# WEAK-TO-STRONG GENERALIZATION VIA DIRECT ON-POLICY DISTILLATION

**Anonymous authors**
Paper under double-blind review

## 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 at 600–1500 teacher RL steps 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.

## 1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a dominant post-training recipe for eliciting strong reasoning in large language models (DeepSeek-AI et al., 2025; He et al., 2025a). 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: run RL on a small model, then reuse its before/after checkpoint pair to train stronger students. The post-RL teacher need not outperform the student. What matters is whether its RL-induced change supplies useful supervision on the student's own states. This makes the same small-model RL run reusable across target models.

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 (DeepSeek-AI et al., 2025) begins at 56.7, already above the post-RL JustRL-1.5B teacher (He et al., 2025a) 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.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

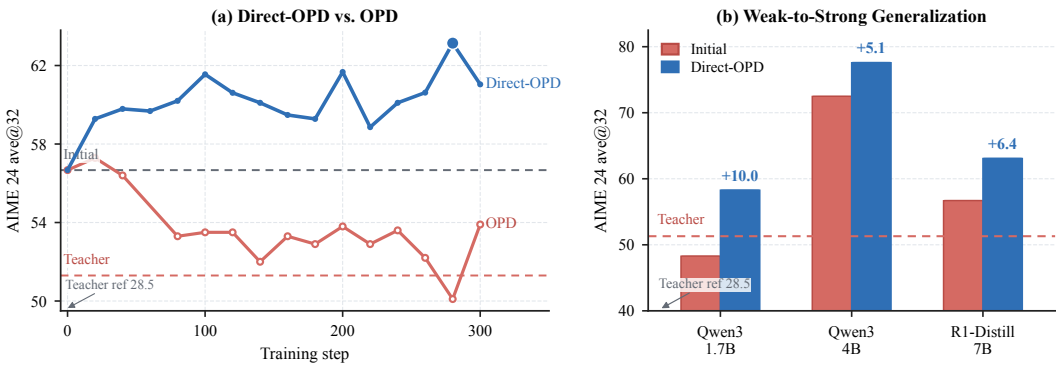


Figure 1: **Direct-OPD transfers the effect of small-model RL rather than imitating the small model.** (a) Starting from R1-Distill-7B, vanilla OPD toward the post-RL JustRL-1.5B teacher degrades performance, whereas Direct-OPD transfers the JustRL-1.5B − R1-Distill-1.5B policy shift and improves the 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. Panel (a) tracks student transfer steps; panel (b) compares initial and best validation scores. Both panels report AIME 2024 ave@32.

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 $\pi_T$ 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 measures change relative to the same teacher's starting point. The teacher/reference log-ratio admits an implicit-reward interpretation under KL-regularized RL (Section 2.2). Direct-OPD applies this signal on the student's own on-policy states, with a KL anchor to the student's initialization, without training an explicit reward model or running sparse-reward RL on the target.

The R1-Distill-1.5B → JustRL-1.5B pair provides the strict weak-to-strong test: Qwen3-4B (Yang et al., 2025) and R1-Distill-7B already outperform JustRL on both AIME benchmarks. At transfer step 300, they improve by 3.87/1.38 and 4.37/6.24 percentage points on AIME 2024/2025, respectively (Table 5). Figure 1 shows the same shift benefiting Qwen3-1.7B; its best AIME 2024 score rises from 48.3 to 58.3 with about 4 hours on 8 A100 GPUs. A second, stronger QuestA teacher pair tests robustness across RL pipelines. Together with the direct-RL route comparison and training diagnostics, these results specify where reuse of a small-model RL shift is effective.

**Contributions.**

- **A weak-to-strong post-training paradigm.** Direct-OPD transfers a small teacher's RL-induced policy shift as an implicit reward on the stronger student's own on-policy states.
- **Consistent gains across teachers and students.** Across two teacher pairs and three students from two model families, Direct-OPD improves every tested student, including those initially stronger than the post-RL teacher.
- **A lower-cost route than direct RL in the tested setting.** At 600–1500 RL steps, small-model RL followed by Direct-OPD achieves higher accuracy with less total compute than direct RL on R1-Distill-7B.
- **Transfer analysis.** We show cross-pattern transfer at low teacher–student overlap and identify response-length and KL conditions under which the implicit reward aligns 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 $\pi$ 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 $\pi_\theta$ and its initial checkpoint $\pi_S$; and a teacher pair consisting of a pre-RL reference $\pi_{T_{\text{ref}}}$ and the corresponding post-RL teacher $\pi_T$. Our weak-to-strong setting is the regime in which $\pi_T$ may be smaller or weaker than the target student, yet the transition from $\pi_{T_{\text{ref}}}$ to $\pi_T$ 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 $S_t$ (Li et al., 2026b). 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 $\pi_T$ 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. Under KL-regularized RL, this log-ratio admits an implicit-reward interpretation.

For a reward $r$, reference policy $\pi_{\text{ref}}$, 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 (Rafailov et al., 2023). Here we use the identity in the opposite direction: we are already *given* a post-RL teacher $\pi_T$ and its pre-RL reference $\pi_{T_{\text{ref}}}$, and we read the reward-like signal back out of their policy ratio. For a chosen scale $\beta > 0$, the checkpoint pair admits a latent-reward representation in which $\pi_T$ is the KL-regularized optimum for $r_T$ anchored at $\pi_{T_{\text{ref}}}$,

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

where $\beta$ sets the scale of this representation independently of the teacher's training KL coefficient. The shift $\Delta_T$ can thus 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 $\pi_S$, Direct-OPD optimizes the student against the teacher-shift signal $\Delta_T$ while regularizing the student toward its own initialization,

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

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 $\pi_S$ 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 $S_t = \mathrm{TopK}_v\,\pi_\theta(v \mid s_t)$ 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 $y \sim \pi_\theta$:

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

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 $\theta$ 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,

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

In practice we do not differentiate the KL analytically; we use the standard KL-penalty implementation in `verl` to anchor $\pi_\theta$ to $\pi_S$.

Appendix C specifies the conditional action expectation and the retained-support mass term of this local surrogate.

### 2.4 Adaptive KL Control

The teacher/reference log-ratio is fixed by the checkpoint pair, while the signal encountered during transfer also depends on the student's visited prefixes and candidate tokens. Its distribution can therefore vary across teacher–student pairs and change during training. In Eq. 9, $\alpha$ controls the strength of the teacher shift relative to the student's KL anchor, so a fixed coefficient need not provide the same balance across settings. We therefore adapt $\alpha$ during training, using the sign of the student-weighted mean shift as a lightweight feedback signal. The update rule is given below.

Let $\alpha_m$ 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; this statistic provides the controller feedback. Before the actor update we set

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

with $\text{sgn}(0) = 0$; by default $\epsilon = 0.01$ and $[\alpha_{\min}, \alpha_{\max}] = [0.5, 2.5]$. Here $\text{sgn}(\bar{r}_m)$ 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* $\alpha$ to discourage over-amplifying that local signal; when the average shift is negative, it *lowers* $\alpha$, 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)

**Experimental setup.** Final transfer runs use Skywork-OR1-RL-Data's math subset (He et al., 2025b), a DAPO-style prompt, 300 training steps, 2,048-token responses, student top-16 support, and learning rate $10^{-6}$. AIME 2024/2025 validation averages 32 samples per problem at temperature 0.7, top-$p = 0.95$, and maximum generation length 31,744. Tables 2–4 report the prompts, shared settings, fixed-KL range $[0.8, 2]$, and RL recipes; Section 2.4 specifies adaptive KL. Table 1 uses benchmark-wise best scores; cost points use the 300-step transfer endpoint, and Figure 4 compares benchmark-averaged scores at step 40.

