Self-Evolution Reasoner: Continuous Optimization via Policy-Intrinsic Exploration and Exploitation

Wenhang Shi[1] **Yiren Chen**[2] **Zhe Zhao**[3] **Jinhao Dong**[1] **Wei Lu**[1] **Xiaoyong Du**[1]*
[1]Renmin University of China
[2]Peking University
[3]Tencent
{wenhangshi, dongjinhao, lu-wei, duyong}@ruc.edu.cn,
yrchen92@pku.edu.cn, nlpzhezhao@tencent.com

Abstract

Reinforcement learning with verifiable rewards (RLVR) enables large language models (LLMs) to continuously self-improve their reasoning capabilities. Compared to RLVR's scalar rewards, recent self-supervised on-policy distillation methods leverage denser supervision signals. However, both paradigms suffer from exploration bottlenecks, struggling to escape local optima when the policy fails to sample valid solutions. Relying on external demonstrations can bypass this issue; however, the off-policy nature hinders genuine self-improvement and the mismatched-distribution reasoning can not be fully exploited. In this work, we propose Self-Evolution Policy Optimization (SEPO), a framework that drives continuous reasoning capability evolution via guided exploration and in-distribution exploitation. Specifically, SEPO utilizes labels to elicit correct reasoning from the policy, unlocking the exploration potential. By treating current policy conditioned on this in-distribution reasoning as a self-teacher, SEPO distills the newly discovered solution back into the policy. Through this iterative cycle of policy-intrinsic exploration and exploitation, SEPO achieves robust and continuous capability evolution. Across scientific, tool-use, and complex logical reasoning domains, SEPO consistently achieves the best performance under identical training time or steps, demonstrating superior efficiency and convergence ceilings. Notably, SEPO maintains stable improvements even in scenarios where baselines severely underperform or fail entirely, such as on complex tasks or with small-scale models. Our code is available at `https://github.com/Eric8932/SEPO`.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) enables large language models (LLMs) to develop sophisticated reasoning capabilities by rewarding their own correct rollouts [30, 3, 36]. Recently, on-policy self-distillation methods have emerged, leveraging denser supervision signals to achieve more efficient self-improvement [22, 10]. They treat the current policy with its own correct rollouts in the context as a self-teacher, distilling the enriched predictions back into the policy, thoroughly exploiting the successful solutions. Although both paradigms can continuously improve reasoning capabilities, their optimization relies entirely on the policy's unguided trials, making them susceptible to exploration bottlenecks. When the policy fails to sample a valid solution, these methods cannot formulate effective supervision signals, causing the optimization to stagnate.

Recent works introduce external solutions from experts or proprietary models to mitigate the exploration bottleneck, which generally fall into two categories. The first category distills expert demonstrations for imitation learning, either by fine-tuning on them as an auxiliary training objective [44, 7], or by placing them within the student's context to construct an expert-imitated distillation

*Corresponding author

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

teacher [31, 46, 38]. The second category integrates expert data to guide RL exploration. They either incorporate it into the same GRPO group with the on-policy rollouts for joint learning [7, 39], or extract prefixes to serve as initial contexts, making it easier to sample correct solutions [33, 25]. While relying on external experts circumvents exploration failures, it introduces severe off-policy challenges and causes policy's own exploration to stagnate. Furthermore, the distribution mismatches prevent the policy from fully internalizing these reasoning patterns. Ultimately, the dual deficits in exploration and exploitation stifle continuous improvement and impose a strict performance ceiling.

In this work, we propose Self-Evolution Policy Optimization (SEPO), a framework that drives continuous capability evolution via guided exploration and in-distribution exploitation. Specifically, SEPO utilizes ground-truth labels as privileged hints to elicit correct reasoning from the policy itself. This alleviates the exploration bottleneck by shifting the learning threshold from generating solutions from scratch to performing guided deduction toward correct answers. By treating the current policy conditioned on this in-distribution reasoning as a teacher, SEPO distills the context-enriched predictions back into the standard policy, ensuring thorough internalization of the newly discovered reasoning. Ultimately, through this iterative cycle of policy-intrinsic exploration and exploitation, SEPO achieves a more sustained capability evolution and a higher performance ceiling.

Across extensive experiments on scientific, tool-use, and logical reasoning tasks, SEPO consistently outperforms strong RLVR and self-distillation methods. It not only demonstrates the best convergence performance, indicating a higher ceiling, but also exhibits superior training efficiency, surpassing baseline performance in significantly less time or steps. Notably, even in scenarios where baselines underperform or fail entirely, such as in highly complex logical reasoning tasks or when scaling down to weak models, SEPO maintains stable and continuous performance improvements. Our contributions are summarized as follows:

- We propose SEPO, an on-policy self-distillation framework that drives continuous reasoning evolution through superior exploration and exploitation. By utilizing label to guide reasoning exploration and constructing in-distribution teacher to exploiting it, SEPO iteratively self-improves.
- We present a theoretical analysis revealing how exploration and exploitation deficits limit the optimization of existing methods. Crucially, our analysis formally validates that SEPO fundamentally circumvents these bottlenecks, guaranteeing sustained and effective self-evolution.
- SEPO demonstrates superior training efficiency and final performance across diverse reasoning tasks. Even in scenarios where baselines fail, SEPO robustly secures continuous improvements.

2 Background

2.1 Reinforcement learning with verifiable rewards

Reinforcement learning with verifiable rewards (RLVR) improves the reasoning ability of large language models by optimizing them against verifiable outcome-level rewards [29, 35]. Given an input problem x with ground-truth label y^*, a policy π_θ generates a solution $\tau = (z, \hat{y})$, where z denotes the intermediate reasoning process and $\hat{y}$ the final prediction, and receives a scalar reward $r(\tau; y^*) = \mathbb{I}[\hat{y} = y^*]$, which is typically binary in reasoning tasks. Recently, RLVR is predominantly implemented via Group Relative Policy Optimization (GRPO) [30]. For each problem x, the policy samples a group of G solutions $\{\tau_i\}_{i=1}^G$. The outcome reward r_i for each solution is converted into a group-relative advantage, e.g., $A_{i,t}^{GPRO} = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G) + \epsilon}$. Crucially, this sequence-level advantage is applied uniformly to every token t within the i-th solution.

Despite its broad application in enhancing reasoning, RLVR suffers from two fundamental limitations. First, its supervision is sparse and purely outcome-level: the uniform advantage $A_{i,t}$ provides no direct credit assignment for intermediate reasoning steps [2, 12]. Second, RLVR relies on the policy possessing robust exploration ability. If the policy fails to sample any correct solutions within a group, the algorithm generates no effective learning signal, causing optimization to stagnate [45, 37].

2.2 On-policy context distillation

Recently, several on-policy context distillation (OPCD) methods leverage denser supervision signals to achieve reasoning self-improvement [10, 31, 46]. OPCD conditions the student policy $\pi_\theta(\cdot \mid x)$ on a privileged context c to construct a stronger self-teacher $\pi_{\text{tea}}(\cdot \mid x, c)$. This context c typically

comprises a valid reasoning path, sourced either from student's successful rollouts or external experts, as a reference to simplify the problem-solving process. Specifically, at each training step, solutions $\tau_i \sim \pi_\theta(\cdot \mid x)$ are sampled from the unhinted student policy. In contrast to GRPO's static sequence-level advantage, OPCD provides a token-specific supervision through on-policy distillation [8, 22]. At each token t within solution i, the student assimilates the teacher's predictions by minimizing the Kullback-Leibler (KL) divergence over the student's own generated prefix $\tau_{i,<t}$: $A_{i,t}^{\text{OPCD}} = \mathbb{D}_{\text{KL}}\left(\pi_{\text{tea}}(\cdot \mid x, c, \tau_{i,<t}) \,\|\, \pi_\theta(\cdot \mid x, \tau_{i,<t})\right)$ [14, 9]. Since the teacher shares the student's evolving weights, its contextual reasoning capabilities progressively improve throughout a iterative process, creating a continuous self-improvement flywheel.

2.3 RLVR and OPCD

Fundamentally, OPCD can be viewed as a proxy objective for RLVR [31]. From a variational perspective, the optimal policy π^* that maximizes expected reward satisfies $\pi^*(\tau \mid x) \propto \pi_{\text{ref}}(\tau \mid x)\exp(r(\tau)/\eta)$, where η is a temperature parameter [13, 26]. RLVR methods such as GRPO move toward π^* using sparse rewards, whereas OPCD instead constructs a teacher policy $\pi_{\text{tea}}(\cdot \mid x, c)$ as an approximation to $\pi^*(\cdot \mid x)$ and minimizes its distance to the student policy (with detailed derivations in Appendix A.1). Therefore, when the teacher is a good proxy for π^*, context distillation can drive iterative capability improvement similarly to RLVR, but with much denser supervision. Its effectiveness and stability, however, critically depend on the choice of privileged context c, which determines the quality and compatibility of the teacher distribution.

Existing OPCD methods have critical issues when constructing the context. Relying on student's direct rollout ($c = \tau_{\text{self}}$) guarantees on-policy learning. However, its existence is bounded by the marginal probability of the model's exploration success, $P_{\text{self}}(x) = \mathbb{E}_{\tau\sim\pi_\theta}[\mathbb{I}(r(\tau) = 1)]$. If the policy fails to solve a complex problem within a rollout group, the empirical estimate $\hat{P}_{\text{self}}(x) \to 0$, rendering the teacher undefined and halting the distillation process. Alternatively, utilizing solutions from external experts ($c = \tau_{\text{expert}}$) bypasses the exploration bottleneck. However, these off-policy contexts cannot co-evolve with the student's optimization, inherently imposing a ceiling on self-improvement. Even worse, this paradigm introduces a severe distribution shift, as the teacher π_{tea} often occupies a reasoning subspace outside the student's natural support ($\text{supp}(\pi_{\text{tea}}) \not\subseteq \text{supp}(\pi_\theta)$) [15, 18]. Minimizing this out-of-distribution divergence forces drastic, localized behavioral shifts rather than naturally internalizing the expert's reasoning patterns. This insufficient exploitation prevents the policy from approaching the ceiling set by experts. Full theoretical analyses are in Appendix A.2.

3 Method: Self-Evolution Policy Optimization

Motivation As established, constructing an effective proxy for the optimal policy π^* requires a privileged context c that is simultaneously reliably accessible (to bypass the exploration bottleneck) and strictly on-policy (to avoid distribution shifts). To achieve this, as illustrated in Figure 1, we develop the Self-Evolution Policy Optimization (SEPO) framework. By harnessing a synergy between guided exploration and in-distribution exploitation, SEPO iteratively mines and elevates policy-intrinsic reasoning capacity, realizing continuous self-evolution.

