

Improving Latent Generalization Using Test-time Compute

Arslan Chaudhry, Sridhar Thiagarajan
 Google DeepMind
 {arslanch, sthiagarajan}@google.com

Andrew Lampinen *
 Anthropic

Abstract

Language Models (LMs) exhibit two distinct mechanisms for knowledge acquisition: in-weights learning (i.e., encoding information within the model weights) and in-context learning (ICL). Although these two modes offer complementary strengths, in-weights learning frequently struggles to facilitate deductive reasoning over the internalized knowledge. We characterize this limitation as a deficit in latent generalization, of which the reversal curse is one example. Conversely, in-context learning demonstrates highly robust latent generalization capabilities. To improve latent generalization from in-weights knowledge, prior approaches rely on train-time data augmentation, yet these techniques are task-specific, scale poorly, and fail to generalize to out-of-distribution knowledge. To overcome these shortcomings, this work studies how models can be taught to use test-time compute, or ‘thinking’, specifically to improve latent generalization. We use Reinforcement Learning (RL) from correctness feedback to train models to produce long chains-of-thought (CoTs) to improve latent generalization. Our experiments show that this thinking approach not only resolves many instances of latent generalization failures on in-distribution knowledge but also, unlike augmentation baselines, generalizes to new knowledge for which no RL was performed. Nevertheless, on pure reversal tasks, we find that thinking does not unlock direct knowledge inversion, but the generate-and-verify ability of thinking models enables them to get well above chance performance. The brittleness of factual self-verification means thinking models still remain well below the performance of in-context learning for this task. Overall, our results establish test-time thinking as a flexible and promising direction for improving the latent generalization of LMs.

1 Introduction

Large Language Models (LLMs) have shown remarkable generalization abilities across many domains especially when the task at hand is being presented in-context. The in-context generalization of LLMs is remarkably consistent even when they are queried for knowledge that is logically implied but not directly stated in the information present in the context. Contrary to in-context learning (ICL), however, when models are finetuned on new tasks, where they are expected to generalize from their in-weights knowledge, the generalization can be surprisingly brittle – particularly for information that is logically deducible but not directly stated in the training set. The well-known cases of such lack of generalization, referred to as latent generalization¹ (Lampinen et al., 2025b) henceforth, include finetuned LLMs’ failure to answer queries about trained facts in the reverse order, studied by (Berglund et al., 2024) as reversal curse phenomenon, and similar failures in syllogistic inference tasks as studied by Lampinen et al. (2025a).

To improve the latent generalization of finetuned models, previous approaches (Golovneva et al., 2024; Yang et al., 2024; Akyürek et al., 2024; Lampinen et al., 2025a) rely on increasing training dataset coverage by targeted data augmentation techniques during the pre-training

*Work done while at Google DeepMind.

¹We borrow this term from latent learning studied in the Psychology literature (Blodgett, 1929; Tolman, 1948)

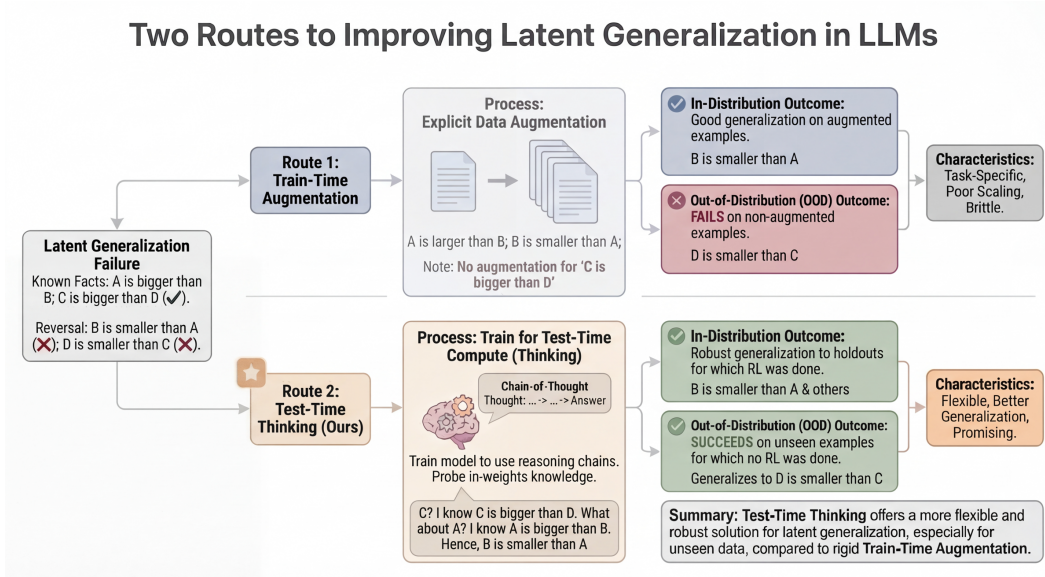


Figure 1: Comparison between train-time augmentation strategies vs test-time thinking to improve latent generalization.

