hupkes, Verna Dankers, Khuyagbaatar Batsuren, Koustuv Sinha, Amirhossein Kazemnejad, Christos Christodoulopoulos, Ryan Cotterell, and Elia Bruni (eds.), *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.genbench-1.14.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2791–2809, Seattle, United States, 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.201.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, 2022b. Association for Computational Linguistics.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.779.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.527.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models, 2023.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. In

Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 11054–11070, 2021.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud D. Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. Parallel context windows for large language models. In *Annual Meeting of the Association for Computational Linguistics*, 2022.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 2009. doi: 10.1561/1500000019.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting, 2023.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. Mining discourse markers for unsupervised sentence representation learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3477–3486, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1351.

Woomin Song, Seunghyuk Oh, Sangwoo Mo, Jaehyung Kim, Sukmin Yun, Jung-Woo Ha, and Jinwoo Shin. Hierarchical context merging: Better long context understanding for pre-trained LLMs. In *The Twelfth International Conference on Learning Representations*, 2024.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

TogetherAI. Llama-2-7b-32k-instruct - and fine-tuning for llama-2 models with together api, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Szymon Tworkowski, Konrad Staniszewski, Mikoł aj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Mił oś. Focused transformer: Contrastive training for context scaling. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2023.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024.

Pawel Swietojanski Xingkun Liu, Arash Eshghi and Verena Rieser. Benchmarking natural language understanding services for building conversational agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, Ortigia, Siracusa (SR), Italy, 2019. Springer.

Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, Madian Khabsa, Han Fang, Yashar Mehdad, Sharan Narang, Kshitiz Malik, Angela Fan, Shruti Bhosale, Sergey Edunov, Mike Lewis, Sinong Wang, and Hao

Ma. Effective long-context scaling of foundation models, 2023.

Paiheng Xu, Fuxiao Liu, Zongxia Li, and Hyemi Song. Towards understanding in-context learning with contrastive demonstrations and saliency maps, 2023.

Howard Yen, Tianyu Gao, and Danqi Chen. Long-context language modeling with parallel context encoding, 2024.

Cecilia Ying and Stephen Thomas. Label errors in BANKING77. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pp. 139–143, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.insights-1.19.

LILI YU, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. MEGABYTE: Predicting million-byte sequences with multiscale transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 35–45, Copenhagen, Denmark, 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1004.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697–12706. PMLR, 2021.

Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Pose: Efficient context window extension of llms via positional skip-wise training, 2024.

A Saturation

One metric we are interested in is the point where the model performance *saturates*, which we define informally as the point where adding more examples is unlikely to meaningfully improve performance. More formally, we define the saturation point as the smallest number of examples tested such that performance reaches 95% of the model's maximum performance.

Saturation points vary by dataset. We define saturation as the first point at which performance reaches 95% of the model's maximum performance on that dataset. Table 3 shows the number of examples at saturation and the maximum number of examples that fit in the context window for each model. For datasets with larger label spaces, saturation generally occurs later; Banking-77 and Clinic-150 do not saturate within the context window of Llama2 (4096 tokens, which represents between 100-162 in-context examples for these datasets). In the longer-context regime, saturation points generally occur slightly later on Llama2-80k, but in both models occur before the model's maximum context length.

This suggests two things. First, given a fixed model, it is often not necessary to use the full context length to extract high performance from that model. Second, current models do not make use of the full potential of ICL; models often saturate in performance before the maximum number of examples, despite longer-context versions revealing that further performance improvements are possible.

The number of classes has some impact on saturation point– but is not fully explanatory. Our results show datasets with more classes benefit from more demonstrations in-context, on average, before saturation. This is to be expected, as the expected number of demonstrations necessary before seeing the correct label increases with the number of total label classes. To test if this is an intrinsic property of these datasets, or truly linked only to the number of label classes, we construct subsets of two high-label-space datasets, Banking-77 and Clinic-150, by randomly selecting half of the labels to exclude from the dataset. These subsets remain in the same domain, but with a smaller label space; if saturation is tied to the number of examples, then this should move the saturation point. Note that this is distinct from *combining* labels (e.g. TREC vs TREC-FINE), as combining finegrained labels into general labels makes the classification task simpler. It's still possible that the subset chosen is a simpler task (e.g. by removing one of a pair of frequently confused labels); to moderate the effect of this change, we average results over 3 randomly chosen subsets.

Table 4 compares the saturation point of the full- and half-label-space runs. For the datasets with the most number of labels, halving the number of labels also reduces the amount of examples that are useful before saturation, albeit not by half. However, the trend is less clear for the tasks with fewer labels; in some cases, reducing the label space actually *increases* the number of demonstrations before saturation. While the size of the label space clearly has some impact on the saturation point, more investigation is necessary to identify other factors impacting this behavior.

Dataset	Llama2	Llama2-32k	Llama2-80k	Mistral
TREC	20 (140)	100 (1129)	75 (2000)	50 (1129)
TREC-fine	75 (131)	250 (1056)	500 (2000)	500 (1091)
NLU	100 (162)	500 (1309)	500 (2000)	250 (1309)
Banking-77	- (100)	500 (838)	750 (1750)	500 (860)
Clinic-150	- (145)	750 (1169)	1000 (2000)	750 (1212)

Table 3: We measure the saturation point as the point at which the model reaches 95% of its maximum accuracy on the dataset; "-" in a column indicates that the maximum performance is achieved by using the full context window. The number in parenthesis represents the maximum number of examples that fit in the context window. As the label space of the dataset increases (from top to bottom row), so does the number of examples that can be used before saturation.

