Easy Samples Are All You Need: Self-Evolving LLMs via Data-Efficient Reinforcement Learning

Zhiyin Yu [♡♠], **Bo Zhang** [♠†], **Qibin Hou** [♢], **Zhonghai Wu** [♡†], **Xiao Luo** [♣], **Lei Bai** [♠]
[♡]Peking University [♠]Shanghai Artificial Intelligence Laboratory
[♢]Nankai University [♣]University of Wisconsin–Madison
zhiyinyu25@stu.pku.edu.cn, zhangbo@pjlab.org.cn, houqb@nankai.edu.cn
wuzh@pku.edu.cn, xiao.luo@wisc.edu, bailei@pjlab.org.cn
Github Repository: https://github.com/YuZhiyin/EasyRL.

Abstract

Previous LLMs-based RL studies typically follow either supervised learning with high annotation costs, or unsupervised paradigms using voting or entropy-based rewards. However, their performance remains far from satisfactory due to the substantial annotation cost and issues such as model collapse or reward hacking. To address these issues, we introduce a new perspective inspired by cognitive learning theory and propose a novel approach called EasyRL. The core of EasyRL is to simulate the human cognitive acquisition curve by integrating reliable knowledge transfer from easy labeled data with a progressive divide-and-conquer strategy that tackles increasingly difficult unlabeled data. Specifically, we initialize a warm-up model using supervised RL with few-shot labeled data. This is followed by a divide-and-conquer pseudo-labeling strategy on difficult unlabeled data, combining consistency-based selection for low-uncertainty cases and reflection-based resolution for medium-uncertainty cases. Finally, difficulty-progressive self-training with iterative pseudo-labeling and RL further strengthens the model's reasoning capability. EasyRL provides a unified self-evolving framework that facilitates data-efficient post-training of LLMs. Experimental results on mathematical and scientific benchmarks demonstrate that EasyRL, using only 10% of easy labeled data, consistently outperforms state-of-the-art baselines.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across diverse domains such as mathematical reasoning (Cui et al., 2025; Luo et al., 2025b; Guan et al., 2025), scientific research (Team et al., 2025; Yu et al., 2025), and code generation (Guo et al., 2024; Qian et al., 2024). Recent advances including DeepSeek-R1 (DeepSeek-AI, 2025), OpenAI's o1 (OpenAI, 2024), and Kimi-1.5 (Team, 2025) have shown that Reinforcement Learning (RL), as a promising post-training paradigm, can effectively elicit and enhance the reasoning abilities of LLMs. Intriguingly, LLMs trained with RL exhibit emergent cognitive behaviors such as self-reflection (Gandhi et al., 2025), and spontaneous "aha moments," while also achieving strong generalization across diverse downstream tasks (Lambert et al., 2025).

In recent advances of LLM reinforcement learning for reasoning, existing approaches can be broadly categorized into supervised and unsupervised paradigms (Zhang et al., 2025e). Supervised methods rely on outcome-based rewards, derived either from verifiable answers or pre-trained reward models (Lyu et al., 2025; Ouyang et al., 2022). While effective, these methods heavily depend on human-labeled data or reward model supervision, resulting in substantial annotation cost (Zhang et al., 2025c) and limited scalability (Yuan et al., 2024; Liu et al., 2025). On the other hand, unsupervised methods attempt to construct reward signals through voting (Zuo et al., 2025) or entropy estimation (Zhang et al., 2025d). However, their performance gains are often marginal and susceptible to issues such as model collapse (Shumailov et al., 2024) or reward hacking (Shafayat et al., 2025), and they struggle to generalize across diverse models (Shao et al., 2025), making stable self-evolution difficult to achieve.

Inspired by Mind in Society (VYGOTSKY, 1978), cognitive development and skill acquisition are known to follow a from-easy-to-hard trajectory. Vygotsky's seminal Zone of Proximal Development (ZPD) theory posits learners master complex tasks by first internalizing knowledge from simple and achievable cases, and then gradually extending this knowledge to more difficult challenges with minimal external guidance. Subsequent studies fur-

[†] Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 15780–15795

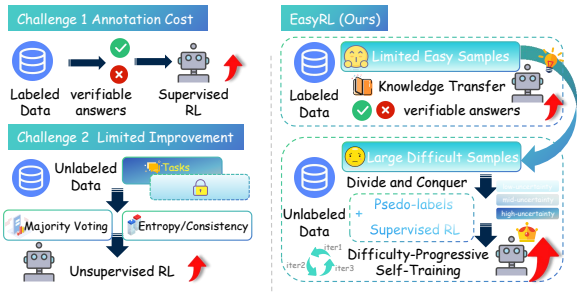

Figure 1: Comparison between (a) existing supervised and unsupervised RL approaches and (b) EasyRL.

ther validate this principle: humans require only a small set of easy, labeled examples to induce generalizable rules (Lake et al., 2016), and can transfer this foundational knowledge to tackle novel, harder problems through analogy and self-reflection (Nam and McClelland, 2024). This insight highlights a critical opportunity: if LLM training can be aligned with human cognitive patterns such that limited easy-labeled cases guide the model's progression toward increasingly difficult tasks, we may achieve a data-efficient RL paradigm that balances annotation cost and generalization. Therefore, we aim to address the following question:

> *Can LLMs gradually evolve with limited easy labeled data and abundant difficult unlabeled data?*

To address this question, we propose EasyRL, a novel reinforcement learning framework that enables LLMs to self-evolve from limited easy cases toward more difficult reasoning tasks. Different from existing supervised and unsupervised approaches, the core of EasyRL is to not only transfer reliable knowledge from easy labeled data, but also leverage difficult unlabeled data in a structured, progressively refined manner to enhance reasoning capability (see Figure 1). Specifically, we first transfer knowledge from labeled data to initialize a warm-up model using supervised reinforcement learning. Next, we adopt a Divide-and-Conquer strategy to construct high-quality pseudo-labeled datasets, where consistency-based selection handles low-uncertainty cases and reflection-based resolution addresses medium-uncertainty ones. To further promote self-improvement, we incorporate difficulty-progressive self-training: the model is iteratively trained on increasingly challenging pseudo-labeled samples, with selection criteria dynamically updated to follow an easy-to-hard progression aligned with human cognitive learning. Comprehensive experiments across multiple reasoning benchmarks demonstrate that EasyRL significantly outperforms competitive baselines, and enables a self-evolving learning trajectory in LLMs. The main contributions are summarized as follows:

❶ *New Perspective.* We introduce a cognitive-inspired framework that enables large language models to self-evolve from limited easy cases to more difficult reasoning tasks.

❷ *Novel Methodology.* EasyRL transfers knowledge from easy samples and progressively incorporates unlabeled data via divide-and-conquer, achieving difficulty-progressive self-training to enhance reasoning.

❸ *Extensive Experiments.* We conduct extensive experiments across multiple benchmarks, showing that EasyRL is **(1) data-efficient**, using only 10% easy labeled data while surpassing GRPO trained on the full dataset; **(2) self-evolving**, with steadily improving pseudo-label quality and an increasing focus on more difficult samples; and **(3) robust-performing**, outperforming supervised and unsupervised RL baselines and transferring effectively to out-of-domain tasks.

2 Related Work

2.1 Reinforcement Learning for Reasoning

Reinforcement learning with Verifiable Rewards (RLVR) has been widely utilized to strengthen the reasoning capabilities of LLMs (DeepSeek-AI, 2025; Yang et al., 2025a,b; Zhang et al., 2025e; Singh et al., 2025; Li et al., 2025c). Recently, an increasing body of research has focused on label-free RL, which eliminates reliance on human annotations (Shao et al., 2025; Zweiger et al., 2025; Xin et al., 2025). Zuo et al. (2025); Shafayat et al. (2025); Wei et al. (2025) adopt majority voting to generate pseudo-labels for correctness rewards. Alternatively, Zhang et al. (2025d) minimize predictive entropy within a latent semantic space, while other approaches (Zhao et al., 2025b; Prabhudesai et al., 2025; Li et al., 2025a; Agarwal et al., 2025) quantify model confidence via self-certainty or token-level entropy. Beyond evaluating only the final outputs, several studies incorporate intermediate reasoning states into reward modeling (Zhang et al., 2025c; Zhou et al., 2025). In parallel, research on self-evolving (Zhao et al., 2025a; Zhang et al., 2025b) and co-evolving (Wang et al., 2025a; Fang et al., 2025) RL paradigms further demonstrates the potential of autonomous adaptation. Despite their progress, self-evolving RL frameworks

