Weak-to-Strong Generalization via Direct On-Policy Distillation

Shiyuan Feng[*1,2,4], Huan-ang Gao[*,‡1,2,3], Haohan Chi[*1,2,3], Hanlin Wu[1,2], Zhilong Zhang[1,2], Zheng Jiang[3], Bingxiang He[3], Wei-Ying Ma[1,2], Ya-Qin Zhang[1,2], Hao Zhou[†1,2]

[1]SIA-Lab of Tsinghua AIR and ByteDance Seed
[2]Institute for AI Industry Research (AIR), Tsinghua University
[3]Department of Computer Science and Technology, Tsinghua University
[4]Peking University

Abstract

Reinforcement learning with verifiable rewards (RLVR) is a powerful recipe for improving language-model reasoning, but it is expensive to repeat on every new strong model because the target model must generate many rollouts during training. As models scale, post-training itself becomes a bottleneck. We study a weak-to-strong alternative: run RL on a smaller model where rollouts are cheaper, then reuse what that RL run learned to improve a stronger target model. Directly distilling the post-RL weak teacher is not enough, because the teacher's final policy mixes useful RL gains with the limitations of the smaller model. We propose **Direct On-Policy Distillation** (**Direct-OPD**), which transfers the teacher's RL-induced **policy shift** instead. Direct-OPD compares the post-RL teacher with its own pre-RL reference and treats their log-ratio as a dense implicit reward for the student. In plain terms, the checkpoint pair tells us which actions RL made the weak model more or less likely to take, and Direct-OPD applies that signal on the stronger student's own on-policy states. This directly reuses the weak model's RL supervision signal without running sparse-reward RL on the target model. Empirically, Direct-OPD consistently leverages weaker teachers to improve stronger target models; notably, it boosts Qwen3-1.7B from 48.3% to 58.3% on AIME 2024 in just 4 hours on 8 A100 GPUs. It outperforms step-matched direct RL and enables the sequential composition of multiple policy shifts. Our results show that RL outcomes can be reused across model scales as implicit reward signals, not merely as final models to imitate.

Date: July 9, 2026
Correspondence: zhouhao@air.tsinghua.edu.cn
Project Page: https://bytedtsinghua-sia.github.io/Direct-OPD/

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a dominant post-training recipe for eliciting strong reasoning in large language models [1, 2]. Its cost, however, is tied to the model being trained: every update requires the current policy to generate rollouts, receive verifiable outcome scores, and update on those trajectories. Larger target models make each rollout slower and each RL iteration more expensive. Thus, as reasoning models continue to scale, re-running RLVR from scratch for every new strong model risks making post-training itself a bottleneck.

In this paper, we propose **Direct On-Policy Distillation** (**Direct-OPD**), a weak-to-strong post-training paradigm.

[*]Equal contribution. [‡]Project Lead. [†]Corresponding author.

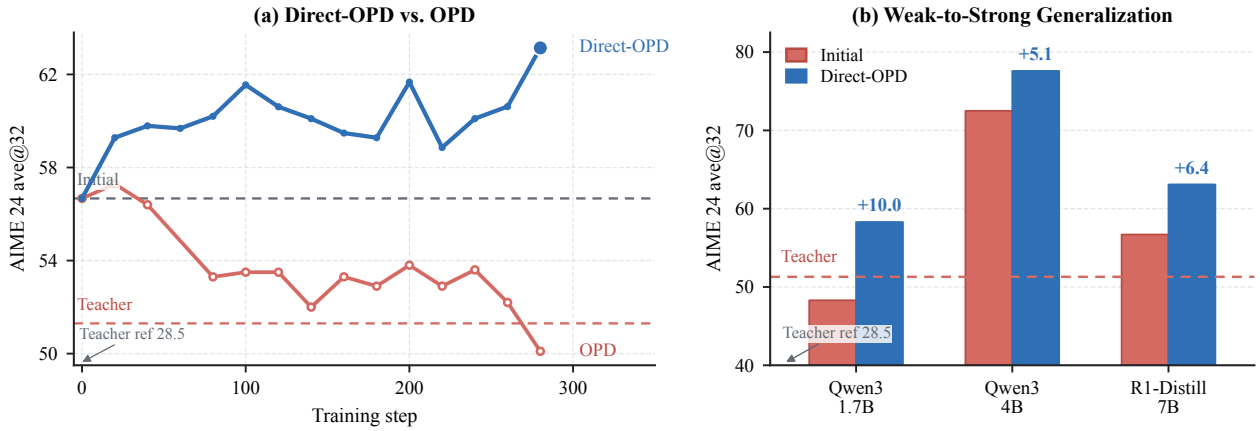


Figure 1 Direct-OPD transfers the effect of small-model RL rather than imitating the small model. (a) Starting from R1-Distill-7B, vanilla OPD toward the post-RL JustRL-1.5B teacher degrades performance, whereas Direct-OPD transfers the JustRL-1.5B − R1-Distill-1.5B policy shift and improves the student. (b) The same policy shift improves Qwen3-1.7B, Qwen3-4B, and R1-Distill-7B on AIME 2024, including students whose initial accuracy already exceeds the post-RL teacher.

The setting is simple: run RL where it is cheap, on a weak model, and use what that RL run learned to train a stronger target model. Crucially, the weak model is not assumed to be a better reasoner than the student; it is merely an inexpensive vehicle which ships the behavior the RL signal has already reshaped. Direct-OPD transfers the improvement induced by weak-model RL, not the weak model itself.

The central question is what exactly to transfer from the weak RL run. One candidate is the post-RL teacher's final policy: on-policy distillation (OPD) lets the student sample its own states and trains it to match the teacher there. Yet this is the wrong object for weak-to-strong transfer, because the final policy entangles the useful change induced by RL with the weak model's intrinsic capacity limits. When the student already surpasses the teacher, imitating that policy can overwrite stronger behavior. Figure 1(a) illustrates this failure: R1-Distill-7B [1] begins at 56.7, already above the post-RL JustRL-1.5B teacher [2] at 51.3, yet vanilla OPD drags it down to roughly 50. The weak RL run carries useful supervision, but its final policy is a poor carrier of that supervision.

Direct-OPD instead isolates the supervision by contrasting the weak model with itself, before and after RL. Let $\pi_{T_{\text{ref}}}$ denote the weak model before RL and π_T its post-RL counterpart. We define the teacher policy shift

$$\Delta_T(y \mid x) = \log \pi_T(y \mid x) - \log \pi_{T_{\text{ref}}}(y \mid x), \tag{1}$$

which is positive on responses that RL made more likely and negative on those it suppressed. The subtraction discards what the weak model already preferred before RL and retains only what RL changed. More importantly, under the KL-regularized RL objective, this policy shift is mathematically equivalent to the reward that trained the weak model: optimizing against Δ_T recovers the same objective form as optimizing against that reward, as proved in Section 2.2. A pair of weak-model checkpoints therefore stores the RL supervision signal directly in policy space. Direct-OPD applies this signal on the student's own on-policy states, anchored to the student initialization by a KL term, without training an explicit reward model or running sparse-reward RL on the target.

This mechanism converts weak-model RL into a transferable post-training signal. Figure 1(b) shows that the policy shift from R1-Distill-1.5B to JustRL-1.5B improves Qwen3-1.7B, Qwen3-4B [3], and R1-Distill-7B alike, including students that already start above the post-RL teacher. The transfer is also inexpensive: with 8 A100 GPUs for about 4 hours, Direct-OPD raises Qwen3-1.7B from 48.3 to 58.3 on AIME 2024, comparable to what Polaris attains by running RL directly on Qwen3-1.7B with 32 A100 GPUs for at least a week [4].

Contributions.

- **A weak-to-strong post-training paradigm.** We recast small-model RL as a cheap generator of an implicit reward for stronger models: rather than copying a weak teacher's final policy, we evaluate its RL-induced policy shift on the student's own on-policy states.
- **Consistent gains across teacher pairs and students.** With two teacher pairs (R1-Distill-1.5B → JustRL-1.5B and Nemotron-1.5B → QuestA-Nemotron-1.5B) and three students from two families (R1-Distill-7B, Qwen3-1.7B, and Qwen3-4B), Direct-OPD improves every student—including R1-Distill-7B and Qwen3-4B, which already start above the post-RL teacher—and lifts Qwen3-1.7B from 48.3 to 58.3 on AIME 2024 in about 4 hours on 8 A100 GPUs.
- **Cheaper than direct large-scale RL.** At matched RL steps, running RL on a 1.5B model and transferring with Direct-OPD outperforms running RL directly on R1-Distill-7B in both accuracy and compute.
- **Analysis of when the signal is reliable.** Unlike imitating the teacher's final policy, Direct-OPD does not require high teacher–student top-k overlap and transfers across thinking patterns; we identify the response-length and KL conditions under which the implicit reward stays aligned with validation accuracy.

2 Direct On-Policy Distillation

2.1 Preliminaries: On-Policy Distillation

We consider autoregressive language-model policies. Let $x \sim \mathcal{D}$ be a prompt and $y = (y_1, \ldots, y_T)$ a response, so that a policy π factorizes as $\log \pi(y \mid x) = \sum_{t=1}^{T} \log \pi(y_t \mid s_t)$ over prefixes $s_t = (x, y_{<t})$. We distinguish four policies: the policy being trained π_θ and its initial checkpoint π_S; and a teacher pair consisting of a pre-RL reference $\pi_{T_{\text{ref}}}$ and the corresponding post-RL teacher π_T. Our weak-to-strong setting is the regime in which π_T may be smaller or weaker than the target student, yet the transition from $\pi_{T_{\text{ref}}}$ to π_T still encodes information worth transferring.

