MetaAct-RL: Training Language Models for Reasoning Through Meta-Action-Based Reinforcement Learning

Zhiheng Xi[*†1], Yuhui Wang[*1], Yiwen Ding[1], Guanyu Li[1], Senjie Jin[1], Shichun Liu[1], Jixuan Huang[1], Dingwen Yang[1], Jiafu Tang[1], Boyang Hong[1], Junjie Ye[1], Shihan Dou[1], Ming Zhang[1], Jian Guan[2], Wei Wu[2], Rui Zheng[1], Tao Gui[†1,3], Qi Zhang[†1,4], Xuanjing Huang[†1]

[1]Fudan University
[2]Ant Group
[3]Shanghai Innovation Institute
[4]wispaper.ai
zhxi22@m.fudan.edu.cn, {tgui, qz, xjhuang}@fudan.edu.cn

Abstract

Outcome-based reinforcement learning has made notable advances in training language models (LMs) for reasoning. However, without explicit incentives and controls, this paradigm has limitations and instability in eliciting high-quality reasoning trajectories with diverse actions—particularly for models whose pretraining lacked extensive reasoning-related data. To this end, we introduce MetaAct-RL, a new RL framework that frames LMs' thinking as sequential decision making over meta-actions. In this framework, the model chooses and executes a high-level action at each step—such as forward reasoning, critique, or refinement—to gradually reach the correct answer. To encourage deeper exploration, richer action diversity, and to improve sampling efficiency in the RL optimization process, MetaAct-RL incorporates appropriate length-based reward and regularization, and a key-state restart mechanism. Extensive experiments across six benchmarks show that MetaAct-RL improves reasoning performance by 7.99 on Llama3.2-1B and 7.17 on Llama3.1-8B relative to vanilla RL method. Moreover, on the challenging AIME-2024, our method outperforms the vanilla RL by 7.5 with Qwen2.5-1.5B.

1 Introduction

When faced with complex problems, humans generally engage in deep, deliberate reasoning, periodically evaluating and validating their thought processes and making appropriate corrections in order to arrive at reliable and optimal decisions (Hegel 1991; Popper 2005; Kahneman 2011). Similarly, recent research and industry efforts have shown that by increasing inference compute and guiding language models to adopt diverse reasoning behaviors (i.e., meta-actions)—such as self-reflection (Dou et al. 2024), self-critique (Xi et al. 2024c), self-correction (Pan et al. 2023), and Critique-RL (Xi et al. 2025c)—the model can effectively review and refine its reasoning process, thereby fostering deeper insights and producing higher-quality outputs. (OpenAI 2024; deepseek 2024; Team et al. 2025; Gou et al. 2024; Gao et al. 2024; Ankner et al. 2024; Welleck et al. 2023; Kumar et al. 2024).

DeepSeek-R1 demonstrated that outcome-based RL alone can elicit different reasoning modes in the model, leading to an "Aha" moment (DeepSeek-AI et al. 2025). Yet recent work (Hu et al. 2025; Gandhi et al. 2025) has highlighted that outcome-based approaches lack explicit incentives and controls and thus cannot stably or consistently guide the generation of high-quality reasoning trajectories with diverse actions—especially for models whose pretraining did not include sufficient reasoning data (AI et al. 2025). As a result, the model may suffer from a collapse of its internal deliberation and tend to output only a small set of meta-actions. The key challenge lies in how to train the model to incorporate diverse behaviors during reasoning and explore more deeply in order to obtain high-quality reasoning trajectories (Jaech et al. 2024; DeepSeek-AI et al. 2025; Xi et al. 2025b).

To this end, we propose MetaAct-RL, a new RL framework for LM reasoning that formulates the reasoning process as a sequence of meta-actions (Figure 1). Specifically, by framing the model's reasoning as a sequential decision-making process, we define a set of high-level meta-actions—such as forward reasoning, critiquing, and refining—through which the model iteratively makes decisions and executes actions to explore the solution space (Section 3). In our implementation, we first leverage a data-synthesis approach to construct reasoning traces containing diverse meta-actions to train models (Section 4.1). Next, we incorporate length-based rewards and regularization, which jointly promote deeper exploration, increased action diversity, and improved sampling efficiency (Section 4.3). Finally, we employ a key-state restart strategy to train the model to precisely generate diverse meta-actions during the RL phase (Section 4.4).

Through detailed analysis, we demonstrate that our method can stably and effectively optimize LM reasoning. Extensive experiments across six reasoning tasks show that our method consistently outperforms RL and SFT baselines (Schulman et al. 2017). Moreover, with Qwen2.5-1.5B, it can outperform

[*]These authors contributed equally.
[†]Corresponding author.