Table 1: AIME scores (percent) for the two teacher pairs and their students. For Direct-OPD, we report the best AIME scores observed during training; the two benchmarks may reach their maxima at different checkpoints. The final block follows Qwen3-1.7B through sequential JustRL and QuestA policy-shift transfer.

| Teacher shift | Model / stage | AIME24 | AIME25 |
|---|---|---|---|
| **JustRL** (He et al., 2025a) | Teacher reference | 28.5 | 24.0 |
| | Post-RL teacher | 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** $^{+8.3}$ |
| **QuestA** (Li et al., 2025a) | Teacher reference | 61.77 | 49.50 |
| | Post-RL teacher | 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** $^{+4.5}$ |
| **Sequential** | Initial Qwen3-1.7B | 48.3 | 36.8 |
| | After JustRL | **58.3** $^{+10.0}$ | **43.2** $^{+6.4}$ |
| | After QuestA | **63.8** $^{+15.5}$ | **46.8** $^{+10.0}$ |

Superscripts: gains (percentage points) over the corresponding student's initial score.

### 3.1 A SMALL RL TEACHER IMPROVES MUCH STRONGER STUDENTS

The R1-Distill-1.5B $\rightarrow$ JustRL-1.5B pair improves three students, including R1-Distill-7B and Qwen3-4B, which already exceed the post-RL teacher. The Nemotron-1.5B $\rightarrow$ QuestA-Nemotron-1.5B pair tests robustness across training pipelines and data; because QuestA is already strong on AIME, it is not a strict weaker-teacher comparison. Table 1 reports benchmark-wise maxima, including sequential composition; Figures 5 and 6 provide full transfer curves.

The gains are also present at the final transfer checkpoint (step 300), with paired AIME 2024/2025 results reported in Appendix B (Table 5).

### 3.2 WEAK-TO-STRONG GENERALIZATION BEATS RL

We train R1-Distill-1.5B on DAPO and transfer checkpoints at 300, 600, 900, 1200, and 1500 RL steps into R1-Distill-7B. For each $N$-step small-model RL checkpoint, we run 300 Direct-OPD steps on the student and compare the result with $N$ steps of direct student RL. On AIME 2025, the 600–1500-step teacher checkpoints yield higher scores than direct R1-Distill-7B RL (Figure 2). The cost of Direct-OPD includes both small-model RL and student transfer. A 1500-step small-model RL run takes about 160 hours on 32 A100 GPUs, versus about 320 hours for 7B RL; transfer adds about four hours on eight A100s. At 1500 RL steps, the total is 5,152 A100 GPU-hours, compared with 10,240 for direct R1-Distill-7B RL. With Qwen3 nonthinking models, transferring a 100-step 1.7B RL shift into 4B reaches the 0.635 direct-RL accuracy on AIME 2025. Teacher and direct-RL settings are in Table 4.

### 3.3 SEQUENTIAL COMPOSITION

For Qwen3-1.7B, applying the JustRL shift and then the QuestA shift raises AIME 2024/2025 scores from 48.3/36.8 to 58.3/43.2 and then 63.8/46.8. Figure 7 in Appendix D.2 shows the training trajectories; Table 1 reports the best scores observed within each stage. The second stage continues the first-stage checkpoint; curves align its local steps 0–300 to global steps 300–600. Separate evaluation samples explain the small stage-boundary discontinuity. The result shows that policy shifts

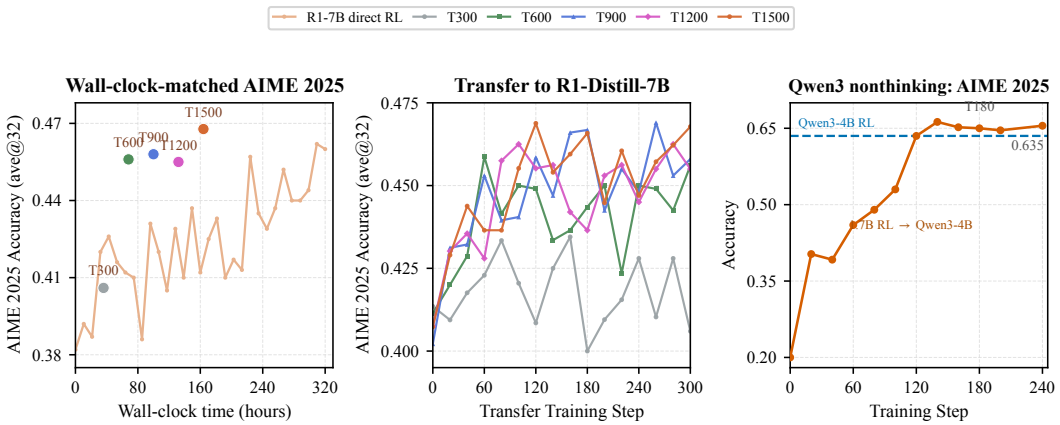


Figure 2: **Small-model RL followed by Direct-OPD versus direct RL on the larger student.** For R1-Distill-7B, we compare direct RL with a two-stage route: RL on R1-Distill-1.5B followed by Direct-OPD transfer into R1-Distill-7B. In the left and middle panels, $T_N$ uses the small model's checkpoint after $N$ RL steps as the post-RL teacher and its initial checkpoint as the reference, with $N \in \{300, 600, 900, 1200, 1500\}$. **Left:** AIME 2025 accuracy versus estimated total wall-clock time. The curve shows direct R1-Distill-7B RL; each $T_N$ point reports the student's accuracy after 300 Direct-OPD steps, with time accounting for both small-model RL and student transfer. Intermediate RL times are linearly interpolated from the reported 1500-step runtimes; RL-step matching does not include the additional 300 student transfer steps. **Middle:** The corresponding Direct-OPD learning curves for the five teacher checkpoints. The horizontal axis counts student transfer steps. **Right:** Transfer from Qwen3-1.7B to Qwen3-4B using nonthinking models. The solid curve shows Direct-OPD training, while the dashed line marks the AIME 2025 accuracy of direct Qwen3-4B RL (0.635). Here, T180 marks student transfer step 180.

learned by different teachers can be applied sequentially to further improve the same student, without imitating either teacher's full policy.

## 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

**Direct-OPD improves student performance even at low teacher–student top-$k$ overlap.** Standard OPD has been associated with increasing teacher–student overlap on high-probability tokens (Li et al., 2026b). To test whether our gains require the same behavior, let $T_k^S(s)$ and $T_k^C(s)$ be the top-$k$ token sets of the student and comparison policy at a student-visited state $s$. We measure $|T_k^S(s) \cap T_k^C(s)|/k$ against both the post-RL teacher and its pre-RL reference on the same states.