On-policy distillation (OPD) supervises the student on trajectories it samples itself. Given a prompt x, the student draws $\hat{y} \sim \pi_\theta(\cdot \mid x)$, and at each visited prefix $s_t = (x, \hat{y}_{<t})$ we read off the student and teacher next-token distributions $p_t(v) = \pi_\theta(v \mid s_t)$ and $q_t(v) = \pi_T(v \mid s_t)$. Standard OPD minimizes the sequence-level KL to the teacher on these rollouts,

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}}\big[\, D_{\text{KL}}\big(\pi_\theta(\cdot \mid x) \,\big\|\, \pi_T(\cdot \mid x)\big)\,\big], \tag{2}$$

which factorizes exactly into dense, per-token supervision on the states the student actually visits,

$$\mathcal{L}_{\text{OPD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, \hat{y} \sim \pi_\theta}\Big[\, \textstyle\sum_{t=1}^{T} D_{\text{KL}}(p_t \,\|\, q_t)\,\Big]. \tag{3}$$

In practice the teacher is queried only on the student's top-k support $S_t = \text{TopK}_v\, p_t$, giving the local estimator

$$\mathcal{L}_{\text{OPD}}^{\text{top-}k}(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, \hat{y} \sim \pi_\theta}\Big[\, \textstyle\sum_{t=1}^{T} D_{\text{KL}}\big(\bar{p}_t^{S_t} \,\big\|\, \bar{q}_t^{S_t}\big)\,\Big], \tag{4}$$

where $\bar{p}_t^{S_t}$ and $\bar{q}_t^{S_t}$ renormalize p_t and q_t on S_t [5]. This top-k KL estimator produces a dense per-token training signal involving the teacher–student log-ratio $\log q_t(v) - \log p_t(v)$. Direct-OPD inherits this on-policy, top-k interface unchanged, and departs from OPD only in *what signal is read from the teacher* (Section 2.2).

2.2 Policy Shifts as Implicit Rewards

Which signal to transfer. Rather than imitating the teacher's absolute output distribution, Direct-OPD transfers its RL-induced **policy shift**: the sequence-level log-ratio between the post-RL teacher π_T and its pre-RL reference $\pi_{T_{\text{ref}}}$,

$$\Delta_T(y \mid x) = \log \pi_T(y \mid x) - \log \pi_{T_{\text{ref}}}(y \mid x), \tag{5}$$

which isolates the teacher's RL-induced *direction* rather than its absolute final distribution. This shift is not an arbitrary heuristic: under the **policy-as-reward** view of KL-regularized RL, it can be interpreted as the teacher's implicit reward.

For a reward r, reference policy π_{ref}, and penalty $\beta > 0$, the KL-regularized objective $\max_\pi \mathbb{E}_{y\sim\pi}\big[r(x,y) - \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}\big]$ admits the closed-form optimum $\pi^* \propto \pi_{\text{ref}} \exp(r/\beta)$, hence

$$\log \frac{\pi^*(y \mid x)}{\pi_{\text{ref}}(y \mid x)} = \tfrac{1}{\beta}\, r(x,y) - \log Z(x). \tag{6}$$

Since $\log Z(x)$ is constant across responses to the same prompt, the policy–reference log-ratio recovers the reward up to a positive scale and a per-prompt constant. This is the same identity that underlies Direct Preference Optimization, which fits a policy from preferences without a separately trained reward model [6]. Here we use the identity in the opposite direction: we are already *given* a post-RL teacher π_T and its pre-RL reference $\pi_{T_{\text{ref}}}$, and we read the reward-like signal back out of their policy ratio. Treating π_T as the KL-regularized optimum of some latent reward r_T anchored at $\pi_{T_{\text{ref}}}$,

$$\Delta_T(y \mid x) = \tfrac{1}{\beta}\, r_T(x,y) - \log Z_T(x), \tag{7}$$

so the shift Δ_T can be interpreted as the teacher's **implicit reward**, up to a positive scale and a per-prompt constant. Direct-OPD transfers this reward-like signal to the student.

2.3 The Direct-OPD Objective

Idealized sequence-level objective. We first write Direct-OPD as a sequence-level objective. With the student initialized from π_S, Direct-OPD optimizes the student against the teacher-shift signal Δ_T while regularizing the student toward its own initialization,

$$J_{\text{Direct-OPD}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\big[\Delta_T(y \mid x)\big] - \alpha\, D_{\text{KL}}\big(\pi_\theta(\cdot \mid x) \,\|\, \pi_S(\cdot \mid x)\big)\right], \tag{8}$$

where $\alpha > 0$ controls the KL penalty. In this idealized form, the objective is itself a KL-regularized RL objective on the student: its optimum is

$$\pi^*(y \mid x) \propto \pi_S(y \mid x)\, \exp\big(\tfrac{1}{\alpha}\Delta_T(y \mid x)\big) = \pi_S(y \mid x)\left(\frac{\pi_T(y \mid x)}{\pi_{T_{\text{ref}}}(y \mid x)}\right)^{1/\alpha}, \tag{9}$$

and substituting the implicit-reward form of Eq. 7 gives $\pi^* \propto \pi_S \exp(r_T/\alpha\beta)$. The student is therefore optimized as if it were running KL-regularized RL with the small teacher's implicit reward, while using its own initialization π_S as the reference. Crucially, it obtains this reward-like signal entirely from the checkpoint pair, without querying a verifiable reward or running sparse-reward RL on the target.

From sequence to token level. Because both teachers factorize autoregressively over the same prefixes, the sequence shift decomposes exactly into a sum of token-level shifts,

$$\Delta_T(y \mid x) = \sum_t r_t(y_t \mid s_t), \qquad r_t(v) = \log \pi_T(v \mid s_t) - \log \pi_{T_{\text{ref}}}(v \mid s_t), \tag{10}$$

so that $r_t(v)$ can be viewed as a dense immediate per-token reward: $r_t(v) > 0$ where the teacher's RL encouraged token v, and $r_t(v) < 0$ where it suppressed it. In practice, we use the corresponding zero-discount token-level surrogate for Eq. 8, crediting each candidate token by its immediate shift $r_t(v)$ rather than by a future return over later shifts. This reuses the on-policy, top-k interface of Section 2.1.

Top-k action-space restriction. The implementation uses a local top-k approximation to this zero-discount token-level surrogate. To keep the signal on actions the student actually considers, at each visited prefix we restrict to its top-k support $S_t = \text{TopK}_v\, \pi_\theta(v \mid s_t)$ and renormalize the student probabilities on it,

$$\bar{p}_t(v) = \frac{\pi_\theta(v \mid s_t)}{\sum_{u\in S_t} \pi_\theta(u \mid s_t)}, \qquad v \in S_t, \tag{11}$$

which is the renormalized student distribution $\bar{p}_t^{S_t}$ of Section 2.1.

Analytical top-k policy gradient. The immediate per-token reward $r_t(v)$ depends only on the teacher pair, so we use it directly as the local advantage. A single-sample, token-level policy gradient for the zero-discount surrogate would weight the log-likelihood of the sampled token by its immediate reward,

$$\nabla_\theta J_{\mathrm{MC}} = \mathbb{E}_{x,y}\Big[\textstyle\sum_t r_t(y_t)\,\nabla_\theta \log \pi_\theta(y_t \mid s_t)\Big], \tag{12}$$

but estimating each step from one token is high-variance. Since we already have the full top-k rewards and probabilities at every visited prefix, we replace the single-token estimate at each step by its expectation over the restricted distribution $\bar{p}_t$, i.e. we Rao–Blackwellize the per-step action while still sampling the trajectory $y \sim \pi_\theta$:

$$\nabla_\theta J_{\mathrm{analytical}} = \mathbb{E}_{x,\, y\sim\pi_\theta}\Big[\sum_t \sum_{v\in S_t} \underbrace{\bar{p}_t(v)}_{\text{weight}}\ \underbrace{r_t(v)}_{\text{reward}}\ \nabla_\theta \underbrace{\log \pi_\theta(v \mid s_t)}_{\text{log-likelihood}}\Big]. \tag{13}$$

This gives a Rao–Blackwellized per-step estimator under the restricted top-k action distribution; it removes the token-sampling variance of $\nabla_\theta J_{\mathrm{MC}}$ while leaving the sampled trajectory distribution untouched.

Stop-gradient coefficient. For this surrogate to match the policy-gradient form, the weight $\bar{p}_t(v)$ must enter only as a scalar. It depends on θ through the student softmax, so differentiating it would inject extra Jacobian terms from the top-k distribution. We therefore detach the weighted reward into a static coefficient,

$$A_t^{\mathrm{w}}(v) = \mathrm{stop\_gradient}\big(\bar{p}_t(v)\cdot r_t(v)\big), \tag{14}$$

giving the final local top-k surrogate for the zero-discount objective,

$$\nabla_\theta J_{\text{Direct-OPD}} \approx \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta}\Big[\sum_t \sum_{v\in S_t} A_t^{\mathrm{w}}(v)\,\nabla_\theta \log \pi_\theta(v \mid s_t)\Big] - \alpha\,\nabla_\theta D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid x)\,\|\,\pi_S(\cdot \mid x)\big). \tag{15}$$

In practice we do not differentiate the KL analytically; we use the standard KL-penalty implementation in `verl` to anchor π_θ to π_S.

2.4 Adaptive KL Control

The reward Direct-OPD transfers carries a scale it does not get to choose. By Eq. 7, $\Delta_T = \frac{1}{\beta} r_T - \log Z_T$: its magnitude is the teacher's reward divided by the teacher's KL budget β, both fixed when the teacher was trained and encoded in how far its RL pushed π_T from $\pi_{T_{\mathrm{ref}}}$, yet neither is recoverable from the checkpoint pair. The per-token shift $r_t(v) = \log \pi_T(v \mid s_t) - \log \pi_{T_{\mathrm{ref}}}(v \mid s_t)$ is likewise a function of the teacher pair alone; the student enters only through where the signal is read off—the prefixes its rollouts visit and the top-k actions retained at each one.