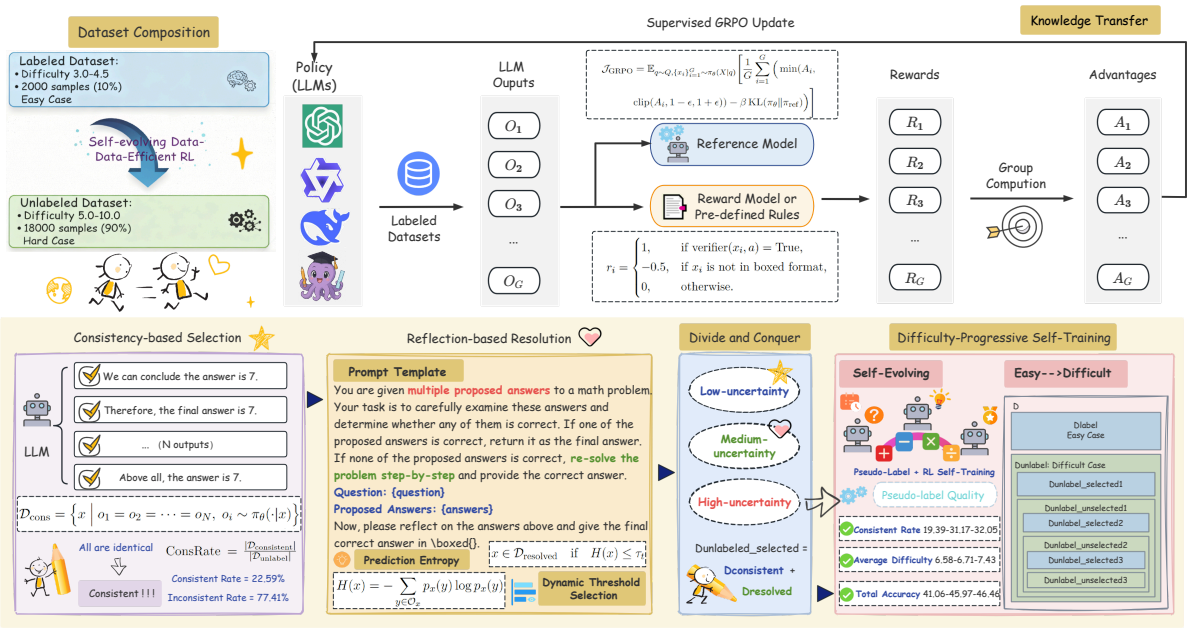

Figure 2: Overview of EasyRL. The workflow consists of three stages: (1) Knowledge Transfer, which initializes a stable warm-up model from easy labeled data using supervised RL; (2) Divide and Conquer, which generates high-quality pseudo-labeled data using consistency-based selection and reflection-based resolution; and (3) Difficulty-Progressive Self-Training, which refines the model by incorporating increasingly difficult pseudo-labeled samples in a staged RL training loop.

from a cognitive perspective remain underexplored, and our work aims to bridge this gap.

2.2 Data-efficient LLM Training

In recent years, achieving data-efficient post-training of LLMs with limited data has become a key research direction (Luo et al., 2025a). Existing approaches primarily focus on five major themes: data selection (Jeong et al., 2025; Liu et al., 2024), data quality enhancement (Li et al., 2024; Dai et al., 2025), synthetic data generation (Dong et al., 2024; Ming et al., 2024), data distillation and compression (Cui et al., 2024; Pan et al., 2024), and self-evolving data ecosystems (You et al., 2024; Madaan et al., 2023). Recent studies have also explored data selection in reinforcement learning (Li et al., 2025b) and even demonstrated that RL with as few as one or four samples can yield improvements (Wang et al., 2025b; Fatemi et al., 2025). In contrast, our work argues that by leveraging a small set of easy samples, the model can self-evolve to tackle increasingly difficult examples, aligning with the intuition of the human learning process.

3 Methodology

3.1 Problem Analysis

In real-world scenarios, obtaining annotations for difficult reasoning tasks is often expensive and time-consuming, while collecting answers for simple problems is much easier. Consider given a small labeled dataset $\mathcal{D}_{\text{label}}$ consisting of easy samples and a large unlabeled dataset $\mathcal{D}_{\text{unlabel}}$ containing difficult samples. In this work, we aim to explore whether a LLM can self-evolve from limited easy cases toward more difficult tasks through data-efficient RL. Inspired by (He et al., 2025), we adopt the difficulty definition provided by the AoPS[1] rating scale (1–10). Specifically, our EasyRL focuses on two objectives: (1) enhancing the model's reasoning capability and generalization performance on test sets, and (2) improving the quality of pseudo-labels generated on unlabeled difficult samples, thereby enabling a self-improving learning cycle without additional human supervision.

3.2 Framework Overview

As illustrated in Figure 2, our EasyRL workflow follows a multi-step process that enables the model to evolve from easy to increasingly difficult tasks. First, we transfer knowledge from the labeled dataset containing easy samples to obtain a stable warm-up model that serves as a reliable policy prior for subsequent pseudo-label generation. Then, we use this warm-up model to pseudo-label unlabeled data and partition samples by uncertainty

[1] https://artofproblemsolving.com/wiki/index.php/AoPS_Wiki:Competition_ratings

into low-, medium-, and high-uncertainty groups, where consistency-based selection handles low-uncertainty cases and reflection-based resolution refines medium-uncertainty ones. Finally, EasyRL iteratively combines labeled data with the selected pseudo-labeled samples in a reinforcement learning training loop, gradually exposing the model to increasingly difficult samples until convergence.

3.3 Knowledge Transfer for Foundational Capability

To provide a stable model π_{warm} for subsequent pseudo-label generation and self-evolution on unlabeled data $\mathcal{D}_{\text{unlabel}}$, we first transfer knowledge from the labeled data $\mathcal{D}_{\text{label}}$ through supervised reinforcement learning. Specifically, we employ the Group Relative Policy Optimization (GRPO) (Shao et al., 2024) algorithm, following DeepSeek-R1-Zero (DeepSeek-AI, 2025), to align the model's outputs with reference answers. The optimization objective is formulated as:

$$\mathcal{J}_{\text{GRPO}} = \mathbb{E}_{q\sim Q,\{x_i\}_{i=1}^{G}\sim\pi_\theta(X|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\Big(\min(A_i, \text{clip}(A_i, 1-\epsilon, 1+\epsilon)) - \beta\,\text{KL}(\pi_\theta\|\pi_{\text{ref}})\Big)\right], \tag{1}$$

where $\{x_1, \ldots, x_G\}$ are a group of outputs sampled from the policy model π_θ, and the KL term, controlled by the coefficient β, limits divergence from the reference policy π_{ref}. The normalized advantage A_i for each output x_i is computed as: $A_i = \frac{r_i - \text{mean}(\{r_1,\cdots,r_G\})}{\text{std}(\{r_1,\cdots,r_G\})}$. The reward r_i is computed from labeled pairs $(q, a) \in \mathcal{D}_{\text{label}}$, based on a correctness verifier that checks whether the model output matches the ground-truth response. Additionally, a format penalty is applied if the output is not in the expected boxed format:

$$r_i = \begin{cases} 1, & \text{if verifier}(x_i, a) = \text{True}, \\ -0.5, & \text{if } x_i \text{ is not in boxed format}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

This supervised RL training embeds domain-specific knowledge from the labeled data into the model. The resulting π_{warm} thus serves as a reliable model for pseudo-label generation and self-evolution on unlabeled data.

3.4 Divide-and-Conquer for Reliable Pseudo-Labeling

To leverage unlabeled data, we adopt a divide-and-conquer strategy for pseudo-labeling. Specifically, we partition samples based on uncertainty of pseudo-labels: (1) *Low-uncertainty*: samples with consistent outputs across multiple reasoning attempts form $D_{\text{consistent}}$ via consistency-based selection. (2) *Medium-uncertainty*: samples with inconsistent outputs are re-evaluated through a reflection mechanism and filtered to form D_{resolved} via reflection-based resolution. (3) *High-uncertainty*: highly uncertain samples are deferred to subsequent rounds for Difficulty-Progressive Self-Training (Section 3.5). The final pseudo-labeled set is $D_{\text{unlabel\_selected}} = D_{\text{consistent}} \cup D_{\text{resolved}}$.

Consistency-based Selection. For each unlabeled query $x \in \mathcal{D}_{\text{unlabel}}$, the model π_{warm} performs N independent inferences:

$$\mathcal{D}_{\text{cons}} = \Big\{x \;\Big|\; o_1 = o_2 = \cdots = o_N,\ o_i \sim \pi_\theta(\cdot|x)\Big\}, \tag{3}$$

where $o_1, \ldots, o_N$ denote the generated outputs. If all outputs are identical, the sample is regarded as consistent. The overall consistency rate is defined as: $\text{ConsRate} = \frac{|\mathcal{D}_{\text{consistent}}|}{|\mathcal{D}_{\text{unlabel}}|}$, which reflects the model's stability and confidence on unlabeled data.

