On the generalization of language models from in-context learning and finetuning: a controlled study

Andrew K. Lampinen*
Google DeepMind
Moutain View, CA
lampinen@google.com

Arslan Chaudry*
Google DeepMind
Moutain View, CA
arslanch@google.com

Stephanie C. Y. Chan*
Google DeepMind
San Francisco, CA
scychan@google.com

Cody Wild
Google DeepMind
Moutain View, CA

Diane Wan
Google DeepMind
Seattle, WA

Alex Ku
Google DeepMind
Mountain View, CA

Jörg Bornschein
Google DeepMind
London, UK

Razvan Pascanu
Google DeepMind
London, UK

Murray Shanahan
Google DeepMind
London, UK

James L. McClelland
Google DeepMind & Stanford University
Mountain View, CA

Abstract

Large language models exhibit exciting capabilities, yet can show surprisingly narrow generalization[2] from finetuning. E.g. they can fail to generalize to simple reversals of relations they are trained on, or fail to make simple logical deductions based on trained information. These failures to generalize factual information from fine-tuning can significantly hinder the reasoning capabilities of these models. On the other hand, language models' in-context learning shows different inductive biases and deductive reasoning capabilities. Here, we explore these differences in generalization and deductive reasoning between in-context- and fine-tuning-based learning. To do so, we constructed several novel datasets to evaluate and improve models' abilities to make generalizations over factual information from novel data. These datasets are designed to create clean tests of generalization, by isolating the knowledge in the dataset from that in pretraining. We expose pretrained large models to controlled subsets of the information in these datasets — either in context, or through fine-tuning — and evaluate their performance on test sets that require various types of generalization. We find overall that in data-matched settings, in-context learning can generalize several types of inferences more flexibly than fine-tuning (though we also find some qualifications of prior findings, such as cases when fine-tuning can generalize to reversals embedded in a larger structure of knowledge). We build on these findings to propose a method to enable improved generalization from fine-tuning: adding in-context reasoning traces to finetuning data. We show that this method improves generalization and deductive reasoning across various splits of our datasets and other benchmarks. Our results have implications for understanding the different forms of reasoning afforded by different modes of learning in language models, and practically improving their performance.

*Equal contribution.

[2]By generalization we mean simple logical deductions of the trained knowledge which is the fundamental aspect of reasoning.

39th Conference on Neural Information Processing Systems (NeurIPS 2025). FoRLM Workshop.

Introduction

Language models (LMs) pretrained on huge corpora of internet text become efficient in-context learners; they can generalize from a few examples of a task to answer new instances (Brown et al., 2020; Gemini Team Google, 2023). Pretrained LMs can also be fine-tuned for downstream tasks using relatively few examples—though achieving good generalization from fine-tuning often requires hundreds, or even thousands of examples (e.g. Kirstain et al., 2022; Vieira et al., 2024). Indeed, the generalization from fine-tuning on a particular example can be surprisingly limited; for example, LMs fine-tuned on a statement like "B's mother is A" fail to generalize to answer "Who is A's son?" (Berglund et al., 2024; cf. Allen-Zhu and Li, 2025). These failures of simple logical deduction of finetuned knowledge can hinder reasoning capabilities of LMs. However, LMs can readily reason through questions about these types of reverse relations in context (e.g. Lampinen et al., 2024b). Furthermore, transformers show different inductive biases when generalizing from weights vs. from context (cf. Chan et al., 2022b; Russin et al.; Shen et al., 2023). How do patterns of generalization differ between in-context reasoning and fine-tuning? What are the implications for how we should adapt models for new tasks or information? In this paper, we explore these questions.

To do so, we construct controlled synthetic datasets of factual knowledge. We design these datasets to have complex and self-consistent structures, while avoiding any overlap with knowledge that may be present in pretraining corpora. We create train and test splits of these datasets that involve different types of generalization: reversals or chaining multiple logical inferences into a syllogism. We then evaluate how well large pretrained language models generalize to these test sets through fine-tuning, or through in-context learning—via placing the entire training set (or large subsets thereof) in context. These results show that models can use long contexts effectively to generalize better than gradient-based learning. We also explore various methods of improving generalization, such as data augmentation. Overall, we find that across various datasets, in-context learning (ICL) generalizes better than finetuning. However, finetuning generalization can be improved, and in fact can even exceed ICL generalization, by spending more train time compute to augment the training dataset using in-context reasoning.

Our contributions are as follows:

- We study the distinct patterns of factual generalization that pretrained LMs exhibit from in-context learning and finetuning.
- We find that in-context learning over the entire training dataset often generalizes better than finetuning, when evaluated on systematic holdouts like reversals, syllogistic inferences, and other logical deductions.
- We propose to bridge this gap via dataset augmentation—prompting an LM to reason about the data in-context, and then adding these reasoning traces to the training set.
- We show that dataset augmentations can bridge the gap to yield improved generalization from fine-tuning.
- We also propose a fine-tuning method that breaks up correlations among sentences, amplifying the benefits of augmentation.

Datasets

We evaluate learning and generalization across several datasets that are crafted to isolate different aspects of generalization, or to situate it within a broader set of learning challenges. We also draw on datasets from prior work.

Reversal curse paper

We first use the reversal dataset proposed by Berglund et al. (2024) that contains one-sentence descriptions of fictional celebrities. Dataset examples can have the celebrity name (e.g. 'Daphne Barrington') preceding the description (e.g. 'the director of "A Journey Through Time."') or vice-versa. The dataset includes two independent sets of celebrities – 'A' and 'B'. Following Berglund et al. (2024), we finetune the model on celebrities from the set 'A' with names preceding the description, and from set 'B' where names and descriptions appear in both orders but in separate examples.

Overall, the train set consists of 3600 examples. The test set evaluates whether the model can infer the celebrity names from the set 'A' given only their descriptions. To add distractors in test examples, we include three randomly selected incorrect celebrity names in the list of options to score.

Simple reversals and syllogisms

We then test our methods with two simple datasets[3] that contain independent examples of reversal and syllogistic reasoning.

Simple reversals: Each training example consists of a brief preamble (e.g. "Did you know that") followed by a single statement comparing two entities, e.g. "femp are more dangerous than glon." There are a hundred such comparisons provided (with the comparison words sampled from a set of 28, e.g. "brighter", "heavier,"). Each comparison is repeated across 10 different training examples, each time paired with a randomly sampled preamble. The test set consists of forced choices between the correct reversal and a contradictory relation, e.g.: "glon are less/more dangerous than femp." Note that the incorrect answer is a tempting alternative, as it uses the same words as the trained statement (just reordered), which means that a model relying on simple features like bag-of-words would tend to prefer the incorrect answer.

Simple syllogisms: There are 69 training examples. Each example consists of a preamble (which specifies the entities under discussion), followed by two statements that form a syllogism. For example: "The relationships between glon, troff and yomp are: All glon are yomp. All troff are glon." The test examples evaluate whether the model makes a correct inference from that syllogistic form. They provide a preamble specifying two entities from the training example, and then score all possible statements about the relationship between the two entities involving the quantifiers and relations in the dataset. Following Lampinen et al. (2024b), we omit the partial negative form (some X are not Y); thus, there are six such possible statements (the product of "all", "some", and "no" together with the two directions each of those relations could take). For example, the preamble for the test example corresponding to the syllogism above would be "The relationships between yomp and troff are:" and the correct answer would be "All troff are yomp."

Semantic structure benchmark

The benchmark is built around a relational semantic hierarchy that allows deductive inferences and abstractions. This hierarchy is based on real-world categories and relations, and draws inspiration from prior work in cognitive science that has studied learning of rich semantic hierarchies (see Fig. 1 for an intuitive example that composes part of the real-world structure). We similarly create a hierarchical knowledge structure involving 110 categories of animals and objects, and their properties and relations. Unlike the original structure, it is not balanced in branching factors or depth, and properties are similarly asymmetrically distributed across entities (1-6 per category, plus inherited ones)—making its distributional statistics more naturalistic (see Appx. A.2 for details). This underlying structure is derived from the real world.

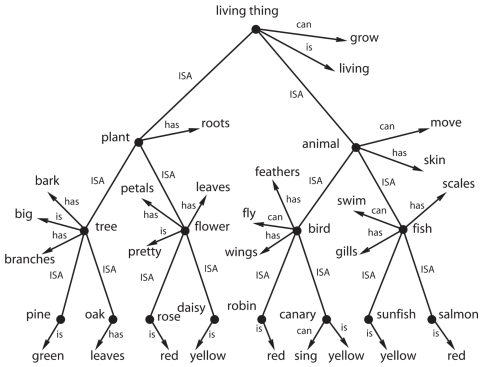

Figure 1: A semantic structure with a hierarchy of properties and relations. (Reproduced with permission from Rogers and McClelland (2008).)

In order to make the structure novel to the pretrained models, however, we replace all nouns, adjectives, and verbs with nonsense terms. This removes the potential overlap with pretraining data, thus ensuring that the data properties follow certain real-world distributional features, but are not contaminated.