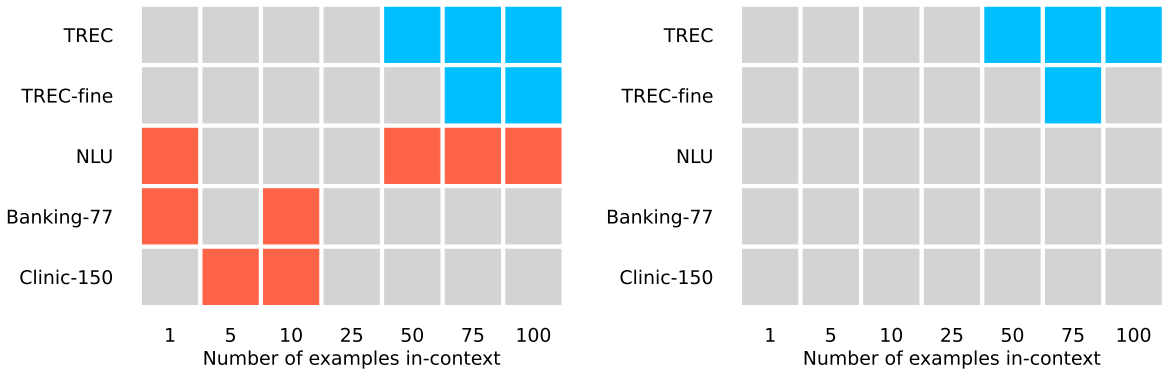
Dataset	Llama2	Llama2-32k	Llama2-80k	Mistral
TREC	14.29 / 24.69	8.86 / 1.77	3.75 / 1.71	4.43 / 6.0
TREC-fine	57.25 / 80.71	23.67 / 27.9	25.0 / 32.38	45.83 / 33.33
NLU	61.73 / 80.71	38.2 / 17.21	25.0 / 26.67	19.1 / 36.67
Banking-77	100.0 / 88.0	59.67 / 37.91	42.86 / 28.57	58.14 / 35.56
Clinic-150	100.0 / 64.04	64.16 / 35.14	50.0 / 22.86	61.88 / 56.67

Table 4: We compare the saturation point between the full-label-space (left) and half-label-space (right) for each model+dataset pair. Here we represent the label space as a percentage of the full context window.

B Using ICL as a testbed for long-context model properties

In this section, we use in-context learning as a testbed to examine several properties of long-context models.

How do long-context models perform in the short-context regime? Llama2-32k and Llama2-80k are finetuned variants of Llama2-7b, adapted for longer contexts. We evaluate how these models perform relative to the base model in short-context tasks (e.g. ICL using less than 4096 tokens of demonstrations) by testing whether the difference in performance is statistically significant (2-sided t-test, $p < 0.05$). Performance is generally similar, with some areas of slight improvement from the base model; Figure 10 shows full results. We observe degradation in performance in some settings for Llama2-32k, highlighting the importance of testing for behavior regression when finetuning for additional capabilities.



(a) Llama2-32k

(b) Llama2-80k

Figure 10: Short-context behavior of long-context models. Each model's performance is compared to the performance of the base model it was finetuned from, on the same amount of data. **Red** represents significantly worse performance; **blue** represents significantly better performance ($p < 0.05$).

.

Input utilization We analyze the performance of all methods on a naturalistic needle-in-the-haystack (Ivgi et al., 2023; Liu et al., 2024) style test. If the model is effectively using the context, then it should be able to exactly recover the label for any example it has seen in-context. Note that, while a model trained on some set of data is not uniformly capable of exact copying from that training data (Biderman et al., 2023), in nearly all of our finetuning runs, the model fits the training data with 100% accuracy.

We examine this behavior by selecting the same set of examples to use in-context and in evaluation; all models should then be able to achieve 100% accuracy. Table 5 shows the results; while all models achieve very high accuracy on the copied data, no model is able to uniformly copy correctly from the input. Surprisingly, performance improves slightly with additional demonstrations for most models, possibly due to additional specification of the task.

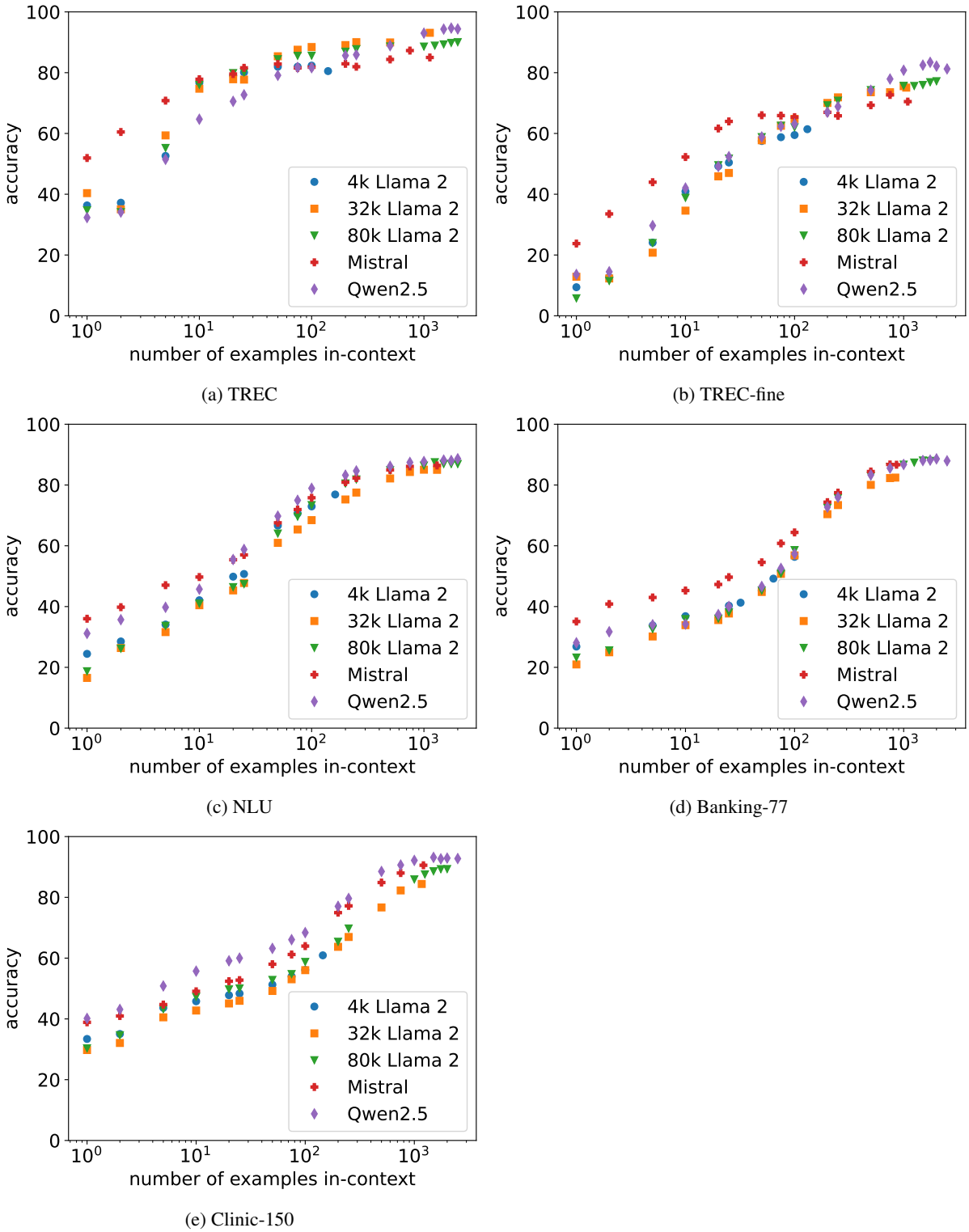
Number of examples	1	5	10	25	50	100	200	250
Llama2	**100.0**	93.0	96.5	97.0	98.6	97.95	-	-
Llama2-32k	80.0	95.0	97.0	96.6	98.4	98.5	98.5	**98.9**
Llama2-80k	90.0	94.0	95.0	98.2	**98.5**	98.3	98.0	97.9