Reflection-based Resolution. For samples that fail the consistency check, we employ a reflection mechanism to assess their reliability. Given a set of proposed answers $\mathcal{O}_x = \{o_1, \ldots, o_N\}$, we compute empirical probability $p_x(y)$ for each distinct answer y, and define the prediction entropy as:

$$H(x) = -\sum_{y\in\mathcal{O}_x} p_x(y)\log p_x(y). \tag{4}$$

A dynamic threshold τ_t is used to determine whether a sample can be confidently resolved:

$$x \in \mathcal{D}_{\text{resolved}} \quad \text{if} \quad H(x) \le \tau_t, \tag{5}$$

where τ_t is a dynamic threshold, set by default to 0.3. The pseudo-label is assigned as $\hat{y}_x = \text{Reflection}(\mathcal{O}_x)$, which is the answer obtained after the reflection-based resolution, while samples with $H(x) > \tau_t$ are reserved for future rounds.

In summary, this divide-and-conquer strategy progressively constructs the reliable pseudo-labeled dataset $\mathcal{D}_{\text{unlabel\_selected}}$ while avoiding noisy supervision from high-uncertainty samples. By iteratively combining Consistency-based Selection

and Reflection-based Resolution, our method ensures stable and high-quality pseudo-labels that effectively support the self-evolution process.

3.5 Difficulty-Progressive Self-Training for Evolution

To further enable self-evolution on $\mathcal{D}_{\text{unlabel\_selected}}$, we adopt a difficulty-progressive self-training strategy that exposes the model to increasingly difficult or uncertain examples.

Specifically, we first combine the pseudo-labeled set $\mathcal{D}^{(0)}_{\text{unlabel\_selected}}$ obtained via the divide-and-conquer strategy with the labeled dataset $\mathcal{D}_{\text{label}}$ to perform supervised RL training. The reward reflects the correctness of pseudo-labels: the model receives 1 if its output matches the pseudo-label, 0 otherwise, with an additional penalty for incorrect format, resulting in the first intermediate model π_1.

We then iteratively apply pseudo-labeling, selection (see Section 3.4), and self-training to progressively refine the model. In each iteration, the current model π_i generates pseudo-labels for the remaining unlabeled samples $\mathcal{D}^{(i)}_{\text{unlabel\_unselected}}$, which correspond to the high-uncertainty samples from the previous round in the divide-and-conquer process. These pseudo-labeled samples are then combined with labeled data for supervised reinforcement learning to obtain the next model π_{i+1}. This process continues until convergence, producing a sequence of models $\pi_1, \pi_2, \ldots, \pi_n$, each increasingly capable of handling more difficult or uncertain cases. Formally, at iteration i:

$$\pi_{i+1} = \text{RL}(\mathcal{D}_{\text{label}} \cup \mathcal{D}^{(i)}_{\text{unlabel\_selected}}), \quad (6)$$

where $\mathcal{D}^{(i)}_{\text{unlabel\_selected}}$ denotes the pseudo-labeled set selected by π_i in the current iteration. This framework enables gradual evolution from easy to difficult instances, leveraging both labeled and pseudo-labeled data.

3.6 Summarization

To summarize, the overall workflow of EasyRL is presented in Algorithm 1. First, in the Knowledge Transfer stage, a supervised RL model π_{warm} is trained on labeled data to provide a reliable policy prior for pseudo-label generation. Next, the Divide-and-Conquer stage leverages π_{warm} to generate multiple candidate outputs for each unlabeled query and organizes them by uncertainty via consistency-based selection and reflection-based resolution, resulting in a high-quality pseudo-labeled dataset $\mathcal{D}_{\text{unlabel\_selected}}$. From a cognitive perspective, $\mathcal{D}_{\text{unlabel\_selected}}$ can be viewed as an analogue of the Zone of Proximal Development (ZPD), which characterizes tasks that slightly exceed an agent's current capability yet remain learnable under appropriate guidance. Although ZPD is traditionally defined in human learning contexts involving more knowledgeable peers or instructors, we adapt this concept to the LLM RL setting by treating intrinsic feedback as implicit instructional signals. It allows the model to progressively expand its capability boundary, facilitating self-evolution through difficulty-progressive self-training.

Algorithm 1: EasyRL framework

Input : Labeled dataset $\mathcal{D}_{\text{label}}$, unlabeled dataset $\mathcal{D}_{\text{unlabel}}$, maximum iterations $I_{\max}$.

// Step 1: Knowledge Transfer;
Train a supervised RL model π_{warm} on $\mathcal{D}_{\text{label}}$ using Eq. 1 and 2;
// Step 2: Divide and Conquer;
foreach $x \in \mathcal{D}_{unlabel}$ **do**
  Generate N outputs $\{o_i\}_{i=1}^{N}$ using π_{warm} by Eq. 3; Consistency-based Selection to obtain $\mathcal{D}_{\text{consistent}}$;
  Reflection-based Resolution by Eq. 4 and Eq. 5 to obtain $\mathcal{D}_{\text{resolved}}$;
$\mathcal{D}_{\text{unlabel\_selected}} = \mathcal{D}_{\text{consistent}} \cup \mathcal{D}_{\text{resolved}}$;
// Step 3: Difficulty-Progressive Self-Training;
for $i \leftarrow 0$ **to** $I_{\max}$ **do**
  Train π_i with RL on $\mathcal{D}_{\text{label}} \cup \mathcal{D}^{(i)}_{\text{unlabel\_selected}}$ to obtain π_{i+1};
  Use π_{i+1} to generate new pseudo-labels for $\mathcal{D}^{(i)}_{\text{unlabel\_unselected}}$ via Step 2;
  Update $\mathcal{D}^{(i+1)}_{\text{unlabel\_selected}}$;

Output : Evolved model π_{final}.

4 Experiment

4.1 Experimental Setup

Datasets. For training, we construct a final dataset consisting of 20,000 samples from DeepMath-103K (He et al., 2025) with each problem assigned a difficulty level. We randomly select 2,000 samples with difficulty levels ranging from 3.0 to 4.5 as the labeled data, and 18,000 samples with difficulty levels between 5.0 and 10.0 as the unla-

