Staying in the Sweet Spot: Boost Reasoning Evolution via Responsive Rollout Difficulty Manipulation

Ziheng Li[1*†], **Zexu Sun**[2*], **Jinman Zhao**[3], **Erxue Min**[2], **Yongcheng Zeng**[4], **Hui Wu**[5], **Hengyi Cai**[2], **Shuaiqiang Wang**[2], **Dawei Yin**[2], **Xu Chen**[6‡], **Zhi-Hong Deng**[1‡]
[1] State Key Laboratory of General Artificial Intelligence,
School of Intelligence Science and Technology, Peking University
[2] Baidu Inc. [3] Department of Computer Science, University of Toronto
[4] Institute of Automation, Chinese Academy of Sciences
[5] Aerospace Information Research Institute, Chinese Academy of Sciences
[6] Gaoling School of Artificial Intelligence, Renmin University of China
{liziheng, zhdeng}@pku.edu.cn, sunzexu0826@gmail.com,
xu.chen@ruc.edu.cn

Abstract

Reinforcement learning with verifiable rewards (RLVR) has advanced the reasoning capabilities of large language models (LLMs). However, existing RLVR methods often suffer from exploration inefficiency due to mismatches between problem difficulty and model capability: overly difficult problems hinder reasoning path discovery, while overly simple problems offer little learning signal. To address this, we first formalize the effect of problem difficulty by quantifying the relationship between loss descent magnitude and rollout accuracy. Building on this analysis, we propose **SEELE**, a supervision-aided RLVR framework that dynamically adjusts problem difficulty to lie within the high-performance region. SEELE augments each training sample by appending a hint (part of a full solution) for difficulty reduction. Unlike previous hint-based approaches, SEELE deliberately computes the hint length for each individual problem to achieve an optimal difficulty. The optimal hint length is determined via multi-round rollout sampling, where an item response theory model fits accuracy–hint pairs from previous rounds to predict the next-round hint. This instance-level, real-time difficulty adjustment aligns problem difficulty with the evolving model capability, thereby improving exploration efficiency. Experiments show that SEELE outperforms Group Relative Policy Optimization (GRPO) and Supervised Fine-tuning (SFT) by **+10.0** and **+8.4** points, respectively, and exceeds the best prior supervision-aided approach by **+3.8** points on average across six math reasoning benchmarks.

GitHub: https://github.com/ChillingDream/seele

1 Introduction

Recent large language models (LLMs) such as GPT-5 [33], DeepSeek-R1 [11], Kimi-K2 [40], and GLM-5 [16] have made remarkable breakthroughs in reasoning ability, benefiting from long chain-of-thought that incorporates self-reflection and revision. Prior works [11] have shown that the capability for complex reasoning can be realized through pure reinforcement learning with verifiable rewards (RLVR), which forgoes memorizing annotated reasoning processes and instead elicits

*The first two authors contribute equally.
†Work done during internship at Baidu.
‡Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

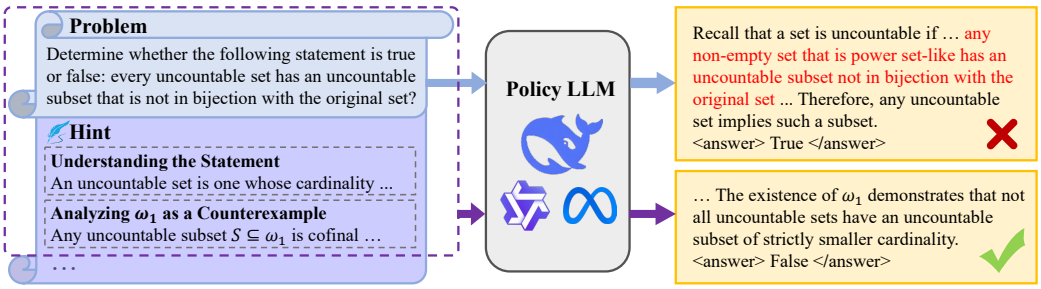

Figure 1: Comparison between the direct rollout (blue) and hinted rollout (purple). The hint consists of the first few steps from an annotated solution. Adding a hint can simplify the problem and guide the LLM toward completing correct solutions.

models' inherent capabilities by extensive on-policy exploration and strengthening verified correct trajectories [7]. At present, RLVR has become the common practice for building high-performance reasoning models.

However, on-policy exploration inherently constrains learning efficiency, exhibiting strong data dependency [15, 12, 36, 46, 50, 39]. RLVR is driven by the rewards from extensive online sampled rollouts, which collapse to zero when the training problems are too difficult for LLMs to produce a correct response. Conversely, overly simple problems yield nearly all correct rewards, producing minor advantage value. It remains unclear what level of problem difficulty maximizes learning efficiency and how to curate such data. Moreover, as recent studies [14, 6, 52] have found, RLVR merely amplifies existing behaviors rather than fostering novel cognitive capabilities, thereby limiting the achievable performance to that of the base model. Supervised fine-tuning (SFT) [26] is a naive way to improve the ability of LLMs before RL with expert data. However, existing works [51, 3] show that directly using SFT-then-RL is not an effective way, which even underperforms pure RL.

To overcome these limitations, recent works have attempted to integrate SFT into the RL framework, enabling synergistic learning when SFT-then-RL is ineffective. LUFFY [44] and SRFT [13] incorporate parallel off-policy guidance into the rollout set, allowing simultaneous exploration and imitation. UFT [29], TAPO [43], StepHint [49], Hint-GRPO [21], and Prefix-RFT [22] append an off-policy hint prefix to each problem to reduce exploration difficulty. While these methods allow for a certain degree of problem difficulty modulation, they have a salient shortcoming: off-policy guidance is applied statically and indiscriminately across problems and hint levels, causing most training samples' difficulty to mismatch the model's evolving capability. Consequently, for those challenging problems, mild off-policy guidance does not effectively resolve the low efficiency of on-policy exploration, whereas for those easy problems, excessive off-policy intervention may impede the LLM from developing its own reasoning patterns. This raises a critical research question:

What is the appropriate problem difficulty, and how can the difficulty be dynamically adjusted in accordance with the model's evolving capability?

In this paper, we propose **SEELE**: ***S**weet-spot st**EE**ring **LE**arning*, a theoretically-grounded supervision-aided RLVR approach that keeps high learning efficiency throughout the whole training stage via dynamically adjusting the proportion of the off-policy prefixes. SEELE explicitly formalizes that an appropriately difficult problem should yield a rollout accuracy of approximately 50% through a theoretical analysis of the loss descent magnitude. By appending an *instance-specific, dynamically determined* hint after the original problem (Figure 1), SEELE is able to control the problem difficulty within the "sweet spot". Unlike previous hint-based methods [29, 22], our approach enables per-instance adaptation and involves the model's real-time feedback as training progresses, thereby achieving a more precise alignment between problem difficulty and model capability. Concretely, we split a rollout sampling stage into several rounds, across which we establish a regression model to predict the accuracy given the hinting rate (proportion of the full solution) under the guidance of item response theory [5]. At each round, we fit the accuracy prediction model using the feedback from the previous rounds and predict how long the current hint should be for a 50% accuracy. We conduct experiments on six math reasoning benchmarks and three out-of-domain general reasoning benchmarks, on which SEELE significantly outperforms previous RLVR methods.

Our contributions can be summarized as:

- We theoretically show that the upper bound on the learning efficiency of RLVR algorithms follows a quadratic relationship with rollout accuracy, peaking at 50% accuracy.
- Guided by our theoretical analysis, we propose a novel capability-adaptive RLVR framework that manipulates the problem difficulty via multi-round rollout sampling and accuracy prediction, maintaining high learning efficiency throughout the entire training process.
- We demonstrate SEELE's superiority over previous RLVR methods on 9 challenging benchmarks, achieving remarkable improvements of **+10.0** points compared with GRPO on math reasoning.

2 Related Work

Reinforcement Learning for LLM Reasoning. Recent work has demonstrated the effectiveness of reinforcement learning (RL) in improving the reasoning capabilities of large language models (LLMs), as exemplified by systems such as DeepSeek-R1 [11], OpenAI-o1 [32], and Kimi-K2 [40]. These studies show that pure RL with verifiable rewards can drive LLMs to autonomously develop extended chain-of-thought reasoning patterns, incorporating substantial self-reflection and iterative refinement. Among these approaches, GRPO-based methods have emerged as a pivotal paradigm for enhancing reasoning performance. Subsequent studies such as DAPO [46], Dr.GRPO [30], GSPO [53] focus on addressing GRPO's optimization limitations by removing length bias, difficulty bias, etc. However, recent works [6, 41] argue that RL approaches are bounded by the capabilities of the base model and do not acquire new reasoning skills, which suggests pure RL may not be the ultimate solution.

Supervision-Aided Reinforcement Learning. Supervised fine-tuning (SFT) on high-quality reasoning chains effectively strengthens reasoning ability and is typically applied before the RL stage [32, 45]. To overcome efficiency and capacity limits of RL, later studies integrate off-policy supervision into RL as a single process. LUFFY [44] enriches RL rollouts with off-policy annotated traces for external guidance, while SRFT [13] jointly optimizes SFT and RL losses using an entropy-based weighting scheme. UFT [29], StepHint [49], and Hint-GRPO [21] append partial solutions (hints) from stronger models to handle difficult problems. Prefix-RFT [22] similarly employs hints but excludes low-entropy hint tokens to avoid over-imitation. TAPO [43] instead introduces high-level "thought patterns" to promote external strategy learning. Yet these methods rely on static supervision, limiting exploration and imitation when tasks are mismatched to the model's capability. In contrast, SEELE grounds hint assignment in principled RL optimization and introduces a capability-adaptive mechanism, enabling dynamic and explicit difficulty adaptation.

Curriculum Learning. Curriculum learning (CL) has been widely studied as a principled strategy for improving optimization by presenting training data in a meaningful order, typically from easy to hard, as first formalized by Bengio et al. [2]. Recent work extends this idea to reinforcement learning for LLM reasoning. Parashar et al. [34] apply a static easy-to-hard task schedule, while Jiang et al. [23], Chen et al. [4] develop dynamic curricula based on reward variance or bandit-based data selection during RL fine-tuning. In parallel, Zhang et al. [48] continuously re-estimate problem difficulty to match the model's evolving capabilities. However, existing approaches mainly rely on heuristic sample selection and therefore cannot achieve precise difficulty matching, while repeated resampling also sacrifices data efficiency. In contrast, our method provides a more direct and efficient solution for capability-aware difficulty control.

3 Preliminaries

Reinforcement Learning with Verifiable Rewards (RLVR) formulates the generation process of an LLM as a Markov Decision Process (MDP), where the state is defined as the concatenation of the prompt x and the tokens generated so far $y_{1:t-1}$, and the action corresponds to selecting the next token y_t from the policy, i.e., $y_t \sim \pi_\theta(\cdot|x, y_{1:t-1})$. The objective of RLVR is to train the policy model to generate outputs that achieve high scores under a rule-based reward function $r(x, y)$. Formally, RLVR optimizes the expected reward over the on-policy rollouts generated from the policy model $\pi_{\theta_{\text{old}}}$ at the previous step, and the objective can be written as

$$\mathcal{L}(\theta) = \underbrace{-\mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\big[A_{\theta_{\text{old}}}(x,y)\big]}_{\mathcal{L}_{\text{pol}}(\theta)} + \beta\, \underbrace{\mathbb{E}_{x\sim\mathcal{D}}[\mathbb{D}_{\text{KL}}(\pi_{\theta_{\text{ref}}}(\cdot|x)\,\|\,\pi_\theta(\cdot|x))]}_{\mathcal{L}_{\text{KL}}(\theta)}, \tag{1}$$

where $A_{\theta_{\text{old}}}(x, y) = r(x, y) - \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)}[r(x, y)]$ denotes the advantage function, pol stands for "policy", and the hyperparameter β controls the strength of KL regularization with respect to the reference model π_{ref}.

GRPO offers an elegant advantage function implementation, which has been widely applied in RLVR studies [37]. Given an input x, GRPO samples a group of outputs $\{y_1, y_2, \cdots, y_n\}$ from $\pi_{\theta_{\text{old}}}(\cdot|x)$ and uses the group mean to replace the expectation. Recently, Liu et al. [30] propose an improved advantage function in their Dr.GRPO approach as

$$A(x, y_i) = r(x, y_i) - \text{mean}(\{r(x, y_j)|j = 1, 2, \cdots, n\}), \tag{2}$$

where the normalization term is removed to mitigate optimization bias. In the following, our analysis is conducted based on this unnormalized formulation.

4 Methodology

In this section, we present SEELE from three aspects: (1) a theoretical foundation that identifies the optimal problem difficulty in terms of rollout accuracy (§ 4.1); (2) a multi-round sampling framework that decomposes rollout generation into sequential rounds, thereby enabling capability-aware adaptation of problem difficulty (§ 4.2); (3) a rollout accuracy prediction model based on Item Response Theory, which supports accurate and responsive adjustment of problem difficulty (§ 4.3).

4.1 Relationship Between Learning Efficiency and Rollout Accuracy

We begin by formulating a quantitative relationship between reinforcement learning (RL) training efficiency and problem difficulty. The model prediction accuracy is employed as the difficulty measure. Given a policy model π_θ and a binary reward function $r(x, y)$, for a problem x, we define the prediction accuracy for x with respect to the policy model π_θ as follows:

$$a_\theta(x) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x, y)]. \tag{3}$$

Prediction accuracy is the expectation of the rollout accuracy and reflects the difficulty of this training instance relative to the current model capability.

