Reinforcement Learning from Rich Feedback with Distributional DAgger

Rishabh Agrawal **Jacob Fein-Ashley** **Paria Rashidinejad**
University of Southern California
rishabha@usc.edu, feinashl@usc.edu, rashidin@usc.edu

Abstract

Reasoning models have advanced rapidly, but the dominant reinforcement learning from verifiable rewards (RLVR) recipe remains surprisingly narrow: sample many responses and reward each with a single bit indicating whether the final answer is correct. Yet many settings provide *rich feedback*, including execution traces, tool outputs, expert corrections, and model self-evaluations. We study how to use such feedback through a distributional variant of the classic imitation learning algorithm DAgger, where the learner has local access to an expert distribution on states visited by the current policy. This yields a simple forward cross-entropy objective that admits a blackbox expert and whose sequence-level gradient conduct rich credit assignment by propagating future expert-student disagreement back to earlier decisions. We show that prior RL with self-distillation objectives based on reverse KL or Jensen-Shannon fail to guarantee monotonic policy improvement: even when the expert has higher reward, their updates may increase probability on worse actions. In contrast, we show that forward cross-entropy admits monotonic policy improvement and enjoys guarantees on regret. We further show that our objective optimizes a teacher-weighted maximum-likelihood RL lower bound, leading to improved Pass@N. Empirically, our approach, DistIL, consistently improves over RLVR and RL with self-distillation baselines across a variety of domains: scientific reasoning, mathematical reasoning, and coding.

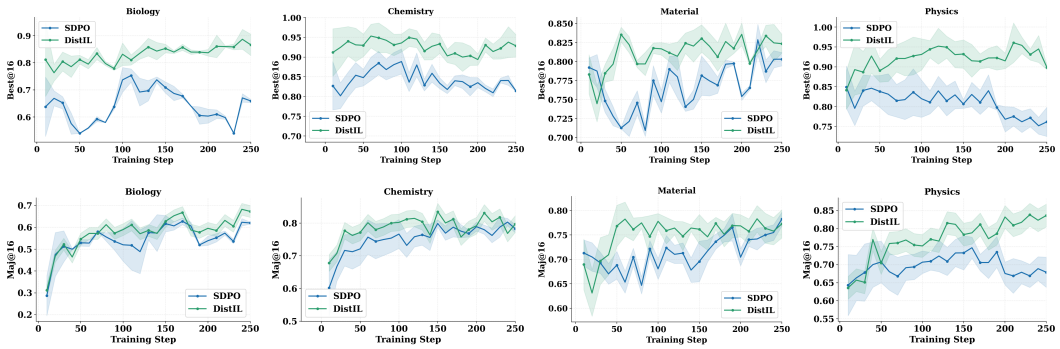

Figure 1: Validation Best@16 (top) and Maj@16 (bottom) of SDPO [16] and DistIL (Ours) across four scientific reasoning domains (biology, chemistry, materials, physics) on Qwen3-8B, plotted against training step. DistIL consistently outperforms SDPO on both metrics across all domains, with gains emerging within the first $\sim$20 steps and sustained throughout training. SDPO exhibits instability under extended training. Notably, Best@16 collapses on biology after step 100 and oscillates on chemistry and physics, while DistIL's curves remain smooth.

43rd International Conference on Machine Learning (ICML 2026). Workshop: RLxF: Reinforcement Learning from World Feedback.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a key paradigm for substantially enhancing the reasoning abilities of large language models (LLMs), as exemplified by advanced models such as OpenAI's O1 [17] and DeepSeek-R1 [14]. By optimizing pre-trained or chain-of-thought (CoT) [43] fine-tuned models through verifiable reward signals, RLVR enables LLMs to excel in complex logical tasks such as mathematics [40, 53, 55] and programming [31, 29]. Despite these advances, a fundamental limitation of RLVR is its reliance on a single binary scalar reward, assigned once per rollout to indicate correctness, which is then broadcast uniformly across all generated tokens via policy gradient. This coarse, uniform signal provides no information about *which* tokens or intermediate reasoning steps were causally responsible for success or failure, weakening token-level guidance and allowing erroneous reasoning steps to remain insufficiently penalized, potentially leading models to implicitly tolerate incorrect intermediate conclusions [52].

A natural remedy is denser supervision. Process reward models (PRMs) [25, 41] provide step-level signals but require costly annotation pipelines or separate critic networks. An orthogonal and increasingly prominent direction, on-policy self-distillation, exploits rich feedback already present in the environment (error traces, failed unit tests, natural-language critiques, or ground-truth solutions) by conditioning the current model on such feedback to produce a privileged self-teacher, then distilling its token-level predictions back into the student [16, 59]. This converts sparse outcome rewards into dense, token-level supervision without any external model or annotated data.

Despite their empirical promise, these methods rest on an implicit and critical assumption: *if the teacher policy is on average better than the student, then minimizing a divergence toward the teacher should monotonically improve the student*. We show this assumption is false in general. In this work, we identify two fundamental theoretical limitations of existing self-distillation methods, propose a principled algorithm, DistIL, that provably overcomes both, and validate it empirically across a suite of challenging reasoning benchmarks. Our contributions are as follows.

Provable limitations of existing self-distillation methods under general f-divergences (§3). We begin by analyzing the objectives of existing self-distillation methods, which typically minimize the reverse-KL or Jensen-Shannon divergence or more broadly, any f-divergence between the student and teacher policies [16, 59]. We identify two fundamental limitations. *First,* we prove that the assumption of monotonic improvement fails for a general family of f-divergence objectives: minimizing divergence to the teacher does *not* guarantee monotonic policy improvement and can actively upweight suboptimal actions (Propositions 1, 2). *Second,* these methods approximate policy gradients via local credit assignment, ignoring how future token differences should accurately inform the credit of past tokens. We show that this can lead to suboptimal policies (Proposition 3).

DistIL: A new algorithm for RL with rich feedback: We address these limitations by framing self-distillation as an on-policy imitation learning (IL) problem and extending DAgger [33], a classical IL algorithm, to a distributional setting, which we call DistIL. The student minimizes a forward cross-entropy loss to match the teacher's action distribution on states induced by its own policy. Unlike divergence-based self-distillation methods, DistIL can be implemented even with black-box teachers. Furthermore, DistIL enables sequence-level credit assignment, allowing feedback from later timesteps to accurately influence earlier decisions.

DistIL enjoys theoretical guarantees: We prove three key results: (i) DistIL enables *monotonic policy improvement* (Proposition 5); (ii) DistIL achieves a sublinear regret guarantee over training iterates (Theorem 1); (iii) DistIL objective maximizes a teacher-weighted lower bound on the expected log-likelihood of success (Proposition 6), connecting our algorithm to MaxRL [39], which implicitly optimizes $\texttt{pass@}n$ for any n.

Empirical results: Empirically, DistIL outperforms SDPO [16], OPSD [59], and GRPO [36] across a variety of benchmarks including: scientific reasoning [11], mathematical reasoning, and coding [18], highlighting the practical benefits of optimizing a theoretically grounded objective with proper temporal credit assignment.

2 Background and Problem Formulation

Reinforcement Learning. We consider a finite-horizon reinforcement learning setting, where a policy π interacts with an environment over a trajectory $y = (y_1, \ldots, y_H)$ of length H, conditioned

on an input $x \sim \rho$ (e.g., a prompt or problem). At each timestep t, the policy produces an action $y_t \sim \pi(\cdot \mid x, y_{1:t-1})$. The objective is to maximize the expected cumulative reward

$$J(\pi) := \mathbb{E}_{x\sim\rho}\, \mathbb{E}_{y\sim\pi(\cdot|x)} \left[\sum_{t=1}^{H} r_t(x, y_{1:t}) \right]. \tag{1}$$

Reinforcement Learning with Verifiable Rewards (RLVR). In many reasoning tasks, such as mathematics or code generation, rewards are *verifiable*: correctness can be checked automatically, often based on the final output. This leads to a sparse reward setting where $r_t = 0$ for $t < H$ and $r_H \in \{0, 1\}$. Methods such as GRPO [36] and its variants [26, 2] optimize policies using such outcome-based rewards. However, this supervision is sparse and delayed, assigning the same trajectory-level signal to all tokens. This makes credit assignment difficult, as the learning signal does not distinguish which actions led to success or failure. Moreover, when all sampled responses receive identical outcomes (e.g., all correct or all incorrect), the gradient signal vanishes, preventing learning and wasting sampling compute.

Learning from Rich Feedback via Distillation. In many settings, supervision extends beyond final-answer correctness and includes richer signals such as execution traces, error messages, expert annotations, or feedback from other models. These sources provide more informative guidance and can be leveraged for improved learning. A common approach is *on-policy distillation*, where a student policy π_S is trained to match a teacher policy π_T along trajectories sampled from the student [1, 48, 44, 54]. In *on-policy self-distillation*, the teacher is derived from the same model but conditioned on additional feedback f:

$$\pi_T(\cdot \mid x, y_{1:t-1}) := \pi_\theta(\cdot \mid x, f, y_{1:t-1}), \quad \pi_S(\cdot \mid x, y_{1:t-1}) := \pi_\theta(\cdot \mid x, y_{1:t-1}),$$

where π_T denotes the feedback-conditioned teacher and π_S the student policy. Given trajectories $y \sim \pi_S(\cdot \mid x)$, training minimizes

$$\mathcal{L}_{\text{self-distill}} := \mathbb{E}_{x\sim\rho,\ y\sim\pi_S(\cdot|x)} \left[\sum_{t=1}^{|y|} D(\pi_S(\cdot \mid x, y_{1:t-1}) \parallel \pi_T(\cdot \mid x, y_{1:t-1})) \right], \tag{2}$$

where D is a divergence measure, such as reverse KL (used in SDPO [16]) or Jensen-Shannon divergence (used in OPSD [59]).

Notation. We use $\mathrm{KL}(p\|q)$ for Kullback–Leibler divergence, $\mathrm{JS}(p\|q)$ for Jensen–Shannon divergence, and $\mathrm{H}^{\times}(p, q) := -\mathbb{E}_p[\log q]$ for cross-entropy. The operator $\mathsf{sg}(\cdot)$ denotes stop-gradient.

Problem formulation: RL from rich feedback. Given a dataset of prompts/problems $\mathcal{D}$, the goal is to learn a policy π_θ that maximizes the RL objective 1. However, the reward function r is unknown. To make the problem well defined, on-policy self-distillation thus assume teacher that is on average better than the student policy. Formally, let π_T denote the teacher and π_θ the student, and assume a positive reward gap

$$\Delta := \mathbb{E}_{x\sim\rho, y\sim\pi_T(\cdot|x)} [r(x, y)] - \mathbb{E}_{x\sim\rho, y\sim\pi_\theta(\cdot|x)} [r(x, y)] > 0.$$

3 Limitations of Existing On-Policy Self-Distillation Methods

3.1 Does divergence minimization guarantee monotonic policy improvement?

A central objective in policy optimization is *monotonic policy improvement* (MPI), wherein each update increases expected reward. This property is satisfied by many widely used RL methods such as natural policy gradient (NPG) methods, and approximately by practical variants such as PPO [34] and GRPO [36]. However, existing self-distillation methods, such as SDPO (reverse KL) [16] and OPSD (Jensen-Shannon) [59], optimize an f-divergence objective instead. We show that such objectives do not, in general, guarantee monotonic improvement, even in the contextual bandit setting.

Proposition 1. *Let π_T be a fixed teacher and π_θ a student policy. For the softmax parameterization, the Natural Policy Gradient (NPG) step minimizing an f-divergence $D_f(\pi_T\|\pi_\theta)$, i.e., $\theta' = \theta - \eta\, F(\theta)^{-1}\nabla_\theta D_f(\pi_T\|\pi_\theta)$, satisfies*

$$J(\pi_{\theta'}) = J(\pi_\theta) - \eta\, \mathbb{C}\mathrm{ov}_{y\sim\pi_\theta}\,(r(y), g(y)) + O(\eta^2),$$

where $g(y) = f\left(\frac{\pi_T(y)}{\pi_\theta(y)}\right) - \frac{\pi_T(y)}{\pi_\theta(y)} f'\left(\frac{\pi_T(y)}{\pi_\theta(y)}\right)$. *Consequently, for sufficiently small* $\eta > 0$*, the NPG step provides a policy improvement if and only if* $\mathbb{C}\mathrm{ov}_{y\sim\pi_\theta}(r(y), g(y)) < 0$. *In particular,* $\Delta > 0$ *does not suffice to guarantee this condition.*

A detailed proof of Proposition 1 can be found in Appendix B.1.

Remark. Using Proposition 1 for SDPO (i.e. reverse KL), we get $\mathbb{C}\mathrm{ov}_{y\sim\pi_\theta}\left(r(y), \log\frac{\pi_\theta(y)}{\pi(y)}\right)$ term. The covariance measures whether actions the student overweighs relative to the teacher tend to be high or low reward. If $\mathbb{C}\mathrm{ov} < 0$, excess mass is on bad actions, and the NPG corrects these errors. If $\mathbb{C}\mathrm{ov} > 0$, excess mass is on good actions and the NPG penalizes them, degrading the policy. Notably, the sign of the reward gap Δ does not constrain the sign of this covariance. Since the covariance depends on the joint structure of rewards and the student–teacher density ratio, its sign can vary arbitrarily across policies. As a result, reverse KL (SDPO) updates do not guarantee monotonic improvement.

Proposition 1 shows that no guarantees of monotonic policy improvement can be provided. To illustrate that such cases indeed exist, we present Proposition 2 below.