Methods	Mathematical Reasoning						Scientific Reasoning			
	MATH	Minerva	Olympiad	AIME24	AMC23	Avg.	Biology	Chemistry	Physics	Avg.
Qwen2.5-Math-1.5B										
Vanilla Base	65.0	18.0	28.3	6.7	45.0	32.6	0.0	2.3	1.4	1.5
w/ Supervised GRPO	66.4$_{\uparrow 1.4}$	27.9$_{\uparrow 9.9}$	30.7$_{\uparrow 2.4}$	3.3$_{\downarrow 3.4}$	50.0$_{\uparrow 5.0}$	35.7$_{\uparrow 3.1}$	11.4$_{\uparrow 11.4}$	10.0$_{\uparrow 7.7}$	4.3$_{\uparrow 2.9}$	7.9$_{\uparrow 6.4}$
w/ Unsupervised EMPO	66.6$_{\uparrow 1.6}$	28.3$_{\uparrow 10.3}$	31.0$_{\uparrow 2.7}$	6.7$_{\uparrow 0.0}$	**60.0**$_{\uparrow 15.0}$	38.5$_{\uparrow 5.9}$	20.0$_{\uparrow 20.0}$	16.9$_{\uparrow 14.6}$	12.1$_{\uparrow 10.7}$	15.6$_{\uparrow 14.1}$
w/ EasyRL Iter1	69.8$_{\uparrow 4.8}$	28.7$_{\uparrow 10.7}$	31.0$_{\uparrow 2.7}$	3.3$_{\downarrow 3.4}$	52.5$_{\uparrow 7.5}$	37.1$_{\uparrow 4.5}$	27.1$_{\uparrow 27.1}$	14.6$_{\uparrow 12.3}$	10.7$_{\uparrow 9.3}$	15.6$_{\uparrow 14.1}$
w/ EasyRL Iter2	70.0$_{\uparrow 5.0}$	**31.6**$_{\uparrow 13.6}$	**31.3**$_{\uparrow 3.0}$	10.0$_{\uparrow 3.3}$	50.0$_{\uparrow 5.0}$	38.6$_{\uparrow 6.0}$	24.3$_{\uparrow 24.3}$	**20.8**$_{\uparrow 18.5}$	11.4$_{\uparrow 10.0}$	17.6$_{\uparrow 16.1}$
w/ EasyRL Iter3	**71.2**$_{\uparrow 6.2}$	30.9$_{\uparrow 12.9}$	**31.3**$_{\uparrow 3.0}$	**13.3**$_{\uparrow 6.6}$	55.0$_{\uparrow 10.0}$	**40.3**$_{\uparrow 7.7}$	**35.7**$_{\uparrow 35.7}$	17.7$_{\uparrow 15.4}$	**12.9**$_{\uparrow 11.5}$	**19.4**$_{\uparrow 17.9}$
Qwen2.5-Math-7B										
Vanilla Base	72.6	16.9	33.8	16.7	52.5	38.5	18.6	24.6	26.4	24.1
Vanilla Instruct	85.0$_{\uparrow 12.4}$	41.5$_{\uparrow 24.6}$	40.4$_{\uparrow 6.6}$	16.7$_{\uparrow 0.0}$	60.0$_{\uparrow 7.5}$	48.7$_{\uparrow 10.2}$	28.6$_{\uparrow 10.0}$	23.8$_{\downarrow 0.8}$	31.4$_{\uparrow 5.0}$	27.9$_{\uparrow 3.8}$
w/ Supervised GRPO	75.6$_{\uparrow 3.0}$	28.3$_{\uparrow 11.4}$	37.8$_{\uparrow 4.0}$	20.0$_{\uparrow 3.3}$	55.0$_{\uparrow 2.5}$	43.3$_{\uparrow 4.8}$	24.3$_{\uparrow 5.7}$	23.1$_{\downarrow 1.5}$	32.9$_{\uparrow 6.5}$	27.4$_{\uparrow 3.3}$
w/ Unsupervised EMPO	74.8$_{\uparrow 2.2}$	35.3$_{\uparrow 18.4}$	37.3$_{\uparrow 3.5}$	16.7$_{\uparrow 0.0}$	57.5$_{\uparrow 5.0}$	44.3$_{\uparrow 5.8}$	25.7$_{\uparrow 7.1}$	23.1$_{\downarrow 1.5}$	30.0$_{\uparrow 3.6}$	26.5$_{\uparrow 2.4}$
w/ EasyRL Iter1	**79.0**$_{\uparrow 6.4}$	38.6$_{\uparrow 21.7}$	**42.1**$_{\uparrow 8.3}$	16.7$_{\uparrow 0.0}$	50.0$_{\downarrow 2.5}$	45.3$_{\uparrow 6.8}$	28.6$_{\uparrow 10.0}$	23.1$_{\downarrow 1.5}$	27.9$_{\uparrow 1.5}$	26.2$_{\uparrow 2.1}$
w/ EasyRL Iter2	78.4$_{\uparrow 5.8}$	**43.8**$_{\uparrow 26.9}$	39.9$_{\uparrow 6.1}$	13.3$_{\downarrow 3.4}$	57.5$_{\uparrow 5.0}$	46.6$_{\uparrow 8.1}$	28.6$_{\uparrow 10.0}$	20.8$_{\downarrow 3.8}$	37.1$_{\uparrow 10.7}$	29.1$_{\uparrow 5.0}$
w/ EasyRL Iter3	**79.0**$_{\uparrow 6.4}$	43.4$_{\uparrow 26.5}$	41.2$_{\uparrow 7.4}$	**26.7**$_{\uparrow 10.0}$	**62.5**$_{\uparrow 10.0}$	**50.6**$_{\uparrow 12.1}$	**31.4**$_{\uparrow 12.8}$	**24.6**$_{\uparrow 0.0}$	**35.7**$_{\uparrow 9.3}$	**30.6**$_{\uparrow 6.5}$
Llama-3.2-3B-Instruct										
Vanilla Base	47.4	21.0	13.0	3.3	22.5	21.4	27.1	17.7	15.0	18.5
w/ Supervised GRPO	47.2$_{\downarrow 0.2}$	18.8$_{\downarrow 2.2}$	14.4$_{\uparrow 1.4}$	10.0$_{\uparrow 6.7}$	**27.5**$_{\uparrow 5.0}$	23.6$_{\uparrow 2.2}$	34.3$_{\uparrow 7.2}$	16.2$_{\downarrow 1.5}$	19.3$_{\uparrow 4.3}$	23.3$_{\uparrow 4.8}$
w/ Unsupervised EMPO	47.6$_{\uparrow 0.2}$	20.6$_{\downarrow 0.4}$	13.5$_{\uparrow 0.5}$	6.7$_{\uparrow 3.4}$	**27.5**$_{\uparrow 5.0}$	23.2$_{\uparrow 1.8}$	30.0$_{\uparrow 2.9}$	17.7$_{\uparrow 0.0}$	13.6$_{\downarrow 1.4}$	18.5$_{\uparrow 0.0}$
w/ EasyRL Iter1	47.2$_{\downarrow 0.2}$	**22.1**$_{\uparrow 1.1}$	15.1$_{\uparrow 2.1}$	10.0$_{\uparrow 6.7}$	**27.5**$_{\uparrow 5.0}$	24.4$_{\uparrow 3.0}$	32.9$_{\uparrow 5.8}$	16.9$_{\downarrow 0.8}$	**23.6**$_{\uparrow 8.6}$	24.5$_{\uparrow 6.0}$
w/ EasyRL Iter2	**49.0**$_{\uparrow 1.6}$	**22.1**$_{\uparrow 1.1}$	**16.0**$_{\uparrow 3.0}$	10.0$_{\uparrow 6.7}$	22.5$_{\uparrow 0.0}$	23.9$_{\uparrow 2.5}$	34.3$_{\uparrow 7.2}$	21.5$_{\uparrow 3.8}$	19.3$_{\uparrow 4.3}$	25.0$_{\uparrow 6.5}$
w/ EasyRL Iter3	47.4$_{\uparrow 0.0}$	21.7$_{\uparrow 0.7}$	14.8$_{\uparrow 1.8}$	**13.3**$_{\uparrow 10.0}$	**27.5**$_{\uparrow 5.0}$	**24.9**$_{\uparrow 3.5}$	**35.7**$_{\uparrow 8.6}$	**23.8**$_{\uparrow 6.1}$	22.1$_{\uparrow 7.1}$	**27.2**$_{\uparrow 8.7}$

Table 1: Performance comparison of different reinforcement learning strategies on LLMs for mathematical and scientific reasoning tasks. The **boldfaced** scores represent the **best** results. The arrows denote the performance change of each method relative to Vanilla Base.

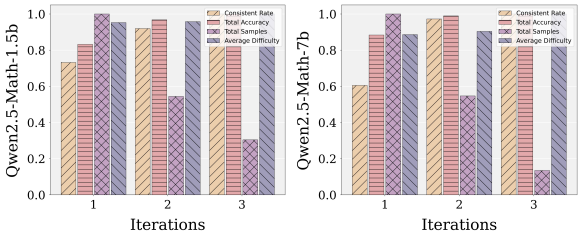

Figure 3: Quality evolution of pseudo-labeled data across three rounds, measured by consistency rate, total accuracy, samples and average difficulty.

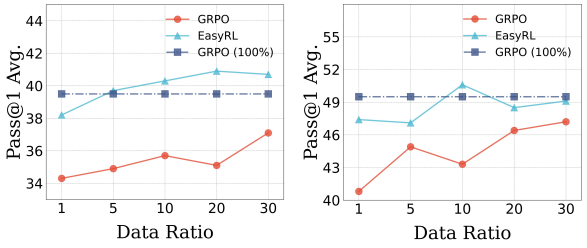

Figure 4: Performance comparison under different proportions of labeled data on Qwen2.5-Math-1.5B (left) and Qwen2.5-Math-7B (right).

beled data to demonstrate the self-evolving process. The selected samples across different difficulty levels are evenly distributed as much as possible. For evaluation, we adopt several well-known mathematical and scientific reasoning benchmarks, including MATH (Hendrycks et al., 2021), Minerva MATH (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME24[2], AMC23[3] and GPQA (Rein et al., 2023).

Baseline. We compare our proposed EasyRL to three types of baselines as follows:

- **Vanilla**: We utilize Qwen2.5-Math-1.5B and Qwen2.5-Math-7B (Yang et al., 2024) to examine the scalability of EasyRL. To assess its generality across architectures, we also include Llama-3.2-3B-Instruct (Grattafiori et al., 2024).
- **Supervised:** We implement supervised GRPO using verl (Sheng et al., 2025) on labeled data. For every prompt, we sample 8 responses. The reward is 1 for a correct answer, 0 for an incorrect one, and -0.5 if no extractable answer is found.
- **Unsupervised:** We include EMPO (Zhang et al., 2025d) as a representative label-free RL baseline. The reward is based on semantic entropy.

Implementation Details. For training, we implement GRPO via (Sheng et al., 2025) and run experiments on 8×140 GB H200 GPUs. We perform two inferences per sample for consistency-based selection. For evaluation, we use zero-shot prompting with greedy decoding and report pass@1 accuracy.

