Diverse Reasoning Paths Matter: Symbolic Equivalence Augmentation for Enhancing RL Exploration

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) has emerged as a powerful paradigm for improving the reasoning capabilities of LLMs. However, the role of supervised fine-tuning (SFT) data in enabling effective downstream RL remains underexplored. In this paper, we investigate a key hypothesis: *the diversity of reasoning steps in chain-of-thought (CoT) during SFT is a critical enabler of RLVR performance*. To test this hypothesis, we introduce EquivAug, a data augmentation framework that uses symbolic equivalence transformations to generate diverse yet provably correct reasoning paths. By converting CoT solutions into formal symbolic representations, applying equivalence transformations, and reconstructing natural-language explanations, EquivAug produces training data that differs in reasoning steps rather than surface-level lexical variation. Experiments on GSM8K and FOLIO across model scales demonstrate that EquivAug consistently achieves the highest reasoning path diversity after SFT and the strongest accuracy after GRPO[1]. Our results establish that CoT diversity during SFT is positively correlated with downstream RLVR performance.

1 Introduction

Most prior work on supervised fine-tuning (SFT) for reasoning focuses primarily on improving accuracy on the SFT task itself (Yuan et al., 2023). However, achieving high SFT accuracy does not necessarily lead to a better model, especially after the success of reinforcement learning with verifiable rewards (RLVR) (Guo et al., 2025; Singh et al., 2025). Recent studies show that high SFT accuracy does not reliably lead to high RL performance, but enabling exploration is more important (Kang et al., 2025). However, existing work has controlled for variables at the problem-level factors, such as task diversity (Kang et al., 2025), which affect RL performance, a fundamentally different dimension remains underexplored: the *reasoning path level*. This raises a natural question: *Does exposure to diverse chain-of-thought during SFT enable more effective exploration and stronger performance in downstream RLVR?*

In this paper, we propose a novel perspective: **the diversity of reasoning steps in chain-of-thought (CoT) at the SFT stage is a critical enabler of RLVR**. A model fine-tuned on a single solution per problem may achieve strong supervised performance, yet it learns a rigid policy that RL struggles to effectively diversify. We hypothesize that exposing the model to *multiple structurally distinct yet correct* reasoning paths during SFT broadens the initial policy's support, thereby enabling more effective exploration during the subsequent RL phase. Despite its intuitive appeal, to the best of our knowledge, no prior work has systematically investigated how the diversity of SFT reasoning data influences downstream RLVR performance.

To systematically analyze the relationship between reasoning diversity during the SFT stage and performance after RLVR, we must first address a closely related challenge: *how to obtain diverse reasoning paths*. Existing data augmentation methods, such as paraphrasing-based augmentation (Yu et al., 2024; Li et al., 2024a,c) and Rejection Sampling Fine-Tuning (RFT) (Yuan et al., 2023), typically rely on LLMs to generate alternative phrasing for a single solution. However, these approaches primarily introduce surface-level lexical variation rather than new reasoning strategies. Therefore, current data augmentation methods cannot provide the diversity of reasoning steps in CoT and thus are not suitable to verify our hypothesis.

To address the limitation of current data augmentation methods and study the relationship between diversity and RL, we introduce **EquivAug**, a novel data augmentation framework that generates di-

[1]Our code can be found at https://anonymous.4open.science/r/March_Arr-0E9D.

verse and verified reasoning paths. Our approach is grounded in a key observation: mathematical and logical reasoning inherently admits rich equivalence structures, in which the same conclusion can be reached through multiple formally equivalent derivations. For example, in mathematics, one can apply the distributive property to alter the computational graph (e.g., transforming $a(b+c)$ to $ab+ac$). By employing symbolic equivalent transformations, EquivAug ensures that the augmented chain-of-thought (CoT) trajectories are structurally distinct at intermediate steps.

We conduct experiments on the GSM8K (Cobbe et al., 2021) and FOLIO (Han et al., 2024) datasets, evaluating our method across model scales ranging from 1.5B to 8B parameters. Using Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as our RL algorithm, we compare different data augmentation strategies followed by an identical post-training process. Our results validate our core hypothesis: introducing diverse CoT data during the SFT stage significantly enhances post-training RLVR performance, while surface-level diversity cannot. This highlights the vital importance of ensuring the *reasoning step diversity* during the SFT stage. Overall, our main contributions can be summarized as follows:

- We systematically investigate the relationship between CoT diversity at the SFT stage and downstream RLVR performance. We demonstrate that diverse and correct SFT data significantly improve RL outcomes. To the best of our knowledge, this is the first work to empirically establish this connection.
- We propose EquivAug, a novel framework that generates diverse and verifiable reasoning paths via equivalent transformations.
- We empirically show that EquivAug, combined with GRPO, consistently outperforms all baseline methods across multiple model scales (1.5B–8B) and benchmark. Furthermore, we provide detailed ablation studies on transformation complexity and data scaling.

2 Related Work

2.1 The Interplay between SFT and Reinforcement Learning

Enhancing exploration has been considered the key to RL training (Song et al., 2025). Considering the SFT-then-RL pipeline (Guo et al., 2025; Shao et al., 2024), SFT provides a cold-start reasoning policy, which decides the quality of exploration (Chu et al., 2025; Chen et al., 2025). Consequently, a growing body of work has analyzed how SFT decides the exploration and the following RL performance. For instance, Yue et al. demonstrates that SFT on data from stronger LLMs can broaden the model's exploration ability. Matsutani et al. (2026) shows that SFT can expand the number of correct trajectories and therefore increase the exploration ability. Kang et al. (2025) demonstrates that high accuracy after SFT does not lead to high accuracy after RL, showing that task diversity or difficulty are more important in deciding performance after RLVR. However, while these studies highlight the importance of SFT data, they primarily analyze factors at the *problem level* such as diversity of questions, ignoring the diversity of reasoning paths. In contrast, we investigate the *reasoning path level*: for a fixed set of problems, how the diversity of chain-of-thought solutions shapes the model's initial exploration space and thereby influences downstream RL optimization to fill the gap.

