ON THE INDUCTIVE REASONING CAPACITY OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The rapid advancement of Large Language Models (LLMs) has sparked attention to their reasoning capacities. In this work, we investigate the underlying mechanics of inductive reasoning in LLMs and discover that internal hidden embedding trajectories capture the respective rule acquisition dynamics. We show that hidden-layer velocity functions serve not only as a strongly correlated signal of the underlying logic, but also as a causal lever capable of steering model behavior during reasoning. Building on this discovery, we introduce a contrastive fine-tuning framework that directly optimizes these representation trajectories to enhance rule inference. Comprehensive evaluation across 8 benchmarks demonstrates the effectiveness of our approach, improving LLM inductive reasoning accuracy by up to 23 percentage points in in-domain generalization and up to 8.6 percentage points in cross-domain generalization.

1 INTRODUCTION

A significant portion of research on large language models (LLMs) revolves around the reasoning behavior and performance. Early models were mostly judged on surface-level tasks like text completion or sentiment analysis, but modern scaling has enabled multi-step problem solving, symbolic manipulation, and complex rule navigation Wei et al. (2022a); Yao et al. (2025). To make sense of these behaviors, existing literature largely divides model capabilities into two formal logical pillars: deduction Li et al. (2026) and induction Flach & Kakas (2000).

In deductive reasoning, models draw necessary conclusions from explicit premises, such as formal math proofs Cabral et al. (2025); Ahn et al. (2024) or code reasoning Liu et al. (2023); Jiang et al. (2024). Supported by techniques like chain-of-thought prompting that externalize intermediate steps, LLMs perform remarkably well in this domain Wei et al. (2022b); Ye et al. (2025). Because explicit premise-to-conclusion paths leave clear internal traces, extensive literature has successfully mapped out how model representations handle deductive tasks Zhou et al. (2026), establishing deduction as a well-understood frontier.

For the case of inductive reasoning, the model is tasked with extracting general governing rules, which are often implicit, from limited observations and inferring the prospective examples. While LLMs often appear to "generalize," it is difficult to separate true inductive rule-learning from simple memorization or retrieval from training data Dong et al. (2024). Notably, the authors of Jin et al. (2026) have found that while LLMs excel at inferring commonly known rules, they struggle on tasks where the rules are specially and sophisticatedly crafted. These findings hint that LLMs may resort to memorization rather than actually reasoning on the tasks. Understanding whether models actually infer abstract underlying rules or merely pattern-match surface similarities is crucial for determining how reliably they can handle novel domain tasks.

Despite the growing evaluation efforts, the studies on inductive reasoning in LLMs remain relatively under-explored in terms of underlying behaviors and improving methods. Some are limited to only benchmark tasks, testing whether models can follow step-by-step rules to arrive at a fixed answer Suzgun et al. (2023); Cobbe et al. (2021). Some other work on inductive reasoning of LLMs only offers high-level surveys regarding the benchmarks or some preliminary approaches Srivastava et al. (2023); Chen et al. (2025). While identifying reasoning patterns in purely deductive contexts has been explored Zhou et al. (2026), the counterpart exploration on inductive reasoning is largely absent.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this work, we bridge this gap by peering directly into the internal mechanics of LLMs during inductive reasoning. By extracting the hidden layers of LLMs as the models progress through inductive reasoning tasks, we investigate the latent patterns that drive the inductive reasoning process of LLMs. We observe that the velocity of hidden representations exhibits remarkably high directional consistency across samples sharing the same underlying logic, regardless of the surface semantics. Furthermore, our steering experiments confirm that the velocity acts as a causal driver of the model's underlying inductive logic. Grounded in these observations, we introduce a contrastive learning fine-tuning framework that aligns the trajectory velocities across instances sharing the same underlying logic. This approach leverages the insights and validates the usefulness of the trajectory velocity, providing an improvement of up to 23% points in performance over the base model.

In short, the key contributions of our work are as follows:

- Observations: We identify hidden representation velocity as a key signal in LLM inductive reasoning, establishing both its correlation with logical task structures and its causal influence on model behavior.
- Method: Exploiting these insights, we devise a contrastive self-supervised fine-tuning framework that explicitly aligns internal representation trajectories to foster robust rule generalization. The method boosts the inductive reasoning capacity of LLMs by up to 23% points in the in-domain and 8.6% points in the cross-domain settings.
- Evaluation: We conduct experiments over 8 benchmarks to show that optimizing these dynamics boosts LLM reasoning performance on both in-distribution tasks and unseen rules.

2 Related Work

2.1 Inductive Reasoning of LLMs

Inductive reasoning refers to the ability to generalize from particular observations to a universal rule governing the observations. The inductive reasoning capacity of LLMs can be measured as the performance in in-context learning Brown et al. (2020), where LLMs are provided with several examples and prompted to infer the underlying rule that describes the several examples. Some other inductive reasoning tasks include linguistic induction Ramji & Ramji (2025) and hypothesis search He & Chen (2025). Recently, researchers have pinpointed that, albeit excellent at inferring common rules, reasoning models tend to struggle with inducing in uncommon tasks Jin et al. (2026), where the underlying rules are more sophisticated. Summarizing the current progress regarding the inductive reasoning capacity of LLMs, a recent survey has provided a broad report on current approaches, tasks, and benchmarks Chen et al. (2025). While a line of work has offered some preliminary investigation on the underlying mechanism driving the in-context learning of LLMs. Olsson et al. (2022); Crosbie & Shutova (2025); Elhage et al. (2021), through the concept of "induction head", the inductive reasoning behavior of LLMs remains largely unexplored.

2.2 Understanding behavior of LLM reasoning

Lots of attention has been paid to interpreting and understanding the reasoning behaviors of Large Language Models. Some work has demonstrated that inference-time scaling is capable of boosting LLM performance on complex tasks Snell et al. (2025). On the theoretical side, this is often explained by the expanded capacity for serial computation Li et al. (2024), superposed reasoning trajectories Zhu et al. (2025), or implicit planning during generation Ye et al. (2025). On the other hand, some work investigates the latent behavior of LLMs when processing through deductive reasoning Zhou et al. (2026). Other work analyzes the underlying graph dynamics of these reasoning chains Minegishi et al. (2025) or isolates critical "thought anchors" that drive the model's decisions Bogdan et al. (2025). However, it was also discovered that accuracy can degrade if the reasoning chain becomes too long Wu et al. (2025). Building on these behavioral insights, recent methods use steering vectors Venhoff et al. (2025) or manifold constraints Huang et al. (2025) to guide the model's internal path, preventing overthinking and enabling earlier termination Fu et al. (2024).

2.3 Post-training Adaptation Methods

Adapting Large Language Models typically relies on alignment training or (self-)supervised fine-tuning. Alignment techniques Bai et al. (2022); Ouyang et al. (2022); Guo et al. (2025), such as preference optimization Rafailov et al. (2023), excel at aligning model behavior to explicit human guidelines but require explicitly expressing human preferences through a training objective. This is typically carried out at the later training stage. More fundamentally, self-supervised fine-tuning He et al. (2022); Hsu et al. (2021); Oquab et al. (2023) leverages intrinsic characteristics to guide the behavior of deep learning models, offering more flexibility without human intervention. A line of work has focused on fine-tuning LLMs with regard to generalization Gupta et al. (2025) or self-alignment Chuang et al. (2025); Sun et al. (2023).

Most closely related to our approach is the work by Zhou et al. (2026). They explore a relationship between deductive logic and the surface semantics with the trajectory of hidden-layer representations, through theoretical and empirical analysis. We not only transition the investigation to inductive reasoning but also make a further step in solidifying the relationship between the underlying inductive logic and velocity of the hidden representations (see Section 3 for details). In particular, we illustrate velocity as a correlational signal and, more importantly, a causal lever to induce the inductive logic (Section 4) through intervention. Furthermore, we demonstrate the utility of this signal by devising a velocity-based training method that yields consistent improvement in inductive reasoning (Section 5).

3 Background

3.1 Preliminaries

In this section, we provide an overview of the fundamental concepts and main focus of this work.