Proposition 2. *There exists a contextual bandit setting such that, even when the teacher policy is strictly better than the student policy (i.e.,* $\Delta > 0$*), an NPG step minimizing the reverse KL divergence, as in SDPO, does not lead to policy improvement.*

The construction for Proposition 2 can be found in Appendix B.2. We construct a simple three-armed bandit where the student over-assigns mass to a suboptimal action; under a reverse-KL NPG update, the normalization term outweighs the local log-ratio, causing the update to further increase probability on this suboptimal action.

3.2 Local credit assignment leads to suboptimal policies

Next, we study the gradient updates used in current self-distillation methods such as SDPO and OPSD. These methods induce *local credit assignment*, whereby gradients are computed token-wise under a frozen trajectory distribution. This arises from the use of a stop-gradient operator in Eq. (2) on the sampling process. Consequently, gradients pass through the expectation and take the following form:

$$\nabla_\theta \mathcal{L}_{\text{self-distill}} := \mathbb{E}_{x\sim\rho,\, y\sim\pi_S(\cdot|x)}\left[\sum_{t=1}^{|y|} \nabla_\theta D(\pi_\theta(\cdot \mid x, y_{1:t-1}) \parallel \pi_T(\cdot \mid x, y_{1:t-1}))\right], \tag{3}$$

While Eq. (3) simplifies gradient computation, it largely ignores how current token choices affect future token distributions due to the stop-gradient operator, thereby failing to perform proper credit assignment. This mismatch between token-wise and sequence-level gradients can lead to incorrect credit assignment to early decisions, whose impact is only realized downstream. To demonstrate this, we provide a simple construction in Proposition 3.

Proposition 3. *There exists a two-step MDP and policy* π_θ *such that the tokenwise gradient provides no update signal for the first-step decision, while the sequence-level gradient is nonzero. Consequently, tokenwise optimization attains a strictly suboptimal solution.*

Proposition 3 highlights a fundamental limitation of tokenwise updates: they may fail to assign proper credit to early decisions, leading to strictly suboptimal solutions. Consequently, methods such as SDPO and OPSD, which rely on such gradient estimators, may suffer from this limitation. A concrete construction is provided in Appendix B.3. We construct a simple two-step problem where only one first-step action leads to informative second-step feedback. Token-wise objectives ignore how this future mismatch depends on the first decision, providing zero gradient and getting stuck at a suboptimal solution, while sequence-level optimization correctly propagates this signal.

Taken together, these results highlight two distinct issues with existing self-distillation approaches: the choice of objective can fail to guarantee monotonic improvement, and the associated token-wise gradient estimators fail to assign credit to early decisions. Addressing these failures requires reconsidering both the objective and how gradients are computed. We turn to this next.

4 RL from Rich Feedback with Distributional Imitation Learning

Motivated by the limitations of existing self-distillation approaches, we seek an objective that ensures consistent policy improvement, along with a training procedure that enables proper credit assignment across the sequence.

4.1 Distributional Imitation Learning Formulation

We view self-distillation through the lens of *on-policy imitation learning*. Our setting closely resembles DAgger [33]: the learner collects data under its own policy, queries an expert on visited states, and updates via imitation. This perspective is particularly natural for LLMs, where the student generates trajectories and the teacher can be queried for supervision at each visited state. A key departure from classical DAgger is our use of a *distributional objective*: rather than supervising individual actions, we minimize a cross-entropy loss between student and teacher distributions. Crucially, this objective does not require access to the full teacher distribution: it admits unbiased estimation from a single sample, can leverage a small set of samples (e.g., top-k), or utilize the full distribution when available. This flexibility allows our method to operate seamlessly across both black-box teachers and fully accessible models. Formally, we consider the objective

$$\mathcal{L}_{\text{DistIL}}(\theta) := \mathbb{E}_{x\sim\rho,\ y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{H} H^{\times}\left(\mathsf{sg}\left(\pi_\theta(\cdot \mid x, y_{1:t-1}, f)\right),\ \pi_\theta(\cdot \mid x, y_{1:t-1})\right)\right] \tag{4}$$

where we used $\pi_T(\cdot \mid x, y_{1:t-1}) = \text{sg}\left(\pi_\theta(\cdot \mid x, y_{1:t-1}, f)\right)$. Our objective has a key advantage over SDPO: it can leverage a black-box teacher (e.g., Claude or human annotators) through sample-based estimation, whereas SDPO requires explicit access to the teacher distribution.

4.2 Local and future credit assignment

Next, we remove the stop-gradient on the sampling process, thereby allowing gradients to correctly reflect the dependence of future outcomes on earlier decisions. We provide the gradient of our $\mathcal{L}_{\text{DistIL}}(\theta)$ in Proposition 4.

Proposition 4. *For $s_t = (x, y_{1:t-1})$, the gradient of $\mathcal{L}_{\text{DistIL}}(\theta)$ is given by*

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\text{DistIL}} = &\underbrace{\mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta H^{\times}\Big(\mathsf{sg}(\pi_\theta(\cdot \mid s_t, f)),\ \pi_\theta(\cdot \mid s_t)\Big)\right]}_{\textit{local credit assignment}} \\ &+ \underbrace{\mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\left(\sum_{i>t} H^{\times}\Big(\mathsf{sg}(\pi_\theta(\cdot \mid s_i, f)),\ \pi_\theta(\cdot \mid s_i)\Big)\right)\right]}_{\textit{future-credit assignment}},\end{aligned} \tag{5}$$

The second term in Eq. (5) propagates future teacher–student disagreement back to earlier decisions through the score function term $\nabla_\theta \log \pi_\theta(a_t \mid s_t)$. This stands in contrast to local token-wise gradient estimators, such as those used in OPSD and SDPO, which omit this dependence by treating the trajectory distribution as fixed, and thus fail to correctly account for how early decisions influence future outcomes. The derivation of Proposition 5 is provided in appendix B.4.

5 Theoretical Properties of DistIL

We now present theoretical guarantees for DistIL. First, we show DistIL enjoys policy improvement. We subsequently derive a regret bound, and finally connect our objective to maximum-likelihood reinforcement learning (MaxRL), providing additional insight into its behavior.

5.1 Monotonic Policy Improvement

Our analysis relies on a mild tangent space assumption, under which we establish a monotonic improvement property of our objective.

Assumption 1 (Tangent space realizability)**.** *Assume that the teacher direction lies in the tangent space of the student policy manifold at θ. That is, there exists a vector $u \in \mathbb{R}^d$ such that for all states s and actions a,*

$$\log \pi_T(a \mid s) - \log \pi_\theta(a \mid s) = u^\top \nabla_\theta \log \pi_\theta(a \mid s).$$

Intuitively, Assumption 1 requires that the discrepancy between the teacher and student policies lies in the span of the student's score functions, meaning that the direction from the student to the teacher is locally realizable by the policy class. This is a mild condition for sufficiently expressive policies. In particular, it holds exactly for tabular softmax policies. A formal statement and proof are provided in Appendix B.5.

Proposition 5. *Under Assumption 1, suppose the teacher policy is strictly better than the current policy (i.e., $\Delta > 0$). Then, the NPG step minimizing $H^\times(\pi_T \| \pi_\theta)$, i.e., $\theta' = \theta - \eta\, F(\theta)^{-1} \nabla_\theta H^\times(\pi_T \| \pi_\theta)$ for sufficiently small $\eta > 0$, satisfies $J(\pi_{\theta'}) = J(\pi_\theta) + \eta\Delta + O(\eta^2)$, and hence the update provides monotonic policy improvement.*

Proposition 5 shows that, under the tangent space assumption, the natural gradient update aligns the policy with the teacher direction, yielding monotonic improvement whenever the teacher is better. A detailed analysis for frozen state distribution is provided in Appendix B.6.

Remark. In the forward Cross Entropy loss, the NPG step becomes a convex interpolation toward the teacher, so improvement follows directly from the fact that we assume the teacher is superior. However, in reverse KL (SDPO), the NPG step is not an interpolation and has an extra weight $\log \pi_\theta / \pi_T$. This term can lead to increased probability on bad actions, even if the teacher is better on average. Note that this argument goes beyond the typical mode-seeking vs. mean-seeking of forward and reverse KL.

5.2 Regret Guarantee of DistIL

Next, we show that our algorithm enjoys regret bound.

Theorem 1. *Let π_T be the teacher policy and π_S, π_S^A be the student policies returned by the updates in the following two per-iteration loss schemes respectively. Then, for the per-iteration loss $\ell_i(\pi) = -\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_t^{\pi^i}} \Big[\sum_{a \in \mathcal{A}} \pi_T(a \mid s_t) \log \pi(a \mid s_t) \Big]$, we have*

$$J(\pi_T) - J(\pi_S) \;\lesssim\; \sqrt{\bar{\sigma}^2_{\pi_T} \frac{HC}{2}} \left(\frac{2C_0}{n} \right)^{1/4} + \frac{\mu_T H C}{2} \sqrt{\frac{2C_0}{n}}.$$

We refer to Appendix B.7 for formal definitions and properties of C, C_0, $\bar{\sigma}_{\pi_T}$ and μ_T, and the proof of Theorem 1. Theorem 1 shows that the regret of the learned policy is controlled through two problem-dependent quantities: $\bar{\sigma}_{\pi_T}$ and μ_T. The term $\bar{\sigma}_{\pi_T}$ captures the stochasticity of the teacher while μ_T measures the worst-case sensitivity of the environment to taking suboptimal actions. When the teacher is nearly deterministic, the second term dominates and yields a faster $O(n^{-1/2})$ rate. In contrast, for highly stochastic teachers, the variance term leads to a slower $O(n^{-1/4})$ rate.

Remark. Our analysis builds on the per-state mirror descent framework of [3], but requires a non-trivial extension to the online setting. In particular, their analysis assumes expectations with respect to a fixed distribution, which does not hold in our setting where the state distribution evolves on every iteration. We address this by deriving a recurrence relation for the intermediate error terms, which enables us to extend the analysis to the online regime. Full details are provided in Appendix B.7. This technique may be of independent interest for analyzing other online learning algorithms.

5.3 Connections to MaxRL

Finally, our forward cross-entropy objective can be interpreted as maximizing a lower bound on a teacher-weighted maximum likelihood RL (MaxRL) objective [39]. In this regard, we have the following proposition.

Proposition 6. *For a fixed prompt x, define the success probability $p_\pi(x) := \mathbb{E}_{y \sim \pi(\cdot \mid x)}[r(x, y)]$, where $r(x, y) \in \{0, 1\}$ indicates correctness. Then*

$$-H^\times(\pi_T(\cdot \mid x), \pi_\theta(\cdot \mid x)) \le p_{\pi_T}(x) \log p_{\pi_\theta}(x),$$

Method	Chemistry		Physics		Biology		Materials		Average	
	1h	5h	1h	5h	1h	5h	1h	5h	1h	5h
Qwen3-8B	41.2		59.2		30.8		58.9		47.5	
+ GRPO (off-policy)	65.9	74.5	63.8	<u>72.7</u>	35.1	59.9	<u>74.3</u>	**77.1**	59.8	71.1
+ GRPO (on-policy)	63.3	63.4	63.6	63.6	49.8	49.8	73.9	74.1	62.7	62.7
+ SDPO	<u>73.0</u>	<u>80.2</u>	<u>68.0</u>	72.4	<u>52.9</u>	<u>63.6</u>	72.2	75.9	<u>66.5</u>	<u>73.0</u>
+ DistIL *(Ours)*	**75.8**	**80.8**	**72.7**	**80.8**	**53.3**	**66.6**	**74.9**	<u>76.2</u>	**69.2**	**76.1**
Olmo3-7B-Instruct	22.8		37.7		16.2		36.7		28.4	
+ GRPO (off-policy)	39.7	56.7	55.3	63.3	35.6	**55.8**	70.9	75.0	50.4	62.7
+ GRPO (on-policy)	51.4	57.5	<u>62.7</u>	62.7	**49.8**	49.8	<u>73.3</u>	73.5	59.3	60.9
+ SDPO	<u>70.2</u>	<u>79.2</u>	59.8	<u>64.9</u>	<u>49.5</u>	52.9	71.8	**78.1**	<u>62.8</u>	<u>68.8</u>
+ DistIL *(Ours)*	**72.1**	**81.0**	**67.4**	**74.5**	47.8	<u>55.3</u>	**73.5**	<u>76.9</u>	**65.2**	**71.9**

Table 1: Comparison on scientific reasoning benchmarks (SciKnowEval L3). We report best avg@16 within 1h and 5h of wall-clock training on $4\times$ NVIDIA H200 GPUs. Average is computed over all four domains. **Bold** denotes the best result per column; <u>underline</u> denotes the second best.

Remark. Proposition 6 shows that minimizing forward cross-entropy maximizes a lower bound on a teacher-weighted success objective, linking our DistIL to MaxRL-style optimization. The result follows by partitioning the cross-entropy over correct and incorrect responses and lower bounding the contribution of correct responses, while noting that the remaining terms are nonnegative. Thus, optimizing cross-entropy implicitly maximizes a teacher-weighted log-likelihood of success. This provides a principled explanation for our improved Pass@n performance, as MaxRL objectives are shown to optimize Pass@n for all n [39]. A detailed discussion is provided in Appendix B.8.

6 Experimental Results

We evaluate our method across three representative settings that capture different feedback regimes and levels of supervision: (i) science reasoning benchmarks without rich environment feedback; (ii) coding tasks with rich execution-based feedback; and (iii) mathematical reasoning on challenging problems with access to ground-truth solutions.

