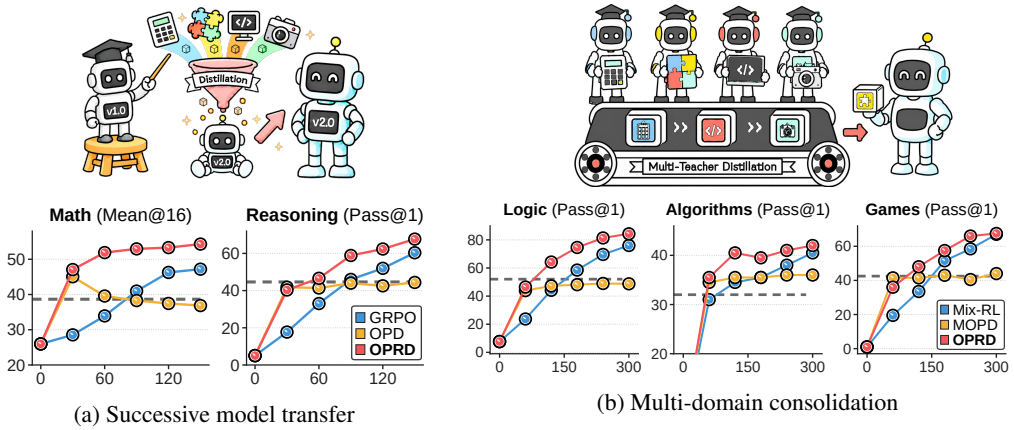
ELICITING WEAK-TO-STRONG GENERALIZATION WITH ON-POLICY REVERSE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Weak-to-strong generalization asks whether stronger models can learn from weaker supervisors and surpass them. This question is particularly important for successive model generations and multi-domain consolidation, where repeating frontier-scale post-training from scratch can be prohibitively expensive. Yet conventional distillation treats the weak teacher as an optimization target, potentially imposing its capacity ceiling on the student. We introduce On-Policy Reverse Distillation (OPRD), which evaluates the teacher's policy shift relative to its reference policy on student rollouts and amplifies the component of the student's verifier-driven policy gradient along that direction. By rescaling only verifier-supported updates, OPRD preserves the stationary points of policy optimization while accelerating learning beyond the teacher. In both successive model transfer and multi-teacher distillation, OPRD achieves higher performance with fewer student updates than existing RL and distillation approaches. Stylistically, OPRD students remain closer to models trained with verifier-based RL alone than to their weak teachers, suggesting that teacher guidance accelerates rather than redirects the students' own optimization. Results in conventional strong-to-weak distillation further demonstrate that OPRD effectively combines verifier-driven policy optimization with teacher guidance regardless of capacity ordering.

1 INTRODUCTION



(a) Successive model transfer

(b) Multi-domain consolidation

Figure 1: **On-policy reverse distillation (OPRD) enables faster and stronger weak-to-strong generalization across two key settings. (a)** For successive model transfer, a checkpoint from a post-trained 4B-scale model serves as the teacher for an 8B-scale student. We average evaluations every 30 training steps: Mean@16 over AIME'24/25, HMMT'25, and OlympiadBench for math, and Pass@1 over Knights & Knaves, Quantum Lock, String Manipulation, and Countdown for reasoning tasks. **(b)** In multi-domain consolidation, four domain-specialized 4B-scale teachers are distilled into one 8B-scale student, with mixed-task batches and the corresponding teacher activated for each example. We report performance every 60 steps for Logic (averaged over Knights & Knaves and Quantum Lock), Algorithms (String Manipulation), and Games (Countdown). Gray dashed lines indicate the corresponding weak teachers' performance.

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.

Knowledge distillation (KD; Hinton et al., 2015) transfers knowledge from a teacher model to a student. For autoregressive language models, conventional distillation on fixed or teacher-generated sequences can mismatch training and inference prefixes. On-policy distillation (OPD) (Gu et al., 2024; Agarwal et al., 2024; Ko et al., 2024) addresses this by training on student-generated responses and querying the teacher at student-visited prefixes. Recent work has applied OPD to efficient reasoning post-training (Qwen Team, 2025b; Xu et al., 2025; GLM-5 Team, 2026) and to consolidating capabilities from multiple domain-specific teachers into a single student (Xiaomi Team, 2026; Yang et al., 2026b; DeepSeek-AI, 2026). Standard OPD optimizes the student toward the teacher policy, making it well suited when matching that policy is the goal.

However, useful supervision need not come from a model that the student should ultimately match. Weaker models can be post-trained at lower cost while still acquiring useful task-specific capabilities, making them practical teachers for stronger students. Weak-to-strong generalization has shown that stronger pretrained models can learn from weaker supervisors and even outperform them across language understanding, reward modeling, and reasoning tasks (Burns et al., 2024; Yang et al., 2024; Lang et al., 2024; Zhou et al., 2025). This regime is especially promising in two key settings in modern foundation model development. (i) *Successive model transfer* (Figure 1a, top): A post-trained model from one generation can supervise a larger-scale successor, enabling it to inherit prior post-training gains and improve beyond its supervisor. (ii) *Multi-domain consolidation* (Figure 1b, top): Domain-specialized policies can be developed independently at smaller scale, enabling efficient iteration on reward functions, environments, and training recipes. Multi-teacher on-policy distillation (MOPD) (Kimi Team, 2026; Ma et al., 2026; Xiaomi Team, 2026) can then consolidate their capabilities into a unified foundation model. Both settings therefore call for *reverse* distillation that transfers post-training gains from weaker models without limiting the eventual performance of higher-capacity students.

Simply applying OPD in the weak-to-strong direction does not resolve this: it matches the weak teacher's full policy at each student-visited prefix, transferring not only post-training gains but also preferences inherited from its reference policy and behavior shaped by its limited capacity. Teacher matching can provide useful guidance when the student underperforms the teacher, but can become counterproductive when the student discovers verifier-supported behavior that the teacher assigns low probability; in this case, the distillation term can directly oppose the student's reward-driven update (Akhondzadeh et al., 2026; Ziheng et al., 2026). Adding reinforcement learning does not remove this tension if teacher matching remains a separate objective, since the matching loss can compete with reward maximization (Xu et al., 2025; Zhang et al., 2026a). Likewise, isolating the teacher's post-training policy change is insufficient if the student is still trained to match it. This change captures only the improvements realized by the weak teacher, not the full range available to the stronger student, so direct matching can impose the same capacity limitation. The central question is therefore **how to leverage weak-model post-training gains to accelerate verifier-driven student optimization without limiting the student to the teacher's capabilities.**

We introduce **On-Policy Reverse Distillation (OPRD)**, which uses the policy change learned during weak-model post-training to accelerate a stronger student's own optimization. On the student's on-policy rollouts, OPRD computes the verifier-driven policy gradient and extracts the weak teacher's policy shift relative to its reference policy. It projects the student gradient onto the direction of this shift and *amplifies the projected component*, leaving the orthogonal component unchanged. Because this transformation positively rescales only a component already present in the student gradient, it preserves the stationary points of policy optimization in logit space while adding a nonnegative first-order alignment gain. When the teacher shift and student gradient align, OPRD reinforces their shared direction and accelerates convergence; when they oppose, it strengthens surprising student behavior supported by the verifier, allowing the student to improve beyond the teacher.