2.2 Data Augmentation for Reasoning

Data augmentation has become a central strategy for improving the reasoning capabilities of LLMs during SFT. A lot of work focuses on increasing the *complexity and diversity of questions*: WizardMath (Luo et al., 2023) applies evolutionary instructions to increase problem difficulty, MetaMath and MuggleMath (Yu et al., 2024; Li et al., 2024a) bootstraps mathematical questions by rewriting them from multiple perspectives. For augmenting the *solution side*, existing approaches fall into two broad categories. *Paraphrase-based* methods employ a stronger model to rewrite ground-truth solutions (Yu et al., 2024; Li et al., 2024c). These methods increase surface-level diversity but preserve the underlying solution strategy. *Sampling-based* methods, exemplified by Rejection Sampling Fine-Tuning (RFT) (Yuan et al., 2023), use the model itself to generate multiple candidate solutions and retain those whose final answer is correct. However, final-answer correctness does not guarantee the validity of intermediate reasoning steps, because LLMs can produce plausible but logically flawed chains of thought. A promising direction to address these limitations is *neuro-symbolic* data augmentation, which leverages symbolic representations to ensure the correctness of generated

data. Recent work has explored symbolic mutation for generating verifiable math problems (Li et al., 2024c) and symbolic verification for logical reasoning data (Li et al., 2024b). Our work builds on this line of research: EquivAug combines equivalence-preserving symbolic transformations with solver-based verification to produce reasoning paths that are both structurally diverse and provably correct at every intermediate step.

3 Motivation and Problem Definition

In GRPO (Shao et al., 2024), the policy is updated based on non-zero advantages, which are computed by normalizing rewards within a group of G sampled responses to the same prompt:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G) + \epsilon}, \tag{1}$$

where $r_i = R(x, y_i)$ is the binary reward for response y_i and ϵ is a small constant for numerical stability. These advantages provide meaningful learning signals only when the group contains responses with different rewards. Since the rewards in RLVR are mostly rule-based accuracy rewards (Xie et al., 2025), if all G responses are uniformly correct or incorrect, the advantages will vanish, leaving the policy with no gradient for that prompt.

Therefore, for an effective RL process, the policy $\pi_{\theta_{\text{SFT}}}$ must be capable of producing sufficiently diverse responses so that sampled groups contain a mixture of correct and incorrect solutions (Zhang et al., 2025). This requires exploration ability for $\pi_{\theta_{\text{SFT}}}$. A policy trained on narrow SFT data tends to generate homogeneous responses, increasing the risk of uninformative reward groups. Moreover, on challenging problems, a narrow policy may fail to produce any correct response (Wu et al., 2026). A broader initial policy, equipped with multiple reasoning strategies, is more likely to succeed on such problems through at least one strategy, reducing the occurrence of all-incorrect groups and ensuring that informative advantage signals persist throughout training. The initial policy could learn from SFT data to obtain multiple reasoning strategies, which motivates our following hypothesis:

Hypothesis 1 *Training the model on diverse reasoning paths during SFT broadens the initial policy's coverage of the solution space, leading to more effective exploration and stronger downstream RLVR performance.*

To test Hypothesis 1, we need SFT datasets with diversity on CoT. However, existing augmentation methods such as paraphrase-based rewriting (Yu et al., 2024; You et al., 2024) primarily introduce surface-level lexical variation that changes the phrases or sentences without altering the underlying reasoning structure. Training on such data does not broaden the reasoning strategies available to the policy, making it unsuitable for testing our Hypothesis 1. This leads to the central technical problem of our work: how can we generate multiple reasoning paths that are step-level diverse while preserving correctness? Formally, we define this as the following augmentation problem:

Problem 1 (Equivalent Solution Augmentation) *Given a dataset $\mathcal{D} = \{(q_i, s_i, a_i)\}_{i=1}^N$, where q_i is a question, s_i is a natural language reasoning path, and a_i is the final answer, the goal is to produce an augmented dataset that i-th sample has K_i reasoning paths:*

$$\mathcal{D}' = \left\{(q_i,\ s'_{i,k},\ a_i) \mid k = 1, \ldots, K_i\right\},$$

where each generated reasoning path $s'_{i,k}$ satisfy two key constraints: (1) correctness*—$s'_{i,k}$ stick to the same answer a_i and (2)* diversity*—two variants $\sigma'_{i,k}$ and $\sigma'_{i,k'}$ have different reasoning paths.*

4 Methodology

In this section, we introduce EquivAug, a framework for symbolic data augmentation that uses equivalence transformations to generate diverse yet provably correct reasoning paths. We first describe the overview and then detail each component.

4.1 Overview

As illustrated in the left portion of Figure 1, EquivAug is a domain-agnostic augmentation framework comprising three modules: (1) **Symbolic Representation**, which converts natural language CoT into formal symbolic forms — specifically, math word problems are translated into algebraic expressions, while logical reasoning problems are encoded as solver-compatible first-order logic formulas; (2) **Equivalence Transformation**, which applies domain-specific equivalence rule libraries to systematically derive multiple alternative yet semantically equivalent symbolic solutions from each original representation; and (3) **Natural Language Reconstruction**, which converts each derived symbolic solution back into natural language, producing the final augmented training dataset.