Figure 3 distinguishes pattern-aligned R1-Distill transfer from the cross-pattern Qwen3 transfer. The former enters a higher-overlap regime; the latter retains lower overlap with both checkpoints despite improving accuracy. The teacher shift can therefore rank actions within the student's own candidate set without requiring full-distribution imitation. This is a diagnostic of the observed transfers, not a

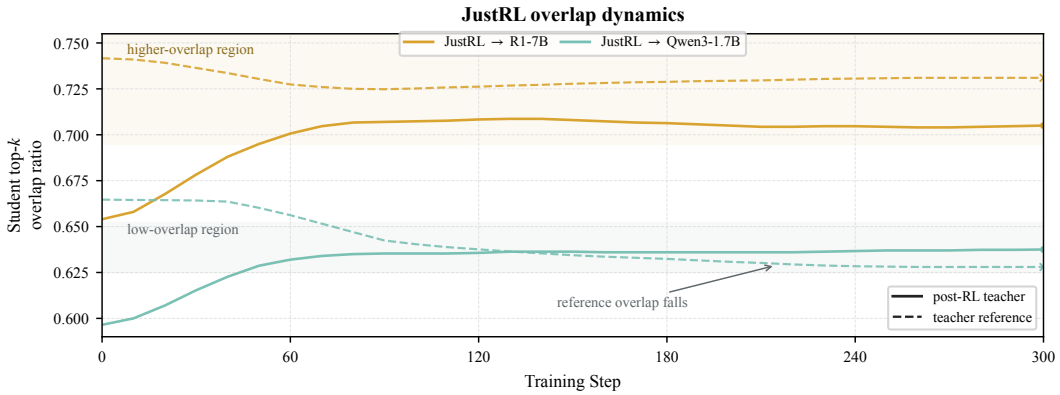


Figure 3: Token-overlap diagnostic for the R1-Distill-1.5B to JustRL-1.5B teacher shift. The students are R1-Distill-7B (pattern-aligned) and Qwen3-1.7B (cross-pattern). The vertical axis is the fraction of shared tokens between the student's and comparison policy's top-$k$ sets at student-visited prefixes; the horizontal axis counts Direct-OPD training steps. Solid curves compare the student with the post-RL teacher; dashed curves compare it with the reference. Pattern-aligned transfer and cross-pattern transfer occupy different overlap regimes. Accuracy improves in both settings (Table 1), although the cross-pattern Qwen3 student remains in a lower-overlap regime with both checkpoints. Curves use centered moving averages of three points for the post-RL teacher and five for the reference.

causal isolation of their mechanism. QuestA overlap and entropy diagnostics extend this comparison and show no accompanying student-entropy collapse (Figures 8, 9, and 10).

### 4.2 Short-horizon training changes longer-rollout behavior

Following prefix-distillation work (Zhang et al., 2026a), we train with a 2k response horizon and test longer generations (Figure 4). On 64 fixed long rollouts, the cumulative teacher/reference gap over actor top-16 tokens shifts after 40 training steps even beyond the 2k supervised prefix, across roughly 16k positions. Specifically, at each fixed prefix $s_t$ we weight the teacher shift by the evaluated actor's probabilities on its top-16 set $K_t$:

$$G_T = \sum_{t=1}^{T} \sum_{v \in K_t} \pi_{\text{actor}}(v \mid s_t) \left[\log \pi_{\text{JustRL}}(v \mid s_t) - \log \pi_{\text{R1}}(v \mid s_t)\right]. \tag{17}$$

Here the plotted diagnostic uses the actor probabilities directly, rather than the renormalized top-$k$ weights of the training surrogate. Higher $G_T$ means that the likely actor tokens receive a larger teacher/reference log-ratio; it does not measure the correctness of the completed response. A 6k-trained actor shifts farther in this diagnostic but validates worse (45.6 versus 48.8 for 2k). Unreliable late-prefix teacher signals are a plausible explanation, not an established causal conclusion. The diagnostic setup and length sweeps are in Appendix E.2 (Figure 11).

### 4.3 KL controls teacher-shift reward reliability

The controller in Section 2.4 adjusts the student's KL constraint in response to the observed teacher shift. We test sensitivity by comparing fixed coefficients and adaptive KL with the same 2k response horizon, tracking validation accuracy and dense token reward.

The KL sweeps in Appendix E.3 (Figure 12) show why a single fixed coefficient is insufficient: different pairs favor different constraints, and dense reward alone does not give a pair-independent selection rule. Weak constraints can permit drift toward states where both teachers have low support; strong constraints can suppress useful transfer. Adaptive KL moves the batch mean reward toward zero after an initial correction, consistent with a balanced signal rather than unconstrained reward maximization. A near-zero mean is not proof of reward reliability, nor do these sweeps establish that adaptive KL dominates every well-tuned fixed coefficient.

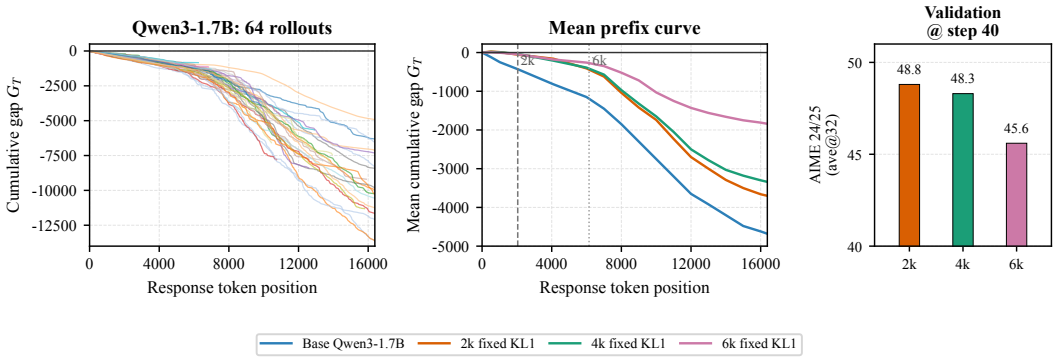


Figure 4: **Short-horizon transfer changes longer-rollout behavior.** On 64 fixed long rollouts from the base Qwen3-1.7B actor, $G_T$ is the cumulative teacher/reference gap through token position $T$: at each prefix, actor probabilities weight the JustRL-1.5B minus R1-Distill-1.5B log-probability difference over the actor's top-16 tokens. **Left:** one cumulative-gap curve per base-actor rollout. **Middle:** mean $G_T$ for the base actor and actors trained for 40 steps with 2k/4k/6k response lengths and fixed KL coefficient 1. The horizontal axes in the left and middle panels show response token position; the vertical dashed line marks the 2k training horizon, while the curves extend to roughly 16k positions. **Right:** the average of AIME 2024 and AIME 2025 accuracy (ave@32) at the same 40-step checkpoints. A larger gap need not yield higher accuracy. Additional diagnostic details are provided in Appendix E.2.