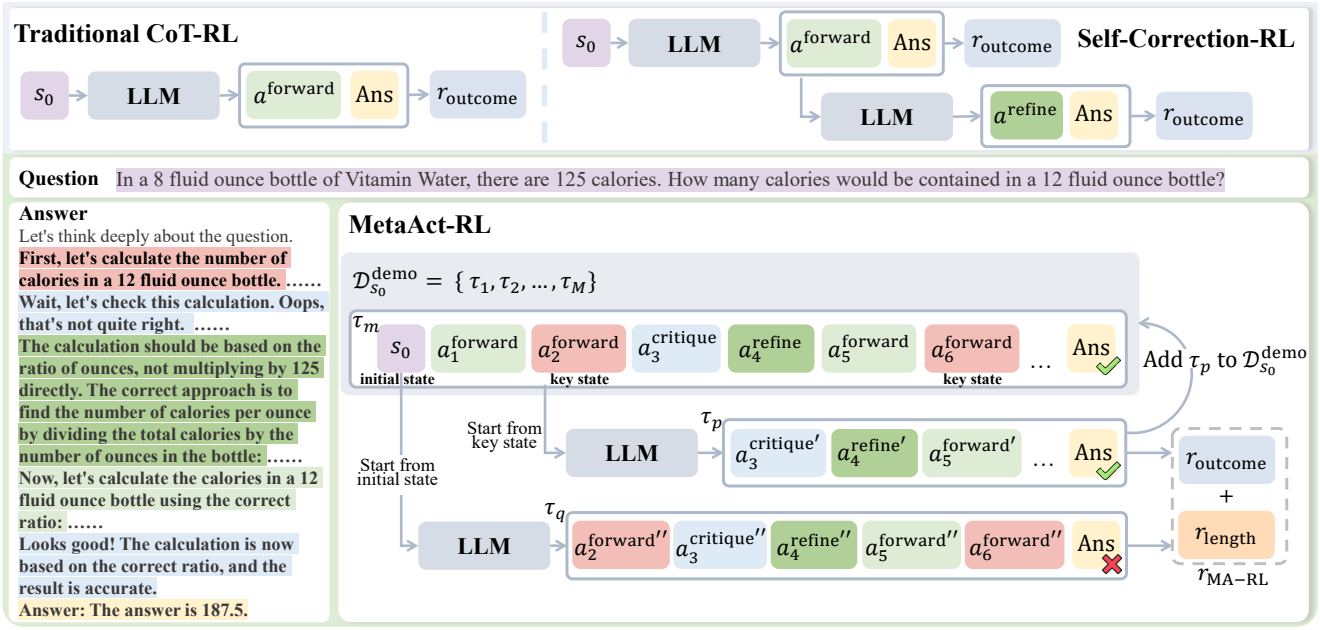


Figure 1: An overview of MetaAct-RL and other RL methods for LM reasoning. In Cot-RL, the LM generates a complete solution and then receives an outcome-based reward. In Self-Correction-RL, the LM generates a solution, corrects it regardless of its correctness, and then obtains an outcome-based reward. In MetaAct-RL, the LM adaptively performs step-level actions such as reasoning, critiquing, and refinement to obtain a reward, which includes an outcome-based reward and a length-based reward to guide deep thinking. MetaAct-RL also employs a key-state restart strategy, which allows the LM to start exploration from a key intermediate step of a demonstration, to improve exploration efficiency and enhance its ability on different actions.

traditional RL by 7.50 on the challenging AIME-2024. When combined with test-compute scaling techniques, it achieves even greater performance gains. Furthermore, MetaAct-RL can elicit new types of actions from LMs, such as divide-and-conquer and case-by-case analysis, further expanding the thinking capacity. In summary, our contributions include:

1. We introduce a meta thinking-action-based RL framework MetaAct-RL for LM reasoning, which trains models to select and execute meta actions during reasoning.
2. We incorporate appropriate length-based reward and regularization, and a key-state restart strategy for RL optimization to encourage deep exploration, diverse actions, and to improve sampling efficiency.
3. We conduct experiments across six benchmark tasks (e.g., MATH, TheoremQA, GSM8K), showing that MetaAct-RL improves reasoning performance by 7.99 on Llama3.2-1B and 7.17 on Llama3.1-8B. Moreover, on the challenging AIME-2024, our method outperforms the vanilla RL by 7.5 with Qwen2.5-1.5B. We also perform in-depth ablation and analysis to offer insights into our method.

2 Related Work

Training language models for reasoning. With the development of language models, researchers have harnessed prompt engineering to elicit their reasoning capabilities (Wei et al. 2023; Kojima et al. 2023). To further strengthen this ability, fine-tuning–based approaches have been introduced, wherein high-quality reasoning traces are annotated and constructed, and the model is trained to imitate the traces step-by-step (Huang et al. 2024b; Min et al. 2024; Team 2025; Zelikman et al. 2022; Singh et al. 2024; Xi et al. 2025a). Other lines of work, represented by OpenAI-o1 (Jaech et al. 2024) and DeepSeek-R1 (DeepSeek-AI et al. 2025) leverage reinforcement learning to encourage exploration and learning, yielding additional gains in model performance (Luong et al. 2024; Xi et al. 2024a; OpenAI et al. 2025; Team et al. 2025; Guo et al. 2025a; Xi et al. 2025d). Our method is built on these works, and it enhances the model's exploration quality and diversity through appropriate regularization and the key-state restart strategy.

Equipping language models with different reasoning actions. Recent work has shown that encouraging models to engage in specific reasoning behaviors—such as self-reflection (Dou et al. 2024), self-critique (Xi et al. 2024c), self-refinement (Xu et al. 2024), and self-correction (Pan et al. 2023)—can improve their performance (Kumar et al. 2024; He et al. 2024; Xi et al. 2023). At its core, this approach increases inference compute by having the model execute a more diverse set of reasoning strategies and perform deeper exploration, resulting in higher-quality reasoning trajectories (Muennighoff et al. 2025). Many works have employed prompt engineering (Shinn et al. 2023; Dou et al. 2024; Pan et al. 2023), supervised fine-tuning (Zheng et al. 2024; Saunders et al. 2022; Guo et al. 2025b), or reinforcement learning (Yao et al. 2024; Keskar et al. 2019; Zhou, Du, and Li 2024;

McAleese et al. 2024) to teach or reinforce these behaviors, and have reported strong empirical gains. Building on these insights, our approach explicitly enables the model to select and execute different reasoning actions, guiding it to generate high-quality and diverse reasoning trajectories. We compare our method and other RL-based methods in Table 1.

Paradigm	Action Gra.	Action Spa.	Op. Rounds	Len.
CoT-RL	Solution-level	FR	Pre-defined	×
SC-RL	Solution-level	FR, RF	Pre-defined	×
MetaAct-RL	Step-level	FR, RF, CR	Adaptive	✓

Table 1: Comparison of different RL paradigms for LM reasoning. Action Gra. means Action Granularity; Action Spa. refers to the available Action Space; Op. Rounds refers to operation rounds for model thinking; Len. refers to Length-based design. Note that in Paradigm column, SC-RL is the abbreviation for Self-Correction-RL; MetaAct-RL is our method. In Action Space column, FR means forward reasoning, RF means refinement, and CR means critiquing.

3 Framing LM Reasoning as Sequential Decision-Making Process

We formulate the reasoning process of LMs as a sequential decision-making process, specifically as a Markov Decision Process (MDP) (Puterman 1990; Qu et al. 2024). This MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma)$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ denotes the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the state transition function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

Given an input prompt $x = (x_0, \ldots, x_L) \in \mathcal{D}$, the initial state s_0 represents the token sequences of the given prompt. The LM π_θ parameterized by θ generates the action $a_t = (h_t^0, \ldots, h_t^M)$ according to its policy: $a_t \sim \pi_\theta(\cdot \mid s_t)$, where each action corresponds to an intermediate reasoning step, consisting of a sequence of tokens, and the state s_t represents the context, including the prompt and the actions generated in the previous $t-1$ steps. Specifically, we define three kinds of step-level meta-actions for the action space $\mathcal{A}$: **Forward Reasoning** ($s_t \to a_t^{\text{forward}}$), where the model generates a reasoning step based on the current state s_t; **Critiquing** ($s_t \to a_t^{\text{critique}}$), where the model produces a critique reflecting on the reasoning step in the context of state s_t; and **Refinement** ($s_t \to a_t^{\text{refine}}$), where the model refines the previous reasoning directly or according to the generated critique. After each action, the transition function $\mathcal{T}$ deterministically updates the state by $\mathcal{T} : s_{t+1} = \{s_t, a_t\}$, concatenating the tokens representing s_t with the action a_t proposed by the model. After T time steps, the trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T)$ is constructed.

Typically, the reward $r(s_t, a_t)$ is assigned to every state-action pair, providing feedback at the intermediate steps. However, in this work, we focus on the case of sparse rewards, where a reward is only assigned at the final step T. Specifically, the outcome-based reward function r verifies whether the final answer is correct.