In-context Learning: In-context learning (ICL) Brown et al. (2020) requires LLMs to infer an underlying rule from several observations. Formally, given examples $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{k}$ governed by a latent function $\mathbf{y}_i = f(\mathbf{x}_i)$, and a query $\mathbf{x}_{k+1}$, the model must implicitly induce f to predict:

$$\mathbf{y}_{k+1} = \arg\max_{\mathbf{y}} P(\mathbf{y} \mid \mathcal{D}, \mathbf{x}_{k+1})$$

In this work, we test the LLM's ability to generalize from observations and infer the underlying governing rule.

Contrastive Learning: Contrastive learning (CL) Chen et al. (2020) trains an encoder to map same-label inputs (positives) close together in an embedding space while pushing different-label inputs (negatives) apart. One common training objective for CL is the InfoNCE loss Gutmann & Hyvärinen (2010):

$$\mathcal{L} = -\log \frac{\exp(\mathbf{z} \cdot \mathbf{z}^{+}/\tau)}{\exp(\mathbf{z} \cdot \mathbf{z}^{+}/\tau) + \sum \exp(\mathbf{z} \cdot \mathbf{z}^{-}/\tau)}$$

where $\mathbf{z}$, $\mathbf{z}^{+}$, and $\mathbf{z}^{-}$ represent normalized anchor, positive, and negative embeddings, and τ is a temperature parameter.

Low-Rank Adaptation: Low-Rank Adaptation (LoRA) Hu et al. (2022) is a parameter-efficient fine-tuning method for LLMs. Instead of re-optimizing all the parameters, LoRA freezes the pre-trained model weights $\mathbf{W}_0 \in \mathbb{R}^{d \times k}$ and injects trainable rank decomposition matrices to track weight updates. For a forward pass $\mathbf{h} = \mathbf{W}_0\mathbf{x}$, it introduces a low-rank parameterization $\Delta\mathbf{W} = \mathbf{BA}$, where $\mathbf{B} \in \mathbb{R}^{d \times r}$ and $\mathbf{A} \in \mathbb{R}^{r \times k}$ with rank $r \ll \min(d, k)$. The updated hidden state calculation is expressed as: $\mathbf{h} = \mathbf{W}_0\mathbf{x} + \frac{\alpha}{r}\mathbf{BAx}$, where α is a hyperparameter that modulates the magnitude of the adapter's structural update.

3.2 Key Objectives

In this paper, we answer the following research questions:

- Q1. What are the relevant signals driving the inductive reasoning process of LLMs?
- Q2. How to exploit such signals to enhance the inductive reasoning capacity of LLMs?
- Q3. How much do the signals boost the reasoning capacity of LLMs?

4 OBSERVATIONS

In this section, we address Q1 by establishing hidden-layer representation velocity as a key mechanism underlying rule acquisition dynamics in inductive reasoning. Specifically, we investigate how LLMs' hidden-layer representations evolve when performing in-context learning tasks and characterize their relationship with the underlying inductive logic. We find a robust relationship between representation velocity and the inductive logic, suggesting that velocity can serve not only as a correlational marker but also as a causal lever for shaping reasoning trajectories. With the usefulness of the velocity established, we leverage it to develop a training scheme in Section 5.

4.1 EXPERIMENTAL DETAILS

In-context Learning Task: We construct an inductive reasoning task based on pattern transformation. In particular, given an underlying transformation rule/instruction $\mathbf{R}$ in each task, we provide a sequence $\mathcal{S}$ of T strings/numbers following $\mathbf{R}$. An LLM $\mathcal{L}$ is prompted to process the sequence $\mathcal{S}$. Note that $\mathbf{R}$ is unknown to $\mathcal{L}$, and $\mathcal{L}$ is tasked to implicitly induce the underlying rule and predict $\mathcal{S}_{T+1}$, the next occurrence/outcome.

Example Processing: For each transformation rule $\mathbf{R}$, we generate several instances, each of which is a sequence of strings/numbers. For investigation, we group instances based on bearing the same underlying rule $\mathbf{R}$ or group those of the same alphabet/starting point/language (different underlying rules). For clarification, instances bearing the same transformation rule are regarded as holding the same logic structures, while those of the same alphabet/starting point/language are considered bearing the same semantics.

Mechanics: We feed each instance $\mathcal{S}$, a sequence of strings/numbers bearing rule/instruction $\mathbf{R}$, into a LLM $\mathcal{L}$ and extract the hidden layer embeddings of $\mathcal{L}$. We measure the *velocity* of the embeddings as $\mathcal{L}$ iterates through each string. For the t-th string $\mathcal{S}_t$ for $t = 1, ..., T$, $\mathcal{S}_t$ ends with a line-break character '$\backslash$n' and we extract the hidden representations $\mathbf{h}^{(L)}_{\backslash \mathrm{n},} \in \mathbb{R}^d$ for the line-break character at layer L. Let $\mathbf{y}_t$ be $\mathbf{h}^{(L)}_{\backslash \mathrm{n},}$.

The ordered sequence of step vectors $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ is the *latent trajectory* of the model's inductive reasoning. The velocity vector of the instance is computed as $v_{\mathbf{y}} = (\mathbf{y}_2 - \mathbf{y}_1, ..., \mathbf{y}_T - \mathbf{y}_{T-1})$. We collect the velocity vectors for investigations: similarity computation and steering.

Orthogonal Alignment: To rule out rotation effects, we also apply orthogonal alignment between two distributions of velocity vectors before comparing any 2 distributions. Details are in Appendix A.1. This alignment further highlights velocity as a strong correlational signal for inductive reasoning, as demonstrated in Section 4.2.

Specifications: For LLMs, we investigate Qwen 3 8B and Llama 8B. We investigate several benchmarks having sequences following transformation/instruction rules: Bit string transformations such as Rule-30 Wolfram (1983) (string transformation), NumSeqBench Tsui (2024) (numerical), and Crypto Li et al. (2025) (natural language). Further details are in Appendix A.2.

Investigations: To demonstrate correlation between velocity and logic, we measure the pairwise similarity of velocity vectors among instances of the same or different groups and collect the respective distributions for the bitstring transformations. To further show that velocity acts as a causal lever of reasoning, we also steer hidden-layer activations along the velocity and observe how the model behavior shifts for the benchmarks.

4.2 OBSERVATIONS

Velocity as a correlational signal of Logic: We collect the distribution of pairwise similarities of velocity vectors and compare the case when grouped differently. Specifically, we group instances either by underlying logic or by semantics. For each grouping, we compare the distribution of pairwise velocity similarity within the same group ("within_group") against that across different groups ("between_group"). Both distributions are computed before ("No Alignment") and after ("Alignment") orthogonal alignment. Figures 1 and 2 illustrate the results for Qwen-3. All the remaining results can be found in Appendix A.3.

It is clear from the plots that when grouped by semantics, velocity vectors exhibit no clear pattern, regardless of alignment. By contrast, vectors sharing the same underlying logic align closely, and this similarity becomes starkly apparent once rotational effects are ruled out. This observation reinforces that representation velocity carries a strong signal for the logic processing of LLMs.

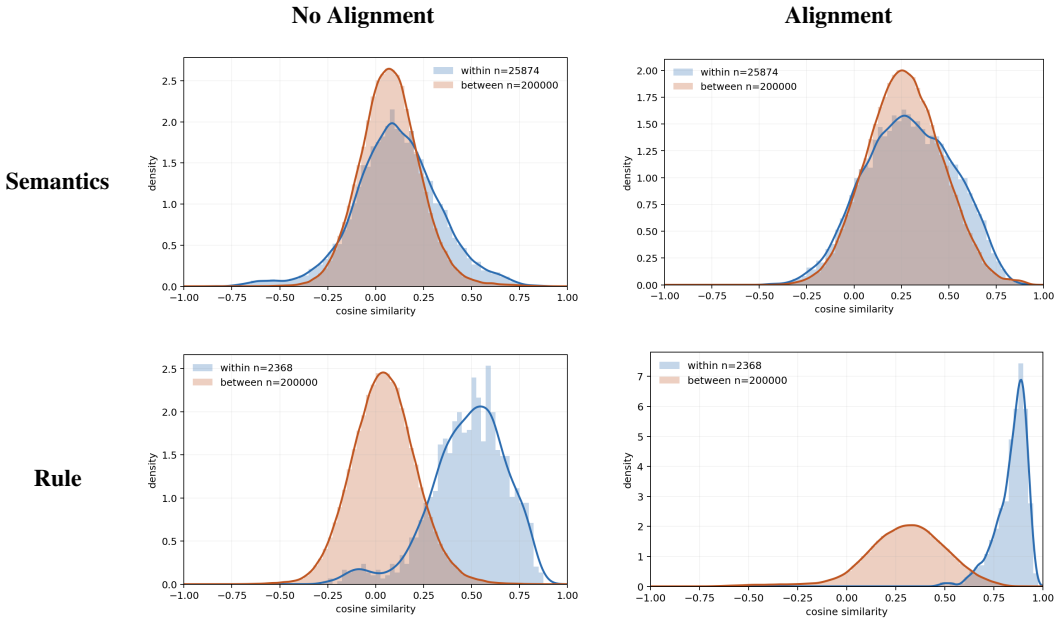

Figure 1: Early step: Similarity distribution of velocity vectors when grouped by semantics vs. by underlying logic with(out) orthogonal alignment. Despite producing no clear velocity patterns by semantics, the instances of the same logic demonstrate highly similar velocity vectors. This similarity becomes more apparent when rotation effects are ruled out by alignment.