We evaluate OPRD across mathematical reasoning, logical reasoning, and code generation. In successive model transfer, OPRD reaches weak-teacher performance with 33–67% fewer student updates than GRPO (Shao et al., 2024) and achieves up to 22.7 percentage points higher performance at early checkpoints. Unlike OPD, it then moves beyond the teacher rather than saturating after the initial transfer (Figure 1a, bottom). In multi-teacher consolidation, OPRD reaches teacher-level performance with 55% fewer updates than Mix-RL, and the resulting student ultimately outperforms all four specialists (Figure 1b, bottom). With the same number of rollouts per update, these gains reflect improved sample efficiency during student training. The benefit extends to conventional strong-to-weak distillation, where OPRD moves beyond OPD's plateau through verifier-driven optimization.

Contributions. In summary, our key contributions in this paper are as follows.

- **Weak-to-Strong Generalization.** We study how post-training gains from weaker models can be transferred to stronger students in two practical scenarios: successive model transfer and multi-domain consolidation. We identify the central challenge as exploiting these gains without making either the weak policy or its policy shift a separate optimization target.
- **On-Policy Reverse Distillation.** We introduce OPRD, which evaluates a weak teacher's policy shift relative to its reference policy on student rollouts and amplifies the component of the student's verifier-driven policy gradient along that direction. Because OPRD only rescales verifier-supported updates, it accelerates the student's own optimization while preserving its stationary points, allowing the student to move beyond the teacher.
- **Empirical Evaluation and Analysis.** Across successive-model and multi-teacher settings, OPRD reaches the final performance of competing methods substantially earlier and ultimately outperforms both RL and distillation baselines (§3.1, §3.2). We further confirm that these gains extend to conventional strong-to-weak distillation (§3.3). We also compare against recent weak-to-strong methods (§4.1), analyze the design and dynamics of teacher guidance (§4.2), and study student reasoning and response style under teacher guidance (§4.3).

2 METHOD

2.1 PRELIMINARY

Reinforcement Learning with Verifiable Rewards (RLVR). RLVR optimizes a language-model policy using rewards computed by programmatic verifiers, such as exact-answer checks or code execution, and has become central to reasoning post-training (Shao et al., 2024; Guo et al., 2025). For $x \sim \mathcal{D}$, the student samples $y \sim \pi_\theta(\cdot \mid x)$ and visits prefixes $s_t = (x, y_{<t})$. Let A_t denote the advantage assigned to token t and $\mathbf{z}_t$ the corresponding next-token logits at prefix s_t. The token-level policy gradient is

$$\mathbf{g}_t := A_t \nabla_{\mathbf{z}_t} \log \pi_\theta(y_t \mid s_t). \tag{2.1}$$

OPRD later rescales this gradient while preserving the RLVR objective, so the student's attainable performance is determined by the verifier objective and its own policy class rather than being bounded by the teacher's capacity.

On-Policy Distillation (OPD). OPD reduces the training–inference distribution mismatch by sampling responses from the student and querying the teacher at each visited prefix, thereby providing dense token-level supervision over the student's inference-time state distribution. A common reverse-KL formulation is

$$\mathcal{L}_{\mathrm{OPD}}(\theta) := \mathop{\mathbb{E}}_{\substack{x\sim\mathcal{D} \\ y\sim\pi_\theta(\cdot|x)}} \left[\sum_t D_{\mathrm{KL}} \left(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_T(\cdot \mid s_t) \right) \right]. \tag{2.2}$$

Recent methods combine teacher matching with reinforcement learning to pair dense teacher supervision with outcome-based optimization (Xu et al., 2025; Ramos et al., 2026). Even in these hybrid methods, however, teacher matching remains a separate objective, leaving the teacher policy as a direct optimization target.

Weak-to-Strong Generalization. Weak-to-strong generalization studies whether a more capable model can learn from weaker supervisors, such as smaller models or imperfect human feedback, and ultimately outperform them (Burns et al., 2024). We review prior approaches in Appendix A. OPD provides the full next-token distribution $\bar{\pi}_T(\cdot \mid s_t)$ at each student-visited prefix, where $\bar{\pi}_T$ denotes either the weak teacher or a target policy derived from it. Under realizability, the resulting KL objective has the pointwise minimizer

$$\arg\min_{\pi(\cdot|s_t)} D_{\mathrm{KL}}(\pi(\cdot \mid s_t) \,\|\, \bar{\pi}_T(\cdot \mid s_t)) = \bar{\pi}_T(\cdot \mid s_t). \tag{2.3}$$

Alternative teacher-derived targets only change which policy the student matches, while adding reinforcement learning yields a compromise between teacher matching and reward maximization (Xu et al., 2025; Ramos et al., 2026). In both cases, the student remains directly optimized toward a policy defined by the weak teacher. OPRD instead extracts the policy change learned during weak-model post-training and uses it only to rescale the stronger student's own policy gradient.

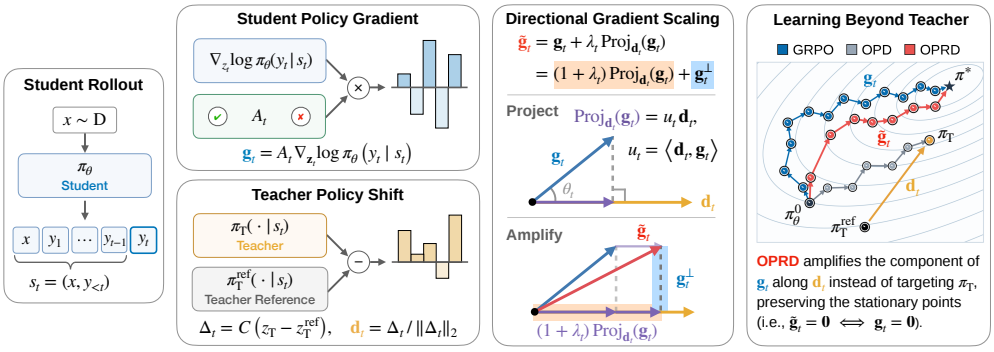


Figure 2: **Conceptual overview of On-Policy Reverse Distillation (OPRD).** Here, $\mathbf{z}_T$ denotes the logits of the weak teacher after post-training and $\mathbf{z}_T^{\mathrm{ref}}$ those of its reference policy, and $\mathcal{C}$ denotes mean-centering. Their centered difference $\boldsymbol{\Delta}_t$ is the teacher's policy shift at that student-visited prefix, and OPRD keeps only its unit direction $\mathbf{d}_t$. In practice, top-10 truncation under the student policy focuses the correction on its high-probability vocabulary region. The rightmost panel shows the student following the verifier-driven policy gradient $\mathbf{g}_t$ with its component along $\mathbf{d}_t$ amplified by $1 + \lambda_t$ and its orthogonal component left unchanged.