Policy gradient for language models. The objective of RL (Sutton and Barto 2018) is to find an optimal policy that maximizes the cumulative reward (i.e., return) $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r(s_{t'}, a_{t'})$. To optimize the policy π_θ, we employ the widely-used policy gradient method (Sutton et al. 1999). The general form of the policy gradient is:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t \mid s_t) R_t \right] \tag{1}$$

and the update step for the policy gradient is given as $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$, where α is the learning rate. In the area of RL for LMs (Ouyang et al. 2022; Bai et al. 2022; Zheng et al. 2023; Wang et al. 2024), proximal policy optimization (PPO) (Schulman et al. 2017) is a popular and effective policy gradient algorithm.

4 Methodology

In this section, we first introduce an approach to construct high-quality, structured traces composed of meta-actions for model initialization. Next, we delve into the vanilla RL optimization process, and reveal its failure modes. To this end, we propose incorporate length-based rewards and regularization to promote deeper exploration. Finally, we employ a key-state restart strategy to elicit deep reasoning with diverse actions.

4.1 Construct Trace and Initialize Base Model

To enable the model to incorporate meta-actions, we first construct a high-quality trace set containing diverse meta-actions $\mathcal{D}_{\text{SFT}}$ to train a base model. The traces need to meet the following requirements: **i)** the final answer must be correct, **ii)** the traces are expected to contain diverse actions, and **iii)** the reasoning process should be natural and smooth, with different actions seamlessly connected.

To achieve this, building on the two-player paradigm of previous work, where an actor reasoning model π_θ performs forward reasoning, a critique model π_ϕ conducts critiquing, and the actor model then refines its output (Xi et al. 2024c), we propose an approach to iteratively construct reasoning traces. This approach consists of three main stages: (1) construct an initial reasoning trace using solution-critique pairs, (2) iteratively refine and critique to optimize the reasoning trace, and (3) smooth the reasoning trace. The details of this data construction are described in Appendix A.

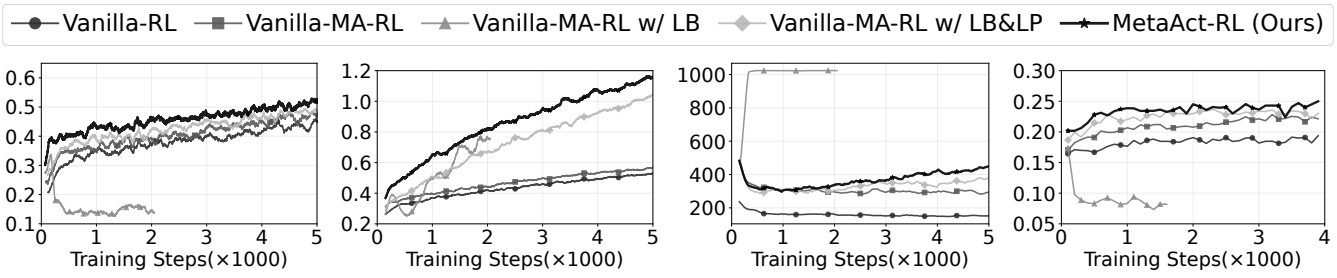
Now given the generated dataset $\mathcal{D}_{\text{SFT}} = \{x_i, y_i\}_{i=1}^{i=|\mathcal{D}_{\text{SFT}}|}$ where x_i is the query and y_i is the reasoning response, we fine-tune a base model π_θ^{SFT} via SFT with the following loss function:

$$\mathcal{L}_{\text{SFT}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{SFT}}} \Big[\log \pi_\theta(y|x) \Big]. \tag{2}$$

4.2 Result and Finding from Vanilla MetaAct-RL

Next, we delve into the RL optimization process. We define the outcome-based reward function as:

$$r_{\text{outcome}}(s_t, a_t) = \begin{cases} 1, & t = T \text{ and answer correct} \\ 0, & t \neq T \text{ or answer not correct} \end{cases}$$



(a) Training Accuracy (b) Training Reward (c) Response Length (d) Evaluating Accuracy

Figure 2: Training dynamics and evaluation accuracy of preliminary experiments. MA-RL refers to MetaAct-RL; "w/" means "with"; LB means length-based bonus; LP means length-based penalty. Our final MetaAct-RL is built upon vanilla MA-RL described in Section 4.2, and then gradually enhanced through the additional design of reward function and the key-state restart strategy (Section 4.3, Section 4.4 and Section 4.5).

Following previous work (Schulman et al. 2017; Ouyang et al. 2022), we set the total reward as sum of the outcome-based correctness score and the Kullback-Leibler (KL) divergence between the learned RL policy π_θ^{RL} and the initial reference policy π_θ^{Ref} scaled by a coefficient factor β:

$$r_{\text{vanilla}}(s_t, a_t) = r_{\text{outcome}}(s_t, a_t) - \beta\,\text{KL}\Big(\pi_\theta^{\text{RL}}(\cdot \mid s_t) \,\|\, \pi_\theta^{\text{Ref}}(\cdot \mid s_t)\Big), \quad (3)$$

where π_θ^{Ref} is the model after SFT. Now given query set $\mathcal{D} = \{s_0^i\}_{i=1}^{i=|D|}$, the policy gradient then can be written as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_0\sim\mathcal{D},\tau\sim\pi_\theta}\Big[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\, R_t\Big]. \quad (4)$$

Result: Vanilla-Meta-RL outperforms traditional CoT-RL. We perform preliminary experiments on MATH (Hendrycks et al. 2021), and the training dynamics and evaluation accuracy of vanilla MetaAct-RL and traditional RL (which perform only forward reasoning at each step) are illustrated in Figure 2. We observe that: (1) the training accuracy of vanilla MetaAct-RL increases more rapidly and significantly, and consistently surpasses that of traditional RL. This suggests that warming up with reasoning data with diverse actions can improve the depth and quality of model exploration. (2) On the test set, vanilla MetaAct-RL consistently outperforms CoT-RL, indicating that it has better generalization by generating different actions.

Finding: Vanilla MetaAct-RL demonstrates failure modes. However, Figure 2c shows that as training progresses, vanilla MetaAct-RL exhibits failure modes: the model's reasoning length shortens, and the analysis of action distribution in Figure 3 shows that compared to SFT model, the mode tends to favor forward reasoning, engaging less in critiquing and refinement. This may cause the model to rely solely on one kind of behavior when encountering unseen challenging problems, making it difficult to self-critique and correct when it generates erroneous reasoning steps, leading to a decline in performance.

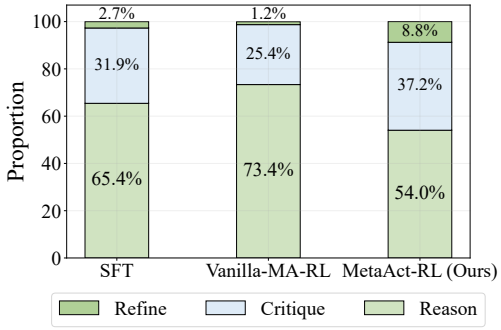


Figure 3: The action distribution of models trained with different methods. Implementation details are in Appendix D.