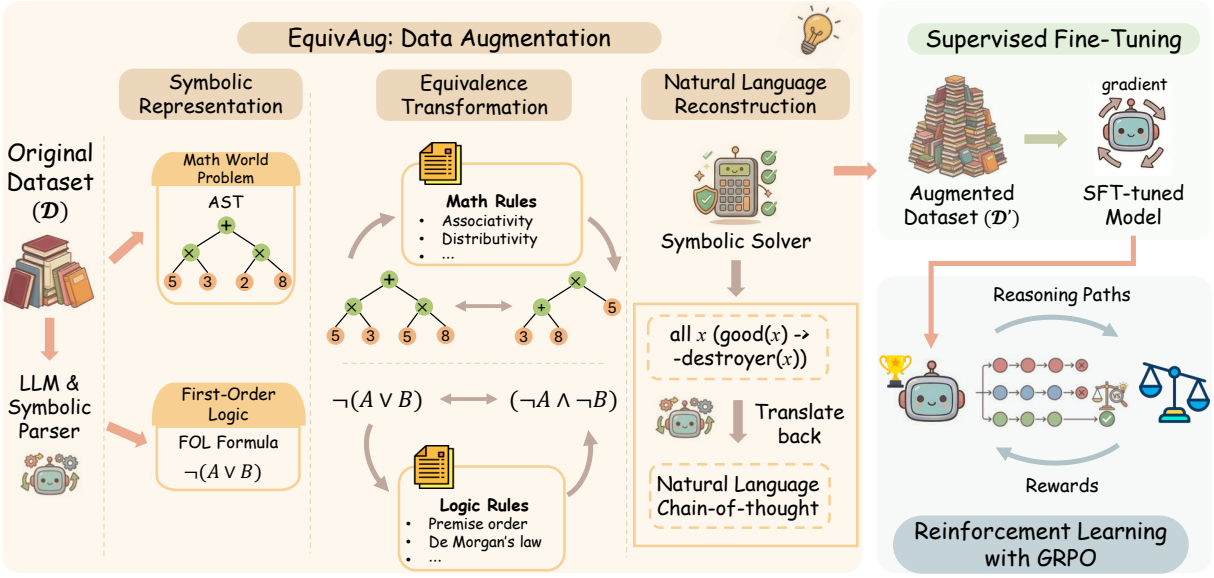

Figure 1: Overview of our pipeline. **Left:** The EquivAug data augmentation framework converts chain-of-thought solutions into symbolic representation, applies equivalence transformation rules to generate structurally diverse yet semantically equivalent variants verified by a symbolic solver, and reconstructs them back into natural language. **Right:** The augmented data is used for supervised fine-tuning, followed by reinforcement learning with GRPO.

Crucially, every augmented CoT solution is verified by a symbolic solver prior to back-translation, ensuring the correctness of not only the final answer but also all intermediate reasoning steps. This symbolic design distinguishes EquivAug from heuristic augmentation methods that risk introducing subtle errors. The entire pipeline is modular: the same three-stage workflow applies uniformly across different domains such as mathematics and logic, differing only in the domain-specific equivalence rule sets plugged into Stage (2). Appendix B summarizes the complete procedure.

4.2 Symbolic Representation

The first module maps each chain-of-thought solution into a formal symbolic representation that supports both symbolic verification and equivalence-preserving transformation. While the specific representation differs across domains, the underlying principle remains the same: the original reasoning process is converted into a structured symbolic object that can be parsed, manipulated, and verified by external solvers. We detail the domain-specific instantiations below.

Math Word Problems. We decompose the solution process into a sequence of step-level algebraic expressions and organize them into a unified abstract syntax tree (AST) (Alfred et al., 2007). In this tree, leaf nodes correspond to numbers and variables, while internal nodes denote arithmetic or algebraic operators such as $+$, $-$, $\times$, and $\div$. To ensure correctness, we execute the complete symbolic expression using a symbolic solver and compare the resulting value against the gold answer; only solutions that pass this verification are retained. This tree-structured formulation enables the equivalence-preserving transformations listed in Table 1 to be applied to arbitrary subtrees.

First-Order Logic. For first-order logic (FOL) reasoning problems, the premises and conclusions in the FOLIO dataset (Han et al., 2024) are already annotated as FOL formulas. To obtain a solver-executable symbolic representation, we convert these formulas into the input format required by automated theorem provers via lightweight rule-based parsing.

We employ the automated theorem prover Prover9 (McCune, 2005) to verify the entailment labels and extract structured proof traces. Prover9 adopts a refutation-based proving strategy: it negates the target conjecture, adds it to the premise set, and attempts to derive a contradiction. Depending on the gold label (TRUE, FALSE, or UNCERTAIN), we configure the conjecture accordingly and invoke either Prover9 or the finite model builder Mace4 to verify the label. Details about Prover9 can be found in Appendix A.

A key property of Prover9 that our framework exploits is that its proof search is highly sensitive to both the syntactic form of the input formulas and the search strategy configuration. Even semantically equivalent reformulations of the premises,

such as reordering clauses, applying De Morgan's laws, or distributing quantifiers, can lead the prover down substantially different search paths, yielding structurally distinct yet equally valid proof traces. This sensitivity provides a natural mechanism for generating diverse reasoning paths: by systematically rewriting the input formulas using semantics-preserving logical equivalences, we can obtain multiple alternative derivations from a single problem instance without compromising logical validity.

4.3 Equivalence Transformation

The second module generates diverse symbolic variants by systematically applying equivalence-preserving transformations to each formal representation. These rewrites alter the structure of the symbolic reasoning process while ensuring that every transformed variant remains consistent with the original answer or logical label. We maintain domain-specific transformation libraries, summarized for mathematics and first-order logic in Table 1 and Table 2, respectively.

Math Word Problems. We apply local rewrites to the symbolic expression tree using algebraic equivalence rules. These rewrites operate on subtrees of the expression AST and include standard properties such as commutativity, associativity, distributivity, and fraction decomposition, listed in Table 1. Because each rewrite is applied to a well-formed subexpression, the transformed expression remains mathematically equivalent to the original one. These transformations can change the decomposition order of arithmetic operations, the grouping structure of intermediate expressions, and the representation of fractions or chained operations, while leaving the final numerical value unchanged. By sampling different valid rewrites over the symbolic tree, we obtain multiple correct symbolic solution variants from a single original CoT.

First-Order Logic. For FOL reasoning, we transform the symbolic premises using logical equivalences, including double negation, contraposition, De Morgan's laws, implication expansion, and related rewrites (Enderton, 2001). Since premise order does not affect logical entailment, we also apply premise-level permutations such as shuffling and reordering. The full set of logical transformations is listed in Table 2. Although these rewrites preserve the logical content of the original premises, they often cause the theorem prover to explore different inference paths and produce distinct proof traces. As a result, multiple transformed premise sets may correspond to the same gold label while exhibiting substantially different derivational structures.

Category	Rule	Equivalence
Arithmetic	Commutativity	$a + b \leftrightarrow b + a$
		$a \times b \leftrightarrow b \times a$
	Associativity	$(a + b) + c \leftrightarrow a + (b + c)$
		$(a \times b) \times c \leftrightarrow a \times (b \times c)$
		$a \times b \div c \leftrightarrow a \div c \times b$
	Distributivity	$a \times (b + c) \leftrightarrow a \times b + a \times c$
		$(a + b) \times c \leftrightarrow a \times c + b \times c$
		$a \times (b - c) \leftrightarrow a \times b - a \times c$

Table 1: Partial equivalence rules for **math world problems**.