The use of nonsense terms potentially poses tokenization challenges; however, we ameliorate these by generating short 4-5 letter nonsense words using plausible combinations of phonemes for English

[3]These datasets are adapted from the nonsense NLI and Syllogism datasets from Lampinen et al. 2024b.

(sampled via Gao et al., 2023). Moreover, we show in the next sections that in practice the models can readily make inferences about these nonsensical entities in context, so use of unfamiliar words is not the primary bottleneck on model performance.

Train set: For training, we assemble train-set facts about this semantic hierarchy into short vaguely-Wikipedia-like synthetic articles with varied formatting and styles, as well as some QA examples (to ensure that fine-tuning on the data doesn't degrade question answering capabilities). We ensure that all facts that are necessary for the test questions (below) are presented in at least one training document. However, some facts will be redundantly presented across multiple documents. We create a total of 2200 training documents of 4-20 sentences in length.

Test sets: Within the semantic structure, our test sets accommodate reversals of relations (gruds [dogs] are a type of abmes [mammals] => abmes include gruds; cf. Berglund et al., 2024), syllogism-like deductive inferences (e.g. gruds [dogs] are abmes [mammals]; abmes are rony [warm-blooded] => gruds are rony), and longer deductions. Specifically, we focus on the following splits (in rough difficulty order):

- *Rephrasings* of trained facts that don't change the direction of wording: used for validation that the material is learned.
- *Reversals* of trained facts, i.e. the two entities are in reverse order, and the relational words are adjusted accordingly.
- *Syllogisms* over trained facts; that is, statements that can be logically inferred by linking two (implicitly) quantified statements in the training corpus, using an entity that overlaps between them.
- *Category holdouts*: only one fact about a category is present in training: what its parent category is. All possible inferences from that fact are tested. This overlaps with some aspects of the syllogism splits, except that: 1) the information about the target category is more strictly limited, thus limiting other cues that could aid generalization; and 2) inferring the fact may require chaining more than two statements together.

When creating the evaluation questions, we choose difficult distractors for the incorrect answers, by choosing entities or properties that have the target relationship to a *different* entity in the dataset. For example, if the correct answer is "gruds are rony" one of the distractors might be "gruds are zept" where there is some other valid statement in the dataset such as "telk are zept." Thus, it is not possible to guess simply by the local context of the words which answer is correct.

We evaluate the models both with and without few-shot QA examples taken from the validation set in the prompt; we find relatively similar results across these conditions so we collapse across them in our main analyses.

Methods

Evaluation

We perform evaluation using multiple-choice likelihood scoring. We do not provide the answer choices in context. We generally construct the datasets to provide challenging alternative answers (see above).

In-context evaluation: To perform full-dataset in-context evaluation, we concatenate the documents in a training dataset and provide them as context to the (instruction-tuned) model. We then prompt the model to use that context to answer the question. We randomly subsample the documents by a factor of 8x when evaluating models in-context on the largest datasets, as there is some redundancy in the datasets, and we find that models experience interference as the context length scales.

Fine-tuning

Our fine-tuning experiments mainly involve tuning Gemini 1.5 Flash (Gemini Team Google, 2024) on our datasets. We generally fine-tune with batch size 8 or 16 and learning rate $3 \cdot 10^{-4}$ for 200-1000 steps (depending on dataset size and loss).

Dataset augmentation

Our key approach to dataset augmentation is to take advantage of the generalization of models during in-context reasoning in order to improve the coverage of the fine-tuning dataset. We approach this via several methods, but all chiefly aim at the goal of spending training time compute for in-context inference to create more finetuning data, in order to improve generalization out-of-context (i.e., without requiring additional information in context) at test time.

Specifically, we consider two types of augmentation: a local strategy that tries to increase the flexibility with which particular pieces of information can be used, and global strategies that attempt to relate between different pieces of information. Each of these strategies uses distinct context and prompts (Appx. A.1).

Local (sentence) augmentation: We prompt an LM to augment each training data point (e.g. sentence) to enhance the flexibility with which the model encodes it. Our prompt includes examples of rephrasings and reversals.

Global (document) augmentation: We concatenate the full training dataset as context, then provide a particular document and prompt the model to generate inferences by linking between that document and the rest of the documents in context. This results in a longer reasoning trace of relevant inferences.

Sentence-splitting

Some datasets, such as the fictional celebrities dataset by Berglund et al. (2024) and our semantic structure dataset, contain documents comprising multiple sentences that are logically or semantically linked. We find that splitting these documents at sentence-level, into multiple finetuning examples, leads to a much improved finetuning performance – even after accounting for the total dataset size and gradient steps. We explore two ways of splitting a document: 1) *Independent Splitting*: wherein all n sentences of a document are split independently into n separate training examples, 2) *Cumulative Splitting*: wherein an n-sentence document is split into n examples, with the ith example contains all sentences up to the ith sentence. We analyze the effect of sentence splitting on the model generalization in appendix B.1. In the following sections, we assume independent sentence splitting unless stated otherwise.

Experiments

Reversal Curse:

In Fig. 2, we first explore generalization in the context of the reversal curse phenomenon and dataset released by Berglund et al. (2024). We replicate the finding that finetuning on the forward directions does not produce generalization to the reversals. The authors of that work noted briefly that in-context the models could perform better at this task. We study this more systematically by presenting the entire dataset in context, and find that the model performs nearly at ceiling on the reversals—thus providing a strong demonstration of the benefits of in-context learning over finetuning. Finetuning with data augmented with in-context inferences yields similarly high test performance. Simple finetuning, on the other hand, has near zero accuracy as the finetuned model always prefers those (incorrect) celebrity name completions that were seen as targets during training, regardless of the context; both ICL and Augmented Finetuning significantly outpeform base finetuning (both $t > 149$, $p < 10^{-16}$). Finally, a pretrained model performs near chance on the test set, indicating a lack of contamination.

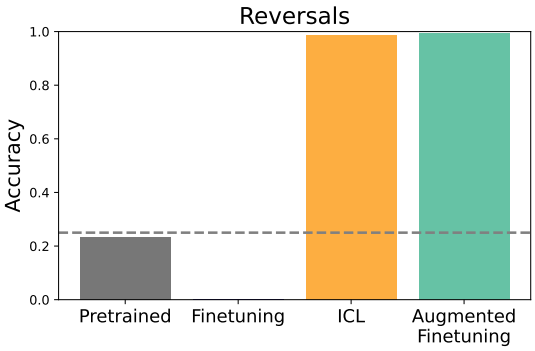

Figure 2: Reversal curse paper results.

Simple nonsense reversals: We then correspondingly test the models on the simple nonsense version of our reversal dataset (Fig 3, left). We find a weaker, but still noticeable, benefit of ICL

over finetuning in this setting, and again stronger benefits of augmented finetuning (respectively $ts = 3.36, 7.25$, $ps < 10^{-3}, 10^{-11}$). The difference in benefits compared to the experiments above may be due to differences in the plausibility of the relationships in question, e.g. the possibility that nonsense words interfere to some degree with the model's reasoning over longer contexts (see Appx. B.2).

Simple syllogisms: We next test the simple syllogisms dataset (Fig 3, right). Again, the pretrained model performs at chance, indicating a lack of contamination. Finetuning on the dataset does yield some non-trivial above-chance generalization; perhaps because for certain flavors of syllogism the logical inferences are compatible with simpler linguistic patterns — e.g., tuning on sequences like "All X include Y." "All Y include Z." might make the model more likely to predict "All X include Z" from pure associations. However, following simpler associative patterns is not valid for most syllogistic forms, e.g. if the universal quantifier is replaced by the existential in the examples above. Perhaps because of this, in-context learning yields much stronger performance; using in-context learning to augment the training dataset further improves performance overall (respectively $ts = 2.83, 6.30$, $ps < 0.01, 10^{-8}$).

Semantic structure benchmark

We then test the semantic structure benchmark (Fig. 4), which integrates multiple kinds of generalization in a richer learning setting. In this setting, we evaluate performance on: a) rephrased training statements (that preserve the direction of the relation) denoted as 'train' in the figure, b) reversals, c) syllogistic inferences, and d) propositions about held out categories. Across these settings, we find an overall benefit of ICL over finetuning, though the magnitude of this benefit varies depending on the split. We find some improvements in generalization to even rephrased information from training, and more dramatic improvements on reversals and syllogisms. However, the category-holdout split remains difficult, and improvements from ICL are minimal. Furthermore, we continue to find that augmenting the finetuning data with in-context inferences improves performance from finetuning, in many cases outperforming even ICL. Most improvements are statistically significant ($ts > 2.79$, $p < 0.01$) except for ICL on Syllogisms and both on Category show only marginal improvements ($1.95 > ts > 1.93$, $p = 0.05$). (N.B., in this setting we *do not* use sentence splitting for the finetuning baseline, as we find it impairs performance and we want to compare to the strongest baselines; results with sentence-splitting for the baseline, along with other ablations, can be found in Appx. B.5.)