2.2 On-Policy Reverse Distillation

Overview. OPRD transfers the policy change learned during teacher post-training rather than matching the teacher's final policy. At each student-visited prefix, it extracts the local direction of this change relative to the teacher's reference policy and uses its alignment with the verifier-driven student gradient to rescale only the gradient component along that direction. Because the teacher signal only rescales the student's own gradient, it can accelerate verifier-supported optimization without defining an independent optimization target. Positive-alignment scaling is active from the outset to amplify updates supported by both the verifier and the teacher, whereas negative-alignment scaling is gradually increased to reinforce verifier-supported departures beyond the weak teacher.

Teacher Policy Shift. The teacher's final policy reflects the change acquired during RL post-training, preferences inherited from its reference policy, and behavior constrained by the weak model's limited capacity. Directly matching it would therefore make all of these part of the student's distillation target. Let π_T denote the frozen RL-trained teacher and π_T^{ref} its frozen pre-RL reference policy, and let $\mathbf{z}_T(s_t)$ and $\mathbf{z}_T^{\mathrm{ref}}(s_t)$ denote their next-token logit vectors at a student-visited prefix s_t. To extract the RL-induced policy delta, we mean-center the difference between the teacher and reference logits, removing a common offset that does not affect relative token preferences. With $\mathcal{C}(\mathbf{v}) := \mathbf{v} - \frac{1}{|\mathcal{V}|}(\mathbf{1}^\top \mathbf{v})\mathbf{1}$, we define

$$\boldsymbol{\Delta}_t := \mathcal{C}\big(\mathbf{z}_T(s_t) - \mathbf{z}_T^{\mathrm{ref}}(s_t)\big) = \mathcal{C}\big(\log \pi_T(\cdot \mid s_t) - \log \pi_T^{\mathrm{ref}}(\cdot \mid s_t)\big)\,. \tag{2.4}$$

Intuitively, $\boldsymbol{\Delta}_t$ captures the change in the teacher's relative next-token preferences induced by post-training, and the corresponding uncentered log-policy ratio admits an implicit-reward interpretation under KL-regularized policy optimization. However, because this shift is learned within the weak teacher's policy class, it need not improve verifier reward for the stronger student. OPRD therefore uses only its unit direction $\mathbf{d}_t := \boldsymbol{\Delta}_t / \|\boldsymbol{\Delta}_t\|_2$ for gradient scaling rather than optimizing toward the shift itself. The student gradient determines whether the resulting correction follows or opposes this direction, independently of its raw magnitude.

Gradient Scaling along the Teacher Direction. At token t, OPRD decomposes the student's policy gradient $\mathbf{g}_t$ (Eq. 2.1) relative to the teacher direction $\mathbf{d}_t$. Let $u_t := \mathbf{d}_t^\top \mathbf{g}_t$ denote their alignment coefficient, and define the projected and orthogonal components as $\mathrm{Proj}_{\mathbf{d}_t}(\mathbf{g}_t) := u_t \mathbf{d}_t$ and $\mathbf{g}_t^\perp := \mathbf{g}_t - \mathrm{Proj}_{\mathbf{d}_t}(\mathbf{g}_t)$, respectively. At optimization step k, OPRD uses a nonnegative scale λ_t:

$$\begin{aligned} \tilde{\mathbf{g}}_t &:= \mathbf{g}_t + \lambda_t \,\mathrm{Proj}_{\mathbf{d}_t}(\mathbf{g}_t) \\ &= \underbrace{(1+\lambda_t)\,\mathrm{Proj}_{\mathbf{d}_t}(\mathbf{g}_t)}_{\textit{amplified}} + \underbrace{\mathbf{g}_t^\perp}_{\textit{unchanged}}\,. \end{aligned} \tag{2.5}$$

OPRD splits the student's policy gradient into components along and orthogonal to the teacher-informed direction, amplifying the former by $1 + \lambda_t$ and leaving the latter unchanged. We backpropagate $\widetilde{\mathbf{g}}_t$ instead of $\mathbf{g}_t$ to update the student.

Learning Beyond the Weak Teacher. Direct teacher matching targets the weak teacher's policy, even when departing from it would yield higher verifier reward. OPRD instead uses the weak teacher only to rescale the student's own policy gradient. At token t, this scaling is the linear map $\widetilde{\mathbf{g}}_t = (\mathbf{I} + \lambda_t \mathbf{d}_t \mathbf{d}_t^\top)\mathbf{g}_t$. For $\lambda_t \geq 0$, the map is invertible and satisfies:

$$\text{(Stationarity)} \qquad \widetilde{\mathbf{g}}_t = \mathbf{0} \quad \text{if and only if} \quad \mathbf{g}_t = \mathbf{0}. \tag{2.6}$$

$$\text{(Alignment Gain)} \qquad \langle \mathbf{g}_t, \widetilde{\mathbf{g}}_t \rangle = \|\mathbf{g}_t\|_2^2 + \lambda_t u_t^2 \geq \|\mathbf{g}_t\|_2^2. \tag{2.7}$$

Since Eq. 2.6 holds at every token, the scaling preserves the stationary points of the verifier objective for a fixed response. Eq. 2.7 shows that $\widetilde{\mathbf{g}}_t$ retains the first-order progress of $\mathbf{g}_t$ and adds a nonnegative gain $\lambda_t u_t^2$ that grows with $|u_t|$. For a fixed response, these gains sum to a nonnegative term in the guaranteed one-step ascent (see Appendix B). Thus OPRD can accelerate student optimization without a teacher-defined target.

Asymmetric Alignment Scaling. The sign of u_t indicates whether $\mathbf{g}_t$ aligns with or opposes $\mathbf{d}_t$, so scaling reinforces teacher-following updates when $u_t \geq 0$ and verifier-supported departures when $u_t < 0$. Both signals may contain reward-irrelevant bias (e.g., $\mathbf{g}_t = \mathbf{g}_t^\star + \boldsymbol{\epsilon}_t$, with $\mathbf{g}_t^\star$ denoting the reward-improving signal and $\boldsymbol{\epsilon}_t$ aggregating structured bias components). Scaling only one sign can then systematically amplify this bias over training (see Appendix C.1 and C.2). Because negative-alignment updates are less reliable on initially weak student rollouts, we activate the positive branch immediately and gradually ramp up the negative branch. At optimization step k, we set the token-wise scaling coefficient as

$$\lambda_t := \begin{cases} \lambda, & u_t \geq 0, \\ \lambda \min\left\{ \frac{k}{K_{\text{warm}}}, 1 \right\}, & u_t < 0, \end{cases} \tag{2.8}$$

where λ is the scaling strength and K_{warm} is the warm-up horizon. Because $\lambda_t \geq 0$, Eqs. 2.6 and 2.7 remain valid. Warming up the negative branch limits early one-sided amplification while preserving immediate teacher-aligned transfer and gradually strengthening verifier-supported departures. We compare alternative branch-scaling strategies in Appendix C.3.