Next, we consider a one-step gradient descent from the last-step policy model $\pi_{\theta_{\text{old}}}$ with the update vector d. For analytical convenience, we assume $\theta_{\text{ref}} = \theta_{\text{old}}$. The minimization objective (Eq. (1)) can be reformulated by letting $\theta = \theta_{\text{old}} + d$:

$$\min_d \mathcal{L}(\theta_{\text{old}} + d) = \min_d \left\{\mathcal{L}_{\text{pol}}(\theta_{\text{old}} + d) + \beta\mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\text{KL}}\left(\pi_{\theta_{\text{old}}}(\cdot|x)||\pi_{\theta_{\text{old}}+d}(\cdot|x)\right)\right]\right\}. \tag{4}$$

We conclude that the magnitude of loss descent is upper bounded by the quadratic envelope of $a_\theta(x)$:

$$\mathcal{L}(\theta_{\text{old}}) - \mathcal{L}(\theta_{\text{old}} + d) \leq \frac{1}{2\beta}\mathbb{E}_{x\sim\mathcal{D}}[a_{\theta_{\text{old}}(x)}(1 - a_{\theta_{\text{old}}}(x))]. \tag{5}$$

Here, we briefly outline the derivation. Since $\mathcal{L}$ is defined as the sum over independent prompts x, we analyze the instance-level loss descent $\mathcal{L}(x; \theta_{\text{old}}) - \mathcal{L}(x; \theta_{\text{old}} + d)$ and then aggregate across all instances. From Eq. (4), for a specific instance x, its minimizer d^*_x can be approximated by applying first-order Taylor expansion on $\mathcal{L}_{\text{pol}}$ and second-order Taylor expansion on $\mathcal{L}_{\text{KL}}$ at θ_{old}:

$$d^*_x \approx \arg\min_d \left\{\mathcal{L}_{\text{pol}}(x; \theta_{\text{old}}) + \nabla_\theta\mathcal{L}_{\text{pol}}(x; \theta)\big|^T_{\theta=\theta_{\text{old}}} d + \frac{\beta}{2}d^T F(\theta_{\text{old}})d\right\}, \tag{6}$$

where $F(\theta_{\text{old}})$ is the Fisher Information Matrix. Since $F(\theta_{\text{old}})$ is always positive semi-definite, the right side of Eq. (6) is convex and has a global minimizer[4]:

$$d^*_x = -\frac{1}{\beta}F^{-1}(\theta_{\text{old}})\nabla_\theta\mathcal{L}_{\text{pol}}(x; \theta_{\text{old}}). \tag{7}$$

[4]The policy gradient lies in the range of F. If F is singular, F^{-1} denotes its inverse restricted to this range.

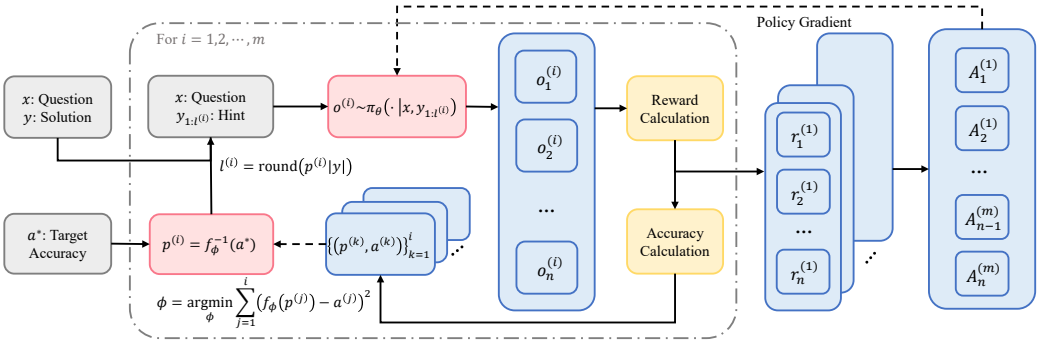

Figure 2: Overview of SEELE. In each step, SEELE conducts m rollout rounds. In round i, an adaptive hint $y_{1:l^{(i)}}$ is appended to the original problem, where $y_{1:l^{(i)}}$ is a prefix of the full solution y with length $l^{(i)}$, determined by the accuracy-hint model f_ϕ to achieve the target rollout accuracy a^*. The output accuracies within each round are then used to update f_ϕ, enabling more accurate predictions in subsequent rounds. Finally, SEELE computes the advantage function over the outputs of all m rounds, which is used to update the policy.

By substituting d_x^* into the Taylor expansion of $\mathcal{L}(x; \theta_{\text{old}} + d)$, we derive the loss descent value:

$$\mathcal{L}(x;\theta_{\text{old}}) - \mathcal{L}(x;\theta_{\text{old}} + d_x^*) \approx \frac{1}{2\beta} \nabla_\theta \mathcal{L}_{\text{pol}}(x;\theta)\big|_{\theta=\theta_{\text{old}}}^T F^{-1}(\theta_{\text{old}}) \nabla_\theta \mathcal{L}_{\text{pol}}(x;\theta)\big|_{\theta=\theta_{\text{old}}} \tag{8}$$

$$= \frac{1}{2\beta} \nabla_\theta a_\theta(x)\big|_{\theta=\theta_{\text{old}}}^T F^{-1}(\theta_{\text{old}}) \nabla_\theta a_\theta(x)\big|_{\theta=\theta_{\text{old}}}. \tag{9}$$

Finally, note that $r(x, y)$ is an unbiased estimator of $a_\theta(x)$. By applying the vector parameter Cramér–Rao bound to Eq. (9) and summing over $x \sim \mathcal{D}$, we get the upper bound shown in Eq. (5) (the equality becomes an inequality because d may not simultaneously satisfy all d_x^*). The full derivation and extension to the multi-step clipped PPO objective are provided in Appendix A. Eq. (5) indicates the convergence rate is correlated with the problem difficulty. The policy model learns little from too easy ($a_\theta(x) \to 1$) or too hard problems ($a_\theta(x) \to 0$), and the maximal efficiency upper bound is achieved at 50% accuracy.

4.2 Difficulty-Aware Hint Manipulation via Multi-Round Sampling

From Eq. (5), we have established how problem difficulty affects learning efficiency. A natural follow-up problem is whether we can deliberately adjust the problem difficulty to lie within the high-efficiency region (around 50%). Recent studies [29, 22, 44, 13] have shown that incorporating off-policy hint guidance into on-policy exploration can enhance the success rate of exploration on challenging samples. However, their strategies for controlling hint length are purely heuristic and lack a principled objective, failing to consider the instance-level difficulty as well as the model's real-time capability. Consequently, the difficulty of hinted problems may deviate from the optimal region, thereby limiting optimization efficiency.

We draw inspiration from these hint-based methods and propose SEELE, a novel instance-level capability-adaptive hint manipulation approach built on GRPO. SEELE adds a dynamic length hint after the original problem to control difficulty to match the increasing model capability, maintaining the prediction accuracy around 50%. To determine the optimal hint length, it is necessary to estimate the difficulty of the problem relative to the model capability at the current timestep. For this purpose, we introduce an accuracy estimator f_ϕ that captures the relationship between prediction accuracy and hint length (the specific form of f_ϕ and its optimization are introduced in Section 4.3).

As shown in Figure 2, we design a multi-round adaptive sampling framework. Different from standard GRPO rollout generation, SEELE distributes the rollouts into m rounds to gradually fit the parameter ϕ. In round i, SEELE first predicts a hinting rate $p^{(i)}$ for reaching target accuracy a^* by the inverse function of f_ϕ. Then, the corresponding part of the solution will be concatenated after x as the input of the policy model for generating n outputs $o_1^{(i)}, o_2^{(i)}, \cdots, o_n^{(i)}$. We calculate the accuracy within

this round as $a^{(i)}$ and add the current hint-accuracy pair $(p^{(i)}, a^{(i)})$ to the memory and update the parameter of f_ϕ. As the number of rounds increases, f_ϕ will model the accuracy more and more accurately and finally make the rollout accuracy stabilize at a^*. Specifically, the first round needs a cold start where we use a default hinting rate $\frac{|y|-1}{|y|}$ in preparation for the worst model capability. The final hinting rate after m rounds will be stored in the dataset so that in the next epoch SEELE can begin exploration from the last predicted rate. Appendix B.3 shows that the computational overhead of the multi-turn rollout is negligible.

After completing all rollout rounds, the mn outputs will be used to calculate the advantages[5]. In GRPO implementation, the advantage is computed at the response level and then distributed to all tokens, which will undermine the model's output probability on input hints if the model fails to explore a correct completion. Hence, we only compute RLVR loss on the generated tokens and impose a supervised loss on the hint tokens to encourage imitation. The final loss is

$$\mathcal{L}(\theta) = -\mathbb{E}_{x\sim\mathcal{D},\, o\sim\pi_\theta(\cdot|\hat{x})}\left[A_{\theta_{\text{old}}}(\hat{x}, o) + \gamma \log \pi_\theta(y_{1:l}|x)\right] + \beta\mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}(\cdot|\hat{x}) \,\|\, \pi_\theta(\cdot|\hat{x}))\right], \tag{10}$$

where $\hat{x}$ denotes the concatenated input $x \oplus y_{1:l}$ and $y_{1:l}$ is the hint decided by multi-round sampling.

4.3 Rollout Accuracy Prediction

The effectiveness of SEELE critically depends on the accuracy of the prediction model f_ϕ, making its design particularly important. f_ϕ should be sufficiently expressive to fit the accuracy-hint relationship while also capable of generalizing from only a few data points. To construct such an accurate yet tractable model, we adopt an established framework used in humans. In psychometrics, Item Response Theory (IRT) [5] studies the relationship between an individual's performance on a test item and the test takers' levels of performance on an overall measure of the ability. The three-parameter logistic (3PL) model is a widely used IRT model that gives the probability that a person with a given ability level will answer correctly:

$$a(\Theta) = b + \frac{1-b}{1+e^{-k_0(\Theta-\nu)}}, \tag{11}$$

where a and Θ indicate the test taker's prediction accuracy and capability, and three model parameters ν, k_0, b represent difficulty, discrimination, and guessing chance, respectively.

In our approach, the difficulty of the problem is tunable depending on the hinting rate. We assume the measure of difficulty ν is linearly related to the hinting rate p as $\nu = -\frac{k}{k_0}p + \nu_0$. Through some symbol substitution and algebraic transformations, we derive the relationship between the accuracy and hinting rate at a certain level of model capability:

$$f_\phi(p) = b + \frac{1-b}{1+e^{-k(p+\mu)}}, \tag{12}$$

where ϕ represents the parameters $\{k, \mu, b\}$ and $\mu = \frac{k_0}{k}(\Theta - \nu_0)$ is the shifted capability measure.

The relationship described in Eq. (12) aligns with both our intuition and empirical observations. When p is small, the problem is too difficult for the model to explore a correct reasoning path, resulting in near-zero accuracy. As p increases and critical steps are gradually revealed, the LLM becomes capable of completing the solution, leading to a rapid increase in accuracy. Once all critical steps are revealed, the model's accuracy approaches 1.0, and providing a longer hint yields no further gain.

Across different problems and training stages, the accuracy-hint curve varies in terms of its starting point, slope, and upper bound. We analyze these curves for 100 randomly sampled problems, with the model generating 100 outputs per problem at each hint level, and find that the 3PL model is sufficiently expressive to capture all observed patterns. Full illustrations are provided in Appendix D.4, while Figure 3 presents representative examples. The observed trends are consistent with IRT predictions, and the 3PL model accurately captures the relationship. Moreover, as the 3PL model involves only three parameters, it requires only a few rounds to obtain an accurate fit. For the model fitting, SEELE employs non-linear least squares:

$$\phi = \arg\min_\phi \sum_{j=1}^{i} \left(f_\phi(p^{(j)}) - \hat{a}^{(j)}\right)^2, \tag{13}$$

[5]Although the hints vary the prompts, the average baseline remains valid, as it is independent of the actions.

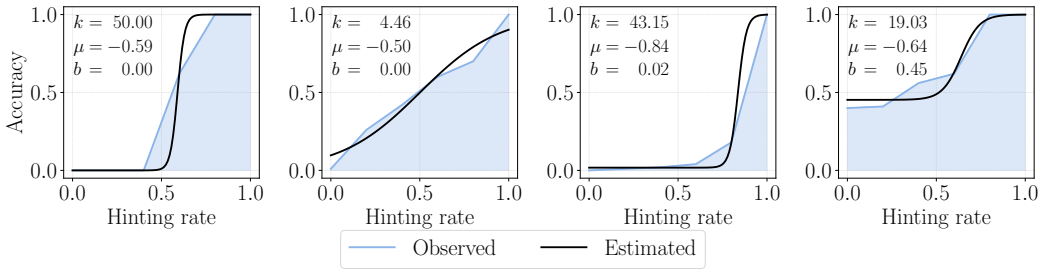

Figure 3: Cases of the accuracy-hint curves and the 3PL fitted curve and parameters. We select 4 typical curves to demonstrate the expressive power of the 3PL model.

where $\hat{a}^{(j)} = \text{mean}(\{a^{(w)} | p^{(w)} = p^{(j)}, w = 1, \cdots, i\})$ is the averaged accuracy across the rounds with the same hinting rate, which helps reduce variance in the accuracy estimation.

5 Experiments

5.1 Setup

Training. We select DeepMath-103K as our training dataset. DeepMath-103K [18] is a large-scale, decontaminated SFT dataset featuring challenging and verifiable mathematical problems, with a strong focus on higher-difficulty problems. To construct a challenging training subset, we retain 22k problems that are particularly difficult, i.e., those on which Qwen2.5-7B [45] fails to produce a correct answer. For these problems, we annotate step-by-step reasoning traces using DeepSeek-V3 [10]. Detailed data synthesis procedure and the annotation prompt are included in Appendix B.4. We adopt GRPO as the RL algorithm, setting the KL coefficient $\beta = 0.001$ and the imitation coefficient $\gamma = 0.001$. We generate 32 rollouts in total with a maximum length of 2,048 tokens for each sample and set $m = 4$ in our multi-round sampling. The tested models include Qwen2.5-Math-7B, Qwen2.5-1.5B/3B, LLaMA 3.2-3B, and Mathstral-7B-v0.1. More hyperparameter, algorithm and prompting details are included in Appendix B.1, B.2 and B.5.