Our results in this section also highlight an important nuance to prior results on the reversal curse (Berglund et al., 2024). When the information being tested forms part of a broader coherent structure of knowledge (such as our semantic structure), finetuning alone does exhibit *some* above-chance generalization to reversals. This generalization may be due to the fact that other information in the training set can support the reversed conclusion; e.g. if the reversal were from "birds include eagles" to "eagles are a type of bird," but the training set also includes statements such as "eagles

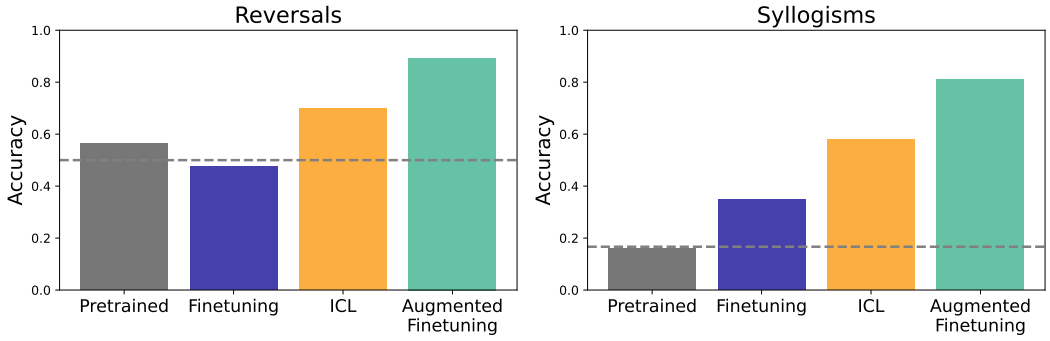

Figure 3: On our simple reversal (left) and syllogism (right) datasets, in-context learning outperforms finetuning. Moreover, augmenting the fine-tuning dataset produces strong improvements in model performance. Pretrained models perform near chance, showing that the datasets are not guessable based on superficial features.

have wings," that information may enable an alternative route to inferring the reversal (associatively if not logically). Nevertheless, in-context learning and augmented finetuning continue to perform substantially better than finetuning alone in most cases.

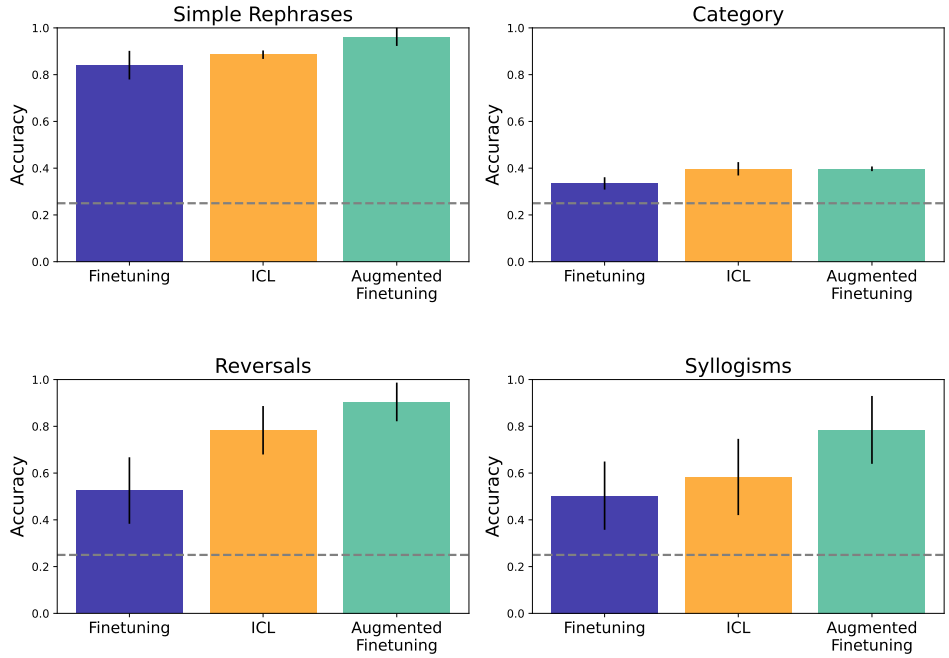

Figure 4: On the more richly structured semantic dataset, in-context learning still moderately outperforms finetuning. Furthermore, augmentation continues to show benefit for generalization from finetuning — even in rephrased questions about trained facts that do not involve reversal (top left). However some generalization splits, such as the category-level holdout, remain very challenging. (Error bars are standard errors computed over task subsets featuring different types of the inferences in question, e.g. reversals of property relations vs. reversals of category inclusion relations.)

Process knowledge: We also performed some preliminary experiments exploring learning and generalization of process-type (rather than semantic) knowledge, i.e. the ability to apply novel procedures to inputs. Specifically, we focused on executing a simple pseudo-mathematical procedure that we call "derivatoids", which transforms math-like expressions in a manner similar to computing a derivative. We created a train-test compositional generalization split in which certain combinations of rules are seen in training (again, either in context or via finetuning), but other combinations are held out for testing. The overall pattern of results is generally consistent with those above. ICL outperforms finetuning, especially in the low-shot regime, and augmentations improve finetuning performance. However, process knowledge requires distinct evaluation methods, and the effects correspondingly appear to be driven by distinct factors compared to the semantic knowledge experiments above; for brevity, we present the experiments and results in full detail in Appendix B.6.

Related work

In-context learning: A variety of works have explored in-context learning (Brown et al., 2020; Dong et al., 2022; Lampinen et al., 2024a). Several works have studied patterns in the learning and generalization of ICL, either empirically (Chan et al., 2022a; Garg et al., 2022), mechanistically (Olsson et al., 2022), or theoretically (Elmoznino et al., 2024; Xie et al., 2023; Zhang et al., 2024). Several recent works have found that even scaling to hundreds or thousands of in-context examples can improve LM performance on challenging datasets (Agarwal et al., 2024; Anil et al., 2024; Bertsch et al., 2024). Our work similarly relies on the ability of models to learn from many documents in context in order to generate the inferences for augmentation. Our work differs from prior works in

focusing more on generalization of broader types of information in context, beyond few-shot tasks (cf. Lampinen et al., 2024a).

Out-of-context learning: Several other papers (e.g. Berglund et al., 2023; Meinke and Evans, 2023) have considered how models can generalize "out-of-context"—that is, how they can use information that is not directly included in the prompt in a flexible way at test time. Our results may connect to these, e.g. in the patterns of partial generalization seen even with finetuning on the semantic structure dataset. However, we generally do not find reliable use of out-of-context information as in-context—i.e., in-context learning tends to outperform finetuning.

Factual learning & generalization: A variety of works have studied how language models learn and generalize factual information, both in pretraining (e.g. Allen-Zhu and Li, 2024, 2025; Zucchet et al., 2025) and during finetuning (e.g. Berglund et al., 2024; Gekhman et al., 2024). Many of these works have focused on failures to generalize like the "Reversal Curse" (Berglund et al., 2024), while others looked into increases in hallucination as LLMs are pretrained (Zucchet et al., 2025) or finetuned on new knowledge (Gekhman et al., 2024). However, these works have generally not performed data-matched experiments comparing the generalization of in-context learning and finetuning.

Data augmentation: A wide variety of works have explored how LLMs can be used to augment data for improved performance from small or narrow datasets, e.g. to improve zero-shot task performance (Maini et al., 2024) or to improve cross-lingual generalization (e.g. Whitehouse et al., 2023). Ding et al. (2024) reviews some of this broader literature. There have also been targeted attempts to fix specific issues like the reversal curse with hardcoded augmentations (Golovneva et al., 2024). A closely related work by Akyürek et al. (2024) proposes "deductive closure training" to improve coverage via prompting language models to generate deductive inferences from training documents. Padmanabhan et al. (2024) similarly propose generating continuations and then distilling them into the model. Several concurrent works Chen et al. (2024); Ruan et al. (2025) suggest that having LMs generate additional reasoning directions, and using these to augment their training data, can improve performance on reasoning tasks. Kujanpää et al. (2024) propose prompt distillation, which distills from larger prompts containing privileged information, and find that this improves generalization from knowledge in finetuning. Yang et al. (2024) use language models to extract entities in the training documents, and generate synthetic data reasoning about links between these entities; like our results, they find this method of "rearranging" knowledge in synthetic data helps to improve downstream performance. Finally, a concurrent work (Park et al., 2025) finds similar benefits to ours of in-context learning over finetuning, and likewise proposes augmenting data to overcome it. Our experiments show that in controlled settings, without the possibility of dataset contamination, similar approaches to augmenting small finetuning datasets can yield improved generalization—and relate the performance to that achieved through more basic finetuning and in-context learning.