3.1 The SEPO framework

To circumvent the exploration bottleneck while preserving on-policy exploitation, the Self-Evolution Policy Optimization (SEPO) framework integrates guided reasoning exploration and in-distribution exploitation processes into a synchronized online training loop. As shown in Algorithm 1, at each training step, given a problem x and its ground-truth label y^*, the student policy π_θ first samples rollouts $\tau \sim \pi_\theta(\cdot \mid x)$. Concurrently, the teacher policy π_{tea} receives a label-informed prompt (e.g., "Correct answer: y^*; Reason naturally ...") to elicit a

Algorithm 1 SEPO Framework

Input: Language Model π_θ, initial weights θ_0, Dataset $\mathcal{D}$, total steps N

Initialize $\theta \leftarrow \theta_0$ and $\theta_{\text{tea}} \leftarrow \theta_0$
for $n = 0$ to $N - 1$ **do**
  Sample a batch $\{(x_i, y_i^*)\}_{i=1}^{B} \sim \mathcal{D}$
  for each (x, y^*) in the batch **do**
    Sample rollouts $\tau \sim \pi_\theta(\cdot \mid x)$
    Elicit reasoning $z_{\text{in}} \sim \pi_{\theta_{\text{tea}}}(\cdot \mid x, y^*)$
  end for
  Update θ by minimizing Eq. 1
  $\theta_{\text{tea}} \leftarrow (1-\beta)\theta_{\text{tea}} + \beta\theta$
end for
return π_θ

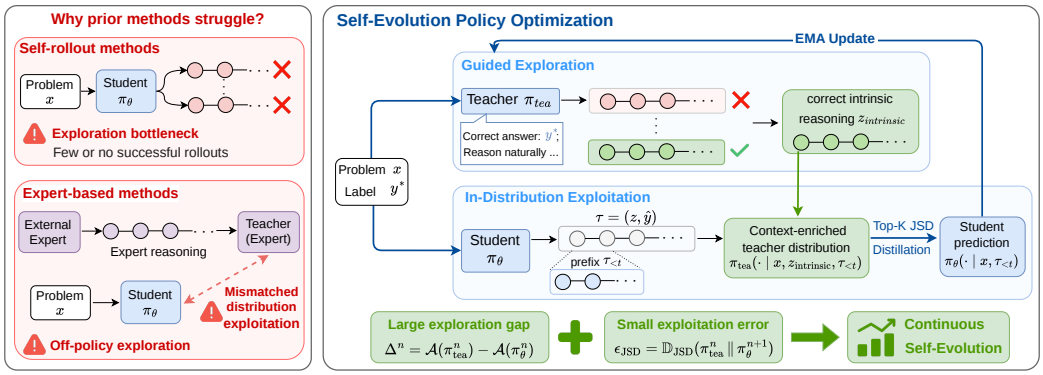

Figure 1: Overview of Self-Evolution Policy Optimization (SEPO) framework. SEPO couples guided exploration with in-distribution exploitation, overcoming the exploration bottlenecks and distribution-mismatched exploitation of prior methods, enabling more sustained self-evolution.

set of reasoning processes and retain only the correct ones, $z_{\text{intrinsic}} \sim \pi_{\text{tea}}(\cdot \mid x, y^*)$. Conditioned on this privileged context $z_{\text{intrinsic}}$, the stop-gradient teacher computes the next-token distribution over student's rollout prefix $\tau_{<t}$. By minimizing the divergence between the student's predictions and the context-informed teacher's distribution, SEPO continuously pulls the student's unhinted reasoning toward its own label-guided potential. The detailed training procedure is provided in Algorithm 2.

Teacher initialization and EMA update. SEPO initializes π_{tea} with the student's weights θ and updates it via an Exponential Moving Average (EMA): $\theta_{\text{tea}} \leftarrow (1-\beta)\theta_{\text{tea}} + \beta\theta$, where $\beta \in [0, 1)$. This momentum-based strategy serves two purposes: it ensures the context $z_{\text{intrinsic}}$ evolves synchronously with the student's growing capabilities to facilitate continuous self-improvement, and it provides a smoothed, implicitly ensembled target that prevents confirmation bias and stabilizes learning.

Stable distillation via bounded divergence and tail truncation. Standard distillation typically relies on Forward KL divergence ($\mathbb{D}_{\text{KL}}(\pi_{\text{tea}} \,\|\, \pi_\theta)$), which enforces mode-covering behavior [23, 9]. It heavily penalizes the student for assigning low probability to preferred actions, frequently triggering gradient explosions and catastrophic forgetting [1]. In contrast, Reverse KL ($\mathbb{D}_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{tea}})$) induces mode-seeking behavior [8]. While this protects the model's base capabilities by only penalizing student's generations outside the teacher's support, it strictly stifles exploration, causing the student to collapse into rigid reasoning paths. To mitigate these extremes, SEPO adopts the Jensen-Shannon Divergence (JSD) [19]. As a symmetric and bounded metric, JSD naturally balances mode-covering and mode-seeking behaviors, providing stable gradients that allow the student to safely assimilate the elicited reasoning. Furthermore, we approximate the analytic KL divergence by restricting the learning distribution to the top-K tokens predicted by the student model [10]. This tail truncation significantly reduces the memory footprint while preserving the performance. Integrating these components, the unified optimization objective for SEPO is formulated as:

$$\mathcal{L}_{\text{SEPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{\tau\sim\pi_\theta}\left[\frac{1}{|\tau|}\sum_{t=1}^{|\tau|}\mathbb{D}_{\text{JSD}}^{\text{Top}_K}\left(\tilde{\pi}_{\text{tea}}(\cdot \mid x, z_{\text{intrinsic}}, \tau_{<t}) \,\|\, \pi_\theta(\cdot \mid x, \tau_{<t})\right)\right]\right]. \quad (1)$$

3.2 Theoretical analysis of SEPO

SEPO achieves continuous self-improvement. To demonstrate how SEPO achieves continuous self-improvement, we analyze its training dynamics using a generalized accuracy lower bound [11, 28]. Let $\mathcal{A}(\pi)$ denote the expected accuracy of a policy. During exploration of privileged context, SEPO leverages label y^* to sample from the posterior reasoning distribution $P(z \mid x, y^*)$ rather than the unguided prior $P(z \mid x)$. This posterior conditioning exponentially prunes the vast search space, yielding a clear capability advantage over the unhinted student at any training step n. We formalize this advantage as the Exploration Gap: $\Delta^n = \mathcal{A}(\pi^n_{\text{tea}}) - \mathcal{A}(\pi^n_\theta)$. During distillation, SEPO optimizes the student to minimize the Jensen-Shannon Divergence. Let $\epsilon_{\text{JSD}} = \mathbb{D}_{\text{JSD}}(\pi^n_{\text{tea}} \,\|\, \pi^{n+1}_\theta)$ denote the

residual exploitation error, which quantifies the inevitable optimization imperfection remaining after step n. By relating JSD to the Total Variation distance via information-theoretic bounds, this residual error induces a worst-case performance penalty of $\mathcal{O}(\sqrt{\epsilon_{\text{JSD}}})$. Consequently, the expected accuracy of the updated student policy at step $n+1$ is bounded by:

$$\mathcal{A}(\pi_\theta^{n+1}) \geq \mathcal{A}(\pi_{\text{tea}}^n) - \mathcal{O}(\sqrt{\epsilon_{\text{JSD}}}) \implies \mathcal{A}(\pi_\theta^{n+1}) \geq \mathcal{A}(\pi_\theta^n) + \Delta^n - \mathcal{O}(\sqrt{\epsilon_{\text{JSD}}}).$$

As long as the exploration gap outweighs the exploitation error ($\Delta^n > \mathcal{O}(\sqrt{\epsilon_{\text{JSD}}})$), the student policy improves. Through SEPO's synchronous evolution, teacher's capability elevates alongside the student, continuously regenerating a substantial Δ^n to drive continuous self-improvement. Ultimately, as the student's intrinsic capabilities grow, the posterior advantage of the label diminishes. Training converges to its empirical upper bound when the exploration gap shrinks to the level of the exploitation error ($\Delta^n \approx \mathcal{O}(\sqrt{\epsilon_{\text{JSD}}})$). Detailed theoretical analysis is in Appendix A.3.

SEPO's superiority over existing methods. The recursive inequality explains the superiority of SEPO over other OPCD methods, as SEPO uniquely optimizes both critical terms to maximize the performance upper bound: 1) For methods using student's own rollouts as context, Δ is bottlenecked by the low prior probability of directly sampling a valid solution. Consequently, their performance ceiling is prematurely locked at the model's unguided exploration limit. In contrast, SEPO guarantees a more persistently positive Δ. By utilizing guided exploration, SEPO raises the upper bound from the autonomous discovery limit to the posterior exploration limit—the true extent of what the model can solve when accurately guided by the outcome. 2) Methods using external solutions as context may initially provide a high teacher accuracy, but they introduce severe support mismatch between the expert's latent manifold and the student's weights. This mismatch makes the exploitation error ϵ_{JSD} intrinsically enormous. When the exploitation error exceeds the exploration gap, the performance gain vanishes, stagnating the optimization. In contrast, SEPO's strictly on-policy exploration guarantees a minimal ϵ_{JSD}, allowing the student to absorb the teacher's capability with little optimization friction.

4 Experiments

4.1 Experimental settings

We evaluate on reasoning tasks spanning diverse difficulties and paradigms (knowledge, planning, and logic), which are widely employed as standard testbeds to demonstrate method efficacy [10, 31, 46]:

- **Science Q&A**: Undergraduate-level scientific reasoning tasks across Chemistry, Physics, Biology, and Material domains, using the L3 subsets from SciKnowEval [6].
- **Tool Use**: Planning tool invocations based on user queries and provided tool specifications, utilizing the ToolAlpaca dataset [34].
- **Logic**: High-difficulty logical reasoning tasks, evaluated on two subsets (Arrow_Maze and Dyck_Language) from SynLogic [20].

We apply a train-test split for each dataset to assess in-domain generalization. We employ models with diverse architectures, specifically Qwen3-8B [36] and Olmo3-7B-Instruct [5], as well as Qwen3 models of varying scales. To ensure robust evaluation, all experimental results are averaged across three independent runs with different random seeds. We report the peak test accuracy (avg@16), measured as the average over 16 sampled responses, achieved within a fixed time or training step.

Baselines We evaluate SEPO against GRPO and other OPCD methods. We implement an enhanced GRPO baseline following [10], integrating several recent algorithmic improvements, such as asymmetric clipping [42] and off-policy data correction [40]. We evaluate both on-policy and off-policy variants of GRPO, and primarily report the results of the better-performing off-policy version. For the OPCD baselines, we compare three distinct teacher contexts: OPCD (*Self*) uses student's own correct rollouts [10]; OPCD (*Expert*) utilizes reasoning processes from DeepSeek-V3.2, which yields the highest performance among several candidate experts [31, 46, 4]; and additionally, we include OPCD (*Label*), a simplified variant of SEPO that solely provides the ground-truth label within the context. To ensure a fair comparison, we maintain identical core training configurations across all methods, such as the global training batch size and the number of sampled responses per prompt. And all OPCD baselines use the same EMA teacher-update rule as SEPO. For other hyperparameters, we perform a rigorous sweep, selecting the configuration for each method that yields the highest

Table 1: Performance comparison on science QA and tool use benchmarks, reporting the highest avg@16 achieved within 1 hour and 10 hours of training time. **Bold** and underlined denote the best and second-best performance, respectively. SEPO consistently achieves the optimal or near-optimal performance at both time marks, demonstrating superior training efficiency and performance ceilings.