## 5 Related Work

**On-policy distillation.** OPD trains on student-generated states (Agarwal et al., 2023; Gu et al., 2023; Ko et al., 2025); recent work studies token overlap, stability, and prefix supervision (Li et al., 2026b; Jang et al., 2026; Zhang et al., 2026a). G-OPD (Yang et al., 2026b) varies the reward/KL balance and studies pre-RL teacher checkpoints as reward references. Its displayed objective uses one reference for both terms. Direct-OPD instead defines the shift against the pre-RL teacher and anchors updates to the student's initialization.

**Weak-to-strong and implicit rewards.** Weak-to-strong generalization seeks to elicit stronger capabilities from weak supervision (Burns et al., 2023). WSPO (Zhu et al., 2025) also transfers a weak aligned/reference log-ratio through offline ratio matching. Direct-OPD instead uses a small model's RL-induced shift as a dense signal on the student's own on-policy states. The teacher's pre-RL reference identifies the RL-induced change, while the student's initialization anchors its update. This transfers the teacher's improvement without requiring imitation of its entire distribution. Our weak-to-strong post-training paradigm reuses small-model RL across stronger students, supported by empirical evidence on transfer, cost, sequential composition, and training dynamics.

**Reusing model updates.** Task arithmetic and model soups combine model weights (Ilharco et al., 2023; Wortsman et al., 2022). Proxy-tuning applies a tuned-minus-untuned small-model logit difference at decoding time (Liu et al., 2024). Here the shift trains the student's parameters. Additional background is in Appendix F.

## 6 Conclusion and Limitations

Direct-OPD introduces a weak-to-strong post-training paradigm that reuses small-model RL shifts to improve stronger students. It evaluates the teacher's RL-induced change on student-visited states while anchoring the student to its initialization. Across the studied teacher pairs and student families, it improves reasoning, lowers cost relative to direct large-model RL in the tested setting, and supports sequential transfer. These results position RL outcomes as reusable improvement signals rather than only final models to imitate. **Limitations.** Transfer requires useful teacher shifts on student-visited states; suitable response lengths and KL strengths remain pair-dependent.

## AI USE STATEMENT

AI tools were used to assist with polishing portions of the text, drafting selected passages, and searching for literature for related work and early-stage research. All AI-assisted content was reviewed by the authors.

## REPRODUCIBILITY STATEMENT

Section 2 specifies the objective, token-level surrogate, and adaptive KL controller. Tables 2, 3, and 4 give evaluation and training settings. The appendix retains full training curves, response-length and KL sweeps, and additional entropy diagnostics.

## REFERENCES

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.

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

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.

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.

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.

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.

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

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.

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

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.

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

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.

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.

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, 2020a.

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, 2020b.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4794–4802, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Cho_On_the_Efficacy_of_Knowledge_Distillation_ICCV_2019_paper.html.

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

Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge. Technical report, Alignment Research Center (ARC), 2021. URL https://www.alignment.org/blog/arcs-first-technical-report-eliciting-latent-knowledge/.

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:2502.01456v1*, 2025.

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

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948v1*, 2025.

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

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562v1*, 2026.

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

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543v5*, 2023.

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.

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.

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.

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*, 2025a.

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.

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*, 2025b.

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*, 2025c.

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

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

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.

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

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

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. *arXiv preprint arXiv:2601.07155v1*, 2026.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, 2020. doi: 10.18653/v1/2020.findings-emnlp.372. URL https://aclanthology.org/2020.findings-emnlp.372/.

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:2603.07079v1*, 2026.

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:2410.01679v1*, 2024.

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.

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

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

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.

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.

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.

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

Jan Leike and Ilya Sutskever. Introducing Superalignment. *OpenAI Blog*, 2023.

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.

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*, 2026a.

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. Hugging Face dataset repository, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-CoT.

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*, 2025a.

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

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*, 2026b.

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. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25366–25394, 2025b. doi: 10.18653/v1/2025.findings-acl.1301. URL https://aclanthology.org/2025.findings-acl.1301/.

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.

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.

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

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

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5191–5198, 2020. doi: 10.1609/aaai.v34i04.5963. URL https://ojs.aaai.org/index.php/AAAI/article/view/5963.

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.

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

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.

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.

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.

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

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*, 2021a.

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, 2021b.

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.

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

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

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:2312.08935v1*, 2023.

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. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

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.

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.

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.

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.

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*, 2025a. URL https://arxiv.org/abs/2506.02208.

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*, 2025b. URL https://arxiv.org/abs/2505.16142.

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*, 2025c. URL https://arxiv.org/abs/2410.11325.

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.

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.

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*, 2026a.

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*, 2026b.

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*, 2025a.

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*, 2025b.

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.

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

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

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:2503.14476v1*, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020v1*, 2024.

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.

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

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. *arXiv preprint arXiv:2604.09459*, 2026.

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*, pp. 25553–25569, 2026a.

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.

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*, 2026b. URL https://arxiv.org/abs/2602.22495.

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

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*, 2026b.

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.

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.

Wenhong Zhu, Zhiwei He, Xiaofeng Wang, Pengfei Liu, and Rui Wang. Weak-to-strong preference optimization: Stealing reward from weak aligned model. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=f7KxfUrRSb.

# 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 (He et al., 2025b), 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.

Table 2: Evaluation protocol for AIME validation.

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

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

| 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 | $1 \times 10^{-6}$ |
| 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 4 reports the GRPO settings used to train the local RL teachers and direct-RL baselines in Section 3.2.

Table 4: RL training settings for the local teacher construction and direct-RL baselines in Section 3.2. We follow the math GRPO setting of Yang et al. (2026b), using train batch size 512 for R1-Distill runs, train batch size 128 for Qwen3-nonthinking runs, and PPO mini batch size 128 throughout.

**R1-Distill RL 1.5B and 7B**

| 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 | $1 \times 10^{-6}$ |
| KL coefficient | 0.0 |
| Clip high | 0.28 |
| Clip low | 0.2 |

**Qwen3-nonthinking RL 1.7B and 4B**

| 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 | $1 \times 10^{-6}$ |
| KL coefficient | 0.0 |

## B Result Reporting and Interpretation

The summary table selects each benchmark's maximum within a transfer run; the pair of maxima need not identify a single checkpoint. Cost-route points instead use 300 transfer steps, while the response-length probe uses 40-step checkpoints. These summaries answer different questions.

**Common 300-step endpoints and persistence.** Table 5 reports both benchmarks at the same 300-step checkpoint of each saved transfer trajectory. For both initially stronger JustRL students, gains remain positive at all 15 post-initialization checkpoints on each benchmark. All five transfers finish above their corresponding initial scores on both benchmarks.