Evaluation. Following prior works [47, 22], we evaluate SEELE on 6 math and 3 general reasoning benchmarks. The math benchmarks are GSM8K [9], MATH500 [19], Minerva [27], Olympiad-Bench [17], AIME24 [28], and AMC23 [1]. The general benchmarks are ARC-Challenge [8], GPQA-Diamond [35], and MMLU-Pro [42]. For AIME24 and AMC23, we report avg@32 due to their small test sets and use a generation temperature of 0.6; for others, we use greedy decoding and report pass@1 accuracy.

Baselines. We compare SEELE with pure GRPO and SFT baselines and recent supervision-aided RLVR methods: LUFFY [44], UFT [29], and Prefix-RFT [22]. Among them, UFT and Prefix-RFT are concurrent hint-based approaches. For a fair comparison, we train all the methods on the same dataset, prompt template, and models and set the hyperparameters following their papers.

5.2 Main results

Table 1 summarizes the performance on Qwen2.5-Math-7B and Qwen2.5-1.5B, while results on other models are included in Appendix C.1. SEELE consistently outperforms all baselines, demonstrating clear advantages across tasks and model scales. For the larger Qwen2.5-Math-7B model, SEELE achieves the best math reasoning average of **59.0%**, surpassing the strongest baseline (Prefix-RFT) by nearly 3 points, and also attains the highest general domain average of **58.5%**. As Qwen2.5-Math-7B is a well-trained math model, it benefits a lot from self-exploration and relies relatively less on the off-policy supervision, which accounts for GRPO's large advantage over SFT. While LUFFY, UFT, and Prefix-RFT incorporate the supervision in the on-policy RL, their strategies are too rigid and fail to boost the training, only achieving marginal improvements over GRPO.

Compared with the 7B setting, SEELE shows even more pronounced advantages on the smaller Qwen2.5-1.5B model, where SEELE improves the math average to **34.2%**, achieving an average improvement of **+10.0** points over GRPO, **+8.4** points over SFT, and **+3.8** points over the second

Table 1: Accuracy on math and general domain reasoning benchmarks.

Model	Math Reasoning							General Domain Reasoning			
	GSM8K	MATH500	Minerva	Olympiad	AIME24	AMC23	Avg.	ARC-C	GPQA-D	MMLU-Pro	Avg.
Qwen2.5-Math-7B	71.6	63.2	25.7	32.0	14.1	45.2	42.0	69.5	24.7	17.7	37.3
+ SFT	89.5	74.6	35.7	37.9	12.4	52.2	50.4	77.6	37.4	47.9	54.3
+ GRPO	92.0	80.6	36.0	41.2	23.9	60.2	55.7	77.1	37.4	45.2	53.2
+ LUFFY	91.7	80.0	35.3	42.4	18.5	66.2	55.7	80.5	39.9	49.9	56.8
+ UFT	92.1	82.4	34.6	40.3	17.6	66.6	55.6	**81.0**	40.9	49.9	57.3
+ Prefix-RFT	92.1	81.6	36.8	43.0	20.9	63.5	56.3	80.3	37.4	49.8	55.8
+ SEELE (Ours)	**92.4**	**82.6**	**37.1**	**46.5**	**25.8**	**69.7**	**59.0**	80.7	**42.9**	**52.0**	**58.5**
Qwen2.5-1.5B	61.9	22.8	9.6	6.7	0.7	9.1	18.5	45.1	15.7	12.2	24.3
+ SFT	67.4	43.6	13.6	12.6	1.4	16.4	25.8	63.7	25.8	30.2	39.9
+ GRPO	70.1	36.4	10.7	11.1	1.8	15.2	24.2	64.8	25.3	23.1	37.7
+ LUFFY	67.2	45.4	11.0	12.9	1.6	16.5	25.8	64.3	26.8	24.7	38.6
+ UFT	72.6	50.4	12.9	15.9	3.9	26.6	30.4	66.1	23.7	28.5	39.4
+ Prefix-RFT	71.5	48.0	13.6	14.5	2.1	22.7	28.7	65.3	23.7	25.0	38.0
+ SEELE (Ours)	**76.5**	**58.0**	**16.2**	**19.9**	**4.1**	**30.4**	**34.2**	**68.3**	**27.8**	**31.7**	**42.6**

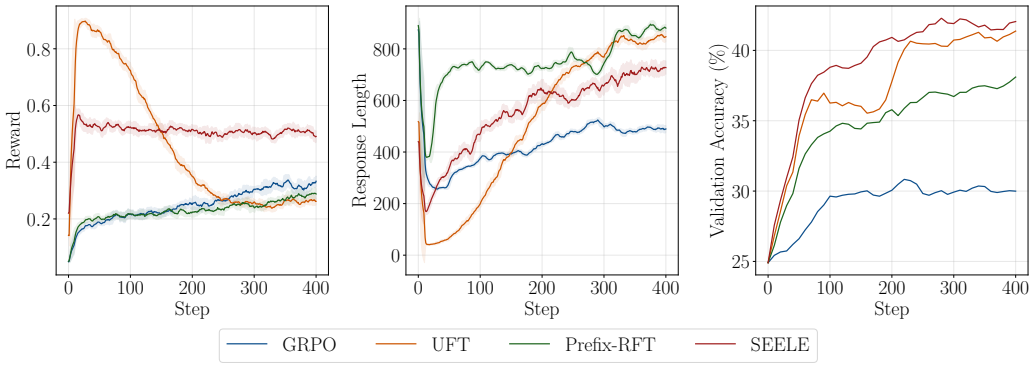

Figure 4: Training dynamics of RL compared with baselines on Qwen2.5-3B. **Left**: training rewards; **Middle**: Response length; **Right**: averaged accuracy on math reasoning validation sets.

best UFT. On the general domain reasoning tasks, SEELE also outperforms the strongest baseline (SFT) by **+2.7** points. The lower initial model capability of Qwen2.5-1.5B underscores the benefits of introducing off-policy demonstrations and leveraging dynamic data difficulty adaptation. We further observe that GRPO generally underperforms SFT on complex reasoning tasks (e.g., MATH500, Minerva, AMC23), while showing comparable or slightly better performance on relatively easy tasks (e.g., GSM8K, ARC-C), demonstrating the limitations of purely on-policy exploration. Moreover, the relatively low performance of GRPO and SFT indicates that exclusive self-exploration or pure imitation alone is insufficient to cultivate strong complex reasoning capabilities.

We provide additional pass@k metrics to assess generation diversity in Appendix C.3, as well as results demonstrating improvements on already strong reasoning models in Appendix C.2.

5.3 Training Dynamics

Figure 4 shows the training dynamics (reward, response length, and validation accuracy) of the SEELE, GRPO, and the two dynamic hinting baselines using Qwen2.5-3B.

Precise Difficulty Control After the initial warm-up, the reward of SEELE rapidly rises to around 0.5 and maintains throughout the remaining training process, indicating our multi-round sampling and regression framework is able to precisely control the rollout accuracy. Due to the adoption of a pessimistic cold-start strategy, the initial reward exceeds the target 0.5 and gradually decreases as training progresses. To further understand SEELE's difficulty control mechanism, we visualize the intermediate results from the multi-round rollout sampling process in Figure 5. Relatively larger fluctuations and deviations are observed in Rounds 1 and 2, primarily because the 3PL regression model receives too few samples to produce accurate fits. Particularly, there is a peak in the first 80 steps (corresponding to the first epoch), which is caused by the high cold-start hinting rate. From the

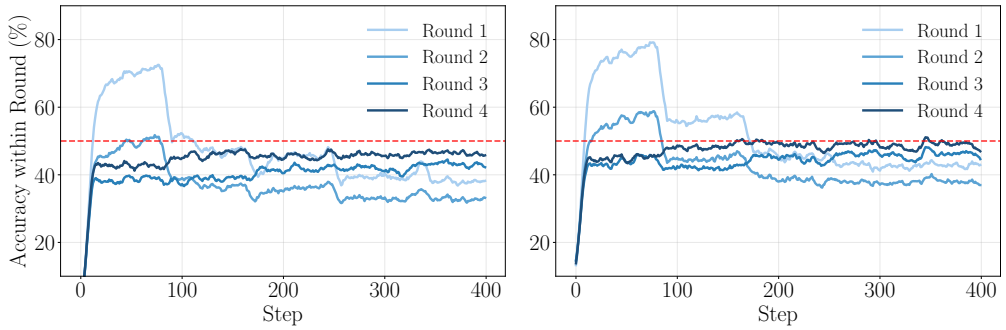

Figure 5: Accuracy within each rollout round during training. The red dotted line denotes the targeted accuracy. **Left**: Qwen2.5-1.5B; **Right**: Qwen2.5-3B.

second epoch onward, SEELE uses the hinting rate rectified in the preceding epoch for the initial round, resulting in more accurate predictions. By Round 3 and Round 4, the accuracy is very close to the target 50%, suggesting that the minimal data requirement for the 3PL model is approximately three samples, while four samples are sufficient to fit a model with adequate precision in practice. We conduct further target control analysis in Appendix D.2 and demonstrate that controlling the accuracy to 50% is beneficial.

Accelerated Learning Compared with GRPO, UFT, and Prefix-RFT, SEELE consistently achieves higher accuracy throughout the entire training process and converges more rapidly. Within the first 100 steps, both SEELE and UFT quickly widen the accuracy gap over GRPO and Prefix-RFT, underscoring the importance of external guidance. GRPO reaches its performance ceiling around step 100, showing minimal improvement thereafter. Notably, its reward continues to increase throughout the remainder of training and ultimately surpasses that of UFT and Prefix-RFT. This behavior suggests that GRPO primarily reinforces previously acquired skills rather than facilitating the learning of new capabilities. From step 100 to step 200, the performance of UFT stagnates because its problem difficulty fails to adapt to the model's evolving capabilities. UFT's holistic pre-designed decaying schedule cannot adequately meet the changing learning requirements. In contrast, SEELE sustains a high growth rate until approximately step 300. From the perspective of response length, SEELE exhibits a more stable growth trend than UFT and Prefix-RFT, which aligns with the performance growth, highlighting the effectiveness of our hint adjustment schedule.

5.4 Ablation Study on Hint-Length Modeling

To verify the effectiveness of the proposed 3PL-based dynamic hinting strategy, we compare SEELE against fixed or random hint lengths $h \in \{1, 2, 3, 4\}$ and a linear regression baseline ($a = kh + b$).

As shown in Table 2, selecting a fixed hint length yields poor performance across tasks. While moderate hint lengths ($h = 2, 3, 4$) substantially outperform the minimal setting ($h = 1$), no single static configuration consistently achieves the best results, indicating that the optimal hint ratio varies across problems. Random hint lengths improve Math Avg over all fixed-length settings but underperform them on Gen Avg, suggesting that varying hint length alone is insufficient. The linear regression baseline further improves over all fixed-length settings, demonstrating the benefit of dynamically adjusting hint length according to problem characteristics. This confirms that adaptive hinting is more effective than manually selecting a constant hint budget.

Table 2: Ablation on hint-length modeling using Qwen2.5-1.5B.

Method	Math Avg	Gen Avg
$h = 1$	24.0	38.7
$h = 2$	31.5	40.7
$h = 3$	31.7	40.5
$h = 4$	31.5	38.6
Random	32.1	38.4
Linear	33.1	41.5
3PL	**34.2**	**42.6**

SEELE achieves the best performance on both math and general benchmarks, surpassing Linear by +1.1 points. These gains suggest that the relationship between hint ratio and learning effectiveness is inherently nonlinear, and therefore better captured by

the proposed 3PL model than by a simple linear approximation. We provide further fitness analysis in Appendix D.4.

6 Conclusion

In this paper, we present SEELE, a novel reinforcement learning with verifiable rewards (RLVR) framework that leverages off-policy demonstrations to dynamically align problem difficulty with the evolving capability of the model, thereby optimizing training efficiency. This framework is motivated by our quantitative analysis, which shows that the upper bound on the learning efficiency of RL algorithms is maximized when the policy model's rollout accuracy is approximately 50%. To maintain rollout accuracy within this "sweet spot", we employ Item Response Theory and construct a multi-round regression model to predict the optimal hint length for fine-grained difficulty adjustment. A key distinguishing feature of our approach, compared with previous supervision-aided RL methods, is that the difficulty manipulation operates at the instance level and incorporates real-time feedback, rendering the training process more responsive and enhancing the utility of each individual training sample. Extensive experiments demonstrate that SEELE significantly outperforms GRPO, SFT, and other RLVR baselines. Our study provides preliminary insights into the types of data favored by RL algorithms and offers a novel direction for improving data efficiency in reinforcement learning.

Limitations

SEELE is primarily designed for scenarios where the difficulty of training problems exceeds the current capability of the model, such that adaptive hinting can effectively adjust the task difficulty into a more learnable regime. When the model is already sufficiently strong on the target tasks, the hinting mechanism may be activated less frequently or become unnecessary, in which case SEELE naturally degenerates to standard GRPO with limited additional benefit. In addition, SEELE relies on decomposing solutions into intermediate hints. For problems with very short reasoning chains or highly compressed solution processes, the granularity of hint segmentation becomes coarser, which may reduce the precision of difficulty control and weaken the advantage of our method. Exploring finer-grained hint construction and extending the framework to broader task settings remain important directions for future work.

Broader Impacts

This work focuses on reinforcement learning with verifiable rewards for improving the reasoning performance of large language models, with experiments mainly conducted on mathematical and other structured tasks with objectively checkable answers. As such, we do not foresee significant new societal risks introduced specifically by our method beyond those already inherent to existing language models. Potential benefits include more accurate and reliable assistance in education, scientific research, and other applications that require verifiable reasoning. Our approach is algorithmic in nature and does not rely on sensitive personal data or deploy domain-specific decision systems.

Disclosure of Funding

This work received no external funding. The authors declare no competing interests.