Table 5: Copying behavior given the test examples in the context window. Results are averaged over Banking-77 and Clinic-150; bold indicates the best performance for that model.

C Full ICL results across datasets

For space, we show 1-2 representative datasets for each point of analysis in the paper. In this appendix, we present results across all classification datasets.

C.1 Random selection ICL across all models



(a) TREC

(b) TREC-fine

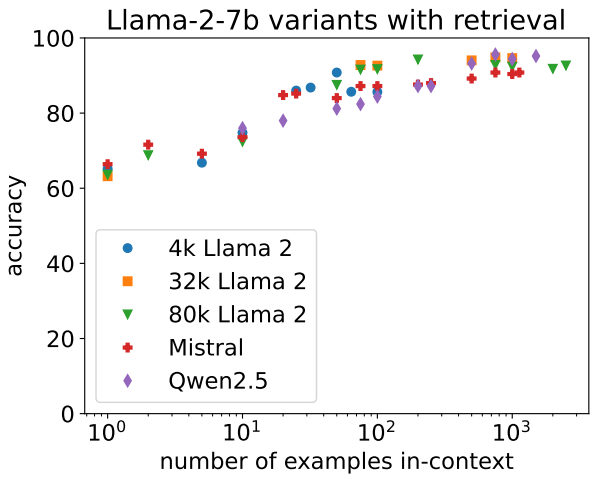
(c) NLU

(d) Banking-77

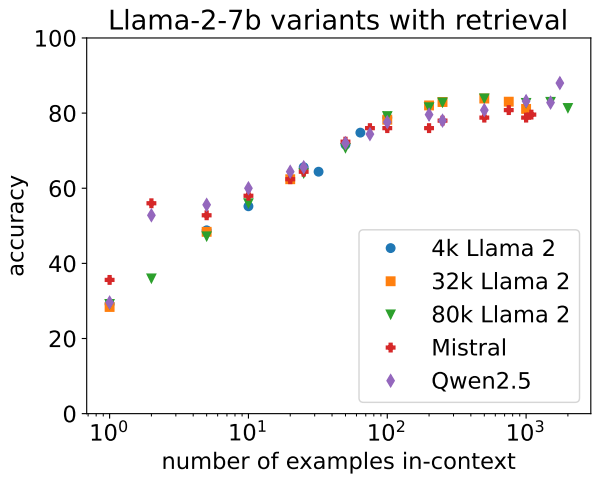
(e) Clinic-150

Figure 11: Performance of random-selection ICL across all models for each classification dataset. Performance continues to increase with additional examples in-context.