	Chemistry		Biology		Physics		Material		Tool use	
	1h	10h	1h	10h	1h	10h	1h	10h	1h	10h
Qwen3-8B	41.5		30.5		58.6		59.8		57.1	
+ GRPO (off-policy)	48.0	73.9	30.5	68.1	59.5	74.8	70.9	76.1	60.9	67.6
+ OPCD (*Self*)	66.4	81.4	49.4	67.9	62.2	**77.0**	67.6	74.5	63.6	65.7
+ OPCD (*Expert*)	72.5	78.4	50.1	57.4	**63.6**	73.8	72.5	76.7	63.3	64.6
+ OPCD (*Label*)	58.8	78.5	38.5	66.9	59.9	68.5	72.9	77.8	62.5	62.5
+ **SEPO**	**73.8**	**82.9**	**55.1**	**69.0**	62.8	76.4	**73.8**	**78.4**	**64.6**	**68.2**
Olmo3-7B-Instruct	25.1		13.0		37.5		37.6		45.3	
+ GRPO (off-policy)	25.4	54.7	26.0	50.6	40.5	53.2	61.1	73.9	52.7	63.4
+ OPCD (*Self*)	58.3	80.9	45.1	51.5	46.3	70.2	**65.4**	76.9	52.7	61.5
+ OPCD (*Expert*)	43.5	64.7	45.9	56.0	**48.4**	64.3	63.9	76.7	52.9	54.6
+ OPCD (*Label*)	34.2	54.3	30.5	46.5	38.7	49.1	62.3	66.2	51.0	60.0
+ **SEPO**	**59.7**	**82.1**	**46.4**	**57.4**	47.0	**71.1**	64.7	**78.4**	**53.6**	**63.8**

validation performance across all tasks. Comprehensive implementation and hyperparameter settings are provided in Appendix D.

4.2 Main results

Table 1 presents a comparison of the peak performance achieved by different methods under a short time budget (1 hour) and at convergence (10 hours) across the scientific and tool-use reasoning domains. Figure 2 (Left) illustrates the training progression of these methods on Chemistry task using Qwen3-8B. As observed, SEPO consistently secures optimal or near-optimal performance, demonstrating both superior short-term training efficiency and a higher long-term convergence ceiling.

Analyzing the 1-hour results, GRPO consistently underperforms, confirming that OPCD's denser supervision accelerates initial training. Among OPCD baselines, the *Expert* method achieves rapid early gains, often reaching near-optimal performance at the 1-hour mark, but its 10-hour performance prematurely plateaus. This indicates that while external demonstrations can bootstrap rapid initial learning, the severe off-policy and distribution mismatch issues impose a strict training ceiling. Furthermore, merely providing the ground-truth label in the context (*Label*) yields poor final performance on tasks such as Physics and Tool use, where performance either stagnates or degrades after a few steps, suggesting its training instability. Moreover, while *Self* and GRPO reach comparable convergence levels, their final performance consistently lags behind SEPO, indicating their exploration bottlenecks prevent them from escaping local optima for further gains. Consequently, by more effectively expanding the exploration boundaries and exploiting the explored paths, SEPO achieves more continuous self-improvement and superior final performance. Additional experimental results, including on-policy GRPO, standard deviations, and fixed training steps, are in Appendix E.

4.3 SEPO overcomes severe exploration bottlenecks

Having demonstrated SEPO's effectiveness on general reasoning tasks, Figure 2(b) further illustrates the 16-hour training processes on Arrow Maze, a highly complex logical reasoning task. In this significantly harder setting, where the initial pass@16 of Qwen3-8B is strictly zero, both GRPO and *Self* fail entirely. Their inability to overcome the exploration bottleneck prevents them from even initiating effective training. Although the *Label* and *Expert* methods manage to trigger training, their performance gains are marginal and plateau after only a few steps. In stark contrast, SEPO exhibits substantial and continuous performance improvements. Its guided exploration successfully resolves the zero-accuracy cold-start problem. By coupling this with the effective exploitation of the explored correct reasoning, SEPO forms a robust self-evolution cycle that stably enhances the model's capabilities. In Appendix E.5, we provide results on another challenging SynLogic task

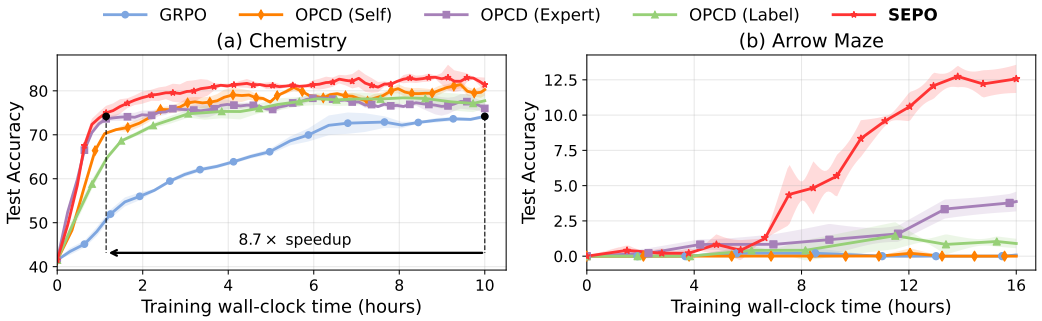

Figure 2: Training process on (a) Chemistry and (b) Arrow Maze tasks using Qwen3-8B. Shaded regions show the standard deviation across 3 runs. SEPO consistently maintains top performance throughout training, achieving effective improvements even on Arrow Maze where baselines fail.

where Qwen3-8B's initial pass@16 is below 20. Although all baseline methods are able to train normally in this regime, SEPO consistently achieves the best performance.

4.4 SEPO achieves scale-invariant self-improvement

Figure 3 presents the peak performance achieved by different methods on the Chemistry task after 10 hours of training, evaluated across Qwen3 model scales ranging from 0.6B to 14B. As observed, SEPO consistently achieves the best performance across all scales. Comparing OPCD baselines against GRPO, a prominent trend emerges from 0.6B to 8B scales: the performance advantage of OPCD widens as the model scale increases. This suggests that the effectiveness of OPCD methods relies on the policy's intrinsic capacity. When scaling down to smaller models (e.g., 0.6B and 1.7B), all OPCD baselines underperform GRPO, indicating models with limited in-context integration capabilities struggle to effectively exploit meaningful signals from self-generated reasoning, heavily distribution-shifted demonstrations or sparse labels. Furthermore, the *Self* method depends not only on model's in-context ability but also on its reasoning exploration capability, resulting in its severely poor performance on small-scale models. In contrast, SEPO maintains a significant performance advantage over GRPO across different model sizes. This highlights a critical synergy in SEPO's design: by resolving the exploration bottleneck, it constructs congruent in-distribution targets with a low assimilation threshold. This enables even lightweight models to smoothly internalize correct reasoning patterns and achieve continuous self-improvement.

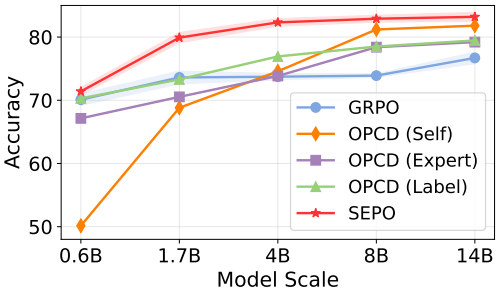

Figure 3: Highest avg@16 on Chemistry after 10h training across Qwen3 scales. SEPO consistently performs best, uniquely outperforming GRPO at 0.6B and 1.7B.

5 Analysis

5.1 SEPO compresses reasoning and maintains exploration diversity

Figure 4 (Left) illustrates the progression of response lengths on Chemistry using Qwen3-8B. While GRPO's lengths increase continuously, OPCD methods maintain or even reduce them. At convergence, SEPO compresses reasoning to be, on average, more than $5\times$ shorter than those of GRPO. Although prevailing RLVR methods often rely on scaling solution lengths to unlock advanced capabilities [4, 24], our results establish that highly concise reasoning can also remain remarkably effective.

To investigate whether this reasoning compression restricts exploration diversity, Figure 4 (Right) plots the best@k performance at the peak avg@16 checkpoint. Here, avg@16 (mathematically equivalent to expected best@1) reflects model's expected abilities, whereas best@16 indicates the capability ceiling driven by exploration diversity. As observed, when $k \geq 4$, all OPCD methods

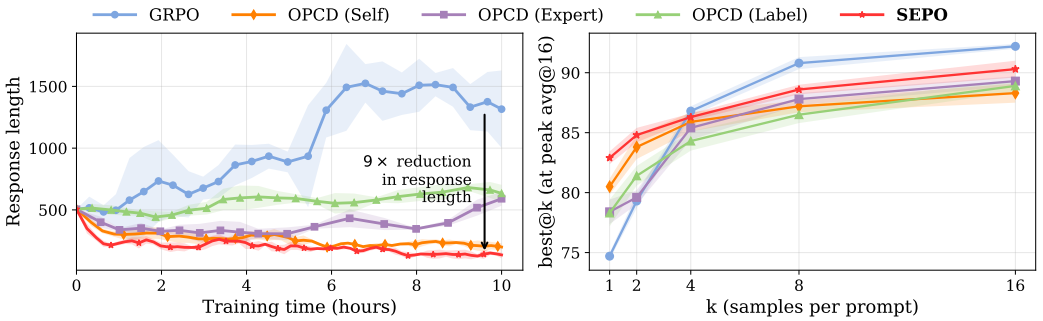

Figure 4: (**Left**) Response length progression. (**Right**) Best@k across different k at the highest avg@16 checkpoint. Both evaluations use Qwen3-8B on Chemistry task. SEPO compresses reasoning while achieving superior best@k, maintaining exploration diversity.

underperform GRPO, revealing a limited sample variance. However, this trade-off is highly favorable for SEPO, which sacrifices merely 2 points in best@16 to secure a massive 10-point advantage in avg@16. Furthermore, SEPO consistently achieves the highest best@k among OPCD methods across all k values, demonstrating its successful preservation of crucial exploration capabilities despite its shortened reasoning. By retaining this diverse solution space, SEPO ensures that each training step continues to yield rich and effective supervision signals, which fundamentally drives its accelerated training efficiency and superior convergence performance.

5.2 SEPO cultivates self-reflective reasoning

Question: What is the molar weight (g/mol) of the molecule with the canonical SMILES representation C1CC1C(=O)NCCC=O?
A: 172.270, B: 141.170, C: 159.180, D: 183.200.