Category	Rule	Equivalence / Transformation
Premise order	Original	Keep the original premise order
	Shuffle	Permute premises arbitrarily
	Reverse	Reverse the premise order
Logical equivalence	Double negation	$A \leftrightarrow \neg\neg A$
	Contraposition	$A \rightarrow B \leftrightarrow \neg B \rightarrow \neg A$
	De Morgan's law	$\neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$
		$\neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$
	Implication expansion	$A \rightarrow B \leftrightarrow \neg A \vee B$
	Biconditional expansion	$A \leftrightarrow B \leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$
Connective restructuring	Commutativity of $\wedge, \vee$	$A \wedge B \leftrightarrow B \wedge A$
		$A \vee B \leftrightarrow B \vee A$
	Distributivity	$A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$
		$A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$
	Antecedent exportation	$(A \wedge B) \rightarrow C \leftrightarrow A \rightarrow (B \rightarrow C)$
Simplification	Idempotent law	$A \wedge A \leftrightarrow A$
		$A \vee A \leftrightarrow A$
	Absorption	$A \vee (A \wedge B) \leftrightarrow A$
		$A \wedge (A \vee B) \leftrightarrow A$

Table 2: Partial equivalence transformations for **FOL**.

Beyond formula-level transformations, we also increase reasoning diversity by varying the search configuration of the theorem prover. Specifically, we use multiple Prover9 strategies that differ in resolution settings and clause-selection heuristics, including hyperresolution, unit-resulting resolution, and different values of `pick_given_ratio`. These strategy variations do not alter the semantics of the input formulas; instead, they change how the prover searches for a refutation.

4.4 Natural Language Reconstruction

The final module converts each transformed symbolic reasoning path back into natural language CoT solutions. We prompt a large language model to translate the symbolic derivation into fluent natural language explanations. The prompt is carefully designed to enforce strict alignment between the symbolic steps and the generated text. The full prompt template is provided in Appendix C.3.

5 Experiments

We conduct comprehensive experiments to validate the effectiveness of EquivAug and to investigate how diverse yet provably correct reasoning paths

during SFT influence both the fine-tuned model itself and its subsequent RL training. Specifically, our experiments are organized around the following research questions:

RQ1: Can EquivAug generate diverse CoT paths, and how do other reasoning augmentation methods perform on generating diverse CoT?

RQ2: Does exposure to diverse chain-of-thought paths during SFT facilitate more effective exploration in post-training reinforcement learning?

RQ3: How do the complexity and quantity of equivalence transformations affect downstream performance? Can EquivAug generalize to out-of-distribution tasks?

5.1 Experiment Settings

Datasets. We evaluate on two benchmarks: (1) GSM8K (Cobbe et al., 2021) is a grade-school math word problems, consisting of 1,319 test examples. (2) FOLIO (Han et al., 2024) is a first-order logic reasoning benchmark where each example comprises a set of natural language premises paired with FOL annotations and a conclusion to be classified as TRUE, FALSE, or UNCERTAIN.

Configurations. We experiment with three backbone large language models of varying scale: Qwen2.5-1.5B, Qwen2.5-3B (Yang et al., 2025), and Llama-3.1-8B (Grattafiori et al., 2024). For supervised fine-tuning, we apply full-parameter training for Qwen2.5-1.5B and Qwen2.5-3B and use LoRA (Hu et al., 2022) for Llama-3.1-8B. Full SFT runs use a cosine learning rate schedule with a peak learning rate of 2e-5, while LoRA uses a learning rate of 2e-4. For reinforcement learning, we adopt GRPO (Shao et al., 2024) as the post-training algorithm. The reward signal is based on exact-match correctness of the final answer. We use a sampling temperature of 0.7, generate 8 rollouts per prompt with a learning rate of 1e-6. All experiments are conducted on 2 Nvidia A100 GPUs. To have a fair comparison, we use the same training samples for all methods. Further implementation details are provided in Appendix C.1.

Compared Methods. We compare several reasoning augmented strategies while keeping all other post-training configurations fixed:

(1) **Ground Truth**: SFT on ground-truth solutions. We directly use the annotated solutions provided by the benchmark. For FOLIO, since the benchmark didn't provide the reasoning traces, we directly use the gold label as the ground-truth reasoning.

(2) **RFT** (Yuan et al., 2023): In the Rejection Sampling Fine-tuning method, we use a model that has been fine-tuned on the ground-truth data to sample multiple candidate solutions, retaining only those whose final answers are correct as the augmented training dataset.

(3) **Paraphrasing**: We use a stronger language model (e.g., GPT-4o-mini) to paraphrase the original annotated solutions in the benchmark, to make sure the augmented solutions that remain correct while increasing diversity in expression. In the case of FOLIO, the benchmark only provides final labels rather than full reasoning paths. We therefore first use the model itself to generate a chain-of-thought that arrives at the correct answer, and then paraphrase this generated solution with the stronger model. Prompt templates are deferred to Appendix C.3.

(4) **EquivAug**: Our method employing the equivalence transformation libraries and symbolic verification pipeline described in Section 4.

Metrics. To evaluate the effectiveness of EquivAug and validate our hypothesis, we report accuracy at two stages: ACC_{SFT}, the final-answer accuracy of the model after SFT, and ACC_{GRPO}, the accuracy after subsequent reinforcement learning with GRPO. We use mean accuracy for 5 runs. To quantify the diversity of reasoning paths generated by the fine-tuned model, we adopt **Reasoning Path Diversity** (RPD) (Ju et al., 2025), a step-level metric that captures semantic variation across intermediate reasoning steps rather than surface-level lexical differences. For each test instance, we sample 5 solutions from the SFT model and compute RPD over the set of generated reasoning paths. Further details on the RPD computation are provided in Appendix C.2.

5.2 Diversity of Generated Reasoning Paths (RQ1)

We first evaluate whether supervised fine-tuning on EquivAug data elicits greater diversity in the model's generated chain-of-thought reasoning compared to baseline methods. This question is central to our framework, since the purpose of symbolic transformation is not only to preserve correctness, but also to expose the model to a broader set of valid reasoning trajectories during training, thereby unlocking greater exploration potential for subsequent reinforcement learning. As shown in Table 3, across all evaluation settings, models trained with EquivAug consistently achieve higher RPD than

the corresponding baselines. It can show that symbolic equivalence-based augmentation effectively broadens the space of reasoning patterns learned during supervised fine-tuning.