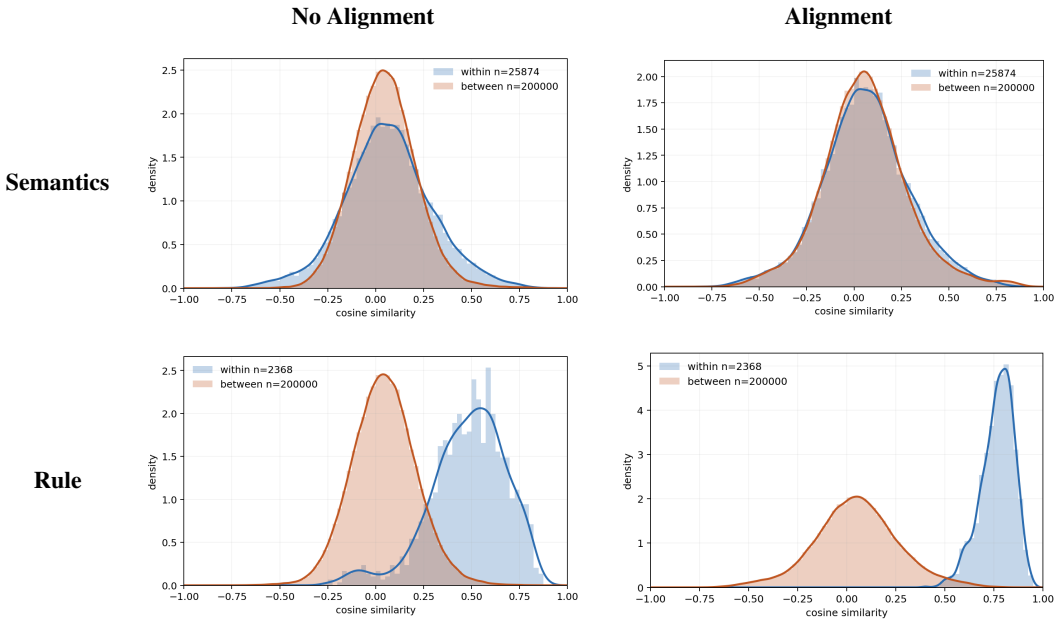

Figure 2: Latter step: Similarity distribution of velocity vectors when grouped by semantics (top row) vs. by underlying logic (bottom) with(out) orthogonal alignment. While producing no clear velocity patterns by semantics, the instances of the same logic demonstrate highly similar velocity vectors. This similarity is more apparent when rotation effects are ruled out by alignment.

Velocity as a causal lever of reasoning: To further investigate velocity as an intervenable mechanism, we perform activation steering on hidden representations. Given hidden-layer representation $\boldsymbol{h}$ of occurrence $\mathcal{S}_t$, we apply steering along vector $\hat{\boldsymbol{u}}$ and employ that to predict occurrence $\mathcal{S}_{t+1}$ via either (i) affine steering: $\hat{\boldsymbol{h}} \leftarrow \boldsymbol{h} + \hat{\boldsymbol{u}}$ or (ii) ablation: $\hat{\boldsymbol{h}} \leftarrow \boldsymbol{h} - (\boldsymbol{h}.\hat{\boldsymbol{u}})\hat{\boldsymbol{u}}$. Vector $\hat{\boldsymbol{u}}$ is either the correct trajectory velocity vector $\boldsymbol{v}$ or an incorrect velocity vector $\boldsymbol{i}$ or a random vector $\boldsymbol{r}$. To avoid any potential leakage, for each sequence $\mathcal{S}$, we obtain the correct velocity $\boldsymbol{v}$ from another sequence $\mathcal{S}'$ bearing the same rule and incorrect velocity $\boldsymbol{i}$ from another sequence $\bar{\mathcal{S}}$ bearing a different rule.

We report the prediction accuracy of such steering mechanisms for Qwen-3 in Table 1. The results for Llama are in Appendix A.3. It is clear that steering hidden states along the target velocity direction consistently yields the highest predictive accuracy across tasks. Conversely, counter-directional steering or ablating the velocity vectors leads to a sharp drop in performance, causing the model to systematically yield incorrect predictions. These results confirm that representation velocity functions as an active causal lever shaping the model's internal reasoning behavior.

Table 1: Prediction accuracy when steering vs. ablating along the correct/incorrect/random directions. The numbers are mean and standard deviation of 5 trials, in Pass@10. Steering representations along the target velocity vector leads the model to more accurate predictions, whereas steering in opposing directions or ablating velocity significantly increases prediction errors. This demonstrates that representation velocity functions as a causal lever in model reasoning.

Benchmark	**Steering**			**Ablation**	
	Velocity	**Incorrect**	**Random**	**Velocity**	**Random**
Rule-30	50.23 ± 3.37	20.39 ± 3.16	26.14 ± 5.87	16.28 ± 2.71	27.55 ± 5.89
NumSeq	42.58 ± 2.76	15.73 ± 2.82	19.73 ± 3.72	15.05 ± 2.46	17.94 ± 4.15
Crypto	36.67 ± 2.83	14.67 ± 3.06	17.68 ± 4.19	13.31 ± 2.42	18.13 ± 3.77

5 EXPERIMENTS

Building on the insights shown in Section 4, we attempt to answer Q2 and Q3. In particular, we utilize the trajectory velocity to drive the inductive reasoning of LLMs. To this end, we introduce a contrastive fine-tuning framework designed to elevate the inductive reasoning capacity of LLMs. Our experiments demonstrate that guiding the model's velocity space yields dramatic gains—boosting performance by up to 23% points in in-domain and 8.6% points in cross-domain settings.

5.1 BOOSTING INDUCTIVE REASONING CAPACITY

This section summarizes our key hypothesis and method.

Hypothesis: Section 4 shows that the velocity of hidden embeddings carries a strong correlated signal and acts as a causal lever for the underlying logic structure that the LLM reasons through. Our hypothesis is that by training the LLM to recognize the velocity pattern, one guides the model to follow the right reasoning trajectory, which in turn helps LLM reach the right destination of the hidden embedding. To that end, the model implicitly infers the underlying rule and reaches the right destination of the reasoning path, enabling it to predict the final step of the sequence correctly.

Fine-tuning with Contrastive Learning:

We devise a contrastive learning fine-tuning method to enable LLMs to recognize the logic structure, follow the right velocity when processing an example, and finally reach the right destination, i.e., the accurate prediction of the last step. That is, we train LLMs so that instances of the same underlying logic structure will have similar velocity vectors. While those of different logic structures will have dissimilar velocity vectors.

The training objective is the InfoNCE van den Oord et al. (2018), which is a multiclass adaptation of the noise-contrastive estimation framework of Gutmann & Hyvärinen (2010), a contrastive objective over velocity vectors $\mathbf{v} \in \mathbb{R}^d$. For each "anchor" instance a, let $\mathcal{P}(a)$ denote the set of instances sharing the same underlying logic structure with anchor a, the set of "positives". Denote $\mathcal{N}(a)$ the

set of instances bearing logic structures different from that of a, the set of "negatives". The objective trains LLMs so that every instance $p \in \mathcal{P}(a)$ will have a velocity vector $\mathbf{v}_p$ highly similar to the velocity vector $\mathbf{v}_a$ of a, whereas every instance $n \in \mathcal{N}(a)$ will have velocity vectors dissimilar to that of a.