[2] https://huggingface.co/datasets/HuggingFaceH4/aime_2024

[3] https://huggingface.co/datasets/AI-MO/aimo-validation-amc

Model	Variant	MATH	Minerva	Olympiad Bench	AIME24	AMC23	GPQA	Avg.
Qwen2.5-Math-1.5B	EasyRL w/o KT	$68.2_{\downarrow 3.0}$	$29.0_{\downarrow 1.9}$	$31.9_{\uparrow 0.6}$	$10.0_{\downarrow 3.3}$	$52.5_{\downarrow 2.5}$	$17.9_{\downarrow 1.5}$	$34.9_{\downarrow 2.0}$
	EasyRL w/o DC	$67.6_{\downarrow 3.6}$	$25.4_{\downarrow 5.5}$	$30.2_{\downarrow 1.1}$	$3.3_{\downarrow 10.0}$	$45.0_{\downarrow 10.0}$	$18.2_{\downarrow 1.2}$	$31.6_{\downarrow 5.3}$
	EasyRL w/o DST	$69.8_{\downarrow 1.4}$	$30.1_{\downarrow 0.8}$	**32.4**$_{\downarrow 1.1}$	$3.3_{\downarrow 10.0}$	**55.0**$_{\downarrow 0.0}$	$17.6_{\downarrow 1.8}$	$34.7_{\downarrow 2.2}$
	EasyRL	**71.2**	**30.9**	31.3	**13.3**	**55.0**	**19.4**	**36.9**
Qwen2.5-Math-7B	EasyRL w/o KT	$76.2_{\downarrow 2.8}$	$39.3_{\downarrow 4.1}$	$40.9_{\downarrow 0.3}$	$20.0_{\downarrow 6.7}$	$57.5_{\downarrow 5.0}$	$28.5_{\downarrow 2.1}$	$43.7_{\downarrow 3.5}$
	EasyRL w/o DC	$76.4_{\downarrow 2.6}$	$39.0_{\downarrow 4.4}$	$38.1_{\downarrow 3.1}$	$23.3_{\downarrow 3.4}$	$57.5_{\downarrow 5.0}$	$25.3_{\downarrow 5.3}$	$43.3_{\downarrow 3.9}$
	EasyRL w/o DST	$78.4_{\downarrow 0.6}$	**44.1**$_{\uparrow 0.7}$	$37.8_{\downarrow 3.4}$	**26.7**$_{\downarrow 0.0}$	$57.5_{\downarrow 5.0}$	$29.1_{\downarrow 1.5}$	$45.6_{\downarrow 1.6}$
	EasyRL	**79.0**	43.4	**41.2**	**26.7**	**62.5**	**30.6**	**47.2**

Table 2: Ablation study via performance comparison of different variants on EasyRL. The **boldfaced** scores represent the **best** results. The arrows denote the performance change of each variant relative to EasyRL.

4.2 Main Results

Performance on Mathematical Reasoning Tasks. We report the main results of EasyRL in Table 1 and summarize several observations. Firstly, EasyRL consistently outperforms both supervised and unsupervised RL baselines across all backbone models. Notably, with only 10% of easy labeled data, it achieves the largest improvements of 7.7% on Qwen2.5-Math-1.5B, 12.1% on Qwen2.5-Math-7B, and 3.5% on LLaMA3.2-3B-Instruct. Secondly, the iterative self-evolution process leads to a steady improvement in model performance, e.g., Qwen2.5-Math-7B improves by 5.3% from Iter1 to Iter3. Thirdly, EasyRL demonstrates strong transferability, effectively enhancing the reasoning capability of language models across different architectures and model scales.

Performance on Scientific Reasoning Tasks. Previous studies have shown that training language models on reasoning-intensive domains such as mathematics can enhance their general reasoning capabilities across other fields (Huan et al., 2025). To evaluate the out-of-domain (OOD) generalization of our method, we train EasyRL on math datasets and assess its transfer performance on biology, chemistry, and physics benchmarks. As shown in Table 1, EasyRL consistently improves over the base model, achieving an average gain of 17.9% on Qwen2.5-Math-1.5B, 6.5% on Qwen2.5-Math-7B, and 8.7% on LLaMA3.2-3B-Instruct. These results demonstrate that EasyRL has strong generalization and exhibits OOD reasoning capability.

Finding 1: EasyRL achieves robust performance gains: it outperforms supervised and unsupervised RL baselines, improves steadily through iterative self-evolution, and transfers effectively to out-of-domain scientific tasks.

4.3 Further Analysis

Pseudo-label Quality Analysis. To assess the quality of pseudo-labels across three iterations, we adopt four metrics: Consistent Rate, the proportion of samples that yield identical outputs across multiple reasoning attempts; Total Accuracy, the fraction of correctly pseudo-labeled samples; Total Samples, the number of selected pseudo-labeled samples in each round; and Average Difficulty, the mean difficulty of samples in $\mathcal{D}^{(i)}_{\text{unlabel\_selected}}$. Figure 3 shows the normalized trends of Consistent Rate and Total Accuracy over $\mathcal{D}_{\text{unlabel}}$, as well as Total Samples and Average Difficulty in $\mathcal{D}^{(i)}_{\text{unlabel\_selected}}$, for Qwen2.5-Math-1.5B and 7B.

From these results, we draw three observations: ❶ Across iterations, both Consistent Rate and Total Accuracy on $\mathcal{D}_{\text{unlabel}}$ steadily increase, indicating that models improve through self-evolution. Consistent Rate and Total Accuracy are positively correlated, which aligns with prior studies (Englesson and Azizpour, 2021; Sohn et al., 2020); ❷ Total Samples decreases over iterations, as each round operates on the remaining subset from the previous iteration; and ❸ Average Difficulty increases, with the overall average difficulty on $\mathcal{D}_{\text{unlabel}}$ being 6.9274, while in iteration 3 it reaches 7.0282 for the 1.5B model and 7.4256 for the 7B model. This pattern suggests that the model's learning process gradually shifts from easier to more difficult cases.

Finding 2: EasyRL exhibits a self-evolving pseudo-labeling process, with steadily improving pseudo-label quality and increasing focus on more difficult samples.

Data Proportion Analysis. We evaluate the performance of supervised GRPO and EasyRL using 1%, 5%, 10%, 20%, and 30% of $\mathcal{D}_{\text{label}}/\mathcal{D}$, as well as supervised GRPO trained on 100% labeled data, on

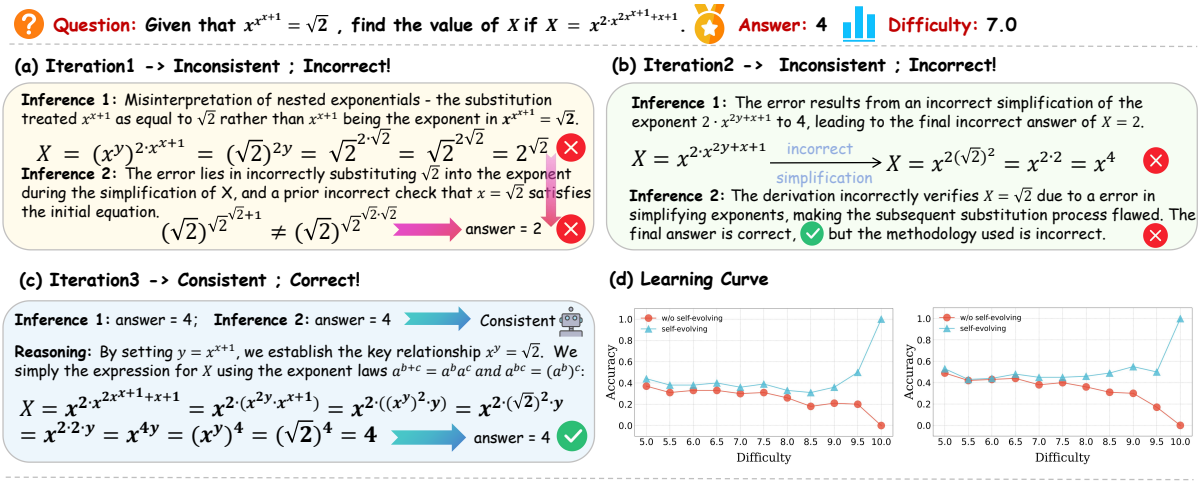

Figure 5: Case Study. Comparison of the pseudo-labeling process, generated by: (a) EasyRL Iteration 1, (b) EasyRL Iteration 2, and (c) EasyRL Iteration 3. Panel (d) shows the learning curve.

Qwen2.5-Math-1.5B and Qwen2.5-Math-7B (see Figure 4). Performance is measured as average accuracy across five math reasoning datasets. We draw two conclusions. First, EasyRL consistently outperforms supervised GRPO when trained with the same amount of labeled data. Second, using only 10% of easy labeled data, EasyRL already surpasses the performance of supervised GRPO trained on 100% labeled data, achieving 40.3 compared to 39.5 on Qwen2.5-Math-1.5B and 50.6 compared to 49.5 on Qwen2.5-Math-7B. Additional results are provided in Appendix B.

Finding 3: EasyRL demonstrates strong data efficiency: with only 10% easy labeled data, it surpasses supervised GRPO trained on 100% labeled data and consistently outperforms GRPO under equal labeled data budgets.