Reasoning: ... a molecule with a six-membered ring containing a nitrogen atom, a carbonyl group... : 144.12 + 10.08 + 32.00 + 14.01 = 200.21. **However, ... recounting gives 7 carbon atoms, 11 hydrogen atom, 1 nitrogen atom, and 2 oxygen atoms.** Thus, the molar mass is 141.168 g/mol, aligning with option B.

Figure 5: An exemplary reasoning process of SEPO on a Chemistry case, showing a natural self-reflective reasoning pattern.

Building on our quantitative observations of reasoning lengths, we qualitatively analyze the underlying reasoning processes of different methods. Appendix Table 11 presents a typical Chemistry case involving molecular weight computation, and Figure 5 visualizes the simplified SEPO's reasoning. All methods employ a common forward-deduction strategy, directly calculating the answer and matching it with the options. However, they diverge significantly in execution. GRPO exhibits inconsistent atom counting, while *Self* and *Expert* derive incorrect molecular formulas. Moreover, GRPO and *Expert* select answers inconsistent with their own calculations, whereas *Self* obtains an incorrect molecular formula whose calculated mass happens to match a distractor. The *Label* baseline attempts to reconsider its answer but repeatedly performs the same incomplete calculation without correcting the error. In contrast, SEPO initially misinterprets the molecular structure but subsequently revisits the SMILES representation, correctly identifies the cyclopropane ring, and revises its atom counts to obtain the correct molecular formula and answer. Notably, the label-guided intrinsic teacher reasoning can also contain meaningful corrective steps, as illustrated in Table 10. Together, these observations suggest that distilling reasoning-rich teacher contexts may help the student internalize self-correction behaviors, beyond merely learning to match final answers

5.3 Ablation study

Ablation on distillation objectives. SEPO introduces critical adjustments to the standard on-policy distillation (OPD) algorithm [8, 22], specifically adopting the JSD objective and restricting the divergence computation to the top-K tokens. Figure 6 presents the 10-hour performance of Qwen3-8B on the Chemistry task across various divergence metrics and K values. For divergence selection, JSD yields the highest performance, demonstrating the advantage of a balanced, symmetric distillation. Furthermore, while traditional OPD favors

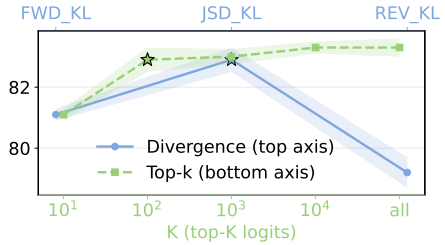

Figure 6: Impact of loss divergence and top-K truncation on 10-hour performance. JSD performs best and top-100 matches full-distribution distillation.

the mode-seeking Reverse KL, the mode-covering Forward KL actually performs better in our framework. Regarding the top-K distillation, we observe that model performance plateaus when $K \geq 100$. Since the long tail of vocabulary is largely uninformative, this top-K truncation effectively circumvents the massive memory overhead associated with storing full logits for the teacher and student, while preserving the distillation efficacy.

Ablation on rollout group size. In Table 1, we maintain a consistent rollout group size G per prompt for fair comparisons. Unlike GRPO and OPCD (*Self*), which rely on sufficiently large rollout pools to discover correct solutions and formulate valid optimization signals, SEPO does not strictly depend on large group sizes. Figure 7 illustrates the peak performance achieved by Qwen3-8B on the Chemistry task at the 1-hour and 10-hour marks under varying group sizes. A clear trade-off emerges: as the rollout size increases, early 1-hour performance decreases while the final 10-hour performance improves, indicating that short-term training efficiency is traded for a higher convergence ceiling. Crucially, regardless of the group size, SEPO consistently outperforms all baseline methods at both time scales, underscoring its robustness.

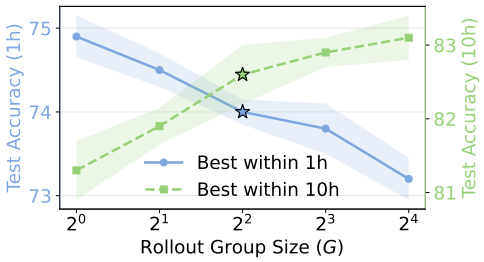

Figure 7: Impact of rollout group size on peak 1-hour and 10-hour performance. Smaller size accelerates training but lower the ceiling, yet any configurations outperform other methods.

Ablation on the teacher-update strategy. We evaluate Qwen3-8B with three teacher-update strategies: fully dynamic (synchronized with the current student after each training-batch update), fixed (frozen at the initial policy), and EMA. Within SEPO, we retain label-guided elicitation and distillation while varying the update strategy; fixed teachers still generate label-conditioned context online. We also evaluate OPCD (Self), the strongest baseline under EMA, with a fixed teacher. Table 2 reports the peak avg@16 achieved within 10 hours on three tasks. The fully dynamic teacher performs substantially worse, suggesting that directly tracking the current policy creates an unstable distillation target. Fixed-teacher SEPO remains effective, and EMA further improves both SEPO and OPCD (Self), supporting the benefit of gradual teacher adaptation. Crucially, SEPO outperforms OPCD (Self) on all three tasks under both fixed and EMA teachers. Thus, its advantage is not attributable solely to a different teacher-update strategy; these controlled comparisons support label-guided, policy-intrinsic context construction as a key driver of its gains.

Table 2: Impact of teacher-update strategy on peak 10-hour performance. EMA performs best, while SEPO outperforms OPCD (Self) under both fixed and EMA teachers.

Method	Chem.	Bio.	Tool Use
SEPO (Dynamic)	47.5	45.2	61.4
OPCD (Self, Fixed)	78.5	64.3	63.7
SEPO (Fixed)	80.6	65.1	66.9
OPCD (Self, EMA)	81.4	67.9	65.7
SEPO (EMA)	**82.9**	**69.0**	**68.2**

6 Related Work

6.1 Reinforcement learning with verifiable rewards

RLVR methods, such as STaR and GRPO [43, 30], are widely adopted in LLM post-training to enhance reasoning capabilities [3, 36]. To overcome the challenges of sparse supervision and exploration bottlenecks [2, 12], recent works leverage external reasoning generated by humans or proprietary models, which fall into two categories. (1) Integrating supervised fine-tuning on expert solutions as an auxiliary loss [44, 7]. (2) Utilizing expert data to guide RL exploration, achieved either by incorporating expert rollouts into the GRPO group for joint learning [7, 39], or by extracting expert prefixes to serve as initial contexts, facilitating the sampling of correct solutions [33, 25]. Unlike these methods that rely on external supervision, SEPO drives simultaneous improvements in training efficiency and final performance entirely through self-evolution distillation.

6.2 On-policy context distillation

Recent on-policy context distillation methods drive reasoning improvement by augmenting the current policy's context to construct a stronger teacher for on-policy distillation [9, 8, 22, 41]. Various contextual augmentations have been explored: some incorporate environmental feedback, whose applicability is restricted to specific tasks [10], while others prompt for simplified teacher responses to compress reasoning [27]. For effective reasoning self-improvement, methods relying on correct student rollouts remain bottlenecked by the model's innate exploration limits, similar to GRPO [10]. Alternatively, injecting external expert reasoning into context suffers from insufficient exploitation of the off-policy solutions [31, 46, 4]. In contrast, the self-explored, in-distribution reasoning within SEPO's context overcomes both the exploration bottleneck and mismatched-distribution exploitation, simultaneously boosting training efficiency and final performance.

7 Conclusion

In this work, we develop Self-Evolution Policy Optimization (SEPO), an on-policy self-distillation framework that drives continuous reasoning evolution through superior exploration and exploitation. By using ground-truth labels as privileged hints, SEPO elicits correct yet strictly in-distribution reasoning from the policy and distills these context-enriched predictions back into the policy. Theoretically, we validate that SEPO mitigates both exploration failures and insufficient exploitation issues, guaranteeing sustained capability evolution. Empirically, experiments across diverse model architectures, scales, and reasoning benchmarks demonstrate SEPO's superior training efficiency and final performance. Notably, in challenging scenarios where existing baselines fail entirely, such as complex logical tasks or lightweight models, SEPO robustly secures continuous improvements.

Acknowledgements

The work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62441230, 62502522, 62072458, 62472429 and 62461146205, and in part by the Outstanding Innovative Talents Cultivation Funded Programs 2024 of Renmin University of China.

References

[1] Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting. *CoRR*, abs/2510.18874, 2025.

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

[3] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

[4] DeepSeek-AI. Deepseek-v3.2: Pushing the frontier of open large language models. *CoRR*, abs/2512.02556, 2025.

[5] Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee F. Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, William Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3. *CoRR*, abs/2512.13961, 2025.

[6] Kehua Feng, Keyan Ding, Weijie Wang, Xiang Zhuang, Zeyuan Wang, Ming Qin, Yu Zhao, Jianhua Yao, Qiang Zhang, and Huajun Chen. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *CoRR*, abs/2406.09098, 2024.

[7] 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. *CoRR*, abs/2506.19767, 2025.

[8] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[9] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[10] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *CoRR*, abs/2601.20802, 2026.

[11] Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In Claude Sammut and Achim G. Hoffmann, editors, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 267–274. Morgan Kaufmann, 2002.

[12] Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *Trans. Mach. Learn. Res.*, 2026, 2026.

[13] Tomasz Korbak, Ethan Perez, and Christopher L. Buckley. RL with KL penalties is better viewed as bayesian inference. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Findings of ACL, pages 1083–1091. Association for Computational Linguistics, 2022.

[14] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[15] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

[16] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[17] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *CoRR*, abs/1805.00909, 2018.

[18] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[19] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theory*, 37(1):145–151, 1991.

[20] Junteng Liu, Yuanxiang Fan, Zhuo Jiang, Han Ding, Yongyi Hu, Chi Zhang, Yiqi Shi, Shitong Weng, Aili Chen, Shiqi Chen, Yunan Huang, Mozhi Zhang, Pengyu Zhao, Junjie Yan, and Junxian He. Synlogic: Synthesizing verifiable reasoning data at scale for learning logical reasoning and beyond. *CoRR*, abs/2505.19641, 2025.

[21] 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.

[22] Kevin Lu and Thinking Machines Lab. On-policy distillation. https://thinkingmachines.ai/blog/on-policy-distillation/, October 2025. Thinking Machines Lab blog. Accessed: 2026-04-16.

[23] Tom Minka et al. Divergence measures and message passing. 2005.

[24] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 20275–20321. Association for Computational Linguistics, 2025.

[25] Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. POPE: learning to reason on hard problems via privileged on-policy exploration. *CoRR*, abs/2601.18779, 2026.

[26] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[27] Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. On-policy self-distillation for reasoning compression. *CoRR*, abs/2603.05433, 2026.

[28] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, JMLR Workshop and Conference Proceedings, pages 1889–1897. JMLR.org, 2015.

[29] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[30] 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. *CoRR*, abs/2402.03300, 2024.

[31] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *CoRR*, abs/2601.19897, 2026.

[32] 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, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM, 2025.