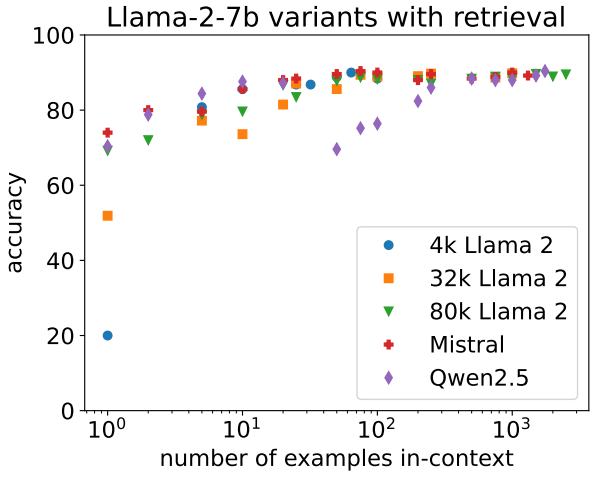
C.2 Retrieval ICL across all models



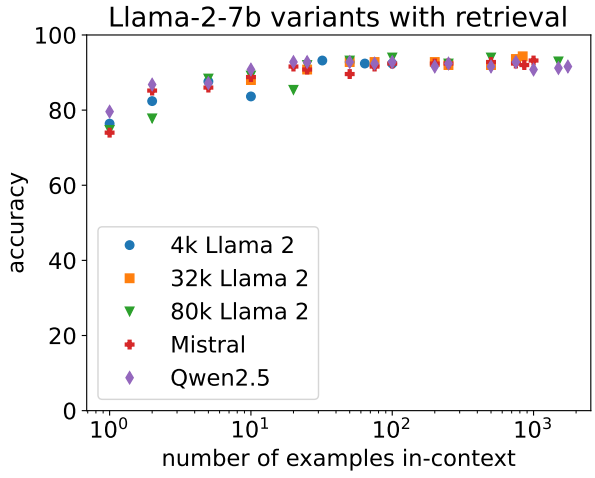
(a) TREC



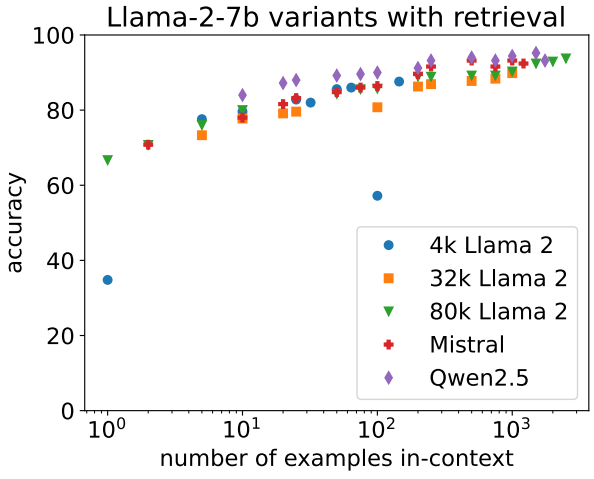
(b) TREC-fine



(c) NLU



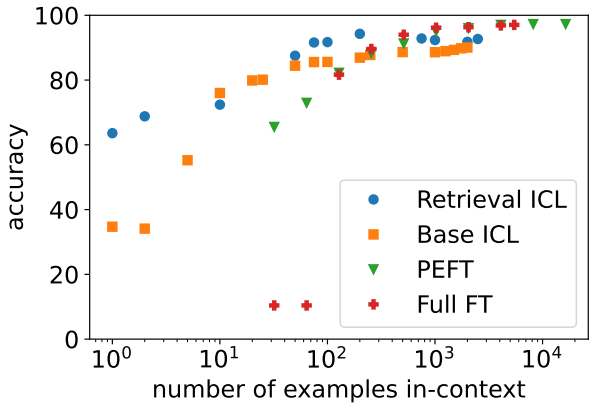
(d) Banking-77



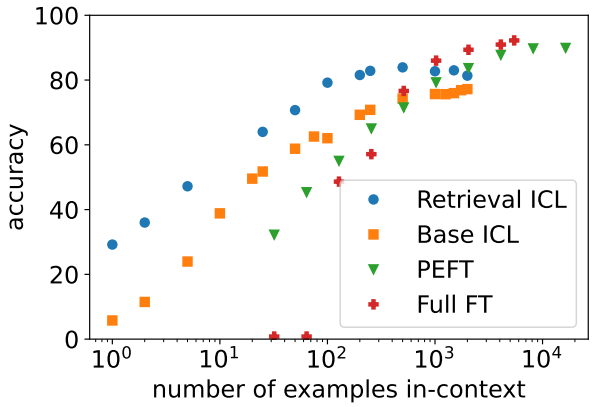
(e) Clinic-150

Figure 12: Performance of retrieval-based ICL across all models for each classification dataset. Short-context performance here is higher than for random-selection, but performance continues to improve with more examples until a saturation point, where performance flattens out.

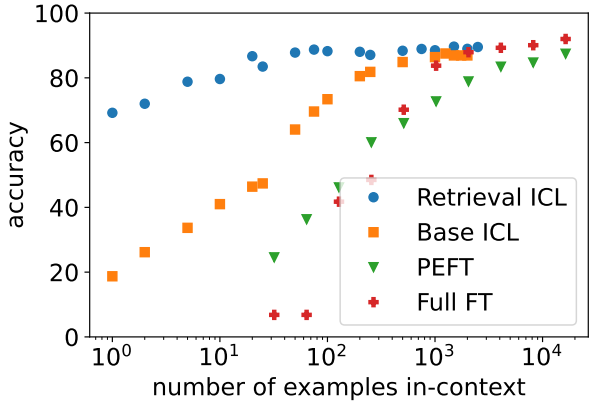
C.3 Comparing retrieval, random selection, and finetuning