Ablation Study. To validate the contribution of each core component, we conduct ablation studies on Qwen2.5-Math-1.5B and Qwen2.5-Math-7B, as shown in Table 2. We compare three variants: ❶ EasyRL w/o Knowledge Transfer (KT): a variant that removes the supervised GRPO warm-up on labeled data; ❷ EasyRL w/o Divide and Conquer (DC): a variant that applies unsupervised RL training directly after π_{warm}; and ❸ EasyRL w/o Difficulty-Progressive Self-Training (DST): a variant that performs only one round of pseudo-labeling and selection after π_{warm}, jointly training on $\mathcal{D}_{\text{label}} \cup \mathcal{D}^{(1)}_{\text{unlabel\_selected}}$ to obtain the final model π_{final}. To ensure fair comparison, we select the top 70% pseudo-labeled samples with the lowest dynamic entropy for training. The results show that removing any of the Knowledge Transfer, Divide and Conquer, or Difficulty-Progressive Self-Training components leads to an average performance drop of 2.8%, 4.60%, and 1.9% on two models, demonstrating that each component plays an indispensable role in the overall performance of EasyRL.

Case Study. We present a case study on a problem from $\mathcal{D}_{\text{unlabel}}$ (see Figure 5). Across iterations, the model shows a refinement process: Iter 1 yields an inconsistent and incorrect solution; in Iter 2, one inferred answer is correct but the reasoning process remains flawed; and Iter 3 finally produces a consistent and correct answer. We further report learning curves for Qwen2.5-Math-1.5B (left) and Qwen2.5-Math-7B (right), showing accuracy on $\mathcal{D}_{\text{unlabel}}$ after pseudo-labeling and selection across difficulty levels. While the non–self-evolving baseline exhibits a steep accuracy drop as difficulty increases, the self-evolving model maintains more stable performance and achieves gains in the 9.0–10.0 difficulty range, highlighting the effectiveness of EasyRL under challenging reasoning conditions.

5 Conclusion

In this work, we present EasyRL, a data-efficient reinforcement learning framework that enables LLMs to self-evolve from limited easy labeled samples. EasyRL first initializes a reliable policy via supervised reinforcement learning, then constructs high-quality pseudo-labeled data through a divide-and-conquer strategy combining consistency and reflection. Finally, EasyRL progressively refines the model by incorporating difficult samples in a staged training loop. Experiments show that EasyRL outperforms strong baselines and significantly improves pseudo-label quality, enabling self-evolving reasoning in LLMs.

6 Limitations

Despite its promising performance, EasyRL still faces several limitations. Our current experiments mainly focus on domains with verifiable rewards. Extending EasyRL to open-ended tasks with non-verifiable rewards, including creative writing and scientific research, remains a challenging but valuable future direction. Moreover, future work may apply EasyRL to real-world scenarios such as embodied intelligence (Zhao et al., 2025a), social simulation, and safety-critical settings (Li et al., 2026).

7 Ethical Statement

This work adheres to ethical research and deployment principles. All experiments are conducted on publicly available datasets, and no private, sensitive, or personally identifiable data are used. Our EasyRL aims to improve model reasoning ability in a transparent and reproducible manner.

Nevertheless, as with all self-evolving AI systems, care should be taken to prevent potential unintended reinforcement of biased behaviors. We encourage future research to integrate fairness-aware reward functions and continuous human oversight to ensure that self-evolving LLMs remain aligned with human values and societal norms.

8 Acknowledgments

The work of Zhiyin Yu, Bo Zhang, and Lei Bai was supported by the Shanghai Artificial Intelligence Laboratory. The authors thank the anonymous reviewers for their valuable comments and suggestions.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. The unreasonable effectiveness of entropy minimization in llm reasoning. *Preprint*, arXiv:2505.15134.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. 2025. Process reinforcement through implicit rewards. *Preprint*, arXiv:2502.01456.

Yu Cui, Feng Liu, Pengbo Wang, Bohao Wang, Heng Tang, Yi Wan, Jun Wang, and Jiawei Chen. 2024. Distillation matters: Empowering sequential recommenders to match the performance of large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 507–517, New York, NY, USA. Association for Computing Machinery.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2025. Auggpt: Leveraging chatgpt for text data augmentation. *IEEE Transactions on Big Data*, 11(3):907–918.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. 2025. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *Preprint*, arXiv:2503.07065.

Qingxiu Dong, Li Dong, Xingxing Zhang, Zhifang Sui, and Furu Wei. 2024. Self-boosting large language models with synthetic preference data. *Preprint*, arXiv:2410.06961.

Erik Englesson and Hossein Azizpour. 2021. Consistency regularization can improve robustness to label noise. *Preprint*, arXiv:2110.01242.

Wenkai Fang, Shunyu Liu, Yang Zhou, Kongcheng Zhang, Tongya Zheng, Kaixuan Chen, Mingli Song, and Dacheng Tao. 2025. Serl: Self-play reinforcement learning for large language models with limited data. *Preprint*, arXiv:2505.20347.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. 2025. Concise reasoning via reinforcement learning. *Preprint*, arXiv:2504.05185.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *Preprint*, arXiv:2503.01307.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and et al Ahmad Al-Dahle. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small LLMs can master math reasoning with self-evolved deep thinking. In *Forty-second International Conference on Machine Learning*.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. Deepseek-coder: When the large language model meets programming – the rise of code intelligence. *Preprint*, arXiv:2401.14196.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *Preprint*, arXiv:2504.11456.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. 2025. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *Preprint*, arXiv:2507.00432.

Daniel P Jeong, Zachary Chase Lipton, and Pradeep Kumar Ravikumar. 2025. LLM-select: Feature selection with large language models. *Transactions on Machine Learning Research*.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2016. Building machines that learn and think like people. *Preprint*, arXiv:1604.00289.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. 2025. Tulu 3: Pushing frontiers in open language model post-training. *Preprint*, arXiv:2411.15124.

Bruce W Lee, Hyunsoo Cho, and Kang Min Yoo. 2024. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1281–1309, Mexico City, Mexico. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857. Curran Associates, Inc.

Pengyi Li, Matvey Skripkin, Alexander Zubrey, Andrey Kuznetsov, and Ivan Oseledets. 2025a. Confidence is all you need: Few-shot rl fine-tuning of language models. *Preprint*, arXiv:2506.06395.

Xinjin Li, Yu Ma, Yangchen Huang, Xingqi Wang, Yuzhen Lin, and Chenxi Zhang. 2024. Synergized data efficiency and compression (sec) optimization for large language models. In *2024 4th International Conference on Electronic Information Engineering and Computer Science (EIECS)*, pages 586–591.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025b. Limr: Less is more for rl scaling. *Preprint*, arXiv:2502.11886.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025c. Torl: Scaling tool-integrated rl. *Preprint*, arXiv:2503.23383.

Zhiheng Li, Zongyang Ma, Yuntong Pan, Ziqi Zhang, Xiaolei Lv, Bo Li, Jun Gao, Jianing Zhang, Chunfeng Yuan, Bing Li, and Weiming Hu. 2026. Making mllms blind: Adversarial smuggling attacks in mllm content moderation. *Preprint*, arXiv:2604.06950.

Shunyu Liu, Wenkai Fang, Zetian Hu, Junjie Zhang, Yang Zhou, Kongcheng Zhang, Rongcheng Tu, Ting-En Lin, Fei Huang, Mingli Song, Yongbin Li, and Dacheng Tao. 2025. A survey of direct preference optimization. *Preprint*, arXiv:2503.11701.

Zichen Liu, Changyu Chen, Chao Du, Wee Sun Lee, and Min Lin. 2024. Sample-efficient alignment for LLMs. In *Language Gamification - NeurIPS 2024 Workshop*.

Junyu Luo, Bohan Wu, Xiao Luo, Zhiping Xiao, Yiqiao Jin, Rong-Cheng Tu, Nan Yin, Yifan Wang, Jingyang Yuan, Wei Ju, and Ming Zhang. 2025a. A survey on efficient large language model training: From data-centric perspectives. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30904–30920, Vienna, Austria. Association for Computational Linguistics.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025b. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2. Notion Blog.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, Weihan Cao, Jiangning Liu, Hongwei Liu, Junnan Liu, Songyang Zhang, Dahua Lin, and Kai Chen. 2025. Exploring the limit

of outcome reward for learning mathematical reasoning. *Preprint*, arXiv:2502.06781.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.

Xuran Ming, Shoubin Li, Mingyang Li, Lvlong He, and Qing Wang. 2024. Autolabel: Automated textual data annotation method based on active learning and large language model. In *Knowledge Science, Engineering and Management: 17th International Conference, KSEM 2024, Birmingham, UK, August 16–18, 2024, Proceedings, Part IV*, page 400–411, Berlin, Heidelberg. Springer-Verlag.

Andrew J. Nam and James L. McClelland. 2024. Systematic human learning and generalization from a brief tutorial with explanatory feedback. *Open Mind*, 8:148–176.

OpenAI. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. 2025. Maximizing confidence alone improves reasoning. *Preprint*, arXiv:2505.22660.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *Preprint*, arXiv:2311.12022.