Synthetic data: An equally broad literature has explored other applications of synthetic data to improving LM performance; see Liu et al. (2024) for a recent survey. Earlier works considered hand-designed synthetic data targeted to improve generalization using domain knowledge of areas like linguistics or mathematics (e.g. Pratapa et al., 2018; Wu et al., 2021). More recent approaches have focused on generating the data directly from language models, either filtering with ground-truth scores (Zelikman et al., 2022) or simply through self-consistency (Huang et al., 2023; Wang et al., 2023). While a recent prominent article argued that training on synthetic data leads models to collapse (Shumailov et al., 2024), other works have noted that in settings where synthetic data is mixed with the original data performance continues to improve (Gerstgrasser et al., 2024). We correspondingly find that in our setting, rather than being harmful, synthesizing augmented data with the models consistently improves performance (even on rephrased information from the training split in the semantic structure dataset). These results contribute to the ongoing discussion of how incorporating synthetic data affects model performance.

Discussion

In this paper, we performed controlled experiments on how language models generalize various types of novel information from in-context learning and finetuning. Overall, we find that the models generalize better on average along several dimensions from in-context learning. Using in-context reasoning to augment the finetuning dataset can exploit the complementary benefits of both to yield better performance.

The distinct inductive biases of in-context learning and finetuning: A variety of works have studied the inductive biases of in-context learning. One common theme has been to emphasize that in-context learning can approximate gradient descent (e.g. Von Oswald et al., 2023), in settings where that behavior is optimal. However, a variety of other works have found that the inductive biases of in-context learning can vary depending on factors such as dataset diversity (Raventós et al., 2024) or model scale (Wei et al., 2023). Indeed, Elmoznino et al. (2024) argue that in-context learning has an "Occam's razor"-like bias towards minimizing complexity, which enhances its generalization. Several works have explicitly noted the distinct inductive biases of in-context vs. in-weights learning (e.g. "rule-based" vs "exemplar-based" generalization) (Awadalla et al., 2022; Chan et al., 2022b; Shen et al., 2023)—while others have argued that they are more similar than they might seem (Mosbach et al., 2023). Our work contributes to this line of findings, by moving beyond the typical input-output tasks considered in these works, to consider other types of knowledge—and illustrates some potential utility of these differences, by showing how the more flexible generalization of in-context learning can be exploited to improve the generalization of fine-tuning.

Accessible information and learning by thinking: Lombrozo (2024) highlights how "learning by thinking" is a unifying theme across cognitive science and recent advances in AI—a system can acquire new information and skills purely through computation, without further inputs. Lombrozo highlights that while superficially this may seem paradoxical—information cannot be created—this further computation can increase the *accessibility* of information and thus improve performance in practice. This argument parallels the theoretical one made by Xu et al. (2020) on how computation can increase the accessibility of information. Our use of in-context reasoning to improve finetuning performance beyond the original data follows this pattern. For example, the information about reversals and syllogisms is always hidden within the data, but finetuning on the in-context inferences make this information more explicit and thus more readily accessible at test time.

Train-time inference scaling: Recently, various works have begun to explore test-time inference scaling to improve performance (e.g. Guo et al., 2025; Jaech et al., 2024). These findings complement prior studies exploring how scaling training compute (e.g. via larger models or more data) improves performance (e.g. Hoffmann et al., 2022; Kaplan et al., 2020). Our results illustrate that scaling train-time compute through in-context inference methods can likewise help to improve some aspects of model generalization.

Limitations: Our work suffers from several limitations. First, our main experiments rely on nonsense words and implausible operations. Although these counterfactual tasks allow us to avoid the possibility of dataset contamination, they may interfere with the performance of the models to some extent. For example, preliminary experiments (Appx. B.2) suggest that the ICL performance of the models on the Reversal Curse dataset degrades if the names are replaced with nonsense. Thus, tasks with more plausible entities may see greater benefits from ICL. It's possible that finetuning makes the nonsense terms more familiar, and that this contributes to the gap between augmented finetuning and ICL. However, in that case we likely *underestimate* the gap between ICL and base finetuning (as the latter would effectively benefit from increased "familiarity" of the nonsense entities). Second, we have not experimented with other language models, which would enhance the generality of our results. However, since the individual phenomena we build upon — e.g. the reversal curse when finetuning (Berglund et al., 2024) vs. the ability to do reversals in context (e.g. Lampinen et al., 2024b) — have been documented across multiple models, we believe it is reasonable to extrapolate our results to other settings. However, future work should study more carefully the differences in how models learn and generalize in these settings, including newer reasoning models (e.g. Guo et al., 2025).

Conclusions

We have explored the generalization of in-context learning and finetuning when LMs are exposed to various types of novel information structures. We find notable differences—often with ICL generalizing better for certain types of inferences—and propose ways of achieving better performance by adding in-context reasoning traces to augment finetuning data. We hope this work will contribute to the science of understanding learning, reasoning, and generalization in foundation models, and the practicalities of adapting them to downstream tasks.

Acknowledgements

We would like to thank the following people for helpful discussions: Sridhar Thiagarajan, Mike Mozer, Amal Rannen-Triki, Andrey Zhmoginov, Preethi Lahoti, Dilan Gorur, Johannes von Oswald. We would also like to thank Shakir Mohamed, Dipanjan Das, Raia Hadsell, Slav Petrov, Andrew Dai, Ruibo Liu, Tom Kwiatkowski, and Lukas Dixon for their support.

References

R. Agarwal, A. Singh, L. M. Zhang, B. Bohnet, L. Rosias, S. C. Chan, B. Zhang, A. Faust, and H. Larochelle. Many-shot in-context learning. In *ICML 2024 Workshop on In-Context Learning*, 2024.

A. F. Akyürek, E. Akyürek, L. Choshen, D. Wijaya, and J. Andreas. Deductive closure training of language models for coherence, accuracy and updatability. *Findings of the Association for Computational Linguistics*, 2024.

Z. Allen-Zhu and Y. Li. Physics of language models: Part 3.1, knowledge storage and extraction. In *International Conference on Machine Learning*, pages 1067–1077. PMLR, 2024.

Z. Allen-Zhu and Y. Li. Physics of language models: Part 3.2, knowledge manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025.

C. Anil, E. Durmus, N. Rimsky, M. Sharma, J. Benton, S. Kundu, J. Batson, M. Tong, J. Mu, D. J. Ford, et al. Many-shot jailbreaking. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

A. Awadalla, M. Wortsman, G. Ilharco, S. Min, I. Magnusson, H. Hajishirzi, and L. Schmidt. Exploring the landscape of distributional robustness for question answering models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5971–5987, 2022.

L. Berglund, A. C. Stickland, M. Balesni, M. Kaufmann, M. Tong, T. Korbak, D. Kokotajlo, and O. Evans. Taken out of context: On measuring situational awareness in llms. *arXiv preprint arXiv:2309.00667*, 2023.

L. Berglund, M. Tong, M. Kaufmann, M. Balesni, A. C. Stickland, T. Korbak, and O. Evans. The reversal curse: Llms trained on "a is b" fail to learn "b is a". In *The Twelfth International Conference on Learning Representations*, 2024.

A. Bertsch, M. Ivgi, U. Alon, J. Berant, M. R. Gormley, and G. Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

S. Chan, A. Santoro, A. Lampinen, J. Wang, A. Singh, P. Richemond, J. McClelland, and F. Hill. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems*, 35:18878–18891, 2022a.

S. C. Chan, I. Dasgupta, J. Kim, D. Kumaran, A. K. Lampinen, and F. Hill. Transformers generalize differently from information stored in context vs in weights. *MemARI Workshop, NeurIPS*, 2022b.

J. C.-Y. Chen, Z. Wang, H. Palangi, R. Han, S. Ebrahimi, L. Le, V. Perot, S. Mishra, M. Bansal, C.-Y. Lee, et al. Reverse thinking makes llms stronger reasoners. *arXiv preprint arXiv:2411.19865*, 2024.

B. Ding, C. Qin, R. Zhao, T. Luo, X. Li, G. Chen, W. Xia, J. Hu, L. A. Tuan, and S. Joty. Data augmentation using llms: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1679–1705, 2024.

Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, T. Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

E. Elmoznino, T. Marty, T. Kasetty, L. Gagnon, S. Mittal, M. Fathi, D. Sridhar, and G. Lajoie. In-context learning and occam's razor. *arXiv preprint arXiv:2410.14086*, 2024.

C. Gao, S. V. Shinkareva, and R. H. Desai. Scope: The south carolina psycholinguistic metabase. *Behavior research methods*, 55(6):2853–2884, 2023.

S. Garg, D. Tsipras, P. S. Liang, and G. Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Z. Gekhman, G. Yona, R. Aharoni, M. Eyal, A. Feder, R. Reichart, and J. Herzig. Does fine-tuning llms on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, 2024.

Gemini Team Google. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemini Team Google. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

M. Gerstgrasser, R. Schaeffer, A. Dey, R. Rafailov, H. Sleight, J. Hughes, T. Korbak, R. Agrawal, D. Pai, A. Gromov, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

M. Geva, R. Schuster, J. Berant, and O. Levy. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL `https://aclanthology.org/2021.emnlp-main.446`.