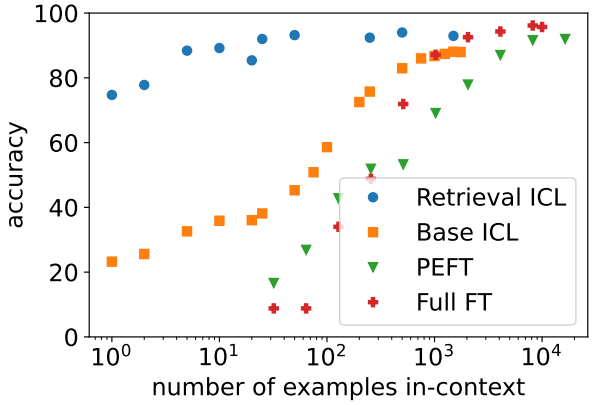
(a) TREC



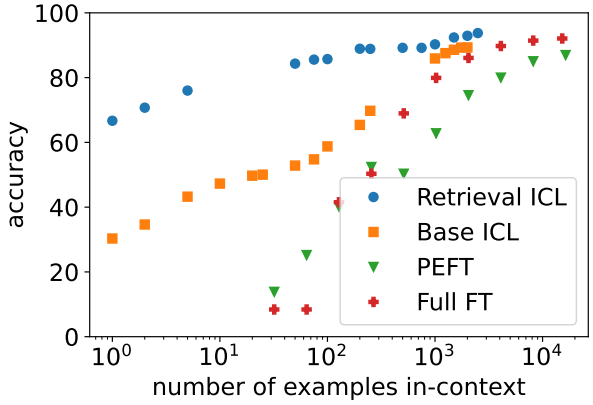
(b) TREC-fine



(c) NLU



(d) Banking-77

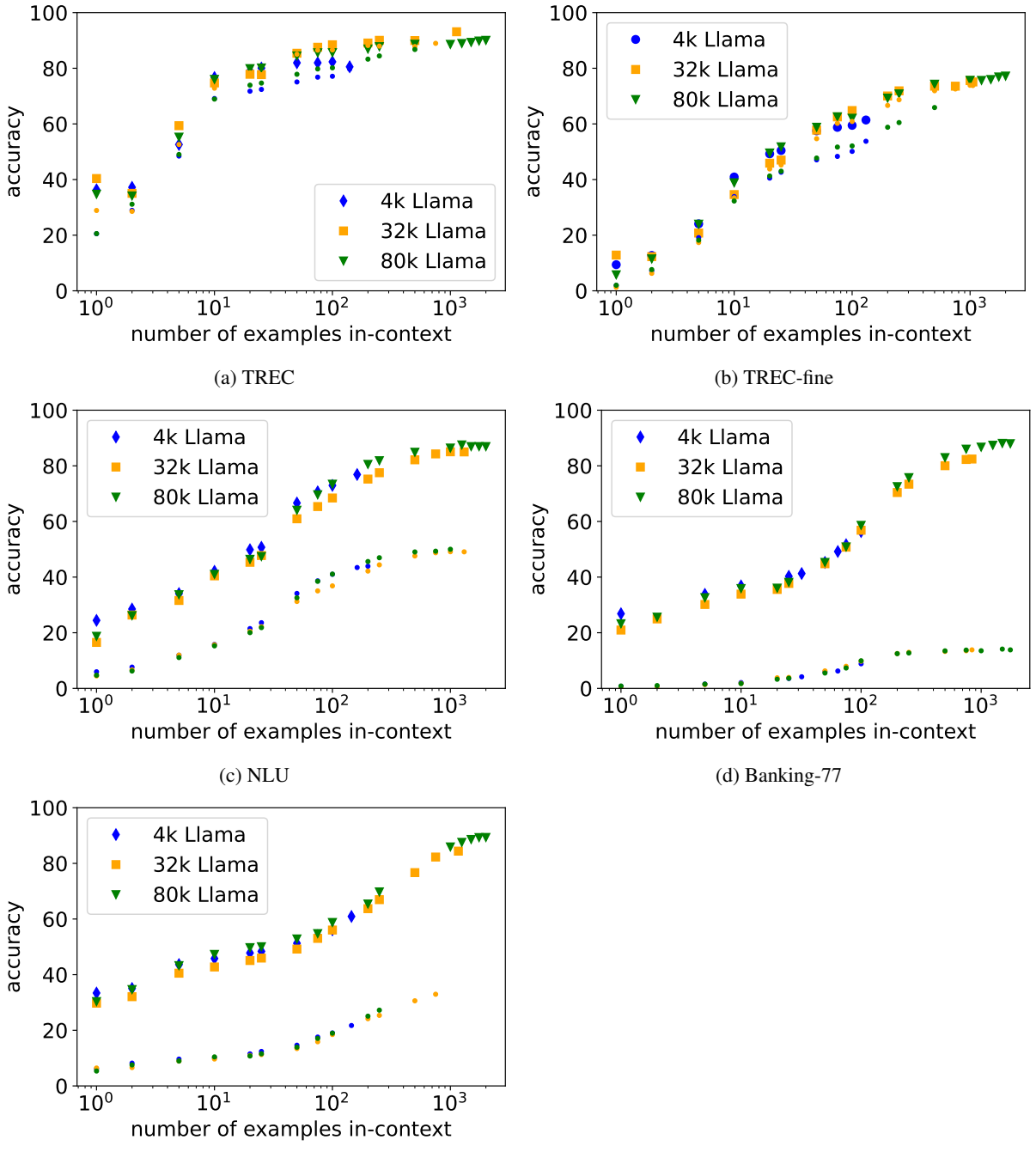


(e) Clinic-150

Figure 13: Performance of retrieval-based ICL, random-selection ICL, and finetuning across 5 datasets. At small example counts, ICL outperforms finetuning; when several thousand examples are used, finetuning outperforms ICL in some datasets.

D Constrained decoding details

To perform constrained decoding, following Ratner et al. (2022), we add a large constant to all logits at each decoding step that could result in a valid label being generated. This strategy is generally not possible for API-access models, or when the label space is not fully known in advance. To determine whether the same trends hold without the use of constrained decoding, we evaluated the Llama-2 family models with and without constrained decoding. Figure 14 shows the comparison; while performance is lower, especially in the higher-label-space datasets, the general trends hold. In Banking77, it appears that performance begins to saturate slightly earlier with unconstrained decoding; we hypothesize that this may be due to more specialized language used in banking domain labels.



(a) TREC

(b) TREC-fine

(c) NLU

(d) Banking-77

(e) Clinic-150

Figure 14: Comparison of constrained decoding (diamond, square, and triangle datapoints on the graph) with the unconstrained decoding variants (dot-like datapoints of the same colors).

E Finetuning

We performed finetuning with HuggingFace transformers (Wolf et al., 2020). To perform parameter efficient finetuning (PEFT), we used the peft (Mangrulkar et al., 2022) package (version 0.9.0). We finetune the model for 30 epochs, evaluating it every epoch on the test set, and ultimately choosing the checkpoint with the highest test accuracy. We note that using the test set to perform model selection presents an unfair advantage to finetuning (compared to ICL) and may not be truly indicative of the generalization error. However, doing so provides the advantage of being both comparable to ICL in terms of the data used, as well as giving an upper bound on the true generalization accuracy of the finetuned model, further emphasizing any observed efficacy gap between it and ICL.

Initialization of the classification head While in our default setting we initialize the classification head from the pretrained LM head, subsampled at the representation of the first token in each label, we investigate the efficacy of this approach by contrasting with a randomly initialized classification head. Figure 15 shows that while in the few-shot regime, this approach has significant advantage, as the training set grows in size the difference shrinks to become negligible. In no case was random initialization better than this approach.

Hyperparameter tuning To remain comparable in terms of compute efficient finetuning, we did not perform extensive hyper-parameter tuning per task, and instead experimented with a good global setting on a single dataset (Banking-77). Specifically, we experimented with different learning rates, different LoRA ranks (r) and α (Hu et al., 2022) and also tried applying RSlora (Kalajdzievski, 2023) which sets the scaling factor to $\frac{\alpha}{\sqrt{r}}$ as some evidence suggest it can outperform the original method. Figure 16 summarizes the results, depicting average test accuracy against training examples with different settings.

Ultimately, we found that using HuggingFace's (Wolf et al., 2020) default parameters of $r = 8$, $\alpha = 32$, LoRA dropout of 0.1 and a learning rate of $1e-3$ to work best. In all cases, we used batch sizes of 32 and weight decay of 0.01.

It is possible that methods specialized for finetuning in small-data regimes, such as T-few (Liu et al., 2022), might close the gap between ICL and PEFT in the small-data regimes. We did not consider T-few in our analysis because of its additional pretraining stage, which imposes substantial additional cost, and because T-few was developed with a focus on encoder-decoder models and we consider only decoder-only models in our setting.