The counts describe one training trajectory per pair. They measure persistence along that trajectory, not independent replications or statistical significance. The common endpoint avoids selecting a separate checkpoint for each benchmark, but remains a descriptive summary of the same validation curves. We report aggregate validation scores without per-problem confidence intervals or independent-seed uncertainty estimates. The QuestA–Qwen3-1.7B curve starts at 49.10 on AIME 2024, whereas Table 1 reports 48.3; we retain these distinct initial measurements and use 49.10 for the endpoint gain in Table 5.

| Teacher shift | Student | Initial 24 / 25 | Step 300 24 / 25 | Gain (pp) 24 / 25 | Above initial 24 / 25 |
|---|---|---|---|---|---|
| JustRL | Qwen3-1.7B | 48.30 / 36.80 | 57.82 / 42.82 | 9.52 / 6.02 | 15 / 15 |
| | Qwen3-4B | 72.50 / 65.60 | 76.37 / 66.98 | 3.87 / 1.38 | 15 / 15 |
| | R1-Distill-7B | 56.67 / 40.53 | 61.04 / 46.77 | 4.37 / 6.24 | 15 / 15 |
| QuestA | Qwen3-1.7B | 49.10 / 36.80 | 56.10 / 41.80 | 7.00 / 5.00 | 15 / 15 |
| | R1-Distill-7B | 56.25 / 39.48 | 60.84 / 42.50 | 4.59 / 3.02 | 14 / 15 |

Table 5: Paired AIME 2024/2025 scores (percent) at step 300, from the saved plotting series for Figures 5 and 6. Gains use each series' own step-0 score. The final column counts saved checkpoints at steps 20, 40, . . ., 300 strictly above that initial score, out of 15 on each benchmark.

**Main results and hyperparameter sweeps.** Table 3 reports shared transfer settings, including the fixed-KL range $[0.8, 2]$ across pairs; it does not assign a coefficient to every result row. Section 2.4 gives the adaptive rule, and Figure 12 compares fixed and adaptive settings within each teacher–student pair. These sweep summaries are distinct from the main transfer curves: the JustRL–Qwen3-1.7B main curve starts at $42.55\%$ averaged across AIME 2024/2025, whereas the corresponding KL

sweep starts at $44.18\%$. We retain these reported initial measurements and interpret the sweep contrasts within their own figures, without combining the curves into a common-run comparison.

## C Local Update and Conditional Action Interpretation

At a sampled prefix $s$, hold the retained set $S$ locally fixed and write $p_\theta(v) = \pi_\theta(v \mid s)$, $m_\theta = \sum_{v \in S} p_\theta(v)$, and $\bar{p}_\theta(v) = p_\theta(v)/m_\theta$. The teacher shift $r(v)$ is fixed with respect to $\theta$. For the restricted mean $U_\theta = \sum_{v \in S} \bar{p}_\theta(v) r(v)$, the reward update in Section 2.3 satisfies

$$g_\theta(s) = \sum_{v \in S} \bar{p}_\theta(v) r(v) \nabla_\theta \log p_\theta(v) = \nabla_\theta U_\theta + U_\theta \nabla_\theta \log m_\theta. \tag{18}$$

This follows from $\nabla \log \bar{p}_\theta(v) = \nabla \log p_\theta(v) - \nabla \log m_\theta$. Thus the stated update changes both preferences within $S$ and its total probability mass; it is not simply the gradient of the normalized restricted mean. Adding a constant $c$ to all retained rewards adds $c \nabla \log m_\theta$ to the update. This identity describes the written surrogate away from changes in top-$k$ membership, without claiming that the mass term was separately tested as a design choice.

For $V \sim p_\theta(\cdot \mid s)$, the conditional law given $V \in S$ is $\bar{p}_\theta$, so $g_\theta(s) = \mathbb{E}[r(V) \nabla \log p_\theta(V) \mid s, V \in S]$. The analytical sum removes action-sampling variance for this conditional distribution; it does not preserve the full-vocabulary expectation, which equals

$$m_\theta g_\theta(s) + \sum_{v \notin S} p_\theta(v) r(v) \nabla_\theta \log p_\theta(v). \tag{19}$$

This is the restricted-action interpretation of Rao–Blackwellization: prefixes remain sampled from the full student, and the truncation and trajectory-sampling effects remain.

At an update point $\theta_0$, freezing sampled prefixes, retained sets, and coefficients gives the local reward surrogate

$$\ell(\theta; \theta_0, s) = \sum_{v \in S} \mathrm{stop_gradient}[\bar{p}_{\theta_0}(v) r(v)] \log p_\theta(v), \qquad \nabla_\theta \ell|_{\theta_0} = g_{\theta_0}(s). \tag{20}$$

The coefficients are refreshed with the student. This is a local objective, not a global objective differentiated through state visitation or top-$k$ selection; Eq. 9 characterizes the idealized sequence objective, not a convergence guarantee for this update. The conditional interpretation identifies what the local action sum computes; it does not turn the truncated immediate-reward update into an exact gradient of the sequence objective.

**Implicit reward and teacher training.** Equation 7 describes a reward representation of the observed teacher checkpoint pair, rather than a reconstruction of its training procedure. For a fixed prompt $x$, assume common positive support and finite relevant expectations. For any $\beta > 0$, define $r_T(x, y) = \beta \Delta_T(y \mid x)$. Then

$$\mathbb{E}_{y \sim \pi(\cdot|x)}[r_T(x, y)] - \beta D_{\mathrm{KL}}\big(\pi(\cdot \mid x) \,\|\, \pi_{T_{\mathrm{ref}}}(\cdot \mid x)\big) = -\beta D_{\mathrm{KL}}\big(\pi(\cdot \mid x) \,\|\, \pi_T(\cdot \mid x)\big).$$

Thus, over unrestricted policies on this support, $\pi_T$ maximizes this represented reward objective. Here $\beta$ is a choice of representation scale, independent of the KL coefficient used to train the teacher; the interpretation also applies when that training coefficient is zero. It does not identify $r_T$ with the original training reward or establish transfer effectiveness by itself. The usefulness of this signal for stronger students is evaluated empirically.