or finetuning phases. Specifically, [Golovneva et al. \(2024\)](#) ameliorated the reversal curse in LLMs by explicitly training on strings in both the forward and reverse directions. [Lampinen et al. \(2025a\)](#), on the other hand, generated dataset augmentations by putting the entire training dataset in-context and generating finetuning data that helps models generalize on reversal and syllogistic tasks. While these works show strong latent generalization of LLMs for specific tasks, we argue that these train-time augmentation techniques are ultimately less flexible as **a)** they assume the latent structure of the task at hand to generate effective augmentations (reversal structure in [Golovneva et al. \(2024\)](#) and syllogistic and reversal structures in [Lampinen et al. \(2025a\)](#)) thus changing the task would require a change in augmentation mechanism, **b)** there is limited to no generalization to data points that are not augmented, **c)** and since each training data point needs to be explicitly augmented, these approaches scale poorly as the size of the training datasets increases.

To ameliorate the aforementioned shortcomings of train-time augmentation techniques, this work studies how LLMs can be trained to spend test-time compute (aka thinking) effectively to improve latent generalization. We train LLMs that can produce chains-of-thoughts (CoTs) probing the knowledge acquired during the finetuning phase. Specifically, we first replicate the lack of latent generalization of LLMs from their in-weights knowledge on different datasets: semantic structure dataset by [Lampinen et al. \(2025a\)](#) and reversal curse dataset by [Berglund et al. \(2024\)](#). We then show that finetuned models that lack latent generalization, once trained to do long CoT through Reinforcement Learning (RL) from correctness feedback can be remarkably effective in latent generalization. We further show that these models not only generate effective long CoTs on the held-out set of the dataset from which the RL prompts are constructed – a case of in-distribution latent generalization – but also generalize to the datasets for which no RL was done – an instance of out-of-distribution latent generalization. This latter point is particularly important as we previously hinted that, contrary to test-time thinking, models trained with train-time augmentations do not generalize to datasets for which no augmentations were generated (i.e.) out-of-distribution latent generalization is not possible with train-time augmented models (Figure 1).

While the experiments reveal that the thinking recipe, that we develop for reasoning about in-weights factual knowledge, is quite effective in improving latent generalization on many deductive reasoning tasks mentioned before, we, however, found that pure reversals are still quite hard for thinking models to perform robustly. We find that a model which is encouraged to do self-verification in its thinking trace shows non-trivial performance on pure reversals in comparison to models which do not use thinking. However, we find that

models are often brittle at factual self-verification, and hence their performance remains well below the ICL baseline. Consequently, strict logical inversions remain persistently challenging to execute, even for models with intermediate reasoning capabilities. Nevertheless, we view the utilization of test-time reasoning traces as a highly promising paradigm for advancing latent generalization in large language models.

Overall we contribute to the literature in the following ways:

1. We show that models with right thinking behaviors to probe their in-weights factual knowledge generalize effectively on latent learning tasks compared to their train-time augmented counterparts in data-matched settings.
2. Further, thinking models robustly generalize to in-weights knowledge for which no specific thinking training/ RL is done. We do not observe such out-of-distribution latent generalization with train-time augmentations driven baselines, even when these baselines undergo reinforcement learning.
3. Pure (0-hop) reversals of knowledge remain difficult even for thinking models, but factual self-verification enables non-trivial performance compared to models which do not utilize thinking.

2 Method

To facilitate latent generalization, our method aims to enhance the model’s capacity to effectively elicit its in-weights knowledge and surface relevant information in-context within its intermediate reasoning traces. We achieve this in the following three stages:

2.1 Knowledge Acquisition

To study the latent generalization under a controlled setup, we first add new knowledge in a pre-trained LLM by finetuning it on a dataset that is not leaked during the pre-training stage. During this supervised finetuning (SFT) phase, we also jointly train on auxiliary datasets to preserve the model’s instruction-following and pre-existing reasoning capabilities. However, we note that this stage is only necessary when the target knowledge required for effective generalization is not already encoded within the model’s parameters.

2.2 Thinking Bootstrapping

Once the new knowledge is incorporated in the LLM (LLM_s) and before starting the reinforcement learning phase, we induce desired reasoning behaviors in LLM_s to improve latent generalization. For this, we first construct a prompt (detailed in Appendix C.5) that explicitly defines the desired properties of a reasoning trace that can aid effective latent generalization: focus-driven recall, self-probing, generate-and-verify etc. The behaviors encouraged by the prompt include behaviors established in prior literature for reasoning benchmarks [Gandhi et al. \(2025\)](#), but we try to encourage behaviors which can help deduce and connect factual knowledge through other knowledge factoids the model may possess. We use this prompt on an off-the-shelf teacher LLM ([Gemini, 2025](#)) (separate from LLM_s) to curate a dataset with golden reasoning traces on a small subset of examples designed to evaluate latent generalization (e.g., queries requiring the reversal of knowledge acquired in the preceding phase). To reduce hallucinations in these reasoning traces, we condition the teacher LLM on the ground-truth answer for each query and discard any completions that fail to arrive at the correct final answer. Once this initial dataset is constructed, we further fine-tune the LLM_s via SFT on this dataset. This procedure is analogous to the Self-Taught Reasoner (STaR) framework by [Zelikman et al. \(2022\)](#), with the distinction that we utilize a more capable teacher model to generate the bootstrapping traces rather than the model being trained (LLM_s). For fair comparison, we use the same teacher model for the augmentation baseline as well.