6.1 Learning under Sparse Feedback: Science

We evaluate in the standard RLVR setting, where only a binary success signal is observed per rollout. Following SDPO, we sample multiple rollouts per input and use successful trajectories as feedback f, training the model to increase their likelihood and converting sparse rewards into a dense, self-supervised signal.

Setup. We compare against SDPO, on-policy GRPO (trajectories from the current policy), and off-policy GRPO (past rollouts reweighted via importance sampling). For all baselines, we adopt the same hyperparameter settings as in SDPO. Scientific reasoning is assessed on undergraduate-level problems in chemistry, physics, biology, and materials science using the L3 reasoning subset of SciKnowEval [11], with a train–test split for in-domain generalization. We initialize from Qwen38B [48] and Olmo3-7B-Instruct [30], and report Best@16 and Maj@16 (Figure 1) and Avg@16 (Table 1) against wall-clock training time using the `verl` framework [37].

Empirical Findings. Table 1 and Figure 1 summarize our results. DistIL achieves the best avg@16 performance in the large majority of columns across both model families, with the largest gains in physics and chemistry domains requiring multi-level reasoning, where sequence-level credit assignment (Proposition 4) is most impactful; physics gains reach $8.1/9.6$ points at 5h over next-best on Qwen3-8B/OLMo3. Beyond accuracy, DistIL's lead emerges within $\sim$20 steps and is largely monotonically sustained (Proposition 5), while SDPO exhibits instability: Best@16 collapses on biology after step 100 and oscillates on chemistry and physics. DistIL's Best@16 advantage is grounded in Proposition 6: our forward cross-entropy objective maximizes a teacher-weighted lower bound on success probability, implicitly optimizing Pass@n for all n [39]. On-policy GRPO saturates within one hour; off-policy GRPO lags DistIL by $5+$ points on reasoning-heavy domains at 5h.

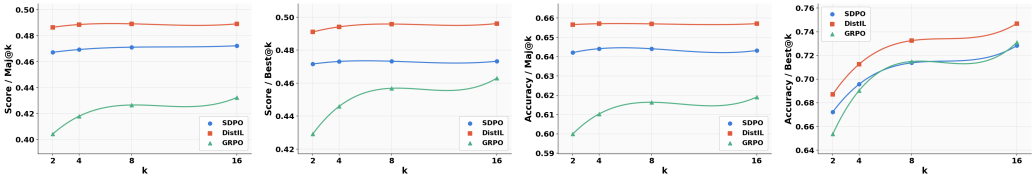

Figure 2: LCBv6 evaluation at τ=0.2 for checkpoint at step-80 (following SDPO), reporting Score and Accuracy at Best@k and Maj@k for $k \in \{2, 4, 8, 16\}$.

Model	Method	AIME24		AIME25		HMMT25		AMC23		Minerva	
		Avg	Pass	Avg	Pass	Avg	Pass	Avg	Pass	Avg	Pass
Qwen3-4B	Base	61.4	82.0	50.3	69.5	30.3	48.8	93.8	98.8	43.2	48.6
	+ SFT	55.8	80.9	43.1	67.4	29.7	46.0	91.7	97.0	41.9	50.4
	+ GRPO	61.4	82.0	50.3	69.5	30.3	48.8	93.8	98.8	43.2	48.6
	+ SDPO	60.9	85.4	49.6	74.1	32.8	50.9	93.9	99.8	43.4	52.0
	+ OPSD	63.2	85.4	51.5	76.1	**33.0**	54.2	94.5	99.6	43.4	51.8
	+ DistIL *(Ours)*	**65.3**	**87.5**	**55.3**	**77.6**	**33.0**	**56.6**	**94.8**	**100.0**	**44.2**	**52.9**
Qwen3-8B	Base	75.9	86.8	66.9	79.6	44.7	65.6	95.6	**100.0**	48.9	56.3
	+ SFT	74.2	85.3	65.3	80.7	42.2	64.4	96.3	**100.0**	48.2	56.2
	+ GRPO	75.9	86.8	66.9	79.6	44.7	65.6	95.6	**100.0**	48.9	56.3
	+ SDPO	**76.9**	89.5	68.3	84.0	45.6	71.2	96.5	**100.0**	49.2	57.7
	+ OPSD	76.5	**91.3**	69.7	83.3	45.5	68.2	96.2	**100.0**	49.2	57.8
	+ DistIL *(Ours)*	76.4	90.7	**71.1**	**85.0**	**46.4**	**71.4**	**96.6**	**100.0**	**49.5**	**58.4**

Table 2: Performance comparison on mathematical reasoning benchmarks for Qwen3 models. Sampling configuration is given in Appendix D.2. **Bold** denotes the best result per column; underline denotes the second best. Following OPSD [59], we report the best checkpoint up to step 100.

6.2 Learning with Rich Environment Feedback: Coding

We evaluate in environments with rich feedback, where signals go beyond binary success, such as execution errors, which standard RLVR cannot exploit. Following SDPO, we sample multiple rollouts per input and use execution-based feedback to construct supervision. Concretely, rollouts are evaluated against public unit tests, and feedback f is derived from the resulting code execution logs.

Setup. We compare against SDPO and GRPO with identical hyperparameters as in SDPO on LiveCodeBench (LCBv6) [18], using public tests for training feedback and private tests for evaluation, initialized from Qwen3-8B [48]. Following [18], we report exact correctness by determining whether all test cases are passed (Score/Avg@k, Score/Maj@k, Score/Best@k) and partial correctness: fraction of test cases passed (Accuracy/Avg@k, Accuracy/Maj@k, Accuracy/Best@k), at temperature 0.2, top_p 0.95.

Empirical Findings. DistIL achieves Accuracy/Avg@16 $= 0.656$ and Score/Avg@16 $= 0.482$, outperforming SDPO (0.643, 0.467) and GRPO (0.600, 0.405). The large gap over GRPO reflects its fundamental inability to exploit execution feedback. Figure 2 further shows consistent gains across all k on both Best@k and Maj@k for score and accuracy, with the advantage most pronounced at small k. The gap between DistIL and SDPO could be attributed to our theoretically grounded-objective and squence-level credit assignment.

6.3 Learning to Solve Very Hard Mathematical Reasoning Problems

We evaluate with access to ground-truth solutions as feedback f on very hard mathematical reasoning problems, enabling learning on problems where the base model never produces a correct rollout. In this regime, RLVR methods such as GRPO receive zero advantage and cannot improve, while distillation-based approaches directly leverage ground-truth solutions to make progress.

Setup. We construct a challenging training set of 738 hard mathematical problems sourced from [8] and OmniMath (problems with pass@512 $= 0$ for Qwen3-4B-Instruct), following [35, 32]. We compare against OPSD (forward-KL), SDPO (reverse-KL), GRPO, and SFT, initialized from Qwen3-

8B and Qwen3-4B-Instruct-2507 models. Evaluation covers AIME24, AIME25, HMMT25, AMC23, and Minerva.

Empirical Findings. Table 2 summarizes results. GRPO exactly matches the base model on both scales, confirming the expected failure: zero pass rate yields zero advantage. SFT degrades below the base model on most benchmarks, indicating overfitting rather than generalizable reasoning. DistIL achieves the best results across the large majority of columns at both model scales. Gains are most substantial on AIME25, where DistIL leads the next-best method on Avg@16 by 3.8 points on Qwen3-4B and 1.4 points on Qwen3-8B. On the stronger Qwen3-8B, where SDPO and OPSD are competitive, DistIL consistently ranks first or second, with no degradation relative to the base model.

An ablation study of DistIL is provided in Appendix E.2.

7 Related Work

Credit Assignment in RLVR. RLVR applies identical trajectory-level signals to all tokens, making credit assignment challenging for long-horizon reasoning [36, 14]. Process reward models introduce intermediate supervision but require additional annotation or auxiliary models [25, 41]. Alternative approaches reweight token updates using heuristic internal signals such as entropy or uncertainty [45, 24]. In contrast, DistIL constructs token-level supervision directly from privileged feedback, yielding structured credit assignment without auxiliary models or heuristics.

On-Policy Distillation and Self-Distillation. On-policy distillation trains a student on its own rollouts under divergence-based teacher supervision, reducing distribution mismatch relative to off-policy approaches [1, 27]. Recent work extends this to self-distillation, where a single model acts as both teacher and student conditioned on privileged feedback [59, 16, 50]. However, self-distillation can suppress epistemic uncertainty [19] and is sensitive to teacher-student alignment [23]; recent methods address this via competence-aware weighting and hybrid routing [47, 21]. Our work revisits the underlying objective: replacing divergence minimization with a forward cross-entropy formulation yields monotonic policy improvement guarantees and stable gradient estimation, connecting self-distillation to DAgger [33]. Extended related work is provided in Appendix C.

8 Conclusion

We view reinforcement learning from rich feedback as an on-policy imitation learning problem and propose DistIL, a distributional extension of DAgger based on forward cross-entropy. We identify two fundamental limitations of existing self-distillation methods: (i) optimizing any f-divergence provides no monotonic policy improvement guarantee and can actively upweight suboptimal actions; and (ii) token-wise gradient estimation performs only local credit assignment, preventing future outcomes from informing past decisions and causing error accumulation over long sequences. DistIL addresses both by replacing divergence minimization with a forward cross-entropy objective that aligns updates with the teacher advantage direction and propagates credit through the student-induced state distribution. We prove monotonic policy improvement under a mild realizability condition, sublinear regret over training iterates, and a connection to MaxRL that implicitly optimizes Pass@n for all n simultaneously. Empirically, DistIL achieves the best performance across most scientific reasoning, competitive programming, and hard mathematical reasoning, with particularly large gains where RLVR methods fail due to zero advantage. Taken together, our results underscore that objective design and gradient structure are as consequential as data or scale for learning from rich feedback, and position distributional imitation learning as a principled and theoretically grounded path toward stronger reasoning in LLMs.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

[2] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization

for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, 2024.

[3] Jonathan D Chang, Wenhao Zhan, Owen Oertell, Kianté Brantley, Dipendra Misra, Jason D Lee, and Wen Sun. Dataset reset policy optimization for rlhf. *arXiv preprint arXiv:2404.08495*, 2024.

[4] Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Step-level value preference optimization for mathematical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7889–7903, 2024.

[5] Minghan Chen, Guikun Chen, Wenguan Wang, and Yi Yang. Seed-grpo: Semantic entropy enhanced grpo for uncertainty-aware policy optimization. *arXiv preprint arXiv:2505.12346*, 2025.

[6] Xinzhu Chen, Xuesheng Li, Zhongxiang Sun, and Weijie Yu. Beyond high-entropy exploration: Correctness-aware low-entropy segment-based advantage shaping for reasoning llms. *arXiv preprint arXiv:2512.00908*, 2025.

[7] Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30377–30385, 2026.

[8] CMU AIRe Lab. Pope-hard-w-oracle-solution, 2026. URL `https://huggingface.co/datasets/CMU-AIRe/POPE-HARD-w-oracle-solution`. Part of the POPE (Privileged On-Policy Exploration) dataset collection.

[9] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

[10] Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*, 2025.

[11] Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

[12] Dylan J Foster, Adam Block, and Dipendra Misra. Is behavior cloning all you need? understanding horizon in imitation learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=8KPyJm4gt5`.

[13] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The twelfth international conference on learning representations*, 2024.

[14] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[15] Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026.

[16] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

[17] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[18] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[19] Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026.

[20] LAS Group, ETH Zurich. Sdpo: Reinforcement learning via self-distillation. `https://github.com/lasgroup/SDPO`, 2026.

[21] Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026.

[22] Yang Li, Zhichen Dong, Yuhan Sun, Weixun Wang, Shaopan Xiong, Yijia Luo, Jiashun Liu, Han Lu, Jiamang Wang, Wenbo Su, et al. Attention illuminates llm reasoning: The preplan-and-anchor rhythm enables fine-grained policy optimization. *arXiv preprint arXiv:2510.13554*, 2025.

[23] Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

[24] Ziheng Li, Liu Kang, Feng Xiao, Luxi Xing, Qingyi Si, Zhuoran Li, Weikang Gong, Deqing Yang, Yanghua Xiao, and Hongcheng Guo. Outcome-grounded advantage reshaping for fine-grained credit assignment in mathematical reasoning. *arXiv preprint arXiv:2601.07408*, 2026.

[25] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The twelfth international conference on learning representations*, 2023.

[26] 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. *arXiv preprint arXiv:2503.20783*, 2025.

[27] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

[28] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

[29] Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, et al. Deepcoder: A fully open-source 14b coder at o3-mini level, 2025. Notion Blog.

[30] Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

[31] Zhenyu Pan and Han Liu. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. *arXiv preprint arXiv:2503.18470*, 2025.

[32] Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. Pope: Learning to reason on hard problems via privileged on-policy exploration. *arXiv preprint arXiv:2601.18779*, 2026.

[33] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[35] Amrith Setlur, Zijian Wang, Andrew Cohen, Paria Rashidinejad, and Sang Michael Xie. Reuse your flops: Scaling rl on hard problems by conditioning on very off-policy prefixes. *arXiv preprint arXiv:2601.18795*, 2026.

[36] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

[38] Wei Sun, Wen Yang, Pu Jian, Qianlong Du, Fuwei Cui, Shuo Ren, and Jiajun Zhang. Ktae: A model-free algorithm to key-tokens advantage estimation in mathematical reasoning. *arXiv preprint arXiv:2505.16826*, 2025.

[39] Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. *arXiv preprint arXiv:2602.02710*, 2026.

[40] Sijun Tan, Michael Luo, Justin Wong, Colin Cai, Xiaoxiang Shi, William Yuan Tang, Manan Roongta, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, et al. Deepscaler: Effective rl scaling of reasoning models via iterative context lengthening. *Notion Blog*, 2025.