[33] Mingyu Su, Jian Guan, Yuxian Gu, Minlie Huang, and Hongning Wang. Trust-region adaptive policy optimization. *CoRR*, abs/2512.17636, 2025.

[34] Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, and Le Sun. Toolalpaca: Generalized tool learning for language models with 3000 simulated cases. *CoRR*, abs/2306.05301, 2023.

[35] Kimi Team. Kimi K2: open agentic intelligence. *CoRR*, abs/2507.20534, 2025.

[36] Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

[37] Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. HINT: helping ineffective rollouts navigate towards effectiveness. *CoRR*, abs/2510.09388, 2025.

[38] Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. PACED: distillation and self-distillation at the frontier of student competence. *CoRR*, abs/2603.11178, 2026.

[39] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *CoRR*, abs/2504.14945, 2025.

[40] Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient rl framework secretly brings you off-policy rl training, august 2025. *URL https://fengyao. notion. site/off-policy-rl*, 2025.

[41] Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

[42] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025.

[43] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. Star: Self-taught reasoner bootstrapping reasoning with reasoning. In *Proc. the 36th International Conference on Neural Information Processing Systems*, volume 1126, pages 0–55, 2024.

[44] 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. *CoRR*, abs/2508.11408, 2025.

[45] Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-grpo: Scaffolded group relative policy optimization for enhancing LLM reasoning. *CoRR*, abs/2510.19807, 2025.

[46] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *CoRR*, abs/2601.18734, 2026.

A Theoretical Analysis

A.1 Equivalence Between RLVR and OPCD

The Variational Perspective of RLVR Standard RLVR methods optimize a policy π_θ to maximize the expected reward while penalizing the deviation from a reference policy π_{ref}. The regularized RL objective can be formulated as:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{\tau\sim\pi_\theta(\cdot|x)}\left[r(\tau)\right] - \eta\mathbb{D}_{\text{KL}}\left(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\right)\right].$$

where $\eta > 0$ is a temperature parameter controlling the KL penalty. By taking the variational derivative of $\mathcal{J}(\theta)$ with respect to π_θ and setting it to zero, the closed-form optimal policy π^* that maximizes this objective follows an energy-based distribution:

$$\pi^*(\tau \mid x) = \frac{1}{Z(x)}\pi_{\text{ref}}(\tau \mid x)\exp\left(\frac{r(\tau)}{\eta}\right),$$

where $Z(x) = \sum_\tau \pi_{\text{ref}}(\tau \mid x)\exp(r(\tau)/\eta)$ is the intractable partition function [13, 26]. Substituting π^* back into the objective reveals that maximizing $\mathcal{J}(\theta)$ is mathematically equivalent to minimizing the reverse KL divergence between the current policy and the optimal policy:

$$\mathcal{J}(\theta) = \eta\log Z(x) - \eta\mathbb{D}_{\text{KL}}\left(\pi_\theta(\cdot \mid x) \,\|\, \pi^*(\cdot \mid x)\right).$$

Equivalence to the Conditional Success Distribution Crucially, the partition function $Z(x)$ depends only on the reference policy π_{ref} and the reward environment, meaning it is entirely independent of the active policy parameters θ. Consequently, the term $\eta\log Z(x)$ is a constant with respect to optimization. Maximizing the RL objective is mathematically equivalent to minimizing the trajectory-level divergence between the student and the optimal policy:

$$\arg\max_\theta \mathcal{J}(\theta) \equiv \arg\min_\theta \mathbb{D}_{\text{KL}}\left(\pi_\theta(\cdot \mid x) \,\|\, \pi^*(\cdot \mid x)\right).$$

To establish the probabilistic interpretation of π^* in the context of logical reasoning, we specialize the environment to binary outcome rewards ($r \in \{0, 1\}$). Under binary rewards, the exponential term $\exp(r(\tau)/\eta)$ acts as a strict filtering mechanism. As $\eta \to 0^+$, the probability mass of incorrect trajectories decays to zero. Consequently, π^* mathematically converges to the normalized reference policy restricted to the domain of successful trajectories, which equates precisely to the conditional distribution (as grounded in the framework of control as inference [17]):

$$\pi^*(\tau \mid x) \approx \pi_{\text{ref}}(\tau \mid x \mid \text{Success}).$$

Thus, RLVR's true underlying objective is to pull the model toward its own intrinsic distribution of successful reasoning paths.

OPCD as Direct Proxy Rather than relying on sparse rewards to iteratively approach π^*, OPCD explicitly constructs a teacher policy $\pi_{\text{tea}}(\cdot \mid x, c)$ using a privileged context c to simulate this conditional success distribution ($\pi_{\text{tea}} \approx \pi^*$). By substituting π_{tea} for π^*, OPCD minimizes their divergence ($\mathbb{D}_{\text{KL}}(\pi_{\text{tea}} \,\|\, \pi_\theta)$) via dense, token-level supervision. However, the validity of this proxy optimization hinges entirely on c faithfully approximating the natural conditional success distribution, $\pi_{\text{ref}}(\tau \mid x \mid \text{Success})$. If the chosen context induces exploration unavailability, severe distribution shifts from external experts, or non-causal reasoning shortcuts, the theoretical equivalence breaks down, rendering the distilled target invalid.

A.2 Theoretical Limitations of OPCD Baselines

The Exploration Bottleneck of Self-Generated Contexts When utilizing the student's own correct rollouts as the privileged context ($c = \tau_{\text{self}}$), the teacher distribution implicitly models the conditional success distribution. For a given problem x, the ability to construct π_{tea} depends on the marginal probability of sampling a successful solution from the current policy π_θ:

$$P_{\text{self}}(x) = \int \pi_\theta(\tau \mid x)\mathbb{I}(r(\tau) = 1)d\tau.$$

In practice, this is approximated by sampling G rollouts. If the task is sufficiently complex or the initial model is weak, the empirical probability $\hat{P}_{\text{self}}(x) = \frac{1}{G}\sum_{i=1}^{G}\mathbb{I}(r(\tau_i) = 1)$ frequently evaluates

to exactly 0. Mathematically, conditioning on an empty set $\emptyset$ renders the conditional probability $\pi_{\text{tea}}(\cdot \mid x, \tau_{\text{correct}})$ undefined. Consequently, the distillation gradient $\nabla_\theta \mathcal{L}_{\text{CD}}(\theta)$ becomes exactly zero for these problems, completely depriving the model of learning signals and resulting in a severe exploration bottleneck.

Static Ceilings and Exploitation Limitations of Expert Contexts To bypass exploration failures, using expert demonstrations ($c = \tau_{\text{expert}} \sim \pi_{\text{expert}}$) guarantees context availability. However, since these off-policy contexts are static, they cannot co-evolve with the student's optimization trajectory, which inherently imposes a performance ceiling on continuous self-improvement. Even worse, the out-of-distribution (OOD) nature of expert contexts prevents the model from fully exploiting these demonstrations, causing the actual performance to fall short of the theoretical ceiling. Let the token-level distillation objective be defined by a divergence $\mathbb{D}$. Because the expert possesses a fundamentally different latent reasoning distribution, their support sets are frequently disjoint ($\text{supp}(\pi_{\text{tea}}) \not\subseteq \text{supp}(\pi_\theta)$) [15, 18]. This mismatch manifests in two extreme pointwise scenarios: expert-exclusive actions a^* where $\pi_{\text{tea}}(a^* \mid s_t) \to 1$ while $\pi_\theta(a^* \mid s_t) \to 0$, and student-intrinsic actions a where $\pi_\theta(a \mid s_t) > 0$ while $\pi_{\text{tea}}(a \mid s_t) \to 0$. When optimizing across these disjoint supports, the model suffers from extreme penalties that distort its natural behavior:

- **Forward KL (Forced Inclusion):** For expert-exclusive actions a^*, the mode-covering penalty $\pi_{\text{tea}}(a^*) \log \frac{\pi_{\text{tea}}(a^*)}{\pi_\theta(a^*)}$ approaches infinity. To strictly minimize this massive penalty, the optimizer aggressively alters parameters to allocate probability mass to a^*. However, this merely forces the inclusion of isolated tokens without reconstructing the underlying causal reasoning chain.
- **Reverse KL (Zero-Forcing):** For student-intrinsic actions a, the mode-seeking penalty $\pi_\theta(a) \log \frac{\pi_\theta(a)}{\pi_{\text{tea}}(a)}$ also diverges to infinity. The optimizer suppresses the student's natural probability ($\pi_\theta(a) \to 0$), eradicating the model's intrinsic deductive stepping stones.

Ultimately, standard divergence metrics in OOD regions impose extreme point-wise constraints rather than smoothly integrating the expert's trajectory into the student's natural manifold. The student only acquires specific, isolated patterns and fails to completely transition into the expert's coherent deductive mode. This severe under-exploitation, compounded by static context, cements a strict performance ceiling.

A.3 SEPO's Continuous Improvement

In this section, we provide the rigorous theoretical proof for the accuracy lower bound of the student policy during the Self-Evolution Policy Optimization (SEPO) process.

Let $x \sim \mathcal{D}$ be the problem sampled from the dataset, and let $Y^*(x)$ denote the set of all valid reasoning trajectories or correct final answers for x. The expected accuracy of a policy π is defined as the expected probability mass it assigns to the correct solutions:$\mathcal{A}(\pi) = \mathbb{E}_{x\sim\mathcal{D}}\left[\sum_{y\in Y^*(x)} \pi(y \mid x)\right]$.During the distillation phase at step n, the student policy is updated from π_θ^n to π_θ^{n+1} by minimizing the Jensen-Shannon Divergence (JSD) against the teacher π_{tea}^n. We define the residual exploitation error ϵ_{JSD} as the expected JSD remaining after the update:$\epsilon_{\text{JSD}} = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\text{JSD}}\left(\pi_{\text{tea}}^n(\cdot \mid x) \,\|\, \pi_\theta^{n+1}(\cdot \mid x)\right)\right]$.