The training loss component with respect to anchor a is:

$$\mathcal{L}_a = \frac{1}{|\mathcal{P}_a|} \sum_{p \in \mathcal{P}_a} -\log \frac{\exp\left(\frac{\text{sim}(\mathbf{v}_a, \mathbf{v}_p)}{\tau}\right)}{\exp\left(\frac{\text{sim}(\mathbf{v}_a, \mathbf{v}_p)}{\tau}\right) + \sum_{n \in \mathcal{N}_a} \exp\left(\frac{\text{sim}(\mathbf{v}_a, \mathbf{v}_n)}{\tau}\right)} \tag{1}$$

, where $\tau > 0$ is the temperature hyperparameter, and $\text{sim}(\cdot, \cdot)$ measures the mean cosine similarity between two velocity vectors.

Let $\mathcal{A}$ denote the set of anchor instances. The total contrastive learning training objective is formulated as a summation over all anchor instances:$\mathcal{L} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \mathcal{L}_a$.

5.2 Experimental Settings

In-context Learning Benchmarks: In each benchmark, the model is provided several examples following an underlying logic rule. The model is tasked with inducing such an underlying rule and inferring the new outcome of applying such rule on a new input. Details of the benchmark are found in Appendix A.1.

Specifications: We provide further details on our models and training scheme

- Models: for our experiments, we consider Qwen-3 8B and Llama 3.1 8B.
- Low-rank adaptation: to enable parameter-efficient training for the contrastive learning objective, we apply LORA rank 16 for our training.
- Variants: to evaluate the enhancement afforded by the proposed contrastive learning fine-tuning, we compare the base and the fine-tuned model. That is, we compare 2 variants of Qwen-3, which are the base model (Qwen-3 Base) and the model after the contrastive learning fine-tuning step (Qwen-3 CL). The same applies to Llama (Llama Base and Llama CL).

Baselines: We include the following baseline supervised fine-tuning (SFT) methods, for each LLM, in our comparison:

- Standard SFT (Standard): it fine-tunes the same LoRA with next-token cross-entropy on the in-context examples $s_2, \ldots, s_m$ given their preceding examples.
- Solution-only SFT (Sol): given the sequence $\mathcal{S}[1:T]$, the model is trained on generating solution $\mathcal{S}_T$ from the input $\mathcal{S}[1:T-1]$.
- Static CL (Static): the model is trained with contrastive learning on the hidden-layer representations. This ablates the velocity component in the training objective.

Competitors: We compare our method with two fine-tuning methods that also aim to enhance the reasoning capacity of LLMs.

- Coconut Hao et al. (2025): it replaces chain-of-thought tokens with the model's own hidden states fed back as continuous thoughts, trained via a curriculum over CoT traces and answers. It performs $(T-1)$ latent reasoning steps followed by a single answer step.
- CODI Shen et al. (2025): it self-distills explicit chain-of-thought into a fixed number of latent tokens by aligning teacher and student hidden states, requiring CoT traces and answer labels. We train CODI on the same T-step sequences, jointly optimizing explicit CoT and implicit continuous-thought objectives with self-distillation at the answer position.

Evaluation Settings: We evaluate each method in the following settings:

- In-domain: for each benchmark, each method is trained on a set of instances and evaluated on *held-out* rules, i.e., predicting the next value of an *unseen* sequence bearing *untrained* rules.

- Compositional: for each benchmark, each method is trained on original instances bearing an individual primitive transformation rule $\mathbf{R}_i (i = 1, ...)$ and tested on augmented instances where each transformation rule is a composition of two primitive rules $\mathbf{R}_i \circ \mathbf{R}_j$.
- Close-domain and Out-of-domain: for the math benchmark family, each method is trained on FACT and evaluated on close-domain benchmarks (NumSeq, OEIS and Riddle) and out-of-domain benchmarks (BBH and BBEH). Their details are in Appendix A.1.
- Semantic Mapping: for the text benchmark family, the evaluation samples are not only held-out but also obtained through an alphabet bijection/linguistic translation from the original dataset. This is to test whether LLMs can be trained to genuinely recognize the logic, not just semantics.

The results reported are the mean and standard deviation of Pass@1 accuracy from 5 iterations. Due to page limits, we present the results for Qwen-3 here and those for Llama in Appendix A.4.

5.3 In-Domain and Compositional Results

We present the evaluation results in Table 2 for the held-out in-domain setting and Table 3 for the compositional setting. Overall, the method Qwen-3 CL is consistently superior in accurately inducing the underlying rule and predicting the correct answer. particularly, the contrastive learning fine-tuning scheme offers a boost of up to 23% points over the base model and up to 10% points over the strongest competitor.

Table 2: Comparison of methods over the benchmarks in the in-distribution held-out setting. We report the mean and standard deviation of Pass@1 accuracy across 5 independent runs. It is clear that Qwen-3 CL is overall the superior method, offering an improvement of up to 23% points over the base model and up to 10% points over the strongest competitor.

	Benchmark							
Method	**List Func.**	**FACT**	**NumSeq**	**OEIS**	**Crypto**	**SyGus**	**Induction**	**Instruct.**
Qwen-3 Base	22.73 ± 1.46	27.62 ± 1.37	41.95 ± 2.37	28.34 ± 2.06	35.15 ± 1.63	50.27 ± 3.51	30.64 ± 2.26	36.82 ± 1.68
Standard	23.58 ± 1.91	21.86 ± 1.72	39.63 ± 2.35	29.17 ± 1.83	35.39 ± 1.54	55.68 ± 3.92	33.48 ± 3.01	38.27 ± 2.24
Sol	21.46 ± 2.13	24.39 ± 1.64	38.73 ± 2.80	32.41 ± 1.98	38.45 ± 1.79	51.47 ± 3.18	30.98 ± 2.77	36.38 ± 1.89
Static	26.92 ± 1.74	28.57 ± 1.89	43.67 ± 2.48	34.55 ± 2.14	37.41 ± 1.55	54.92 ± 4.05	34.57 ± 1.89	37.73 ± 1.95
Coconut	28.23 ± 1.85	29.48 ± 1.76	43.83 ± 3.07	33.59 ± 1.86	40.87 ± 1.29	59.24 ± 2.85	$\mathbf{44.16 \pm 2.09}$	41.72 ± 1.85
CODI	27.49 ± 2.28	32.17 ± 1.90	45.16 ± 2.79	36.87 ± 2.31	40.57 ± 1.51	63.38 ± 3.47	42.87 ± 2.79	40.28 ± 1.92
Qwen-3 CL	$\mathbf{31.37 \pm 2.19}$	$\mathbf{39.64 \pm 1.97}$	$\mathbf{54.72 \pm 2.86}$	$\mathbf{41.48 \pm 2.27}$	$\mathbf{42.63 \pm 1.77}$	$\mathbf{73.16 \pm 3.05}$	$\mathbf{44.93 \pm 2.20}$	$\mathbf{44.79 \pm 1.68}$

Notably, because Standard and Sol are fine-tuned to predict specifically the next step, they struggle to generalize to held-out sequences and can even degrade performance. Meanwhile, although Static incorporates contrastive learning, its formulation omits representation velocity—a key lever associated with reasoning—leading to sub-optimal results. These findings support the utility of representation velocity as an effective training signal for enhancing reasoning performance.

Table 3: Evaluation of methods over the benchmarks in the compositional setting. Results are reported as the mean and standard deviation of Pass@1 accuracy over 5 iterations. Qwen-3 CL outperforms in the majority of cases, improving from the base model by up to 19% points.