4.3 Length-based Reward Design for Deep and Reliable Reasoning

To address the aforementioned challenge, we propose a length-based reward design, which aims to elicit deep exploration while ensuring the reliability. This aligns with the current research trend of scaling inference compute for better reasoning performance (Brown et al. 2024; Snell et al. 2024).

Length-based bonus for eliciting long reasoning with rich actions. As the varying difficulty of queries requires reasoning traces of different lengths, the challenge lies in determining how to allocate the bonus appropriately for responses of different lengths across queries. To address this, we first establish a baseline response length for each query. In concrete, we sample 16 responses from the SFT model for each query and calculate the average response length, which serves as the baseline length L_x for that query. During RL, given a query x_i and a sampled response y_i, we compute the ratio of the current response length to the baseline length, $\frac{|y_i|}{L_{x_i}}$, and set the bonus as $\eta_1 \cdot \frac{|y_i|}{L_{x_i}}$, where η_1 is a hyperparameter that controls the bonus strength.

Length-based penalty to make reasoning more reliable. In preliminary analysis, we find that directly using the afore-

mentioned mechanism may lead to reward hacking (Amodei et al. 2016), where the model simply generates meaningless or hallucinated content (Shen et al. 2023; Chen et al. 2024a) to increase the length of its responses and, thereby, obtain a higher reward. Yet this may cause overthinking (Chen et al. 2024b) and incorrect final answer, and as RL progresses, the optimization process may collapse. This is illustrated by the orange curve in Figure 2, where the response length and training reward increase rapidly, but the training accuracy and evaluating accuracy drops, forming a reward hacking phenomenon.

In light of this, we apply a penalty for incorrect but long sampled traces. Similar to the bonus, given a sampled response y_i and its corresponding baseline length L_{x_i}, the length-based penalty is set as $-\eta_2 \cdot \frac{|y_i|}{L_{x_i}}$. Moreover, we apply lower and upper bounds for the ratio of length to ensure it stays within a reasonable range. Hence, the length-based reward function r_{length} becomes:

$$r_{\text{length}}(s_t, a_t) = \begin{cases} \text{Bonus}, & t = T \wedge \text{answer correct}, \\ \text{Penalty}, & t = T \wedge \text{answer not correct}, \\ 0, & t \neq T, \end{cases}$$

$$\text{where} \quad \text{Bonus} = \eta_1 \min\Big\{\max\big(\tfrac{|y_i|}{L_{x_i}}, \epsilon_1\big), \epsilon_2\Big\},$$

$$\text{and} \quad \text{Penalty} = -\eta_2 \min\Big\{\max\big(\tfrac{|y_i|}{L_{x_i}}, \epsilon_1\big), \epsilon_2\Big\}.$$

$$r_{\text{MA-RL}}(s_t, a_t) = r_{\text{outcome}} + r_{\text{length}}. \tag{5}$$

4.4 Key-State Restart for Generalization

Previous work has pointed out that, compared to forward reasoning, models' ability to critique and correct is more limited and harder to acquire (Huang et al. 2024a), which may be due to the fact that in pre-training stage LMs are exposed to relatively more forward reasoning data but lack data related to critique or correction (Pan et al. 2023; Huang et al. 2024a). However, such abilities are crucial for generalization performance. To address this, we propose the key-state restart strategy.

State restart for efficient and high-quality exploration. State restart, or learning from demonstration, is a widely used technique in RL (Subramanian, Jr., and Thomaz 2016; Popov et al. 2017; Plappert et al. 2018; Xi et al. 2024b). This technique allows the model to begin exploration from an intermediate step of a given demonstration (i.e., correct trace), thereby reducing sampling space, lowering exploration difficulty, and improving exploration efficiency.

In our setting, we define the demonstration set for a query s_0 as $\mathcal{D}_{s_0}^{\text{demo}} = \{\tau_1, \tau_2, \ldots, \tau_M\}$, where M is the number of demonstrations for s_0, and each demonstration is a reasoning trace that arrives at correct answer. During exploration of RL, we randomly select a demonstration τ_m, and set the state to a particular $s_i \in \tau_m$, allowing the model to begin exploring from there. In other words, the model is provided with prefix steps and is prompted to continue exploring from that point, as shown in Figure 1.

Restart from key state for better critique and refinement ability. Since we aim to enhance the model's critiquing and refinement abilities, we define the key state as the state immediately following an incorrect reasoning step and introduce a key-state restart strategy. By resetting to the states, the model is then tasked with performing appropriate critique and refinement actions to reach the correct result. In this way, we allow the model to engage in more critiquing and refinement steps to reinforce the relevant ability.

In implementation, we randomly assign the model a 50% chance to begin exploration from the initial state s_0 and a 50% chance to start from the key state for stable training. The demonstration set is initialized with the SFT traces and correct traces explored by models are added to the set. Based on this, we construct the RL training set for each round as $S_{\text{all}} = \{s_{\text{start}}^i\}_{i=1}^{|S_{\text{all}}|}$, where s_{start}^i represents the starting state of the i-th query, which could either be s_0 or the key state. Thus, our policy gradient becomes:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_{\text{start}} \sim S_{\text{all}}, \tau \sim \pi_\theta} \left[\sum_{t=j}^{T} \nabla_\theta \log \pi_\theta(a_t \mid s_t) R_t \right], \tag{6}$$

where j is the state position of s_{start} in the corresponding demonstration.

4.5 Brief Summary

Our algorithm is summarized in B. Thanks to the aforementioned designs, our MetaAct-RL demonstrates an efficient and stable optimization process (Figure 2), and rich actions (Figure 3).

5 Experiments

5.1 Experimental Setup

Datasets. In this paper, we focus on mathematical reasoning tasks. Therefore, we select six commonly used math datasets, including both free-form and multiple-choice-form, namely MATH (Hendrycks et al. 2021), GSM8K (Cobbe et al. 2021), AQuA (Ling et al. 2017), SVAMP (Patel, Bhattamishra, and Goyal 2021), MathQA (Amini et al. 2019), and TheoremQA (Chen et al. 2023). We modify their formats based on Yue et al. (2024) to facilitate answer extraction. We use the training splits of MATH, GSM8K and AQuA as our training set. We select the test splits of the three tasks as our in-domain test set; we select the test sets of SVAMP, MathQA and TheoremQA as our out-of-domain (OOD) test set. Additionally, we evaluate our method on the more challenging AIME-2024 and AIME-2025, using the filtered NuminaMath (LI et al. 2024) as the corresponding training set.