Sheikh Shafayat, Fahim Tajwar, Ruslan Salakhutdinov, Jeff Schneider, and Andrea Zanette. 2025. Can large reasoning models self-train? *Preprint*, arXiv:2505.21444.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. 2025. Spurious rewards: Rethinking training signals in rlvr. *Preprint*, arXiv:2506.10947.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297. ACM.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. 2025. Agentic reasoning and tool integration for llms via reinforcement learning. *Preprint*, arXiv:2505.01441.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. Fixmatch: simplifying semi-supervised learning with consistency and confidence. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

InternAgent Team, Bo Zhang, Shiyang Feng, Xiangchao Yan, Jiakang Yuan, Runmin Ma, Yusong Hu, Zhiyin Yu, Xiaohan He, Songtao Huang, Shaowei Hou, Zheng Nie, Zhilong Wang, Jinyao Liu, Tianshuo Peng, Peng Ye, Dongzhan Zhou, Shufei Zhang, Xiaosong Wang, Yilan Zhang, Meng Li, Zhongying Tu,

Xiangyu Yue, Wangli Ouyang, Bowen Zhou, and Lei Bai. 2025. Internagent: When agent becomes the scientist – building closed-loop system from hypothesis to verification. *Preprint*, arXiv:2505.16938.

Kimi Team. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

L. S. VYGOTSKY. 1978. *Mind in Society: Development of Higher Psychological Processes*. Harvard University Press.

Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. 2025a. Co-evolving llm coder and unit tester via reinforcement learning. *Preprint*, arXiv:2506.03136.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. 2025b. Reinforcement learning for reasoning in large language models with one training example. *Preprint*, arXiv:2504.20571.

Zhenting Wang, Guofeng Cui, Yu-Jhe Li, Kun Wan, and Wentian Zhao. 2025c. Dump: Automated distribution-level curriculum learning for rl-based llm post-training. *Preprint*, arXiv:2504.09710.

Lai Wei, Yuting Li, Chen Wang, Yue Wang, Linghe Kong, Weiran Huang, and Lichao Sun. 2025. Unsupervised post-training for multi-modal llm reasoning via grpo. *Preprint*, arXiv:2505.22453.

Rihui Xin, Han Liu, Zecheng Wang, Yupeng Zhang, Dianbo Sui, Xiaolin Hu, and Bingning Wang. 2025. Surrogate signals from format and length: Reinforcement learning for solving mathematical problems without ground truth answers. *Preprint*, arXiv:2505.19439.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Chenyu Yang, Shiqian Su, Shi Liu, Xuan Dong, Yue Yu, Weijie Su, Xuehui Wang, Zhaoyang Liu, Jinguo Zhu, Hao Li, Wenhai Wang, Yu Qiao, Xizhou Zhu, and Jifeng Dai. 2025a. Zerogui: Automating online gui learning at zero human cost. *Preprint*, arXiv:2505.23762.

Wenjie Yang, Mao Zheng, Mingyang Song, Zheng Li, and Sitong Wang. 2025b. Ssr-zero: Simple self-rewarding reinforcement learning for machine translation. *Preprint*, arXiv:2505.16637.

Jiaxuan You, Mingjie Liu, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. LLM-evolve: Evaluation for LLM's evolving capability on benchmarks. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16937–16942, Miami, Florida, USA. Association for Computational Linguistics.

Zhiyin Yu, Chao Zheng, Chong Chen, Xian-Sheng Hua, and Xiao Luo. 2025. scRAG: Hybrid retrieval-augmented generation for LLM-based cross-tissue single-cell annotation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 954–970, Vienna, Austria. Association for Computational Linguistics.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 2025. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. `https://hkust-nlp.notion.site/simplerl-reason`. Notion Blog.

Enci Zhang, Xingang Yan, Wei Lin, Tianxiang. Zhang, and Lu Qianchun. 2025a. Learning like humans: Advancing LLM reasoning capabilities via adaptive difficulty curriculum learning and expert-guided self-reformulation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6619–6633, Suzhou, China. Association for Computational Linguistics.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. 2025b. Darwin godel machine: Open-ended evolution of self-improving agents. *Preprint*, arXiv:2505.22954.

Kongcheng Zhang, Qi Yao, Shunyu Liu, Yingjie Wang, Baisheng Lai, Jieping Ye, Mingli Song, and Dacheng Tao. 2025c. Consistent paths lead to truth: Self-rewarding reinforcement learning for llm reasoning. *Preprint*, arXiv:2506.08745.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. 2025d. Right question is already half the answer: Fully unsupervisedd LLM reasoning incentization. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*.

Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. 2025e. No free lunch: Rethinking internal feedback for llm reasoning. *Preprint*, arXiv:2506.17219.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025a. Absolute zero: Reinforced self-play reasoning with zero data. *Preprint*, arXiv:2505.03335.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. 2025b. Learning to reason without external rewards. *Preprint*, arXiv:2505.19590.

Yujun Zhou, Zhenwen Liang, Haolin Liu, Wenhao Yu, Kishan Panaganti, Linfeng Song, Dian Yu, Xiangliang Zhang, Haitao Mi, and Dong Yu. 2025. Evolving language models without labels: Majority drives selection, novelty promotes variation. *Preprint*, arXiv:2509.15194.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. 2025. Ttrl: Test-time reinforcement learning. *Preprint*, arXiv:2504.16084.

Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. 2025. Self-adapting language models. *Preprint*, arXiv:2506.10943.

A Experimental Settings

A.1 Parameter Settings

Following previous work, we build on the SimpleRL codebase (Zeng et al., 2025) and adopt the standard evaluation prompt (Zhang et al., 2025d): "Let's think step by step and output the final answer within \boxed{}." During inference, we use a greedy decoding configuration with temperature set to 0, top-p set to 1, and a fixed random seed of 0. All experiments are conducted with the vLLM framework on 2 x NVIDIA H200 GPUs (140GB).

For RL training, we adopt the GRPO algorithm with a maximum sequence length of 4096. The training batch size is set to 8, with a mini-batch size of 2 and a micro-batch size of 1. The actor and critic are optimized using learning rates of 5×10^{-7} and 9×10^{-6}, respectively. All experiments are conducted on eight NVIDIA H200 GPUs (140GB each), and the model is trained for a single epoch.

A.2 Instruction Settings

Mathematical Reasoning Training and Evaluation Template

<|im_start|> system
Please reason step by step, and output your final answer within \boxed{}.
<|im_end|>
<|im_start|>user
{Question} Let's think step by step and output the final answer within \boxed{}.
<|im_end|>
<|im_start|>assistant

GPQA Test Prompt

<|im_start|>system
Reason step by step, and output your final answer (A, B, C, or D) within \boxed{}.
<|im_end|>
<|im_start|>user
{Question} Reason step by step and output the final answer (A, B, C, or D) within \boxed{}.
<|im_end|>
<|im_start|>assistant

Reflection-based Resolution

You are given multiple proposed answers to a math problem. Your task is to carefully examine these answers and determine whether any of them is correct.
- If one of the proposed answers is correct, return it as the final answer.
- If none of the proposed answers is correct, re-solve the problem step-by-step and provide the correct answer.
- Always show the final answer clearly inside \boxed{}.
Question: {question}
Proposed Answers: {answers}
Now, please reflect on the answers above and give the final correct answer in \boxed{}.

B Further Analysis

Table 3 reports the results of the consistency-based selection during pseudo-labeling, where we vary the number of inferences (N) as 2, 3, and 4. We compare the consistent rate, average entropy, total number of selected samples, and accuracy. As expected, a larger N leads to a lower consistent rate. As discussed in Section 3.3, the consistent rate is positively correlated with the total accuracy. Therefore, we set N=2 as the default configuration, which achieves a good balance between performance and computational efficiency.

N	Cons Rate	Entropy	Samples	Accuracy
2	0.1653	0.1550	7483	0.3186
3	0.0983	0.1593	6638	0.3039
4	0.0706	0.1613	6289	0.2822

Table 3: Analysis of the number of inferences (N) in consistency-based selection.

We provide analysis on the threshold setting (0.1–0.5) using Qwen2.5-Math-1.5B across multiple benchmarks. From the results shown in Table 4, we make the following observations: (1) EasyRL demonstrates robust performance across various settings, indicating low sensitivity to the threshold. (2) The threshold determines the percentage of inconsistent samples with the lowest entropy that are selected in each iteration. As the threshold increases, more samples are selected. However, a higher threshold also introduces samples with greater uncertainty or noise. Therefore, when the threshold becomes too large, performance drops, suggesting that the excessive inclusion of uncertain samples negatively affects training. (3) As the threshold decreases, only a limited number of samples are selected per iteration. This restricts the model's exposure to diverse training signals and leads to suboptimal improvement. We choose 0.3 as the default value, as it provides the best balance between sample quality and quantity.

Thres	MATH	Minerva	Olympiad	AIME24	AMC23	Avg.
0.1	69.8	26.5	30.7	13.3	52.5	38.6
0.2	70.0	30.1	34.2	13.3	52.5	40.0
0.3	71.2	30.9	31.3	13.3	55.0	40.3
0.4	71.0	28.3	32.7	6.7	50.0	37.7
0.5	68.2	31.6	31.6	6.7	50.0	37.6

Table 4: Sensitivity analysis of the threshold.