This scale mismatch makes a fixed α unlikely to transfer robustly across settings. From the optimum in Eq. 9, $\pi^* \propto \pi_S \exp(\frac{1}{\alpha}\Delta_T)$, α is the temperature of an exponential tilt of π_S, and is meaningful only relative to the scale of Δ_T. But Δ_T lives in the teacher's reward units ($\frac{1}{\beta} r_T$) while α is a coefficient on the student's KL—two scales with no a priori conversion, the teacher's being unobservable. A single α therefore cannot be calibrated in advance across teacher–student pairs, and we instead adopt a lightweight controller that adapts α to the running scale of the dense signal.

Let α_m be the KL coefficient at training iteration m, and let $\bar{r}_m$ be the batch-level mean of the student-weighted shift $\bar{p}_t(v)\, r_t(v)$ over the visited prefixes and their top-k candidates—the same dense reward the gradient optimizes. Before the actor update we set

$$\alpha_{m+1} = \mathrm{clip}\left(\alpha_m\,(1 + \epsilon\ \mathrm{sgn}(\bar{r}_m))\,, \alpha_{\min}, \alpha_{\max}\right), \tag{16}$$

with $\mathrm{sgn}(0) = 0$; by default $\epsilon = 0.01$ and $[\alpha_{\min}, \alpha_{\max}] = [0.5, 2.5]$. Here $\mathrm{sgn}(\bar{r}_m)$ reports whether, on the prefixes the student currently visits, the teacher's RL on average raised or lowered the retained candidate tokens—a batch-level sign that the student-weighted shift carries directly.

The controller is intended to keep the dense signal balanced. When the visited prefixes carry positive teacher shift on average, it raises α to discourage over-amplifying that local signal; when the average shift is negative, it

lowers α, weakening the anchor so the dense gradient can move probability away from tokens whose likelihood the teacher's RL reduced. This differs from the standard adaptive controller for an in-reward KL penalty, which steers toward a target KL value: here the update is driven by the sign of the Direct-OPD dense reward and acts on the explicit actor KL loss coefficient. We analyze KL-coefficient sensitivity in Section 4.3.

3 Experiments

In this section we answer the following research questions:

- **RQ1.** Can a small teacher's RL-induced policy shift improve students that already match or exceed the teacher's own ability, and does this hold across different teacher pairs and student families? (Section 3.1)
- **RQ2.** Under a fixed RL-step budget, does running RL on a small model and transferring its policy shift with Direct-OPD outperform running RL directly on the large target—in both accuracy and compute? (Section 3.2)
- **RQ3.** Can several independently learned policy shifts be composed sequentially to accumulate their gains in a single student? (Section 3.3)

3.1 A small RL teacher improves much stronger students

Our first experiment asks whether an RL improvement learned by a small teacher can be used to improve students that are stronger than the post-RL teacher. We use R1-Distill-1.5B as the teacher reference and JustRL-1.5B as the post-RL teacher, then distill the policy shift from this pair into three target students: R1-Distill-7B, Qwen3-1.7B, and Qwen3-4B. In this setting, R1-Distill-7B and Qwen3-4B already outperform the post-RL JustRL teacher before Direct-OPD, making it a direct weak-to-strong transfer test.

We further test a second teacher pair, Nemotron-1.5B $\rightarrow$ QuestA-Nemotron-1.5B, which comes from a different training pipeline and data source. This setting is not intended as a strict weaker-teacher-to-stronger-student comparison, since the post-RL QuestA teacher is already strong on AIME. Instead, it tests whether Direct-OPD can transfer a policy-shift signal beyond a single teacher family, training recipe, and data source. We transfer this policy shift into R1-Distill-7B and Qwen3-1.7B.

Figure 2 shows that the transferred policy shift improves students across these settings, including R1-Distill-7B and Qwen3-4B students whose starting accuracies already exceed the post-RL JustRL teacher. The QuestA results serve as an additional robustness check: they show that the effect is not specific to the JustRL teacher pair, and can also appear with a different teacher training pipeline, data source, and post-RL checkpoint. Taken together, these results suggest that Direct-OPD transfers a reusable policy-shift signal rather than simply imitating the post-RL teacher's final policy.

This result supports the interpretation that Direct-OPD uses an RL-induced direction of improvement rather than the teacher's full policy.

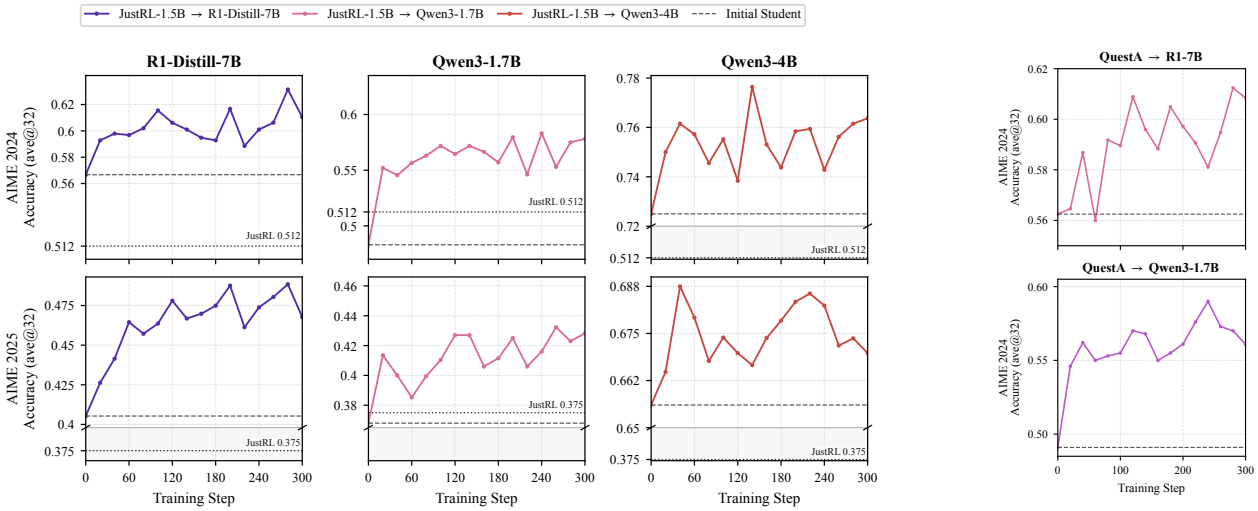


Figure 2 Direct-OPD transfers RL-induced policy shifts across teacher pairs and student families. **Left:** R1-Distill-1.5B → JustRL-1.5B transfer into R1-Distill-7B, Qwen3-1.7B, and Qwen3-4B, evaluated on AIME 2024 and AIME 2025. **Right:** Nemotron-1.5B → QuestA-Nemotron-1.5B transfer into R1-Distill-7B and Qwen3-1.7B on AIME 2024. The two teacher pairs come from different training data and pipelines, showing that Direct-OPD is not specific to a single post-RL teacher.

(a) JustRL policy-shift transfer

Model	AIME24	AIME25
Teacher ref	28.5	24.0
Teacher RL [2]	51.3	37.5
Qwen3-1.7B	48.3	36.8
+ Direct-OPD	**58.3** +10.0	**43.2** +6.4
Qwen3-4B	72.5	65.6
+ Direct-OPD	**77.6** +5.1	**68.8** +3.2
R1-Distill-7B	56.7	40.5
+ Direct-OPD	**63.1** +6.4	**48.8** +8.3

(b) QuestA policy-shift transfer

Model	AIME24	AIME25
Teacher ref	61.77	49.50
Teacher RL [7]	72.50	62.29
Qwen3-1.7B	48.3	36.8
+ Direct-OPD	**59.0** +10.7	**43.1** +6.3
R1-Distill-7B	56.3	39.5
+ Direct-OPD	**61.2** +4.9	**44.0** +4.5

Table 1 Score summaries for the different students and teacher pairs

3.2 Weak-to-strong generalization beats RL

We next ask whether Direct-OPD is only a way to reuse an already trained small teacher, or whether it can be a better training path than running RL directly on the larger target model. We compare two matched-step routes. The direct-RL route trains R1-Distill-7B with RL. The weak-to-strong route first trains the smaller R1-Distill-1.5B model with RL, then transfers the resulting RL/reference policy shift into R1-Distill-7B using Direct-OPD.

We test that for the same number of RL steps, a smaller model is a more efficient place to find a useful policy-improvement direction, and Direct-OPD can then transfer that direction to the larger student. We trained R1-Distill-1.5B on DAPO dataset, and selected R1-Distill-1.5B RL checkpoints at steps 300, 600, 900, 1200, 1500. For each selected checkpoint, we construct a teacher pair using the base R1-Distill-1.5B model as $\pi_{T_{\text{ref}}}$ and the selected RL checkpoint as π_T. We then train R1-Distill-7B with Direct-OPD and compare the resulting validation score against direct R1-Distill-7B RL at the matched small-teacher RL step. In the figure, T300 denotes the route that first trains R1-Distill-1.5B with RL for 300 steps and then applies Direct-OPD to R1-Distill-7B; T600–T1500 are defined analogously.

We find that, for the same number of RL steps, running RL on the small model and then transferring the learned policy shift to the large model gives better R1-Distill-7B performance than running RL directly on R1-Distill-7B.

This matched-step advantage also translates into a compute advantage. In our setup, a 1500-step RL run on R1-Distill-1.5B takes about 160 hours on 32 A100 GPUs, while RL on R1-Distill-7B takes about 320 hours. After small-model RL, the Direct-OPD transfer stage adds only about 4 hours on 8 A100 GPUs, which is negligible compared with the RL cost. This low transfer cost comes from two factors: Direct-OPD requires only a short training run, as discussed in Section 4.2, and the teacher models used for scoring are small enough that scoring is not the speed bottleneck.