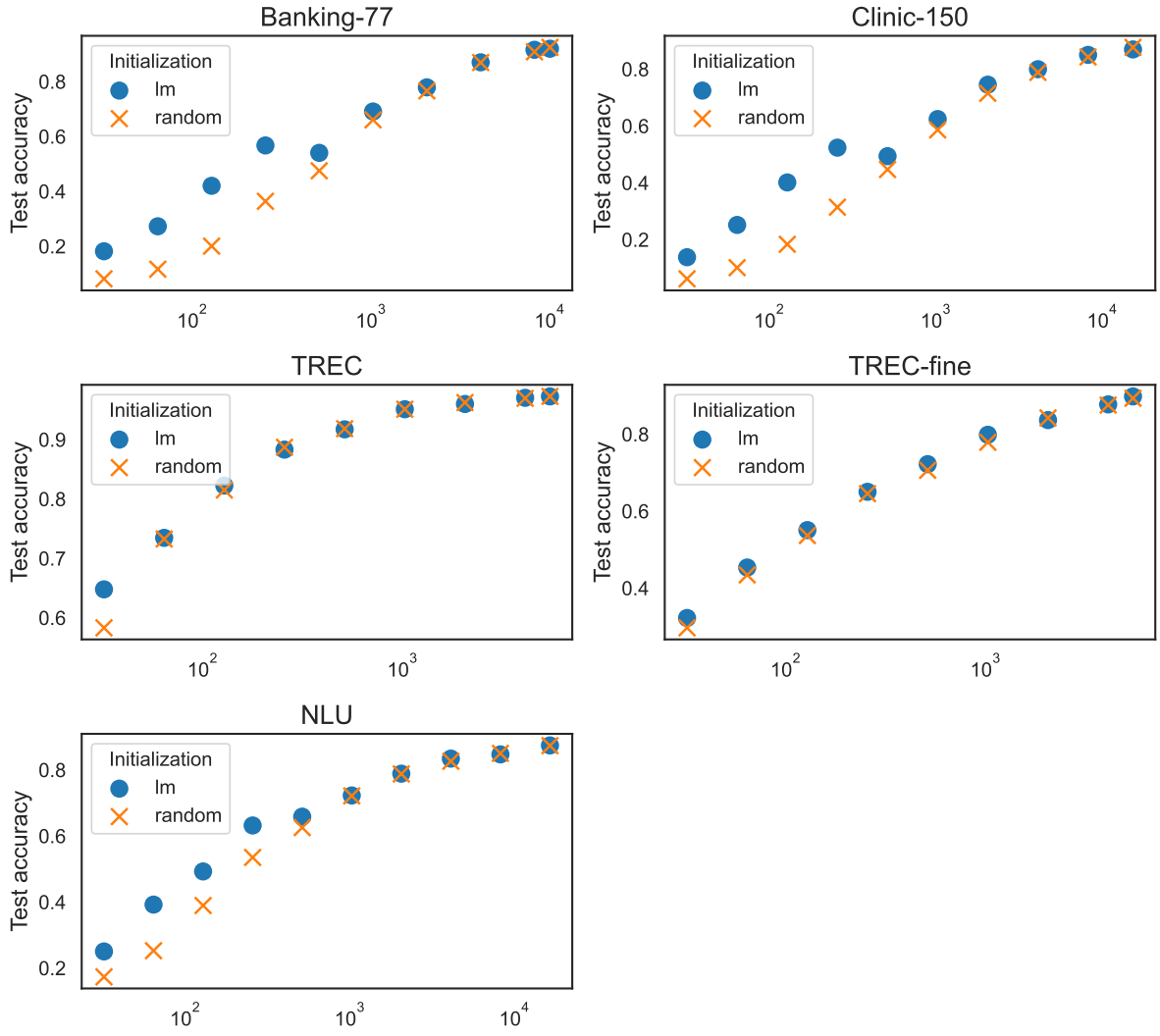


Figure 15: Comparing initialization methods of the classification head when finetuning a PEFT llama-2-7b model. Averaged (best) test accuracy over 5 random seeds. Initialization with *lm* subsamples the pretrained language-modeling head at the first token of the target label, while random samples random weights.

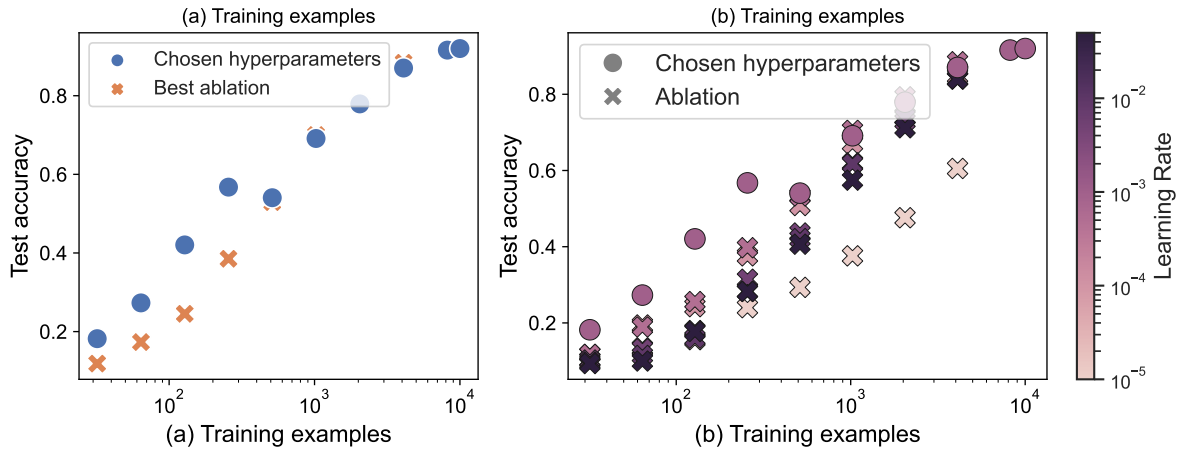
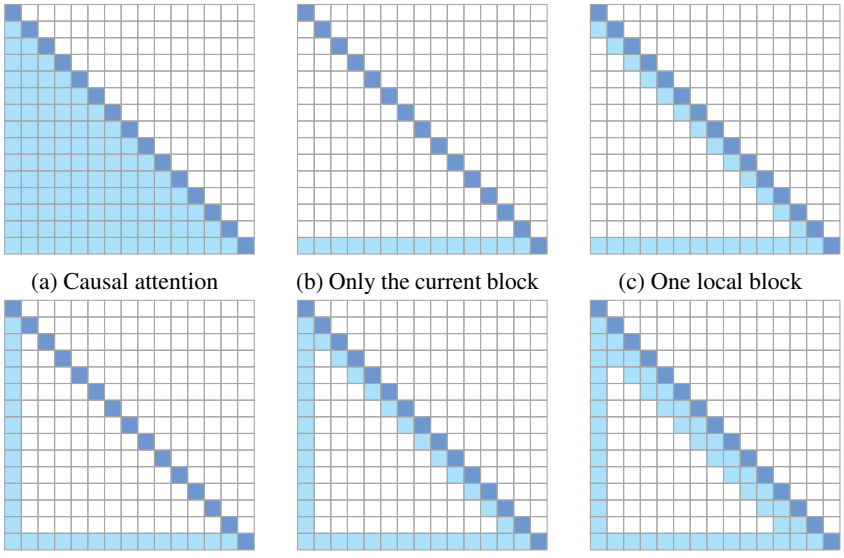