2.3 Reinforcement Learning (RL)

Using the resulting checkpoint from the thinking bootstrapping as the initial policy, we perform RL on a combined dataset comprising prompts that necessitate latent generalization, alongside prompts that directly evaluate knowledge present in the training distribution. The policy is updated using a variant of the REINFORCE algorithm (Williams, 1992).

To test out-of-distribution generalization, we keep a disjoint dataset that is strictly excluded from both the bootstrapping and RL training mixtures, and is exposed to the model only during the knowledge acquisition phase.

3 Experiments

3.1 Model & Datasets

Our objective is to investigate the effect of increased test-time compute on latent generalization. Therefore, we start off with a pre-trained model that has already undergone standard reasoning fine-tuning (i.e., it is capable of generating extended Chain-of-Thought (CoT) traces for specific prompts already). Specifically, we utilize Gemini 2.5 Flash (Gemini, 2025) as the base model for all baselines and subsequently perform fine-tuning on controlled datasets to evaluate latent generalization.

For our controlled experimental setup, we utilize two datasets, ensuring strict exclusion from the model’s pre-training corpus to prevent data contamination:

Semantic Structure Benchmark (Lampinen et al., 2025a) This benchmark consists of real-world categories and relations, enabling the evaluation of various forms of knowledge generalization, including logical reversals and syllogistic inferences. For instance, the model might be exposed to the premise “dogs are a type of mammal” during training, and subsequently queried on the reversal “mammals include dogs” at test time. Alternatively, if exposed to an additional premise, “mammals are warm-blooded,” the model could be evaluated on the syllogistic inference “dogs are warm-blooded,” which requires composing the two independent facts acquired during training. For the out-of-distribution structure, we keep similar depth and branching factor as the in-distribution structure, but modify several categories to have different properties, for example a capability instead of an attribute. More details on how we use this dataset for our experiments are given in Appendix B.1.

This dataset enables us to evaluate latent generalization under the following conditions:

- **Syllogisms:** Requires the logical composition of two distinct facts present in the training documents to deduce the correct answer.
- **Reversals:** Requires the logical inversion of a trained fact. We evaluate two specific sub-types: **(a) non-strict reversals**, where the model can deduce the answer via an alternative multi-hop reasoning path, and **(b) strict reversals**, where no alternative reasoning path exists aside from the direct inversion of the premise.
- **Category holdouts:** Evaluates generalization when the training documents contain only a single fact about a specific category (e.g., explicitly specifying its parent category), while all other valid inferences are held out for testing. We further partition these holdouts into: **(a) category as subject**, where the held-out category acts as the subject in the test proposition, and **(b) category as target**, where the held-out category acts as the object or target.

Reversal Curse Dataset (Berglund et al., 2024) This dataset comprises fictional celebrity names paired with their corresponding descriptions. During training, the dataset presents specific celebrity names followed by their descriptions (i.e., name-to-description). At test time, the model is provided with a description and queried to generate the corresponding celebrity name (i.e., description-to-name). During the knowledge acquisition phase, we introduce a data augmentation step requiring the model to unconditionally output a randomly sampled name from the dataset. This ensures the model acquires a non-zero probability of generating these fictional names, as it must observe them as target tokens

during training. This dataset parallels the strict reversal condition of the semantic structure benchmark, as no alternative reasoning path exists to deduce the correct answer other than the direct reversals. Further details on dataset splits for different training and testing phases are given in Appendix B.2

3.2 Baselines

We evaluate our approach against the following baselines:

1. **Fine-tuning (SFT):** Standard supervised fine-tuning conducted exclusively during the knowledge acquisition phase.
2. **Fine-tuning + RL:** The initial knowledge acquisition phase followed by RL, but without generating intermediate reasoning tokens. This baseline isolates the performance gains attributable specifically to the reasoning process, ruling out improvements stemming solely from RL optimization on the factual prompts, which is known to independently enhance factual accuracy (Tian et al., 2023).
3. **Augmentation:** Following Lampinen et al. (2025a), we apply data augmentation by providing the entire dataset in-context and subsequently training on this augmented dataset during the knowledge acquisition phase.
4. **Augmentation + RL:** The aforementioned augmentation strategy followed by the RL phase, again omitting the generation of intermediate reasoning tokens.
5. **In-Context Learning (ICL):** The model is evaluated with the entire training dataset provided in the context window. Large language models typically exhibit strong latent generalization under this ICL paradigm.
6. **Reasoning (Ours):** Initial SFT during the knowledge acquisition phase, followed by reasoning bootstrapping and RL, adhering to the methodology detailed in Section 2.