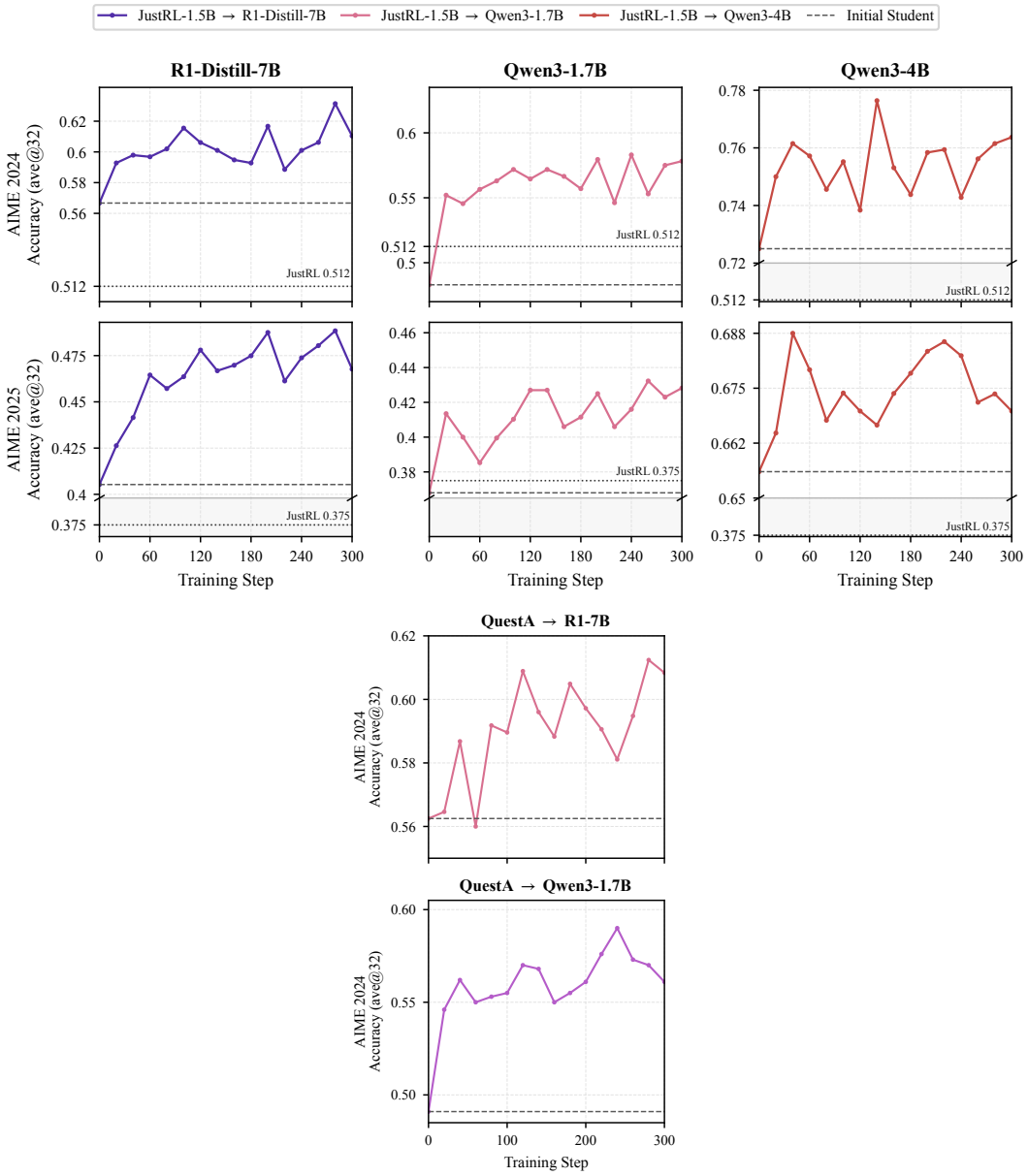


Figure 5: Direct-OPD transfers RL-induced policy shifts across teacher pairs and student families. **Top:** R1-Distill-1.5B → JustRL-1.5B transfer into R1-Distill-7B, Qwen3-1.7B, and Qwen3-4B, evaluated on AIME 2024 and AIME 2025. **Bottom:** Nemotron-1.5B → QuestA-Nemotron-1.5B transfer into R1-Distill-7B 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.

# D Additional Experimental Results

## D.1 Transfer learning curves

Figure 5 provides the full learning curves for the teacher pairs and students in Section 3.1. The corresponding scores are in Table 1; additional QuestA AIME 2025 curves are in Figure 6.

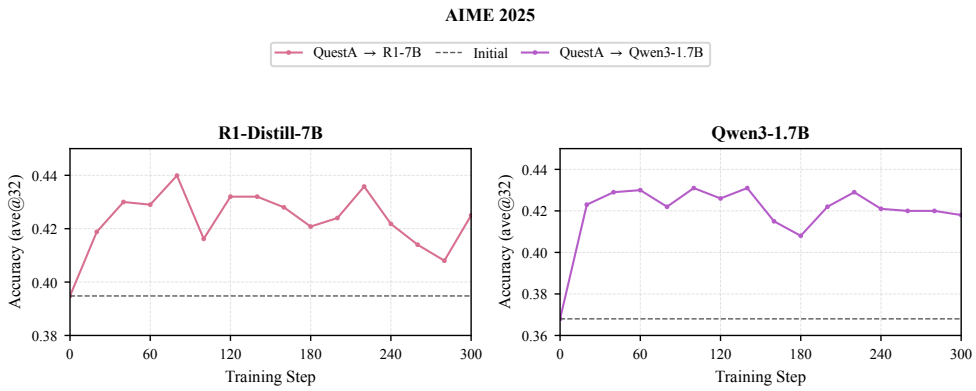


Figure 6: QuestA transfer curves on AIME 2025 for the cross-pattern transfer setting. Figure 5 reports the corresponding AIME 2024 curves.

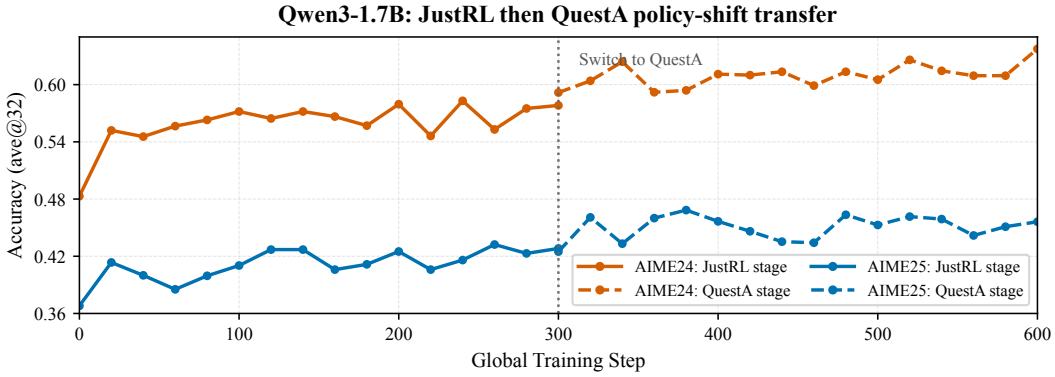


Figure 7: Sequential policy-shift transfer into Qwen3-1.7B on AIME 2024 and AIME 2025. The first stage uses the R1-Distill-1.5B → JustRL-1.5B signal; the second stage continues from that checkpoint with the Nemotron-1.5B → QuestA-Nemotron-1.5B signal. The second stage's local steps 0–300 are plotted as global steps 300–600. The first-stage endpoint and second-stage starting point are evaluated from separate samples, so the small boundary discontinuity reflects evaluation variance. Best scores observed during each stage are reported in Table 1.

### D.2 Sequential composition

Figure 7 provides the two-stage training trajectories for the sequential-transfer experiment in Section 3.3.

## E Additional Analyses

### E.1 Cross-pattern transfer and entropy

We use the top-$k$ set-overlap metric defined in Section 4.1, comparing the student with both the post-RL teacher and its reference on the same student-visited states. Figure 8 extends the JustRL comparison in Figure 3 to the QuestA teacher pair.

Figure 9 checks whether the low-overlap transfers are accompanied by entropy collapse. It reports student, post-RL teacher, and teacher-reference entropy for the JustRL pair; the corresponding QuestA diagnostics are in Figure 10.