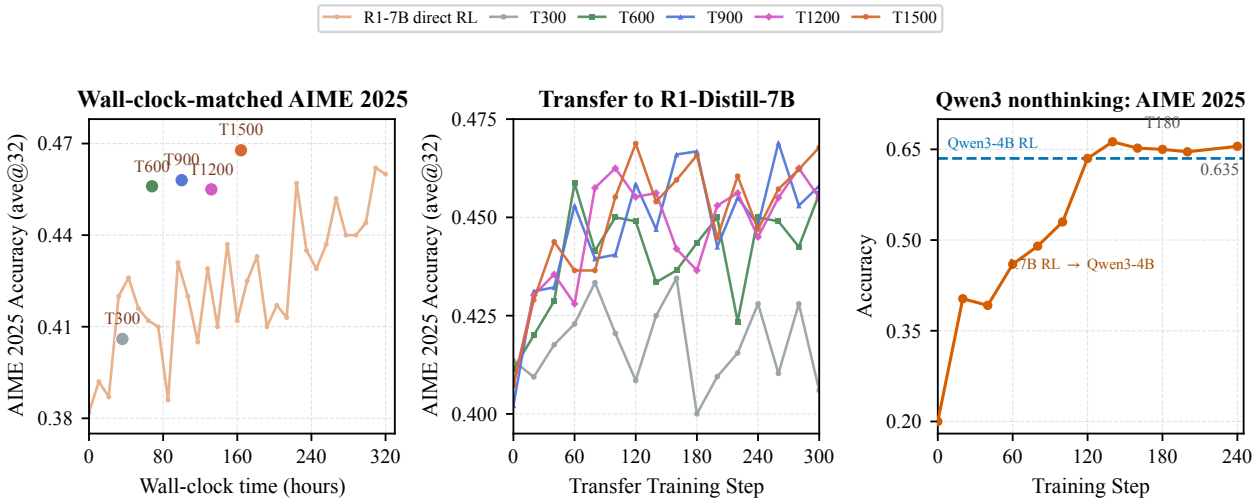


Figure 3 Running RL on a small model and transferring its policy shift with Direct-OPD beats running RL directly on the large target under a shorter wall-clock training path. We compare two routes to improving R1-Distill-7B: direct RL on R1-Distill-7B, versus a weak-to-strong route that runs RL on the smaller R1-Distill-1.5B and transfers the resulting policy shift into R1-Distill-7B with Direct-OPD. TN denotes the transfer that uses the R1-Distill-1.5B RL checkpoint at step N as the post-RL teacher π_T, with the base model as $\pi_{T_{\text{ref}}}$. **Left:** AIME 2025 accuracy against total wall-clock time; the wiggly curve is direct R1-Distill-7B RL, and each TN point sums the elapsed time for small-model RL and the short Direct-OPD transfer. Later transfers (T600–T1500) sit above the direct-RL curve at comparable elapsed time. **Middle:** Direct-OPD transfer trajectories into R1-Distill-7B from the five small-teacher checkpoints; the early T300 carries a weaker shift than T900–T1500. **Right:** the same recipe with Qwen3 nonthinking models—transferring a Qwen3-1.7B RL shift into Qwen3-4B reaches the 0.635 accuracy of direct Qwen3-4B RL (dashed) on AIME 2025.

Figure 3 shows that the transferred signal depends on which small-teacher RL checkpoint is used. This dependence is expected: different points in the small-model RL trajectory encode different policy shifts, and not every shift is equally useful on the larger student's state distribution.

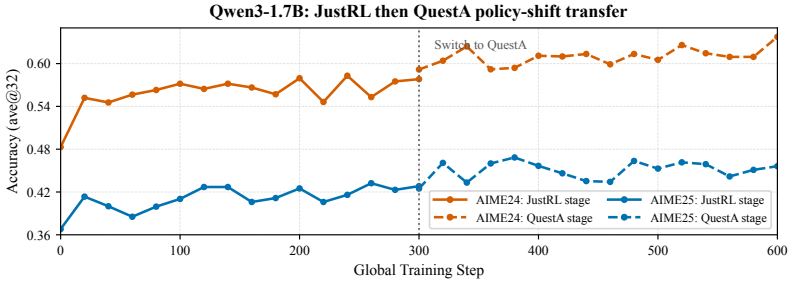
This result changes the role of the small model. The small teacher is not used because its endpoint policy is stronger than R1-Distill-7B. Instead, the small model provides a cheaper environment in which RL discovers a direction of improvement. Direct-OPD then transfers that direction to the larger model by evaluating the teacher/reference log-ratio on the larger student's own on-policy states. Direct RL on the large model must discover the same kind of direction through its own rollouts; weak-to-strong transfer reuses the direction already found by the small model.

We observe the same pattern with Qwen3 nonthinking models. After a 100-step RL run on the 1.7B model, transferring the resulting policy shift to Qwen3-4B-nonthinking reaches the 0.635 direct-RL level on AIME 2025. This suggests that a small-model RL run can recover the useful RL direction even when the target is a stronger 4B model.

Takeaway. Compared with running RL directly on the large model, first running RL on the small model and then transferring to the large model has two advantages: small-model RL is more stable, easier, and cheaper, while the following Direct-OPD stage provides fast transfer to the large model while preserving the performance advantage.

3.3 Sequential composition

This experiment asks whether two independently learned policy shifts can be applied in sequence to the same student. The student is first trained with the R1-Distill-1.5B → JustRL-1.5B signal and then continued with the Nemotron-1.5B → QuestA-Nemotron-1.5B signal. Figure 4 shows the AIME 2024 and AIME 2025 trajectories after aligning the second stage's local 0–300 training steps to global steps 300–600. The first-stage endpoint and the second-stage starting point are evaluated from separate samples, so the small discontinuity at the stage boundary reflects evaluation variance.



Score summary

Stage	AIME24	AIME25
Initial	48.3	36.8
After JustRL	**58.3** +10.0	**43.2** +6.4
After QuestA	**63.8** +15.5	**46.8** +10.0

Figure 4 Sequential policy-shift transfer into Qwen3-1.7B on AIME 2024 and AIME 2025. Left: trajectories after aligning the second stage to global steps 300–600. Right: endpoint scores on AIME 2024/2025. The first stage uses the R1-Distill-1.5B → JustRL-1.5B signal; the second stage continues from that checkpoint with the Nemotron-1.5B → QuestA-Nemotron-1.5B signal.

Takeaway. Different RL training runs can learn different abilities, and Direct-OPD can sequentially compose these abilities into the same student.

4 Analysis and Training Dynamics

The previous section showed that Direct-OPD transfers RL gains weak-to-strong; here we analyze when and why it works, examining three aspects of the transfer:

- **What is transferred?** Whether Direct-OPD needs the student to imitate the teacher's high-probability tokens, or whether it transfers even when teacher–student top-k overlap stays low (Section 4.1).
- **Does short-horizon training generalize?** Whether training on short rollouts changes only the supervised prefixes or shifts behavior across much longer responses (Section 4.2).
- **Why adapt the KL coefficient?** How the KL coefficient affects the reliability of the teacher-shift reward across teacher–student pairs (Section 4.3).

4.1 Cross-pattern transfer without token-overlap imitation

Prior work on standard OPD argues that teacher-student thinking-pattern compatibility is a key success condition: successful OPD is reflected in progressive alignment on high-probability tokens at student-visited states, with a small shared top-k token set carrying most of the probability mass [5]. This requirement is natural for raw teacher imitation, because the student is asked to match the teacher's final distribution. In that setting, increasing top-k overlap is both a diagnostic and a plausible carrier of the training signal.

We compute top-k overlap using the same set-overlap metric as prior OPD work. For a student policy π_S and a comparison policy π_C at state s_t, let

$$T_k^S(s_t) = \mathrm{TopK}_v \ \ \pi_S(v \mid s_t), \tag{17}$$

$$T_k^C(s_t) = \mathrm{TopK}_v \ \ \pi_C(v \mid s_t). \tag{18}$$

The per-state overlap ratio is

$$\mathrm{Overlap}_k(S, C; s_t) = \frac{\left|T_k^S(s_t) \cap T_k^C(s_t)\right|}{k}. \tag{19}$$

For Direct-OPD, we report this metric against both the post-RL teacher ($C = T$) and the teacher reference ($C = T_{\mathrm{ref}}$) on the same student-visited states.

Direct-OPD changes the object being transferred. Instead of matching the post-RL teacher distribution, it evaluates the teacher/reference log-ratio on student-visited candidate tokens. This signal can rank actions within the student's own support, so useful transfer need not require the student to enter the teacher's high-overlap token regime.

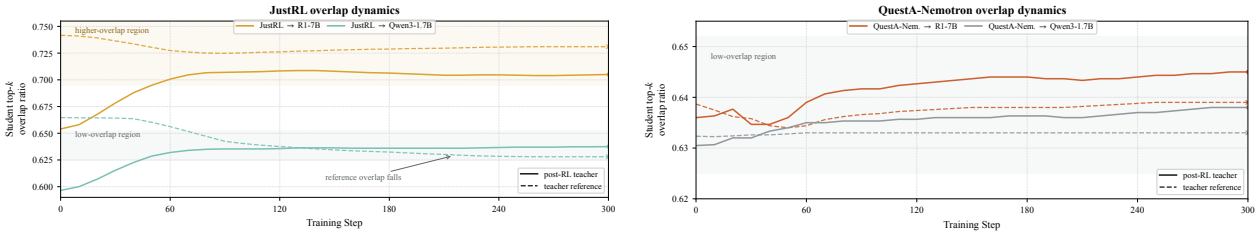


Figure 5 Teacher–student top-k overlap during Direct-OPD training. Left: R1-Distill-1.5B → JustRL-1.5B teacher pair. Right: Nemotron-1.5B → QuestA-Nemotron-1.5B teacher pair. Solid curves measure overlap with the post-RL teacher, and dashed curves measure overlap with the teacher reference. The pattern-aligned R1-Distill transfer enters a higher-overlap regime, while the cross-pattern transfers remain lower and do not become imitation of the post-RL teacher.