Notably, the comparison with other baselines is particularly informative: while paraphrasing by GPT-4o introduces lexical variation, it largely preserves the original reasoning structure, resulting in lower RPD. Similarly, RFT improves diversity to some extent by sampling multiple solutions from the model itself, but the generated paths tend to cluster around the patterns already learned during initial SFT, limiting the effective diversity gain. In contrast, EquivAug produces paths that differ at the level of reasoning steps, such as restructuring intermediate computations or deriving conclusions through alternative inference chains, yielding substantially higher diversity. This suggests that systematic symbolic transformations, rather than mere surface-level rephrasing, are the key driver of reasoning diversity.

5.3 Post-training RL with Diverse SFT (RQ2)

We then analyze how different SFT training datasets affect downstream RL performance. As shown in Table 3, models fine-tuned with EquivAug consistently achieve the highest ACC_{GRPO} across all model scales and datasets, confirming that equivalence-based augmentation provides the strongest foundation for post-training RL. More importantly, we highlight two key observations:

Reasoning diversity correlates with RL performance. We observe a clear positive relationship between RPD after SFT and accuracy after GRPO: methods that produce higher reasoning diversity during SFT tend to yield stronger RL performance. For instance, on Qwen2.5-1.5B, EquivAug achieves both the highest RPD, followed by Ground Truth, which ranks second in both diversity and post-RL accuracy To quantify this relationship, we compute the Spearman rank correlation (Zar, 2005) between RPD and ACC_{GRPO} rank across all methods for each dataset, averaged over model scales. **The resulting correlations are $\rho = 0.87$ on GSM8K and $\rho = 0.67$ on FOLIO, with an overall average of $\rho = 0.77$.** This strong positive correlation substantiates our central Hypothesis 1: training on diverse reasoning paths during SFT broadens the initial policy's coverage of the solution space, facilitating more effective exploration and stronger downstream RL performance.

SFT accuracy does not predict RL accuracy. Interestingly, EquivAug is not consistently the best method at the SFT stage. Its ACC_{SFT} often falls below the strongest baseline. However, after GRPO, EquivAug surpasses all baselines across settings. We attribute this to the nature of the cross-entropy training objective used in SFT: fine-tuning on more diverse reasoning paths increases the entropy of the model's output distribution, making it appear less confident and thus less accurate under greedy decoding at the SFT stage. Yet this higher entropy is precisely what benefits RL, as it preserves a broader policy distribution from which the RL algorithm can explore and optimize. We provide empirical evidence for this by visualizing the entropy trajectories of different models during GRPO training in Appendix E. This finding reveals that the primary advantage of EquivAug lies not in higher immediate supervised accuracy, but in better compatibility with RL-based optimization. Even when this advantage is not fully reflected in SFT accuracy, it becomes evident once the model is further optimized with GRPO.

5.4 Ablation and OOD Generalization (RQ3)

To further analyze the relationship between diversity and RL performance, we conduct several additional experiments on EquivAug. More results on entropy and accuracy can be found in Appendix D.

Amount of augmented data. First, we aim to investigate how the number of diverse reasoning paths impacts downstream RL performance. To do this, we vary the maximum number of augmented solutions generated per problem ($K \in \{3, 5, 10, 20\}$) and train the Qwen2.5-1.5B model on GSM8K. The results are summarized in Table 4. We find that a moderate amount of augmentation (max 5 solutions per problem) yields the best GRPO accuracy, while further increasing the number of solutions degrades performance. We attribute this to a trade-off: too few variants provide insufficient diversity for exploration, whereas too many variants over-smooth the output distribution. Based on this finding, we use a maximum of 5 solutions per problem in all GSM8K experiments.

Complexity of transformations. Next, we examine whether applying more transformation rules to each solution leads to better performance. We conduct an experiment by controlling the transformation complexity, defined as the number of times transformation rules are sequentially applied to an already modified reasoning path on GSM8K and Qwen2.5-1.5B. The relationship between transfor-

LLM	Qwen2.5-1.5B			Qwen2.5-3B			Llama3.1-8B		
Metrics	RPD	ACC_{SFT}	ACC_{GRPO}	RPD	ACC_{SFT}	ACC_{GRPO}	RPD	ACC_{SFT}	ACC_{GRPO}
Dataset: GSM8K									
Ground Truth	0.1132	55.88	76.87	0.1560	63.59	83.00	0.0952	62.08	72.47
RFT	0.0815	55.66	74.37	0.1087	67.61	80.66	0.0619	66.35	71.87
Paraphrasing	0.0318	**62.29**	74.52	0.1925	**75.31**	82.90	0.1051	**72.22**	75.20
EquivAug	**0.1200**	61.71	**76.95**	**0.2087**	73.36	**83.09**	**0.1102**	67.81	**76.35**
Dataset: FOLIO									
Ground Truth	0.0001	44.83	64.08	0.0001	**69.56**	75.61	0.0001	64.24	73.39
RFT	**0.4998**	49.75	64.97	0.1338	60.30	68.97	0.0878	69.56	74.28
Paraphrasing	0.1280	**61.58**	62.56	0.1255	65.02	72.51	0.1213	65.71	75.17
EquivAug	0.1720	56.06	**65.41**	**0.1745**	67.49	**78.27**	**0.1718**	**70.94**	**75.61**

Table 3: Results on GSM8K and FOLIO across different backbone models. We report RPD, ACC_{SFT}, and ACC_{GRPO} for each data augmentation method. EquivAug produces more diverse reasoning paths after SFT and achieves the best performance after GRPO across all settings.

Max solutions per problem	# Training samples	ACC_{SFT}	ACC_{GRPO}
3	14972	0.6118	74.22
5	21834	**0.6171**	**75.06**
10	36222	0.5991	74.38
20	56529	0.6068	73.09

Table 4: Effect of the maximum number of solutions per problem on performance (GSM8K on Qwen2.5-1.5B).