Proposition 1 (Monotonic Improvement Bound). Assuming the residual exploitation error is ϵ_{JSD}, the expected accuracy of the updated student policy π_θ^{n+1} is strictly bounded by:

$$\mathcal{A}(\pi_\theta^{n+1}) \geq \mathcal{A}(\pi_\theta^n) + \Delta^n - \mathcal{O}(\sqrt{\epsilon_{\text{JSD}}})$$

where $\Delta^n = \mathcal{A}(\pi_{\text{tea}}^n) - \mathcal{A}(\pi_\theta^n)$ is the capability advantage (exploration gap) provided by the teacher. The proof proceeds in three steps:

Step 1: Bounding Accuracy Difference via Total Variation Distance. For any given problem x, the difference in the probability of generating a correct answer between the teacher π_{tea}^n and the updated student π_θ^{n+1} is bounded by the Total Variation distance between their respective distributions. By the definition of TV distance, for any event A (including the event of correctness $Y^*(x)$):

$$|\pi_{\text{tea}}^n(Y^*(x) \mid x) - \pi_\theta^{n+1}(Y^*(x) \mid x)| \leq \mathbb{D}_{\text{TV}}(\pi_{\text{tea}}^n(\cdot \mid x), \pi_\theta^{n+1}(\cdot \mid x))$$

Taking the expectation over the data distribution $x \sim \mathcal{D}$:

$$|\mathcal{A}(\pi_{\text{tea}}^n) - \mathcal{A}(\pi_\theta^{n+1})| \leq \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\text{TV}}(\pi_{\text{tea}}^n(\cdot \mid x), \pi_\theta^{n+1}(\cdot \mid x))\right]$$

Removing the absolute value yields the lower bound for the student's accuracy:

$$\mathcal{A}(\pi_\theta^{n+1}) \geq \mathcal{A}(\pi_{\text{tea}}^n) - \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\text{TV}}(\pi_{\text{tea}}^n(\cdot \mid x), \pi_\theta^{n+1}(\cdot \mid x))\right]$$

Step 2: Relating Total Variation Distance to Jensen-Shannon Divergence To connect the empirical distillation objective to the accuracy bound, we utilize the information-theoretic relationship between TV distance and JSD [19] . For any two probability distributions P and Q:

$$\mathbb{D}_{\text{TV}}(P, Q) \leq \sqrt{2\ln 2 \cdot \mathbb{D}_{\text{JSD}}(P \,\|\, Q)}$$

(Note: If the base of the logarithm in JSD is 2, the constant $2\ln 2$ is replaced by 2. For generality, we represent this constant scaling as $\mathcal{O}(\cdot)$).Applying this inequality to our policies for a specific x:

$$\mathbb{D}_{\text{TV}}(\pi_{\text{tea}}^n(\cdot \mid x), \pi_\theta^{n+1}(\cdot \mid x)) \leq \sqrt{2\ln 2 \cdot \mathbb{D}_{\text{JSD}}(\pi_{\text{tea}}^n(\cdot \mid x) \,\|\, \pi_\theta^{n+1}(\cdot \mid x))}$$

Taking the expectation over $\mathcal{D}$ and applying Jensen's Inequality (since $f(u) = \sqrt{u}$ is a concave function):

$$\mathbb{E}_{x\sim\mathcal{D}}[\mathbb{D}_{\text{TV}}] \leq \mathbb{E}_{x\sim\mathcal{D}}\left[\sqrt{2\ln 2 \cdot \mathbb{D}_{\text{JSD}}}\right] \leq \sqrt{2\ln 2 \cdot \mathbb{E}_{x\sim\mathcal{D}}[\mathbb{D}_{\text{JSD}}]}$$

Substituting our definition of ϵ_{JSD}:

$$\mathbb{E}_{x\sim\mathcal{D}}[\mathbb{D}_{\text{TV}}] \leq \sqrt{2\ln 2 \cdot \epsilon_{\text{JSD}}} = \mathcal{O}(\sqrt{\epsilon_{\text{JSD}}})$$

Step 3: Synthesizing the Final Recursive Bound. Substituting the bound from Step 2 into the inequality from Step 1, we obtain the worst-case performance penalty induced by the exploitation error:

$$\mathcal{A}(\pi_\theta^{n+1}) \geq \mathcal{A}(\pi_{\text{tea}}^n) - \mathcal{O}(\sqrt{\epsilon_{\text{JSD}}})$$

By definition, the exploration gap at step n is $\Delta^n = \mathcal{A}(\pi_{\text{tea}}^n) - \mathcal{A}(\pi_\theta^n)$, which implies $\mathcal{A}(\pi_{\text{tea}}^n) = \mathcal{A}(\pi_\theta^n) + \Delta^n$. Substituting this into the inequality, we arrive at the final recursive bound:

$$\mathcal{A}(\pi_\theta^{n+1}) \geq \mathcal{A}(\pi_\theta^n) + \Delta^n - \mathcal{O}(\sqrt{\epsilon_{\text{JSD}}})$$

B Limitations

B.1 Expert Selection in Baselines

To ensure a rigorous evaluation of the OPCD (*Expert*) baseline, we evaluate three highly capable models: GPT-OSS-120B, Qwen3-235B, and DeepSeek-V3.2, to generate expert reasoning trajectories. We ultimately select DeepSeek-V3.2 as the external expert, as it yields the highest self-distillation performance. In our comparative analysis, OPCD (*Expert*) exhibits rapid initial training efficiency but suffers from a lower final performance ceiling. Nevertheless, we acknowledge that employing an even stronger future expert, or an expert whose generative distribution is more intrinsically aligned with the student model's preferences, might significantly alter the performance dynamics and upper bounds of this baseline.

B.2 Computational Overhead of the Elicitation Pipeline

Compared to the standard on-policy training loops of baseline methods, the SEPO pipeline introduces an additional generation step: it requires the teacher to conditionally elicit the reasoning process based on the ground-truth label. Although SEPO ultimately achieves superior overall training efficiency and convergence performance despite this added step, exploring architectural or procedural optimizations to further streamline this pipeline remains an important direction for future work.

B.3 Evaluation at Larger Model Scales

Our experiments evaluate models up to 14B parameters. Although SEPO performs favorably across the evaluated scales, whether these advantages persist for larger models, such as 32B or 72B, remains unverified. Evaluating SEPO at these scales is an important direction for future work.

Algorithm 2 SEPO with label-conditioned elicitation and outcome-filtered context

Input: Initial parameters θ_0; dataset $\mathcal{D}$; task verifier R; training steps N; batch size B; student samples G; teacher candidates H; threshold $r_{\min}$; EMA rate β; elicitation template E; context template C

$\theta \leftarrow \theta_0$; $\theta_{\text{tea}} \leftarrow \theta_0$
for $n = 0, \ldots, N-1$ **do**
  Sample $\{(x_i, y_i^*)\}_{i=1}^{B} \sim \mathcal{D}$; set $\mathcal{P} \leftarrow \varnothing$
  for $i = 1, \ldots, B$ **do**
    Sample $\tau_{ig} \sim \pi_\theta(\cdot \mid x_i)$ for $g = 1, \ldots, G$
    Sample $\bar{\tau}_{ih} = (\bar{z}_{ih}, \hat{y}_{ih}) \sim \pi_{\theta_{\text{tea}}}(\cdot \mid E(x_i, y_i^*))$ for $h = 1, \ldots, H$
    $\mathcal{Z}_i \leftarrow \{\bar{z}_{ih} : R(x_i, \bar{\tau}_{ih}, y_i^*) \geq r_{\min}\}$
    if $\mathcal{Z}_i \neq \varnothing$ **then**
      for $g = 1, \ldots, G$ **do**
        Draw $z_{ig}^{\text{intrinsic}} \sim \text{Uniform}(\mathcal{Z}_i)$
        Add $(x_i, \tau_{ig}, z_{ig}^{\text{intrinsic}})$ to $\mathcal{P}$
      end for
    end if
  end for
  for each optimization mini-batch $\mathcal{M} \subseteq \mathcal{P}$ **do**
    for each $(x, \tau, z) \in \mathcal{M}$ and valid response position t **do**
      $p_t \leftarrow \pi_\theta(\cdot \mid x, \tau_{<t})$
      $q_t \leftarrow \text{stopgrad}(\pi_{\theta_{\text{tea}}}(\cdot \mid C(x, z), \tau_{<t}))$
      $S_t \leftarrow \text{TopK}(p_t, K)$
      $\bar{p}_t \leftarrow (p_t(v))_{v \in S_t} \parallel \left(1 - \sum_{v \in S_t} p_t(v)\right)$
      $\bar{q}_t \leftarrow (q_t(v))_{v \in S_t} \parallel \left(1 - \sum_{v \in S_t} q_t(v)\right)$
      $m_t \leftarrow (\bar{p}_t + \bar{q}_t)/2$
      $d_t \leftarrow \frac{1}{2} D_{\text{KL}}(\bar{q}_t \| m_t) + \frac{1}{2} D_{\text{KL}}(\bar{p}_t \| m_t)$
    end for
    $\mathcal{L}_{\mathcal{M}} \leftarrow \dfrac{1}{|\mathcal{M}|} \displaystyle\sum_{j=1}^{|\mathcal{M}|} \frac{1}{|\mathcal{T}_j|} \sum_{t \in \mathcal{T}_j} d_{j,t}$
    $\theta \leftarrow \text{OptimizerStep}(\theta, \nabla_\theta \mathcal{L}_{\mathcal{M}})$
  end for
  if a student optimizer update was performed **then**
    $\theta_{\text{tea}} \leftarrow (1 - \beta)\theta_{\text{tea}} + \beta\theta$
  end if
end for
return π_θ

C Broader Impacts

Large Language Models (LLMs) are increasingly deployed across diverse industries, where robust in-domain reasoning capabilities are critical. However, conventional RLVR methods frequently suffer from sparse supervision signals and inherent exploration bottlenecks. Our method, SEPO, offers an efficient and effective self-distillation framework to reliably enhance these reasoning capabilities. This can significantly accelerate the development of highly capable, domain-specific AI systems while democratizing access by reducing reliance on expensive external supervision.

Despite these positive impacts, SEPO also presents potential risks. Because it enables autonomous self-improvement, malicious actors could exploit this framework to accelerate model capabilities in harmful or illicit domains (e.g., generating malicious code or synthesizing dangerous chemical compounds). However, this risk is not unique to SEPO; rather, it is a broader systemic challenge inherent to all algorithms designed for generalized capability enhancement.

D Implementation Details

D.1 SEPO's Implementations

Algorithm 2 expands the high-level training loop in Algorithm 1. For every question in each batch, the student generates $G = 8$ unhinted rollouts, while the EMA teacher generates $H = 4$ candidates

using a task-specific label-informed prompt. Teacher elicitation is performed for all questions, rather than only when the student fails. We retain candidates whose final answers pass the task verifier with reward at least $r_{\min} = 1$. This outcome-based filter does not certify the correctness of intermediate reasoning. If no candidate is retained for a question, all its student rollouts receive zero distillation weight for that batch; no additional resampling is required. Otherwise, each student rollout is independently paired with a uniformly sampled retained candidate from the same question's pool, with replacement. We use JSD with mixing coefficient $\lambda = 0.5$. At each response position, the student selects the top-K token indices ($K = 100$), and both distributions are evaluated on this shared support. Each distribution's remaining probability mass is aggregated into one additional tail category. To stabilize updates against distribution shifts, we apply a token-level importance sampling correction, clipping the importance-sampling ratios at 2.0. Finally, the teacher model is updated via an Exponential Moving Average (EMA) of the student's weights with an update rate of 0.05.

We implement our training pipeline on top of the verl reinforcement learning framework [32], utilizing PyTorch Fully Sharded Data Parallel (FSDP2) distributed across 8 H20 GPUs. During the rollout phase, which is accelerated by vLLM [16], we sample $G = 8$ student rollouts per prompt at a temperature of 1.0. To accommodate extended reasoning traces, we set a maximum prompt length of 2048 tokens and a maximum response length of 22,528 tokens. The policy is optimized using AdamW with a peak learning rate of 1×10^{-5}, zero weight decay, and a linear warmup over the first 10 steps. Each training step processes a global batch of 32 prompts, which is partitioned into mini-batches of 32 for policy optimization.