During all fine-tuning procedures, we jointly train on the base model’s (Gemini 2.5 Flash) original instruction-tuning and reasoning datasets. This co-training ensures that our empirical results are not confounded by the catastrophic forgetting of pre-existing instruction-following and reasoning capabilities. Training is performed using the Adam optimizer with a learning rate of 3×10^{-4} , a batch size of 16, and a maximum sequence length of 8192 tokens. For all baselines, we employ early stopping based on the validation loss.

3.3 Evaluation & Metrics

For evaluation, we employ a closed-book Question Answering (QA) framework, sampling generations from the model at a temperature of $T = 1.0$, rather than computing the log-likelihood of ground-truth completions. We ensure un-leaked evaluation prompts during the Reinforcement Learning (RL) phase, and take additional care to prevent leakage via thought traces, which we discuss in C.1. For all baselines, we compute best@1 and best@N metrics, corresponding to a single attempt and N attempts, respectively, and use a bootstrapped estimator over all completions to reduce variance.

For the **Semantic Structure Benchmark**, the model is prompted to generate a comprehensive list of all valid completions for a given test query. The ground truth comprises a set $\mathcal{A}_{all}$ representing all valid completions—some of which may have been explicitly stated in the training documents or RL prompts—and a subset $\mathcal{A}_{heldout} \subset \mathcal{A}_{all}$ representing completions strictly held out from the training documents to evaluate true latent generalization (see Figure 4 in the appendix). Because the model’s output constitutes a set of strings, we evaluate performance using Precision, Recall, and F1 scores. Specifically, we report:

- **Overall Precision, Recall, and F1:** Computed with respect to the complete set of valid answers, $\mathcal{A}_{all}$.
- **Train Recall:** The recall evaluated strictly on the valid answers exposed during training, defined by the set difference $\mathcal{A}_{all} \setminus \mathcal{A}_{heldout}$.
- **Held-out Recall:** The recall evaluated exclusively on the held-out subset $\mathcal{A}_{heldout}$, serving as a direct measure of latent generalization.

- **Pass@N:** The rate at which the model’s predicted set exactly matches $\mathcal{A}_{all}$ within N independent attempts. This is a rigorous exact-match metric that yields a score of zero if the generated set contains any omissions or false inclusions.
- **Primary Reference Inclusion:** A binary indicator of whether the specific completion targeting the held-out condition (e.g., a specific reversal or syllogism) is present in the model’s output. This provides an additional proxy for true latent generalization.

Note that these metrics are inherently correlated; for instance, a model achieving a higher Overall F1 score is proportionally more likely to achieve a higher Pass@ N . However, it is possible for a model to obtain reasonable F1 score without necessarily predicting the completion which tests the latent-learning behavior, so it is useful to look at all these metrics independently. For the **Reversal Curse Dataset**, each query possesses a single unique correct completion; therefore, we exclusively report exact string-match accuracy.

3.4 Results

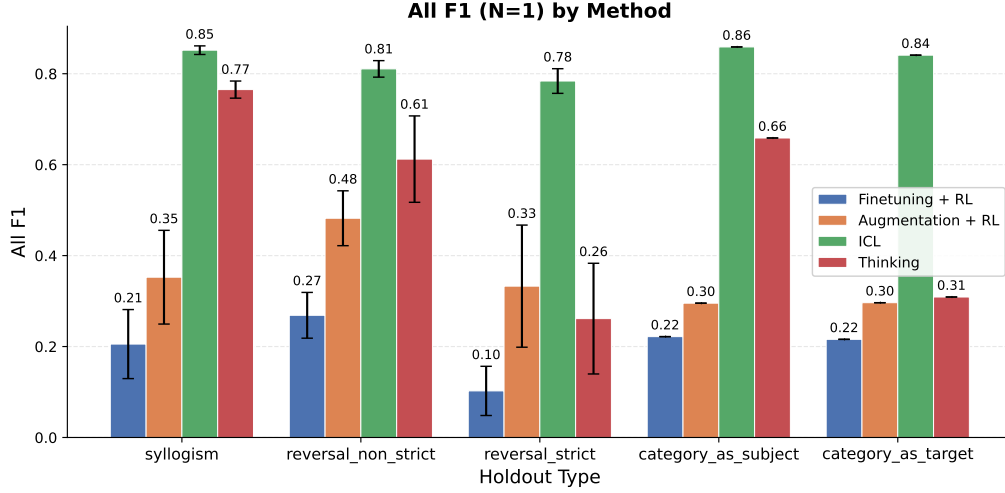
3.4.1 Semantic Structure Benchmark

Strengthening the Baselines Prior research, such as Lampinen et al. (2025a), evaluates latent generalization by computing the log-likelihood of a single ground-truth answer. In contrast, our evaluation protocol relies on a generative sampling procedure. We observe that under this generative setting, where the model must output the complete set of valid answers for a query, standard supervised fine-tuning (SFT)—even when supplemented with data augmentation—yields suboptimal performance. As it is well known in literature that reinforcement learning (RL) on its own can help factuality Tian et al. (2023), we apply RL to these baselines, optimizing a reward signal that explicitly incentivizes the correctness of all valid completions for the RL prompts. Figure 5 in the appendix demonstrates that this RL phase significantly enhances baseline performance. In subsequent analyses, we utilize these enhanced baselines “Fine-tuning + RL” and “Augmentation + RL.” and compare their performance against In-Context Learning (ICL) and our proposed methodology.