References

[1] Art of Problem Solving. AMC problems and solutions. `https://artofproblemsolving.com/wiki/index.php?title=AMC_Problems_and_Solutions`.

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[3] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. SFT or RL? An Early Investigation into Training R1-Like Reasoning Large Vision-

Language Models. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=wZI5qkQeDF.

[4] Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-Evolving Curriculum for LLM Reasoning. *CoRR*, abs/2505.14970, 2025. doi: 10.48550/ARXIV.2505.14970. URL https://doi.org/10.48550/arXiv.2505.14970.

[5] Yunxiao Chen, Xiaoou Li, Jingchen Liu, and Zhiliang Ying. Item Response Theory—A Statistical Framework for Educational and Psychological Measurement. *Statistical Science*, 40 (2):167–194, 2025. doi: 10.1214/23-STS896.

[6] Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/537d5aa768c2d534016a4d06f87bc8fb-Abstract-Conference.html.

[7] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training. In *Forty-Second International Conference on Machine Learning*, June 2025.

[8] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. doi: 10.48550/ARXIV.1803.05457. URL https://doi.org/10.48550/arXiv.1803.05457.

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168, 2021. doi: 10.48550/ARXIV.2110.14168. URL https://doi.org/10.48550/arXiv.2110.14168.

[10] DeepSeek-AI. DeepSeek-V3 Technical Report. *CoRR*, abs/2412.19437, 2024. doi: 10.48550/ARXIV.2412.19437. URL https://doi.org/10.48550/arXiv.2412.19437.

[11] DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

[12] Shihan Dou, Muling Wu, Jingwen Xu, Rui Zheng, Tao Gui, and Qi Zhang. Improving RL exploration for LLM reasoning through retrospective replay. In Xian-Ling Mao, Zhaochun Ren, and Muyun Yang, editors, *Natural Language Processing and Chinese Computing - 14th National CCF Conference, NLPCC 2025, Urumqi, China, August 7-9, 2025, Proceedings, Part I*, volume 16102 of *Lecture Notes in Computer Science*, pages 594–606. Springer, 2025. doi: 10.1007/978-981-95-3343-5_46. URL https://doi.org/10.1007/978-981-95-3343-5_46.

[13] Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. SRFT: A Single-Stage Method with Supervised and Reinforcement Fine-Tuning for Reasoning. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust, editors, *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. proceedings.iclr.cc, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a79699db176ed0efc04a9da171e52112-Abstract-Conference.html.

[14] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive Behaviors that Enable Self-Improving Reasoners, or, Four Habits of Highly Effective STaRs. *CoRR*, abs/2503.01307, 2025. doi: 10.48550/ARXIV.2503.01307. URL https://doi.org/10.48550/arXiv.2503.01307.

[15] Jingtong Gao, Ling Pan, Yejing Wang, Rui Zhong, Chi Lu, Qingpeng Cai, Peng Jiang, and Xiangyu Zhao. Navigate the Unknown: Enhancing LLM Reasoning with Intrinsic Motivation Guided Exploration. *CoRR*, abs/2505.17621, 2025. doi: 10.48550/ARXIV.2505.17621. URL https://doi.org/10.48550/arXiv.2505.17621.

[16] GLM. GLM-5: from Vibe Coding to Agentic Engineering. *CoRR*, abs/2602.15763, 2026. doi: 10.48550/ARXIV.2602.15763. URL https://doi.org/10.48550/arXiv.2602.15763.

[17] 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. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211.

[18] 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. DeepMath-103K: A Large-Scale, Challenging, Decontaminated, and Verifiable Mathematical Dataset for Advancing Reasoning. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust, editors, *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. proceedings.iclr.cc, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/df7b97ccf54c528516b62aa9a012d02a-Abstract-Conference.html.

[19] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, Virtual*, 2021.

[20] hiyouga. Mathruler. https://github.com/hiyouga/MathRuler, 2025.

[21] Qihan Huang, Weilong Dai, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, Jingyuan Chen, Chang Yao, and Jie Song. Boosting MLLM Reasoning with Text-Debiased Hint-GRPO. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 4848–4857. IEEE, 2025. doi: 10.1109/ICCV51701.2025.00461. URL https://doi.org/10.1109/ICCV51701.2025.00461.

[22] Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Yinghui Xu, Edoardo M. Ponti, and Ivan Titov. Blending Supervised and Reinforcement Fine-Tuning with Prefix Sampling. *CoRR*, abs/2507.01679, 2025. doi: 10.48550/ARXIV.2507.01679. URL https://doi.org/10.48550/arXiv.2507.01679.

[23] Guochao Jiang, Wenfeng Feng, Guofeng Quan, Chuzhan Hao, Yuewei Zhang, Guohua Liu, and Hao Wang. VCRL: Variance-based Curriculum Reinforcement Learning for Large Language Models. *CoRR*, abs/2509.19803, 2025. doi: 10.48550/ARXIV.2509.19803. URL https://doi.org/10.48550/arXiv.2509.19803.

[24] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. doi: 10.48550/ARXIV.2001.08361. URL https://doi.org/10.48550/arXiv.2001.08361.

[25] Steven M Kay. *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[26] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. OpenAssistant Conversations - Democratizing Large Language Model Alignment. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *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*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/949f0f8f32267d297c2d4e3ee10a2e7e-Abstract-Datasets_and_Benchmarks.html.

[27] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving Quantitative Reasoning Problems with Language Models. In *Advances in Neural Information Processing Systems*, October 2022.

[28] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. https://github.com/project-numina/aimo-progress-prize, 2024.

[29] Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar. UFT: Unifying Supervised and Reinforcement Fine-Tuning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/92e886487a8354b03d8bf4416eae6d7d-Abstract-Conference.html.

[30] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. *CoRR*, abs/2503.20783, 2025. doi: 10.48550/ARXIV.2503.20783. URL https://doi.org/10.48550/arXiv.2503.20783.

[31] Matthew Newville, Renee Otten, Andrew Nelson, Till Stensitzki, Antonino Ingargiola, Daniel Allan, Austin Fox, Faustin Carter, and Michal Rawlik. LMFIT: Non-Linear Least-Squares Minimization and Curve-Fitting for Python, July 2025. URL https://doi.org/10.5281/zenodo.16175987.

[32] OpenAI. OpenAI o1 System Card. *CoRR*, abs/2412.16720, 2024. doi: 10.48550/ARXIV.2412.16720. URL https://doi.org/10.48550/arXiv.2412.16720.

[33] OpenAI. OpenAI GPT-5 System Card. *CoRR*, abs/2601.03267, 2026. doi: 10.48550/ARXIV.2601.03267. URL https://doi.org/10.48550/arXiv.2601.03267.

[34] Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum Reinforcement Learning from Easy to Hard Tasks Improves LLM Reasoning. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust, editors, *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. proceedings.iclr.cc, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fe1d9d163a8b88f413553d99cc48fa5a-Abstract-Conference.html.

[35] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *CoRR*, abs/2311.12022, 2023. doi: 10.48550/ARXIV.2311.12022. URL https://doi.org/10.48550/arXiv.2311.12022.

[36] Thomas Schmied, Jörg Bornschein, Jordi Grau-Moya, Markus Wulfmeier, and Razvan Pascanu. LLMs are Greedy Agents: Effects of RL Fine-tuning on Decision-Making Abilities. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust, editors, *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. proceedings.iclr.cc, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cf2fcda40de8520027af2eadcafdecd6-Abstract-Conference.html.

[37] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL https://doi.org/10.48550/arXiv.2402.03300.

[38] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, March 2025. doi: 10.1145/3689031.3696075.

[39] Zexu Sun, Yiju Guo, Yankai Lin, Xu Chen, Qi Qi, Xing Tang, Xiuqiang He, and Ji-Rong Wen. Uncertainty and Influence aware Reward Model Refinement for Reinforcement Learning from Human Feedback. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=iamWnRpMuQ.

[40] Kimi Team. Kimi K2: Open Agentic Intelligence. *CoRR*, abs/2507.20534, 2025. doi: 10.48550/ARXIV.2507.20534. URL https://doi.org/10.48550/arXiv.2507.20534.

[41] 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 S. Du, and Yelong Shen. Reinforcement Learning for Reasoning in Large Language Models with One Training Example. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/b1ea3f93167c55f3fc2999b68a170ff3-Abstract-Conference.html.

[42] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

[43] Jinyang Wu, Chonghua Liao, Mingkuan Feng, Shuai Zhang, Zhengqi Wen, Pengpeng Shao, Huazhe Xu, and Jianhua Tao. Thought-Augmented Policy Optimization: Bridging External Guidance and Internal Capabilities. *CoRR*, abs/2505.15692, 2025. doi: 10.48550/ARXIV.2505.15692. URL https://doi.org/10.48550/arXiv.2505.15692.

[44] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to Reason under Off-Policy Guidance. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a9d5c33e38442450f9781bd7bad3c81e-Abstract-Conference.html.

[45] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

[46] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

[47] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. SimpleRL-Zoo: Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild. *CoRR*, abs/2503.18892, 2025. doi: 10.48550/ARXIV.2503.18892. URL https://doi.org/10.48550/arXiv.2503.18892.

[48] Enci Zhang, Xingang Yan, Wei Lin, Tianxiang Zhang, and Qianchun Lu. Learning Like Humans: Advancing LLM Reasoning Capabilities via Adaptive Difficulty Curriculum Learning and Expert-Guided Self-Reformulation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6619–6633, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.336. URL https://aclanthology.org/2025.emnlp-main.336/.

[49] Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. StepHint: Multi-level Stepwise Hints Enhance Reinforcement Learning to Reason. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pages 37846–37864. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1755. URL https://doi.org/10.18653/v1/2026.acl-long.1755.

[50] Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. What, How, Where, and How Well? A Survey on Test-Time Scaling in Large Language Models. *CoRR*, abs/2503.24235, 2025. doi: 10.48550/ARXIV.2503.24235. URL https://doi.org/10.48550/arXiv.2503.24235.

[51] Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy RL meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. In Carl Vondrick, Bharath Hariharan, Colin Raffel, Lerrel Pinto, Diyi Yang, and Aleksandra Faust, editors, *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. proceedings.iclr.cc, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c44a04289beaf0a7d968a94066a1d696-Abstract-Conference.html.

[52] Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo Chamber: RL Post-training Amplifies Behaviors Learned in Pretraining. *CoRR*, abs/2504.07912, 2025. doi: 10.48550/ARXIV.2504.07912. URL https://doi.org/10.48550/arXiv.2504.07912.

[53] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group Sequence Policy Optimization. *CoRR*, abs/2507.18071, 2025. doi: 10.48550/ARXIV.2507.18071. URL `https://doi.org/10.48550/arXiv.2507.18071`.

A Derivation of the RL Optimization Efficiency and Rollout Accuracy

A.1 One-step Unclipped PPO Objective

We start from Eq. (1) and assume the reference distribution π_{ref} equals π_{old}. Considering a one-step gradient update with the update vector d, we substitute $\theta = \theta_{\text{old}} + d$ into the loss function and obtain the new loss value as

$$\mathcal{L}(\theta_{\text{old}} + d) = \mathcal{L}_{\text{pol}}(\theta_{\text{old}} + d) + \beta \mathbb{E}_{x\sim\mathcal{D}} \left[\mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_{\theta_{\text{old}}+d}(\cdot|x) \right) \right]. \tag{14}$$

Note that $\mathcal{L}$ is defined as the independent sum of loss on each prompt x. So, we can compute the individual loss for each x and then aggregate the results. For a specific x, we approximate $\mathcal{L}_{\text{pol}}(x; \theta_{\text{old}} + d)$ and $\mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_{\theta_{\text{old}}+d}(\cdot|x) \right)$ using first-order and second-order Taylor expansion at θ_{old}, respectively. First, we expand the policy loss:

$$\mathcal{L}_{\text{pol}}(x; \theta_{\text{old}} + d) \approx \mathcal{L}_{\text{pol}}(x; \theta_{\text{old}}) + \nabla_\theta \mathcal{L}_{\text{pol}}(x; \theta)\big|_{\theta=\theta_{\text{old}}}^T d. \tag{15}$$

$$\begin{aligned} \mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_{\theta_{\text{old}}+d}(\cdot|x) \right) \approx & \mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_{\theta_{\text{old}}}(\cdot|x) \right) \\ & + \nabla_\theta \mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_\theta(\cdot|x) \right) \big|_{\theta=\theta_{\text{old}}}^T d \\ & + \frac{1}{2} d^T \nabla_\theta^2 \mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_\theta(\cdot|x) \right) \big|_{\theta=\theta_{\text{old}}} d. \end{aligned} \tag{16}$$

Next, we compute the first- and second-order derivatives of KL-divergence.