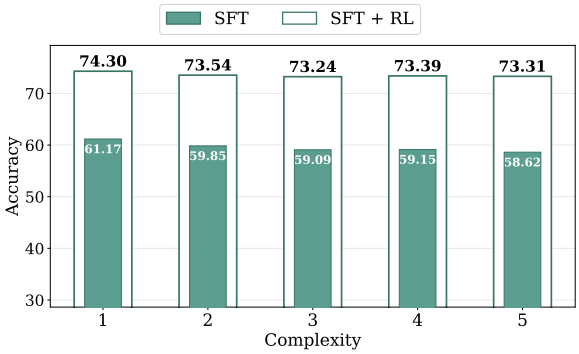

Figure 2: Effect of transformation complexity on performance (GSM8K, Qwen2.5-1.5B).

mation complexity and downstream performance is illustrated in Figure 2. The results show that a complexity level of 1 yields the best performance. Applying too many sequential transformations degrades accuracy, likely because heavily manipulated reasoning paths become unnatural or structurally strange, which confuses the policy.

OOD Generalization. Finally, We evaluate whether the diversity introduced by EquivAug transfers to out-of-distribution settings. Using Qwern2.5-1.5B trained on GSM8K, we evaluate on SVAMP (Patel et al., 2021), a distribution-shifted arithmetic reasoning benchmark. Results are shown in Figure 3. Similarly, EquivAug outperforms all baselines after RL, indicating that the diverse reasoning strategies acquired during SFT are not merely dataset-specific shortcuts but generalizable problem-solving capabilities that remain beneficial under distribution shift.

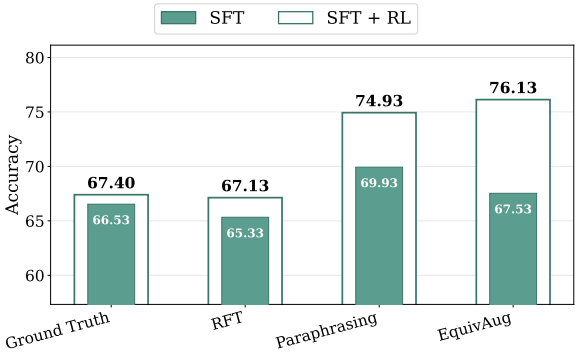

Figure 3: Out-of-distribution performance of different data augmentation methods on SVAMP.

6 Conclusion

In this work, we investigated the relationship between chain-of-thought diversity during supervised fine-tuning and downstream reinforcement learning with verifiable reward performance. We proposed EquivAug, a symbolic data augmentation framework that generates structurally diverse yet provably correct reasoning paths through equivalence-preserving transformations and solver-based verification. Experiments on GSM8K and FOLIO across multiple model scales consistently demonstrate that EquivAug produces more diverse reasoning paths after SFT and achieves stronger performance after GRPO. Our findings establish that reasoning-step-level diversity, rather than surface-level lexical variation, is the key factor in preparing an effective policy foundation for RLVR.

Limitations

This work has two main limitations. First, the equivalence rule libraries used in EquivAug are manually designed based on well-known mathematical and logical identities. While these rules are sufficient to produce meaningful diversity in the current evaluation domains, automatically discovering, learning, or expanding domain-specific equivalence rules is a promising direction that we leave for future work. Second, our experiments are conducted on models ranging from 1.5B to 8B parameters. Although the benefits of reasoning diversity are consistent across these scales, it remains an open question whether these gains persist, diminish, or amplify as model size increases further. Investigating the interplay between reasoning diversity and model capacity at larger scales is an important avenue for future research.

References

V Alfred, S Monica, Sethi Ravi, Ullman Jeffrey D, and 1 others. 2007. *Compilers Principles, Techniques*. Pearson.

Jack Chen, Fazhong Liu, Naruto Liu, Yuhan Luo, Erqu Qin, Harry Zheng, Tian Dong, Haojin Zhu, Yan Meng, and Xiao Wang. 2025. Step-wise adaptive integration of supervised fine-tuning and reinforcement learning for task-specific llms. *arXiv preprint arXiv:2505.13026*.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. In *International Conference on Machine Learning*, pages 10818–10838. PMLR.

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

Herbert B Enderton. 2001. *A mathematical introduction to logic*. Elsevier.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. Folio: Natural language reasoning with first-order logic. In *Proceedings of EMNLP*, pages 22017–22031.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Feng Ju, Zeyu Qin, Rui Min, Zhitao He, Lingpeng Kong, and Yi R Fung. 2025. Reasoning path divergence: A new metric and curation strategy to unlock llm diverse thinking. *arXiv preprint arXiv:2510.26122*.

Feiyang Kang, Michael Kuchnik, Karthik Padthe, Marin Vlastelica, Ruoxi Jia, Carole-Jean Wu, and Newsha Ardalani. 2025. Quagmires in sft-rl post-training: When high sft scores mislead and what to use instead. *arXiv preprint arXiv:2510.01624*.

Chengpeng Li, Zheng Yuan, Hongyi Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. 2024a. Mugglemath: Assessing the impact of query and response augmentation on math reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10230–10258.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024b. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2961–2984.

Zenan Li, Zhi Zhou, Yuan Yao, Xian Zhang, Yu-Feng Li, Chun Cao, Fan Yang, and Xiaoxing Ma. 2024c. Neuro-symbolic data generation for math reasoning. *Advances in Neural Information Processing Systems*, 37:23488–23515.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Kohsei Matsutani, Shota Takashiro, Gouki Minegishi, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. 2026. RL squeezes, SFT expands: A comparative study of reasoning LLMs. In *The Fourteenth International Conference on Learning Representations*.

William McCune. 2005. Release of prover9. In *Mile high conference on quasigroups, loops and nonassociative systems, Denver, Colorado*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2080–2094.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Yuda Song, Julia Kempe, and Remi Munos. 2025. Outcome-based exploration for llm reasoning. *arXiv preprint arXiv:2509.06941*.

Yangzhen Wu, Shanda Li, Zixin Wen, Xin Zhou, Ameet Talwalkar, Yiming Yang, Wenhao Huang, and Tianle Cai. 2026. Learn hard problems during rl with reference guided fine-tuning. *arXiv preprint arXiv:2603.01223*.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Weihao You, Shuo Yin, Xudong Zhao, Zhilong Ji, Guoqiang Zhong, and Jinfeng Bai. 2024. Mumath: Multi-perspective data augmentation for mathematical reasoning in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2932–2958.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *ICLR*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *Preprint*, arXiv:2308.01825.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?

Jerrold H Zar. 2005. Spearman rank correlation. *Encyclopedia of biostatistics*, 7.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. 2025. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*.