Figure 16: Comparing hyperparameters when finetuning a PEFT llama-2-7b model on Banking-77. Averaged (best) test accuracy over 3 random seeds. (a) Comparing our fixed LoRA configurations to the best alternative configuration (at each scale) we tried. (b) Comparing different learning rates.

F Block attention patterns

We aim to construct a modified attention pattern where demonstrations can only attend to a limited block of local demonstrations.

The most naive strategy is to only allow attention within a local block (Figure 17b). Other variants relax the mask to allow attention to the first block, which acts as an attention sink (Xiao et al., 2024) (Figure 17d), or allowing attention back to the immediately prior local block(s) (as seen as part of Acharya et al. (2024))(Figure 17c).



(a) Causal attention (b) Only the current block (c) One local block

(d) Attention sink (e) StarAttention (f) Our chosen attention

Figure 17: Representative examples of attention masks we considered. In each diagram, the squares represent a block of b examples each, *not* individual tokens.

As a test of these methods, we perform block attention with each attention mask on Banking-77 with Llama2-80k, with the block size $b = 50$ and 500 demonstrations in-context.

Diagram	Block size b	Sink blocks	Local blocks	Accuracy
Figure 17a	500 (full causal)	-	-	80.04
Figure 17b	50	0	0	18.72
Figure 17c	50	0	1	51.16
Figure 17d	50	1	0	62.15
Figure 17e	50	1	1	75.12
Figure 17f	50	1	2	**77.51**

Table 6: Comparison of attention strategies and performance.

Neither a sink block nor local attention alone recovers close to full attention performance, although both are substantial improvements over the naive strategy. The best performance is achieved with both sink and local blocks; although one local block is sufficient to recover 94% of the performance of the full attention method, we choose two local blocks to minimize performance degradation. As a result, we use the attention pattern in Figure 17f for the experiments in the paper.

G Prompt formatting and examples from datasets

As a demonstration of the datasets, we provide an example of 3-shot prompting for each dataset with the prompt formatting we used (and with examples drawn from the training set of each dataset).

Prompt formatting and instruction phrasing can have significant impact on performance (Sclar et al., 2023); we keep the formatting consistent with prior work (Ratner et al., 2022), with prefixes for the input and output for each exemplar. Because we use predominately non-instruction-tuned models, we do not add an additional instruction or system prompt.

G.1 TREC

TREC (Hovy et al., 2001; Li & Roth, 2002) is a question classification dataset with two granularities of labels. We refer to the 6-label coarse classification as TREC.

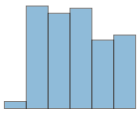


Figure 18: The label distribution of TREC. One label is much less frequent than the rest.

Question: How does light travel through the void of space if there is no medium for it to ' wave ' or ' pulse ' .
Type: description
==
Question: What cathedral was Thomas Becket murdered in ?
Type: location
==
Question: The lawyer who represented Randy Craft , what was his name ?
Type: human
==
Question: What are the rites accompanying the circumcision of a newly born-child in Judaism called ?
Type:

Figure 19: Example 3-shot prompt for TREC.

G.2 TREC-fine

We refer to TREC's 50-label finegrained classification as TREC-fine.

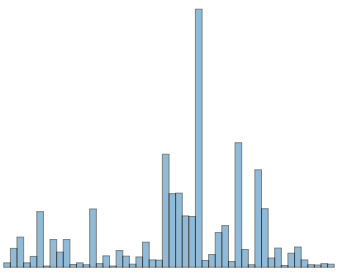


Figure 20: The label distribution of TREC-fine.

Question: How does light travel through the void of space if there is no medium for it to ‘ wave ’ or ‘ pulse ’ .
Type: description manner
==
Question: What cathedral was Thomas Becket murdered in ?
Type: location other
==
Question: The lawyer who represented Randy Craft , what was his name ?
Type: human individual
==
Question: What are the rites accompanying the circumcision of a newly born-child in Judaism called ?
Type:

Figure 21: Example 3-shot prompt for TREC-fine.

G.3 NLU

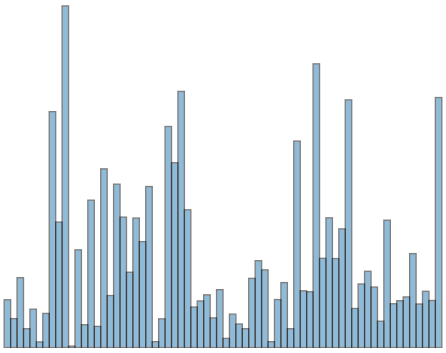


Figure 22: The label distribution of NLU.

NLU (Xingkun Liu & Rieser, 2019) is a 68-way intent classification dataset in the conversational domain. The original paper evaluates on 64 of the intents; we use all 68. The data is licensed under CC BY 4.0.

utterance: oh it is nice one, olly.
intent: general praise
==
utterance: nope wrong.
intent: general negate
==
utterance: what events near hear are happening this week
intent: recommendation events
==
utterance: play fishing podcasts that are favorited
intent:

Figure 23: Example 3-shot prompt for NLU.