$$\begin{aligned} \nabla_\theta \mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_\theta(\cdot|x) \right) =& \nabla_\theta \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \log \frac{\pi_{\theta_{\text{old}}}(y|x)}{\pi_\theta(y|x)} && (17) \\ =& - \nabla_\theta \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \log \pi_\theta(y|x) && (18) \\ =& - \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \frac{\nabla_\theta \pi_\theta(y|x)}{\pi_\theta(y|x)} && (19) \\ =& - \sum_y \frac{\pi_{\theta_{\text{old}}}(y|x)}{\pi_\theta(y|x)} \nabla_\theta \pi_\theta(y|x). && (20) \end{aligned}$$

$$\begin{aligned} \nabla_\theta \mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_\theta(\cdot|x) \right) \big|_{\theta=\theta_{\text{old}}} =& - \sum_y \frac{\pi_{\theta_{\text{old}}}(y|x)}{\pi_{\theta_{\text{old}}}(y|x)} \nabla_\theta \pi_\theta(y|x) \big|_{\theta=\theta_{\text{old}}} && (21) \\ =& - \nabla_\theta \left[\sum_y \pi_\theta(y|x) \right] \Bigg|_{\theta=\theta_{\text{old}}} && (22) \\ =& 0. && (23) \end{aligned}$$

$$\begin{aligned} \nabla_\theta^2 \mathbb{D}_{\text{KL}} \left(\pi_{\theta_{\text{old}}}(\cdot|x) || \pi_\theta(\cdot|x) \right) =& \nabla_\theta^2 \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \log \frac{\pi_{\theta_{\text{old}}}(y|x)}{\pi_\theta(y|x)} && (24) \\ =& - \nabla_\theta^2 \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \log \pi_\theta(y|x) && (25) \\ =& - \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{\nabla_\theta^2 \pi_\theta(y|x)}{\pi_\theta(y|x)} - \frac{\nabla_\theta \pi_\theta(y|x) \nabla_\theta \pi_\theta(y|x)^T}{\pi_\theta^2(y|x)} \right] && (26) \\ =& \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[-\frac{\nabla_\theta^2 \pi_\theta(y|x)}{\pi_\theta(y|x)} + \frac{\nabla_\theta \pi_\theta(y|x)}{\pi_\theta(y|x)} \frac{\nabla_\theta \pi_\theta(y|x)^T}{\pi_\theta(y|x)} \right] && (27) \\ =& \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[-\frac{\nabla_\theta^2 \pi_\theta(y|x)}{\pi_\theta(y|x)} + \nabla_\theta \log \pi_\theta(y|x) \nabla_\theta \log \pi_\theta(y|x)^T \right] && (28) \end{aligned}$$

We substitute $\theta = \theta_{\text{old}}$ for the first term.

$$\mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{\nabla_\theta^2\pi_\theta(y|x)}{\pi_\theta(y|x)}\right]\Big|_{\theta=\theta_{\text{old}}} = \sum_y \frac{\pi_{\theta_{\text{old}}}(y|x)}{\pi_{\theta_{\text{old}}}(y|x)}\nabla_\theta^2\pi_\theta(y|x)\big|_{\theta=\theta_{\text{old}}} \tag{29}$$

$$= \nabla_\theta^2\left[\sum_y \pi_\theta(y|x)\right]\Big|_{\theta=\theta_{\text{old}}} \tag{30}$$

$$= 0. \tag{31}$$

The second term is $F(\theta_{\text{old}})$ when substituting $\theta = \theta_{\text{old}}$.

So, we get

$$\mathcal{L}(x;\theta_{\text{old}}+d) \approx \mathcal{L}_{\text{pol}}(x;\theta_{\text{old}}) + \nabla_\theta\mathcal{L}_{\text{pol}}(x;\theta)\big|_{\theta=\theta_{\text{old}}}^T d + \frac{\beta}{2}d^T F(\theta_{\text{old}})d. \tag{32}$$

Since the Fisher information matrix is positive semi-definite, the right-hand side of Eq. (32) is convex and has a global minimizer. To find the minimizer d_x^*, let the derivative be zero.

$$\nabla_\theta\mathcal{L}_{\text{pol}}(x;\theta)\big|_{\theta=\theta_{\text{old}}} + \beta F(\theta_{\text{old}})d_x^* = 0. \tag{33}$$

$$d_x^* = -\frac{1}{\beta}F(\theta_{\text{old}})^{-1}\nabla_\theta\mathcal{L}_{\text{pol}}(x;\theta)\big|_{\theta=\theta_{\text{old}}}. \tag{34}$$

Substitute back to Eq. (32)

$$\mathcal{L}(x;\theta_{\text{old}}) - \mathcal{L}(x;\theta_{\text{old}}+d_x^*) \approx \frac{1}{2\beta}\nabla_\theta\mathcal{L}_{\text{pol}}(x;\theta)\big|_{\theta=\theta_{\text{old}}}^T F^{-1}(\theta_{\text{old}})\nabla_\theta\mathcal{L}_{\text{pol}}(x;\theta)\big|_{\theta=\theta_{\text{old}}}. \tag{35}$$

Now, we need to connect $\nabla_\theta\mathcal{L}_{\text{pol}}(x;\theta)\big|_{\theta=\theta_{\text{old}}}$ with $a_{\theta_{\text{old}}}(x)$. According to the definition

$$\nabla_\theta\mathcal{L}_{\text{pol}}(x;\theta) = -\nabla_\theta\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}A_{\theta_{\text{old}}}(x,y) \tag{36}$$

$$= -\nabla_\theta\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r(x,y) - \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)}r(x,y)\right] \tag{37}$$

$$= -\nabla_\theta\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r(x,y)\right] \tag{38}$$

$$= -\nabla_\theta a_\theta(x). \tag{39}$$

$$\mathcal{L}(x;\theta_{\text{old}}) - \mathcal{L}(x;\theta_{\text{old}}+d_x^*) \approx \frac{1}{2\beta}\nabla_\theta a_\theta(x)\big|_{\theta=\theta_{\text{old}}}^T F^{-1}(\theta_{\text{old}})\nabla_\theta a_\theta(x)\big|_{\theta=\theta_{\text{old}}}. \tag{40}$$

Note that $r(x,y)$ is an unbiased estimator of $a_\theta(x)$. By applying the vector parameter Cramér–Rao bound [25, Section 3.8] to Eq. (40), we get

$$\mathcal{L}(x;\theta_{\text{old}}) - \mathcal{L}(x;\theta_{\text{old}}+d_x^*) \le \frac{1}{2\beta}\text{Var}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x)}[r(x,y)] \tag{41}$$

$$= \frac{1}{2\beta}a_{\theta_{\text{old}}}(x)(1-a_{\theta_{\text{old}}}(x)). \tag{42}$$

So, for an arbitrary d

$$\mathcal{L}(x;\theta_{\text{old}}) - \mathcal{L}(x;\theta_{\text{old}}+d) \le \frac{1}{2\beta}a_{\theta_{\text{old}}}(x)(1-a_{\theta_{\text{old}}}(x)). \tag{43}$$

Now we have got the inequality for a single x, taking the expectation we get

$$\mathcal{L}(\theta_{\text{old}}) - \mathcal{L}(\theta_{\text{old}}+d) \le \frac{1}{2\beta}\mathbb{E}_{x\sim\mathcal{D}}[a_{\theta_{\text{old}}}(x)(1-a_{\theta_{\text{old}}}(x))]. \tag{44}$$

A.2 Multi-Step Clipped PPO Objective

Training performs K updates using different mini-batches from a fixed rollout batch, while keeping the sampling policy $\pi_{\text{old}} = \pi_{\theta_{\text{old}}}$ and the advantages fixed. We take $\pi_{\text{ref}} = \pi_{\text{old}}$. To examine how the accuracy affects the decrease in the clipped loss, we analyze a local, per-prompt model of these updates. The proof proceeds in three steps. First, we bound the parameter displacement after K updates. Second, we combine this displacement bound with an upper bound on the clipped advantage and a lower bound on the KL cost, obtaining a scalar maximization problem that bounds the loss decrease. Third, we derive a closed-form upper bound on this maximum.

Preliminaries: Local loss and update rule. Fix a prompt x. Write $a = a_{\theta_{\text{old}}}(x)$ and $v = a(1-a)$, and abbreviate the binary reward advantage $A_{\theta_{\text{old}}}(x, y)$ by $A = r(x, y) - a$. Let $\mathbb{E}_{\text{old}}$ denote expectation over $y \sim \pi_{\text{old}}(\cdot|x)$ and $\rho = \pi_\theta(y|x)/\pi_{\text{old}}(y|x)$. Omitting the fixed argument x inside KL divergences, the loss is

$$\mathcal{L}^{\text{clip}}(x;\theta) = -\mathbb{E}_{\text{old}}[\min\{\rho A, \text{clip}(\rho, 1-\epsilon, 1+\epsilon)A\}] + \beta\, \mathbb{D}_{\text{KL}}(\pi_{\text{old}}\|\pi_\theta). \tag{45}$$

Here $\beta > 0$ and $0 < \epsilon < 1$. Define the score, Fisher matrix, and initial policy gradient by

$$s(y) = \nabla_\theta \log \pi_\theta(y|x)|_{\theta=\theta_{\text{old}}}, \qquad F = \mathbb{E}_{\text{old}}[s(y)s(y)^T], \qquad g = \mathbb{E}_{\text{old}}[As(y)]. \tag{46}$$

We assume locally fixed policy support, twice differentiable probabilities, valid differentiation under the expectations, and finite second moments of s. Then $\mathbb{E}_{\text{old}}[s] = 0$, g is the accuracy gradient at θ_{old}, and F is the Hessian of the KL term there.

We first express the loss in terms of the displacement $d = \theta - \theta_{\text{old}}$. This requires the following local approximation.

Assumption 1 (Local approximation). All K updates remain close enough to θ_{old} for the first-order expansion of the policy ratio and the second-order expansion of the KL term to remain accurate:

$$\rho \approx 1 + s(y)^T d, \qquad \mathbb{D}_{\text{KL}}(\pi_{\text{old}}\|\pi_{\theta_{\text{old}}+d}) \approx \frac{1}{2} d^T F d. \tag{47}$$

The identity $\min\{\rho A, \text{clip}(\rho, 1-\epsilon, 1+\epsilon)A\} = A + \min\{A(\rho-1), \epsilon|A|\}$ and $\mathbb{E}_{\text{old}}[A] = 0$ give the local loss

$$\widetilde{\mathcal{L}}^{\text{clip}}(x;d) = -J(d) + \frac{\beta}{2} d^T F d, \qquad J(d) = \mathbb{E}_{\text{old}}\left[\min\{As(y)^T d, \epsilon|A|\}\right]. \tag{48}$$

Thus $\mathcal{L}^{\text{clip}}(x;\theta_{\text{old}} + d) \approx \widetilde{\mathcal{L}}^{\text{clip}}(x;d)$, with the clipping operation retained. To obtain the update rule, we first differentiate this local loss. A response contributes $As(y)$ to the gradient of J while its contribution is on the linear branch, and zero when it is on the constant, clipped branch. Hence, at $d_k = \theta_k - \theta_{\text{old}}$, define

$$g_k = \mathbb{E}_{\text{old}}[\chi_k As], \qquad \nabla_d \widetilde{\mathcal{L}}^{\text{clip}}(x;d_k) = -g_k + \beta F d_k. \tag{49}$$

Here $\chi_k(y) = 1$ if $As(y)^T d_k < \epsilon|A|$ and $\chi_k(y) = 0$ if the inequality is reversed. At equality, any value in $[0, 1]$ gives a valid subgradient in place of the gradient above.

Having obtained the loss gradient, we next specify how it changes the parameters.

Assumption 2 (Gradient updates). We represent mini-batch gradients by their expectations under π_{old} and apply gradient descent with step size $\eta > 0$, starting from $d_0 = 0$.

Since θ_{old} is fixed, updating θ is equivalent to updating d. Substituting Eq. (49) into the gradient descent step gives

$$d_{k+1} = d_k - \eta(-g_k + \beta F d_k) = (I - \eta\beta F)d_k + \eta g_k. \tag{50}$$

At each step, we recompute g_k according to which responses are clipped under the current parameters.

Because the local KL cost is $\beta d^T F d/2$, we measure displacement by $\|d\|_F = (d^T F d)^{1/2}$. The local loss starts at zero, so its decrease after K steps is

$$\Delta_K(x) := \widetilde{\mathcal{L}}^{\text{clip}}(x;0) - \widetilde{\mathcal{L}}^{\text{clip}}(x;d_K) = J(d_K) - \frac{\beta}{2}\|d_K\|_F^2. \tag{51}$$

We first bound the displacement reached by Eq. (50), then combine an upper bound on the clipped gain with a lower bound on the KL cost. The following inequalities are exact for the local model under the stated conditions.

Step 1: Bounding the parameter displacement. Equation (50) contains two terms to control: the new gradient contribution ηg_k and the previous displacement multiplied by $I - \eta\beta F$. We start with the gradient. Since r is binary, $\mathbb{E}_{\text{old}}[A^2] = a(1-a) = v$. For any direction w, Cauchy–Schwarz and $0 \le \chi_k \le 1$ give

$$(w^T g_k)^2 \le \mathbb{E}_{\text{old}}[\chi_k^2 A^2]\, \mathbb{E}_{\text{old}}[(w^T s)^2] \le v\, w^T F w. \tag{52}$$

Taking $w = F^{-1}g_k$[6] yields

$$\|g_k\|_{F^{-1}}^2 = g_k^T F^{-1} g_k \le v. \tag{53}$$

To use the gradient bound in the displacement recursion, we still need to convert it from the F^{-1}-norm to the F-norm. We also need to control how $I - \eta\beta F$ changes the previous displacement. Both depend on the eigenvalues of F. In particular, along an eigenvector with eigenvalue λ, this matrix multiplies that component by $1 - \eta\beta\lambda$. We therefore introduce the following bounds.

Assumption 3 (Curvature and step-size bounds). Let $\mu > 0$ and $M \ge \mu$ be lower and upper bounds on the positive eigenvalues of F, respectively. We assume

$$\mu \le \lambda \le M \quad \text{for every positive eigenvalue } \lambda \text{ of } F, \qquad 0 < \eta\beta M \le 1. \tag{54}$$

The step-size condition keeps $1 - \eta\beta\lambda$ nonnegative, while the lower curvature bound makes it at most $q := 1 - \eta\beta\mu < 1$. Consequently, $\|(I - \eta\beta F)d\|_F \le q\|d\|_F$. The upper curvature bound also gives $\|g_k\|_F \le M\|g_k\|_{F^{-1}} \le M\sqrt{v}$. Applying these two bounds to Eq. (50) yields

$$\|d_{k+1}\|_F \le q\|d_k\|_F + \eta M\sqrt{v}. \tag{55}$$

Starting from $d_0 = 0$ and summing the geometric series, we obtain

$$\|d_K\|_F \le \sqrt{v}\, h_K, \qquad h_K = \eta M \sum_{j=0}^{K-1} q^j = \frac{M}{\beta\mu}\left[1 - (1 - \eta\beta\mu)^K\right]. \tag{56}$$

This bound tracks the accumulation of successive clipped gradients together with the contraction caused by the KL term.