Models and baselines. We use models of varying sizes from the Llama series as our backbone models, specifically Llama3.2-1B-Instruct and Llama3.1-8B-Instruct (Dubey et al. 2024). For a more comprehensive evaluation, we also test models from the Qwen (QwenTeam 2024) series, as detailed in Appendix E. We include various reasoning baselines, including traditional Chain-of-Thought (CoT) (Wei

Base-model	Training-method	Held-in Datasets			Held-out Datasets			Average
		AQuA	GSM8K	MATH	MathQA	SVAMP	TheoremQA	
Llama3.2-1B-Instruct	CoT-STaR	34.25	33.74	14.50	35.08	49.50	8.63	29.28
	COT-RL	37.95	45.90	19.38	45.44	46.15	10.64	34.24
	Self-Correction-STaR	35.04	37.07	18.18	39.23	46.80	7.00	30.55
	Self-Correction-RL	45.28	46.70	23.30	49.55	51.90	10.13	37.81
	MetaAct-SFT	37.55	45.22	21.72	44.81	62.76	10.39	37.08
	MetaAct-STaR	36.22	33.02	14.06	35.51	54.80	8.38	30.33
	MetaAct-RL	**48.43**	**52.92**	**25.00**	**52.03**	**64.00**	**11.00**	**42.23**
Llama3.1-8B-Instruct	CoT-STaR	55.91	74.41	31.58	61.37	68.90	10.88	50.51
	CoT-RL	58.66	78.62	31.64	62.61	71.90	17.50	53.49
	Self-Correction-STaR	57.09	81.73	35.10	68.81	75.90	16.38	55.84
	Self-Correction-RL	63.39	81.35	35.40	71.42	78.60	16.38	57.76
	MetaAct-SFT	54.72	72.63	36.20	64.29	82.10	14.00	53.99
	MetaAct-STaR	59.45	81.05	35.42	71.83	83.40	16.88	58.00
	MetaAct-RL	**66.14**	**83.85**	**38.26**	**72.93**	**83.80**	**19.00**	**60.66**

Table 2: Main evaluation results. The best performance is in bold and underlined, while the second-best performance is underlined.

Method	AQuA	GSM8K	MATH	Length
MetaAct-RL (Ours)	**48.43**	**52.92**	**25.00**	513.3
Component ablation				
-w/o length-based rewards	43.70	51.78	23.38	332.7
-w/o key-state restart	44.49	50.41	23.31	481.6
-w/ restart from reasoning	43.48	51.25	23.03	476.8
Parameter ablation				
-w/ $\epsilon_1 = 0.5, \epsilon_2 = \infty$	38.58	31.46	8.46	1024.0
-w/ $\epsilon_1 = 0.0, \epsilon_2 = 1.5$	46.85	51.25	23.00	389.5

Table 3: Ablation study on model components and length reward parameters using Llama-3.2-1B. $\epsilon_2 = \infty$ represents no upper bound for $\frac{|y_i|}{L_{x_i}}$, and $\epsilon_1 = 0.0$ represents no lower bound.

et al. 2022) which performs only forward reasoning, self-correction (Madaan et al. 2023) which performs solution-level corrections. To implement these reasoning baselines, we use the SFT method, the STaR method (Zelikman et al. 2022; Huang et al. 2023) which samples high-quality data from the model itself and fine-tunes itself, and methods based on online RL (Schulman et al. 2017).

Implementation details. For evaluation, we set the temperature to 0, i.e., greedy decoding, except for the inference-compute scaling analysis where we set the temperature to 0.7. We construct a reasoning set of 24307 samples from MATH, GSM8K and AQuA. Before applying RL/STaR, we perform SFT to initialize a base model. For CoT, we follow Ding et al. (2024) to construct our training set. For self-correction, we mix the CoT reasoning data and the correction data for training. For our method, we set $\eta_1 = 1$, $\eta_2 = 0.5$, $\epsilon_1 = 0.5$, and $\epsilon_2 = 1.5$. See Appendix C for more details.

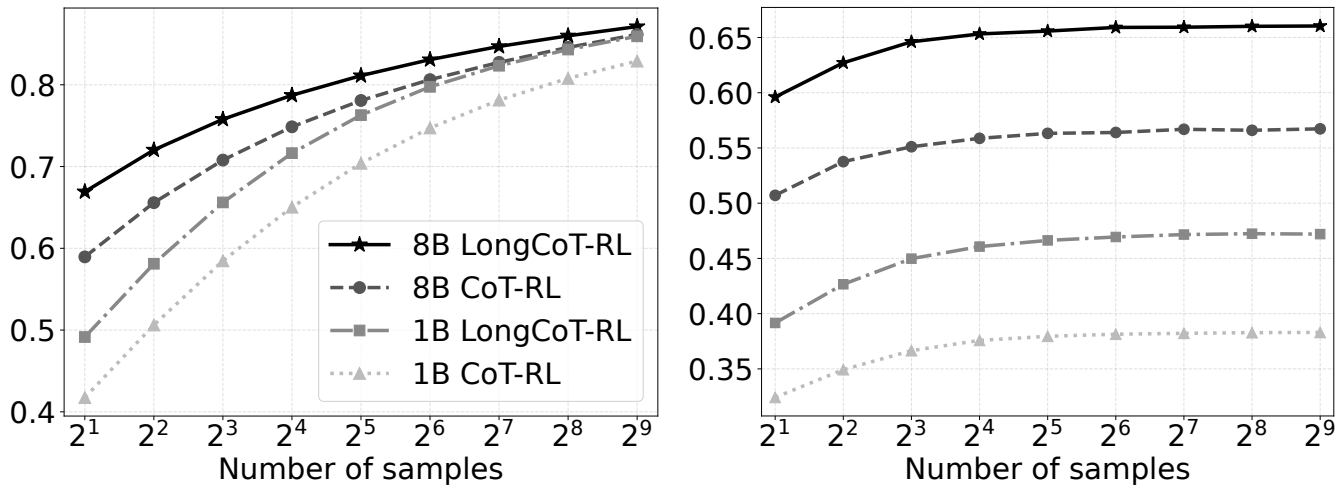
Method	AIME-2024	AIME-2025
Qwen2.5-1.5B		
-w/ CoT-RL	28.33	24.17
-w/ K1.5 (Team et al. 2025)	33.33	23.33
-w/ ADORA (Gui and Ren 2025)	31.93	25.16
-w/ MetaAct-RL (Ours)	**35.83**	**25.42**

Table 4: Results on challenging benchmarks.

5.2 Main Results

The evaluation results are listed in Table 2, we find that: (1) Overall, compared to traditional CoT methods, training LMs to engage in diverse actions for deeper thinking significantly improves their reasoning performance across various datasets and models. (2) Our constructed diverse-action-traces help build a strong base model (i.e., MetaAct-SFT), which greatly facilitates subsequent optimization. (3) MetaAct-STaR performs well on the larger 8B model but shows weaker performance on smaller 1B model. This might be because smaller models struggle to sample high-quality traces (especially on difficult queries) for self-training, leading to performance degradation. (4) In contrast, our MetaAct-RL performs well across models of different sizes, demonstrating the advantage of the on-policy RL approach. Our key-state restart strategy also contributes to this by reducing the sampling space and lowering exploration difficulty, thereby efficiently yielding high-quality traces.