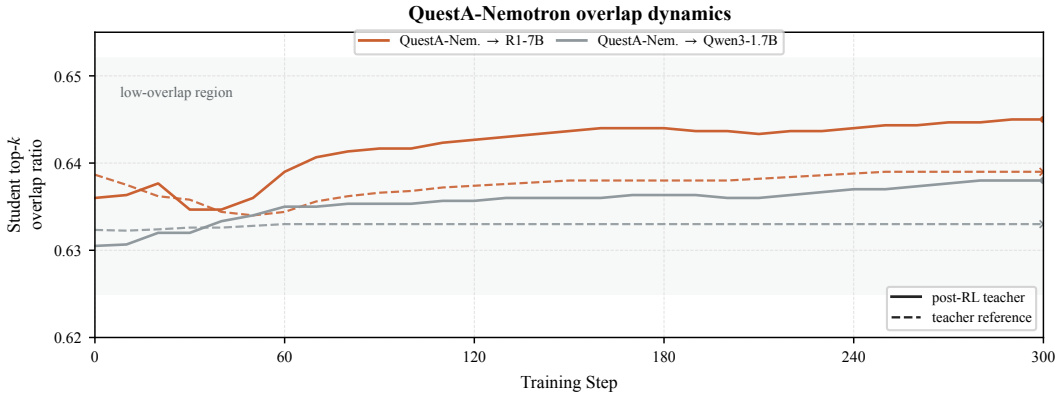


Figure 8: Teacher–student top-$k$ overlap for the Nemotron-1.5B → QuestA-Nemotron-1.5B teacher pair during Direct-OPD transfer into R1-Distill-7B and Qwen3-1.7B. The vertical axis is the fraction of shared tokens between the student's and comparison policy's top-$k$ sets at student-visited prefixes; the horizontal axis counts student training steps. Solid curves compare the student with the post-RL teacher, and dashed curves compare it with the teacher reference. The corresponding JustRL results are shown in Figure 3.

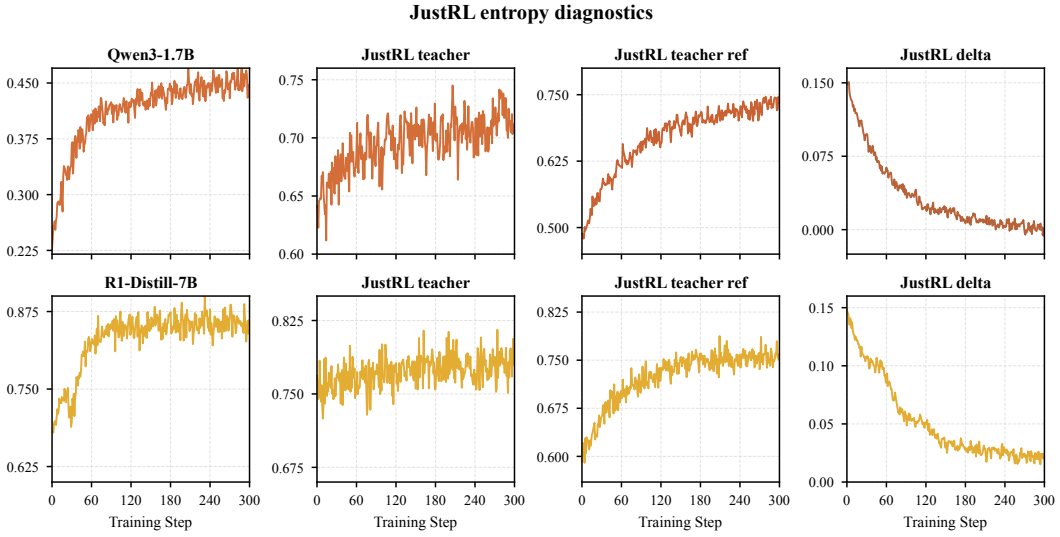


Figure 9: Entropy diagnostics for R1-Distill-1.5B → JustRL-1.5B policy-shift transfer. **Top row:** transfer into Qwen3-1.7B. **Bottom row:** transfer into R1-Distill-7B. Each row shows student entropy, post-RL teacher entropy, teacher-reference entropy, and teacher entropy minus reference entropy. Actor entropy does not collapse, while the teacher/reference entropy gap narrows over training.

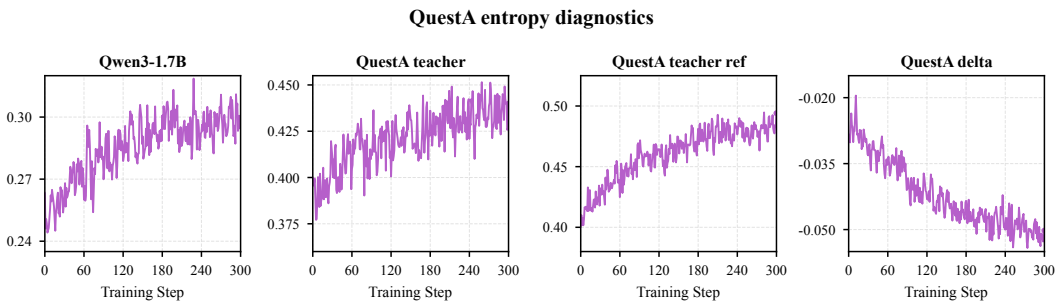


Figure 10: Entropy diagnostics for QuestA-Nemotron into Qwen3-1.7B. The panels show student entropy, post-RL teacher entropy, teacher-reference entropy, and teacher entropy minus reference entropy. Together with Figure 9, this shows that the non-collapse pattern is not specific to the JustRL teacher pair.

### E.2 Response-length sweep and diagnostic setup

Figure 11 compares response lengths for two students with the JustRL teacher pair and a fixed KL coefficient of 1.

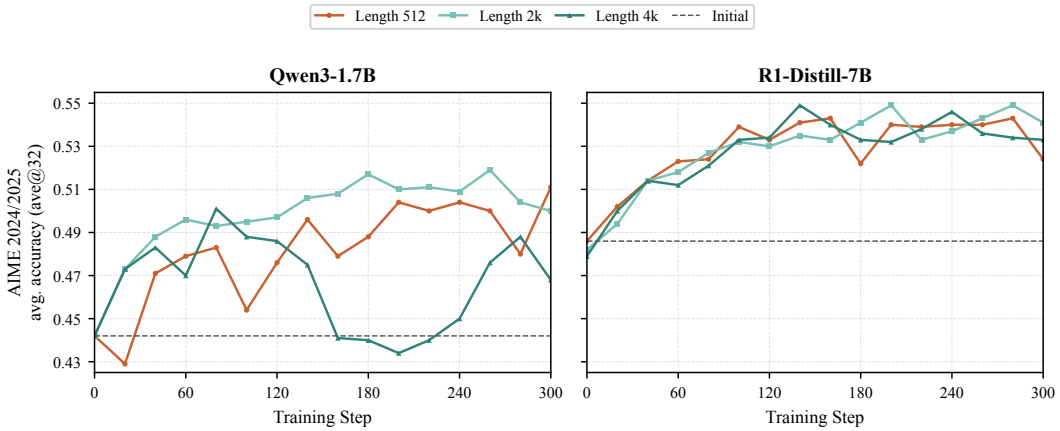


Figure 11: Response-length sweep for R1-Distill-1.5B $\rightarrow$ JustRL-1.5B transfer with fixed KL coefficient 1. We report the average of AIME 2024 and AIME 2025 validation accuracy (ave@32) during training for Qwen3-1.7B and R1-Distill-7B students. The 2k setting gives stable validation behavior across the two students, while shorter or longer rollouts do not consistently improve the validation curves.