Step 2: Accounting for clipping and the KL cost. We now bound the clipped gain $J(d)$ within the displacement bound. Since binary rewards give two advantage values, we treat correct and incorrect responses separately. For $0 < a < 1$, set $m = \min(a, 1-a)$ and $z = g^T d/v$. Here $vz = g^T d$ is the first-order change in the prompt's accuracy. Using $\mathbb{E}_{\text{old}}[s] = 0$, we have $g^T d = \mathbb{E}_{\text{old}}[r\, s^T d] = a\,\mathbb{E}_{\text{old}}[s^T d \mid r = 1]$. Combining this with the zero mean of $s^T d$ gives

$$\mathbb{E}_{\text{old}}[s^T d \mid r = 1] = (1-a)z, \qquad \mathbb{E}_{\text{old}}[-s^T d \mid r = 0] = az. \tag{57}$$

For correct and incorrect responses, the clipped contributions to $J(d)$ are $(1-a)\min\{s^T d, \epsilon\}$ and $a\min\{-s^T d, \epsilon\}$, respectively. Weighting the two classes by their probabilities a and $1-a$ gives

$$J(d) = v\big[\mathbb{E}_{\text{old}}[\min\{s^T d, \epsilon\} \mid r = 1] + \mathbb{E}_{\text{old}}[\min\{-s^T d, \epsilon\} \mid r = 0]\big].$$

Since $u \mapsto \min(u, \epsilon)$ is concave, we apply Jensen's inequality to $s^T d$ conditional on $r = 1$ and to $-s^T d$ conditional on $r = 0$. Using the conditional means derived above yields

$$\mathbb{E}_{\text{old}}[\min\{s^T d, \epsilon\} \mid r = 1] \le \min\{\mathbb{E}_{\text{old}}[s^T d \mid r = 1], \epsilon\} = \min\{(1-a)z, \epsilon\}, \tag{58}$$

$$\mathbb{E}_{\text{old}}[\min\{-s^T d, \epsilon\} \mid r = 0] \le \min\{\mathbb{E}_{\text{old}}[-s^T d \mid r = 0], \epsilon\} = \min\{az, \epsilon\}. \tag{59}$$

Substituting these two bounds into the expression for $J(d)$ gives

$$J(d) \le v[\min\{(1-a)z, \epsilon\} + \min\{az, \epsilon\}] = vS_a(z), \qquad S_a(z) = \min\{z, \epsilon + mz, 2\epsilon\}. \tag{60}$$

For $z \ge 0$, the slope of S_a decreases from 1 to m and then to 0, at $z = \epsilon/(1-m)$ and $z = \epsilon/m$. The first change occurs when one class reaches its clipping limit in this bound; the second occurs when both do. At $a = 1/2$, the thresholds coincide. These are thresholds of the bound on the expected contribution; individual responses can become clipped at different steps.

By Eq. (51), the loss decrease is $\Delta_K(x) = J(d_K) - \frac{\beta}{2} d_K^T F d_K$. We have bounded the policy gain from above in terms of $z = g^T d/v$. We next express a lower bound on the KL cost in terms of the same variable, so that the two bounds can be combined.

At the initial point $d_0 = 0$, clipping is inactive and hence $g_0 = g$. Applying Eq. (52) with $k = 0$ and $w = d$ gives $(g^T d)^2 \le v\, d^T F d$. Since $g^T d = vz$, this implies $d^T F d \ge vz^2$. Combining this lower bound with Eq. (60) at $d = d_K$, where $z_K = g^T d_K/v$, yields

$$\Delta_K(x) = J(d_K) - \frac{\beta}{2} d_K^T F d_K \le v\left[S_a(z_K) - \frac{\beta}{2} z_K^2\right].$$

[6] If F has zero eigenvalues, its inverse is understood on the subspace spanned by eigenvectors with positive eigenvalues. Indeed, $Fw = 0$ implies $\mathbb{E}_{\text{old}}[(w^T s)^2] = 0$, hence $w^T g_k = 0$. Starting from $d_0 = 0$, the updates have no component in these zero-eigenvalue directions. If $F = 0$, all gradients and the loss decrease are zero.

This bound still depends on the value of z_K reached after K updates. To bound its range, Eq. (56) gives $d_K^T F d_K \le v h_K^2$. Together with the inequality above, we obtain

$$v^2 z_K^2 = (g^T d_K)^2 \le v\, d_K^T F d_K \le v^2 h_K^2,$$

and therefore $|z_K| \le h_K$. We can thus bound the loss decrease by maximizing $v[S_a(z) - \beta z^2/2]$ over $[-h_K, h_K]$. For $z < 0$, we have $S_a(z) = z$, so this expression is negative, whereas its value at $z = 0$ is zero. The maximum can therefore be restricted to $[0, h_K]$, giving

$$\Delta_K(x) \le v \max_{0 \le z \le h_K} \left[S_a(z) - \frac{\beta}{2} z^2 \right]. \tag{61}$$

Step 3: Derivation of the bound. Write $f_a(z) = S_a(z) - \beta z^2/2$. Since all slopes of S_a are at most 1, for $z \ge 1/\beta$,

$$f_a(z) - f_a(1/\beta) \le z - \frac{1}{\beta} - \frac{\beta}{2}\left(z^2 - \frac{1}{\beta^2}\right) = -\frac{\beta}{2}\left(z - \frac{1}{\beta}\right)^2 \le 0. \tag{62}$$

Thus, the maximum over $[0, h_K]$ equals that over $[0, \min\{h_K, 1/\beta\}]$. Define

$$t_K = \min\{1, \beta h_K\} = \min\left\{1, \frac{M}{\mu}\left[1 - (1 - \eta\beta\mu)^K\right]\right\}. \tag{63}$$

The maximization interval is therefore $[0, t_K/\beta]$.

To evaluate an upper bound on this maximum, use $S_a(z) \le \min\{z, 2\epsilon\}$ and $S_a(z) \le \epsilon + mz$. Define

$$U_K := \max_{0 \le z \le t_K/\beta} \left[\min\{z, 2\epsilon\} - \frac{\beta}{2} z^2 \right], \tag{64}$$

$$V_K := \max_{0 \le z \le t_K/\beta} \left[\epsilon + mz - \frac{\beta}{2} z^2 \right]. \tag{65}$$

Both expressions bound the same maximum from above, so

$$\Delta_K(x) \le v \max_{0 \le z \le t_K/\beta} f_a(z) \le v \min\{U_K, V_K\}. \tag{66}$$

Since $t_K \le 1$, the maximizers defining U_K and V_K are, respectively,

$$z_1^* = \min\{t_K/\beta, 2\epsilon\}, \qquad z_2^* = \min\{t_K, m\}/\beta. \tag{67}$$

Substituting these maximizers gives the following conclusion.

Final bound. With $(u)_+ = \max\{u, 0\}$ and

$$U_K = \frac{1 - \max\{1 - t_K, (1 - 2\beta\epsilon)_+\}^2}{2\beta}, \tag{68}$$

the loss decrease after K updates satisfies

$$\Delta_K(x) \le B_K(a) := a(1-a) \min\left\{ U_K,\ \epsilon + \frac{m^2 - (m - t_K)_+^2}{2\beta} \right\}. \tag{69}$$

For batch updates, we assume that the common displacement d_K satisfies the radius condition in Eq. (56) for every prompt. The preceding pointwise bound then applies along this shared trajectory, and averaging gives

$$\mathbb{E}_{x \sim \mathcal{D}}[\Delta_K(x)] \le \mathbb{E}_{x \sim \mathcal{D}}[B_K(a_{\theta_{\text{old}}}(x))]. \tag{70}$$

Interpretation. The factor t_K captures the cumulative effect of K updates under the specified step size, KL strength, and curvature bounds. The term U_K combines the clipping limit 2ϵ with the quadratic KL cost. We assume common curvature bounds μ, M and fixed hyperparameters η, β, ϵ across prompts, so that t_K and U_K are independent of a. The accuracy dependence of $B_K(a)$ is then entirely through $m = \min\{a, 1-a\}$. As m increases from 0 to $1/2$, the factor $a(1-a) = m(1-m)$ strictly increases, while $m^2 - (m - t_K)_+^2$ is nondecreasing. For $K \ge 1$, the second factor in Eq. (69) is therefore positive and nondecreasing, making $B_K(a)$ strictly increasing in m. Thus, $B_K(a)$ is symmetric about $a = 1/2$, vanishes at $a \in \{0, 1\}$, and attains its unique maximum at $a = 1/2$. The bound retains the sweet spot at $a = 1/2$ while accounting for repeated updates, clipping, and KL regularization.

Algorithm 1 SEELE Policy Optimization

Input: initial policy model π_{init}; training set $\mathcal{D}$; initial hint rate p_0

policy model $\pi_\theta \leftarrow \pi_{\text{init}}$
reference model $\pi_{\text{ref}} \leftarrow \pi_{\text{init}}$
for step=$1, 2, \cdots, T$ **do**
  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  Sample a batch B from $\mathcal{D}$
  $C \leftarrow \{x : \emptyset\}_{x \in B}$ (Here C_x refers to $\{(p_x^{(j)}, a_x^{(j)})\}_{j=1}^{i-1}$)
  for $i = 1, 2, \cdots, m$ **do**
    $B_i \leftarrow \emptyset$
    for $x, y \in B$ **do**
      if $i == 1$ **then**
        $p_x^{(i)} \leftarrow p_0$ if epoch $== 1$, else $p_x^{(m)}$ from the previous epoch
      else
        $\hat{C}_x \leftarrow C_x \cup \{(w, w) | w \in \{0, 1\} \backslash \{p_x^{(j)}\}_{j=1}^{i-1}\}$
        Optimize ϕ on $\hat{C}_x$ (Follow Eq. (13).)
        $p_x^{(i)} = f_\phi^{-1}(a^*)$
      end if
      $\hat{x} = x \oplus y_{1:l}, \quad l = \text{round}(p_x^{(i)}|y|)$
      $B_i \leftarrow B_i \cup \{(\hat{x}, y)\}$
    end for
    Sample n outputs $\{o_j^{(i)}\}_{j=1}^n \sim \pi_{\theta_{\text{old}}}(\cdot|\hat{x})$ for each $\hat{x} \in B_i$
    Compute rewards $\{r_j^{(i)}\}_{j=1}^n$ for each sampled output
    Compute rollout accuracy $a_x^{(i)}$ over $\{r_j^{(i)}\}_{j=1}^n$ for each problem x
    for $x \in B$ **do**
      $C_x \leftarrow C_x \cup \{(p_x^{(i)}, a_x^{(i)})\}$
    end for
  end for
  Compute advantage $\{A_{j,t}^{(i)}\}_{i=1,j=1}^{m,n}$ for each sampled token
  for GRPO iteration=$1, 2 \cdots, u$ **do**
    Update the policy model π_θ by minimizing the objective in Eq. (71).
  end for
end for

Output: π_θ

B Training Details

B.1 Hyperparameter Setting

We adopt GRPO as the RL algorithm, setting the KL coefficient $\beta = 0.001$ and the imitation coefficient $\gamma = 0.001$. Our rollout batch size is 256 and the update batch size is 64. For our approach and all other RL-based baselines, we generate 32 rollouts in total with a maximum length of 2,048 tokens for each sample. For our multi-round sampling, we set the number of rounds $m = 4$ and each round will generate $n = 8$ rollouts. Temperature is set to 1.0 for the rollout generation. Our training is based on veRL [38]. We use MathRuler [20] to verify the correctness of the model's outputs and use the TRF non-linear least squares algorithm provided by the LMFIT library [31] to fit f_ϕ. Our experiments are conducted under the Zero-RL setting, where we use the DeepSeek-R1-Zero prompting template and train the base model for 400 steps. We train SEELE for 2 days on the 7B model using 8 A100 GPUs.

B.2 Policy Optimization Implementation Details

In the practical implementation, we apply the clipping techniques used in GRPO [37] and use π_{ref} rather than π_{old} in KL regularization for simplicity, as shown in Eq. (72) and Eq. (71), which slightly

differs from the formulation in Eq. (10).

$$\mathcal{L}(\theta) = -\mathbb{E}_{x\sim\mathcal{D},o\sim\pi_{\theta_{\text{old}}}(\cdot|\hat{x})}[\hat{A}_{\theta_{\text{old}}}(\hat{x},o) + \gamma \log \pi_\theta(y_{1:l}|x)] + \beta\mathbb{E}_{x\sim\mathcal{D}}[\mathbb{D}_{\text{KL}}(\pi_{\theta_{\text{ref}}}(\cdot|\hat{x})||\pi_\theta(\cdot|\hat{x}))], \tag{71}$$

$$\hat{A}_{\theta_{\text{old}}}(\hat{x},o) = \sum_{t=1}^{|o|}\left\{\min\left[\frac{\pi_\theta(o_t|\hat{x},o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|\hat{x},o_{<t})}A_t, \text{clip}\left(\frac{\pi_\theta(o_t|\hat{x},o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|\hat{x},o_{<t})}, 1-\epsilon, 1+\epsilon\right)A_t\right]\right\} \tag{72}$$

In Algorithm 1, we present the detailed workflow of SEELE policy optimization. At each training step, SEELE first samples a batch from the training set and constructs a hint–accuracy collection C_x for each problem x. Then, SEELE performs m rounds of generation. In each round, C_x is augmented by adding the margin points $(0,0)$ and $(1,1)$ whenever $p_x = 0$ or $p_x = 1$ has not yet been evaluated. This augmentation substantially enhances fitting accuracy and stability during the early rounds. Subsequently, the predictor f_ϕ is optimized on the augmented $\hat{C}_x$ and used to estimate the optimal hint length l. With the estimated hint length, SEELE invokes the policy model to generate n rollouts for the hinted problem $\hat{x} = x \oplus y_{1:l}$ and computes the corresponding rewards. These rewards are then aggregated to calculate the intra-round accuracy a_x, which is used to update C_x. Finally, after m rounds of rollouts, SEELE computes the advantages over all mn outputs following the Dr.GRPO formulation [30], and updates the policy model according to Eq. (71).