Algorithm 1 EquivAug

Require: Seed set $\mathcal{D}$ of (problem, solution) pairs; rule sets $\mathcal{R}_{\text{math}}$, $\mathcal{R}_{\text{logic}}$; domain $\delta \in \{\text{math}, \text{logic}\}$
Ensure: Augmented dataset $\mathcal{D}'$

```
D' ← ∅
for each (p, s) ∈ D do
    Parse s into symbolic representation (SR) φ
    S ← ∅
    repeat
        Sample rule r ∈ R_δ and apply r to φ to get ψ
        if ψ ≢ φ ∧ ψ ∉ S then
            if VERIFY(ψ) then
                S ← S ∪ {ψ}
            end if
        end if
    until iteration bound reached
    for each ψ ∈ S do
        Render ψ to text
        D' ← D' ∪ {(p, render(ψ))}
    end for
end for
return D'
```

A Theorem Prover: Prover9

For TRUE instances, the original conclusion C serves as the conjecture, so a successful refutation establishes $\text{Premises} \models C$. For FALSE instances, we instead use $\neg C$ as the conjecture; a successful refutation then demonstrates $\text{Premises} \models \neg C$. In both cases, the prover returns a refutation trace of the same structural form. For UNCERTAIN instances, neither C nor $\neg C$ is derivable from the premises. We verify this by employing a finite model builder, Mace4, to construct one model satisfying $\text{Premises} \cup \{C\}$ and another satisfying $\text{Premises} \cup \{\neg C\}$. The existence of both models confirms that the premises do not determine the truth value of C.

B Algorithm

In Algorithm 1, we show the pseudo code for EquivAug.

C Experiment Details

C.1 Configurations

Supervised Fine-Tuning. All SFT experiments use the Hugging Face Transformers library. We apply full-parameter fine-tuning for Qwen2.5-1.5B and Qwen2.5-3B, and LoRA (Hu et al., 2022) for Llama-3.1-8B. The two settings differ only in learning rate: full fine-tuning uses 2e-5, while LoRA uses 2e-4. All other hyperparameters are shared: AdamW optimizer with weight decay 0.01, cosine learning rate schedule with 100 warmup steps, per-device batch size of 16, and BF16 mixed precision. We train for 3 epochs on all datasets. For all methods except for the ground-truth, we train models on approximately 30,000 samples for GSM8K and 8,000 for FOLIO to ensure a fair comparison between EquivAug and other methods.

Reinforcement Learning (GRPO). We adopt GRPO (Shao et al., 2024) implemented via verl 0.8.0dev0 [2] with vLLM [3] 0.11.2 as the inference backend. All RL experiments are conducted on 2 Nvidia A100-80G GPUs using SDPA attention. The reward signal is binary, exact-match correctness of the final answer. We use a learning rate of 1e-6, KL penalty coefficient $\beta = 0.0005$, sampling temperature of 0.7, 8 rollouts per prompt, and BF16 precision. We use 3 epochs for GSM8K and 10 epochs for Folio. For all methods, we use the original training set from both datasets as the training set for GRPO.

Adopted from metamath (Yu et al., 2024), we use the following training prompt for both SFT and RL:

Training Template for SFT and RL

Below is an instruction that describes a task. Write a response that appropriately completes the request.
Instruction:
{{ message['content'] }}
Response:

C.2 Metric: RPD

To quantify the step-level diversity between reasoning paths, we adopt the Reasoning Path Divergence (RPD) metric (Ju et al., 2025). Given two solutions S_A and S_B, RPD first uses an LLM to decompose each into sequences of core logical step summaries: $S_A \rightarrow \{a_1, a_2, \ldots, a_m\}$ and $S_B \rightarrow \{b_1, b_2, \ldots, b_n\}$, where each a_i or b_j is a concise semantic description of a single reasoning step, abstracting away surface-level narrative and computational details. All step summaries are then encoded into vector representations $\mathbf{e}(\cdot)$ via

[2] https://github.com/volcengine/verl
[3] https://github.com/vllm-project/vllm

an embedding model. Without loss of generality, let $m \leq n$. For each step a_i in the shorter solution, RPD computes its minimum cosine distance to all steps in the longer solution, and the final score averages these distances:

$$\mathrm{RPD}(S_A, S_B) = \frac{1}{m} \sum_{i=1}^{m} \min_{j=1,\ldots,n} \left(1 - \frac{\mathbf{e}(a_i) \cdot \mathbf{e}(b_j)}{\|\mathbf{e}(a_i)\| \, \|\mathbf{e}(b_j)\|}\right). \tag{2}$$

This asymmetric design that always matches from the shorter to the longer solution ensures that a genuinely novel reasoning step is recognized as distant even when the remaining steps are subsumed by the longer solution. A higher RPD score indicates greater strategic divergence between two reasoning paths. In this paper, following the setting in the original RPD paper, we use Qwen-3-embedding-8B (Yang et al., 2025) as the embedding model.

C.3 Prompt Template

Prompt Template for Translating GSM8K to Symbolic Presentation

System Prompt:
You convert a natural-language math solution into exactly ONE arithmetic expression.
Output rules:
Output ONLY a single expression. Do not output any text, explanation, equals sign, numeric result, or markdown/code fences.
Allowed characters: digits 0-9, decimal point '.', operators '+-*/', and parentheses '()' only. No variables, no units, no commas, no words.
If the answer contains annotated intermediate steps of the form «expr=result»result (or similar like expr=«expr=result»result), then reconstruct the final expression by replacing numeric intermediate results with their original expressions recursively. Wrap each replacement in parentheses when substituting to preserve order.
If the answer shows intermediate calculations without explicit «...» annotations (for example 3*2=6 then 6*2=12 then 12*52=624), infer the chain and expand likewise, replacing each derived number with its generating expression.
Do not evaluate or simplify numeric subexpressions. Keep the original operations. For fractions prefer the original fraction notation rather than its decimal form.
If an intermediate number has no known generating expression in the answer, leave that number as-is.
Keep parentheses to preserve the intended order of operations. Add parentheses around substituted subexpressions to avoid ambiguity.
If you cannot reconstruct an expanded expression, output a single expression built from the final arithmetic line as-is (still obey rules 1 and 2).
Use deterministic settings (temperature=0) and return a single-line expression.