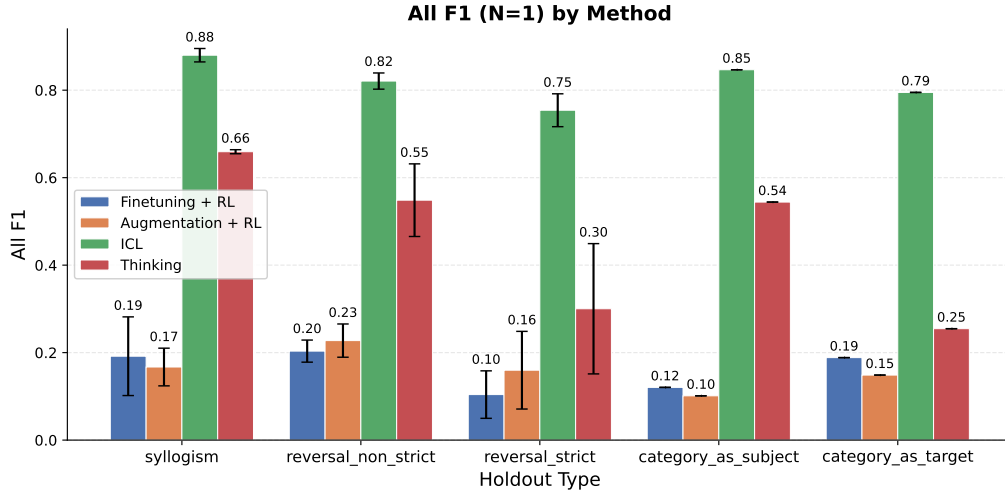
In-Distribution Latent Generalization Figure 2a shows performance for the in-distribution case, measured in terms of all F1 scores (plots for other metrics are available in the appendix in Figure 8, and point to similar conclusions). One can make several observations from this figure. First on holdouts for which the latent completions can be found by employing multi-hop reasoning, such as syllogisms, non-strict reversals and where category appears as a subject for different propositions in the dataset, thinking generalizes much better than the train-time augmentation strategy. On some of these holdouts, such as syllogisms, thinking achieves performance comparable to the ICL baseline, which is provided with the entire training dataset within its context window. We attribute this success to the model’s enhanced capacity to perform multi-hop reasoning within its intermediate traces, effectively composing multiple independent facts observed during training to deduce the correct latent answer.

On holdout conditions that inherently preclude multi-hop reasoning, such as strict reversals, the improvement of ‘Augmentation’ over fine-tuning and thinking on strict reversals is most likely due to the exact replication of test queries in the training set during the augmentation process. As we will show in the next section, this improvement of the ‘Augmentation’ baseline completely vanishes when being tested on the dataset for which no augmentation was performed. In fact, thinking outperforms the ‘Augmentation’ baseline substantially even on strict reversal in the out-of-distribution case.

Out-of-Distribution Latent Generalization We now evaluate out-of-distribution (OOD) latent generalization. In this setting, we measure performance on holdouts from a distinct dataset that was only available to the model during the initial knowledge acquisition phase; crucially, no train-time data augmentation or RL fine-tuning (with or without intermediate reasoning) was applied to this secondary dataset. We previously hypothesized that while train-time data augmentations are intrinsically tied to the specific training distribution and fail to generalize to novel datasets, training a model to utilize test-time compute to reason over its internal parametric knowledge achieves a task-agnostic latent generalization.



(a) In-Distribution Latent Generalization



(b) Out-of-Distribution Latent Generalization

Figure 2: **Semantic Structure Experiment:** F1 scores on in- and out-of-distribution hold-outs. Both Finetuning and Augmentation undergo RL training as described in Section 3.2. Thinking shows significant and consistent improvement over the baselines on both in- and out-of-distribution holdout splits, particularly for those splits that require multi-hop reasoning. Train-time augmentations completely fail to generalize on a new knowledge structures (out-of-distribution generalization).