Results on AIME-2024 and AIME-2025. We evaluate our method on more challenging benchmarks, and the results are in Table 4. Here we compare more baselines in length-based reward like K1.5 (Team et al. 2025), ADORA (Gui and Ren 2025). We can find that our method consistently outperforms other baselines, e.g., it surpasses the vanilla CoT-RL by 7.5 on AIME-2024.



(a) Pass@K

(b) Majority Vote@K

Figure 4: Inference-compute scaling with MetaAct-RL.

6 Discussion and Analysis

Ablation study on different components of MetaAct-RL. We perform ablation experiments to verify the importance of each component of our method. The results are presented in Table 3. We observe the following: (1) The length-based reward function r_{length} plays a crucial role in our method, as it improves model performance by encouraging long, deep and reliable thinking. (2) The key-state restart is also an important component; it enhances the model's ability to perform different actions, thereby improving performance. If we change the restart point from the key critiquing step to the forward reasoning step, which means we encourage the model to engage more in reasoning and strengthen its reasoning ability, we find that the length is reduced, and the performance drops. This also highlights the importance of reinforcing critiquing and refinement during RL.

In-depth ablation on r_{length}. Table 3 shows that if we do not set a lower bound for the length reward, performance decreases, but the model still performs reasonably well. However, if we do not set an upper bound for the length reward, it easily falls into reward hacking (similar to the orange curves in Figure 2), where the output length quickly reaches the upper limit, i.e., 1024, causing training collapse and a drop in test performance.

Inference-compute scaling with MetaAct-RL. We investigate whether MetaAct-RL can be combined with inference compute scaling strategies. We evaluate the model's Pass@K performance, which measures whether at least one correct answer exists among K sampled responses (Kulal et al. 2019), and majority vote performance, which measures whether the most frequent answer among K parallel sampled responses is correct, denoted as Maj@K (Wang et al. 2023). The results in Figure 4 show that our method is more efficient compared to traditional CoT-RL. As the sampling number scales, the Pass@K of the 1B model of MetaAct-RL approaches that of the 8B model of CoT-RL. When using majority vote, the MetaAct-RL method significantly improves the model's performance ceiling, for example, Maj@4 of MetaAct-RL substantially outperforms Maj@128 of traditional CoT-RL.

MetaAct-RL elicits new types of actions from LMs. When constructing the SFT dataset, we focus on three principal thinking actions for language models: forward reasoning, critique, and refinement. However, we find that MetaAct-RL can stimulate models to generate new types of actions, such as divide and conquer, or case-by-case analysis. These new actions can further expand the model's reasoning capacity and generalization ability.

More experiments, discussion and analysis. We include more experiments of MetaAct-RL on Qwen2.5 (QwenTeam 2024) models (1.5B and 7B) in Appendix E. We analyze **error types** that MetaAct-RL model identifies and handles in Appendix F. We also perform **qualitative case study** on MetaAct-RL and other methods in Appendix G.

7 Conclusion

In this paper, we explore training LMs for reasoning with diverse meta-actions. First, we frame LM reasoning as a decision-making process, and propose a data synthesizing method to construct a trace set for initialization. Building on this, we delve into the RL process, identify challenges, and propose length-based reward and regularization, and a key-state restart strategy for deeper exploration, richer action diversity, and sampling efficiency. We conduct extensive experiments and analyses to show the effectiveness of our method and its working mechanism. We hope our work provides insights for the community of RL for LM reasoning.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by the Sci-

ence and Technology Commission of Shanghai Municipality (No. 24511103100), National Natural Science Foundation of China (No.62576106, 62376061, 62476061), Shanghai Rising-Star Program (23QA1400200), and Natural Science Foundation of Shanghai (23ZR1403500).

References

AI, E.; :; Shah, D. J.; Rushton, P.; et al. 2025. Rethinking Reflection in Pre-Training. arXiv:2504.04022.

Amini, A.; Gabriel, S.; Lin, S.; Koncel-Kedziorski, R.; Choi, Y.; and Hajishirzi, H. 2019. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 2357–2367. Association for Computational Linguistics.

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete Problems in AI Safety. arXiv:1606.06565.

Ankner, Z.; Paul, M.; Cui, B.; Chang, J. D.; and Ammanabrolu, P. 2024. Critique-out-Loud Reward Models. *CoRR*, abs/2408.11791.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Brown, B. C. A.; Juravsky, J.; Ehrlich, R. S.; Clark, R.; Le, Q. V.; Ré, C.; and Mirhoseini, A. 2024. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. *CoRR*, abs/2407.21787.

Chen, L.; Zhu, C.; Chen, J.; Soselia, D.; Zhou, T.; Goldstein, T.; Huang, H.; Shoeybi, M.; and Catanzaro, B. 2024a. ODIN: Disentangled Reward Mitigates Hacking in RLHF. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Chen, W.; Yin, M.; Ku, M.; Lu, P.; Wan, Y.; Ma, X.; Xu, J.; Wang, X.; and Xia, T. 2023. TheoremQA: A Theorem-driven Question Answering Dataset. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 7889–7901. Association for Computational Linguistics.

Chen, X.; Xu, J.; Liang, T.; He, Z.; Pang, J.; Yu, D.; Song, L.; Liu, Q.; Zhou, M.; Zhang, Z.; Wang, R.; Tu, Z.; Mi, H.; and Yu, D. 2024b. Do NOT Think That Much for 2+3=? On the Overthinking of o1-Like LLMs. arXiv:2412.21187.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168.

deepseek. 2024. deepseekr1lite. *website*.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Ding, Y.; Xi, Z.; He, W.; Li, Z.; Zhai, Y.; Shi, X.; Cai, X.; Gui, T.; Zhang, Q.; and Huang, X. 2024. Mitigating Tail Narrowing in LLM Self-Improvement via Socratic-Guided Sampling. *arXiv preprint arXiv:2411.00750*.

Dou, Z.; Yang, C.; Wu, X.; Chang, K.; and Peng, N. 2024. ReReST: Reflection-Reinforced Self-Training for Language Agents. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, 15394–15411. Association for Computational Linguistics.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; et al. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.

Gandhi, K.; Chakravarthy, A.; Singh, A.; Lile, N.; and Goodman, N. D. 2025. Cognitive Behaviors that Enable Self-Improving Reasoners, or, Four Habits of Highly Effective STaRs. arXiv:2503.01307.

Gao, B.; Cai, Z.; Xu, R.; Wang, P.; Zheng, C.; Lin, R.; Lu, K.; Lin, J.; Zhou, C.; Xiao, W.; Hu, J.; Liu, T.; and Chang, B. 2024. LLM Critics Help Catch Bugs in Mathematics: Towards a Better Mathematical Verifier with Natural Language Feedback. *CoRR*, abs/2406.14024.

Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; and Chen, W. 2024. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Gui, L.; and Ren, Q. 2025. Training Reasoning Model with Dynamic Advantage Estimation on Reinforcement Learning. https://www.notion.so/ADORA-Enhancing-RL-with-Advantage-Dynamics-and-Online-Rollout-Adaptation-1a830cc0904681fa9df3e076b6557a3e. Notion Blog.

Guo, D.; Wu, F.; Zhu, F.; Leng, F.; Shi, G.; et al. 2025a. Seed1.5-VL Technical Report. arXiv:2505.07062.

Guo, X.; Xi, Z.; Ding, Y.; Zhai, Y.; Shi, X.; Cai, X.; Gui, T.; Zhang, Q.; and Huang, X. 2025b. Counteracting Matthew Effect in Self-Improvement of LVLMs through Head-Tail Re-balancing. arXiv:2510.26474.

He, W.; Liu, S.; Zhao, J.; Ding, Y.; Lu, Y.; Xi, Z.; Gui, T.; Zhang, Q.; and Huang, X. 2024. Self-Demos: Eliciting Out-of-Demonstration Generalizability in Large Language Models. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Findings of the Association for Computational Linguistics: NAACL 2024*, 3829–3845. Mexico City, Mexico: Association for Computational Linguistics.

Hegel, G. W. F. 1991. *The Encyclopaedia Logic, with the Zus tze: Part I of the Encyclopaedia of Philosophical Sciences with the Zusätze*, volume 1. Hackett Publishing.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In Vanschoren, J.; and Yeung, S., eds., *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Hu, Z.; Wang, Y.; Dong, H.; Xu, Y.; Saha, A.; Xiong, C.; Hooi, B.; and Li, J. 2025. Beyond 'Aha!': Toward Systematic Meta-Abilities Alignment in Large Reasoning Models. arXiv:2505.10554.

Huang, J.; Chen, X.; Mishra, S.; Zheng, H. S.; Yu, A. W.; Song, X.; and Zhou, D. 2024a. Large Language Models Cannot Self-Correct Reasoning Yet. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Huang, J.; Gu, S.; Hou, L.; Wu, Y.; Wang, X.; Yu, H.; and Han, J. 2023. Large Language Models Can Self-Improve. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 1051–1068. Association for Computational Linguistics.

Huang, Z.; Zou, H.; Li, X.; Liu, Y.; Zheng, Y.; Chern, E.; Xia, S.; Qin, Y.; Yuan, W.; and Liu, P. 2024b. O1 Replication Journey - Part 2: Surpassing O1-preview through Simple Distillation, Big Progress or Bitter Lesson? *CoRR*, abs/2411.16489.

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

Kahneman, D. 2011. Thinking, fast and slow. *Farrar, Straus and Giroux*.

Keskar, N. S.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. arXiv:1909.05858.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2023. Large Language Models are Zero-Shot Reasoners. arXiv:2205.11916.

Kulal, S.; Pasupat, P.; Chandra, K.; Lee, M.; Padon, O.; Aiken, A.; and Liang, P. 2019. SPoC: Search-based Pseudocode to Code. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 11883–11894.

Kumar, A.; Zhuang, V.; Agarwal, R.; Su, Y.; et al. 2024. Training Language Models to Self-Correct via Reinforcement Learning. *CoRR*, abs/2409.12917.

LI, J.; Beeching, E.; Tunstall, L.; et al. 2024. Numina-Math. [https://github.com/project-numina/aimo-progress-prize](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).

Ling, W.; Yogatama, D.; Dyer, C.; and Blunsom, P. 2017. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 158–167. Association for Computational Linguistics.

Luong, T. Q.; Zhang, X.; Jie, Z.; Sun, P.; Jin, X.; and Li, H. 2024. ReFT: Reasoning with Reinforced Fine-Tuning. arXiv:2401.08967.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Gupta, S.; Majumder, B. P.; Hermann, K.; Welleck, S.; Yazdanbakhsh, A.; and Clark, P. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

McAleese, N.; Pokorny, R. M.; Uribe, J. F. C.; Nitishinskaya, E.; Trebacz, M.; and Leike, J. 2024. LLM Critics Help Catch LLM Bugs. arXiv:2407.00215.

Min, Y.; Chen, Z.; Jiang, J.; Chen, J.; Deng, J.; Hu, Y.; Tang, Y.; Wang, J.; Cheng, X.; Song, H.; Zhao, W. X.; Liu, Z.; Wang, Z.; and Wen, J. 2024. Imitate, Explore, and Self-Improve: A Reproduction Report on Slow-thinking Reasoning Systems. *CoRR*, abs/2412.09413.

Muennighoff, N.; Yang, Z.; Shi, W.; Li, X. L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; and Hashimoto, T. 2025. s1: Simple test-time scaling. arXiv:2501.19393.

OpenAI; :; El-Kishky, A.; Wei, A.; Saraiva, A.; Minaiev, B.; Selsam, D.; Dohan, D.; Song, F.; Lightman, H.; Clavera, I.; Pachocki, J.; Tworek, J.; Kuhn, L.; Kaiser, L.; Chen, M.; Schwarzer, M.; Rohaninejad, M.; McAleese, N.; o3 contributors; Mürk, O.; Garg, R.; Shu, R.; Sidor, S.; Kosaraju, V.; and Zhou, W. 2025. Competitive Programming with Large Reasoning Models. arXiv:2502.06807.

OpenAI. 2024. Learning to Reason with LLMs.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; et al. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Pan, L.; Saxon, M.; Xu, W.; Nathani, D.; Wang, X.; and Wang, W. Y. 2023. Automatically Correcting Large Language Models: Surveying the landscape of diverse self-correction strategies. *CoRR*, abs/2308.03188.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tür, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 2080–2094. Association for Computational Linguistics.

Plappert, M.; Andrychowicz, M.; Ray, A.; McGrew, B.; Baker, B.; Powell, G.; Schneider, J.; Tobin, J.; Chociej, M.; Welinder, P.; et al. 2018. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*.

Popov, I.; Heess, N.; Lillicrap, T. P.; Hafner, R.; Barth-Maron, G.; Vecerík, M.; Lampe, T.; Tassa, Y.; Erez, T.; and Riedmiller, M. A. 2017. Data-efficient Deep Reinforcement Learning for Dexterous Manipulation. *CoRR*, abs/1704.03073.

Popper, K. 2005. *The logic of scientific discovery*. Routledge.

Puterman, M. L. 1990. Markov decision processes. *Handbooks in operations research and management science*, 2: 331–434.

Qu, Y.; Zhang, T.; Garg, N.; and Kumar, A. 2024. Recursive Introspection: Teaching Language Model Agents How to Self-Improve. *CoRR*, abs/2407.18219.

QwenTeam. 2024. Qwen2.5: A Party of Foundation Models.

Saunders, W.; Yeh, C.; Wu, J.; Bills, S.; Ouyang, L.; Ward, J.; and Leike, J. 2022. Self-critiquing models for assisting human evaluators. *CoRR*, abs/2206.05802.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347.