O. Golovneva, Z. Allen-Zhu, J. Weston, and S. Sukhbaatar. Reverse training to nurse the reversal curse. *arXiv preprint arXiv:2403.13799*, 2024.

D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 30016–30030, 2022.

J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han. Large language models can self-improve. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Y. Kirstain, P. Lewis, S. Riedel, and O. Levy. A few more examples may be worth billions of parameters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, volume 2022, pages 1017–1029. ACL Anthology, 2022.

K. Kujanpää, H. Valpola, and A. Ilin. Knowledge injection via prompt distillation. *arXiv preprint arXiv:2412.14964*, 2024.

A. K. Lampinen, S. C. Chan, A. K. Singh, and M. Shanahan. The broader spectrum of in-context learning. *arXiv preprint arXiv:2412.03782*, 2024a.

A. K. Lampinen, I. Dasgupta, S. C. Chan, H. R. Sheahan, A. Creswell, D. Kumaran, J. L. McClelland, and F. Hill. Language models, like humans, show content effects on reasoning tasks. *PNAS nexus*, 3(7):pgae233, 2024b.

R. Liu, J. Wei, F. Liu, C. Si, Y. Zhang, J. Rao, S. Zheng, D. Peng, D. Yang, D. Zhou, et al. Best practices and lessons learned on synthetic data for language models. *arXiv preprint arXiv:2404.07503*, 2024.

T. Lombrozo. Learning by thinking in natural and artificial minds. *Trends in Cognitive Sciences*, 2024.

P. Maini, S. Seto, R. Bai, D. Grangier, Y. Zhang, and N. Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14044–14072, 2024.

A. Meinke and O. Evans. Tell, don't show: Declarative facts influence how llms generalize. *arXiv preprint arXiv:2312.07779*, 2023.

M. Mosbach, T. Pimentel, S. Ravfogel, D. Klakow, and Y. Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.779. URL `https://aclanthology.org/2023.findings-acl.779/`.

C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

S. Padmanabhan, Y. Onoe, M. Zhang, G. Durrett, and E. Choi. Propagating knowledge updates to lms through distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

C. F. Park, Z. Zhang, and H. Tanaka. New news: System-2 fine-tuning for robust integration of new knowledge. *arXiv preprint arXiv:2505.01812*, 2025.

A. Pratapa, G. Bhat, M. Choudhury, S. Sitaram, S. Dandapat, and K. Bali. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553, 2018.

A. Raventós, M. Paul, F. Chen, and S. Ganguli. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. *Advances in Neural Information Processing Systems*, 36, 2024.

T. T. Rogers and J. L. McClelland. Précis of semantic cognition: A parallel distributed processing approach. *Behavioral and Brain Sciences*, 31(6):689–714, 2008.

Y. Ruan, N. Band, C. J. Maddison, and T. Hashimoto. Reasoning to learn from latent thoughts. *arXiv preprint arXiv:2503.18866*, 2025.

L. Ruis, M. Mozes, J. Bae, S. R. Kamalakara, D. Talupuru, A. Locatelli, R. Kirk, T. Rocktäschel, E. Grefenstette, and M. Bartolo. Procedural Knowledge in Pretraining Drives Reasoning in Large Language Models, Nov. 2024. URL `http://arxiv.org/abs/2411.12580`. arXiv:2411.12580 [cs].

J. Russin, E. Pavlick, and M. J. Frank. Human curriculum effects emerge with in-context learning in neural networks. *ArXiv*.

L. Shen, A. Mishra, and D. Khashabi. Do pretrained transformers really learn in-context by gradient descent? *arXiv preprint arXiv:2310.08540*, 2023.

I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

J. Treutlein, D. Choi, J. Betley, S. Marks, C. Anil, R. Grosse, and O. Evans. Connecting the dots: Llms can infer and verbalize latent structure from disparate training data. *arXiv preprint arXiv:2406.14546*, 2024.

I. Vieira, W. Allred, S. Lankford, S. Castilho, and A. Way. How much data is enough data? fine-tuning large language models for in-house translation: Performance evaluation across multiple dataset sizes. In *Proceedings of the 16th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 236–249, 2024.

J. Von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *arXiv:2201.11903 [cs]*, Jan. 2022. URL `http://arxiv.org/abs/2201.11903`. arXiv: 2201.11903.

J. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

C. Whitehouse, M. Choudhury, and A. F. Aji. Llm-powered data augmentation for enhanced cross-lingual performance. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Y. Wu, M. N. Rabe, W. Li, J. Ba, R. B. Grosse, and C. Szegedy. Lime: Learning inductive bias for primitives of mathematical reasoning. In *International Conference on Machine Learning*, pages 11251–11262. PMLR, 2021.

S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2023.

Y. Xu, S. Zhao, J. Song, R. Stewart, and S. Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020.

Z. Yang, N. Band, S. Li, E. Candes, and T. Hashimoto. Synthetic continued pretraining. *arXiv preprint arXiv:2409.07431*, 2024.

E. Zelikman, Y. Wu, J. Mu, and N. Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

R. Zhang, S. Frei, and P. L. Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

N. Zucchet, J. Bornschein, S. Chan, A. Lampinen, R. Pascanu, and S. De. How do language models learn facts? dynamics, curricula and hallucinations. *arXiv preprint arXiv:2503.21676*, 2025.

A Supplemental methods

A.1 Prompts

In this section we provide the prompts used for the augmentations. The relevant portions of the prompt were formatted with the corresponding content.

```
LOCAL_PROMPT = (
    'Please generate possible novel statements and rephrasings that can be '
    'inferred from each sentence on its own. Even a simple sentence has '
    'some alternative phrasings that are logically equivalent. Please '
    'only use logic and language to draw your conclusions, regardless of '
    'whether the entities in question actually exist.\n',
    'Statement: trillips are taller than zax.',
    'Inferences: trillips have greater height than zax. zax are shorter'
    ' than trillips. zax have lower heights than trillips.',
    'Statement: Note: Engineering is simpler than science.',
    'Inferences: Science is more complex than engineering. Engineering is '
    'less complex than science. Engineering is not as complex as science.',
    'Statement: "{text_to_augment}"',
)

GLOBAL_PROMPT = (
    'I am going to give you a bunch of documents, and then please use them as '
    'context to reason through all the logical consequences you can produce '
    'from a final target document. First, here are the source documents:\n\n'
    '{full_context}\n\n'
    'Now, please use that context to help me rephrase this document or reason '
    'through the consequences of the statements it contains. Please state all '
    'consequences as explicitly as possible, following the format of the '
    'source documents, and be as complete as possible.\n'
    '{target_document}\n'
)
```

For some of our ablation experiments we also used a document-level prompt without the global context:

```
DOCUMENT_PROMPT = (
    'Please generate possible novel propositions that can be logically '
    'inferred from each document on its own. Even a simple sentence has '
    'some alternative phrasings that are logically equivalent, and '
    'combining sentences can often yield new logical inferences. Please '
    'only use logic and language to draw your conclusions, regardless of '
    'whether the entities in question actually exist.\n'
)
```

A.2 Datasets

Semantic structure

The full tree structure used to derive the benchmark has 110 entities structured into a tree with depth of up to 7, with branching factors ranging from 1-8, and numbers of node-specific properties ranging from 1-6.

The (non-nonsense) relations used, e.g. “are a type of” were converted into natural language by a range of templates (per relation). In turn, these were embedded in a larger range of phrasing and formatting structures designed to mimic some superficial details of an article about a topic.

An example training document is shown below.

```
Jund
--------
```

```
Jund have yact. ("Phropt are a component of jund.") "Swug is a capability
    ↪  of jund." Jund are one kind of hipt. Jund can sunt; however, jund
    ↪ have the component splect.

* Thrund are the parent category of jund.
* (Jund are a subset of thont.)
* Jund have clud.
* Jund can glon.
* (Ust are a component of jund.)
* Spluff is a capability of jund.

Splect are a part of jund. Jund are a subset of thrund, and jund can
    ↪ swug.
```

Assuming this document were the only information in the training corpus, one example reversal QA pair that could be generated relative to it would be "Q: Clud are a part of...? A: Jund."

A.3 Compute resources

Finetuning and evaluation experiments used TPUv4 and v5 resources. For the largest (semantic) dataset each finetuning pass took about 10-20 minutes on 128 devices; evaluating a finetuned model took about 10-20 minutes on 8 devices. ICL evaluation took about 1.5-2 hours on 64 devices, and augmentation took about 3 hours on 256 devices. As the other dataset are smaller, we estimate reproducing all experiments in the main text of this paper would take around 2000 TPU-hours.