Figure 5 separates the pattern-aligned case from the cross-pattern cases. The aligned transfer follows the standard OPD intuition: the student moves into a higher-overlap regime with the post-RL teacher. The cross-pattern transfers behave differently. Their overlap with the post-RL teacher remains lower, and their overlap with the teacher reference does not show a compensating rise. Thus the gains in Figure 2 are not explained by progressive imitation of either teacher checkpoint. They are better explained by using the RL-induced direction from reference to post-RL teacher on the student's own visited states.

We next check whether this low-overlap transfer is a degenerate sharpening effect. If Direct-OPD simply collapsed the actor or forced a trivial low-entropy distribution, the overlap diagnostic would be less informative.

Figure 6 shows that the actor entropy remains controlled rather than collapsing. At the same time, the teacher/reference entropy gap narrows, indicating that training changes how the student samples regions where the teacher's RL-induced shift is evaluated. This supports the same mechanism as the overlap result: Direct-OPD transfers a local policy-shift signal within the student's distribution, not a full teacher policy.

> **Takeaway.** Direct-OPD does not require progressive token-overlap imitation. It can transfer the teacher's RL-induced direction on student-visited tokens while keeping the actor distribution controlled.

4.2 Short-horizon training changes longer-rollout behavior

Recent work on on-policy prefix distillation shows that full-length supervision is not always necessary for reasoning transfer: student rollouts can be truncated to short prefixes, while the resulting model is still

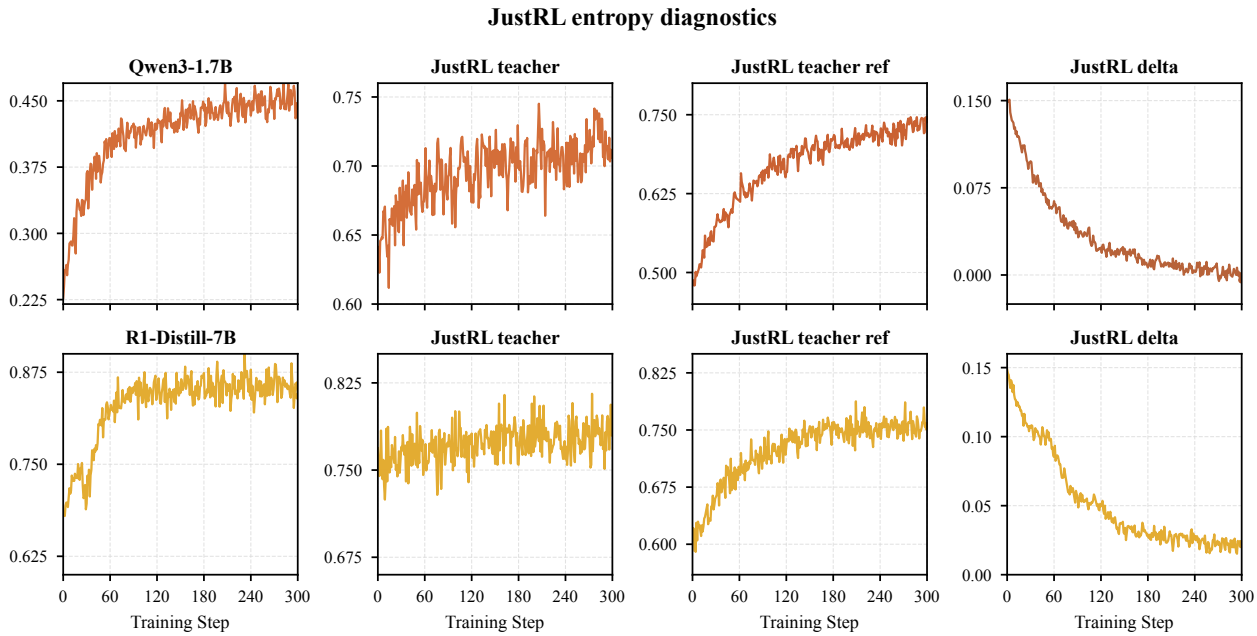


Figure 6 Entropy diagnostics for R1-Distill-1.5B → JustRL-1.5B policy-shift transfer. **Top row:** transfer into Qwen3-1.7B. **Bottom row:** transfer into R1-Distill-7B. Each row shows student entropy, post-RL teacher entropy, teacher-reference entropy, and teacher entropy minus reference entropy. Actor entropy does not collapse, while the teacher/reference entropy gap narrows over training.

evaluated with long generations at test time [8]. Following this view, we do not treat the training horizon merely as a finite rollout length. We deliberately use a short 2k-token response horizon for Direct-OPD training, and ask whether the resulting policy change generalizes beyond the directly supervised prefix.

A natural concern is that such short-horizon training may only change the early prefixes that receive direct supervision, leaving later reasoning behavior unchanged. We test this by evaluating trained actors on fixed long rollouts and measuring whether the actor's likely next tokens move toward the post-RL teacher relative to the teacher reference.

For each response prefix $x_{\le t}$, we take the actor's top-16 token set K_t and compute the actor-probability-weighted teacher gap

$$g_t = \sum_{a \in K_t} \pi_{\text{actor}}(a \mid x_{\le t}) \left[\log \pi_{\text{JustRL}}(a \mid x_{\le t}) - \log \pi_{\text{R1}}(a \mid x_{\le t})\right]. \tag{20}$$

The plotted quantity is the prefix cumulative gap

$$G_T = \sum_{t=1}^{T} g_t. \tag{21}$$

Larger G_T means that, along the actor's own likely tokens, the JustRL teacher assigns higher weighted log-probability than the R1-Distill-1.5B teacher; smaller G_T means that the actor remains closer to R1-Distill-1.5B.

Figure 8 shows that the effect of short Direct-OPD training is not confined to the supervised prefix range: after 40 training steps with a 2k response length, the actor moves toward the teacher-shift direction across much longer fixed rollouts, so the short horizon already captures the transferable signal. Training with a 6k response length shifts the actor even further in this diagnostic (middle panel), yet it validates worst (right panel, 45.6 versus 48.8 for 2k). One plausible explanation is that the extra movement comes from late prefixes,

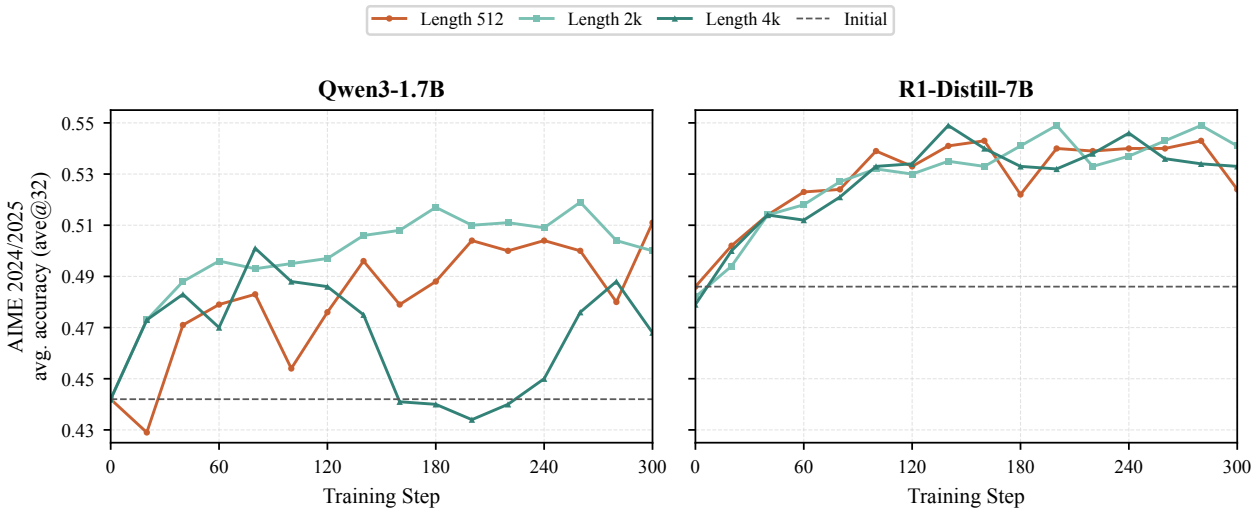


Figure 7 Response-length sweep for R1-Distill-1.5B → JustRL-1.5B transfer with fixed KL coefficient 1. We report the average of AIME 2024 and AIME 2025 validation accuracy (ave@32) during training for Qwen3-1.7B and R1-Distill-7B students. The 2k setting gives stable validation behavior across the two students, while shorter or longer rollouts do not consistently improve the validation curves.