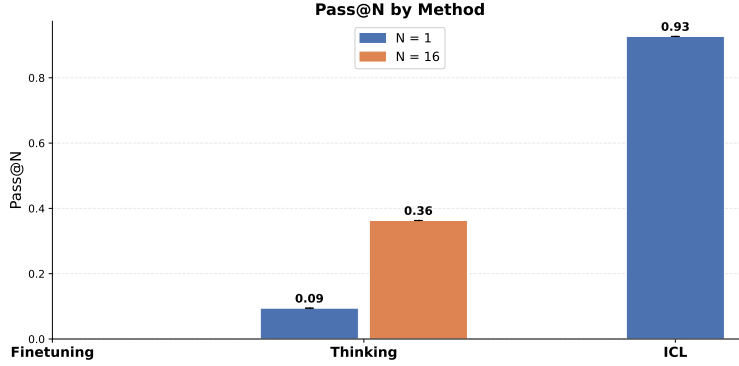


Figure 3: **Pass@N Accuracy:** Exact string match accuracy of reversal curse paper dataset. Finetuning completely fails to generalize even with multiple attempts. Thinking gives non-trivial generalization but overall for such strict reversals ICL is the strongest baseline.

Figure 2b presents the results for this OOD evaluation (only all F1 is reported here, other metrics can be found in Figures 8 and 9 where similar results hold). The empirical findings demonstrate that, in this regime, the data augmentation baseline yields no measurable improvement over standard supervised fine-tuning. The small improvement of the augmentation baseline on the strict reversals seen in the in-distribution case is completely gone once the potential leakage from the augmentation process is stopped by using a disjoint dataset. In stark contrast, our reasoning-based approach maintains OOD latent generalization performance comparable to its in-distribution results. On strict reversals, the performance of thinking, although better than augmentation, significantly trails ICL. This indicates that although intermediate reasoning enables LLMs to better elicit and generalize over their internalized parametric knowledge, this capability is fundamentally constrained when a query cannot be resolved by logically chaining multiple training facts. In other words, LLMs employing intermediate reasoning still struggle to execute direct, zero-hop logical reversals. We investigate reversal performance of thinking models in more detail on the Reversal Curse dataset (Section 3.4.2). **Overall, the out-of-distribution structure results confirms that leveraging test-time compute to dynamically elicit and reason over parametric knowledge is a more robust and flexible mechanism than relying on static train-time data augmentations, thereby validating the central hypothesis of this work.**

3.4.2 Reversal Curse Paper Dataset

We now present the results on the Reversal Curse dataset (Berglund et al., 2024). This dataset is analogous to the strict reversal condition in the Semantic Structure Benchmark: the training documents contain no intermediate reasoning chains that the model can exploit to deduce the correct answer. Instead, the model must directly invert the learned factual association. The corresponding results are illustrated in Figure 3. We omit the data augmentation baseline for this experiment, as augmenting the dataset with direct reversals would explicitly leak the test distribution during training. Consequently, we evaluate only the standard supervised fine-tuning (SFT), reasoning (ours), and ICL baselines.

As shown in the Figure 3, standard SFT fails to generalize, and scaling the number of parallel attempts (up to 16) yields no performance improvement. On the other hand, the model trained to generate intermediate reasoning traces exhibits measurable improvement, yet its performance remains inferior to that of ICL. An example of a successful reasoning trace generated by our model is provided in Appendix C.4. The trace reveals that the specifically encouraged generate-and-verify behavior in the golden traces helps the model as it proposes candidate entities acquired during the knowledge acquisition phase and subsequently self-verifies these candidates using its internalized parametric knowledge regarding each entity. Notably, this self-verification step does not require logical inversion, because the forward name-to-description mapping is explicitly encoded in the training data. If the retrieved description aligns with the target description provided in the query, the model terminates the reasoning process and outputs the verified entity.

However, this heuristic is not entirely reliable, as LLMs frequently exhibit brittle factual self-verification capabilities. In Table 2 and Table 1, we present an analysis of search depth and failures mode that commonly occur in the the thinking traces. We see that although the model engages deeply in search and self-verification of factual knowledge, robust factual verification remains one of the bottleneck to improve latent generalization on this task. Furthermore, the efficacy of this strategy is strictly bottlenecked by the requirement that the model possesses a non-zero probability of initially generating the correct entity as a hypothesis that undergoes verification. While scaling test-time compute via multiple attempts predictably increases the overall success rate, the pass@1 accuracy remains low. **Ultimately, these findings indicate that executing strict reversals derived purely from parametric knowledge remains a persistent challenge, even when models are augmented with intermediate reasoning capabilities.** ICL continues to be the most effective paradigm for resolving such direct logical inversions.

4 Related Work

Generalization from fine-tuning and latent learning: Several works look at some of the surprising ways in which models generalize from their dataset (Berglund et al., 2023; Meinke & Evans, 2023), or fail to generalize (Berglund et al., 2024; Allen-Zhu & Li, 2023; 2024) study the generalization abilities of fine-tuning in contrast to in-context learning. Borrowing terminology from (Blodgett, 1929; Tolman, 1948), the problem of latent learning is discussed in (Lampinen et al., 2025b), as one in which the model learns information that is not relevant to the task at hand, but that might be useful for a different future task. We operate in a similar train-test setup as some of these works studying generalization, but specifically focus on the ability of test-time compute to improve the latent learning ability of these models. We focus on the comparison of this test-time thinking strategy to other established methods like train-time augmentation.