Shen, W.; Zheng, R.; Zhan, W.; Zhao, J.; Dou, S.; Gui, T.; Zhang, Q.; and Huang, X. 2023. Loose lips sink ships: Mitigating Length Bias in Reinforcement Learning from Human Feedback. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, 2859–2873. Association for Computational Linguistics.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: language agents with verbal reinforcement learning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Singh, A.; Co-Reyes, J. D.; Agarwal, R.; Anand, A.; et al. 2024. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. *Trans. Mach. Learn. Res.*, 2024.

Snell, C.; Lee, J.; Xu, K.; and Kumar, A. 2024. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *CoRR*, abs/2408.03314.

Subramanian, K.; Jr., C. L. I.; and Thomaz, A. L. 2016. Exploration from Demonstration for Interactive Reinforcement Learning. In

Jonker, C. M.; Marsella, S.; Thangarajah, J.; and Tuyls, K., eds., *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, 447–456. ACM.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Team, K.; Du, A.; Gao, B.; Xing, B.; et al. 2025. Kimi k1.5: Scaling Reinforcement Learning with LLMs. arXiv:2501.12599.

Team, N. 2025. Sky-T1: Train your own O1 preview model within $450. https://novasky-ai.github.io/posts/sky-t1. Accessed: 2025-01-09.

Wang, B.; Zheng, R.; Chen, L.; Liu, Y.; et al. 2024. Secrets of RLHF in Large Language Models Part II: Reward Modeling. *CoRR*, abs/2401.06080.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Welleck, S.; Lu, X.; West, P.; Brahman, F.; Shen, T.; Khashabi, D.; and Choi, Y. 2023. Generating Sequences by Learning to Self-Correct. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xi, Z.; Chen, W.; Hong, B.; Jin, S.; Zheng, R.; He, W.; Ding, Y.; Liu, S.; Guo, X.; Wang, J.; Guo, H.; Shen, W.; Fan, X.; Zhou, Y.; Dou, S.; Wang, X.; Zhang, X.; Sun, P.; Gui, T.; Zhang, Q.; and Huang, X. 2024a. Training Large Language Models for Reasoning through Reverse Curriculum Reinforcement Learning. arXiv:2402.05808.

Xi, Z.; Chen, W.; Hong, B.; Jin, S.; et al. 2024b. Training Large Language Models for Reasoning through Reverse Curriculum Reinforcement Learning. *CoRR*, abs/2402.05808.

Xi, Z.; Ding, Y.; Chen, W.; Hong, B.; Guo, H.; Wang, J.; Guo, X.; Yang, D.; Liao, C.; He, W.; Gao, S.; Chen, L.; Zheng, R.; Zou, Y.; Gui, T.; Zhang, Q.; Qiu, X.; Huang, X.; Wu, Z.; and Jiang, Y.-G. 2025a. AgentGym: Evaluating and Training Large Language Model-based Agents across Diverse Environments. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 27914–27961. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Xi, Z.; Guo, X.; Nan, Y.; Zhou, E.; Shen, J.; Chen, W.; Liu, J.; Huang, J.; Zhang, Z.; Guo, H.; Deng, X.; Lei, Z.; Zheng, M.; Wang, G.; Zhang, S.; Sun, P.; Zheng, R.; Yan, H.; Gui, T.; Zhang, Q.; and Huang, X. 2025b. BAPO: Stabilizing Off-Policy Reinforcement Learning for LLMs via Balanced Policy Optimization with Adaptive Clipping. arXiv:2510.18927.

Xi, Z.; Huang, J.; Guo, X.; Hong, B.; Yang, D.; Fan, X.; Li, S.; Chen, Z.; Ye, J.; Yuan, S.; Du, Z.; Yao, X.; Xu, Y.; Chen, J.; Zheng, R.; Gui, T.; Zhang, Q.; and Huang, X. 2025c. Critique-RL: Training Language Models for Critiquing through Two-Stage Reinforcement Learning. arXiv:2510.24320.

Xi, Z.; Huang, J.; Liao, C.; Huang, B.; Guo, H.; Liu, J.; Zheng, R.; Ye, J.; Zhang, J.; Chen, W.; He, W.; Ding, Y.; Li, G.; Chen, Z.; Du, Z.; Yao, X.; Xu, Y.; Chen, J.; Gui, T.; Wu, Z.; Zhang, Q.; Huang, X.; and Jiang, Y.-G. 2025d. AgentGym-RL: Training LLM Agents for Long-Horizon Decision Making through Multi-Turn Reinforcement Learning. arXiv:2509.08755.

Xi, Z.; Jin, S.; Zhou, Y.; Zheng, R.; Gao, S.; Liu, J.; Gui, T.; Zhang, Q.; and Huang, X. 2023. Self-Polish: Enhance Reasoning in Large Language Models via Problem Refinement. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 11383–11406. Singapore: Association for Computational Linguistics.

Xi, Z.; Yang, D.; Huang, J.; Tang, J.; Li, G.; Ding, Y.; He, W.; Hong, B.; Dou, S.; Zhan, W.; Wang, X.; Zheng, R.; Ji, T.; Shi, X.; Zhai, Y.; Weng, R.; Wang, J.; Cai, X.; Gui, T.; Wu, Z.; Zhang, Q.; Qiu, X.; Huang, X.; and Jiang, Y. 2024c. Enhancing LLM Reasoning via Critique Models with Test-Time and Training-Time Supervision. *CoRR*, abs/2411.16579.

Xu, W.; Zhu, G.; Zhao, X.; Pan, L.; Li, L.; and Wang, W. 2024. Pride and Prejudice: LLM Amplifies Self-Bias in Self-Refinement. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 15474–15492. Association for Computational Linguistics.

Yao, W.; Heinecke, S.; Niebles, J. C.; Liu, Z.; Feng, Y.; Xue, L.; N., R. R.; Chen, Z.; Zhang, J.; Arpit, D.; Xu, R.; Mui, P.; Wang, H.; Xiong, C.; and Savarese, S. 2024. Retroformer: Retrospective Large Language Agents with Policy Gradient Optimization. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yue, X.; Qu, X.; Zhang, G.; Fu, Y.; Huang, W.; Sun, H.; Su, Y.; and Chen, W. 2024. MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. D. 2022. STaR: Bootstrapping Reasoning With Reasoning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Zheng, R.; Dou, S.; Gao, S.; Hua, Y.; et al. 2023. Secrets of RLHF in Large Language Models Part I: PPO. *CoRR*, abs/2307.04964.

Zheng, X.; Lou, J.; Cao, B.; Wen, X.; Ji, Y.; Lin, H.; Lu, Y.; Han, X.; Zhang, D.; and Sun, L. 2024. Critic-CoT: Boosting the reasoning abilities of large language model via Chain-of-thoughts Critic. arXiv:2408.16326.

Zhou, R.; Du, S. S.; and Li, B. 2024. Reflect-RL: Two-Player Online RL Fine-Tuning for LMs. In Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 995–1015. Association for Computational Linguistics.