User Prompt:
For example, given the natural-language answer:
He writes each friend 3*2=«3*2=6»6 pages a week. So he writes 6*2=«6*2=12»12 pages every week. That means he writes 12*52=«12*52=624»624 pages a year.
The correct final arithmetic expression is:
((3*2)*2)*52
Natural-language answer:
{answer}
Return only the single final arithmetic expression that expands intermediate results into their original subexpressions (see system rules).

Prompt Template for Translating Back GSM8K to Natural Language

User Prompt:
Your task is to write a natural language solution for a math word problem based on a given algebraic expression.
The solution should:
1) Imitate the style, tone, and level of detail of the original natural language solution, but do not use any special formatting such as « » or dollar signs.
2) Include the final line starting with '####' and the numeric answer.
3) Do NOT simply copy the original solution. Adjust the order and grouping of computations so that they match the NEW expression.
4) Only output the new natural language solution, with no extra comments or explanations.
Here is the math problem, its ORIGINAL algebraic solution expression, and its original natural language solution:
[Problem]
{question}
[Original Expression]
{expr_original}
[Original Natural Language Solution]
{answer_text}
Here is the NEW algebraic solution expression that is mathematically equivalent and still solves the SAME word problem:
[New Expression]
{expr_variant_clean}
Write your new natural language solution below:

Prompt Template for Translating FOLIO Proofs Back to Natural Language

System Prompt:
You are an expert at explaining logical reasoning in clear, natural English. You translate formal logical proofs into plain-language explanations that read like a thoughtful person working through the problem. Vary your style — sometimes start with the conclusion and work backward, sometimes build up from the premises, sometimes highlight the key insight first. Do NOT use rigid formatting like 'Step 1:' or 'Step 2:'. Write flowing prose, as if explaining to a smart friend.

True User Template:
Below is a logical reasoning problem. The premises logically entail the conclusion.
I will show you the premises (in both natural language and formal logic), the conclusion, and a formal proof

from an automated theorem prover.
Your job: write a natural language explanation of WHY the conclusion follows from the premises. Trace the reasoning in your own words, referencing the key logical steps. End your explanation with a clear verdict line: 'Therefore, the answer is: True'
[Premises - Natural Language]
{record.get('premises_nl', '(not available)')}
[Premises - Formal Logic]
{record.get('premises_fol', '')}
[Conclusion]
{record.get('conclusion_fol', '')}
[Formal Proof (Prover9)]
{record.get('solution', '')}
Write your natural language explanation below. Be natural and varied in style — do not mechanically list steps. End with 'The answer is: True'

False User Template:
Below is a logical reasoning problem. The premises logically CONTRADICT the conclusion — the negation of the conclusion can be proved from the premises.
I will show you the premises, the conclusion, and a formal proof that the negation holds.
Your job: write a natural language explanation of WHY the conclusion is false given the premises. Show how the premises lead to the opposite of what the conclusion claims. End your explanation with: 'Therefore, the answer is: False'
[Premises - Natural Language]
{record.get('premises_nl', '(not available)')}
[Premises - Formal Logic]
{record.get('premises_fol', '')}
[Conclusion]
{record.get('conclusion_fol', '')}
[Formal Proof (negation proved)]
{record.get('solution', '')}
Write your natural language explanation below. Be natural — do not mechanically list steps. End with 'The answer is: False'

Uncertain User Template:
Below is a logical reasoning problem. The conclusion is INDEPENDENT of the premises — it can be neither proved nor disproved from what is given. There exist scenarios where the premises are true and the conclusion holds, and other scenarios where the premises are true but the conclusion fails.
Your job: write a natural language explanation of WHY we cannot determine the truth value. Point out what information is missing or why the premises don't constrain the conclusion. End your explanation with: 'Therefore, the answer is: Uncertain'
[Premises - Natural Language]
{record.get('premises_nl', '(not available)')}
[Premises - Formal Logic]
{record.get('premises_fol', '')}
[Conclusion]
{record.get('conclusion_fol', '')}
[Solver Output]
{record.get('solution', '')}
Write your natural language explanation below. Be natural — explain what's missing or ambiguous. End with 'The answer is: Uncertain'

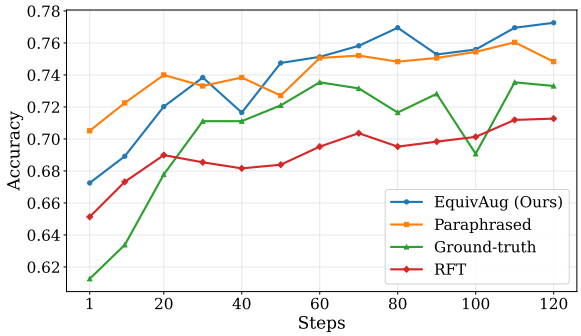

Figure 4: Test accuracy for different models during GRPO training for GSM8K on Llama3.1-8B.

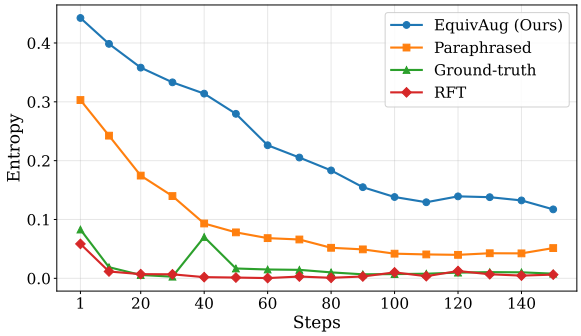

Figure 5: Entropy of different models during GRPO training for folio on Llama3.1-8B.

D More Experimental Results

Here, we present more results on the accuracy trends and entropy.

D.1 Accuracy Trends

In Figure 4, we show the evaluation accuracy trend during the GRPO training for GSM8K on Llama3.1-8B. The evaluation accuracy shows that even though EquivAug has lower initial accuracy, models trained on EquivAug can learn more effectively and outperform other methods soon.

D.2 Entropy

In Figure 5, we show how entropy changes for all methods on GSM8K and Llama3.1-8B.The results show that EquivAug has the highest entropy, further validating that EquivAug has a better exploration ability. Besides, the results show that with the training of GRPO, the entropy will decrease, which might be the reason that RFT has the lowest entropy since RFT can be seen as a weak version of RL without advantages and negative samples.

E AI Assistant Usage

We used GPT for grammar checking and Copilot as an assistive tool.