RuleReasoner: Reinforced Rule-based Reasoning via Domain-aware Dynamic Sampling

Yang Liu⋆, Jiaqi Li⋆, and Zilong Zheng✉

April 25, 2026

What is Rule Reasoning?

[Taken Fact] Socrates is a man.
[Taken Rule] All men are mortal.
[Derived Fact] Therefore, Socrates is mortal.

Task	Context (Explicit or Implicit Rules)	Question	Answer
ProofWriter	**RULES:** *If the bear needs the dog and the dog visits the bear then the bear likes the cat.If something is rough then it likes the dog* **FACTS:** *The bear is round.The bear visits the catc*	The bear needs the cat?	True
ProntoQA	**RULES:** *Everything that is earthy and a wumpus is an impus. Everything that is dull and a brimpus is a numpus* **FACTS:** *Sally is dull. Sally is a brimpus.*	Sally is dull and a brimpus?	True
Clutrr	**RULES:** *If B is the son of A, and C is the grandmother of B, then C is the mother of A.* **FACTS:** *Pedro is taking his wife Dorothy out to dinner for their date tonight. Tracy loves cooking for her son. Tracy went to the store with her sister Dorothy.*	How is Shantel related to Pedro?	Shantel is the mother -in-law of Pedro.
LogicNLI	**RULES:** *All not fierce people are not brainy. If there is at least one people who is not intelligent, then Keaton is fragile and Jaime is fierce.* **FACTS:** *Jaime is fragile. Philip is not sociable. Jaime is brainy.*	Landon is not intelligent.	Entailment
FOLIO	*Rafa Nadal was born in Mallorca. Rafa Nadal is a professional tennis player. Nadal's win ratio is higher than 80%. All players in the Big 3 are professionals who have a high win ratio.*	Nadal was not born in Mallorca.	False
Logical Deduction	*On a shelf, there are five books: a blue book, a red book, a purple book, a gray book, and a white book. The white book is to the right of the gray book. The blue book is the leftmost. The red book is to the left of the gray book. The red book is the third from the left.*	Which of the following is true? A) The blue book is the second from the right. B) ... C) ...	D
AR-LSAT	*Eight new students—R, S, T, V, W, X, Y, Z—are being divided among exactly three classes—class 1, class 2, and class 3. Classes 1 and 2 will gain three new students each; class 3 will gain two new students.*	If T is added to class 3, which one of the following is a student who must be added to class 2?	C
LogiQA	*Xiao Ming forgot what day it was today, so he asked O, P, and Q. O replied I also forgot what day it is today, but you can ask P and Q both. P replied Yesterday It's the day when I lied. Q's answer is the same as P. It is known that 1.O never lied;*	What day is today? A) Monday B) Tuseday C) Thursday D) Sunday	C

Figure: Demonstration of rule reasoning data examples.

Why does Rule Reasoning matter?

- Rule-based reasoning is a sort of the very basic and important, yet not well solved task, for example, law reasoning and every other tasks can be formulated as "syllogism";
- Previous work (*e.g.*, prompt engineering, suprevised distillation, and external memory bank) fails on long-context, complex, and noisy rule application. These methods obeys the "First principles thinking";
- Even frontier large reasoning models (*e.g.*, O1, R1, Gemini-2.5-Pro, and Claude-Sonnet-3.7) fail on many cases from rule-based reasoning.

What is RuleReasoner?

- Rule-based reasoning is a fundamental problem in AI, but real-world rule variations pose challenges.
- Large reasoning models (LRMs) show great promise, especially with reinforcement learning with verifiable reward (RLVR), yet limited in math & code problems.
- However, it is an open question whether (small) models can effectively learn rule-based reasoning.
- RuleReasoner is a new method that helps smaller models learn rule-based reasoning using a large set of curated tasks and a useful domain-aware dynamic sampling algorithm.
- It dynamically adjusts the sampling of training data based on past rewards, which helps the model learn more efficiently.

Data Curation

The following table shows the statistics of the curated tasks and data used in this work.