B Supplemental experiments and ablations

B.1 Effect of sentence splitting

We analyze the effect of splitting the training documents at sentence-level. Specifically, we study the effect of sentence splitting on the semantic structure benchmark with and without augmentations. For this experiment, we use two local augmentation methods, 1) *Sentence Augmentation*: wherein a prompted language model rephrases each sentence in the document using the local prompt (cf. A.1), 2) *Document Augmentation*: wherein a prompted LM generates augmentation at the document-level using the document prompt (cf. A.1). The results of this experiment are reported in Fig. 5. It can be seen from the figure that sentence splitting consistently helps when augmentation is employed. However, on the vanilla finetuning baseline (i.e.) the one with no augmentation, sentence splitting sometimes makes the performance worse. This suggests that when a fact is presented in multiple ways in a document – the case for the augmentation baselines – then splitting those rephrases into different examples as targets provides a better learning signal for the model. For example, in the independent sentence splitting case, the benefit for augmented data may come from avoiding a kind of "explaining-away" phenomenon — where the model does not learn as effectively from gradient updates on a piece of information if the context already makes that information likely.

B.2 Non-sensification and Long-context ICL

In the main paper, we observed that full-dataset in-context evaluation performs very well for the reversal task of Berglund et al. (2024) (cf. Fig. 2). However, for the nonsense reversals, the ICL performance is relatively low (cf. Fig. 3). We conjecture that this difference in performance can be attributed to the nonsense nouns used in the simple reversal task as an LLM does not have a good semantic prior over these nonsense nouns so it can't utilize its long-context as effectively to do ICL. To test this hypothesis, we modify the dataset of Berglund et al. (2024) by nonsensifying all the celebrity names.

Doing full-dataset ICL on this modified dataset results in a significantly reduced performance, as shown in Fig. 6. This experiment suggests that models may suffer from using their long-context effectively if they don't already have a good prior over the context—e.g. if the context contains many nonsense words. We leave a more systematic study of this hypothesis to future work.

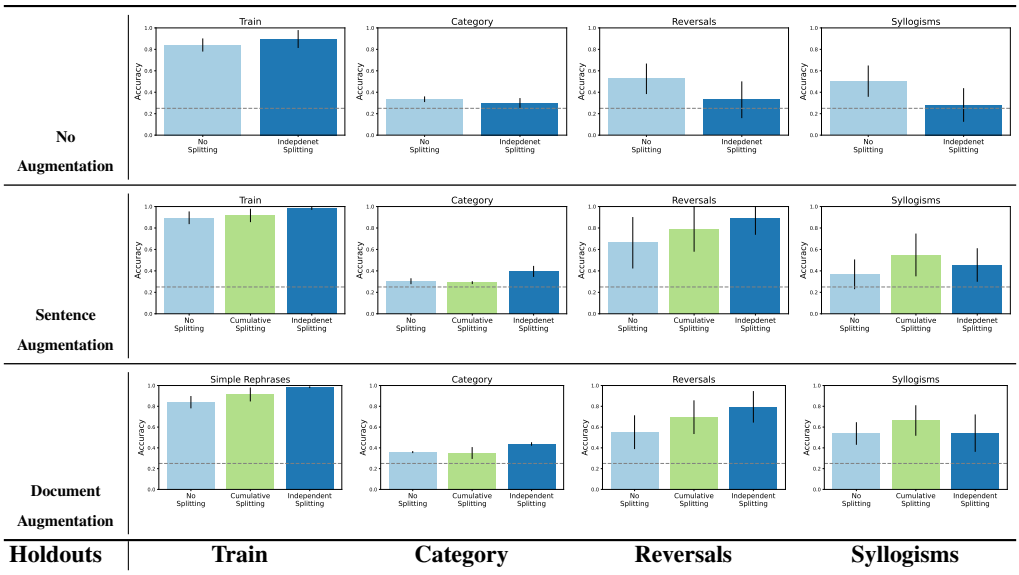

Figure 5: **Sentence splitting analysis on the semantic structure benchmark**. These plots show the finetuning performance when the documents in the training dataset are split at sentence-level. We can observe that except for augmented variants of the dataset, sentence splitting consistently improves performance.

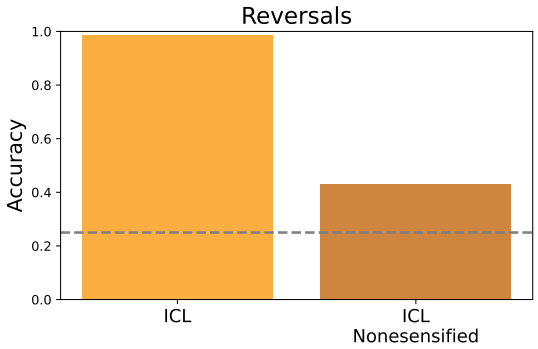

Figure 6: Comparison of ICL with and without nonsensification of celebrity names in the Reversal Curse Paper (Berglund et al., 2024) dataset.

B.3 Effect of Model Size

In the main paper, we reported results on the Gemini 1.5 Flash model. In order to see how model size affects the results, we compare the performance of a smaller Gemini 1.5 Flash-8B model with the larger Gemini 1.5 Flash model. The comparison of the two models is reported in Fig. 7. We can see that across both model sizes augmented finetuning performs better. Additionally, for a smaller Flash-8B model, in-context full-dataset evaluation (ICL) performs worse than vanilla finetuning on some splits (syllogisms, for example). This is inline with the existing literature that suggests that small models are not efficient in-context learners. However, note that augmented finetuning gives stronger generalization even in this smaller model regime.

B.4 Hyper-parameter Sweeps

We ablate batch size and training steps for the finetuning baseline. The results are reported in Fig. 8 and Fig. 9. The numbers reported in the main paper are based on the strongest performing baseline.

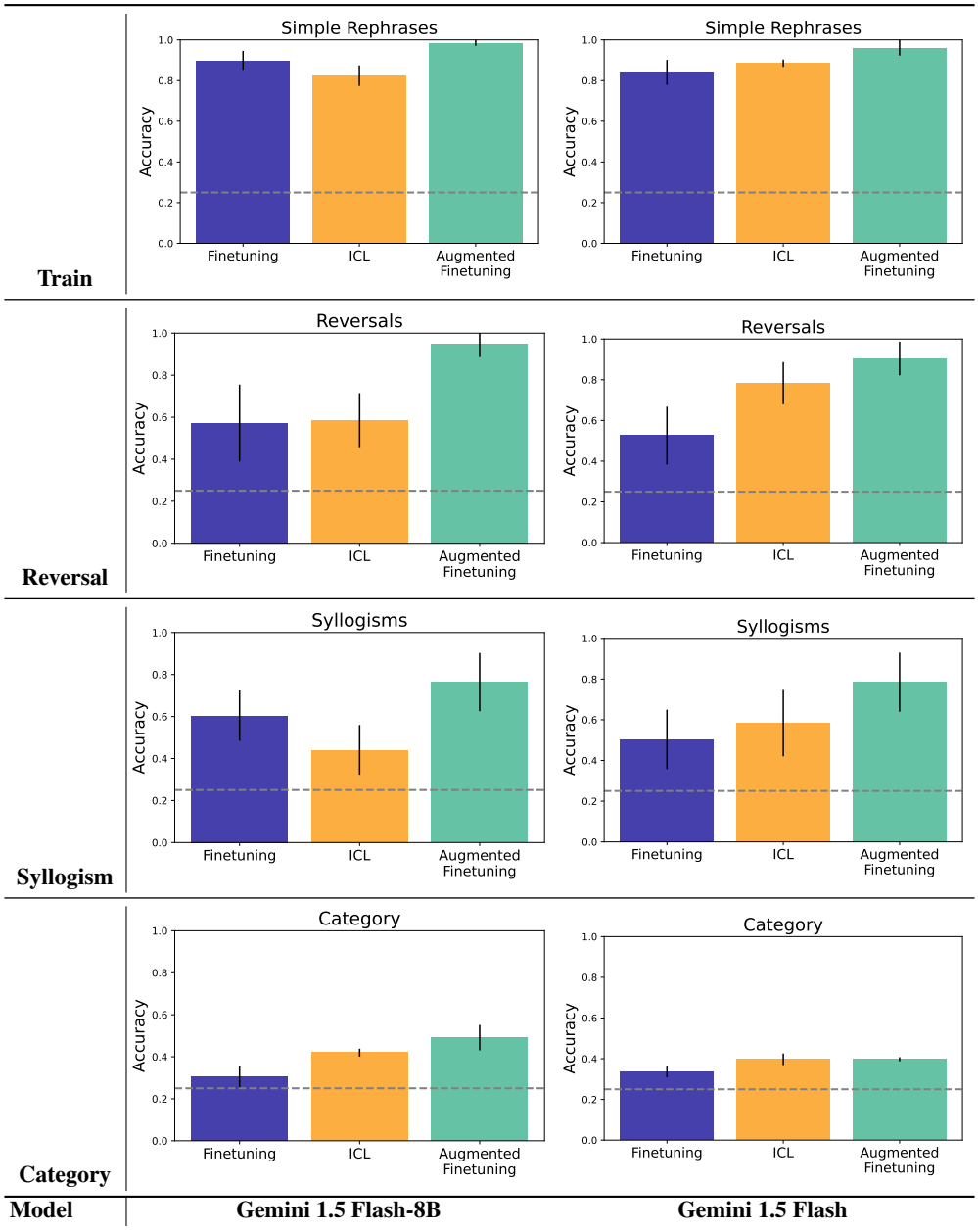

Figure 7: **Effect of model size**. Here we compare the results of full-dataset evaluation (ICL) with finetuning on the semantic structure benchmark using two different sized models. We see consistent gains with augmented finetuning at different model scales.