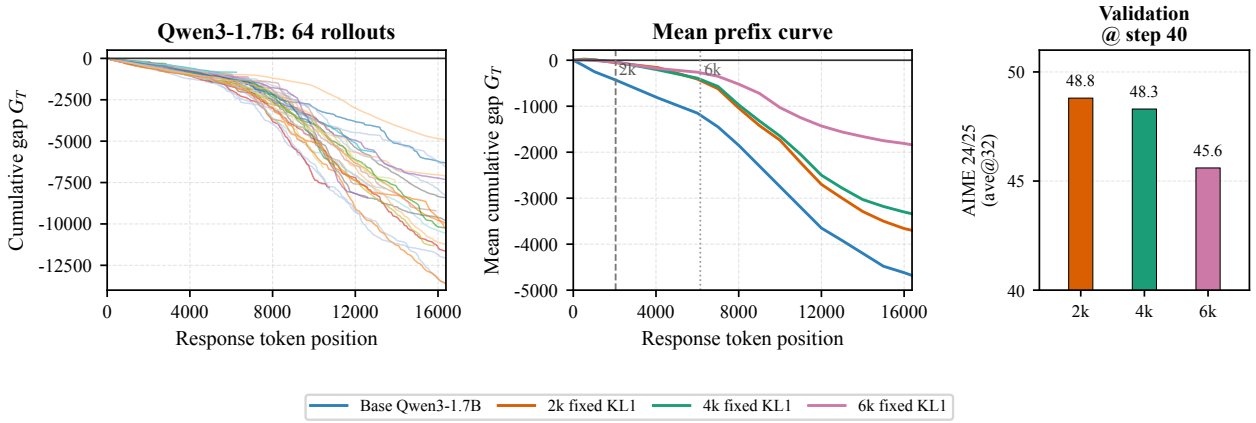


Figure 8 Short-horizon Direct-OPD training changes behavior beyond the supervised prefix. On a fixed set of 64 long rollouts we track the cumulative gap $G_T = \sum_{t \le T} g_t$, where g_t weights, over the actor's top-16 tokens at position t, the log-probability the post-RL teacher (JustRL) assigns minus that of the reference (R1-Distill-1.5B); higher G_T means the actor's likely tokens look more like the post-RL teacher, while lower means the actor remains closer to the reference. **Left:** the 64 per-rollout trajectories for the untrained Qwen3-1.7B actor all drift negative—its long rollouts are reference-like, and this is not driven by a single outlier. **Middle:** the mean G_T for the base actor and for actors trained with 2k/4k/6k response length (40 steps, fixed KL=1); every trained actor sits above the base across the full ∼16k positions, and the 2k actor shifts well past its 2k training horizon (dashed line). **Right:** AIME 2024/2025 (ave@32) at the same 40-step checkpoint—the 2k setting validates best.

where the small teacher pair is increasingly off-distribution and the teacher/reference log-ratio is large but unreliable (Section 4.3). A 6k horizon may over-drive the actor on this unreliable late-prefix signal without improving the answer, whereas a 2k horizon avoids ingesting it while still capturing—and generalizing—the reliable early-prefix direction.

> **Takeaway.** Short-horizon Direct-OPD training can generalize changes in thinking patterns to longer rollouts. A moderate response length avoids the noise and instability of very long responses while capturing more useful signal than overly short training horizons.

4.3 KL controls teacher-shift reward reliability

The teacher/reference log-ratio acts as a dense reward, but this reward is only reliable on states where the teacher shift is meaningful for the student. This makes KL more than a regularization coefficient: it controls which rollout states the student visits, and therefore whether the dense reward remains reliable. If KL is too weak, the student can drift into regions where both teacher and reference assign low probability and the log-ratio becomes a noisy training target. If KL is too strong, the student cannot follow the teacher's policy shift. We test this by sweeping fixed KL coefficients under the same 2k-response setting and comparing them with an adaptive-KL run.

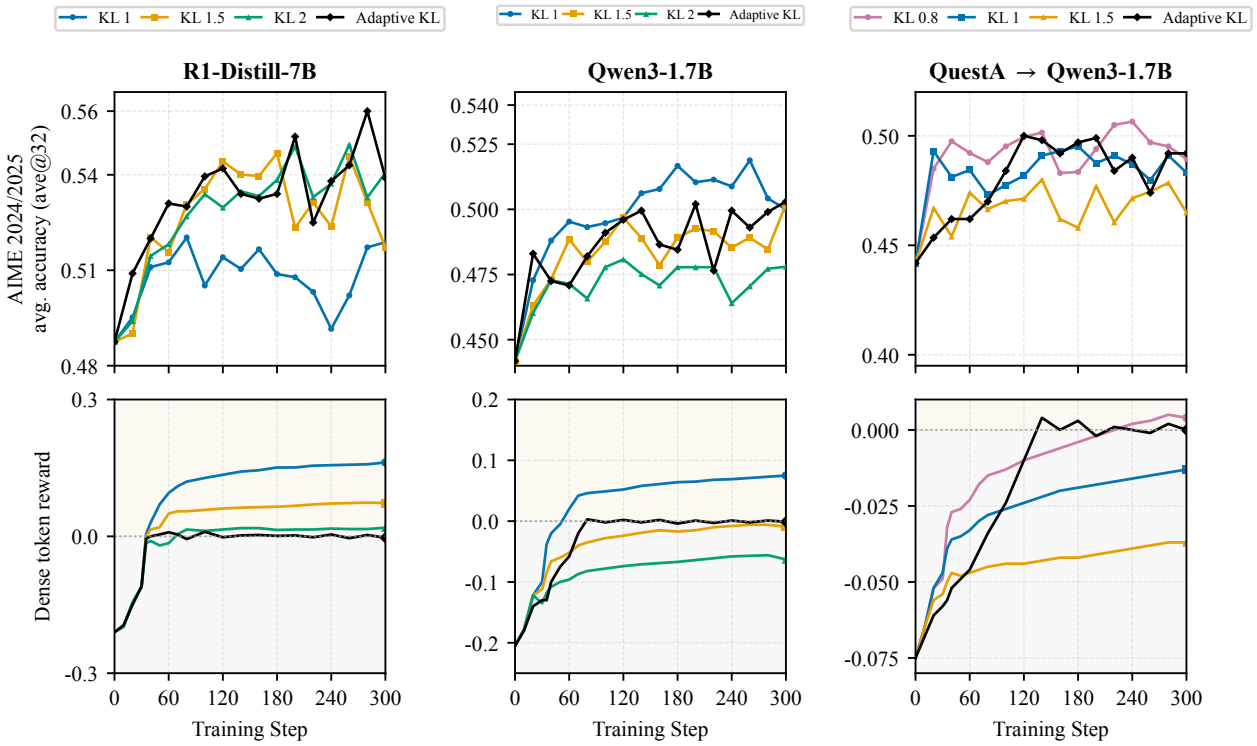


Figure 9 The best KL coefficient is pair-dependent, and adaptive KL pulls the dense token reward toward a balanced regime. 2k-response runs across fixed KL coefficients, with adaptive KL as the black curve. **Top row:** AIME 2024/2025 validation accuracy (ave@32); **bottom row:** dense token reward. The best fixed coefficient differs across teacher-student pairs, and a larger reward does not imply better validation. Adaptive KL instead pulls the dense token reward toward zero after an initial correction, keeping the student from simply maximizing the dense teacher/reference reward.

Figure 9 shows that the best fixed KL value is not universal: different teacher-student pairs prefer different constraint strengths. The dense token reward also does not provide a pair-independent rule; a larger positive value can coincide with worse validation in one setting but track the best validation curve in another. Thus the

dense reward should not be treated as a scalar objective to maximize independently of the rollout distribution. The useful regime depends on the teacher-student pair and on where the student currently samples.

The adaptive-KL runs provide a complementary diagnostic. After an initial correction phase, their dense token rewards move toward zero rather than remaining strongly positive or negative. This behavior is consistent with the intended role of adaptive KL: keep the student in a region where the teacher/reference comparison remains informative, instead of letting the student either ignore the teacher shift or exploit an unreliable local signal. A persistently large reward can indicate that the student is sampling tokens on which the post-RL teacher and its reference strongly disagree—often because it has drifted away from both models' support, where the log-ratio is large in magnitude but no longer a trustworthy improvement direction. A near-zero batch mean suggests a more balanced regime, where local per-token shifts can still rank actions without a global drift.

Takeaway. The dense teacher/reference reward should be treated as rollout-distribution dependent rather than optimized in isolation. Fixed-KL sweeps show that the right constraint is pair-dependent, while adaptive KL empirically pulls the mean reward toward a more balanced regime.

5 Related Work

On-policy distillation of reasoning models. Reinforcement learning with verifiable rewards now produces strong reasoning models [1, 2, 4, 7, 9–17], a paradigm now standard in frontier systems [3, 18–22]. A standard way to spread such gains to other models is knowledge distillation, from classic and sequence-level distillation to its modern variants and scaling behavior [23–30], with a large body of work on divergences and estimators for distilling LLMs [31–39]. On-policy distillation (OPD), where the student is trained on its own sampled states under teacher supervision, has become the dominant variant for reasoning models and is the subject of rapid recent work [5, 8, 40–54, 54–57], including hybrids that couple distillation with RL [51, 58–62]. Analyses tie OPD success to teacher–student top-k overlap on student-visited states [5], note that small students struggle to copy much stronger reasoners [63], and reinterpret OPD as dense KL-constrained RL that rescales the reward to extrapolate beyond the teacher [64]. All of these imitate, or push past, the teacher's final policy; when the teacher is a small post-RL model weaker than the target, this also imports its capacity ceiling. Direct-OPD instead distills only the teacher's RL-induced policy shift $\log \pi_T - \log \pi_{T_{\text{ref}}}$, discarding the teacher's absolute policy.

Weak-to-strong generalization. Weak-to-strong generalization asks whether a weak supervisor can elicit capabilities from a stronger model [65, 66], building on a long line of learning from imperfect supervision—semi-supervised learning and learning from noisy labels [67–76]. It is central to scalable oversight when reliable human or stronger-model supervision is scarce [77–82], and is related to eliciting latent knowledge and to easy-to-hard generalization [83–86]. Recent work studies its theory and extends it to reasoning and automated alignment [87–91]. Most approaches still supervise with the weak model's labels or distribution, which caps the student near the supervisor; self-rewarding schemes bootstrap a model with its own judgments [92], and, closest to us, weak-to-strong preference optimization reuses a weak aligned model's implicit reward to steer a stronger one [93]. We share the goal of eliciting rather than imitating, but our supervisor is a pair of small RL checkpoints: we transfer the reward implied by their difference, and show it improves students that already exceed the post-RL teacher.