For the long-prefix diagnostic in Figure 4, we use 64 fixed long rollouts from the base Qwen3-1.7B actor and evaluate the base actor and checkpoints trained for 40 steps with 2k, 4k, or 6k response lengths and fixed KL coefficient 1. At each fixed prefix, each evaluated actor supplies its own top-16 token set and raw token probabilities for Eq. 17. We compute $G_T$ for each rollout and then average across the 64 rollouts for the middle panel. The right panel reports the average of AIME 2024 and AIME 2025 accuracy at those same checkpoints.

### E.3 KL coefficient sweep

Figure 12 provides the full fixed- and adaptive-KL curves discussed in Section 4.3. All runs use the same 2k response horizon; each column pairs validation accuracy with the dense token reward for one teacher–student combination.

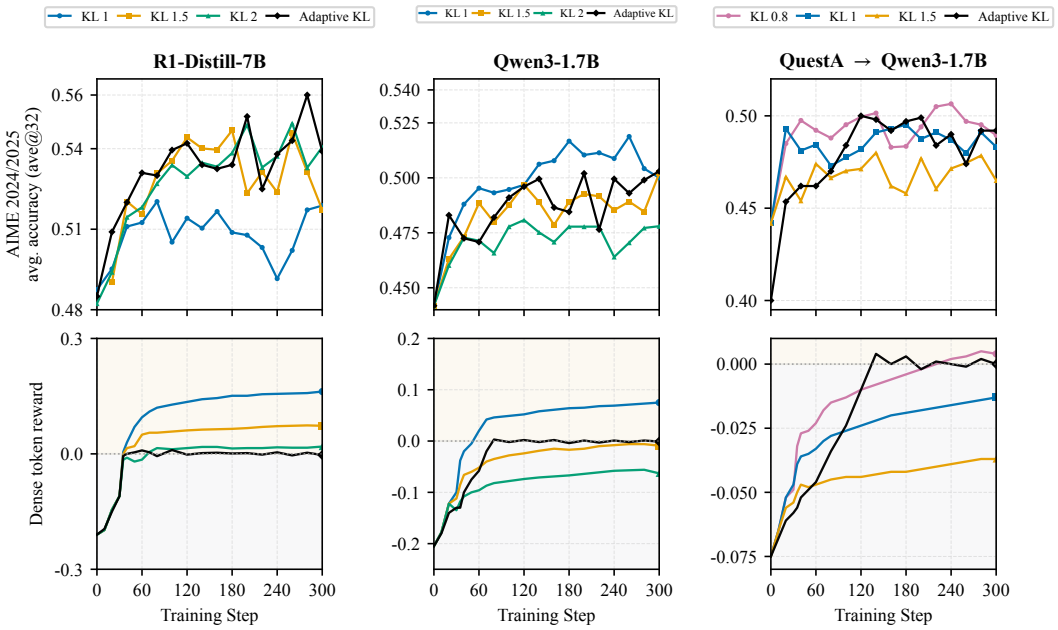


Figure 12: **The best KL coefficient is pair-dependent, and adaptive KL pulls the dense token reward toward a balanced regime.** 2k-response runs across fixed KL coefficients, with adaptive KL as the black curve. The left and middle columns use the R1-Distill-1.5B → JustRL-1.5B teacher pair with R1-Distill-7B and Qwen3-1.7B students, respectively; the right column uses Nemotron-1.5B → QuestA-Nemotron-1.5B with Qwen3-1.7B. Legend values are fixed KL coefficients, and the horizontal axes count student training steps. **Top row:** the average of AIME 2024 and AIME 2025 accuracy (ave@32); **bottom row:** dense token reward. Each score curve keeps its own recorded initial value; missing step-zero evaluations are omitted. 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.

# F Additional Related Work

**Reasoning models and distillation.** Reinforcement learning with verifiable rewards now produces strong reasoning models (DeepSeek-AI et al., 2025; Shao et al., 2024; Schulman et al., 2017; Kimi Team, 2025; Qwen Team, 2025; Hu et al., 2025; Yu et al., 2025; He et al., 2025a; An et al., 2025; Li et al., 2025a; He et al., 2025b;c; 2026), a paradigm now standard in frontier systems (Yang et al., 2025; DeepSeek-AI, 2026; Z.ai, 2026; Moshkov et al., 2025; Guha et al., 2025; Li et al., 2024). Knowledge distillation spans classic and sequence-level methods, modern variants, and scaling behavior (Hinton et al., 2015; Kim & Rush, 2016; Sanh et al., 2019; Jiao et al., 2020; Wang et al., 2020; Mirzadeh et al., 2020; Cho & Hariharan, 2019; Busbridge et al., 2025), with a large body of work on divergences and estimators for distilling LLMs (Gu et al., 2023; Wen et al., 2023; Zhong et al., 2024; Wu et al., 2025; Ko et al., 2024; 2025; Gu et al., 2025; Zhou et al., 2024; Xu et al., 2025c). Recent OPD work studies reasoning-model distillation (Agarwal et al., 2023; Lu & Lab, 2025; Li et al., 2026b; Song & Zheng, 2026; Fu et al., 2026; Jin et al., 2026; Jang et al., 2026; Li et al., 2026a; Ye et al., 2026b;a; Zhao et al., 2026b;a; Hübotter et al., 2026; Shenfeld et al., 2026; Zhang et al., 2026a; Xu et al., 2026; Ko et al., 2026; Kim et al., 2026; Ye et al., 2025; Xiao et al., 2026). Small students can struggle to copy much stronger reasoners (Li et al., 2025b). Related hybrids couple distillation with RL (Xu et al., 2025b;a; Zhang et al., 2026b; 2025; Hübotter et al., 2026; Yang et al., 2026a).

**Weak supervision and scalable oversight.** Weak supervision connects to semi-supervised learning and learning from noisy labels (Grandvalet & Bengio, 2004; Dai & Le, 2015; Laine & Aila, 2016; Athiwaratkun et al., 2018; Han et al., 2018; Berthelot et al., 2019; Li et al., 2020; Chen et al., 2020a;b; 2022). It also connects to scalable oversight when reliable human or stronger-model supervision is scarce (Leike & Sutskever, 2023; Christiano et al., 2018; Leike et al., 2018; Irving et al., 2018; Bowman et al., 2022; Bowman, 2022; Bai et al., 2022), and is related to eliciting latent knowledge and to easy-to-hard generalization (Christiano et al., 2021; Burns et al., 2022; Schwarzschild et al., 2021b;a). Recent work studies its theory and extends it to reasoning and automated alignment (Cao et al., 2024; Yao et al., 2025b;a; Sang et al., 2024; Yuan et al., 2025). Self-rewarding schemes bootstrap a model with its own judgments (Yuan et al., 2024).

**Reward learning and token-level supervision.** DPO variants fit policies directly from preferences without an explicit reward model (Azar et al., 2024; Ethayarajh et al., 2024; Meng et al., 2024). Dense and process rewards for reasoning (Lightman et al., 2023; Wang et al., 2023; Cui et al., 2025) and per-token credit assignment (Kazemnejad et al., 2024; Zhang, 2026) provide additional context for token-level supervision. Reward over-optimization is studied by Gao et al. (2023).