3 Experiments

We compare OPRD with verifier-only (Shao et al., 2024), distillation-only (Agarwal et al., 2024; Ma et al., 2026), and hybrid (Xu et al., 2025) baselines under matched student initialization and training budgets. Teachers are task-specific GRPO checkpoints held fixed during student training. Our experiments cover mathematics, Reasoning Gym, and code generation, with datasets and settings detailed in Appendices D and E.

3.1 Weak-to-Strong Distillation for a Successor Model

Settings. We pair Qwen3-4B teachers with Qwen3-8B students for mathematics and their base counterparts for Reasoning Gym. For code generation, we pair a Qwen3-8B-Base teacher with a Qwen3-14B-Base student. Further results for all three task families appear in Appendix F.2.

OPRD Accelerates Learning While Continuing Beyond the Weak Teacher. Figure 1a (bottom) shows that OPD improves rapidly but plateaus near the teacher average, whereas GRPO progresses more gradually. OPRD matches OPD's initial acceleration, quickly surpasses the weak teacher, and reaches GRPO's end-of-training performance substantially earlier. Averaged over five evenly spaced checkpoints to summarize the learning curve, Table 1 shows gains of 7.92 points on mathematics and 10.80 points on Reasoning Gym over the strongest baseline. The initially similar trajectories of OPD and OPRD indicate that teacher guidance is useful while the student still trails it, but their later divergence suggests that direct policy matching becomes restrictive once the student discovers reward-supported improvements beyond the teacher.

Table 1: **Main experimental results for successive model transfer in mathematics and Reasoning Gym.** We report Mean@16 for mathematics and Pass@1 for Reasoning Gym. Teacher and initial-student rows use fixed checkpoints, and trained rows average evaluations at steps 30, 60, 90, 120, and 150. Scores are percentages, and Avg. is the mean across benchmarks. Best results per column are bold. Appendix F.1 gives learning curves, F.2 covers mathematics with Qwen3-Base, three additional Reasoning Gym tasks, and code generation, and F.3 reports standard deviations.

	Math Reasoning					Reasoning Gym				
Policy	AIME'24	AIME'25	HMMT'25	Olympiad	**Avg.**	Knights	Quantum	String	Count	**Avg.**
	Qwen3-4B (Teacher) → Qwen3-8B (Student)					**Qwen3-4B-Base (Teacher) → Qwen3-8B-Base (Student)**				
Teacher	42.50	38.75	21.25	52.15	38.66	57.50	46.58	32.00	42.50	44.65
Student	25.63	19.58	12.50	46.22	25.98	11.00	3.14	3.00	3.00	5.04
+ GRPO	46.63	36.42	21.96	52.52	39.38	54.20	36.24	35.50	41.30	41.81
+ OPD	46.92	38.21	21.42	51.24	39.44	55.00	39.62	35.10	41.60	42.83
+ KDRL	53.46	43.13	26.08	**53.28**	43.99	53.30	36.63	38.10	49.50	44.38
+ OPRD	**66.92**	**56.04**	**31.42**	53.26	**51.91**	**73.30**	**51.11**	**42.20**	**54.10**	**55.18**

Table 2: **(Left) Multi-teacher distillation results on Reasoning Gym.** Four task-specific Qwen3-4B-Base specialists supervise one Qwen3-8B-Base student. Teacher and initial-student rows report fixed-checkpoint scores, while trained rows average Pass@1 at updates 60, 120, 180, 240, and 300. **(Right) Strong-to-weak distillation results.** We evaluate Qwen3-8B → Qwen3-1.7B on AIME'24 and Qwen3-8B-Base → Qwen3-0.6B on Knights & Knaves. Teacher and initial-student rows report fixed-checkpoint scores, while trained rows average Mean@16 and Pass@1, respectively, over five checkpoints. Detailed learning curves appear in Appendices G and H. Best scores are bold.

Policy	**Knights**	**Quantum**	**String**	**Count**	**Avg.**
4 Teachers (Qwen3-4B-Base) → 1 Student (Qwen3-8B-Base)					
Teachers	57.50	46.58	32.00	42.50	44.65
Student	11.00	3.14	3.00	3.00	5.04
+ Mix-RL	64.90	43.91	35.90	46.00	47.68
+ MOPD	57.10	37.70	35.50	42.10	43.10
+ KDRL	65.90	44.09	35.00	44.90	47.47
+ OPRD	**80.70**	**59.68**	**39.70**	**55.00**	**58.77**

Policy	**AIME'24**	**Knights Knaves**	**Avg.**
	8B → 1.7B	**8B-Base → 0.6B**	
Teacher	52.29	64.50	58.40
Student	10.00	5.00	7.50
+ GRPO	18.92	20.60	19.76
+ OPD	29.79	20.20	25.00
+ KDRL	25.33	33.70	29.52
+ OPRD	**33.58**	**49.40**	**41.49**

3.2 Multi-Teacher Weak-to-Strong Distillation

Settings. We consolidate four Qwen3-4B-Base specialists into one Qwen3-8B-Base student. Each batch mixes the four Reasoning Gym tasks equally and pairs each example with its corresponding teacher. We train for 300 policy updates because mixing reduces exposure to each task, retaining the single-teacher coefficient schedules.

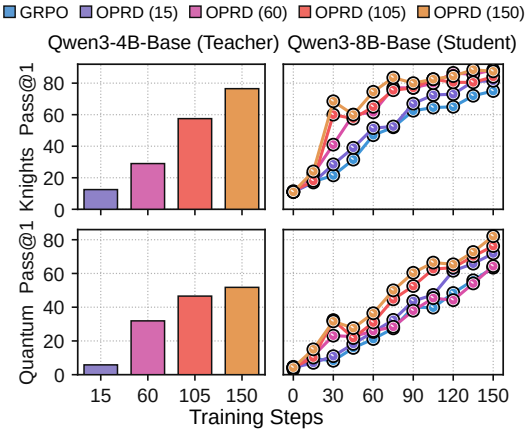
OPRD Consolidates Heterogeneous Specialists without Cross-Task Tradeoffs. Figure 1b (bottom) shows MOPD quickly approaches the specialist average before plateauing, while Mix-RL improves more gradually. OPRD combines early transfer with sustained improvement. Table 2 reports 58.77, 11.09 points above Mix-RL and 14.12 above the specialist average. OPRD surpasses the corresponding specialist on all four tasks despite their distinct structures and objectives, indicating joint gains without cross-task tradeoffs. OPRD may reduce cross-task interference by amplifying only the component of each task's verifier-driven student gradient along its teacher-shift direction, rather than matching the full specialist policy. This projection resembles PCGrad (Yu et al., 2020), but uses each task's teacher direction rather than another task gradient.

3.3 Strong-to-Weak Distillation

Settings. We pair a Qwen3-8B teacher with a Qwen3-1.7B student for mathematics, and a Qwen3-8B-Base teacher with a Qwen3-0.6B student for Knights & Knaves. Both teachers are step-105 task-specific GRPO checkpoints. Appendix H gives batch sizes and coefficient schedules.