We also extend our experiments to five self-training iterations using Qwen2.5-Math-1.5B. The results are summarized in Table 5. From Iter 1 to Iter 5, we observe the following: (1) Performance steadily improves from Iteration 1 to Iteration 3, demonstrating that iterative self-training effectively enhances the model's reasoning capability in the early stages. (2) After Iteration 3, performance gradually converges, with the average score showing only marginal improvement or slight decline. This suggests that excessive self-training may introduce accumulated noise from pseudo-labels, leading to minor performance degradation.

Iter	MATH	Minerva	Olympiad	AIME24	AMC23	Avg.
1	69.8	28.7	31.0	3.3	52.5	37.1
2	70.0	31.6	31.3	10.0	50.0	38.6
3	71.2	30.9	31.3	13.3	55.0	40.3
4	73.8	32.7	33.0	10.0	52.5	40.4
5	74.2	32.0	33.8	10.0	47.5	39.5

Table 5: Sensitivity analysis of iterations.

Table 6 presents the sensitivity analysis of our method under different ratios of labeled data. The GRPO variants represent standard RL trained with varying amounts of labeled samples, while the EasyRL variants apply our self-evolving mechanism under the same limited supervision. Notably, EasyRL achieves comparable or even superior performance to fully supervised GRPO, demonstrating the effectiveness of our data-efficient RL even with only 1–10% labeled data.

Table 7 summarizes the self-evolution dynamics across three pseudo-labeling iterations. As iterations proceed, the consistency rate and accuracy on D_{unlabel} consistently improve, indicating more reliable pseudo-label generation. Meanwhile, the average difficulty of the selected samples gradually increases, showing that the model progressively shifts its focus from easy to harder problems. In contrast, the accuracy on D_{selected} declines as samples become more challenging.

C Related Work

Curriculum Learning in LLMs. Curriculum Learning trains models on samples of increasing difficulty to improve convergence and generalization, and has been widely applied in computer vision (CV) (Deng et al., 2025), natural language processing (NLP) (Platanios et al., 2019), and reinforcement learning (RL) (Wang et al., 2025c). Traditional curriculum learning methods typically rely on predefined static difficulty metrics to sort labeled training data offline (Lee et al., 2024; Team, 2025). More recent work explores dynamic or adaptive curriculum learning strategies, where sample difficulty is re-estimated during training to adjust batch ordering (Zhang et al., 2025a). In contrast, EasyRL targets a data-scarce setting, where models evolve from limited easy samples to more difficult reasoning tasks. For methodology, EasyRL transfers knowledge from easy samples and progressively incorporates unlabeled data via a divide-and-conquer strategy, enabling difficulty-progressive self-training to enhance reasoning ability.

D Case Study

As shown in Table D, the model demonstrates an instance of self-reflection. Initially, it explores potential solutions by testing the symmetric case $a = b$ and then evaluates specific candidate values such as $a = 1$ or $b = 1$. Upon finding that these attempts yield no valid solutions, the model revisits its reasoning strategy, systematically reconsidering the structure of the equation and leveraging symmetry to identify the correct solution $(a, b) = (\frac{1}{2}, \frac{1}{2})$.

Model	Variant	MATH-500	Minerva MATH	Olympiad Bench	AIME24	AMC23	Avg.
	1% GRPO	65.2	19.1	29.8	10.0	47.5	34.3
	5% GRPO	66.4	24.6	28.3	10.0	45.0	34.9
	10% GRPO	66.4	27.9	30.7	3.3	50.0	35.7
	20% GRPO	68.2	26.8	29.5	3.3	47.5	35.1
	30% GRPO	69.4	27.9	31.3	6.7	50.0	37.1
Qwen2.5-Math-1.5B	100% GRPO	72.6	32.7	33.9	3.3	55.0	39.5
	1% EasyRL	69.8	31.2	33.2	6.7	50.0	38.2
	5% EasyRL	71.0	30.9	33.3	13.3	50.0	39.7
	10% EasyRL	71.2	30.9	31.3	13.3	55.0	40.3
	20% EasyRL	71.4	30.5	32.0	13.3	57.5	40.9
	30% EasyRL	71.6	31.2	34.8	13.3	52.5	40.7
	1% GRPO	71.0	21.7	34.7	26.7	50.0	40.8
	5% GRPO	73.8	32.0	35.6	23.3	60.0	44.9
	10% GRPO	75.6	28.3	37.8	20.0	55.0	43.3
	20% GRPO	77.8	33.5	40.6	20.0	60.0	44.9
	30% GRPO	78.0	39.3	39.4	16.7	62.5	47.2
Qwen2.5-Math-7B	100% GRPO	80.8	39.3	42.2	20.0	65.0	49.5
	1% EasyRL	77.4	42.3	39.1	13.3	65.0	47.4
	5% EasyRL	77.8	41.9	40.0	13.3	62.5	47.1
	10% EasyRL	79.0	43.4	41.2	26.7	62.5	50.6
	20% EasyRL	79.0	39.3	38.5	23.3	62.5	48.5
	30% EasyRL	80.6	39.7	38.7	16.7	70.0	49.1

Table 6: Performance comparison of GRPO and EasyRL under different labeled data ratios.

Models	Iteration	Data	Consistent Rate	Total Samples	Accuracy	Difficulty
	1	Dunlabel	0.1653	7483	0.3186	6.6933
	2	Dunlabel	0.2077	8017	0.3706	6.9274
Qwen2.5-Math-1.5B		Dselected1	0.1251	4078	0.2469	6.7308
	3	Dunlabel	0.2259	8247	0.3829	6.9274
		Dselected2	0.0777	2284	0.1375	7.0282
	1	Dunlabel	0.1939	7844	0.4106	6.5788
	2	Dunlabel	0.3117	9328	0.4597	6.5110
Qwen2.5-Math-7B		Dselected1	0.1747	4289	0.2949	6.7093
	3	Dunlabel	0.3205	9438	0.4646	6.5388
		Dselected2	0.0614	1055	0.1412	7.4256
	1	Dunlabel	0.0982	6637	0.1770	6.8001
	2	Dunlabel	0.2687	8785	0.2444	6.7625
Llama-3.2-3B-Instruct		Dselected1	0.1368	4497	0.1841	6.8611
	3	Dunlabel	0.3168	9392	0.2528	6.7593
		Dselected2	0.1486	2774	0.1388	6.9638

Table 7: Self-evolution dynamics during iterative pseudo-labeling.

Qwen2.5-Math-7B's Answer to the 13-th Question in AMC23

Question: How many ordered pairs of positive real numbers (a, b) satisfy the equation $(1 + 2a)(2 + 2b)(2a + b) = 32ab$?
Response: To solve the equation $(1 + 2a)(2 + 2b)(2a + b) = 32ab$ for positive real numbers a and b, we can use the method of substitution and symmetry. Let's start by expanding and simplifying the left-hand side of the equation.
First, let's expand the left-hand side:

$$(1 + 2a)(2 + 2b)(2a + b) = (1 + 2a)(2(1 + b)(2a + b)) = 2(1 + 2a)(2a + b)(1 + b).$$

... Now, let's check if there are any obvious solutions. One way to do this is to check if $a = b$. If $a = b$, then the equation becomes:

$$(1 + 2a)(2a + a)(1 + a) = 16a^2,$$

... Since a is a positive real number, $a \neq 0$. So we need to solve the quadratic equation:

$$6a^2 - 7a + 3 = 0.$$

The discriminant of this quadratic equation is:

$$\Delta = (-7)^2 - 4 \cdot 6 \cdot 3 = 49 - 72 = -23,$$

which is negative. Therefore, the quadratic equation has no real solutions. This means that $a = b$ does not provide any valid solutions. Next, let's check if there are any other solutions. We can use the method of symmetry and substitution. Let's try $a = 1$ and solve for b:

$$(1 + 2 \cdot 1)(2 + 2b)(2 \cdot 1 + b) = 32 \cdot 1 \cdot b,$$

... The discriminant of this quadratic equation is:

$$\Delta = (-7)^2 - 4 \cdot 3 \cdot 6 = 49 - 72 = -23,$$

which is negative. Therefore, the quadratic equation has no real solutions. This means that $a = 1$ does not provide any valid solutions.
By symmetry, we can try $b = 1$ and solve for a:

$$(1 + 2a)(2 + 2 \cdot 1)(2a + 1) = 32a \cdot 1,$$

... So the solution is: $a = \frac{1}{2}$. By symmetry, if $a = \frac{1}{2}$, then $b = \frac{1}{2}$ is also a solution. Therefore, the only solution is $(a, b) = \left(\frac{1}{2}, \frac{1}{2}\right)$.
Thus, there is only one ordered pair of positive real numbers (a, b) that satisfies the equation. The final answer is: $\boxed{1}$.