[41] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

[42] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

[43] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[44] Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

[45] Can Xie, Ruotong Pan, Xiangyu Wu, Yunfei Zhang, Jiayi Fu, Tingting Gao, and Guorui Zhou. Unlocking exploration in rlvr: Uncertainty-aware advantage shaping for deeper reasoning. *arXiv preprint arXiv:2510.10649*, 2025.

[46] Wenda Xu, Rujun Han, Zifeng Wang, Long T Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. *arXiv preprint arXiv:2410.11325*, 2024.

[47] Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. Paced: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*, 2026.

[48] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[49] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*, 2025.

[50] Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

[51] Chenxu Yang, Qingyi Si, Mz Dai, Dingyu Yao, Mingyu Zheng, Minghui Chen, Zheng Lin, and Weiping Wang. Test-time prompt intervention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34223–34231, 2026.

[52] Xinhao Yao, Lu Yu, Xiaolin Hu, Fengwei Teng, Qing Cui, Jun Zhou, and Yong Liu. The debate on rlvr reasoning capability boundary: Shrinkage, expansion, or both? a two-stage dynamic view. *arXiv preprint arXiv:2510.04028*, 2025.

[53] Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

[54] Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

[55] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

[56] Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*, 2024.

[57] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516, 2025.

[58] Siyan Zhao. Opsd: On-policy self-distillation for large language models. `https://github.com/siyan-zhao/OPSD`, 2026.

[59] 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. *arXiv preprint arXiv:2601.18734*, 2026.

Appendices

A Useful Lemmas **15**

B Theoretical Results **17**

B.1 Lack of Monotonic Improvement Guarantees under f-Divergences 17

B.2 Existence of Non-Monotonic Improvement Cases for Reverse-KL 18

B.3 Local Credit Assignment leads to Sub-Optimal Policies 19

B.4 DistIL's Gradient Derivation 20

B.5 Tangent-Space Realizability 21

B.6 Monotonic Improvement of DistIL 22

B.7 Regret Analysis 23

B.8 Connections with Max RL 27

C Extended Related Work **27**

D Experiment Details **28**

D.1 Hyperparameters for Science and Coding Benchmark 28

D.2 Hyperparameters for Maths task 28

E Additional Results **28**

E.1 More results for Coding. 28

E.2 Ablation Study 29

E.2.1 Credit Assignment 29

E.2.2 Top-1 v/s Top k v/s Full distribution 29

F Limitations **29**

G LLM Usage **30**

A Useful Lemmas

Lemma 1 (Fisher identity for softmax policies)**.** *Let π_θ be a softmax policy over a finite action set $\mathcal{A}$, with*

$$\pi_\theta(a) = \frac{e^{\theta_a}}{\sum_{b\in\mathcal{A}} e^{\theta_b}}.$$

Define the score vector $s_a := \nabla_\theta \log \pi_\theta(a)$, and the Fisher information matrix $F(\theta) = \mathbb{E}_{a\sim\pi_\theta}\big[s_a s_a^\top\big]$. Then, for any $a, b \in \mathcal{A}$,

$$s_a^\top F(\theta)^{-1} s_b = \frac{\mathbf{1}\{a=b\}}{\pi_\theta(b)} - 1.$$

Equivalently, $\pi_\theta(a)\, s_a^\top F(\theta)^{-1} s_b = \mathbf{1}\{a=b\} - \pi_\theta(a)$.

Proof. Let $m := |\mathcal{A}|$ and we write $p_a := \pi_\theta(a)$. Let $e_a \in \mathbb{R}^m$ denote the a-th standard basis vector, and let $\mathbf{1} \in \mathbb{R}^m$ denote the all-ones vector. Then, for the softmax parameterization, we have

$$\frac{\partial \log \pi_\theta(a)}{\partial \theta_c} = \mathbf{1}\{a=c\} - p_c.$$

Hence, $s_a = e_a - p$, where $p = (p_1, \ldots, p_m)^\top$.

Now, by definition, $F(\theta) = \mathbb{E}_{a\sim p}\big[(e_a - p)(e_a - p)^\top\big]$. Upon expanding, we get $F(\theta) = \mathbb{E}_{a\sim p}[e_a e_a^\top] - pp^\top$. Since $\mathbb{E}_{a\sim p}[e_a e_a^\top] = \mathrm{diag}(p)$, we obtain

$$F(\theta) = \mathrm{diag}(p) - pp^\top.$$

Next, we fix $b \in \mathcal{A}$, and define $v := \frac{e_b}{p_b} - \mathbf{1}$. We claim that $F(\theta)v = s_b$. For this, we note that $p^\top v = \sum_{a\in\mathcal{A}} p_a \left(\frac{\mathbf{1}\{a=b\}}{p_b} - 1\right) = 1 - 1 = 0$. Therefore,

$$F(\theta)v = (\mathrm{diag}(p) - pp^\top)v = \mathrm{diag}(p)v - p(p^\top v) = \mathrm{diag}(p)v.$$

Now,

$$\mathrm{diag}(p)v = \mathrm{diag}(p)\left(\frac{e_b}{p_b} - \mathbf{1}\right) = e_b - p = s_b.$$

Thus, $F(\theta)v = s_b$. Now, assuming that inverse of $F(\theta)$ exists, we have

$$F(\theta)^{-1} s_b = v = \frac{e_b}{p_b} - \mathbf{1}.$$

Finally, using $(s_a = e_a - p)$, we have $s_a^\top F(\theta)^{-1} s_b = s_a^\top \left(\frac{e_b}{p_b} - \mathbf{1}\right)$. Since $s_a^\top \mathbf{1} = 0$, this simplifies to

$$s_a^\top F(\theta)^{-1} s_b = \frac{s_a^\top e_b}{p_b} = \frac{\mathbf{1}\{a=b\} - p_b}{p_b} = \frac{\mathbf{1}\{a=b\}}{p_b} - 1.$$

Moreover, multiplying both sides by (p_a) gives the equivalent form

$$p_a\, s_a^\top F(\theta)^{-1} s_b = \mathbf{1}\{a=b\} - p_a.$$

□

Lemma 2 (Trajectory score-function identity)**.** *Let $\tau \sim p_\theta(\tau)$ be a differentiable trajectory distribution, and let $f(\tau, \theta)$ be any differentiable function. Then*

$$\nabla_\theta \mathbb{E}_{\tau\sim p_\theta}[f(\tau,\theta)] = \mathbb{E}_{\tau\sim p_\theta}[f(\tau,\theta)\nabla_\theta \log p_\theta(\tau) + \nabla_\theta f(\tau,\theta)].$$

In particular, if f does not depend on θ, then $\nabla_\theta \mathbb{E}_{\tau\sim p_\theta}[f(\tau)] = \mathbb{E}_{\tau\sim p_\theta}[f(\tau)\nabla_\theta \log p_\theta(\tau)]$. Moreover,

$$\mathbb{E}_{\tau\sim p_\theta}[\nabla_\theta \log p_\theta(\tau)] = 0.$$

Proof. Using the definition of expectation and differentiating it gives,

$$\nabla_\theta \mathbb{E}_{\tau\sim p_\theta}[f(\tau,\theta)] = \nabla_\theta \sum_\tau p_\theta(\tau) f(\tau,\theta) = \sum_\tau \nabla_\theta p_\theta(\tau)\, f(\tau,\theta) + \sum_\tau p_\theta(\tau)\nabla_\theta f(\tau,\theta).$$

Next, we use $\nabla_\theta p_\theta(\tau) = p_\theta(\tau)\nabla_\theta \log p_\theta(\tau)$, to get

$$\begin{aligned}\nabla_\theta \mathbb{E}_{\tau\sim p_\theta}[f(\tau,\theta)] &= \sum_\tau p_\theta(\tau) f(\tau,\theta)\nabla_\theta \log p_\theta(\tau) + \sum_\tau p_\theta(\tau)\nabla_\theta f(\tau,\theta)\\ &= \mathbb{E}_{\tau\sim p_\theta}[f(\tau,\theta)\nabla_\theta \log p_\theta(\tau) + \nabla_\theta f(\tau,\theta)],\end{aligned}$$

Now, if f does not depend on θ, the second term vanishes. Finally,

$$\mathbb{E}_{\tau\sim p_\theta}[\nabla_\theta \log p_\theta(\tau)] = \sum_\tau p_\theta(\tau)\nabla_\theta \log p_\theta(\tau) = \sum_\tau \nabla_\theta p_\theta(\tau) = \nabla_\theta \sum_\tau p_\theta(\tau) = \nabla_\theta 1 = 0.$$

□

Lemma 3. *Let $\Phi : \Pi \to \mathbb{R}$ be a differentiable strictly convex function, and let*

$$D_\Phi(\pi\|\rho) := \Phi(\pi) - \Phi(\rho) - \langle \nabla\Phi(\rho),\, \pi - \rho\rangle$$

denote the Bregman divergence induced by Φ. For the following mirror-descent trust-region update:

$$\pi^{i+1} \in \arg\min_{\pi\in\Pi} \left\{ \langle \nabla \ell_i(\pi^i), \pi\rangle + \frac{1}{\eta} D_\Phi(\pi\|\pi^i) \right\},$$

we have, for all $\pi \in \Pi$,

$$\langle \eta\nabla\ell_i(\pi^i) + \nabla\Phi(\pi^{i+1}) - \nabla\Phi(\pi^i),\, \pi - \pi^{i+1}\rangle \geq 0.$$

Proof. The objective is convex in π, and π^{i+1} is a minimizer over the convex set Π. Therefore, the standard first-order optimality condition gives

$$\langle \nabla\ell_i(\pi^i) + \frac{1}{\eta}(\nabla\Phi(\pi^{i+1}) - \nabla\Phi(\pi^i)),\, \pi - \pi^{i+1}\rangle \geq 0,$$

for all $\pi \in \Pi$. Multiplying by η gives the final inequality. □

Lemma 4. *For any differentiable strictly convex Φ and any u, v, w in its domain,*

$$\langle \nabla\Phi(u) - \nabla\Phi(v),\, w - u\rangle = D_\Phi(w,v) - D_\Phi(w,u) - D_\Phi(u,v).$$

Proof. By definition,

$$D_\Phi(x,y) = \Phi(x) - \Phi(y) - \langle \nabla\Phi(y), x - y\rangle.$$

Upon expanding $D_\Phi(w,v) - D_\Phi(w,u) - D_\Phi(u,v)$ and rearranging terms gives

$$\Phi(w) - \Phi(v) - \langle \nabla\Phi(v), w - v\rangle - \Phi(w) + \Phi(u) + \langle \nabla\Phi(u), w - u\rangle - \Phi(u) + \Phi(v) + \langle \nabla\Phi(v), u - v\rangle,$$

which simplifies to

$$\langle \nabla\Phi(u) - \nabla\Phi(v),\, w - u\rangle.$$

□

Lemma 5. *Let $\{a_i\}_{i\geq 1}$ be a sequence satisfying $a_{i+1} \leq (1-\eta)a_i + b$, where $\eta \in (0,1)$. Then, for all $i \geq 1$,*

$$a_i \leq (1-\eta)^{i-1} a_1 + b \sum_{k=0}^{i-2} (1-\eta)^k.$$

Proof. We prove by induction on i. For $i = 1$, the bound reduces to $a_1 \leq a_1$, which holds trivially. Next, assume the statement holds for some $i \geq 1$. Using the recursion and the induction hypothesis,

$$\begin{aligned} a_{i+1} &\leq (1-\eta)\left[(1-\eta)^{i-1}a_1 + b\sum_{k=0}^{i-2}(1-\eta)^k\right] + b \\ &= (1-\eta)^i a_1 + b\sum_{k=0}^{i-2}(1-\eta)^{k+1} + b \\ &= (1-\eta)^i a_1 + b\sum_{k=0}^{i-1}(1-\eta)^k. \end{aligned}$$

This establishes the claim for $i + 1$ and completes the proof. □

B Theoretical Results

B.1 Lack of Monotonic Improvement Guarantees under f-Divergences

Proposition 1. *Let π_T be a fixed teacher and π_θ a student policy. For the softmax parameterization, the Natural Policy Gradient (NPG) step minimizing an f-divergence $D_f(\pi_T\|\pi_\theta)$, i.e., $\theta' = \theta - \eta\, F(\theta)^{-1}\nabla_\theta D_f(\pi_T\|\pi_\theta)$, satisfies*

$$J(\pi_{\theta'}) = J(\pi_\theta) - \eta\, \mathbb{C}\mathrm{ov}_{y\sim\pi_\theta}\left(r(y), g(y)\right) + O(\eta^2),$$

where $g(y) = f\left(\frac{\pi_T(y)}{\pi_\theta(y)}\right) - \frac{\pi_T(y)}{\pi_\theta(y)}\, f'\left(\frac{\pi_T(y)}{\pi_\theta(y)}\right)$. Consequently, for sufficiently small $\eta > 0$, the NPG step provides a policy improvement if and only if $\mathbb{C}\mathrm{ov}_{y\sim\pi_\theta}\left(r(y), g(y)\right) < 0$. In particular, $\Delta > 0$ does not suffice to guarantee this condition.