Table 3: **(Left) Weak-to-strong baseline results.** For 4B-to-8B transfer, we use instruction-tuned models for math and base models for reasoning. Teacher and initial-student rows show fixed-checkpoint scores, while trained rows average checkpoint evaluations. Best scores per column are bold. Implementation details and learning curves appear in Appendix I.1 and Appendix I.2, respectively. **(Right) Teacher-checkpoint quality ablation.** Bars show Qwen3-4B-Base teacher performance at GRPO steps 15, 60, 105, and 150. Lines show learning curves for Qwen3-8B-Base OPRD students trained with those checkpoints. Other settings follow Appendix D.

Policy	AIME'24	Knights	String	Avg.
4B (-Base) → 8B (-Base)				
Teacher	42.50	57.50	32.00	44.00
Student	25.63	11.00	3.00	13.21
+ GRPO	46.63	54.20	35.50	45.44
+ OPD	46.92	55.00	35.10	45.67
+ W2SR-P	45.42	62.50	38.00	48.64
+ S2L-PO	60.54	63.40	38.70	54.21
+ OPSD	15.88	24.60	24.60	21.69
+ Direct-OPD	35.04	49.20	29.20	37.81
+ W2S-OPD	52.96	59.50	35.20	49.22
+ OPRD	**66.92**	**73.30**	**42.20**	**60.81**



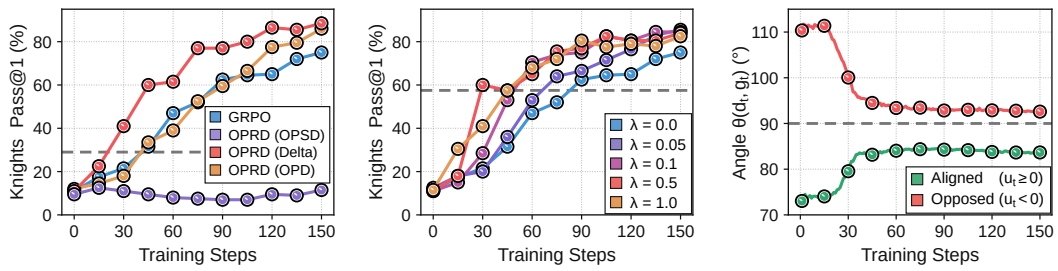
OPRD Does Not Depend on Teacher–Student Capacity Ordering. Table 2 shows OPRD outperforms OPD in strong-to-weak distillation by 3.79 points on AIME'24 and 29.20 on Knights & Knaves. Prior studies show that capacity or distributional gaps can make OPD fail, so a stronger teacher need not yield a better student (Li et al., 2026; Fu et al., 2026). OPD improves initially in both settings but quickly saturates well below OPRD. KDRL's lower score suggests that adding verifier feedback to policy matching does not fully resolve this issue. OPRD instead amplifies only the component of the student's verifier-driven gradient aligned with the teacher's policy delta. This allows the student to benefit from teacher guidance along reward-supported directions it can realize, without having to reproduce the stronger policy in full.

4 Analysis

4.1 Broader Comparison with Weak-to-Strong Methods

OPRD Outperforms Methods Using Off-Policy Generations from Weak Teacher. Table 3 (left) compares OPRD with three baselines built on weak-teacher generations. W2SR-P (Yuan et al., 2026) performs SFT on verified-correct teacher trajectories. S2L-PO (Ren et al., 2026) mixes off-policy weak-explorer rollouts with student rollouts in shared GRPO groups, then transitions to fully on-policy RLVR. Our OPSD variant (Zhao et al., 2026) uses a verified weak-teacher draft as privileged context for self-distillation. W2SR-P and S2L-PO improve over the initial student, showing that weak-teacher trajectories can bootstrap learning within the same Qwen3 family. However, reliance on off-policy teacher trajectories can create train–inference mismatch (Agarwal et al., 2024) and need not transfer underlying capabilities across model gaps (Gudibande et al., 2023). OPRD instead remains fully on-policy and uses the teacher shift only to rescale the aligned component of the verifier gradient. OPRD reaches 60.81, 6.60 points above the strongest alternative, S2L-PO, and leads on all three tasks.

Rescaling the Verifier Gradient Outperforms Direct Optimization of the Weak Policy Delta. The lower rows of Table 3 (left) compare OPRD with Direct-OPD (Feng et al., 2026) and W2S-OPD (Yu et al., 2026), two concurrent methods that derive their objectives directly from the weak policy shift. Direct-OPD uses the corresponding log-ratio as a dense reward, whereas W2S-OPD reanchors the shift to the student's base policy and distills the resulting proxy teacher. Because both methods are sensitive to the transferred shift's relative strength, we use their reported hyperparameters. Both still rely solely on the information encoded in the shift. OPRD instead retains verifier supervision through the orthogonal component $\mathbf{g}_t^{\perp}$, allowing the student to pursue reward-supported directions not captured by the weak policy delta. OPRD reaches 60.81, outperforming W2S-OPD by 11.59 points and Direct-OPD by 23.00 points on average.



(a) Construction of $\mathbf{d}_t$ (b) Amplification strength λ (c) Gradient alignment

Figure 3: **(a) Ablations of scaling-direction construction.** All OPRD variants use the step-60 GRPO teacher. For OPSD, a verified teacher draft is provided as privileged context. **(b) Ablation of directional amplification strength.** We vary OPRD's directional-correction coefficient λ. $\lambda = 0$ corresponds to GRPO. The gray dashed line marks teacher performance. Other settings follow Appendix D. **(c) Alignment dynamics between $\mathbf{d}_t$ and $\mathbf{g}_t$.** On Knights & Knaves, we track their angle θ_t during OPRD with a Qwen3-8B-Base student and Qwen3-4B-Base weak teacher. Excluding rollout groups with $\mathbf{g}_t = \mathbf{0}$ (identical rewards within the group), we report token-averaged angles for aligned ($u_t \geq 0$) and opposed ($u_t < 0$) tokens.

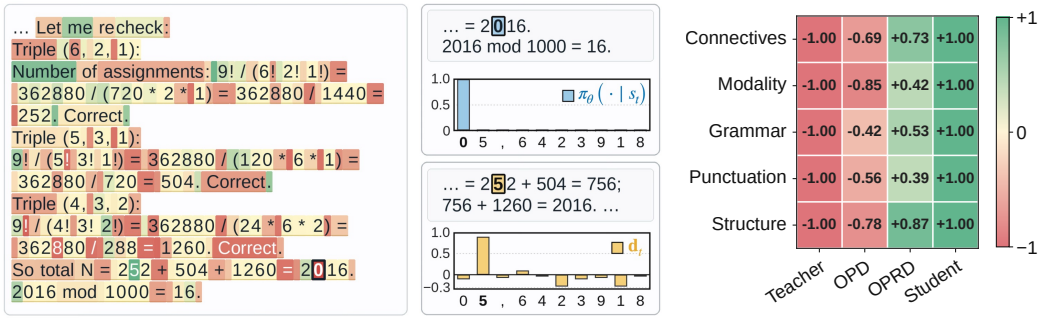
4.2 Design and Dynamics of Teacher Guidance

Better-Trained Weak Teachers Provide More Effective Guidance. Table 3 (right) shows that later, better-performing GRPO checkpoints of the 4B teacher generally accelerate 8B OPRD students on both reasoning tasks. Because $\mathbf{\Delta}_t$ is normalized before scaling, this benefit cannot reflect shift magnitude alone. Later checkpoints appear to encode a more reward-informative direction, giving OPRD a larger verifier-gradient component to amplify. Notably, the step-60 teacher achieves only 29.0% Pass@1 on Knights & Knaves, yet the corresponding OPRD student rapidly reaches approximately 88%, far surpassing both the teacher and GRPO. Thus, while teacher quality affects the strength of OPRD's acceleration, the teacher's absolute performance need not impose a ceiling on the student.