	Benchmark							
Method	**List Func.**	**FACT**	**NumSeq**	**OEIS**	**Crypto**	**SyGus**	**Induction**	**Instruct.**
Qwen-3 Base	22.15 ± 1.78	29.13 ± 1.85	39.87 ± 2.18	29.15 ± 1.97	33.92 ± 2.04	51.16 ± 3.48	31.54 ± 2.07	34.23 ± 1.89
Standard	22.82 ± 1.29	28.45 ± 1.72	39.41 ± 2.23	29.66 ± 1.89	34.78 ± 2.19	54.82 ± 3.43	33.19 ± 2.52	37.82 ± 2.35
Sol	23.49 ± 1.26	26.57 ± 1.88	40.55 ± 2.91	31.76 ± 2.14	38.13 ± 2.24	53.16 ± 3.27	31.18 ± 2.41	36.85 ± 1.97
Static	25.41 ± 1.88	27.61 ± 2.23	43.81 ± 2.17	35.38 ± 2.32	38.65 ± 1.82	53.56 ± 3.62	34.49 ± 1.95	37.22 ± 1.94
Coconut	27.18 ± 1.92	29.35 ± 2.26	42.47 ± 2.93	33.92 ± 1.95	39.24 ± 2.17	59.36 ± 4.38	40.27 ± 2.26	41.06 ± 2.18
CODI	29.65 ± 2.07	31.84 ± 2.72	43.59 ± 2.06	36.81 ± 2.53	$\mathbf{40.18 \pm 2.09}$	62.45 ± 4.28	40.27 ± 2.26	40.23 ± 2.17
Qwen-3 CL	$\mathbf{32.27 \pm 1.95}$	$\mathbf{37.72 \pm 2.91}$	$\mathbf{51.63 \pm 3.46}$	$\mathbf{41.22 \pm 2.53}$	$\mathbf{40.89 \pm 2.73}$	$\mathbf{70.28 \pm 3.65}$	$\mathbf{43.58 \pm 2.71}$	$\mathbf{44.51 \pm 2.23}$

5.4 Generalization into Out-of-Domain Datasets

The cross-domain and out-of-domain results for the math-field benchmarks are reported in Table 4. While performance gains over the base model diminish across all methods as task difficulty increases, our proposed contrastive fine-tuning approach enables Qwen-3 to achieve reasoning accuracy improvements of up to 9 percentage points over the base model and outpace competing base-

Table 4: Empirical results across close-domain and out-of-domain benchmarks. Reported values indicate Pass@1 accuracy (mean ± std) across 5 iterations. While generalization to other domains proves a challenging task QWEN-3 CL is superior in the majority of cases.

Method	In-Domain	Close-Domain			Out-Of-Domain	
	FACT	NumSeq	OEIS	Riddle	BBH	BBEH
QWEN-3 BASE	27.62 ± 1.37	41.67 ± 2.02	28.15 ± 1.98	18.47 ± 1.85	41.36 ± 1.62	19.36 ± 2.05
STANDARD	21.86 ± 1.72	39.55 ± 1.89	27.92 ± 1.76	19.45 ± 1.68	42.57 ± 1.88	18.65 ± 2.19
SOL	24.39 ± 1.64	41.26 ± 2.24	29.35 ± 2.14	16.93 ± 1.41	39.87 ± 1.97	19.43 ± 1.95
STATIC	28.57 ± 1.89	40.27 ± 2.35	28.17 ± 2.36	18.21 ± 1.57	41.06 ± 2.15	20.49 ± 1.86
COCONUT	29.48 ± 1.76	43.61 ± 2.15	31.58 ± 1.89	23.48 ± 1.35	44.16 ± 2.28	22.85 ± 2.37
CODI	32.17 ± 1.90	43.47 ± 2.06	30.28 ± 2.19	21.57 ± 1.96	45.82 ± 2.19	22.28 ± 2.15
QWEN-3 CL	$\mathbf{39.64 \pm 1.97}$	$\mathbf{50.29 \pm 2.05}$	$\mathbf{36.52 \pm 2.41}$	$\mathbf{25.49 \pm 1.87}$	$\mathbf{46.79 \pm 2.05}$	$\mathbf{25.87 \pm 1.58}$

lines by up to 7 percentage points. This performance margin narrows slightly on more challenging benchmarks such as BBH and BBEH

5.5 SEMANTIC MAPPING

Table 5: Assessment across the benchmarks in the semantic-mapping setting. Results are reported as the mean and standard deviation of Pass@1 accuracy over 5 iterations. QWEN-3 CL yields the most overall enhancement over the base model, up to 21% points.

Method	Benchmark			
	Crypto	SyGus	Induction	Instruct.
QWEN-3 BASE	33.78 ± 2.19	49.63 ± 3.28	30.35 ± 2.17	35.39 ± 1.83
STANDARD	34.85 ± 2.23	51.27 ± 3.84	32.41 ± 2.85	37.83 ± 2.16
SOL	37.61 ± 2.04	54.31 ± 3.29	31.54 ± 1.97	36.49 ± 1.82
STATIC	39.45 ± 1.97	53.96 ± 2.93	34.29 ± 1.82	39.26 ± 2.04
COCONUT	40.26 ± 2.14	60.88 ± 2.91	43.87 ± 2.11	40.47 ± 1.94
CODI	40.19 ± 2.17	62.57 ± 3.36	42.59 ± 2.26	40.63 ± 2.21
QWEN-3 CL	$\mathbf{42.18 \pm 2.45}$	$\mathbf{70.56 \pm 3.15}$	$\mathbf{44.93 \pm 2.47}$	$\mathbf{44.48 \pm 2.07}$

Examining the semantic mapping results in Table 5, we observe trends consistent with those reported in Section 5.3, as models must navigate the same underlying reasoning logic despite variations in surface semantics. By incorporating representation velocity—which serves as a key lever for reasoning—QWEN-3 CL yields reliable performance gains over QWEN-3 BASE and achieves competitive performance alongside strong baselines like COCONUT and CODI.

6 CONCLUSION

In this work, we investigate the inductive reasoning behavior of LLMs, from discovering patterns to designing mechanisms to boost the inductive reasoning capacity based on such observations. We discover that the velocity of hidden representations provides a meaningful signal that correlates with the underlying logical structure and offers an intervenable vector to steer the model's reasoning dynamics. Exploiting such signals, we devise a lightweight fine-tuning scheme based on contrastive learning for the language model. This training scheme guides the model to steer its reasoning trajectory in the proper velocity, which in turn enables the model to reach the right destination and arrive at the correct solution. Our experiments across several benchmarks demonstrate the effectiveness of our proposed method in boosting the inductive reasoning capacity and ability to generalize to unseen rules of LLMs. Our findings reveal a promising potential for optimizing representation trajectories, opening new avenues for developing more robust and interpretable inductive reasoners.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools to: generate synthetic datasets, implement methods, assist with translation, and clean and reformat the datasets. We have not used generative AI tools to: propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, or interpret results. The rest of the required disclosure tasks are not applicable to this work.

Additionally, we used generative AI tools to: edit the paper to enhance readability, suggest experimental hyperparameters, and summarize or analyse existing literature. We have reviewed all AI-assisted work. We checked LLM-generated relevant literature for authenticity and genuine relevance. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Peter Belcák, Ard Kastrati, Flavio Schenker, and Roger Wattenhofer. Fact: Learning governing abstractions behind integer sequences. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022) Track on Datasets and Benchmarks*, 2022.

Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*, 2025. URL https://arxiv.org/abs/2506.19143.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Rafael Cabral, Tuan Manh Do, Xuejun Yu, Wai Ming Tai, Zijin Feng, and Xin Shen. Proofflow: A dependency graph approach to faithful proof autoformalization, 2025. URL https://arxiv.org/abs/2510.15981.

Kedi Chen, Dezhao Ruan, Yuhao Dan, Yaoting Wang, Siyu Yan, Xuecheng Wu, Yinqi Zhang, Qin Chen, Jie Zhou, Liang He, et al. A survey of inductive reasoning for large language models. *arXiv preprint arXiv:2510.10182*, 2025.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 2020.

Yung-Sung Chuang, Benjamin Cohen-Wang, Shannon Zejiang Shen, Zhaofeng Wu, Hu Xu, Xi Victoria Lin, James Glass, Shang-Wen Li, and Wen-tau Yih. Selfcite: Self-supervised alignment for context attribution in large language models. *arXiv preprint arXiv:2502.09604*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Joy Crosbie and Ekaterina Shutova. Induction heads as an essential mechanism for pattern matching in in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5049–5111, 2025.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.716. URL https://aclanthology.org/2024.findings-acl.716/.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

Peter A. Flach and Antonis C. Kakas. Abductive and inductive reasoning: Background and issues. pp. 1–27, 2000. doi: 10.1007/978-94-017-0606-3_1. URL https://doi.org/10.1007/978-94-017-0606-3_1.