Implicit rewards and reusing trained models. Direct-OPD rests on the policy-as-reward identity: under KL-regularized RL the policy/reference log-ratio recovers the reward up to a constant, which Direct Preference Optimization and its variants use to fit a policy directly from preferences without an explicit reward model [6, 94–96]. We use this identity in reverse—reading a dense implicit reward off a post-RL checkpoint and its reference—which connects to dense and process rewards for reasoning [97–99] and to per-token credit assignment in RL [100, 101], while avoiding reward over-optimization from an explicit model [102].

A complementary line reuses trained models in weight space—task arithmetic, model merging, and proxy tuning [103–105]. Direct-OPD differs on both counts: the signal is a behavioral log-ratio rather than a weight delta, and it is applied on the student's own on-policy states, so it transfers across model families and scales.

6 Conclusion and Limitations

Direct-OPD shows that the policy shift learned by a small RL teacher is a useful dense reward for weak-to-strong generalization. The transferable object is not the post-RL teacher's policy, but its log-ratio against its own pre-RL reference, evaluated on student-visited tokens; this is why a smaller, weaker teacher can still improve stronger students. Empirically, this direction transfers across teacher pairs and student families, outperforms step-matched direct RL at a fraction of the compute, and composes across teachers. It reframes RL outcomes as reusable improvement signals rather than final models to imitate. **Limitations.** The signal is conditional: Direct-OPD can fail when the teacher/reference improvement is not meaningful on student-visited states, and the best response length and KL strength remain teacher–student dependent.

References

[1] DeepSeek-AI, 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. Nature, 645:633–638, 2025.

[2] Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, Ning Ding, and Zhiyuan Liu. JustRL: Scaling a 1.5B LLM with a simple RL recipe. arXiv preprint arXiv:2512.16649, 2025.

[3] 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.

[4] Chenxin An et al. POLARIS: A post-training recipe for scaling reinforcement learning on reasoning models. `https://github.com/ChenxinAn-fdu/POLARIS`, 2025.

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

[6] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290, 2023.

[7] Jiazheng Li, Hongzhou Lin, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Yi Wu, and Jingzhao Zhang. QuestA: Expanding reasoning capacity in LLMs via question augmentation. arXiv preprint arXiv:2507.13266, 2025.

[8] Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler II, Qian Qian, Gregory D. Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and effective on-policy distillation from reasoning prefixes. In Findings of the Association for Computational Linguistics: ACL 2026, pages 25553–25569, 2026.

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

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

[11] Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms. arXiv preprint arXiv:2501.12599, 2025.

[12] Qwen Team. QwQ-32B: Embracing the power of reinforcement learning. `https://qwenlm.github.io/blog/qwq-32b/`, 2025.

[13] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling reinforcement learning on the base model. `https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero`, 2025.

[14] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. DAPO: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476, 2025.

[15] Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. arXiv preprint arXiv:2505.22312, 2025.

[16] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. arXiv preprint arXiv:2504.11456, 2025.

[17] Bingxiang He, Yuxin Zuo, Zeyuan Liu, Shangziqi Zhao, Zixuan Fu, Junlin Yang, Cheng Qian, Kaiyan Zhang, Yuchen Fan, Ganqu Cui, et al. How far can unsupervised rlvr scale llm training? arXiv preprint arXiv:2603.08660, 2026.

[18] DeepSeek-AI. DeepSeek V4 preview release. `https://api-docs.deepseek.com/news/news260424`, 2026.

[19] Z.ai. GLM-5.2: Built for long-horizon tasks. `https://z.ai/blog/glm-5.2`, 2026.

[20] Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. AIMO-2 winning solution: Building state-of-the-art mathematical reasoning models with OpenMathReasoning dataset. arXiv preprint arXiv:2504.16891, 2025.

[21] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. arXiv preprint arXiv:2506.04178, 2025.

[22] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. 2024.

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[24] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In Proceedings of EMNLP, 2016.

[25] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108, 2019.

[26] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. 2020.

[27] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. 2020.

[28] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. 2020.

[29] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. 2019.

[30] Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. Distillation scaling laws. arXiv preprint arXiv:2502.08606, 2025.

[31] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. arXiv preprint arXiv:2306.08543, 2023.

[32] Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-Divergence Minimization for Sequence-Level Knowledge Distillation. Proceedings of ACL, 2023. URL https://arxiv.org/abs/2307.15190.

[33] Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. Revisiting Knowledge Distillation for Autoregressive Language Models. Proceedings of ACL, 2024. URL https://arxiv.org/abs/2402.11890.

[34] Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking Kullback-Leibler Divergence in Knowledge Distillation for Large Language Models. Proceedings of COLING, 2025. URL https://arxiv.org/abs/2404.02657.

[35] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards Streamlined Distillation for Large Language Models. Proceedings of ICML, 2024. URL https://arxiv.org/abs/2402.03898.

[36] Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A Contrastive Approach Boosts the Distillation of LLMs. In Proceedings of ICML, 2025. URL https://arxiv.org/abs/2503.07067.

[37] Yuxian Gu, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. MiniPLM: Knowledge Distillation for Pre-Training Language Models. Proceedings of ICLR, 2025. URL https://arxiv.org/abs/2410.17215.

[38] Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. DistillSpec: Improving Speculative Decoding via Knowledge Distillation. Proceedings of ICLR, 2024. URL https://arxiv.org/abs/2310.08461.

[39] 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. Proceedings of ICLR, 2025. URL https://arxiv.org/abs/2410.11325.

[40] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. arXiv preprint arXiv:2306.13649, 2023.

[41] Kevin Lu and Thinking Machines Lab. On-policy distillation. 2025.

[42] Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. arXiv preprint arXiv:2604.00626, 2026.

[43] Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. arXiv preprint arXiv:2603.25562, 2026.

[44] Woogyeol Jin, Taywon Min, Yongjin Yang, Swanand Ravindra Kadhe, Yi Zhou, Dennis Wei, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. arXiv preprint arXiv:2603.07079, 2026.

[45] Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. arXiv preprint arXiv:2601.07155, 2026.

[46] 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.

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

[48] Tianzhu Ye, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Online experiential learning for language models. arXiv preprint arXiv:2603.16856, 2026.

[49] 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.

[50] Guoliang Zhao, Ruobing Xie, An Wang, Shuaipeng Li, Huaibing Xie, and Xingwu Sun. Self-distillation for multi-token prediction, 2026. arXiv preprint arXiv:2603.23911, 2026.

[51] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement Learning via Self-Distillation. arXiv preprint arXiv:2601.20802, 2026. URL https://arxiv.org/abs/2601.20802.

[52] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. arXiv preprint arXiv:2601.19897, 2026.

[53] 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. URL https://arxiv.org/abs/2603.11178.

[54] Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. arXiv preprint arXiv:2603.11137, 2026.

[55] 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.

[56] Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-Box On-Policy Distillation of Large Language Models. arXiv preprint arXiv:2511.10643, 2025. URL https://arxiv.org/abs/2511.10643.

[57] 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.

[58] Shicheng Xu, Liang Pang, Yunchang Zhu, Jia Gu, Zihao Wei, Jingcheng Deng, Feiyang Pan, Huawei Shen, and Xueqi Cheng. RLKD: Distilling LLMs' Reasoning via Reinforcement Learning. arXiv preprint arXiv:2505.16142, 2025. URL https://arxiv.org/abs/2505.16142.

[59] Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. KDRL: Post-Training Reasoning LLMs via Unified Knowledge Distillation and Reinforcement Learning. arXiv preprint arXiv:2506.02208, 2025. URL https://arxiv.org/abs/2506.02208.

[60] Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware Knowledge Distillation for LLM Reasoning. arXiv preprint arXiv:2602.22495, 2026. URL https://arxiv.org/abs/2602.22495.

[61] Songming Zhang, Xue Zhang, Tong Zhang, Bojie Hu, Yufeng Chen, and Jinan Xu. AlignDistil: Token-Level Language Model Alignment as Adaptive Policy Distillation. In Proceedings of ACL, 2025. URL https://arxiv.org/abs/2503.02832.

[62] 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.

[63] Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. 2025.

[64] Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. arXiv preprint arXiv:2602.12125, 2026.

[65] Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. arXiv preprint arXiv:2312.09390, 2023.

[66] Jan Leike and Ilya Sutskever. Introducing Superalignment. OpenAI Blog, 2023.

[67] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. Advances in neural information processing systems, 17, 2004.

[68] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. Advances in neural information processing systems, 28, 2015.

[69] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242, 2016.

[70] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. arXiv preprint arXiv:1806.05594, 2018.

[71] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. Advances in neural information processing systems, 31, 2018.

[72] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32, 2019.

[73] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. arXiv preprint arXiv:2002.07394, 2020.

[74] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. Advances in neural information processing systems, 33:22243–22255, 2020.

[75] Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. Advances in Neural Information Processing Systems, 33:21061–21071, 2020.

[76] Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debiased self-training for semi-supervised learning. Advances in Neural Information Processing Systems, 35:32424–32437, 2022.

[77] Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. arXiv preprint arXiv:1810.08575, 2018.

[78] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. arXiv preprint arXiv:1811.07871, 2018.

[79] Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. arXiv preprint arXiv:1805.00899, 2018.

[80] Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamile Lukosuite, Amanda Askell, Andy Jones, Anna Chen, et al. Measuring progress on scalable oversight for large language models. arXiv preprint arXiv:2211.03540, 2022.

[81] Sam Bowman. Artificial Sandwiching: When can we test scalable alignment protocols without humans? AI Alignment Forum, 2022.

[82] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. arXiv preprint arXiv:2212.08073, 2022.

[83] Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge. Technical report, Alignment Research Center (ARC), 2022.

[84] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. arXiv preprint arXiv:2212.03827, 2022.

[85] Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. Advances in Neural Information Processing Systems, 34:6695–6706, 2021.