Proof. Let us define $p_a := \pi_\theta(a)$, $q_a := \pi_T(a)$, and $u_a := \frac{q_a}{p_a}$. Then, we have

$$\mathcal{L}(\theta) := D_f(\pi_T\|\pi_\theta) = \sum_{a\in\mathcal{A}} p_a\, f(u_a).$$

Next, we compute the gradient of $\mathcal{L}(\theta)$. By the chain rule,

$$\frac{\partial L(\theta)}{\partial p_a} = \frac{\partial}{\partial p_a}\left[p_a f\left(\frac{q_a}{p_a}\right)\right] = f(u_a) - u_a f'(u_a) = g(a),$$

where we used the definition of $g(a)$ from the statement in the Proposition. Next, we define the score vector $s_a := \nabla_\theta \log \pi_\theta(a)$. Using $\nabla_\theta p_a = p_a s_a$ and linearity, we have

$$\nabla_\theta \mathcal{L}(\theta) = \sum_{b\in\mathcal{A}} \frac{\partial \mathcal{L}(\theta)}{\partial p_b}\, \nabla_\theta p_b = \sum_{b\in\mathcal{A}} g(b)\, p_b\, s_b = \mathbb{E}_{b\sim\pi_\theta}\left[g(b) s_b\right].$$

Now, the natural gradient update is given as $\theta' - \theta = -\eta F(\theta)^{-1}\nabla_\theta\mathcal{L}(\theta)$. Upon expanding the policy map to first order gives

$$\pi_{\theta'}(a) = \pi_\theta(a) + \nabla_\theta \pi_\theta(a)^\top(\theta' - \theta) + O(\eta^2).$$

Since $\nabla_\theta \pi_\theta(a) = \pi_\theta(a) s_a$ and using the gradient update, we get

$$\pi_{\theta'}(a) = \pi_\theta(a) - \eta\, \pi_\theta(a)\, s_a^\top F(\theta)^{-1}\nabla_\theta\mathcal{L}(\theta) + O(\eta^2).$$

Now, we substitute the expression for $\nabla_\theta\mathcal{L}(\theta)$ to get $s_a^\top F(\theta)^{-1}\nabla_\theta\mathcal{L}(\theta) = \mathbb{E}_{b\sim\pi_\theta}\left[g(b)\, s_a^\top F(\theta)^{-1} s_b\right]$. By Lemma 1,

$$s_a^\top F(\theta)^{-1} s_b = \frac{\mathbf{1}\{a = b\}}{\pi_\theta(b)} - 1.$$

Therefore,

$$
\begin{aligned}
s_a^\top F(\theta)^{-1}\nabla_\theta \mathcal{L}(\theta) &= \mathbb{E}_{b\sim\pi_\theta}\left[g(b)\left(\frac{\mathbf{1}\{a=b\}}{\pi_\theta(b)} - 1\right)\right] \\
&= \mathbb{E}_{b\sim\pi_\theta}\left[g(b)\frac{\mathbf{1}\{a=b\}}{\pi_\theta(b)}\right] - \mathbb{E}_{b\sim\pi_\theta}[g(b)] \\
&= g(a) - \mathbb{E}_{b\sim\pi_\theta}[g(b)].
\end{aligned}
$$

Thus, we have

$$
\pi_{\theta'}(a) = \pi_\theta(a) - \eta\,\pi_\theta(a)\Big(g(a) - \mathbb{E}_{b\sim\pi_\theta}[g(b)]\Big) + O(\eta^2).
$$

Next, we consider that change in value of the policy following gradient update. Upon using the definition of $J(\pi)$, we have

$$
\begin{aligned}
J(\pi_{\theta'}) - J(\pi_\theta) &= \sum_{a\in\mathcal{A}} r(a)(\pi_{\theta'}(a) - \pi_\theta(a)) \\
&= -\eta \sum_{a\in\mathcal{A}} \pi_\theta(a)\, r(a)\Big(g(a) - \mathbb{E}_{b\sim\pi_\theta}[g(b)]\Big) + O(\eta^2) \\
&= -\eta\left(\mathbb{E}_{a\sim\pi_\theta}[r(a)g(a)] - \mathbb{E}_{a\sim\pi_\theta}[r(a)]\,\mathbb{E}_{a\sim\pi_\theta}[g(a)]\right) + O(\eta^2) \\
&= -\eta\,\mathbb{C}\mathrm{ov}_{a\sim\pi_\theta}(r(a), g(a)) + O(\eta^2).
\end{aligned}
$$

Hence, we have $J(\pi_{\theta'}) - J(\pi_\theta) - \eta\,\mathbb{C}\mathrm{ov}_{y\sim\pi_\theta}(r(y), g(y)) + O(\eta^2)$. □

B.2 Existence of Non-Monotonic Improvement Cases for Reverse-KL

Proposition 2. *There exists a contextual bandit setting such that, even when the teacher policy is strictly better than the student policy (i.e., $\Delta > 0$), an NPG step minimizing the reverse KL divergence, as in SDPO, does not lead to policy improvement.*

Proof. We provide our construction as follows. We consider a single-context multi-armed bandit with three actions and rewards given as

$$
r(1) = 1, \quad r(2) = 1, \quad r(3) = 0.
$$

Let the student and teacher policies be

$$
\pi_\theta = (0.05,\, 0.10,\, 0.85), \qquad \pi = (0.15,\, 0.05,\, 0.80).
$$

Then, we have

$$
J(\pi_\theta) = 0.15, \qquad J(\pi) = 0.20,
$$

which implies $\Delta > 0$.

Next, we consider the NPG update for reverse-KL which is given as

$$
\pi_{\theta'}(y) = \pi_\theta(y) - \eta\,\pi_\theta(y)\left(\log\frac{\pi_\theta(y)}{\pi(y)} - \mathrm{KL}(\pi_\theta\|\pi)\right).
$$

Let us now focus on the suboptimal action $y = 3$. We compute

$$
\log\frac{\pi_\theta(3)}{\pi(3)} \approx 0.0606, \qquad \mathrm{KL}(\pi_\theta\|\pi) \approx 0.0659,
$$

This gives

$$
\log\frac{\pi_\theta(3)}{\pi(3)} - \mathrm{KL} \approx -0.0053.
$$

Thus,

$$
\pi_{\theta'}(3) = \pi_\theta(3) - \eta\cdot(-0.0045) > \pi_\theta(3).
$$

Therefore, the probability of the suboptimal action increases after the update, leading to a degradation in performance. Hence, even when $\Delta > 0$, reverse KL NPG does not guarantee monotonic improvement. □

B.3 Local Credit Assignment leads to Sub-Optimal Policies

To show that Proposition 3 holds, we provide a detailed construction in Proposition 7.

Proposition 7. *Let us consider the action-space $\mathcal{A} = \{a, b, c\}$ and consider a two-step decision process $(y_1, y_2) \in \mathcal{A}^2$. We define the student policy $\pi_{u,v}$ by*

$$\begin{aligned}
\pi_{u,v}(y_1 = \cdot) &= \mathrm{softmax}(u, 0, 0), \\
\pi_{u,v}(y_2 = \cdot \mid y_1 = a) &= \mathrm{softmax}(v, 0, 0), \\
\pi_{u,v}(y_2 = \cdot \mid y_1 = b) &= U, \\
\pi_{u,v}(y_2 = \cdot \mid y_1 = c) &= U,
\end{aligned}$$

where U is the uniform distribution over actions. Let us define the teacher policy π_T as

$$\pi_T(y_1) = U, \qquad \pi_T(\cdot \mid y_1 = a) = R, \qquad \pi_T(\cdot \mid y_1 = b) = \pi_T(\cdot \mid y_1 = c) = U,$$

where R is any distribution satisfying $\inf_v \mathrm{KL}(\mathrm{softmax}(v, 0, 0) \| R) = D_{\min} > 0$. Next, we define the sequence-level objective

$$\mathcal{L}_{\mathrm{seq}}(u, v) = \mathrm{KL}(\pi_{u,v}(y_1, y_2) \| \pi_T(y_1, y_2)).$$

Let $\mathcal{L}_{\mathrm{curr}}$ denote the token-wise objective that sums per-step KL terms but treats the prefix distribution as fixed when differentiating (for example, in SDPO/OPSD).

Then, we have

1. *At $(u, v) = (0, 0)$,*
$$\frac{\partial \mathcal{L}_{\mathrm{curr}}}{\partial u}(0, 0) = 0, \qquad \frac{\partial \mathcal{L}_{\mathrm{seq}}}{\partial u}(0, 0) > 0.$$
2. *Token-wise updates leaves $u = 0$. At the resulting solution v^*, the true objective satisfies*
$$\mathcal{L}_{\mathrm{seq}}(0, v^*) = \frac{1}{3} D_{\min} > 0.$$
3. *In contrast,*
$$\inf_{u,v} \mathcal{L}_{\mathrm{seq}}(u, v) = 0.$$

Thus, token-wise gradients provide no learning signal for the first-step decision and lead to a strictly suboptimal solution, whereas sequence-level optimization admits strictly better solutions.

Proof. We start by using the chain rule for KL divergence which gives

$$\mathcal{L}_{\mathrm{seq}}(u, v) = \mathrm{KL}(\pi_{u,v}(y_1) \| U) + \mathbb{E}_{y_1 \sim \pi_{u,v}} \mathrm{KL}(\pi_{u,v}(y_2 \mid y_1) \| \pi_T(y_2 \mid y_1)).$$

Let us define $p(u) := \pi_{u,v}(y_1 = a) = \frac{e^u}{e^u + 2}$. Then, we have

$$\pi_{u,v}(y_1) = \left(p(u), \frac{1 - p(u)}{2}, \frac{1 - p(u)}{2} \right).$$

Upon simplification, we get

$$\mathrm{KL}(\pi_{u,v}(y_1) \| U) = K(p(u)),$$

where

$$K(p) = p \log(3p) + (1 - p) \log\left(\frac{3(1 - p)}{2} \right).$$

Next, for $y_1 \in \{b, c\}$, both student and teacher distributions are U, so the KL divergence is zero. For $y_1 = a$, let us denote the KL divergence as $D(v)$, where $D(v) = \mathrm{KL}(\mathrm{softmax}(v, 0, 0) \| R)$. Hence, we have $\mathbb{E}_{y_1 \sim \pi_{u,v}} \mathrm{KL}(\pi_{u,v}(y_2 \mid y_1) \| \pi_T(y_2 \mid y_1)) = \mathbb{E}_{y_1} \mathrm{KL}(\cdot) = p(u) D(v)$, and therefore

$$\mathcal{L}_{\mathrm{seq}}(u, v) = K(p(u)) + p(u) D(v).$$

Next, we compute the gradient of sequence-level objective. Upon calculation, we have

$$K'(p) = \log \frac{2p}{1-p}, \qquad p'(u) = \frac{2e^u}{(e^u+2)^2}.$$

At $u = 0$, we have $p(0) = 1/3$, so

$$K'(1/3) = \log 1 = 0, \qquad p'(0) = \frac{2}{9}.$$

Therefore, we have

$$\frac{\partial \mathcal{L}_{\text{seq}}}{\partial u}(0,0) = K'(1/3)p'(0) + p'(0)D(0) = \frac{2}{9}D(0) > 0.$$

Next, we focus on the token-wise objective that treats $p(u)$ as fixed when differentiating the second term,

$$\frac{\partial \mathcal{L}_{\text{curr}}}{\partial u} = \frac{d}{du}K(p(u)).$$

Thus, we get

$$\frac{\partial \mathcal{L}_{\text{curr}}}{\partial u}(0,0) = 0.$$

Therefore, token-wise updates keep $u = 0$ and minimize $D(v)$, giving $D(v^*) = D_{\min}$. Hence,

$$\mathcal{L}_{\text{seq}}(0, v^*) = \frac{1}{3}D_{\min}.$$

On the other hand, as $u \to -\infty$, we have $p(u) \to 0$, and therefore

$$L_{\text{seq}}(u, v) \to 0.$$

To conclude, token-wise optimization is stuck at a strictly suboptimal value, while sequence-level optimization can achieve arbitrarily small loss. □

B.4 DistIL's Gradient Derivation

Proposition 4. *The gradient of $\mathcal{L}_{\text{DistIL}}(\theta)$ is given by The gradient of $\mathcal{L}_{\text{DistIL}}(\theta)$ is given by*

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\text{DistIL}} =& \mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta H^{\times}\Big(\mathsf{sg}(\pi_\theta(\cdot \mid s_t, f)),\ \pi_\theta(\cdot \mid s_t)\Big)\right] \\ &+ \mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\left(\sum_{i>t} H^{\times}\Big(\mathsf{sg}(\pi_\theta(\cdot \mid s_i, f)),\ \pi_\theta(\cdot \mid s_i)\Big)\right)\right],\end{aligned}$$

where $s_t = (x, y_{1:t-1})$.