D.2 Enhanced GRPO Implementation

Following [10], we implement an enhanced GRPO baseline that integrates several recent algorithmic improvements. These include asymmetric clipping [42], the avoidance of biased normalization [21], and off-policy data correction [40].

D.3 Baseline Implementations

The implementation of OPCD (*Self*) follows the methodology of [10]. To construct the self-teacher, each rollout is conditioned on a distinct successful rollout randomly selected from the same group. For OPCD (*Expert*), we adapt the frameworks established in [31, 46]. To construct the expert dataset, we sample 8 reasoning trajectories per problem and randomly retain one successful rollout to serve as the context. We try three highly capable reasoning models to generate these expert trajectories: GPT-OSS-120B, Qwen3-235B, and DeepSeek-V3.2. We ultimately select DeepSeek-V3.2 as our primary external expert, as it yields the highest self-distillation performance. The OPCD (*Label*) baseline serves as an ablated variant of the SEPO framework, where the label-guided deduction mechanism is explicitly removed (i.e., the teacher model is merely conditioned on the label, but is not prompted to perform guided deduction toward it). Following an extensive hyperparameter search, the optimal hyperparameters for all OPCD baselines align with those established for SEPO. For both the on-policy and off-policy variants of GRPO, the optimal learning rate is 1×10^{-5}. In addition, for a controlled comparison, SEPO and all three OPCD baselines use an EMA-updated teacher initialized from the same initial student weights.

D.4 Meta-Prompts

We detail the specific meta-prompts utilized within the SEPO framework, using the Science QA task as a representative example. Table 3 presents the standard base prompt used during student rollouts. Table 4 displays the prompt used to elicit label-conditioned reasoning from the teacher. Table 5 outlines the prompt employed during the teacher distillation phase.

E Additional Experimental Results

E.1 Off-Policy vs. On-Policy GRPO

In our primary evaluations, we report the off-policy variant of GRPO as the standard GRPO baseline. Table 6 details a comprehensive performance comparison between the off-policy and on-policy GRPO

Table 3: Base Meta-Prompt utilized for standard student rollouts.

Given a question and four options, please select the right answer. Respond in the following format: <reasoning> ... </reasoning> <answer> ... </answer> For the answer, only output the letter corresponding to the correct option (A, B, C, or D), and nothing else. Do not restate the answer text. For example, if the answer is "A", just output: <answer>A</answer> {Question} Please reason step by step.

Table 4: Label-Conditioned Reasoning Prompt utilized for teacher exploration.

{Base Meta-Prompt} (Correct answer: {label}. Reason as you naturally would without this hint — arrive at the correct answer through genuine understanding, not by working backward from it.)

variants across various scenarios, utilizing identical experimental settings to those in Table 1. As observed, the off-policy variant achieves superior performance in the majority of cases.

E.2 Results under Fixed Training Steps

In Section 4.2 and Table 1, we report the peak avg@16 performance achieved by various methods under wall-clock time constraints (1 hour and 10 hours). To complement these findings, Table 7 details the highest avg@16 performance attained under fixed budgets of 50 and 400 training steps. As observed, the comparative trends closely mirror those in Table 1. SEPO consistently secures the highest performance across both limited and extended training steps, further validating its superior training efficiency and higher final convergence ceiling.

E.3 Standard Deviations Across Independent Runs

In Table 1, we report the peak performance of various methods across different scenarios, averaged over three independent runs with different random seeds. To provide a comprehensive view of training stability, Table 8 details the corresponding standard deviations. As observed, the standard deviations for all methods fall consistently within a tight range of 0.2 to 1.2. This indicates that the observed variances primarily reflect natural experimental fluctuations introduced by stochastic trajectory sampling, rather than underlying algorithmic instability. Consequently, this confirms that the performance gains achieved by SEPO are highly robust and represent stable convergence, independent of specific random seed initializations.

E.4 Thinking Mode

In Table 1, we report the model's performance in the standard Instruct (non-thinking) mode Table 9 extends this evaluation to the Chemistry and Physics tasks under a "Thinking Mode" setting using Qwen3-8B. As observed, SEPO remains highly effective in this extended reasoning regime, consistently achieving the best performance at both the 1-hour and 10-hour training milestones.

E.5 Results on the Dyck Language Task

In Section 4.3, we demonstrated SEPO's unique ability to initiate training on the Arrow Maze task, an extreme scenario where standard exploration completely fails due to a strict zero initial pass rate. To provide a comprehensive evaluation, Table 9 details the performance on Dyck Language, another highly complex SynLogic task. Unlike Arrow Maze, Qwen3-8B exhibits a non-zero initial avg@16 (12.5) on the Dyck Language task. This minimal baseline competence allows all standard methods, including GRPO and *Self*, to successfully bypass the exploration bottleneck and train

Table 5: Contextual Distillation Prompt utilized during the self-distillation phase.

{Base Meta-Prompt} Here is a reference solution: {solution} Now, solve the problem on your own.

Table 6: Supplementary performance comparison between the off-policy and on-policy GRPO variants, extending the results from Table 1.

	Chemistry		Biology		Physics		Material		Tool use	
	1h	10h	1h	10h	1h	10h	1h	10h	1h	10h
Qwen3-8B	41.5		30.5		58.6		59.8		57.1	
+ GRPO (off-policy)	48.0	73.9	30.5	68.1	59.5	74.8	70.9	76.1	60.9	67.6
+ GRPO (on-policy)	42.5	70.5	32.4	58.0	60.5	70.2	69.3	78.1	59.3	66.5
Olmo3-7B-Instruct	25.1		13.0		37.5		37.6		45.3	
+ GRPO (off-policy)	25.4	54.7	26.0	50.6	40.5	53.2	61.1	73.9	52.7	63.4
+ GRPO (on-policy)	25.9	71.8	27.5	41.6	39.5	51.6	55.1	73.8	52.1	63.4

normally. However, despite this alleviated learning threshold, SEPO still demonstrates superior sample efficiency at 1 hour and achieves the highest final convergence at 10 hours. This confirms that SEPO consistently provides higher-quality optimization signals to maximize performance, regardless of the exploration difficulty.

F Case Study

F.1 Example of Intrinsic Teacher Context

To illustrate the privileged context $z_{\text{intrinsic}}$, Table 10 presents a label-conditioned teacher reasoning trace for a Chemistry question about hydrogen-bond acceptors. The teacher initially identifies three candidate acceptors, including both nitrogen atoms and the carbonyl oxygen. Upon reconsideration, it correctly excludes the terminal amide nitrogen, whose lone pair is delocalized into the adjacent carbonyl group, and identifies the non-amide nitrogen and carbonyl oxygen as the two acceptors. This example illustrates that $z_{\text{intrinsic}}$ can provide meaningful intermediate reasoning and explicit corrective steps, offering richer contextual guidance than the ground-truth label alone.

F.2 Comparison of Student Reasoning Trajectories

Appendix Table 11 compares representative reasoning trajectories generated by different student policies after 100 training steps on a Chemistry problem involving molecular weight calculation. GRPO initially recognizes the cyclopropane ring but subsequently produces inconsistent atom counts, leading to an incorrect answer. *Self* and *Expert* also derive incorrect molecular formulas, with *Self* obtaining a mass that happens to match an incorrect option and *Expert* selecting an option inconsistent with its own calculation. The *Label* baseline attempts to reconsider its answer but repeatedly performs the same incomplete calculation without correcting the underlying error. In contrast, SEPO initially misinterprets the molecular structure as containing a six-membered ring, but subsequently revisits the SMILES representation and correctly identifies the three-membered cyclopropane ring. It then revises the atom counts to obtain the correct molecular formula, C7H11NO2, yielding a molecular weight of 141.168 g/mol and selecting the correct answer. This case illustrates that SEPO can exhibit meaningful structural correction within its reasoning process, rather than merely matching the final answer.

Table 7: Performance comparison on science QA and tool-use benchmarks, reporting the highest avg@16 achieved within 50 and 400 training steps. Bold and underlined values denote the best and second-best performance, respectively.

	Chemistry		Biology		Physics		Material		Tool use	
	50	400	50	400	50	400	50	400	50	400
Qwen3-8B	41.5		30.5		58.6		59.8		57.1	
+ GRPO (off-policy)	63.2	73.6	51.7	<u>68.1</u>	67.5	76.1	72.9	77.1	60.9	<u>68.0</u>
+ OPCD (*Self*)	76.1	<u>81.7</u>	<u>55.0</u>	67.9	68.2	**77.0**	70.1	74.5	63.4	65.7
+ OPCD (*Expert*)	<u>76.2</u>	78.3	49.2	57.4	<u>68.3</u>	73.8	73.0	78.1	<u>63.7</u>	64.6
+ OPCD (*Label*)	74.2	78.9	48.2	66.9	67.3	68.5	**75.7**	<u>77.8</u>	62.5	62.5
+ **SEPO**	**76.5**	**82.9**	**58.1**	**69.0**	**68.8**	<u>76.4</u>	<u>73.8</u>	**78.4**	**64.4**	**68.2**
Olmo3-7B-Instruct	25.1		13.0		37.5		37.6		45.3	
+ GRPO (off-policy)	47.6	57.2	38.9	53.6	55.9	68.3	73.7	**79.8**	57.5	<u>63.4</u>
+ OPCD (*Self*)	<u>74.0</u>	<u>80.9</u>	50.5	50.5	<u>66.9</u>	<u>70.2</u>	74.3	77.9	<u>60.7</u>	61.5
+ OPCD (*Expert*)	64.7	64.7	<u>52.0</u>	<u>56.9</u>	64.3	69.8	<u>75.1</u>	76.7	54.6	54.6
+ OPCD (*Label*)	44.3	54.3	46.5	46.5	49.1	49.1	66.0	66.0	60.0	60.0
+ SEPO	**77.4**	**82.1**	<u>56.0</u>	**57.4**	**70.1**	**73.8**	**75.7**	<u>79.1</u>	**62.8**	**63.8**

Table 8: Standard deviations across three independent runs corresponding to the main results in Table 1.