Weak Policy Delta Provides the Most Effective Scaling Direction. Figure 3a compares three choices for the guidance direction $\mathbf{d}_t$: the normalized weak policy delta $\mathbf{\Delta}_t$, the OPD teacher-matching gradient, and the OPSD self-distillation gradient. The weak policy delta yields the fastest and most sustained gains. Comparing the post-trained teacher with its reference isolates the reward-relevant update, and their log-policy ratio admits an implicit-reward interpretation. OPRD projects the verifier gradient onto this direction and amplifies its aligned component, exploiting the teacher's reward information without inheriting its capacity ceiling. In contrast, OPD captures the full teacher–student mismatch and offers limited acceleration when the teacher is too weak, while OPSD's off-policy supervision can restrict exploration of alternative reasoning paths (Kim et al., 2026; Kaur et al., 2026). Although OPD improves with a better-trained teacher, the weak policy delta remains the most reliable guidance signal (see Appendix J).

Sufficient Directional Amplification Enables Early Acceleration. Figure 3b varies λ, which scales OPRD's directional correction and controls teacher guidance. Every $\lambda > 0$ improves final Pass@1 over $\lambda = 0$ (GRPO). Larger values of λ up to 0.5 also yield faster gains early in training. This relationship between guidance strength and learning speed confirms that directional correction drives OPRD's acceleration. Performance changes little beyond $\lambda = 0.5$, so precise tuning is unnecessary once amplification is sufficiently strong. We therefore use $\lambda = 0.5$ as the default.

Teacher Guidance Bootstraps Early Learning but Becomes Less Influential over Time. Figure 3c tracks the mean angle θ_t between the guidance direction $\mathbf{d}_t$ and policy gradient $\mathbf{g}_t$. Early in training, the two directions exhibit substantial alignment for $u_t > 0$ and opposition for $u_t < 0$. This strong directional coupling allows the weak policy delta to bootstrap student learning. As training proceeds, the mean angle for $u_t > 0$ increases toward 90°, while that for $u_t < 0$ decreases toward 90°. Since $\|\mathrm{Proj}_{\mathbf{d}_t}(\mathbf{g}_t)\|_2 / \|\mathbf{g}_t\|_2 = |\cos\theta_t|$, this convergence toward orthogonality means that the component of $\mathbf{g}_t$ along $\mathbf{d}_t$ becomes smaller relative to the full policy gradient. This indicates that the evolving student gradient increasingly follows verifier-supported directions not captured by the teacher shift, so teacher guidance becomes less influential over time. Practical challenges and mitigations are discussed in Appendix K.



(a) Token alignment and reasoning paths

(b) Response style similarity

Figure 4: **(a) Visualizing token alignment and reasoning continuations.** An AIME'25 response from the OPRD student at update 150. Green and red mark positive and negative cosine similarity between $\mathbf{d}_t$ and the student policy gradient $\mathbf{g}_t$ when $A_t = 1$. The black-outlined 0 has the lowest similarity among displayed tokens. The student's top-1 token 0 completes 2016 directly. Forcing $\mathbf{d}_t$'s top-1 token 5 leads the same student to 2016 through an intermediate sum. The plots show the student's top-10 token probabilities above their $\mathbf{d}_t$ values. **(b) Measuring similarity to teacher and student response styles.** On AIME'24, we compare 101 standardized style features across five categories. Normalized distance differences indicate whether each method's average style is closer to the weak teacher (red) or the GRPO-trained student at update 150 (green).

4.3 Student Behavior under Teacher Guidance

OPRD Can Move Beyond the Teacher's Reasoning Paths. Figure 4a illustrates how OPRD can exploit an informative teacher shift while allowing the stronger student to follow its own, more direct reasoning path rather than the one favored by the teacher. At the selected AIME'25 prefix, the student's top-1 prediction is 0, which immediately completes 2016. By contrast, the top-1 token under the weak policy shift $\mathbf{d}_t$ is 5. Forcing 5 and continuing with the same student produces 252 + 504 = 756, followed by 756 + 1260 = 2016. This detour also reaches the correct result, showing that the teacher shift provides a valid direction that may be useful earlier in training. Here, however, the student can already complete the calculation directly. This is reflected in the highlighted 0, which has the most negative alignment with $\mathbf{d}_t$ among the displayed tokens. OPRD therefore raises the logit of 0 and lowers that of 5 (when $u_t < 0$, OPRD amplifies the component of the verifier-driven policy gradient that opposes the teacher shift). The negative-alignment branch thus favors the student's shorter solution over the valid teacher-favored detour, providing a token-level example of how OPRD can move beyond the teacher.

OPRD Remains Stylistically Closer to the Stronger Student. Figure 4b examines how teacher guidance affects response style on AIME'24. We summarize each method's average response style using 101 standardized features grouped into five categories: connectives, modality, grammar, punctuation, and sentence and paragraph structure. For each category, a normalized distance difference indicates whether the average style is closer to the teacher (negative) or the GRPO student at update 150 (positive) (see Appendix L.2 for details). At update 150, OPD is closer to the teacher in all five categories, whereas OPRD is closer to the GRPO student. This pattern suggests that the OPRD student can benefit from what the teacher learned without inheriting its response style, consistent with using the teacher shift to rescale the student's own policy gradient rather than matching the teacher policy.

5 Conclusion

OPRD amplifies the component of a stronger student's policy gradient aligned with a weak teacher's post-training shift. This preserves the policy objective's stationary points in logit space without targeting the teacher policy. Across successive model transfer and multi-domain consolidation, OPRD reaches teacher-level performance in substantially fewer updates than policy optimization alone and improves beyond the teacher where on-policy distillation plateaus. Gains also extend to strong-to-weak distillation. The OPRD student's response style remains closer to the reward-only baseline than to the teacher, consistent with optimizing its own policy. We discuss practical limitations and mitigations in Appendix K and future work in Appendix N.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity and readability of the manuscript and to assist with code review and verification. We have reviewed all AI-assisted work, including the suggested revisions and code feedback. We take responsibility for the final content of this work, including text and code produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation. *arXiv preprint arXiv:2607.04037*, 2026.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=ghNRg2mEgN.