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, Yian Ma, Aurick Qiao, Tajana Rosing, Ion Stoica, and Hao Zhang. Efficiently scaling llm reasoning with certaindex. *arXiv preprint arXiv:2412.20993*, 2024. doi: 10.48550/arXiv.2412.20993.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Sonam Gupta, Yatin Nandwani, Asaf Yehudai, Dinesh Khandelwal, Dinesh Raghu, and Sachindra Joshi. Selective self-to-supervised fine-tuning for generalization in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6255–6264. Association for Computational Linguistics, 2025.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

Deepon Halder, Alan Saji, Thanmay Jayakumar, Anoop Kunchukuttan, Ratish Puduppully, and Raj Dabre. Riddlebench: A new generative reasoning benchmark for llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4363–4372, 2026.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling (COLM)*, 2025.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, 2022.

Kaiyu He and Zhiyu Chen. From reasoning to learning: A survey on hypothesis discovery and rule learning with large language models. *Transactions on Machine Learning Research*, 2025.

Or Honovich, Uri Shaham, Samuel Bowman, and Omer Levy. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1935–1952, 2023.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Wenyue Hua, Tyler Wong, Fei Sun, Liangming Pan, Adam Jardine, and William Yang Wang. Inductionbench: Llms fail in the simplest complexity class. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26526–26546, 2025.

Yao Huang, Huanran Chen, Shouwei Ruan, Yichi Zhang, Xingxing Wei, and Yinpeng Dong. Mitigating overthinking in large reasoning models via manifold steering. *arXiv preprint arXiv:2505.22411*, 2025. doi: 10.48550/arXiv.2505.22411.

Xue Jiang, Yihong Dong, Lecheng Wang, Zheng Fang, Qiwei Shang, Ge Li, Zhi Jin, and Wenpin Jiao. Self-planning code generation with large language models. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–30, 2024.

Haibo Jin, Peiyan Zhang, Man Luo, and Haohan Wang. Evaluating the inductive abilities of large language models: Why chain-of-thought reasoning sometimes hurts more than helps. *Advances in Neural Information Processing Systems*, 38:102916–102949, 2026.

Mehran Kazemi, Bahare Fatemi, Hritik Bansal, John Palowitch, Chrysovalantis Anastasiou, Sanket Vaibhav Mehta, Lalit K Jain, Virginia Aglietti, Disha Jindal, Yuanzhu Peter Chen, et al. Big-bench extra hard. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26473–26501, 2025.

Chunyang Li, Weiqi Wang, Tianshi Zheng, and Yangqiu Song. Patterns over principles: The fragility of inductive reasoning in llms under noisy observations. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19608–19626, 2025.

Peiji Li, Jiasheng Ye, Yongkang Chen, Yichuan Ma, Zijie Yu, Kedi Chen, Xiaozhe Li, Ganqu Cui, Haozhan Li, Jiacheng Chen, Chengqi Lyu, Wenwei Zhang, Linyang Li, Qipeng Guo, Dahua Lin, Bowen Zhou, and Kai Chen. Internbootcamp technical report: Boosting llm reasoning with verifiable task scaling, 2026. URL https://arxiv.org/abs/2508.08636.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=3EWTEy9MTM.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2023. Curran Associates Inc.

Gouki Minegishi, Hiroki Furuta, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. Topology of reasoning: Understanding large reasoning models through reasoning graph properties. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=o1g8NWkxqf.

Augustus Odena, Kensen Shi, David Bieber, Rishabh Singh, Charles Sutton, and Hanjun Dai. Bustle: Bottom-up program synthesis through learning-guided exploration. In *International Conference on Learning Representations*, 2021.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Daniel O'Malley, Manish Bhattarai, Nishath Ranasinghe, Erick Draayer, and Javier E Santos. Benchmarking large language models with integer sequence generation tasks. *Advances in Neural Information Processing Systems*, 38, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Raghav Ramji and Keshav Ramji. Inductive linguistic reasoning with large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22783–22810, 2025.

Joshua S. Rule. *The child as hacker: building more human-like models of learning*. PhD thesis, Massachusetts Institute of Technology, 2020. URL https://hdl.handle.net/1721.1/129232.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, 2025.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint arXiv:2305.03047*, 2023.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Ken Tsui. Numseqbench: Benchmarking inductive reasoning in language models via number sequences, 2024. URL https://huggingface.co/blog/kenhktsui/numseqbench.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. In *ICLR Workshop on Reasoning and Planning for Large Language Models*, 2025. URL https://openreview.net/forum?id=OwhVWNOBcz.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 24824–24837, 2022b.

Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3): 601–644, 1983. doi: 10.1103/RevModPhys.55.601.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in LLMs. *arXiv preprint arXiv:2502.07266*, 2025.

Junjie Yao, Zhongwang Zhang, and Zhi-Qin John Xu. An analysis for reasoning bias of language models with small initialization. In *International Conference on Machine Learning (ICML)*, 2025.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=Tn5B6Udq3E.

Yufa Zhou, Yixiao Wang, Xunjian Yin, Shuyan Zhou, and Anru R. Zhang. The geometry of reasoning: Flowing logics in representation space. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=ixr5Pcabq7.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A Appendix

A.1 Additional Experimental Details

We provide more technical details regarding our experiments presented in Sections 4 and 5.

Orthogonal Alignment: To orthogonally align two matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times d}$, we find the orthogonal rotation matrix $\mathbf{R} \in \mathbb{R}^{d \times d}$ that minimizes the squared Frobenius norm of their difference. This objective is formulated as:

$$\min_{\mathbf{R}} \|\mathbf{X}\mathbf{R} - \mathbf{Y}\|_F^2 \quad \text{subject to} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}$$

The optimal rotation matrix is analytically computed via Singular Value Decomposition (SVD) of the cross-covariance matrix $\mathbf{X}^T\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, yielding $\mathbf{R} = \mathbf{U}\mathbf{V}^T$, which aligns $\mathbf{X}$ into the coordinate space of $\mathbf{Y}$.

Apart from collecting the pairwise similarity distribution between $\mathbf{X}$ and $\mathbf{Y}$, the post-alignment similarity distribution is collected between $\mathbf{X}\mathbf{R}$ and $\mathbf{Y}$. As illustrated in Section 4.2, applying this orthogonal alignment helps rule out rotational effects and amplify the stark differences between the velocity similarity distributions when grouped by surface semantics vs. by the underlying inductive logic.

A.2 Benchmark Details

In this section, we describe the benchmarks utilized for our observations in Section 4 and experiments in Section 5.

Rule 30: Rule 30 is an elementary cellular automaton introduced by Wolfram (1983): a row of binary cells updates in discrete time steps, with each cell's next state determined by a fixed function of itself and its two immediate neighbors (specifically, left XOR (center OR right)). Despite this simple, deterministic local rule, the resulting pattern is highly irregular and difficult to predict, making it a useful test case for inductive reasoning over symbolic sequences.

For the observations presented in Section 4 and Appendix A.3, we generate sequences of binary strings whose transformation rule follows Rule 30,

Main Benchmarks: We describe the benchmarks utilized for our experiments in Section 5.

- List Function Rule (2020) Given several pairs of number lists (input– output) with a transformation rule mapping from the input to the output, the task is to induce such rule and predict new output from the new input list.

- FACT Belcák et al. (2022) Given a sequence of integers, the task is to infer its underlying governing rule and predict missing or subsequent terms.
- NumSeqBench Tsui (2024) Given a (incomplete) sequence of numbers, infer the generating rule and predict the next number or missing number.
- OEIS O'Malley et al. (2026) Given a sequence of numbers, infer the generating rule and predict the next number.
- CrypTo Li et al. (2025) Given several pairs of English words, where the output is obtained from the input word with a transformation rule, the task is to predict the output word from a new input word by applying the rule.
- SyGuS Odena et al. (2021) Given several pairs of strings, each contains an input and output where a transformation rule applies, the task is to generate a program to show the transformation process.
- Induction Bench Hua et al. (2025) Given a pair of strings, where the transformation from the input string to the output string follows a certain rule, the task is to predict the rule.
- Instruction Honovich et al. (2023) Given a pair of natural language statements input - output, where the output is obtained by applying an instruction to the input, the task is to output such an instruction.