Dataset	# Train/Test	Task Format	Reasoning Form	Reasoning Depth	Fiction Rule	Rule of Inference
ProofWriter (Tafjord et al., 2021)	7,997/500	Boolean	Deduction	[0, 5]	✓	MP, UI
ProntoQA (Saparov & He, 2023)	8,000/500	Boolean	Deduction	{1, 3, 5}	✓	UI, Conjunction Simplification
Clutrr (Sinha et al., 2019)	268/67	Free Text	Induction†	–	✗	HS
FOLIO (Han et al., 2024)	1,208/242	MC	FOL	[0, 7]	✓	MT, DS, UI
LogicNLI (Tian et al., 2021)	8,000/500	MC	FOL	[1, 5]	✓	MP, MT
AR-LSAT (Zhong et al., 2022)	1,636/410	MC	AR	–	✓	MP, MT
Logic. Dedu. (Xu et al., 2024a)	1,200/300	MC	CS	{1, 3, 5}	✓	MP, MT
LogiQA (Liu et al., 2023a)	264/67	MC	CCR	–	✓	MP, MT

Figure: Data statistics of curated tasks and data.

Training Recipe & Optimization.

- We employ verifiable reward via strict exact match for NL answers;
- We discard KL loss, encouraging exploration instead of exploitation;
- We discard the entropy bonus, to avoid entropy explosion in Zero-RL.

$$\mathcal{R}_{\text{EM}}(\hat{y}, y) = \begin{cases} 1 & \text{is\_equivalent}(\hat{y}, y), \\ -1 & \text{otherwise}. \end{cases} \quad (1)$$

Policy Optimization. We adopt the basic form of GRPO (Shao et al., 2024) but discard the part of KL loss, encouraging the model to explore various solutions. For each question-answer pair (q, y), the policy model $\pi_{\theta_{\text{old}}}$ samples to generate a group of responses $\{y_1, y_2, \ldots, y_G\}$ and calculates the associated rewards $\{r_1, r_2, \ldots, r_G\}$, given the oracle answer y, using the aforementioned reward function $\mathcal{R}_{\text{EM}}$.

$$\mathcal{J}(\theta) = \mathbb{E}_{(q,y)\sim\mathcal{D},\, \{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\left(\min\left(r_{i,t}(\theta)A_{i,t}, \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)A_{i,t}\right)\right)\right], \quad (2)$$

where $r_{i,t}(\theta)$ is the rate of importance sampling for domain d_i at the t-th token for y_i, and A_i is the advantage as the critic obtained by normalizing the rewards within each group. We strictly follow the on-policy training method, performing only one gradient update after the policy model $\pi_{\theta_{\text{old}}}$ generates a group of G rollouts, to enable stable RL training and prevent entropy collapse.

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid q, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid q, y_{i,<t})}, \; A_i := \tilde{r}_i = \frac{r_i - \text{mean}\left(\{r_1, r_2, \cdots, r_G\}\right)}{\text{std}\left(\{r_1, r_2, \cdots, r_G\}\right)}. \quad (3)$$

Figure: Objective for RLVR optimization.

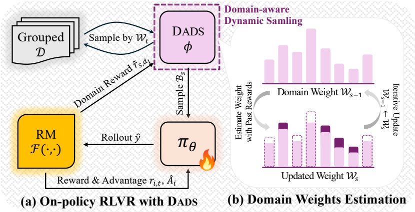

Figure: Diagram of RuleReasoner training recipe.

The following algorithm describes the Domain-aware Dynamic Sampling (DADS) method, enabling policy conduct active learning via auto historical domain reward tracking & domain weight estimation for training samples.

Algorithm 1 Domain-aware Dynamic Sampling

Input: Policy model: $\pi_\theta : \mathcal{X} \rightarrow \mathcal{Y}$;
Reward model: $\mathcal{R}_{\text{EM}}(\cdot,\cdot) : \mathcal{Y}, \hat{\mathcal{Y}} \rightarrow \{0,1\}$;
Last weight: $\mathcal{W}_{s-1} := \{w_1, w_2, \ldots, w_n\}$;
Grouped data: $\mathcal{D} := \{(d_i, q_i, r_i, y_i)\}_{i=1}^{n}$
where domain: $d_i \in \{d_1, \cdots, d_n\}$;
Hyperparameters: $\{\alpha, \epsilon, \tau\} \subset \mathbb{R}^+$.
Output: Constructed batch of samples: $\mathcal{B}_s$.