	Chemistry		Biology		Physics		Material		Tool use	
	1h	10h	1h	10h	1h	10h	1h	10h	1h	10h
Qwen3-8B	–		–		–		–		–	
+ GRPO (off-policy)	0.42	0.27	1.05	0.41	0.82	0.44	0.96	1.05	0.85	0.49
+ OPCD (*Self*)	0.31	0.67	0.92	0.55	1.01	0.53	0.88	0.42	0.62	0.94
+ OPCD (*Expert*)	0.56	1.02	0.45	0.88	0.94	0.47	1.12	0.55	1.08	0.51
+ OPCD (*Label*)	0.48	1.13	0.68	0.62	0.55	0.92	0.76	0.39	0.93	0.45
+ **SEPO**	0.35	0.53	0.58	0.37	0.33	0.72	0.42	0.91	0.65	0.36
Olmo3-7B-Instruct	–		–		–		–		–	
+ GRPO (off-policy)	0.94	0.58	0.77	0.92	1.08	0.65	1.12	0.57	0.88	0.46
+ OPCD (*Self*)	0.95	0.62	0.73	0.58	0.55	0.87	0.93	0.48	0.95	0.52
+ OPCD (*Expert*)	0.88	0.49	0.89	0.44	0.96	0.52	0.74	1.05	1.16	0.64
+ OPCD (*Label*)	1.12	0.55	1.02	0.78	0.83	0.63	1.01	0.59	0.42	0.98
+ **SEPO**	0.52	0.37	0.41	0.83	0.48	0.45	0.39	0.51	0.57	0.68

Table 9: Performance comparison when models are in thinking mode on science tasks, alongside results on the Dyck Language logical reasoning task.

	Chemistry (Thinking)		Physics(Thinking)		Dyck Language	
	1h	10h	1h	10h	1h	10h
Qwen3-8B	56.3		70.2		12.5	
+ GRPO (off-policy)	57.3	62.5	71.3	72.6	15.0	73.8
+ OPCD (*Self*)	58.3	<u>78.3</u>	**72.2**	<u>76.3</u>	<u>19.8</u>	<u>96.7</u>
+ OPCD (*Expert*)	<u>59.8</u>	65.8	71.4	75.7	18.8	86.9
+ OPCD (*Label*)	57.2	63.4	70.5	74.3	14.3	64.4
+ **SEPO**	<u>60.2</u>	**79.1**	<u>71.9</u>	**79.2**	**20.3**	**98.1**

Table 10: A complete retained intrinsic teacher trace for the Chemistry question.

Question

How many hydrogen bond acceptors are there in the molecule with the canonical SMILES representation 'C=CNCC(=O)N'?
A: 2 B: 8 C: 6 D: 9 (answer: A=2)

Complete teacher reasoning

To determine the number of hydrogen bond acceptors in the molecule represented by the SMILES string 'C=CNCC(=O)N', we need to identify all atoms that can accept hydrogen bonds.
Hydrogen bond acceptors are typically atoms that have lone pairs of electrons and are capable of forming hydrogen bonds with hydrogen atoms. These include: Oxygen (O), Nitrogen (N), Fluorine (F), Sulfur (S)
In the given molecule, the atoms of interest are: 1. The nitrogen in the 'C=CN' group: This is a nitrogen atom in a double bond and has a lone pair, making it a hydrogen bond acceptor. 2. The oxygen in the 'C(=O)N' group: This is an oxygen double-bonded to carbon and has two lone pairs, making it a hydrogen bond acceptor. 3. The nitrogen in the 'N' at the end: This is a terminal nitrogen atom that has a lone pair and acts as a hydrogen bond acceptor.
No other atoms in the molecule are capable of hydrogen bonding, as the other carbon and hydrogen atoms cannot act as hydrogen bond acceptors. Thus, there are a total of 3 hydrogen bond acceptors in the molecule: 1 nitrogen, 1 oxygen, and 1 nitrogen.
However, the terminal nitrogen (the last N in C(=O)N) is not a hydrogen-bond acceptor in this context. Let's re-identify the atoms: the non-amide nitrogen in the C=CN group has an available lone pair and can act as a hydrogen-bond acceptor; the oxygen in the C(=O)N group is a double-bonded oxygen with lone pairs and can act as a hydrogen bond acceptor; the carbon and hydrogen atoms can not act as hydrogen bond acceptors.
Thus, there are a total of 2 hydrogen bond acceptors: one nitrogen and one oxygen.

Table 11: Responses from different methods after 100 training steps to the question: "What is the molar weight (g/mol) of the molecule with the canonical SMILES representation C1CC1C(=O)NCCC=O? A: 172.270, B: 141.170, C: 159.180, D: 183.200." The correct answer is B. Red marks erroneous structural claims, calculations, or an incorrect final answer. Blue marks superficial mechanical reasoning in GRPO, and repeated reflection in Label when the derived answer is inconsistent with the options. Green denotes spontaneous and corrective reflection in SEPO.

Method	Output
GRPO	To calculate the molar mass of a molecule given its SMILES string 'C1CC1C(=O)NCCC=O', we need to determine the molecular formula and sum the atomic masses: 1. Decompose the SMILES string: - C1CC1: This is a cyclopropane ring (C3H6). - C(=O): This is a carbonyl group (C=O). - N: A single nitrogen atom. - CCC=O: This is an ethyl group with a double bond to oxygen (C2H5O). 2. Break it down into atoms: Carbon (C): There are 5 carbon atoms in the ring (C1CC1) + 1 from C(=O) + 2 from CCC=O = 8 C atoms. Hydrogen (H): There are 6 H atoms in the ring (C1CC1) + 2 H atoms in CCC=O = 8 H atoms. Oxygen (O): There is 1 O from C(=O) and 1 O from CCC=O = 2 O atoms. Nitrogen (N): 1 atom. 3. Atomic masses (g/mol):- Carbon (C) = 12.011. - Hydrogen (H) = 1.008. - Oxygen (O) = 16.00.- Nitrogen (N) = 14.007. 4. Calculate molar mass: $(8 \times 12.011) + (8 \times 1.008) + (2 \times 16.00) + (1 \times 14.007) = 96.088 + 8.064 + 32.00 + 14.007 = 150.159$ g/mol (rounded to 3 decimal places). \n Thus, the correct answer is C: 159.180.
Self	The SMILES representation 'C1CC1C(=O)NCCC=O' creates a molecule that contains: A five-membered ring with two carbon atoms, one nitrogen, and one oxygen atom (C2N1O1), A side chain of three carbon atoms, one hydrogen atom, and two oxygen atoms (C3H1O2), A carbonyl group and a nitrogen group, thus summing up the full molecular composition. After combining these, the molecular formula is correctly determined as $C_6H_{11}N_2O_3$, reflecting the accurate count of atoms and their sources. Here are the calculations: Carbon (C): 6 × 12.01 = 72.06 g/mol; Hydrogen (H): 11 × 1.008 = 11.09 g/mol; Nitrogen (N): 2 × 14.01 = 28.02 g/mol; Oxygen (O): 3 × 16.00 = 48.00 g/mol. Total molar mass = 72.06 + 11.09 + 28.02 + 48.00 = 159.17 g/mol. This matches Option C (159.180 g/mol), which is the only option that closely aligns with the calculated value and corresponds to the correct molecular formula. Hence, Option C is the most accurate answer.
Expert	The SMILES string 'C1CC1C(=O)NCCC=O' represents a molecule with a cyclobutane ring and a side chain. By breaking down the structure, we determine the molecular formula to be C8H12NO2. Calculating the molar mass: Carbon: 8 carbons * 12.01 g/mol = 96.08 g/mol, Hydrogen: 12 hydrogens * 1.008 g/mol = 12.096 g/mol, Nitrogen: 1 nitrogen * 14.01 g/mol = 14.01 g/mol, Oxygen: 2 oxygens * 16.00 g/mol = 32.00 g/mol. Total molar mass = 96.08 + 12.096 + 14.01 + 32.00 = 154.186 g/mol. After careful analysis, the molecular formula calculation results in a molar mass consistent with the option C: 159.18 g/mol.
Label	Let's break down the SMILES string and identify the atoms and bonds in the molecule. - 'C1CC1': This forms a cyclopropane ring with two carbon atoms connected to the ring. - 'C(=O)': This is a carbonyl group, indicating a carbon double-bonded to an oxygen. - 'N': This is a nitrogen atom. - 'CCC=O': This is a chain of three carbon atoms ending in a carbonyl group (C=O). So the molecule is a cyclopropane ring (C3), with a carbonyl group (C=O), a nitrogen (N), and a chain of three carbons ending in a carbonyl group. Now, we count each atom and calculate the total molecular weight based on standard atomic weights: Carbon: 8 × 12.011 = 96.088 g/mol, Oxygen: 2 × 16.00 = 32.00 g/mol, Nitrogen: 1 × 14.007 = 14.007 g/mol. Total molecular weight: 96.088 + 32.00 + 14.007 = 142.095 g/mol But wait—this calculation assumes the structure is that simple. [...] Now, repeating the molecular weight calculation: [...] Total molecular weight: 96.088 + 32.00 + 14.007 = 142.095 g/mol [Wait—this still doesn't match the correct answer. Let's go back and accurately analyze the SMILES again.] *4 Adding gives: 96.088 + 32.00 + 14.007 = 142.095 g/mol. So the correct answer is C.
SEPO	The SMILES string 'C1CC1C(=O)NCCC=O' describes a molecule with a six-membered ring containing a nitrogen atom, a carbonyl group, and a terminal carbonyl group. A careful count of the atoms gives:- 12 carbon atoms: the ring has 5 carbons, and the remaining 7 carbons from the side chain and other groups. - 10 hydrogen atoms: the hydrogens are located on the ring and the substituents. - 1 nitrogen atom: part of the ring structure. - 2 oxygen atoms: one from the carbonyl group and another from the amide group. Using the standard atomic weights: Carbon: $12 \times 12.01 = 144.12$ g/mol, Hydrogen: $10 \times 1.008 = 10.08$ g/mol, Oxygen: $2 \times 16.00 = 32.00$ g/mol, Nitrogen: $1 \times 14.01 = 14.01$ g/mol. Adding these: $144.12+10.08+32.00+14.01 = 200.21$ g/mol However, after revisiting the SMILES, C1CC1 is a three-membered cyclopropane ring. Recounting gives 7 carbon atoms, 11 hydrogen atoms, 1 nitrogen atom, and 2 oxygen atoms. Thus, the molar mass is 141.168 g/mol, aligning with option B.

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: The abstract and introduction accurately reflect our contributions and scope.

   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: We clearly discuss the limitations on our work in Appendix B.

   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: We provide the full set of assumptions and a complete proof in Section 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: We fully disclose the datasets and models used, which are publicly accessible. We include all discussions about the implementation details in Appendix D.

   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 provide the data, code and sufficient instructions in our supplemental material.

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: We discuss all the details in Section 4.1 and Appendix D.

   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: [Yes]

   Justification: All reported results are averaged over three independent runs. We provide the standard deviations in Table 8, and the shaded regions in all experimental figures denote standard deviations.

   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: For each experiment, we specify the runtime or number of steps, as well as the exact experimental setup.

   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: Our research 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: In Appendix C, we discuss the potential positive and negative societal impacts of our work.

    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 paper does not release new models or scraped datasets but rather derive from existing models and datasets. Therefore, we do not anticipate such risks.

    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 have properly credited all the code sources, data sources, and open-source models in our paper following their license and terms.

    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: [N/A]

    Justification: The paper does not release new assets. We entirely use assets that have already been made available.

    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: Our paper does not involve crowdsourcing nor 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: Our paper does not involve crowdsourcing nor research with human subjects.

    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: In Section 4.1 and Appendix D, we describe in detail how LLM are used in certain baseline methods.

    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.