Close-distribution and Out-of-distribution evaluation benchmarks: We provide some details for the additional benchmarks employed for the cross-domain and out-of-domain evaluation settings in Section 5.4.

- Riddle Bench Halder et al. (2026) The benchmark contains numerical/text sequence and encoding-decoding tasks.
- BBH Suzgun et al. (2023) The benchmark contains difficult tasks challenging the reasoning abilities of LLMs.
- BBEH Kazemi et al. (2025) The benchmark replaces BBH tasks with harder counterparts that test similar abilities using longer inputs, deeper reasoning, larger output spaces, or fewer shortcuts.

A.3 Additional Observations

In this section, we present the additional observations complementary to those highlighted in Section 4. In particular, we visualize the similarity distribution for Llama. In addition, we repeat the steering experiment in Section 4.2 to take a further step and illustrate the velocity as a causal lever with the underlying inductive logic of Llama.

Velocity as a correlational signal of logic:

We visualize the similarity distributions of velocities. The results are illustrated in Figure 3 and Figure 4.

Velocity as a causal lever of reasoning: The results for steering intervention in Llama are summarized in Table 6.

Table 6: Prediction accuracy when steering vs. ablating along the correct/incorrect/random directions. The numbers are mean and standard deviation of 5 trials, in Pass@10. Steering representations along the target velocity vector leads the model to more accurate predictions, whereas steering in opposing directions or ablating velocity significantly increases prediction errors. This demonstrates that representation velocity functions as a causal lever in model reasoning.

Benchmark	**Steering**			**Ablation**	
	Velocity	**Incorrect**	**Random**	**Velocity**	**Random**
Rule-30	45.26 ± 2.62	19.67 ± 3.40	25.81 ± 6.28	14.73 ± 2.59	22.45 ± 5.27
NumSeq	38.85 ± 2.97	15.41 ± 3.08	21.67 ± 4.49	16.28 ± 2.36	23.18 ± 5.41
Crypto	27.11 ± 2.58	12.27 ± 2.95	13.48 ± 2.53	10.82 ± 2.98	14.77 ± 3.85

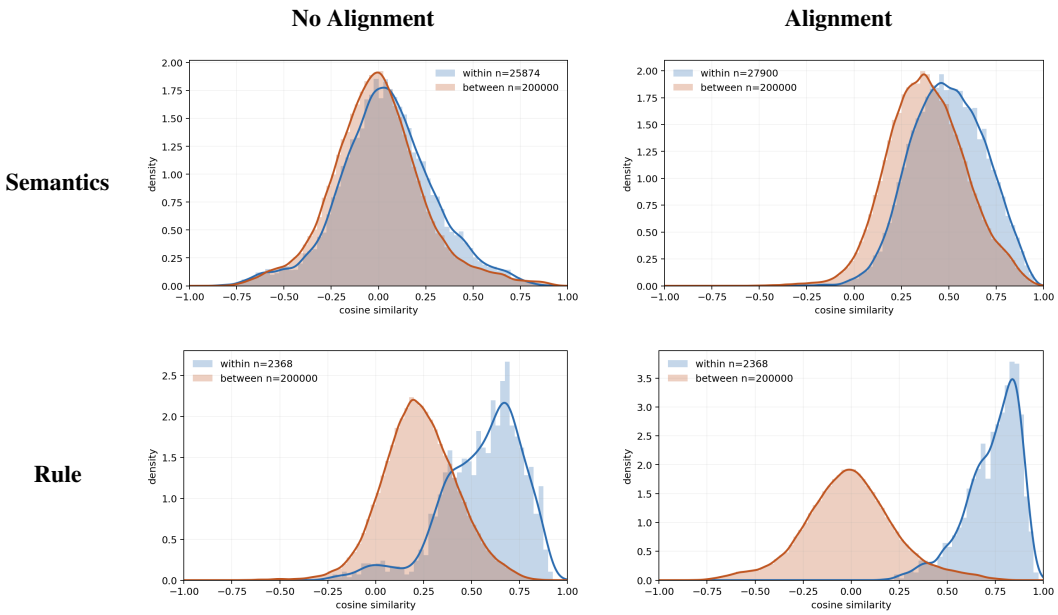

Figure 3: Early step: Similarity distribution of velocity vectors when grouped by alphabet (top row) vs. by underlying logic (bottom row) with(out) orthogonal alignment. While producing no clear velocity patterns by alphabet, the instances of the same underlying logic demonstrate highly similar velocity vectors. This similarity is more apparent when rotation effects are ruled out.

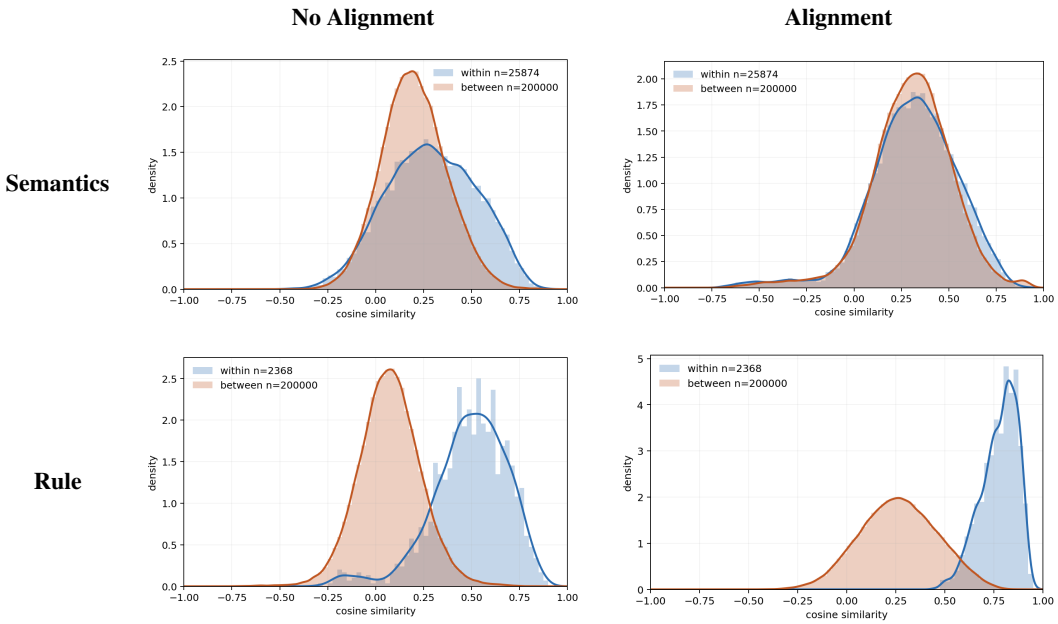

Figure 4: Latter step: Similarity distribution of velocity vectors when grouped by alphabet (top row) vs. by underlying logic (bottom) without and with orthogonal alignment. While producing no clear velocity patterns by alphabet, the instances of the same underlying logic demonstrate highly similar velocity vectors. This similarity is more apparent when rotation effects are ruled out by alignment.

Directing internal representations along the correct velocity vector maximizes task performance, whereas counter-steering or stripping out velocity components significantly degrades reasoning accuracy. This bidirectional behavior validates that velocity is a causal lever capable of steering model inference, rather than a mere correlational artifact.

A.4 Experimental Results for Llama

In this section, we report the experimental results for Llama, complementary to Section 5. Overall, Llama CL provides superior performance over the base model Llama Base and generally outperforms the competitors in the majority of cases, ranging from inducing held-out rules in-domain to generalizing to out-of-domain benchmarks or semantically translated settings.