B.3 Computation Overhead of SEELE

Although adopting a multi-turn rollout strategy, SEELE has nearly identical FLOPs to other RL methods (under the same number of rollouts). The only additional cost over GRPO comes from multi-round sampling because the prompt KV-cache is not shared across rounds. The extra cost from the non-linear least squares fitting is negligible, as the computation is lightweight (3 parameters) and can be parallelized across rollouts on CPU cores.

To provide a quantitative comparison, we report the FLOPs of each phase in a single RL step in Table 3. Following the prior work [24], we approximate the FLOPs of the forward and backward passes as $2NL$ and $4NL$, respectively, where N denotes the amount of LLM's non-embedding parameters and L denotes the sequence length.

Table 3: FLOPs comparison with GRPO.

Phase	GRPO	SEELE
Rollout	$2NL_p + 2mnNL_r$	$2mNL_p + 2mnNL_r$
Old Prob.	$2mnN(L_p + L_r)$	$2mnN(L_p + L_r)$
Ref. Prob.	$2mnN(L_p + L_r)$	$2mnN(L_p + L_r)$
Update	$6mnN(L_p + L_r)$	$6mnN(L_p + L_r)$
Total	$2NL_p + 2mnNL_r$ $+ 10mnN(L_p + L_r)$	$2mNL_p + 2mnNL_r$ $+ 10mnN(L_p + L_r)$

Under our experimental configuration, $m = 4, n = 8, L_p = 1024, L_r = 2048$, we obtain:

$$\frac{FLOPs(\text{SEELE})}{FLOPs(\text{GRPO})} = 1.0055.$$

Thus, the FLOPs increment introduced by multi-round sampling is only about 0.55%, which is effectively negligible.

B.4 Training Data Synthesis

Our data synthesis procedure consists of two phases: filtering and annotation. We use DeepMath-103K [18] as the initial dataset. First, to construct a more challenging subset, we employ Qwen2.5-7B [45] to sample 8 reasoning traces per instance (temperature $= 0.6$, maximum length $= 2048$) and retain only those instances for which all traces are incorrect. Second, we use DeepSeek-V3 [10] to generate step-by-step reasoning annotations. The annotation prompt is presented as follows. We instruct DeepSeek-V3 to produce a logically complete and concise solution based on the reference solution provided in the original dataset.

Step-by-Step Annotation Prompt

Task: Generate a clear, step-by-step, and complete solution to the following problem with these requirements:
1. **Five or fewer Numbered Steps:**
   Ensure no logical jumps—every non-trivial inference must be justified.
2. **Key Explanations Included:**
   a. Briefly explain "why" for non-obvious steps (e.g., "We use X method because...").
   b. Avoid redefining terms/concepts already introduced.
3. **Full Calculations:**
   a. Show at least one intermediate step for computations (e.g., a = b + c $\rightarrow$ a = 5 + 3 = 8).
   b. For symbolic math, state the rules/theorems used (e.g., "By the chain rule...").
4. **Final Answer:**
   Mark clearly with \boxed{}.

Original Question:
{*question*}
Reference Solution (for guidance only):
{*reference_solution*}
Begin your new solution:

B.5 Chain-of-Thought Prompt Template

Following previous work [29], we adopt the DeepSeek-R1 prompt template for both training and evaluation. This template contains minimal special tokens and is well-suited for the base model. When generation is complete, we first attempt to extract the text enclosed within `<answer> ... </answer>`, which is then used for subsequent answer extraction. If no match is found, we instead extract the content within the \boxed{} tag.

Chain-of-Thought Prompt

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer.
User: {*question*}
Show your work in `<think> </think>` tags. And return the final answer in `<answer> </answer>` tags, for example `<answer>` 12 `</answer>`.
Assistant: Let me solve this step by step.
`<think>`
{*completion*}

C Extended Empirical Analysis

C.1 Performance on Various Models

In addition to Qwen2.5-Math-7B and Qwen2.5-1.5B, we further evaluate SEELE on three base models from different families: Qwen2.5-3B, LLaMA 3.2-3B, and Mathstral-7B-v0.1, and summarize the results in Table 4. Overall, SEELE consistently delivers the strongest performance across both the math reasoning and general domain reasoning benchmarks, demonstrating its wide applicability.

Qwen2.5-3B. In the in-domain mathematical reasoning setting, SEELE achieves an average improvement of **+11.8** points over GRPO and **+10.5** points over SFT, surpassing the best baseline by 1.7%. The improvement is consistent across all six math benchmarks, underscoring the benefits of introducing off-policy demonstrations and dynamically adapting data difficulty. We further observe that GRPO generally underperforms SFT on complex reasoning tasks (e.g., MATH500, AIME24), while showing comparable or slightly better performance on relatively easy tasks (e.g., GSM8K, ARC-C), highlighting the limitations of purely on-policy exploration. Moreover, the relatively low performance of GRPO and SFT indicates that exclusive self-exploration or pure imitation alone is

Table 4: Performance on math and general domain reasoning benchmarks using Qwen2.5-3B, LLaMA 3.2-3B, and Mathstral-7B-v0.1.

Model	Math Reasoning							General Domain Reasoning			
	GSM8K	MATH500	Minerva	Olympiad	AIME24	AMC23	Avg.	ARC-C	GPQA-D	MMLU-Pro	Avg.
Qwen2.5-3B	73.3	29.0	7.0	10.7	0.6	11.6	22.0	66.6	23.7	15.2	35.2
+ SFT	73.1	52.6	18.4	18.5	2.3	25.0	31.7	75.7	30.3	40.1	48.7
+ GRPO	78.5	45.8	17.3	15.0	1.7	23.9	30.4	75.6	**36.9**	40.5	51.0
+ LUFFY	81.1	56.6	21.0	21.3	2.7	27.0	35.0	78.4	32.8	41.5	50.9
+ UFT	84.8	62.6	22.1	25.9	5.8	**41.6**	40.5	79.4	35.4	42.9	52.6
+ Prefix-RFT	77.4	57.0	21.3	21.9	**6.0**	31.5	35.9	**82.2**	35.9	37.9	52.0
+ SEELE (Ours)	**86.3**	**66.4**	**26.1**	**28.9**	5.9	39.4	**42.2**	81.2	34.3	**44.0**	**53.2**
LLaMA 3.2-3B	8.1	4.2	2.6	1.9	0.0	2.7	3.3	13.7	15.2	7.3	12.1
+ SFT	20.7	8.8	4.0	3.0	0.2	4.0	6.8	59.0	27.3	**26.3**	37.5
+ GRPO	5.9	8.2	5.5	2.5	0.0	3.2	4.2	43.5	28.8	23.9	32.1
+ LUFFY	11.9	9.6	4.8	2.4	0.2	3.5	5.4	58.8	**30.8**	26.0	38.5
+ UFT	24.2	11.4	**8.1**	3.6	0.2	**7.5**	9.2	56.2	27.3	25.2	36.2
+ Prefix-RFT	21.1	9.2	5.8	3.6	0.1	2.8	7.1	55.5	28.4	24.9	36.3
+ SEELE (Ours)	**29.5**	**12.8**	7.4	**5.2**	**0.7**	6.5	**10.4**	**62.4**	28.8	26.1	**39.1**
Mathstral-7B	76.0	34.6	15.8	14.5	1.4	15.7	26.3	62.9	29.3	16.7	36.3
+ SFT	80.1	53.4	24.6	21.6	1.8	26.6	34.7	73.1	**44.4**	42.2	53.2
+ GRPO	85.6	44.0	22.4	16.0	0.9	21.7	31.8	80.0	38.4	43.5	54.0
+ LUFFY	88.4	60.0	25.0	24.4	6.4	32.4	39.4	80.2	32.8	44.1	52.4
+ UFT	87.9	57.6	23.2	20.6	4.3	33.1	37.8	81.8	37.9	47.7	55.8
+ Prefix-RFT	86.3	59.6	24.3	22.5	3.6	31.7	38.0	79.7	35.4	46.2	53.8
+ SEELE (Ours)	**90.2**	**63.4**	**27.9**	**29.5**	**6.7**	**38.4**	**42.7**	**82.9**	38.9	**50.5**	**57.4**

insufficient to cultivate strong complex reasoning capabilities. In the out-of-domain general reasoning setting, SEELE demonstrates strong generalization, achieving an average improvement of **+2.2** points over GRPO, while other baselines exhibit only comparable performance.

LLaMA 3.2-3B. A similar trend is observed on the smaller LLaMA 3.2-3B model, where overall math reasoning is more challenging. Despite the low base performance, SEELE improves the math average to **10.4**, outperforming SFT and GRPO by **+3.6** and **+6.2** points respectively. Notably, GRPO again fails to consistently surpass SFT, confirming that naive on-policy reinforcement struggles with scarce reasoning signals. Other supervision-aided RL methods (LUFFY, UFT, Prefix-RFT) provide moderate gains, showing incompetence on the weak initial model. For general domain reasoning, SEELE also reaches the highest average (39.1), exceeding GRPO by **+7.0** and SFT by **+1.6**, showing that our strategy scales effectively even for smaller and weaker base models.

Mathstral-7B. On the larger Mathstral-7B, which already exhibits strong math ability, SEELE continues to be the state-of-the-art. It raises the math average to **42.7%**, surpassing SFT by **+8.0** and GRPO by **+10.9** points, and outperforming the strongest competing RL-with-supervision baseline (LUFFY) by **+3.3** points. In challenging tasks such as Olympiad and AIME24, SEELE achieves the largest absolute gains, demonstrating its ability to handle complex reasoning even when the base model is powerful. For general domain reasoning, SEELE further improves the average to **57.4%**, a **+1.6%** gain over the best baseline, again highlighting superior out-of-domain generalization.

In summary, SEELE is generally effective for training models of different families (Qwen, LLaMA, and Mathstral) and sizes (1.5B, 3B, and 7B), showing great robustness. Notably, for models with very weak math ability (e.g., LLaMA 3.2-3B), SEELE still achieves considerable improvement.

C.2 Performance on Strong Reasoning and Instruction Models

Table 5 further demonstrates that SEELE remains effective beyond base models, consistently improving already strong reasoning and instruction-tuned models. On DeepSeek-R1-Distill-Qwen-7B, a capable reasoning model, SEELE surpasses GRPO by a clear margin, improving Math Avg. from 60.0% to 63.9% and General Avg. from 57.1% to 58.4%, with especially notable gains on more challenging benchmarks such as AIME24 (+11.0%) and Minerva (+3.7%). This suggests that SEELE can further unlock latent reasoning potential even when the model already possesses strong chain-of-thought capabilities.

Similarly, on Qwen3-4B-Instruct-2507, a high-quality instruction-tuned model, SEELE consistently outperforms both SFT and GRPO baselines, achieving the best overall Math Avg. (68.5%) and

Table 5: Performance on math and general domain reasoning benchmarks using DeepSeek-R1-Distill-Qwen-7B and Qwen3-4B-Instruct-2507.

Model	Math Reasoning							General Domain Reasoning			
	GSM8K	MATH500	Minerva	Olympiad	AIME24	AMC23	Avg.	ARC-C	GPQA-D	MMLU-Pro	Avg.
DeepSeek-R1-Distill-Qwen-7B	89.5	72.6	25.7	37.9	17.3	53.3	49.4	69.7	34.8	30.7	45.1
+ GRPO	90.1	84.4	31.2	51.1	27.9	75.5	60.0	75.9	**44.4**	51.1	57.1
+ SEELE (Ours)	**90.3**	**87.2**	**34.9**	**53.0**	**38.9**	**79.0**	**63.9**	**79.1**	42.4	**53.6**	**58.4**
Qwen3-4B-Instruct-2507	94.1	86.6	35.7	49.0	29.4	75.2	61.7	90.2	52.0	60.0	67.4
+ SFT	83.5	65.2	30.1	28.3	7.2	41.0	42.6	89.5	45.5	60.2	65.1
+ GRPO	93.7	90.4	**39.7**	55.4	38.6	87.3	67.5	**92.7**	55.0	70.5	72.7
+ SEELE (Ours)	**94.5**	**92.0**	**39.7**	**56.1**	**40.9**	**88.0**	**68.5**	92.6	**58.1**	**70.8**	**73.8**

Table 6: Pass@k comparison across math and general reasoning benchmarks using Qwen2.5-Math-7B. SEELE consistently achieves the best performance for all k values.

Method	Math Average				General Average			
	P@1	P@4	P@8	P@16	P@1	P@4	P@8	P@16
GRPO	55.7	61.8	64.2	66.3	53.2	57.2	58.6	63.1
UFT	55.6	60.7	63.2	65.3	57.3	64.1	66.0	67.1
Prefix-RFT	56.3	62.1	64.8	67.5	55.8	60.1	61.4	62.2
SEELE (Ours)	**59.0**	**64.5**	**66.0**	**69.6**	**58.5**	**64.9**	**67.0**	**68.2**

General Avg. (73.8%). The gains are particularly evident on difficult mathematical reasoning tasks such as MATH500, Olympiad, and AIME24. These results indicate that SEELE is not limited to weak or untuned base models, but can also serve as an effective zero-RL enhancement for advanced instruction-following models. Overall, the strong performance across these diverse starting points highlights the robustness and broad applicability of SEELE.

C.3 Pass@k Evaluation and Diversity Analysis

To further examine whether SEELE merely follows a single hinted trajectory or truly improves the underlying reasoning policy, we evaluate all methods using the Pass@k metric with $k \in \{1, 4, 8, 16\}$. Pass@k measures whether at least one correct solution appears among k independently sampled rollouts, and thus reflects both single-sample accuracy and the diversity of successful reasoning paths. If gains only arise from collapsing onto one narrow solution mode induced by hints, improvements would mainly appear in Pass@1 while higher-k gains would be limited. In contrast, consistent improvements across all k values indicate that the model discovers a broader and more effective set of reasoning strategies.