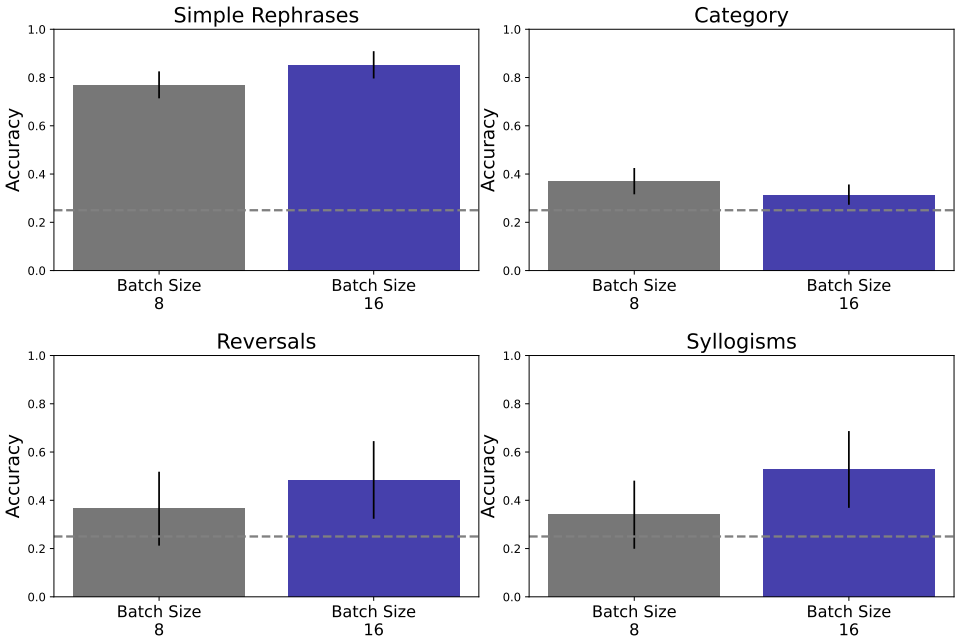

Figure 8: Batch size ablation on the semantic structure benchmark on the Finetuning baseline. The performance of batch size of 16 is slightly better so we use that in the main paper. However, changing the batch size doesn't improve the performance of the finetuning baseline significantly and the gains we see for the ICL and Augmentation methods in the main paper are not due to the poor choice of batch size selection for the baseline.

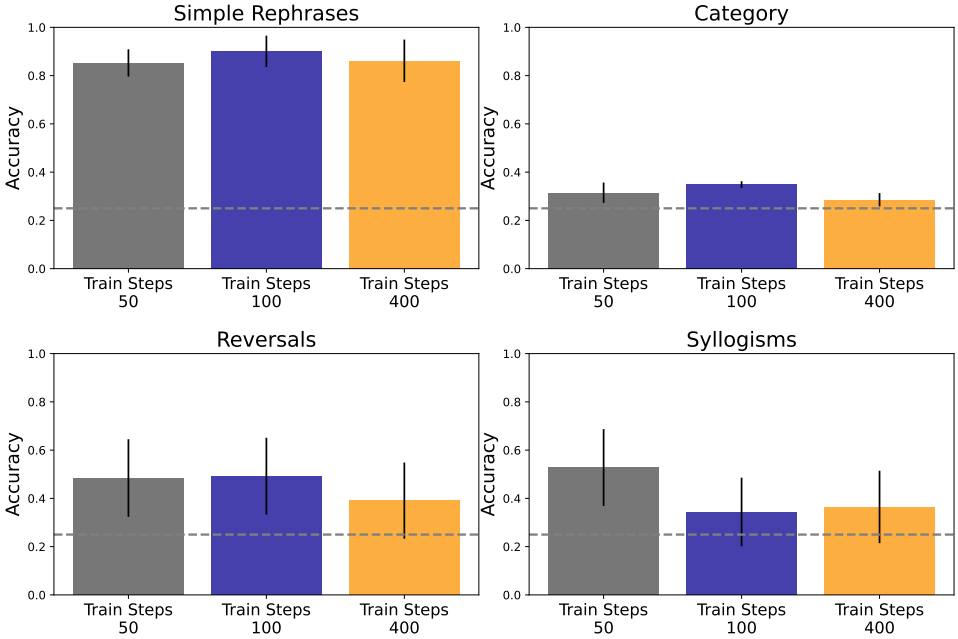

Figure 9: The effect of varying the training steps in the Finetuning baseline on the semantic structure benchmark. Overall the results are fairly stable across the training steps. Note that changing the number of training steps doesn't improve the performance of the finetuning baseline significantly and the gains we see for ICL and Augmentation methods in the main paper are not due to the under or over-training of the baseline.

B.5 Efficacy of different prompts

We now compare the performance of different augmentation methods proposed in the main paper. For this experiment, we use two local augmentation methods, *sentence augmentation*: wherein a prompted language model rephrases each sentence in the document using the local prompt, 2) *document augmentation*: wherein a prompted LM generates augmentation at the document-level using the document-only prompt, and a *global augmentation* method whereby a prompted LLM generates inferences by concatenating all the documents in the training dataset and using the global prompt. The results of the experiment are reported in Fig. 10. It can be seen from the figure that any type of augmentation consistently improves over no-augmentation baseline (confirming the findings in the main paper). Additionally, we can see that, across different holdout splits, different augmentation methods perform better (although the difference is not huge). Overall, we find that global augmentation generally performs better across all the holdout types of the semantic structure dataset.

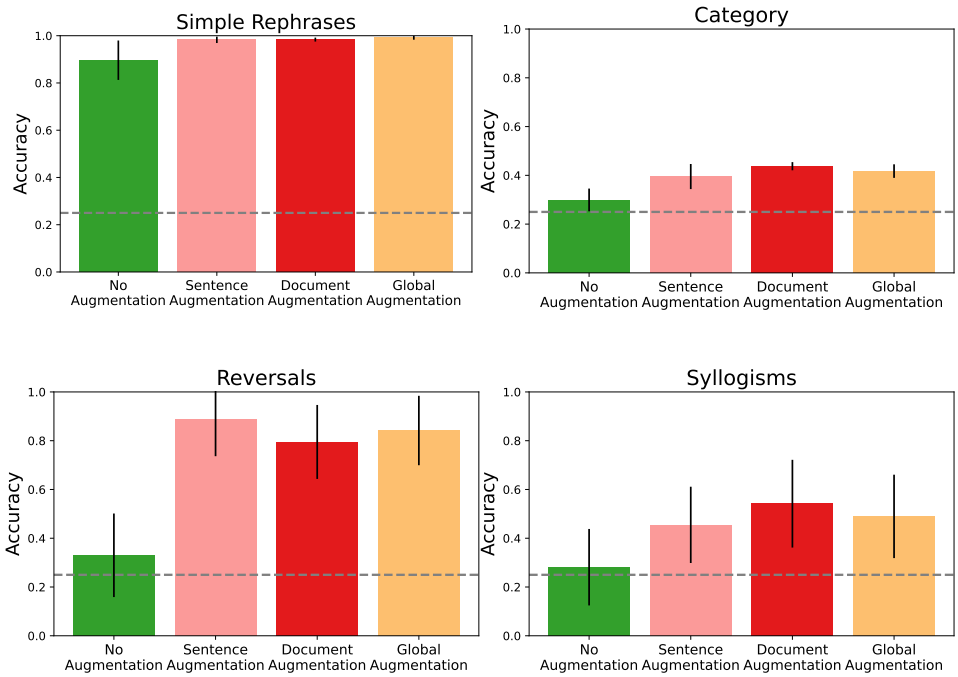

Figure 10: Comparison of different augmentation methods on the semantic structure dataset. Augmentations give significant improvement over finetuning without augmentations. Across different holdouts, different augmentation methods perform better or worse; thus, combining them (as in the main results) offers complementary benefits. (All methods use independent sentence splitting, including the no-augmentation baseline. Error bars are computed over task subsets featuring different types of the inferences in question.)

B.6 Process benchmark

We performed exploratory analyses on a "Process" benchmark that tests a model's ability to apply a novel *procedure* to inputs. This is distinct from learning "semantic" factual data, as the other benchmarks are designed to test.

We found preliminary results showing benefits of data augmentation for low-shot finetuning. We include them here for completeness, and as a remaining open challenge for data-efficient finetuning.