G.4 Banking-77

Banking-77 (Casanueva et al., 2020) is a 77-way intent classification task in the financial domain. Although the accuracy of some labels in BANKING77 has been criticized (Ying & Thomas, 2022), we

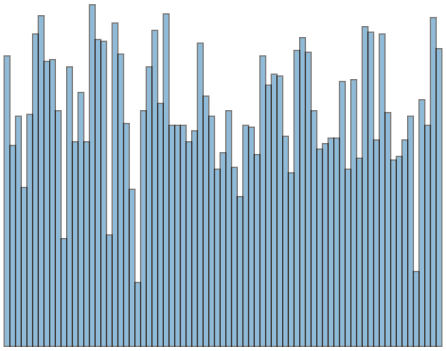


Figure 24: The label distribution of Banking-77.

report results here on the original dataset for consistency with prior work. The data is licensed under CC BY 4.0.

query: How long will my payment be pending?
intent: pending card payment
==
query: My physical card is not working
intent: card not working
==
query: i cant seem to activate card
intent: activate my card
==
query: I didn't set up a direct debit payment on my account.
intent:

Figure 25: Example 3-shot prompt for Banking-77.

G.5 Clinic-150

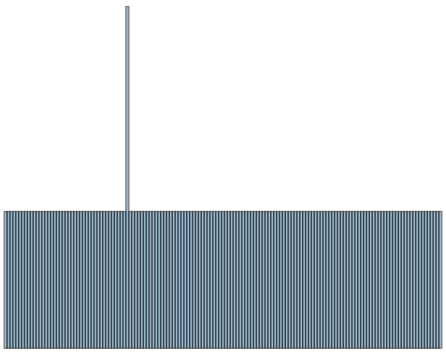


Figure 26: The label distribution of Clinic-150. It is balanced except for the "out of scope" label, which has additional data points in the split we use.

Clinic-150 (Larson et al., 2019) is a 151-way, multi-domain intent classification task; examples are either labeled with an intent from one of 10 domains or with the catch-all "out-of-scope" label. We use the "plus" train split from the original paper, which adds additional "out-of-scope" examples to the dataset. The data is licensed under CC BY 3.0.

utterance: how much is my comcast bill
intent: bill balance
==
utterance: tell me about yourself
intent: what is your name
==
utterance: how to build up my credit score
intent: improve credit score
==
utterance: are you employed by me
intent:

Figure 27: Example 3-shot prompt for Clinic-150.

G.6 SAMSum

SAMSum (Gliwa et al., 2019) is a text message summarization corpus; both the text conversations and summaries were written by linguists. The conversations can be two or more participants, and may contain emojis, emoticons, or tags that indicate the use of an image or gif; the summaries are generally 1-3 sentences long. The data is licensed under CC BY-NC-ND 4.0.

Conversation: Zara: <file_gif>
Zara: something terrible happened :(
Stanley: what? You OK?
Zara: yes, I'm fine! I went to the swimming pool
Zara: and lost the earring you gave me for birthday :(
Stanley: oh Jesus, I thought something bad happened to you!
Summary: Zara has lost the earring, which Stanley gave her for birthday, when she was at the swimming pool.
==
Conversation: Rose: hey congratulations for the baby boy! i wish him all the health and happiness in life.
Ela: thank you so much for the wishes.
Rose: your welcome.. so hows he and you? all ok
Ela: yes everything is great thanks
Rose: will come to see you and the baby boy
Ela: Sure will wait to see you..
Summary: Ela just gave birth to a boy. Rose will visit them shortly.
==
Conversation: Kim: Are you going to the conference in SF?
Jenny: I should, I know, it would be good for my career
Jeff: no, not so much, I think it's bullshit that it's so important
Simon: is it?
Jeff: sure, the whole net-working thing doesn't really matter, I think
Jeff: nobody offers you a job at a conference
Jeff: and it costs so much to fly to SF
Kim: I would like to go also to see what's going on in the field Kim: to meet people, see new trends, ideas
Kim: I think it's important for an academic
Jeff: this may be true, if you can afford
Kim: the flight is about €500, right?
Simon: true
Jeff: and then more money for accommodation
Jeff: it can easily pile up to €2000
Kim: you're quite right, unfortunately
Jeff: because it also doesn't make sense to fly to California for 3 days
Jeff: it would be also extremely disturbing, with the jet lag etc.
Kim: you're so right :(
Jeff: so think about it first
Summary: Jeff is not going to the conference in SF. The flight is expensive.
==
Conversation: Amanda: I baked cookies. Do you want some?
Jerry: Sure!
Amanda: I'll bring you tomorrow :-)
Summary:

Figure 28: Example 3-shot prompt for SAMSum.

H Additional details

H.1 Models selected

The majority of the analysis in the paper concerns these five models:

1. **Llama2** (Touvron et al., 2023) is a decoder-only model trained with a 4096 context length. We use the non-instruct (non-chat) variant because it is more commonly used as a base model for long-context finetuning and because we observed very similar performance between the chat and non-chat variants in our initial experiments.
2. **Llama2-32k** (TogetherAI, 2023) is a version of Llama-2-7b finetuned by TogetherAI for a 32k context window. We use the non-instruct version.
3. **Llama2-80k** (Fu et al., 2024) is a version of Llama-2-7b finetuned with 80k context and a carefully designed long-document data mixture.
4. **Mistral-7b-v0.2** (Jiang et al., 2023). is the instruct version of Mistral-v0.2 (the non-instruct model is not publicly available). The trained context length of Mistral-7B-Instruct-v0.2 is 32k tokens.
5. **Qwen2.5-7B** (Team, 2024) is a decoder-only model with multilingual support. The trained context length of Qwen2.5-7B is 128k tokens.

While all of these models can extrapolate to inputs longer than their trained context length, we restrict the lengths of inputs to fit within the trained context length; this represents the best case performance without the additional confound of the extrapolation strategy.

H.2 Computational cost

For our finetuning experiments, we used approximately 50 GPU-days of compute on 80GB A100 GPUs. The computational cost of the in-context learning experiments was approximately 75 GPU-days, primarily using 48GB A6000 GPUs. All experiments were run on local clusters (i.e. not using cloud providers), with the exception of the frontier models experiments, which used Together AI's API (for Llama 3.1-405B) and the Anthropic API (for Claude).