As shown in Table 6, SEELE consistently outperforms all baselines across both domains and all sampling budgets. On math tasks, SEELE improves over GRPO from 55.7% to 59.0% at Pass@1, and maintains an advantage at larger sampling budgets, reaching 69.6% at Pass@16. This suggests that SEELE not only increases single-rollout correctness, but also enhances the probability of generating multiple valid solution trajectories.

A similar trend is observed on general reasoning tasks. SEELE attains the strongest Pass@1 performance (58.5%), while maintaining leading results at Pass@4, Pass@8, and Pass@16. Notably, methods that rely on stronger supervised prefixes or direct fine-tuning may improve some low-k settings, but their gains are less consistent as k increases, implying more limited exploration diversity.

Overall, these results support our central hypothesis: strategic hints do not simply steer the model toward one predefined reasoning path. Instead, they reshape the optimization landscape so that reinforcement learning can discover a wider set of high-quality reasoning behaviors.

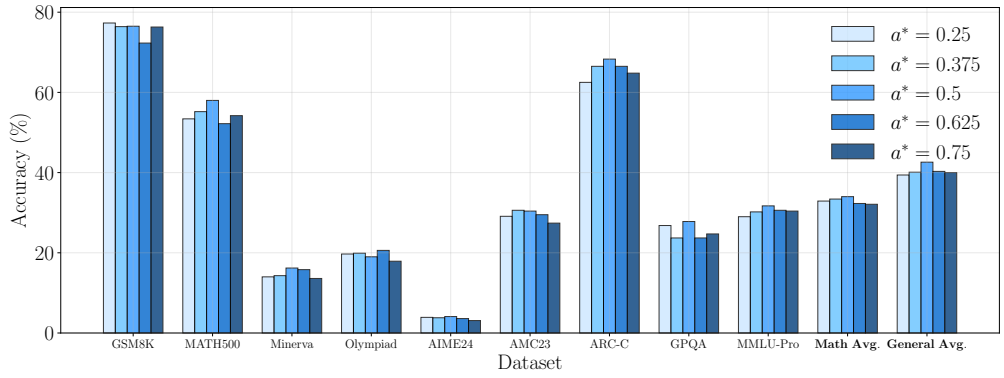

Figure 6: Performance for different target accuracies a^* using Qwen2.5-1.5B.

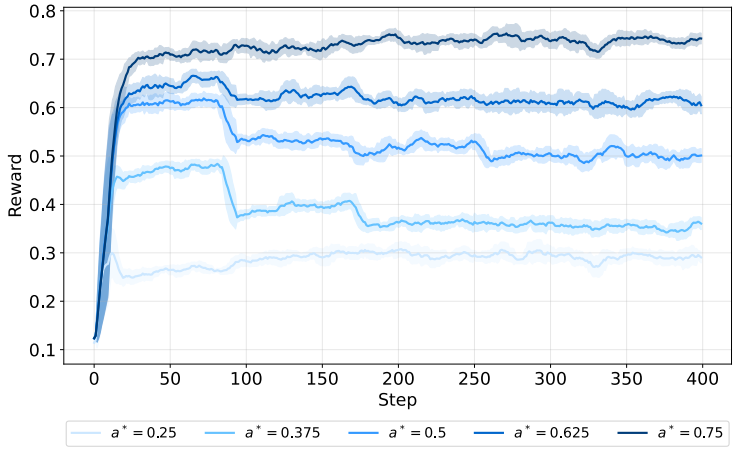

Figure 7: Reward across the training steps for various target accuracies.

D Rollout Manipulation Analysis

D.1 Target Accuracy Analysis

To verify the critical role of rollout accuracy in RLVR, we conduct an ablation study by varying the target accuracy a^* and examining its impact on performance. Specifically, we set a^* to values in the range $\{i/n \mid i = 2, 3, \ldots, n-2\}$, while excluding the boundary cases $i = 1$ and $i = n-1$ to avoid situations where rollouts become entirely incorrect or entirely correct. The results, summarized in Figure 6, lead to the following observations: (1) Setting the target accuracy to 0.5 yields the best performance, while performance degrades gradually as a^* deviates from this value; (2) The degradation trend is approximately symmetric, whether the target accuracy is increased or decreased. These findings are consistent with both intuition and theoretical analysis: data that is either overly difficult or overly simple hinders effective training. For completeness, we also report detailed accuracy control results in Appendix D.2, which further demonstrate that our method can precisely regulate the rollout accuracy to any desired target value.

D.2 Rollout Accuracy Manipulation

To verify that our multi-round sampling framework can arbitrarily manipulate the rollout accuracy, we set the target accuracy to $i/n, i = 2, 3, \cdots, n-2$. We exclude $1/n$ and $(n-1)/n$ because they are too close to the boundary, which easily leads to all wrong/correct rollout sampling. We set $n = 8$ following the main experiment setup. The results are shown in Figure 7. After the cold start for an epoch, the reward converges to the target accuracy across all settings with very little error (less than 0.02 for $a^* > 0.25$), showing a stable trend until the end of the training. The fluctuation and

Table 7: Performance of different multi-round configurations using Qwen2.5-1.5B.

Multi-Round		Math Reasoning							General Domain Reasoning			
Rollout Scheme		GSM8K	MATH500	Minerva	Olympiad	AIME24	AMC23	Avg.	ARC-C	GPQA-D	MMLU-Pro	Avg.
$mn = 8$	$m = 2$	76.0	50.8	16.2	18.2	3.0	28.1	32.1	65.2	26.3	28.1	39.9
$mn = 16$	$m = 4$	76.8	51.0	15.4	19.0	3.3	28.4	32.3	66.3	24.7	29.7	40.2
$mn = 24$	$m = 3$	75.1	53.4	14.7	**20.0**	**4.6**	31.8	33.3	67.9	21.9	29.8	39.9
	$m = 4$	77.1	54.4	**16.9**	19.9	3.0	**32.1**	33.9	66.6	24.7	**33.2**	41.5
	$m = 6$	**77.6**	53.8	15.8	18.5	2.9	30.5	33.2	68.9	23.8	33.0	41.9
$mn = 32$	$m = 4$	76.5	**58.0**	16.2	19.9	4.1	30.4	**34.2**	68.3	**27.8**	31.7	**42.6**
	$m = 8$	76.9	54.2	16.5	19.3	3.3	31.4	33.6	**69.8**	23.2	32.4	41.8

deviation diminish as the training progresses. The results of $a^* = 0.25$ are slightly higher than the target because they touch the difficulty lower bound, where the policy model can generate sufficiently correct outputs without the aid of hints.

D.3 Rollout Scheme Analysis

Given a fixed total number of rollouts, the number of rounds can be configured in multiple ways. Increasing the number of rounds provides more samples for regression, but it reduces the number of samples per round, which in turn increases the variance in estimating accuracy for a specific hinting rate. To examine the actual effect of the multi-round rollout configuration and identify the optimal setting, we train Qwen2.5-1.5B with varying numbers of rollout rounds under different total rollout budgets. The results are presented in Table 7. Our observations are as follows: (1) The performance of SEELE is relatively insensitive to the specific rollout configuration; (2) A three-round scheme underperforms the four-round scheme on both domain averages, as only two sample points are available for constructing the three-parameter logistic (3PL) model, which is insufficient for accurate estimation; (3) A four-round scheme generally achieves the highest performance. Increasing the number of rounds beyond four reduces the number of samples per round, leading to higher estimation variance in single-round accuracy. Based on these findings, we therefore select $m = 4$ as a practical choice for optimal performance.

D.4 Relationship between Prediction Accuracy and Hinting Rate

We analyze the relationship between prediction accuracy and hinting rate using the Qwen2.5-3B checkpoint after 30 training steps, corresponding to the early rising stage of learning. For each example, we enumerate the hint length and prompt the model to complete the hinted problem. Accuracy is computed over 100 randomly sampled traces with a temperature setting of 1.

Figure 8 illustrates the accuracy–hinting rate curves for these 100 examples. In most cases, the curves exhibit an S-shaped trend: accuracy remains close to zero until a critical proportion of the solution is revealed, after which it rises rapidly to nearly 1. This pattern is highly consistent with the predictions of the three-parameter logistic (3PL) model.

To quantitatively evaluate the fitness of different hint-length modeling strategies, we calculate the coefficient of determination (R^2) and root mean squared error (RMSE) over the training set using Qwen2.5-Math-7B as the evaluator model. For each hint length, we generate 100 rollouts to approximate the underlying performance curve, and then fit different predictive models to the observed results.

As shown in Table 8, the 3PL model achieves the strongest goodness-of-fit, with an R^2 of 0.929 and an RMSE of only 0.031, significantly surpassing the Linear alternative. These results indicate that the true relationship between hint proportion and model performance is highly nonlinear, and cannot be adequately captured by a simple linear approximation. Overall, the superior fit quality of 3PL provides strong empirical support for SEELE's dynamic adjustment mechanism. By more accurately estimating the capability-aware

Table 8: Goodness-of-fit comparison.

Method	R^2	RMSE
Linear	0.694	0.164
3PL	**0.929**	**0.031**

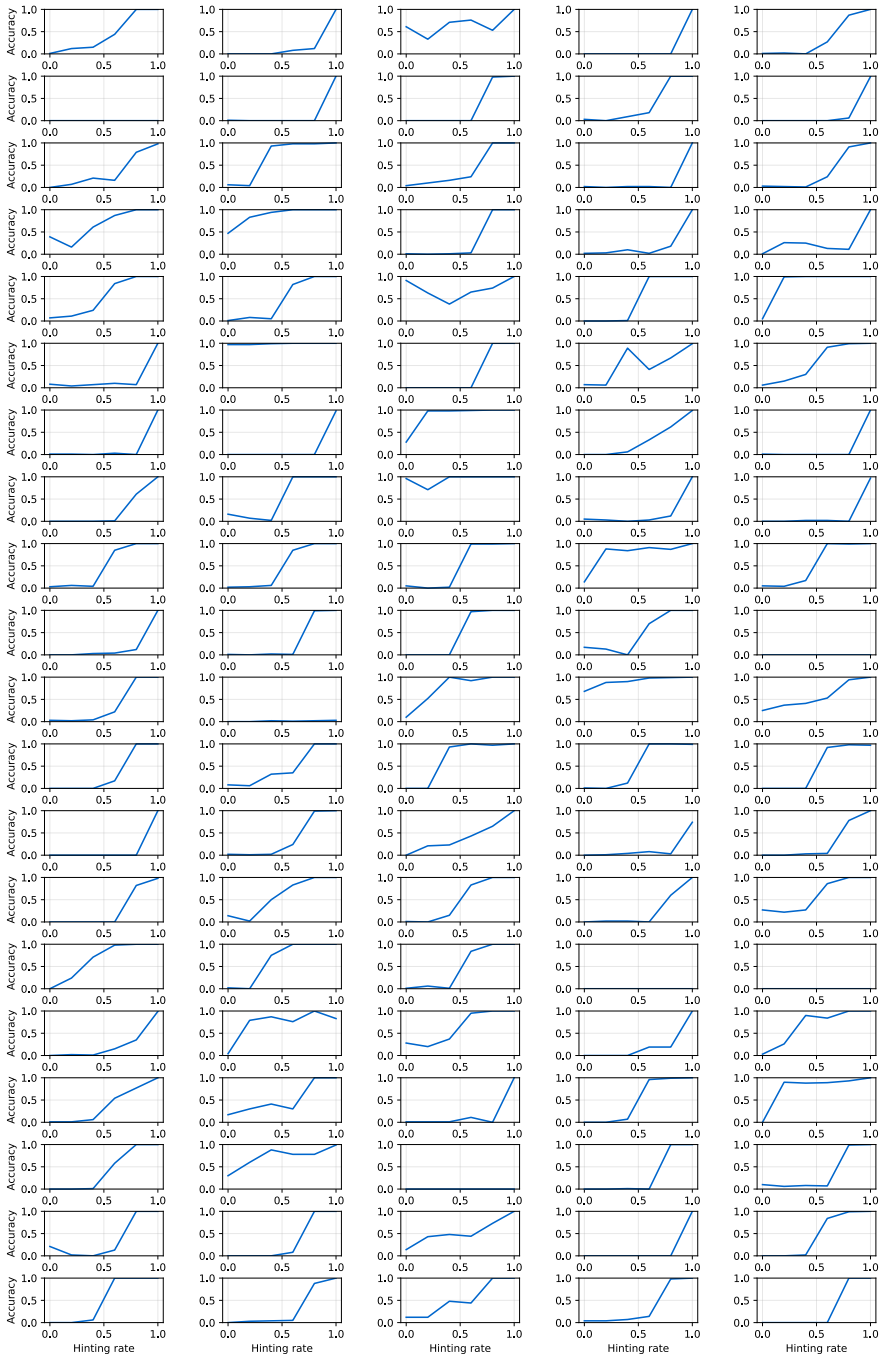

Figure 8: Accuracy with respect to the hinting rate for 100 training examples using Qwen2.5-3B.

optimal hint ratio, SEELE can better match problem difficulty with model competence, leading to improved training efficiency and downstream reasoning performance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to Section 4.1 and Section 5.2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please refer to Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release the code and data at https://github.com/ChillingDream/seele as stated in the abstract.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Appendix B.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We take the average results of multiple runs and find that the variance is small. In the literature, reporting only average accuracy is common practice.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Appendix B.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Broader Impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work does not involve releasing assets that pose a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the relevant dataset, model, and code papers in Section 5.1 and Section B.1 and respect their licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code and data are documented in the accompanying repository. Appendix B provides details of the training procedure and data construction.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve research with human subjects and therefore does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Appendix B.4 describes the use of LLMs for training data filtering and annotation.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.