In-domain and Compositional Settings:

The comparison of methods is reported in Table 7 for the in-domain held-out setting and in Table 8 for the compositional setting. In summary, the contrastive learning fine-tuning in Section 5.1 affords Llama CL an enhancement of up to 19% points from Llama Base and 8% points over the competitors. The baselines are trained to optimize with respect to predicting the next step or aligning the hidden-layer representations, and thus struggle to reason in held-out rules. This indicates that the trajectory velocity acts as a useful signal to improve the reasoning capacity.

Table 7: Comparison of methods over the benchmarks in the in-distribution held-out setting. We report the mean and standard deviation of Pass@1 accuracy across 5 independent runs. It is clear that Llama CL is overall the most superior method, offering an improvement of up to 19% points over the base model and up to 8% points over the strongest competitor.

	Benchmark							
Method	**List Func.**	**FACT**	**NumSeq**	**OEIS**	**Crypto**	**SyGus**	**Induction**	**Instruct.**
Llama Base	19.18 ± 1.47	26.14 ± 1.95	37.72 ± 2.13	28.08 ± 2.69	25.68 ± 2.29	45.25 ± 3.63	29.38 ± 2.13	35.76 ± 1.67
Standard	21.33 ± 1.56	23.57 ± 2.09	38.06 ± 1.95	27.62 ± 2.41	27.45 ± 2.19	48.25 ± 3.57	31.44 ± 2.07	36.19 ± 2.28
Sol	20.35 ± 1.28	26.17 ± 2.45	37.94 ± 2.07	26.59 ± 2.98	27.43 ± 1.74	46.37 ± 3.29	30.42 ± 1.83	33.27 ± 2.08
Static	20.86 ± 1.74	27.36 ± 2.21	40.34 ± 2.72	28.37 ± 2.35	26.94 ± 1.83	45.28 ± 3.76	31.63 ± 2.16	36.38 ± 1.63
Coconut	23.17 ± 1.54	31.36 ± 2.07	44.27 ± 1.89	32.27 ± 2.53	29.48 ± 2.15	55.27 ± 3.42	34.18 ± 2.59	41.18 ± 2.09
CODI	21.98 ± 1.83	30.57 ± 2.36	45.18 ± 2.26	33.65 ± 3.19	29.42 ± 2.15	58.22 ± 2.95	33.27 ± 2.18	40.72 ± 2.25
Llama CL	$\mathbf{26.22 \pm 2.15}$	$\mathbf{38.13 \pm 2.65}$	$\mathbf{50.79 \pm 2.03}$	$\mathbf{40.27 \pm 3.21}$	$\mathbf{34.29 \pm 2.87}$	$\mathbf{64.21 \pm 3.20}$	$\mathbf{38.25 \pm 3.28}$	$\mathbf{43.39 \pm 2.04}$

Table 8: Evaluation of methods over the benchmarks in the compositional setting. Results are reported as the mean and standard deviation of Pass@1 accuracy over 5 iterations. Llama CL outperforms in the majority of cases, improving from the base model by up to 17% points.

	Benchmark							
Method	**List Func.**	**FACT**	**NumSeq**	**OEIS**	**Crypto**	**SyGus**	**Induction**	**Instruct.**
Llama Base	17.43 ± 1.51	22.29 ± 1.93	38.27 ± 1.87	26.41 ± 2.72	22.37 ± 2.15	46.39 ± 3.54	30.27 ± 2.26	36.18 ± 1.72
Standard	20.39 ± 1.46	22.76 ± 1.58	38.46 ± 1.32	26.37 ± 2.73	23.19 ± 2.06	47.35 ± 3.68	30.51 ± 2.19	37.73 ± 2.06
Sol	18.79 ± 1.30	24.47 ± 1.82	37.52 ± 1.29	28.87 ± 2.84	23.53 ± 1.86	46.18 ± 2.73	29.83 ± 1.98	36.41 ± 2.05
Static	18.57 ± 1.62	25.41 ± 2.20	39.41 ± 1.53	30.82 ± 2.98	22.35 ± 1.91	50.76 ± 3.13	31.47 ± 2.05	38.51 ± 1.84
Coconut	21.83 ± 2.04	27.85 ± 1.87	43.15 ± 1.49	33.65 ± 2.73	27.71 ± 1.73	59.52 ± 2.81	33.09 ± 1.97	$\mathbf{43.24 \pm 2.35}$
CODI	22.57 ± 1.91	27.14 ± 2.03	44.61 ± 1.63	33.41 ± 3.06	30.65 ± 1.82	60.38 ± 2.91	35.39 ± 1.85	42.58 ± 2.27
Llama CL	$\mathbf{26.89 \pm 1.65}$	$\mathbf{35.38 \pm 2.14}$	$\mathbf{49.35 \pm 1.51}$	$\mathbf{41.38 \pm 3.16}$	$\mathbf{36.22 \pm 1.91}$	$\mathbf{65.21 \pm 3.07}$	$\mathbf{37.45 \pm 2.94}$	42.29 ± 2.16

Close-domain and Out-of-domain results: We summarize the close-domain and out-of-domain comparison in Table 9. Generalization to other benchmarks proves a challenging task, and Llama CL yields a considerable boost over Llama Base.

Semantic mapping results:

Table 10 illustrates the results for the semantic mapping setting. The comparison is similar to that reported in Table 7, as the underlying logic is preserved and only the surface semantics are translated.

Table 9: Empirical results across close-domain and out-of-domain benchmarks. Reported values indicate Pass@1 accuracy (mean ± std) across 5 iterations. While generalization to other domains proves a challenging task LLAMA CL is superior in the majority of cases

Method	In-Domain	Close-Domain			Out-Of-Domain	
	FACT	NumSeq	OEIS	Riddle	BBH	BBEH
LLAMA BASE	26.14 ± 1.95	36.51 ± 2.35	27.15 ± 1.89	20.84 ± 1.76	40.28 ± 1.43	14.57 ± 2.23
STANDARD	23.57 ± 2.09	35.62 ± 2.08	26.67 ± 2.14	22.38 ± 1.65	42.73 ± 2.41	17.63 ± 2.41
SOL	26.17 ± 2.45	35.88 ± 2.09	26.89 ± 1.94	22.58 ± 1.46	40.05 ± 2.27	18.46 ± 1.95
STATIC	27.36 ± 2.21	36.57 ± 1.93	27.51 ± 2.19	22.72 ± 1.69	43.66 ± 2.15	17.72 ± 2.11
COCONUT	31.36 ± 2.07	40.18 ± 2.01	30.05 ± 1.86	$\mathbf{25.18 \pm 1.23}$	45.72 ± 2.44	17.38 ± 2.18
CODI	30.57 ± 2.36	41.72 ± 1.75	$\mathbf{32.67 \pm 2.05}$	23.26 ± 1.74	44.91 ± 2.29	18.85 ± 2.06
LLAMA CL	$\mathbf{38.13 \pm 2.65}$	$\mathbf{43.98 \pm 1.81}$	$\mathbf{32.15 \pm 2.27}$	$\mathbf{25.86 \pm 1.98}$	$\mathbf{48.73 \pm 2.31}$	$\mathbf{22.37 \pm 2.28}$

Table 10: Assessment across the benchmarks in the semantic-mapping setting. Results are reported as the mean and standard deviation of Pass@1 accuracy over 5 iterations. LLAMA CL yields the most overall enhancement over the base model, up to 19.9% points.

Method	Benchmark			
	Crypto	SyGus	Induction	Instruct.
LLAMA BASE	26.35 ± 2.14	43.39 ± 2.68	27.92 ± 2.04	36.43 ± 1.74
STANDARD	27.48 ± 2.06	47.42 ± 2.95	30.61 ± 2.13	36.58 ± 1.99
SOL	27.13 ± 1.81	47.24 ± 3.05	30.29 ± 1.92	35.71 ± 1.96
STATIC	26.95 ± 1.87	46.51 ± 2.82	30.93 ± 1.95	36.85 ± 1.83
COCONUT	30.38 ± 2.16	55.37 ± 3.11	33.58 ± 2.27	41.57 ± 2.06
CODI	29.67 ± 1.99	57.14 ± 2.81	33.86 ± 2.15	41.82 ± 2.15
LLAMA CL	$\mathbf{34.27 \pm 1.85}$	$\mathbf{62.45 \pm 2.77}$	$\mathbf{37.81 \pm 2.48}$	$\mathbf{43.36 \pm 2.27}$