procedure TRAIN STEP s SAMPLING
  Initialize: $\mathcal{B}_{s-1} \leftarrow \mathcal{W}_{s-1} \times \mathcal{D}$; $\tilde{r}_{0,d_i} \leftarrow 0$.
  $\mathcal{Y}_{s-1} \leftarrow \pi_\theta(\mathcal{B}_{s-1})$ ▷ ROLLOUT
  $\{\{r_{s,d_i,j}\}_{j=1}^{m_i}\}_{i=1}^{n} \leftarrow \mathcal{R}_{\text{EM}}(\mathcal{Y}_{s-1}, \hat{\mathcal{Y}}_{s-1})$
  /* Update estimated rewards */
  $\{\bar{r}_{s,d_i}\}_{i=1}^{n} \leftarrow \{\frac{1}{m_i} \sum_{j=1}^{m_i} r_{s,d_i,j}\}_{i=1}^{n}$
  $\{\widetilde{r}_{s,d_i}\}_{i=1}^{n} \leftarrow \{\alpha \widetilde{r}_{s-1,d_i} + (1-\alpha)\bar{r}_{s,d_i}\}_{i=1}^{n}$
  /* Calculate weights by rewards */
  for $i = 1, 2, \ldots, n$ **do**
    $v_{s,d_i} \leftarrow 1 - \widetilde{r}_{s,d_i}$
    $w_{s,d_i} \leftarrow \exp(v_{s,d_i}/\tau) + \epsilon$
  $\mathcal{W}_s := \{w^{\text{norm}}_{s,d_i}\}_{i=1}^{n}$ ▷ NORMALIZING
    $= \{w_{s,d_i} / \sum_{j=1}^{n} w_{s,d_j}\}_{i=1}^{n}$
  /* Re-sample w.r.t. optimized weights */
  $\mathcal{B}_s \leftarrow \mathcal{W}_s \times \mathcal{D}$ ▷ SAMPLING BY $\mathcal{W}_s$
  return $\mathcal{B}_s$

Figure: Domain-aware Dynamic Sampling Algorithm.

In-Distribution (ID) Performance

The following table compares the in-distribution performance of RuleReasoner with a large spectrum of compared baselines.

	Induction	Deduction		FOL		Others			Avg. Results
	Clutrr	ProntoQA	ProofWriter	FOLIO	LogicNLI	AR-LSAT	Logic. Dedu.	LogiQA	
Prior RBRs									
HtT (Zhu et al., 2023)	40.3	92.0	88.0	71.0	54.0	97.0	100.0	79.1	77.7
RGFB (Diallo et al., 2025)	31.3	94.0	88.0	74.0	55.0	95.0	100.0	79.1	77.1
Chain-of-Logic (Servantez et al., 2024a)	44.8	91.0	92.0	80.0	54.0	97.0	100.0	80.6	80.0
Frontier Reasoners									
OpenAI o1 (Jaech et al., 2024)	52.2	91.0	91.0	77.0	60.0	98.0	88.0	82.1	79.9
OpenAI o3-mini (Zhang et al., 2025a)	40.3	94.0	93.0	74.0	55.0	96.3	100.0	77.6	78.8
Claude-3.7-Sonnet (Anthropic, 2025)	65.7	92.8	90.0	74.7	58.0	76.2	97.0	81.5	79.5
DeepSeek-R1 (Guo et al., 2025)	71.6	40.0	27.0	72.7	49.0	89.7	98.3	85.0	66.7
Behavioral Cloning									
SFT w/o CoT (Wei et al., 2022a)	37.5	96.0	88.8	73.4	74.8	37.5	85.9	76.1	71.2
SFT w/ Short CoT (Yeo et al., 2025)	77.6	92.6	87.0	82.9	73.8	54.8	87.6	88.0	80.9
SFT w/ Long CoT (Yeo et al., 2025)	83.5	95.6	89.2	83.4	76.6	68.6	79.6	79.1	81.9
Advanced RLVRs									
GRPO (Shao et al., 2024)	73.1	95.4	96.4	72.3	66.6	36.3	90.3	70.1	75.0
Dr. GRPO (Liu et al., 2025b)	68.6	96.0	95.6	73.9	75.4	32.1	84.3	65.6	73.9
DAPO (Yu et al., 2025)	86.5	96.0	94.8	80.9	65.8	40.0	95.3	74.6	79.2
Curriculum Learning									
Data-balance RL (Parashar et al., 2025)	86.5	95.8	95.6	76.8	64.4	45.6	95.3	73.1	79.1
Easy-to-hard RL (Parashar et al., 2025)	88.0	96.2	96.8	78.9	66.6	46.3	96.0	74.6	80.4
AdaRFT (Shi et al., 2025a)	92.5	96.0	97.4	81.8	64.4	44.6	96.6	80.5	81.7
RuleReasoner (Ours)									
RuleReasoner-4B	$82.0_{0.4}$	$95.0_{0.6}$	$96.3_{0.3}$	$78.9_{0.8}$	$66.6_{0.4}$	$38.6_{0.5}$	$96.3_{0.2}$	$80.5_{0.7}$	$79.2_{0.6}$
RuleReasoner-8B	$\mathbf{95.5}_{0.3}$	$\mathbf{96.4}_{0.4}$	$\mathbf{97.0}_{0.2}$	$\mathbf{84.7}_{0.6}$	$70.4_{0.1}$	$46.8_{0.3}$	$98.3_{0.4}$	$83.5_{0.3}$	$\mathbf{84.0}_{0.5}$

Figure: Comparison with all baselines in eight ID tasks.