[86] Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Arpit Bansal, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. Datasets for studying generalization from easy to hard examples. arXiv preprint arXiv:2108.06011, 2021.

[87] Boxi Cao, Keming Lu, Xinyu Lu, Jiawei Chen, Mengjie Ren, Hao Xiang, Peilin Liu, Yaojie Lu, Ben He, Xianpei Han, Le Sun, Hongyu Lin, et al. Towards scalable automated alignment of llms: A survey. arXiv preprint arXiv:2406.01252, 2024.

[88] Wei Yao, Wenkai Yang, Ziqiao Wang, Yankai Lin, and Yong Liu. Revisiting weak-to-strong generalization in theory and practice: Reverse kl vs. forward kl. arXiv preprint arXiv:2502.11107, 2025.

[89] Wei Yao, Gengze Xu, Huayi Tang, Wenkai Yang, Donglin Di, Ziqiao Wang, and Yong Liu. On weak-to-strong generalization and f-divergence. arXiv preprint arXiv:2506.03109, 2025.

[90] Jitao Sang, Yuhang Wang, Jing Zhang, Yanxu Zhu, Chao Kong, Junhong Ye, Shuyu Wei, and Jinlin Xiao. Improving weak-to-strong generalization with scalable oversight and ensemble learning. arXiv preprint arXiv:2402.00667, 2024.

[91] Yige Yuan, Teng Xiao, Shuchang Tao, Xue Wang, Jinyang Gao, Bolin Ding, and Bingbing Xu. Incentivizing strong reasoning from weak supervision. arXiv preprint arXiv:2505.20072, 2025.

[92] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. arXiv preprint arXiv:2401.10020, 2024.

[93] Wenhong Zhu, Zhiwei He, Xiaofeng Wang, Pengfei Liu, and Rui Wang. Weak-to-strong preference optimization: Stealing reward from weak aligned model. arXiv preprint arXiv:2410.18640, 2025.

[94] Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. AISTATS, 2024.

[95] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. arXiv preprint arXiv:2402.01306, 2024.

[96] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. NeurIPS, 2024.

[97] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. arXiv preprint arXiv:2305.20050, 2023.

[98] Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui. Math-shepherd: A label-free step-by-step verifier for LLMs in mathematical reasoning. arXiv preprint arXiv:2312.08935, 2023.

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

[100] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Unlocking rl potential for llm reasoning through refined credit assignment. arXiv preprint arXiv:2410.01679, 2024.

[101] Chenchen Zhang et al. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. arXiv preprint arXiv:2604.09459, 2026.

[102] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. ICML, 2023.

[103] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In ICLR, 2023.

[104] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In ICML, 2022.

[105] Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. Tuning language models by proxy. arXiv preprint arXiv:2401.08565, 2024.

A. Experimental Details

Direct-OPD data and prompt. For the final Direct-OPD runs, we use the math subset of Skywork-OR1-RL-Data, released with Skywork-OR1 [15], and apply the following DAPO-style math prompt template.

```
Solve the following math problem step by step.
The last line of your response should be of the form
Answer:  $Answer (without quotes) where $Answer is the answer to the problem.

{Question}

Remember to put your answer on its own line after "Answer:".
```

We use this template for Direct-OPD training rollouts and evaluation prompts. This template differs from the boxed-answer prompt used during teacher RL; in our runs, the DAPO-style prompt gives slightly better transfer. Since prompt template divergence is not the focus of this paper, we use this setting throughout. We also observe similar transfer trends when replacing the Skywork training dataset with DAPO-Math-17K while keeping the same prompt format, suggesting that the result is not specific to a single training dataset.

Evaluation. Table 2 summarizes the evaluation protocol used for all reported AIME results unless otherwise specified.

Evaluation setting	Value
Benchmarks	AIME 2024, AIME 2025
Samples per problem	32
Sampling temperature	0.7
Top-p	0.95
Maximum generation length	31,744

Table 2 Evaluation protocol for AIME validation.

Table 3 lists the default Direct-OPD training hyperparameters used with the data and prompt format above.

Hyperparameter	Value
Training framework	`verl`
Global batch size	64
Mini batch size	64
Rollout n	4
Maximum prompt length	1,024
Maximum response length	2,048
Sampling temperature	1.0
Top-p	1.0
Learning rate	1×10^{-6}
Training steps	300
KL coefficient	$[0.8, 2]$ for different student–teacher pairs if not using adaptive KL
Student top-k support	16
Top-k strategy	Student top-k

Table 3 Training hyperparameters for Direct-OPD unless otherwise specified.

Table 4 reports the GRPO settings used to train RL teachers and direct-RL baselines.

R1-Distill-1.5B RL

Hyperparameter	Value
Algorithm	GRPO
Train batch size	512
PPO mini batch size	128
Rollout n	8
Maximum prompt length	2,048
Maximum response length	16,384
Temperature	1.0
Learning rate	1×10^{-6}
KL coefficient	0.0
Clip high	0.28
Clip low	0.2

R1-Distill-7B RL

Hyperparameter	Value
Algorithm	GRPO
Train batch size	512
PPO mini batch size	128
Rollout n	8
Maximum prompt length	2,048
Maximum response length	16,384
Temperature	1.0
Learning rate	1×10^{-6}
KL coefficient	0.0
Clip high	0.28
Clip low	0.2

Qwen3-1.7B-nonthinking RL

Hyperparameter	Value
Algorithm	GRPO
Train batch size	128
PPO mini batch size	128
Rollout n	8
Maximum prompt length	2,048
Maximum response length	16,384
Temperature	1.0
Learning rate	1×10^{-6}
KL coefficient	0.0

Qwen3-4B-nonthinking RL

Hyperparameter	Value
Algorithm	GRPO
Train batch size	128
PPO mini batch size	128
Rollout n	8
Maximum prompt length	2,048
Maximum response length	16,384
Temperature	1.0
Learning rate	1×10^{-6}
KL coefficient	0.0

Table 4 RL training settings for teacher construction and direct-RL baselines. We follow the math GRPO setting of Yang et al. [64], using train batch size 512 for R1-Distill runs, train batch size 128 for Qwen3-nonthinking runs, and PPO mini batch size 128 throughout.

B. Additional Entropy Diagnostics

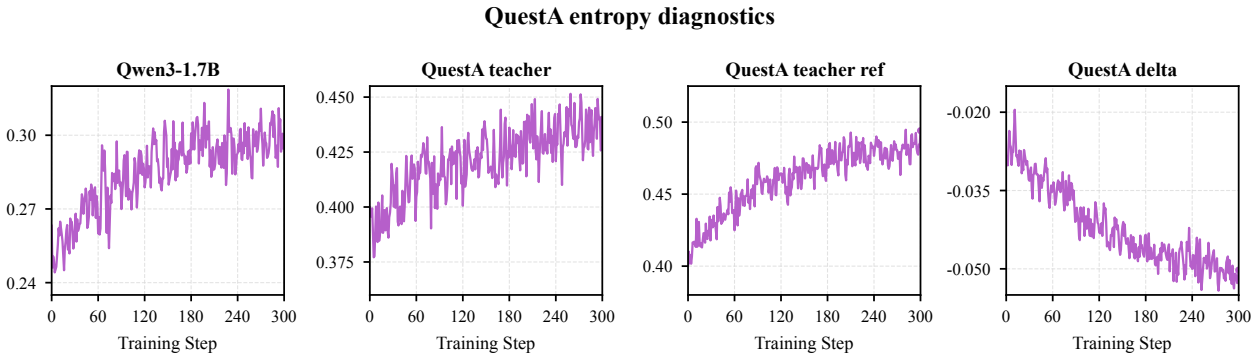


Figure 10 Entropy diagnostics for QuestA-Nemotron into Qwen3-1.7B. The panels show student entropy, post-RL teacher entropy, teacher-reference entropy, and teacher entropy minus reference entropy. Together with Figure 6, this shows that the non-collapse pattern is not specific to the JustRL teacher pair.

C. Additional Results

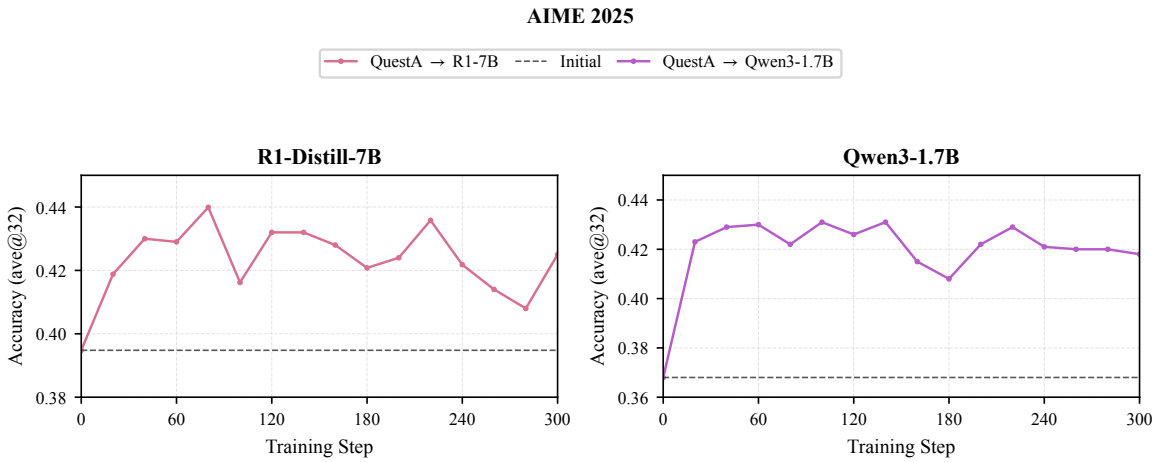


Figure 11 QuestA transfer curves on AIME 2025 for the cross-pattern transfer setting. The main text reports the corresponding AIME 2024 curves.