Proof. We start by defining the per-step loss

$$\ell_t(y,\theta) := H^{\times}\Big(\mathsf{sg}(\pi_\theta(\cdot \mid x, y_{1:t-1}, f)),\ \pi_\theta(\cdot \mid x, y_{1:t-1})\Big),$$

so that

$$\mathcal{L}_{\text{DistIL}}(\theta) = \mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{H} \ell_t(y,\theta)\right].$$

By Lemma 2, applied to the trajectory distribution $y \sim \pi_\theta$, we have

$$\nabla_\theta \mathcal{L}_{\text{DistIL}}(\theta) = \mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta \ell_t(y,\theta)\right] + \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=1}^{H} \ell_t(\tau,\theta)\right)\nabla_\theta \log \pi_\theta(\tau)\right]. \tag{6}$$

Since the policy is autoregressive, we have $\log \pi_\theta(\tau) = \sum_{t=1}^{H} \log \pi_\theta(a_t \mid s_t)$ and $\nabla_\theta \log \pi_\theta(\tau) = \sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_t \mid s_t)$. Upon using this in Eq. 6 gives

$$\nabla_\theta \mathcal{L}_{\text{DistIL}}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta \ell_t(\tau, \theta)\right] + \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=1}^{H} \sum_{i=1}^{H} \ell_i(\tau, \theta)\, \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right].$$

We now show that only future terms $i > t$ survive in the second expectation. Let us fix t and consider any $i \leq t$. The quantity $\ell_i(\tau, \theta)$ depends only on the prefix up to time i, and therefore is independent of the sampled action a_t once the state s_t is fixed. Consequently,

$$\mathbb{E}_{a_t \sim \pi_\theta(\cdot \mid s_t)}\Big[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \ell_i(\tau, \theta)\Big] = \ell_i(\tau, \theta)\, \mathbb{E}_{a_t \sim \pi_\theta(\cdot \mid s_t)}\Big[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big].$$

By Lemma 2,

$$\mathbb{E}_{a_t \sim \pi_\theta(\cdot \mid s_t)}\Big[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\Big] = 0,$$

so the contribution from all $i \leq t$ vanishes. Therefore,

$$\mathbb{E}_{\tau \sim \pi_\theta}\left[\left(\sum_{t=1}^{H} \ell_t(\tau, \theta)\right) \nabla_\theta \log \pi_\theta(\tau)\right] = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \left(\sum_{i > t} \ell_i(\tau, \theta)\right)\right].$$

Substituting back the definition of ℓ_i gives

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\text{DistIL}} =& \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta H^{\times}\Big(\mathsf{sg}(\pi_\theta(\cdot \mid s_t, f)),\ \pi_\theta(\cdot \mid s_t)\Big)\right] \\ &+ \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=1}^{H} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \left(\sum_{i > t} H^{\times}\Big(\mathsf{sg}(\pi_\theta(\cdot \mid s_i, f)),\ \pi_\theta(\cdot \mid s_i)\Big)\right)\right].\end{aligned}$$

□

B.5 Tangent-Space Realizability

Proposition 8. *Let $\mathcal{S}$ and $\mathcal{A}$ be finite sets, and consider a tabular softmax policy*

$$\pi_\theta(a \mid s) = \frac{\exp(\theta(s, a))}{\sum_{a' \in \mathcal{A}} \exp(\theta(s, a'))}.$$

Then, for any fixed θ and any teacher policy π_T, there exists a vector $u : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ such that for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$,

$$\pi_T(a \mid s) - \pi_\theta(a \mid s) = \pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s)^\top u.$$

Proof. We start by fixing $s \in \mathcal{S}$. Since $\pi_\theta(\cdot \mid s)$ is a softmax distribution, $\pi_\theta(a \mid s) > 0$ for all $a \in \mathcal{A}$, so the ratio $\pi_T(a \mid s)/\pi_\theta(a \mid s)$ is well-defined. For tabular softmax, the score function satisfies

$$\frac{\partial \log \pi_\theta(a \mid s)}{\partial \theta(s, a')} = \mathbf{1}\{a' = a\} - \pi_\theta(a' \mid s).$$

Thus, for any vector u, we have

$$\nabla_\theta \log \pi_\theta(a \mid s)^\top u = u(s, a) - \sum_{a' \in \mathcal{A}} \pi_\theta(a' \mid s)\, u(s, a').$$

Now, let us define

$$u(s, a) := \frac{\pi_T(a \mid s)}{\pi_\theta(a \mid s)}.$$

Then, we have

$$\sum_{a'} \pi_\theta(a' \mid s)\, u(s, a') = \sum_{a'} \pi_T(a' \mid s) = 1.$$

Hence,

$$\nabla_\theta \log \pi_\theta(a \mid s)^\top u = \frac{\pi_T(a \mid s)}{\pi_\theta(a \mid s)} - 1.$$

Multiplying both sides by $\pi_\theta(a \mid s)$ gives

$$\pi_\theta(a \mid s)\, \nabla_\theta \log \pi_\theta(a \mid s)^\top u = \pi_T(a \mid s) - \pi_\theta(a \mid s).$$

□

B.6 Monotonic Improvement of DistIL

To show that Proposition 5 holds, we provide the assumptions made in it, along with the complete proof in Proposition 9.

Proposition 9. *Let us consider our objective $\mathcal{L}_{\mathrm{DistIL}}(\theta)$. Assume the following holds*

1. *$\theta \mapsto \pi_\theta$ is twice continuously differentiable,*
2. *the reward $r(x, y)$ is bounded,*
3. *the teacher is strictly better than the student:*
$$\Delta := \mathbb{E}_{x,y\sim\pi_T}[r(x,y)] - \mathbb{E}_{x,y\sim\pi_\theta}[r(x,y)] > 0,$$
4. *(Tangent assumption) there exists $u \in \mathbb{R}^d$ such that for all (x, y),*
$$\pi_T(y \mid x) - \pi_\theta(y \mid x) = \pi_\theta(y \mid x)\, \nabla_\theta \log \pi_\theta(y \mid x)^\top u.$$

Then, the natural gradient update $\theta' = \theta - \eta F(\theta)^{-1} \nabla_\theta \mathcal{L}_{\mathrm{DistIL}}(\theta)$ satisfies

$$\mathbb{E}_{x,y\sim\pi_{\theta'}}[r(x,y)] = \mathbb{E}_{x,y\sim\pi_\theta}[r(x,y)] + \eta\Delta + O(\eta^2).$$

Equivalently, we have $J(\pi_{\theta'}) = J(\pi_\theta) + \eta\Delta + O(\eta^2)$.

Proof. Under the frozen state distribution, the gradient of $\mathcal{L}_{\mathrm{DistIL}}$ is given as

$$\nabla_\theta \mathcal{L}_{\mathrm{DistIL}}(\theta) = -\mathbb{E}_{x\sim\rho,\, y\sim\pi_T(\cdot|x)} \left[\nabla_\theta \log \pi_\theta(y \mid x)\right].$$

From Lemma 2, $\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[\nabla_\theta \log \pi_\theta(y \mid x)] = 0$. Therefore, the gradient can also be written as

$$\nabla_\theta \mathcal{L}_{\mathrm{DistIL}}(\theta) = -\mathbb{E}_{x\sim\rho} \left[\sum_y \left(\pi_T(y \mid x) - \pi_\theta(y \mid x)\right) \nabla \log \pi_\theta(y \mid x)\right].$$

Next, we apply the tangent assumption $\pi_T(y \mid x) - \pi_\theta(y \mid x) = \pi_\theta(y \mid x)\, \nabla_\theta \log \pi_\theta(y \mid x)^\top u$ to obtain

$$\nabla_\theta \mathcal{L}_{\mathrm{DistIL}}(\theta) = -\mathbb{E}_{x\sim\rho} \left[\sum_y \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x)^\top\right] u.$$

We now recognize the Fisher information matrix,

$$F(\theta) = \mathbb{E}_{x\sim\rho,\, y\sim\pi_\theta(\cdot|x)} \left[\nabla_\theta \log \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x)^\top\right]$$

to conclude

$$\nabla_\theta \mathcal{L}_{\mathrm{DistIL}}(\theta) = -F(\theta) u.$$

Thus, the natural gradient update becomes

$$\theta' = \theta + \eta u.$$

We now analyze the change in the policy using a first-order Taylor expansion

$$\pi_{\theta'}(y \mid x) = \pi_\theta(y \mid x) + \nabla_\theta \pi_\theta(y \mid x)^\top (\theta' - \theta) + O(\eta^2).$$

Substituting $\theta' - \theta = \eta u$ gives

$$\pi_{\theta'}(y \mid x) = \pi_\theta(y \mid x) + \eta \nabla_\theta \pi_\theta(y \mid x)^\top u + O(\eta^2).$$

Using $\nabla_\theta \pi_\theta(y \mid x) = \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x)$, we obtain

$$\pi_{\theta'}(y \mid x) = \pi_\theta(y \mid x) + \eta \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x)^\top u + O(\eta^2).$$

Next, we again apply the tangent assumption to obtain

$$\pi_{\theta'}(y \mid x) = \pi_\theta(y \mid x) + \eta(\pi_T(y \mid x) - \pi_\theta(y \mid x)) + O(\eta^2).$$

Multiplying by $r(x, y)$, summing over y, and taking expectation over $x \sim \rho$, we obtain

$$\mathbb{E}_{x,y\sim\pi_{\theta'}}[r(x,y)] = \mathbb{E}_{x,y\sim\pi_\theta}[r(x,y)] + \eta\left(\mathbb{E}_{x,y\sim\pi_T}[r(x,y)] - \mathbb{E}_{x,y\sim\pi_\theta}[r(x,y)]\right) + O(\eta^2).$$

Finally, using the definition of Δ and $J(\pi)$, this becomes

$$J(\pi_{\theta'}) = J(\pi_\theta) + \eta\Delta + O(\eta^2),$$

Since $\Delta > 0$, for sufficiently small $\eta > 0$, the update gives strict improvement. □

B.7 Regret Analysis

Let us denote $\pi^\star := \pi_T$. We start by assuming that $\left|\frac{\pi^\star(a|s)}{\pi^i(a|s)}\right| \le C, \forall s \in \mathcal{S}, a \in \mathcal{A}$. This is true due to trust-region and also teacher is just feedback conditioned policy.

Theorem 2. *Let $\ell_i(\pi)$ be defined as*

$$\ell_i(\pi) := -\sum_{t=1}^{H} \mathbb{E}_{s_t\sim d_t^{\pi^i}}\left[\sum_{a\in\mathcal{A}} \pi^\star(a \mid s_t) \log \pi(a \mid s_t)\right].$$

Let us denote $\ell_s(\pi) = -\sum_{a\in\mathcal{A}} \pi^\star(a \mid s) \log \pi(a \mid s)$ for any state $s \in \mathcal{S}$ and $\{\pi^i\}_{i=1}^{n+1}$ be generated by

$$\pi^{i+1}(\cdot \mid s) = \arg\min_{q\in\Delta(\mathcal{A})} \left\{\langle \nabla \ell_s(\pi^i), q\rangle + \frac{1}{\eta} D_{\mathrm{KL}}(q \| \pi^i(\cdot \mid s))\right\}.$$

Under the assumption that

$$\left|\frac{\pi^\star(a|s)}{\pi^i(a|s)}\right| \le C \quad \forall s, a \quad \textit{and} \quad D_{\mathrm{KL}}(\pi^\star(\cdot \mid s) \| \pi^1(\cdot \mid s)) \le C_0, \forall s.$$

we have

$$\frac{1}{n}\sum_{i=1}^{n}\left(\ell_i(\pi^i) - \ell_i(\pi^\star)\right) \le \frac{HC_0}{\eta n} + \frac{\eta H C^2}{2}.$$

Therefore, upon choosing $\eta = \sqrt{\frac{2C_0}{C^2 n}}$, we obtain

$$\frac{1}{n}\sum_{i=1}^{n}(\ell_i(\pi^i) - \ell_i(\pi^\star)) \le HC\sqrt{\frac{2C_0}{n}}.$$

Proof. We start by observing from the definition of $\ell_s(\pi)$ that $\ell_i(\pi) = \sum_{t=1}^{H} \mathbb{E}_{s_t\sim d_t^{\pi^i}}[\ell_{s_t}(\pi)]$. Let us consider the loss difference

$$\ell_s(\pi) - \ell_s(\pi^\star) = -\sum_{a\in\mathcal{A}} \pi^\star(a \mid s) \log \pi(a \mid s) + \sum_{a\in\mathcal{A}} \pi^\star(a \mid s) \log \pi^\star(a \mid s) = D_{\mathrm{KL}}(\pi^\star(\cdot \mid s) \,\|\, \pi(\cdot \mid s)).$$

Thus,

$$\ell_i(\pi) - \ell_i(\pi^\star) = \sum_{t=1}^{H} \mathbb{E}_{s_t\sim d_t^{\pi^i}}\left[D_{\mathrm{KL}}(\pi^\star(\cdot \mid s_t) \,\|\, \pi(\cdot \mid s_t))\right]. \tag{7}$$

We use shorthand $\pi^i(s) := \pi^i(\cdot \mid s)$ and $D_i(s) := D_{\mathrm{KL}}(\pi^\star(\cdot \mid s) \,\|\, \pi^i(\cdot \mid s))$. Let us now consider the mirror map to be the negative entropy, so that the associated Bregman divergence is the KL divergence. We run the state-conditioned mirror descent update:

$$\pi^{i+1}(s) \in \arg\min_{q \in \Delta(\mathcal{A})} \left\{ \langle \nabla \ell_s(\pi^i), q \rangle + \frac{1}{\eta} D_{\mathrm{KL}}(q \| \pi^i(s)) \right\}.$$

From Lemma 3, we have

$$\langle \eta \nabla \ell_s(\pi^i) + \nabla \Phi(\pi^{i+1}(s)) - \nabla \Phi(\pi^i(s)),\, \pi^\star(s) - \pi^{i+1}(s) \rangle \geq 0.$$

Upon rearranging, we have

$$\eta \langle \nabla \ell_s(\pi^i), \pi^{i+1}(s) - \pi^\star(s) \rangle \leq \langle \nabla \Phi(\pi^{i+1}(s)) - \nabla \Phi(\pi^i(s)),\, \pi^\star(s) - \pi^{i+1}(s) \rangle.$$

Next, we use Lemma 4 to obtain

$$\eta \langle \nabla \ell_s(\pi^i), \pi^{i+1}(s) - \pi^\star(s) \rangle \leq D_i(s) - D_{i+1}(s) - D_{\mathrm{KL}}(\pi^{i+1}(s) \| \pi^i(s)). \tag{8}$$

Now, from assumption, we have,

$$\left| \frac{\partial \ell_s(\pi^i)}{\partial \pi(a \mid s)} \right| = \left| \frac{\pi^\star(a|s)}{\pi^i(a|s)} \right| \leq C.$$