The following figure shows the out-of-distribution performance comparison between RuleReasoner and other frontier reasoning models.

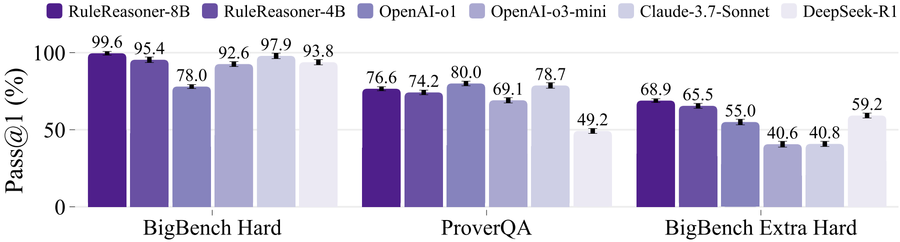

Figure: Out-of-distribution performance comparison.

The following table shows the performance gains of RuleReasoner compared to other training methods.

Model	Pass@1	Avg. Δ
Qwen3-8B	27.4 / 34.2	–
+ SFT†	81.9 / 66.6	54.5 / 34.4
+ GRPO	75.0 / 75.8	47.6 / 41.6
+ Ours	**84.0 / 81.7**	**56.6 / 47.5**

Figure: Comparison of average improvement.

Learning Dynamics by Domain

The following figure illustrates the learning dynamics by domain, showing the training reward, validation pass@1 performance, and domain weights.

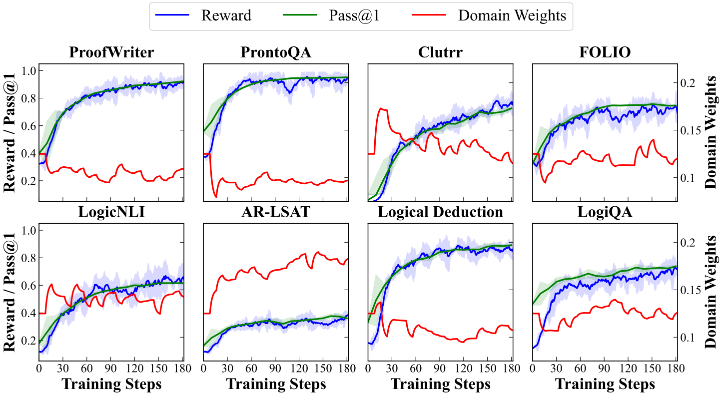

Figure: Learning dynamics by domains.

Additional Analyses

The following figure shows the performance of data curricula of task types and training sample efficiency, as well as training method efficiency.

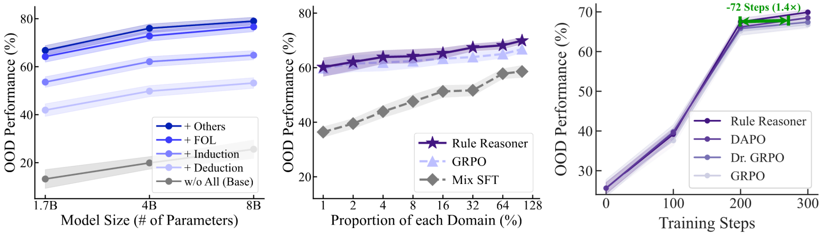

Figure: Additional evaluation and training method comparison.

Challenges of Different Rule Settings

Comparison of performance of RuleReasoner on challenging rule setups.

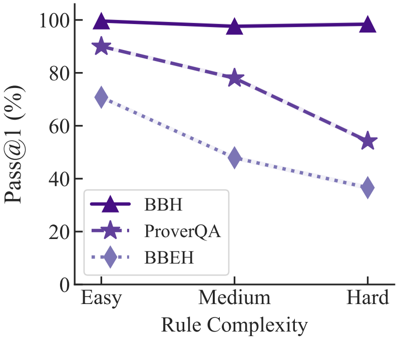

Figure: Comparison of performance on varying task complexity.