Data augmentation to improve generalization: Data augmentation is a long-established solution to improving the generalization of models after fine-tuning. For specific instances of the latent learning problem such as the reversal curse, augmentation strategies, such as adding inverse inference to the training data (Golovneva et al., 2024) have been proposed in prior works. Knowledge augmentation is also extensively studied in (Allen-Zhu & Li, 2024), shedding light on the role of augmentation in both pre-training and fine-tuning phases and its impact on generalization. More general augmentation mechanisms were proposed in (Lampinen et al., 2025a), where model’s in-context inferences on a subset of the training data are added to the training dataset. They show that this improves performance at least on the knowledge structure where this augmentation is performed. Another variant proposed in (Park et al., 2025), leveraged similar in-context learning ability along with additional question-answers to improve model performance. A more comprehensive review on the topic of data augmentations to improve model performance is provided by (Chai et al., 2025; Ding et al., 2024). We compare our method to a similarly flavored augmentation to (Lampinen et al., 2025a) in our work.

Thinking to improve reasoning: The use of CoT thinking to greatly improve performance on reasoning tasks has been studied in several prior works (Lightman et al., 2023; OpenAI, 2024; DeepSeek-AI, 2025). We refer the reader to (Chen et al., 2025a) for a more comprehensive overview of this area. This line of research typically focuses on mathematical reasoning, science, and programming, and does not specifically look at how thinking can be leveraged to improve the use of factual knowledge stored in a model’s weights.

Reinforcement learning for factuality: Several works study how reinforcement learning in the post-training phase can greatly improve the factuality of language models in the absence of test-time compute (Tian et al., 2023; Lin et al., 2024). In our work, we leverage these findings by ensuring that both the train-time augmentation and fine-tuning baselines undergo RL. On the other hand, for methods that explicitly use test-time compute (thinking) Allen-Zhu & Li (2023) study the utility of chain-of-thought in the ability of a model to access and manipulate its parametric knowledge, finding that chain-of-thought is critical

for utilizing knowledge even if the knowledge is encoded perfectly. For the specific case of the reversal curse (termed inverse knowledge search), they find that language models cannot perform this task even with chain-of-thought. In our work, we find that when sufficient support exists in the data, or if the model can efficiently leverage self-verification, the task of reversal might be tractable. (Sun et al., 2022) propose a method where the model is encouraged to recite relevant passages from its pre-training data before producing an answer, acting as a form of chain-of-thought for factual knowledge. Chen et al. (2025b) focus on improving long-form factuality using reasoning, where the model can make a tradeoff between specificity and recall in various claims of a response. Closely related to our work is concurrent work by Ma & Hewitt (2026) and Zhang et al. (2025). Ma & Hewitt (2026) study how chain-of-thought can help factuality on single-hop questions such as those found in SimpleQA (Wei et al., 2024). Zhang et al. (2025) find that reinforcement learning and reasoning improves parametric knowledge traversal on a wide variety of datasets, and can improve knowledge access for rare facts unseen in training. Another closely related concurrent work by Gekhman et al. (2026) studies the effect of reasoning on single-hop factual queries. They find that encouraging thinking can greatly improve pass@k without any training, and identify two key drivers behind the mechanism: thinking about related topics (factual priming), and the ability of the model to leverage additional thinking-token compute independent of the actual thinking content.

Our work distinguishes ourselves in that we control the facts the model sees during training using our synthetic datasets, and test on facts which are latent in the dataset and not explicitly stated. Unlike some of the prior work which care about factuality with respect to the direct knowledge the model is exposed to during training, we also carefully study factuality with respect to facts which are latent and not directly stated in the data. The structure of our data enables us to make fine-grained conclusions about the generalization of various methods on difficult subsets, such as strict reversals. We also compare the efficacy of test-time compute (thinking) vs. train-time compute (augmentation) on these subsets.

5 Discussion & Conclusion

Recent advances in test-time scaling, often implemented via intermediate reasoning or “thinking” traces, have enhanced the reasoning capabilities of Large Language Models (LLMs). We investigate whether models can leverage this additional test-time compute to improve latent generalization whereby they can resolve deductive reasoning challenges, such as syllogisms and logical reversals, when generating responses exclusively from parametric knowledge. Through controlled experiments, we demonstrate that LLMs can indeed utilize test-time compute to enhance latent generalization, proving to be a more flexible and domain-agnostic strategy than relying on static train-time data augmentations.