Next, we define

$$\Delta^i(s) := \pi^i(s) - \pi^{i+1}(s).$$

Then

$$\eta \langle \nabla \ell_s(\pi^i), \pi^i - \pi^{i+1} \rangle \leq \eta C \|\Delta^i(s)\|_1.$$

Upon applying Young's inequality ($ab \leq \frac{a^2}{2} + \frac{b^2}{2}$), with $a = \eta C$ and $b = \Delta^i(s)$, we get

$$\eta \langle \nabla \ell_s(\pi^i), \pi^i - \pi^{i+1} \rangle \leq \frac{\eta^2 C^2}{2} + \frac{1}{2} \|\Delta^i(s)\|_1^2.$$

Using Pinsker's inequality,

$$\frac{1}{2} \|\Delta^i(s)\|_1^2 \leq D_{\mathrm{KL}}(\pi^{i+1}(s) \| \pi^i(s)).$$

Thus,

$$\eta \langle \nabla \ell_s(\pi^i), \pi^i(s) - \pi^{i+1}(s) \rangle \leq \frac{\eta^2 C^2}{2} + D_{\mathrm{KL}}(\pi^{i+1}(s) \| \pi^i(s)). \tag{9}$$

Now, from convexity of ℓ_s, we have,

$$\eta(\ell_s(\pi^i) - \ell_s(\pi^\star)) \leq \eta \langle \nabla \ell_s(\pi^i),\, \pi^i(s) - \pi^{i+1}(s) \rangle + \eta \langle \nabla \ell_s(\pi^i),\, \pi^{i+1}(s) - \pi^\star(s) \rangle.$$

Upon combining Equations 8 and 9, we obtain

$$\eta(\ell_s(\pi^i) - \ell_s(\pi^\star)) \leq \frac{\eta^2 C^2}{2} + D_i(s) - D_{i+1}(s).$$

Now, we reuse the definition of $D_i(s) = \ell_s(\pi^i) - \ell_s(\pi^\star)$, this gives

$$\eta D_i(s) \leq \frac{\eta^2 C^2}{2} + D_i(s) - D_{i+1}(s).$$

Thus,

$$D_{i+1}(s) \leq (1 - \eta) D_i(s) + \frac{\eta^2 C^2}{2}. \tag{10}$$

Next, we simplify this recursion. In particular, for a given state s, let us define

$$a_i := D_i(s), \quad b := \frac{\eta^2 C^2}{2}.$$

Therefore, Equation 10 can be re-written as

$$a_{i+1} \leq (1 - \eta) a_i + b.$$

Using Lemma 5, we get

$$D_i(s) \leq (1-\eta)^{i-1} D_1(s) + \frac{\eta^2 C^2}{2} \sum_{k=0}^{i-2} (1-\eta)^k, \quad \forall i \geq 1.$$

Next, we bound the geometric sum:

$$\sum_{k=0}^{i-2} (1-\eta)^k \leq \sum_{k=0}^{\infty} (1-\eta)^k = \frac{1}{\eta}.$$

Therefore,

$$D_i(s) \leq (1-\eta)^{i-1} D_1(s) + \frac{\eta^2 C^2}{2} \cdot \frac{1}{\eta} = (1-\eta)^{i-1} D_1(s) + \frac{\eta C^2}{2}.$$

Using $D_1(s) \leq C_0$, we obtain

$$D_i(s) \leq (1-\eta)^{i-1} C_0 + \frac{\eta C^2}{2}.$$

Now, considering Eq. 7, we have

$$\ell_i(\pi^i) - \ell_i(\pi^\star) = \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_t^{\pi^i}} [D_i(s_t)] \leq H(1-\eta)^{i-1} C_0 + H \frac{\eta C^2}{2}.$$

Upon summing from $i = 1$ to n gives

$$\begin{aligned}
\sum_{i=1}^{n} (\ell_i(\pi^i) - \ell_i(\pi^\star)) &\leq H \sum_{i=1}^{n} \left((1-\eta)^{i-1} C_0 + \frac{\eta C^2}{2} \right) \\
&= \frac{n\eta H C^2}{2} + H C_0 \sum_{i=1}^{n} (1-\eta)^{i-1} \\
&\leq \frac{n\eta H C^2}{2} + \frac{H C_0}{\eta} \qquad \left(\because \sum_{i=1}^{n} (1-\eta)^{i-1} \leq \frac{1}{\eta} \right)
\end{aligned}$$

Upon dividing by n gives

$$\frac{\sum_{i=1}^{n} (\ell_i(\pi^i) - \ell_i(\pi^\star))}{n} \leq \frac{\eta H C^2}{2} + \frac{H C_0}{\eta n}$$

Finally, choosing $\eta = \sqrt{\frac{2C_0}{C^2 n}}$ completes the proof. □

We now introduce the quantities that characterize the difficulty of imitation under a stochastic teacher, following Foster et al. [12].

Teacher variance and recoverability. Following Foster et al. [12], for any policy π, define

$$\bar{\sigma}^2_{\pi_T|\pi} := \sum_{h=1}^{H} \mathbb{E}_{P^{\pi \circ_h \pi_T}} \left[(Q_h^{\pi_T}(x_h, a_h) - V_h^{\pi_T}(x_h))^2 \right],$$

where $P^{\pi \circ_h \pi_T}$ denotes the trajectory distribution that follows π up to time $h-1$ and switches to π_T thereafter. We then define the worst-case teacher variance

$$\bar{\sigma}^2_{\pi_T} := \sup_{\pi} \bar{\sigma}^2_{\pi_T|\pi}.$$

We also define the signed recoverability parameter

$$\mu_T := \max_{x \in \mathcal{X},\, a \in \mathcal{A},\, h \in [H]} \left| Q_h^{\pi_T}(x, \pi_{T,h}(x)) - Q_h^{\pi_T}(x, a) \right|.$$

Remarks. The quantity $\bar{\sigma}^2_{\pi_T}$ captures the stochasticity of the teacher policy, and in particular, $\bar{\sigma}^2_{\pi_T} = 0$ when π_T is deterministic. The constant μ_T measures the worst-case degradation in value from taking a suboptimal action under the teacher's value function, and satisfies $0 \leq \mu_T \leq R$ when rewards are bounded by R. Together, these quantities characterize the difficulty of imitation. When μ_T is small, the environment is forgiving and errors are easily recoverable; in contrast, large μ_T corresponds to settings where early mistakes can lead to significant long-term loss. Moreover, the two quantities are related via the bound $\bar{\sigma}^2_{\pi_T} \leq \mu_T^2 H$, which implies that the stochasticity of the teacher is controlled by the recoverability of the environment.

Theorem 1. *Let π_T be the teacher policy and π_S, π_S^A be the student policies returned by the updates in Theorem 2. Then, for the per-iteration loss $\ell_i(\pi) = -\sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_t^{\pi^i}} \Big[\sum_{a \in \mathcal{A}} \pi_T(a \mid s_t) \log \pi(a \mid s_t) \Big]$, we have*

$$J(\pi_T) - J(\pi_S) \;\lesssim\; \sqrt{\bar{\sigma}^2_{\pi_T} \frac{HC}{2}} \left(\frac{2C_0}{n}\right)^{1/4} + \frac{\mu_T H C}{2} \sqrt{\frac{2C_0}{n}}.$$

Proof. Let us start by proving the first case. For a fixed state s_t, we define the statewise cross-entropy

$$\ell_{s_t}(\pi) := -\sum_{a \in \mathcal{A}} \pi_T(a \mid s_t) \log \pi(a \mid s_t).$$

Then, $\ell_{s_t}(\pi) - \ell_{s_t}(\pi_T) = D_{\mathrm{KL}}(\pi_T(\cdot \mid s_t) \,\|\, \pi(\cdot \mid s_t))$. Therefore, using the definition of ℓ_i,

$$\ell_i(\pi^i) - \ell_i(\pi_T) = \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_t^{\pi^i}} \Big[D_{\mathrm{KL}}(\pi_T(\cdot \mid s_t) \,\|\, \pi^i(\cdot \mid s_t)) \Big].$$

Averaging over i gives

$$\epsilon_n := \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_t^{\pi^i}} \Big[D_{\mathrm{KL}}(\pi_T(\cdot \mid s_t) \,\|\, \pi^i(\cdot \mid s_t)) \Big].$$

Now, for any pair of distributions (p, q), we have $D^2_{\mathrm{H}}(p, q) \leq \frac{1}{2} D_{\mathrm{KL}}(p \| q)$, where D_{H} is the Hellinger distance. Applying this pointwise with $p = \pi_T(\cdot \mid s_t)$ and $q = \pi^i(\cdot \mid s_t)$, then taking expectation over $s_t \sim d_t^{\pi^i}$, summing over t, and averaging over i, gives

$$\mathcal{E}_n := \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{H} \mathbb{E}_{s_t \sim d_t^{\pi^i}} \Big[D^2_{\mathrm{H}}(\pi_T(\cdot \mid s_t), \pi^i(\cdot \mid s_t)) \Big] \leq \frac{1}{2} \epsilon_n.$$

Next, we make use of the Proposition C.1 of [12], which works for our online setting, to bound the rollout suboptimality by a variance term and a recoverability term as

$$J(\pi_T) - J(\pi_S) \;\lesssim\; \sqrt{\bar{\sigma}^2_{\pi_T} \, \mathcal{E}_n} + \mu_e \, \mathcal{E}_n.$$

Substituting $\mathcal{E}_n \leq \epsilon_n / 2$ gives

$$J(\pi_T) - J(\pi_b) \;\lesssim\; \sqrt{\frac{\bar{\sigma}^2_{\pi_T}}{2} \, \epsilon_n} + \frac{\mu_T}{2} \, \epsilon_n.$$

Finally, plugging in the Theorem 2 gives

$$J(\pi_T) - J(\pi_b) \;\lesssim\; \sqrt{\bar{\sigma}^2_{\pi_T} \frac{HC}{2}} \left(\frac{2C_0}{n}\right)^{1/4} + \frac{\mu_T H C}{2} \sqrt{\frac{2C_0}{n}}.$$

□

B.8 Connections with Max RL

We show that the forward cross-entropy objective induces a lower bound on a teacher-weighted maximum likelihood RL objective.

Proposition 6. *For a fixed prompt x, we define the success probability $p_\pi(x) := \mathbb{E}_{y\sim\pi(\cdot|x)}[r(x,y)]$, where $r(x,y) \in \{0,1\}$ indicates correctness. Then*

$$-H^{\times}(\pi_T(\cdot \mid x), \pi_\theta(\cdot \mid x)) \le p_{\pi_T}(x) \log p_{\pi_\theta}(x),$$

where

$$p_{\pi_\theta}(x) := \mathbb{P}_{y\sim\pi_\theta(\cdot|x)}[r(x,y) = 1].$$

Proof. To show this, we begin with the definition of forward cross-entropy:

$$H^{\times}(\pi_T(\cdot \mid x), \pi_\theta(\cdot \mid x)) = \sum_y \pi_T(y \mid x)[-\log \pi_\theta(y \mid x)].$$

Next, we partition the sum into correct and incorrect responses:

$$H^{\times}(\pi_T(\cdot \mid x), \pi_\theta(\cdot \mid x)) = \sum_{y:r(x,y)=1} \pi_T(y \mid x)[-\log \pi_\theta(y \mid x)] + \sum_{y:r(x,y)=0} \pi_T(y \mid x)[-\log \pi_\theta(y \mid x)].$$

For any y such that $r(x,y) = 1$, we can write

$$\pi_\theta(y \mid x) = p_{\pi_\theta}(x)\, \pi_\theta(y \mid x, r = 1),$$

where

$$p_{\pi_\theta}(x) = \mathbb{P}_{y\sim\pi_\theta(\cdot|x)}[r(x,y) = 1].$$

Thus,

$$-\log \pi_\theta(y \mid x) = -\log p_{\pi_\theta}(x) - \log \pi_\theta(y \mid x, r = 1).$$

Substituting this into the first term, we get

$$\begin{aligned} H^{\times}(\pi_T(\cdot \mid x), \pi_\theta(\cdot \mid x)) = -p_{\pi_T}(x) \log p_{\pi_\theta}(x) + p_{\pi_T}(x)\, H^{\times}(\pi_T(\cdot \mid x, r = 1), \pi_\theta(\cdot \mid x, r = 1)) \\ + (1 - p_{\pi_T}(x))\, \mathbb{E}_{y\sim\pi_T(\cdot|x,r=0)}[-\log \pi_\theta(y \mid x)]. \end{aligned}$$

Next, we define

$$\mathcal{R}(x) := p_{\pi_T}(x)\, H^{\times}(\pi_T(\cdot \mid x, r = 1), \pi_\theta(\cdot \mid x, r = 1)) + (1 - p_{\pi_T}(x))\, \mathbb{E}_{y\sim\pi_T(\cdot|x,r=0)}[-\log \pi_\theta(y \mid x)].$$

Since both terms are nonnegative, we have $\mathcal{R}(x) \ge 0$. Therefore,

$$H^{\times}(\pi_T(\cdot \mid x), \pi_\theta(\cdot \mid x)) = -p_{\pi_T}(x) \log p_{\pi_\theta}(x) + \mathcal{R}(x) \ge -p_{\pi_T}(x) \log p_{\pi_\theta}(x).$$

Rearranging gives

$$-H^{\times}(\pi_T(\cdot \mid x), \pi_\theta(\cdot \mid x)) \le p_{\pi_T}(x) \log p_{\pi_\theta}(x).$$

□

C Extended Related Work

Credit Assignment in RLVR. RLVR enables training from automatically verifiable outcomes without human preference annotations [36, 14]. However, supervision is applied at the trajectory level, assigning identical signals to all tokens despite their differing roles, making credit assignment especially challenging for long-horizon reasoning. One line of work addresses this by introducing intermediate supervision via process-reward models or step-wise evaluators [25, 57, 41, 51, 28, 4, 56, 49, 10, 9]. While effective, these approaches require additional annotation or auxiliary models. An alternative direction avoids explicit supervision by reweighting updates using internal signals such as entropy or uncertainty, but these rely on heuristic notions of token importance [45, 24, 7, 42, 5, 38, 22, 6]. In our work, instead of estimating token importance heuristically or learning separate reward models, we construct token-level supervision directly through privileged

feedback. This yields structured credit assignment while preserving the simplicity of the verifier-only training pipeline.