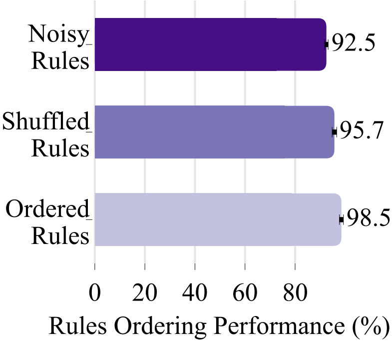

Figure: Comparison of performance on challenging rule setups.

Case Studies

The following figure presents case studies showing the cherry-picked reasoning chains before and after training with RuleReasoner.

Case Type	Input	Before Training	After Training
Explicit Rules (ProofWriter)	**Rules**: If something is smart then it is furry. If something is cold it is green. Cold, smart things are green. All furry things are white. If something is green, it is smart. All furry things are green. All cold, green things are rough. Cold things are green. **Facts**: Dave is blue. Dave is cold. Dave is green. **Question**: Dave is furry?	Step 1: Analyze the given rules and facts. Rule 1: If something is smart, then it is furry. ... Fact 1: Dave is blue. Fact 2: Dave is cold. ... Step 2: Determine if Dave is furry based on the given rules and facts. From Fact 2 and Rule 2, we know that Dave is green. From Fact 2 and Rule 8, we know that Dave is green. However, we also have Rule 4 stating that all furry things are white, and Dave is blue (Fact 1). This creates a contradiction, as Dave cannot be both furry and blue. Since we have a contradiction, we cannot determine if Dave is furry. The answer is Unknown. Answer: Unknown. ✗	Step-by-Step Reasoning: 1. Dave is cold. From Rule 8: If something is cold, then it is green. Since Dave is cold, Dave is green. This is consistent with the given fact that Dave is green. From Rule 5: If something is green, then it is smart. Since Dave is green, Dave is smart. From Rule 1: If something is smart, then it is furry. Since Dave is smart, Dave is furry. From Rule 6: All furry things are green. This is consistent with the given fact that Dave is green. 3. Conclusion: Since Dave is smart (from Rule 5), and from Rule 1, if something is smart, then it is furry, we can conclude Dave is furry. Answer: True. ✓
Implicit Rules (BBH)	**Context**: The following paragraphs each describe a set of three objects arranged in a fixed order. The statements are logically consistent within each paragraph. On a branch, there are three birds: a robin, an owl, and a hawk. The robin is to the right of the hawk. The owl is the leftmost. **Options**: (A) The robin is the rightmost (B) The owl is the rightmost (C) The hawk is the rightmost. **Question**: Which of the following options is true?	To answer this question, let's analyze the information provided in the context step by step: 1. There are three birds on a branch: ... Now, let's evaluate the options: (A) The robin is the rightmost: This option is not true because the robin is to the right of the hawk, but not necessarily the rightmost bird. (B) The owl is the rightmost: This option is not true because the owl is the leftmost bird, not the rightmost. Based on the information provided, none of the options (A, B, or C) are true. However, since we must choose one of the given options, we can conclude that the correct answer is: (B) Please note that this answer is not entirely accurate, as the owl is actually the leftmost bird. The question might have a mistake or missing information. Answer: B. ✗	To determine the correct order of the three birds on the branch, let's analyze the information provided: 1. The owl is the leftmost. This means the owl is in the first position on the branch. 2. The robin is to the right of the hawk. This means the robin comes after the hawk in the order. Now, let's determine the positions of the hawk and the robin: Since the owl is the leftmost, the owl must be in the first position. The robin is to the right of the hawk, so the hawk must be in the second position, and the robin must be in the third position. So, the order from left to right is: 1. Owl 2. Hawk 3. Robin Now, let's evaluate the options: (A) The robin is the rightmost. So, this statement is true. (B) The owl is the rightmost. ... So, this statement is false. Answer: A. ✓

Figure: Reasoning quality before and after training by RuleReasoner.

Conclusion

- RuleReasoner is a helpful training framework that enhances the effectiveness and efficiency of multi-task rule reasoning via RLVR.
- It achieves higher ID and OOD performance compared to recent RLVR methods, with redundant rollout cost (Make SRMs stronger!).
- Future work will explore more approaches to further improve reasoning model post-training efficiency, to unify all reasoning tasks.