Process benchmark description: "Derivatoids"

This benchmark is built to test a model's ability to learn a novel "process" from examples. Here, a process is a transformation of an input. This contrasts with learning facts or semantics (which is tested

Primitive family	original → derivatoid
exponential	$b^x \rightarrow x^b$
polynomial	$c \times x^e \rightarrow (c+e) \times x^{(e+2)}$ (for each term)
logarithmic	$log(x, base) \rightarrow log_b log_b(x)$
trigonometric	sin → meep cos → morp tan → moop

Table 1: Derivatoid transformation rules for "primitives".

Combination type	original → derivatoid
addition	$u + v \rightarrow du - dv$
multiplication	$u \times v \rightarrow v * u + du * dv$
composition	$u(v) \rightarrow du * dv * dv(u)$

Table 2: Derivatoid transformation rules for combinations of "primitive" expressions, where u and v are primitives, and du and dv are their derivatoids (as defined in Table 1)

by the semantic structure benchmark above), given that processes may be learned and represented differently than facts (e.g., Geva et al., 2021; Ruis et al., 2024).

We target a few desiderata: (a) The processes are not familiar to the pretrained LM, requiring the model to go beyond mapping a familiar process to novel symbols, given that models can simply learn a latent mapping to that existing process (Treutlein et al., 2024). (b) At the same time, the task uses mostly familiar words and symbols, to avoid tokenization issues.

To fulfill these desiderata, we designed "derivatoids", which are a derivative-like transformation of math expression. "Primitive" expressions are transformed according to Table 1, and combinations of primitives are transformed according to to Table 2. For example, the input $(log(x, 39)) * (64 * *x)$ should be converted to $(64 * *x) * (log(x, 39)) + (log39log39(x)) * (x * *64)$.

The composition of primitives allows us to evaluate *combinatorial generalization* – the ability of models to perform on compositions where the primitives were seen in training, but not in those particular combinations. For example, we might train on [exponentials multiplied with polynomials] and [logarithmic multiplied by trigonometric functions], and evaluate on [exponentials multiplied by logarithmic functions].

In our experiments, we explore the data efficiency of ICL and SFT by creating datasets with k "shots". E.g. an 8 "shot" dataset for SFT consists of 8 examples in a standard training dataset. An 8-shot dataset for ICL includes 8 process examples (expression input and expression output) in context to demonstrate the derivatoids process, and a final query expression for which the model must provide the correct derivatoid.

Methods

We perform evaluation by computing ROUGE-L on the model's outputs, using the correct derivatoid output as the reference.

We generally fine-tune with batch size 8 and learning rate $1 \cdot 10^{-5}$. We selected both learning rates and fine-tuning checkpoints based on a validation loss, computed on an unseen set of derivatoid examples drawn from the same distribution as the train set for the unaugmented data.

Experiments

In Fig. 11, we inspect data efficiency for both in-context learning (ICL) and supervised finetuning (SFT), on the process benchmark. Data efficiency is measured as a function of the number of unique examples seen k (number of "shots", using the terminology of few-shot learning). For ICL, k is the number of examples seen in context, and for SFT, k is the number of unique examples in the

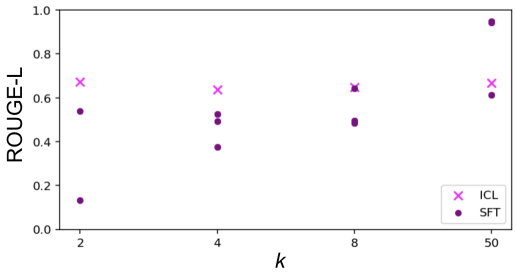

Figure 11: Data efficiency for in-context learning (ICL) vs supervised fine-tuning (SFT). k is the number of distinct examples viewed by each method – the number of "shots" in context (for ICL) or the number of training examples (for SFT). For the ICL results, we average across 3000 samples, each of which contains a k-shot example. For SFT, given the relative cost of finetuning, we only evaluate 3 different datasets, each of which contains k examples; each purple point represents one dataset and training run.

training data (the models are trained on multiple epochs). We find that ICL is surprisingly stable as we increase k, but SFT's performance predictably improves we increase k. ICL generally outperforms SFT for low k, exhibiting greater data efficiency.

All results in this section are for the version of the process benchmark where primitives are combined via multiplication, and evaluations are performed on the model's ability to perform combinatorial generalization, i.e. on a held-out set of combinations where the individual primitives (e.g. polynomials and trig functions) were seen in training, but the particular combinations were not seen in training. Surprisingly, across all of our experiments, performance on combinatorial generalization was very similar to generalization within distribution (i.e. novel examples of the same combinations that were seen in training, e.g. new examples of polynomials combined with logarithmic functions).

We explored augmented finetuning for improving the performance of SFT, for $k = 4$ and $k = 8$, given that SFT was generally outperformed by ICL in this low-shot regime, indicating potential room for improvement.

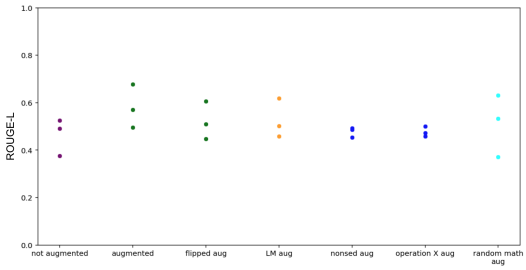

Figure 12: Augmentation results for 4-shot SFT.

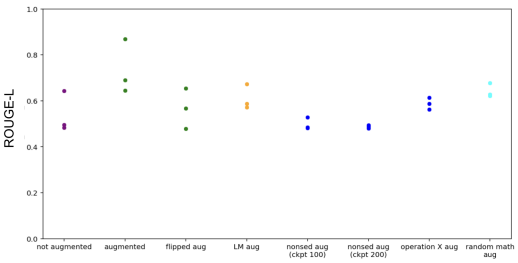

Figure 13: Augmentation results for 8-shot SFT.

We first applied train-time augmentations which consisted of "reflections" on the primitive categories and combination for the prompt input. E.g. for an input "(35**x) * (sin(x))", the full response

should look like “Reflection: This looks like a Exponential function multiplied by a Trigonometric function. Answer: (sin(x)) * (35**x) + (x**35) * (meep(x)).” These augmentations were generated programmatically in the training data, as a kind of upper bound on the effects of such augmentations (to distinguish from the effects of the augmentation generation itself, which can be very sensitive to the prompt etc). We found that such augmentations led to improvements for 8-shot SFT (Fig 13), and less so for 4-shot SFT (Fig 12).

To better understand the origin of the benefit for 8-shot SFT, we ran a series of additional experiments:

- To understand whether the effect occurs at train-time or test-time, we also tried flipping the ordering of the responses so that the reflections came *after* the answers. E.g. “Answer: (sin(x)) * (35**x) + (x**35) * (meep(x)). Reflection: This looks like a Exponential function multiplied by a Trigonometric function.” This did not lead to the same improved performance. This may be because the augmentation effect is mostly a test-time effect (c.f. “chain of thought reasoning” (Wei et al., 2022)), or because the augmentations cannot affect representations in the right way at train-time, especially given the particular form of the augmentations “This looks like...”, which aren’t as sensical when they follow the final answer rather than the input.
- To understand the sensitivity of the approach to the particular augmentations, we used a language model to generate similar augmentations. These prompted augmentations did not lead to the same performance enhancement; however, this prompt was not tuned in any way (see prompt in Appendix A.1).
- To understand whether the augmentations act as “clustering labels” for similar examples (e.g. so that related examples could better scaffold each others’ learning), we replaced certain words in the reflections with nonsense words, but using a consistent mapping so that e.g. “Linear” always mapped to “Glon”. The nonsense words were more out-of-distribution for the model, which may actually have caused a decrement in performance. Thus, we also tried training the model for longer (200 steps instead of 100 steps), and using real words instead of nonsense words e.g. "Operation A" and "Operation B". However, none of these interventions exceeded baseline non-augmented performance.
- To understand whether the models benefit from augmentations that are math-like without being specific to the problem at hand, e.g. by shifting the model more towards a math-like persona, we also tried augmenting with random statements about math (see examples in Appendix A.1). These augmentations did not lead to the same performance enhancement.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims in the abstract and introduction focus on the experimental results given in the experiments section.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Limitations in the Discussion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No proofs given.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Details on the datasets and training/evaluation methods are provided in the methods section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While some of the prior datasets used in this paper are available, and we believe we have provided sufficient information to reproduce the results, exact reproduction code cannot be released due to use of proprietary tools.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Methods and Experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We include error bars in figures for our main dataset (computed across dataset subsets testing e.g. different types of reversals); we do not for some of the simpler datasets such as the reversal curse because they lack these subsets.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix A.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We followed the code of ethics; see answers above and below.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This is a science-of-learning investigation of fundamental machine learning concepts; there are no direct societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No model or data release.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Datasets used in the paper (directly or as inspiration) are referenced appropriately.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets released, see above.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human subjects were involved.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: As above.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Insofar as this project is researching LLM learning and generalization, their use is described.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.