On-Policy Distillation and On-Policy Self-Distillation. On-policy distillation trains a student on its own rollouts while a teacher provides token-level supervision via divergence-based objectives [1, 46, 27, 44, 48]. Aligning training with the student's visitation distribution reduces the distribution mismatch inherent in off-policy approaches, but it typically relies on a stronger external teacher. Recent work extends this paradigm to self-distillation, where a single model provides both supervision and updates. OPSD [59] formulates this as minimizing the divergence between a privileged teacher, which is conditioned on ground-truth reasoning traces, and a student, thereby improving token efficiency. SDPO [16] generalizes this setup to incorporate arbitrary feedback signals (e.g., code execution error logs), while RLSD [50] integrates self-distillation with the GRPO methodologically to stabilize training. However, self-distillation introduces new challenges. It can degrade reasoning by suppressing epistemic uncertainty [19], and its success depends critically on alignment between teacher and student distributions as well as the quality of self-generated targets [23]. Recent methods address these issues through improved supervision and routing strategies, including competence-aware weighting (PACED), iterative self-revision (SD-Zero), and hybrid routing between distillation and RL updates (SRPO) [47, 15, 21]. In contrast, our approach revisits the objective underlying self-distillation. We adopt a cross-entropy-based formulation with well-behaved optimization properties, resulting in more stable training dynamics and improved gradient estimation. This is related to methods such as DAgger [33] in imitation learning that query the expert for supervision on the learner's induced state distribution.

D Experiment Details

We use 4 H200 GPUs for all our experiments. For prompts and implementation, we use SDPO's codebase [20] on Science and Coding benchmarks whereas OPSD's codebase [58] on Mathematics benchmarks.

For DistIL, we observe performance degradation for longer sequences $y \sim \pi_\theta(\cdot \mid x)$, as the cumulative future loss $\sum_{i>t} \mathrm{H}^{\times}(\cdot)$ grows with sequence length, leading to disproportionately large gradient magnitudes. To mitigate this effect, we apply length normalization, similar to Gu et al. [13]. Specifically, we replace the cumulative loss in future-credit assignment in Eq. (5) with

$$\tilde{L}_{t+1} = \frac{1}{T-t-1} \sum_{i>t} \mathrm{CE}(\text{stopgrad}(\pi_\theta(\cdot \mid x, y_{1:i-1}, f)) \parallel \pi_\theta(\cdot \mid s_i)).$$

D.1 Hyperparameters for Science and Coding Benchmark

For Science and Coding benchmarks, we follow the implementation of SDPO and use their reported hyperparameters for baselines. The hyperparameters for SDPO, DistIL (Ours), and GRPO with its off-policy and on-policy variants are provided in Table 3, 4, and 5 respectively.

D.2 Hyperparameters for Maths task

For Mathematics benchmarks, we follow the implementation of OPSD and use their reported hyperparameters for baselines. The hyperparameters for SDPO, OPSD, DistIL (Ours), and GRPO are provided in Table 6 and that for SFT in Table 7. Evaluation hyperparameters are provided in Table 8 for Qwen3-4B-2507-Instruct model and Table 9 for Qwen3-8B model. The only difference between OPSD and SDPO in this scenario is that OPSD uses Forward-KL divergence whereas SDPO uses reverse-KL divergence in these mathematics experiments.

E Additional Results

E.1 More results for Coding.

SDPO evaluated their coding experiment at temperature = 0.6. We include results for that scenario in here. DistIL achieves Acc/Mean@16 $= 0.660$ and Score/Mean@16 $= 0.482$, outperforming SDPO (0.643, 0.467) and GRPO (0.639, 0.411). The large gap over GRPO reflects its fundamental inability

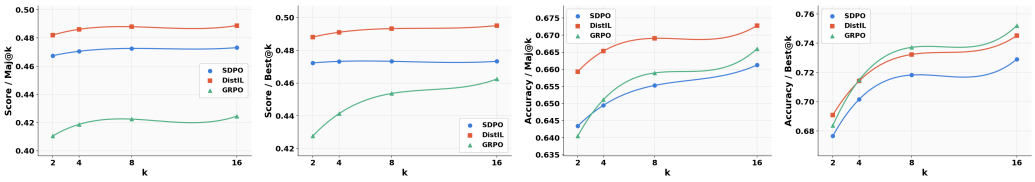

Figure 3: LCBv6 evaluation at τ=0.2 (checkpoint-step 80), reporting Score and Accuracy at Best@k and Maj@k for $k \in \{2, 4, 8, 16\}$.

to exploit execution feedback. Figure 3 notice that the same trend follows as that for temperature=0.2, except that GRPO becomes slightly better on Accuracy/Best@k for k=16.

E.2 Ablation Study

E.2.1 Credit Assignment

We compare DistIL against a CE (cross-entropy) baseline that retains only the cross-entropy term of the gradient, performing local credit assignment analogous to SDPO and OPSD. Figure 4 shows results on the Material domain. DistIL consistently outperforms CE throughout training, demonstrating that the full credit assignment across the entire response distribution rather than assigning it locally, is the key driver of performance gains. The CE baseline exhibits higher variance and fails to improve to the level of DistIL's performance, suggesting that local credit assignment is insufficient for learning robust scientific reasoning.

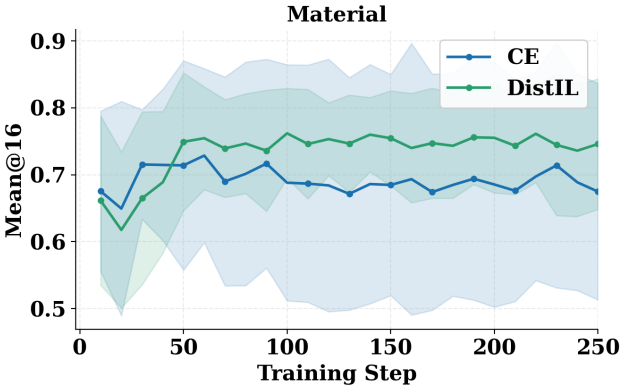

Figure 4: Comparison between DistIL (Ours) and CE. The difference between these methods lies in the credit assignment. CE only does local token-wise creadit assignment similar to SDPO and OPSD, whereas DistIL does full credit assignment as given in Eq. 5.

E.2.2 Top-1 v/s Top k v/s Full distribution

We ablate the number of teacher tokens used in the distillation objective across $K \in \{1, 100, \text{Full}\}$ on the Physics domain (Figure 5). Top-1 distillation, which aligns the student only to the single highest-probability teacher token, underperforms throughout training, reflecting insufficient signal from a single token. Top-100 achieves the best final performance and most stable training trajectory, outperforming both Top-1 and Full distillation. Full distillation, while competitive, introduces noise from low-probability tokens that are unlikely to be informative, slightly degrading performance relative to Top-100. Based on this analysis, we use K=100 in all experiments.

F Limitations

We assume that the teacher when conditioned on feedback has better expected reward than the student which does not get to see this feedback. Although, this does make sense but there might

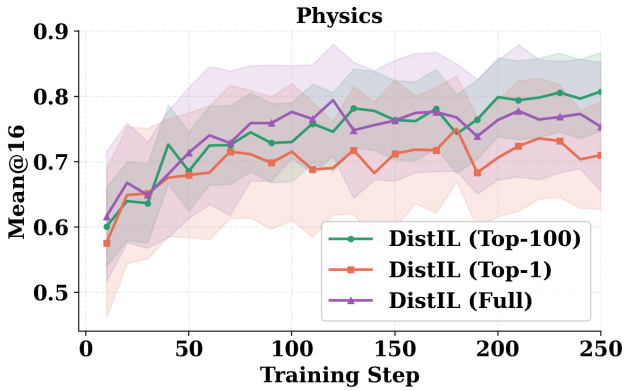

Figure 5: Comparison of Avg@16 between different number of top-k tokens used for training DistIL on Material benchmark.

exist some corner cases where such phenomenon does not happen. Also, in the analysis we assumed $\left|\frac{\pi^\star(a|s)}{\pi^i(a|s)}\right| \leq C, \forall s \in \mathcal{S}, a \in \mathcal{A}$. Again, this makes sense but there might lie some corner cases where this does not happen.

G LLM Usage

We used LLM to assist with grammar and readability of the paper. All research ideas, analysis, and technical content are original from authors.

Parameters	Science	Coding
General		
Model	Qwen/Qwen3-8B allenai/Olmo3-7B-Instruct	Qwen/Qwen3-8B
Thinking	False	False
Data		
Max. prompt length	2048	2048
Max. response length	8192	8192
Batching		
Question batch size	32	32
Mini batch size	32	1
Number of rollouts	8	8
Rollout		
Inference engine	vllm	vllm
Temperature	1.0	1.0
Validation		
Number of rollouts	16	4
Temperature	0.6	0.6
Top-p	0.95	0.95
SDPO Loss		
Top-K distillation	100	20
Distillation divergence	Jensen–Shannon	Reverse-KL
Clip advantages	–	–
Teacher-EMA update rate	0.05	0.01
Rollout importance sampling clip	2	2
Training		
Optimizer	AdamW	AdamW
Learning rate	1×10^{-5} (constant)	1×10^{-6} (constant)
Warmup steps	10	0
Weight decay	0.01	0.01
Gradient clip norm	1.0	1.0

Table 3: Hyperparameters used for **SDPO** for each experimental setup.

Parameters	Science	Coding
General		
Model	Qwen/Qwen3-8B allenai/Olmo3-7B-Instruct	Qwen/Qwen3-8B
Thinking	False	False
Data		
Max. prompt length	2048	2048
Max. response length	8192	8192
Batching		
Question batch size	32	32
Mini batch size	32	1
Number of rollouts	8	8
Rollout		
Inference engine	vllm	vllm
Temperature	1.0	1.0
Validation		
Number of rollouts	16	4
Temperature	0.6	0.6
Top-p	0.95	0.95
SDPO Loss		
Top-K distillation	100	20
Distillation divergence	Jensen–Shannon	Reverse-KL
Clip advantages	–	–
Teacher-EMA update rate	0.01	0.01
Rollout importance sampling clip	2	2
Training		
Optimizer	AdamW	AdamW
Learning rate	5×10^{-5} (constant)	1×10^{-6} (constant)
Warmup steps	10	0
Weight decay	0.01	0.01
Gradient clip norm	1.0	1.0

Table 4: Hyperparameters used for **DistIL** for each experimental setup.

Parameters	Value
General	
Model	Qwen/Qwen3-8B allenai/Olmo3-7B-Instruct
Thinking	False
Data	
Max. prompt length	2048
Max. response length	8192
Batching	
Question batch size	32
Mini batch size	8 (default) / 32 (on-policy)
Number of rollouts	8
Rollout	
Inference engine	vllm
Temperature	1.0
Validation	
Temperature	0.6
Top-p	0.95
Number of rollouts	16
Loss	
ϵ-high	0.28
Rollout importance sampling clip	2
KL coefficient (λ)	0.0
Training	
Optimizer	AdamW
Learning rate	1×10^{-6} (default) / 1×10^{-5} (on-policy)
Warmup steps	10
Weight decay	0.01
Gradient clip norm	1.0

Table 5: Hyperparameters used for **GRPO** (On-Policy) and (Off-Policy) for Science and Coding.

Parameter	GRPO	OPSD/SDPO/DistIL
Learning Rate	5×10^{-6}	5×10^{-6}
Effective Batch Size	32	32
LoRA Rank (r)	64	64
LoRA Alpha (α)	128	128
LoRA Target Modules	`q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj`	
Max Completion Length	16,000	16,384
Number of Generations per Prompt	8	1
Sampling Temperature	0.7	0.7
KL Coefficient (β)	0.0	–
Training Steps	500	100

Table 6: Training configuration for GRPO, OPSD, SDPO, DistIL on Mathematics benchmarks.

Parameter	SFT
Learning Rate	5×10^{-6}
Effective Batch Size	32
LoRA Rank (r)	64
LoRA Alpha (α)	128
LoRA Target Modules	`q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj`
Max Sequence Length	16,000
Number of Training Steps	100

Table 7: Training configuration for SFT on Mathematics benchmarks.

Parameter	Value
Max New Tokens	16,384
Thinking Mode	Enabled
Top-p	0.95
Top-k	20
Min-p	0.0
Presence Penalty	0.0
Samples per Prompt	64
Temperature	0.7

Table 8: Evaluation parameters for Qwen3-4B-Instruct-2507 Model on Maths Task.

Parameter	Value
Max New Tokens	38,912
Thinking Mode	Enabled
Top-p	1.0
Top-k	-1
Min-p	0.0
Presence Penalty	0.0
Samples per Prompt	64
Temperature	1.0

Table 9: Evaluation parameters for Qwen3-8B Model on Maths Task.