Moses Charikar, Chirag Pabbaraju, and Kirankumar Shiragur. Quantifying the gain in weak-to-strong generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=MyVyH5Jo1l.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *arXiv preprint arXiv:2605.00674*, 2026.

Yijun Dong, Yicheng Li, Yunai Li, Jason D. Lee, and Qi Lei. Discrepancies are virtue: Weak-to-strong generalization through lens of intrinsic dimension. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 14079–14113. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/dong25g.html.

Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity, 2019. URL https://openreview.net/forum?id=r1gl7hC5Km.

Shiyuan Feng, Huan-ang Gao, Haohan Chi, Hanlin Wu, Zhilong Zhang, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Weak-to-strong generalization via direct on-policy distillation. *arXiv preprint arXiv:2607.05394*, 2026.

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

Gemini Team. Introducing Gemini 3.8 Flash and 3.8 Flash Cyber, September 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/3-8-flash-and-3-8-flash-cyber/.

GLM-5 Team. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*, 2021.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

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

Yu-Guan Hsieh, James Thornton, Eugene Ndiaye, Michal Klein, Marco Cuturi, and Pierre Ablin. Careful with that scalpel: Improving gradient surgery with an EMA. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19085–19100. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hsieh24a.html.

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

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. *arXiv preprint arXiv:2605.06387*, 2026.

Can Jin, Tristan J. Li, Rui Wu, Eddy Z. Zhang, and Dimitris N. Metaxas. Weak critics make strong learners: On-policy critique distillation for scalable oversight. In *3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents*, 2026. URL https://openreview.net/forum?id=oEfedgUChS.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026.

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 Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Kimi Team. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=lsHZNNoC7r.

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.

Hunter Lang, David Sontag, and Aravindan Vijayaraghavan. Theoretical analysis of weak-to-strong generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=HOSh0SKklE.

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

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. Tuning language models by proxy. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=dribhnhm1i.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

MAA. American invitational mathematics examination (AIME), 2024–2025, 2024–2025. URL https://maa.org/maa-invitational-competitions/.

Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: augmenting random search with surrogate gradients. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4264–4273. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/maheswaranathan19a.html.

Marko Medvedev, Kaifeng Lyu, Dingli Yu, Sanjeev Arora, Zhiyuan Li, and Nathan Srebro. Weak-to-strong generalization even in random feature networks, provably. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43519–43556. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/medvedev25a.html.

OpenAI. GPT-6 Astra: A new generation of intelligence, 2026. URL https://openai.com/index/gpt-6-astra/.

Qwen Team. Qwen2.5 technical report, 2025a. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025b.

Miguel Moura Ramos, Duarte M Alves, and André FT Martins. A recipe for long-context reasoning in large language models via on-policy optimization and distillation. *arXiv preprint arXiv:2605.12227*, 2026.

Yiming Ren, Yiran Xu, Zicheng Lin, Chufan Shi, Yukang Chen, Dingdong WANG, Tianhe Wu, Junjie Wang, Yujiu Yang, Yu Qiao, and Ruihang Chu. Smaller models are natural explorers for policy-level diversity in GRPO. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=PI2xku6EDA.

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

Junhao Shi, Qinyuan Cheng, Zhaoye Fei, Yining Zheng, Qipeng Guo, and Xipeng Qiu. How to mitigate overfitting in weak-to-strong generalization? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16100–16118, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.784. URL https://aclanthology.org/2025.acl-long.784/.

Hadar Sivan, Moshe Gabel, and Assaf Schuster. FOSI: Hybrid first and second order optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NvbeD9Ttkx.

Seamus Somerstep, Felipe Maia Polo, Moulinath Banerjee, Yaacov Ritov, Mikhail Yurochkin, and Yuekai Sun. A transfer learning framework for weak to strong generalization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=PeLLMw3wLX.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=GqYSunGmp7.

Dayu Wang, Jiaye Yang, Weikang Li, Jiahui Liang, Liwei Qian, Xin Pei, and Jizhou Huang. It takes 8 tokens: Weak-to-strong off-policy rl via auxiliary branches. *arXiv preprint arXiv:2607.16205*, 2026a.

Zechuan Wang, Siyuan Lu, Hongxuan Zhang, Linjian Mo, Chenyi Zhuang, and Leilei Gan. Teach the magnitude, not the direction: Verifier-bounded credit assignment for multi-turn multi-step llm agents. *arXiv preprint arXiv:2608.13179*, 2026b.

Xiaomi Team. 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*, 2025.

Yihao Xue, Jiping Li, and Baharan Mirzasoleiman. Representations shape weak-to-strong generalization: Theoretical insights and empirical predictions. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ypEW077kle.

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*, 2026a.

Yuqing Yang, Yan Ma, and Pengfei Liu. Weak-to-strong reasoning. In *Findings of the association for computational linguistics: EMNLP 2024*, pp. 8350–8367, 2024.

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, et al. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026b.

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. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2860–2888, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.148. URL https://aclanthology.org/2025.findings-acl.148/.

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

Fangxu Yu, Zinan Lin, Xiaodong Liu, Weijia Xu, Michael Xu, Tianyi Zhou, and Jianfeng Gao. Weak-to-strong on-policy distillation. *arXiv preprint arXiv:2607.26246*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Yige Yuan, Teng Xiao, Shuchang Tao, Xue Wang, Jinyang Gao, Bolin Ding, and Bingbing Xu. Incentivizing strong reasoning from weak supervision. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7138–7156, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.336. URL https://aclanthology.org/2026.eacl-long.336/.

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*, 2026a.

Zhu Zhang, Jixun Wang, Xiaoang Xu, Xiaorong Wang, Zihan Zhou, Zhiyuan Wang, Shuo Wang, Chaojun Xiao, and Yuezhi Zhou. Beyond teacher likelihood: Group-calibrated on-policy distillation for long-context reasoning. *arXiv preprint arXiv:2608.19181*, 2026b.

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. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Shiji Zhou, Wenpeng Zhang, Jiyan Jiang, Wenliang Zhong, Jinjie GU, and Wenwu Zhu. On the convergence of stochastic multi-objective gradient manipulation and beyond. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=ScwfQ7hdwyP.

Yucheng Zhou, Jianbing Shen, and Yu Cheng. Weak to strong generalization for large language models with multi-capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 11583–11612, 2025.

Zhanhui Zhou, Zhixuan Liu, Jie Liu, Zhichen Dong, Chao Yang, and Yu Qiao. Weak-to-strong search: Align large language models via searching over small language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dOJ6CqWDf1.

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

Zhou Ziheng, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation. *arXiv preprint arXiv:2605.27028*, 2026.