Despite this success, strictly pure logical reversals remain elusive, even for models augmented with intermediate reasoning and self-verification. We hypothesize this limitation stems from the rigid key-value associative memory structures that models acquire as a consequence of the transformer architecture and the autoregressive next-token prediction objective. The reversal curse and related failures of latent generalization highlight fundamental deficiencies in the contemporary LLM paradigm, encompassing both the underlying architecture and the optimization objective. While scaling compute, whether during training (e.g., via data augmentation) or at inference time (e.g., via reasoning traces), may address some of these limitations, we believe exploring stronger inductive biases, novel architectures, alternative loss functions, or a combination thereof in parallel could provide a more efficient path to obtaining better models.

References

- Afra Feyza Akyürek, Ekin Akyürek, Leshem Choshen, Derry Wijaya, and Jacob Andreas. Deductive closure training of language models for coherence, accuracy and updatability. *Findings of the Association for Computational Linguistics*, 2024.
- Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation. *arXiv preprint arXiv:2309.14402*, 2023.
- Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction, 2024. URL <https://arxiv.org/abs/2309.14316>.
- Lukas Berglund, Asa Cooper Stickland, Mikita Balesni, Max Kaufmann, Meg Tong, Tomasz Korbak, Daniel Kokotajlo, and Owain Evans. Taken out of context: On measuring situational awareness in llms. *arXiv preprint arXiv:2309.00667*, 2023.
- Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain 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.
- Hugh Carlton Blodgett. The effect of the introduction of reward upon the maze performance of rats. *University of California publications in psychology*, 1929.
- Yaping Chai, Haoran Xie, and Joe S. Qin. Text data augmentation for large language models: A comprehensive survey of methods, challenges, and opportunities, 2025. URL <https://arxiv.org/abs/2501.18845>.
- Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models, 2025a. URL <https://arxiv.org/abs/2503.09567>.
- Xilun Chen, Ilia Kulikov, Vincent-Pierre Berges, Barlas Oğuz, Rulin Shao, Gargi Ghosh, Jason Weston, and Wen-tau Yih. Learning to reason for factuality. *arXiv preprint arXiv:2508.05618*, 2025b.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1679–1705, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.97. URL <https://aclanthology.org/2024.findings-acl.97/>.
- Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.
- Zorik Gekhman, Roei Aharoni, Eran Ofek, Mor Geva, Roi Reichart, and Jonathan Herzig. Thinking to recall: How reasoning unlocks parametric knowledge in llms. *arXiv preprint arXiv:2603.09906*, 2026.
- Gemini. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL <https://arxiv.org/abs/2507.06261>.
- Olga Golovneva, Zeyuan Allen-Zhu, Jason Weston, and Sainbayar Sukhbaatar. Reverse training to nurse the reversal curse. *arXiv preprint arXiv:2403.13799*, 2024.

- Andrew K Lampinen, Arslan Chaudhry, Stephanie CY Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study. arXiv preprint arXiv:2505.00661, 2025a.
- Andrew Kyle Lampinen, Martin Engelcke, Yuxuan Li, Arslan Chaudhry, and James L McClelland. Latent learning: episodic memory complements parametric learning by enabling flexible reuse of experiences. arXiv preprint arXiv:2509.16189, 2025b.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. arXiv preprint arXiv:2305.20050, 2023.
- Sheng-Chieh Lin, Luyu Gao, Barlas Oguz, Wenhan Xiong, Jimmy Lin, Wen-tau Yih, and Xilun Chen. Flame : Factuality-aware alignment for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), Advances in Neural Information Processing Systems, volume 37, pp. 115588–115614. Curran Associates, Inc., 2024. doi: 10.52202/079017-3671. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d16152d53088ad779ffa634e7bf66166-Paper-Conference.pdf.
- Melody Ma and John Hewitt. Improving parametric knowledge access in reasoning language models. arXiv preprint arXiv:2602.22193, 2026.
- Alexander Meinke and Owain Evans. Tell, don’t show: Declarative facts influence how llms generalize. arXiv preprint arXiv:2312.07779, 2023.
- OpenAI. Openai o1 system card. arXiv preprint arXiv:2412.16720, 2024.
- Core Francisco Park, Zechen Zhang, and Hidenori Tanaka. New news: System-2 fine-tuning for robust integration of new knowledge. arXiv preprint arXiv:2505.01812, 2025.
- Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms, 2025. URL <https://arxiv.org/abs/2505.00127>.
- Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. Recitation-augmented language models. arXiv preprint arXiv:2210.01296, 2022.
- Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. In The Twelfth International Conference on Learning Representations, 2023.
- Edward C Tolman. Cognitive maps in rats and men. Psychological review, 55(4):189, 1948.
- Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. arXiv preprint arXiv:2411.04368, 2024.
- Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3):229–256, 1992.
- Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candes, and Tatsunori Hashimoto. Synthetic continued pretraining. arXiv preprint arXiv:2409.07431, 2024.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. Advances in Neural Information Processing Systems, 35:15476–15488, 2022.
- Renfei Zhang, Manasa Kaniselman, and Rylan Schaeffer abd Niloofar Mireshghallah. Reinforcement learning improves traversal of parametric knowledge in llms. arXiv preprint arXiv:2511.05933, 2025.