Contents

1 Introduction 1

2 Method 3
- 2.1 Preliminary . . . 3
- 2.2 On-Policy Reverse Distillation . . . 4

3 Experiments 5
- 3.1 Weak-to-Strong Distillation for a Successor Model . . . 5
- 3.2 Multi-Teacher Weak-to-Strong Distillation . . . 6
- 3.3 Strong-to-Weak Distillation . . . 6

4 Analysis 7
- 4.1 Broader Comparison with Weak-to-Strong Methods . . . 7
- 4.2 Design and Dynamics of Teacher Guidance . . . 8
- 4.3 Student Behavior under Teacher Guidance . . . 9

5 Conclusion 9

A Related Work 18

B Optimization Properties of Teacher-Direction Scaling 19

C Analysis of Asymmetric Alignment Scaling 20
- C.1 One-Sided Amplification under Positive-Only Scaling . . . 20
- C.2 Isolating the Positive and Negative Alignment Branches . . . 21
- C.3 Mitigating One-Sided Amplification through Branch Scheduling . . . 22

D Training Details 23

E Evaluation Details 23

F Detailed Results for Weak-to-Strong Model Transfer 26
- F.1 Detailed Learning Curves . . . 26
- F.2 Additional Results Across Model Variants and Tasks . . . 27
- F.3 Evaluation Results with Standard Deviations . . . 29

G Multi-Teacher Weak-to-Strong Distillation 30

H Detailed Results for Strong-to-Weak Distillation 30

I Detailed Results for Weak-to-Strong Method Comparisons 31
- I.1 Baseline Implementation Details . . . 31
- I.2 Detailed Learning Curve . . . 32

J Additional Results on Guidance-Direction Construction 33

K Discussion of Key Challenges 34
K.1 Vanishing Policy Gradients 34
K.2 Reference Policy Selection and Length Bias 34

L Detailed Analysis of Student Behavior 36
L.1 Token Alignment Analysis 36
L.2 Response Style Analysis 36

M Reverse Distillation without Policy Matching 37

N Future Works 38

O Computational Cost and Memory Usage 39

A Related Work

Weak-to-Strong Generalization. Weak-to-strong generalization has been observed across language understanding, reward modeling, and reasoning, although weak supervision typically recovers only part of the gap to strong supervision (Burns et al., 2024; Yang et al., 2024). Analyses attribute the gains to correcting weak pseudo-labels, extending coverage beyond the weak teacher, and differences between teacher and student hypothesis classes or representations (Lang et al., 2024; Charikar et al., 2024; Dong et al., 2025; Xue et al., 2025; Medvedev et al., 2025), while naive fine-tuning can instead overfit weak errors (Somerstep et al., 2025; Yao et al., 2025; Shi et al., 2025). For reasoning, W2SR-P trains stronger students on verified weak-model trajectories, S2L-PO and related methods use weaker policies to broaden the student's rollouts, and weak critiques can generate and filter improved responses (Yuan et al., 2026; Ren et al., 2026; Wang et al., 2026a; Jin et al., 2026). These methods change the student's training data, exploration, or feedback, whereas OPRD leaves all three unchanged and only rescales the student's own policy gradient.

On-Policy Distillation. Knowledge distillation for language generation has moved from matching teacher distributions on fixed or teacher-generated sequences (Hinton et al., 2015; Kim & Rush, 2016) to objectives evaluated on student-generated sequences (Gu et al., 2024; Ko et al., 2024). OPD makes this supervision fully on-policy by querying the teacher along the student's current rollouts, addressing the mismatch between the prefixes seen in training and those the student visits at inference (Agarwal et al., 2024). It is now a common step in reasoning post-training (Qwen Team, 2025b; GLM-5 Team, 2026), and later work uses the same interface to consolidate several specialist teachers into one student (Ma et al., 2026; Kimi Team, 2026; Xiaomi Team, 2026), to exploit privileged information available only during training (Zhao et al., 2026; Ye et al., 2026), or to extrapolate the reward implicit in OPD beyond the teacher (Yang et al., 2026a). In all of these, the student is still trained to match a token distribution that the teacher defines, so in the weak-to-strong setting the optimum of the objective is the weak policy itself or a target derived from it.

Distillation with Reinforcement Learning. Methods that combine distillation with verifier-based reinforcement learning differ in how the teacher signal enters optimization. KDRL and later work add a teacher-matching term to the reward objective (Xu et al., 2025; Ramos et al., 2026). Others modify teacher guidance through policy ratios, reward-based selection, group-level calibration, or token-level interventions (Zhang et al., 2026a; Akhondzadeh et al., 2026; Zhang et al., 2026b; Ko et al., 2026; Jia et al., 2026), and another uses a privileged self-teacher to control the magnitude of token-level credit (Wang et al., 2026b). However, a teacher-matching loss introduces a second objective that can compete with reward maximization when the teacher favors a solution the verifier does not reward. OPRD adds no such objective and optimizes reward alone.

Transferring Policy Shifts. Several methods transfer the shift between a post-trained policy and its reference rather than the final policy alone. During decoding, this shift can steer a larger frozen model (Liu et al., 2024; Zhou et al., 2024). During training, it has been used as an alignment target for a stronger model (Zhu et al., 2025), as a proxy teacher built on the student's base policy in W2S-OPD (Yu et al., 2026), and as a dense reward on student rollouts in Direct-OPD (Heo et al., 2026; Feng et al., 2026). In each case the shift itself becomes an optimization target, and it carries only the improvements the weak teacher realized. OPRD instead uses the weak policy delta to rescale the student's policy gradient, so the direction transfers without the shift becoming a target.

Gradient Manipulation. Multi-task optimization combines objectives at the level of gradients rather than losses. Gradient surgery projects one task gradient onto the normal plane of another when the two conflict (Yu et al., 2020), a moving average of past gradients makes this projection more stable (Hsieh et al., 2024), and auxiliary gradients can be gated by their cosine similarity with the main gradient (Du et al., 2019; Zhou et al., 2022). In these methods, every direction is the gradient of a loss the model itself optimizes, and conflicting components are removed or down-weighted. OPRD is closest to this family in form, but it removes nothing and only amplifies the component of the student's policy gradient that already points along the teacher direction, so the stationary points of the student objective in logit space do not move.

B Optimization Properties of Teacher-Direction Scaling

OPRD multiplies the token-level policy gradient by $\mathbf{I} + \lambda_t \mathbf{d}_t \mathbf{d}_t^\top$, which amplifies the component along $\mathbf{d}_t$ by $1 + \lambda_t$ and leaves the orthogonal component unchanged. Over a full response, the transformed token gradients vanish exactly where the original token gradients do, and the corresponding logit update has an ascent bound with an additional nonnegative term.

Fix a response y with T valid tokens and let $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_T)$ collect its next-token logits, with $\pi(\cdot \mid \mathbf{z}_t) = \mathrm{softmax}(\mathbf{z}_t)$. The advantages A_t and the directions $\mathbf{d}_t$ do not depend on $\mathbf{z}$, and the token gradient in Eq. 2.1 is the gradient of the objective for this response,

$$J(\mathbf{z}) = \sum_{t=1}^{T} A_t \log \pi(y_t \mid \mathbf{z}_t), \qquad \mathbf{g}_t = \nabla_{\mathbf{z}_t} J(\mathbf{z}). \tag{B.1}$$

Each block of the Hessian of J is A_t times that of $\log \pi(y_t \mid \mathbf{z}_t)$, whose eigenvalues lie in $[-\frac{1}{2}, 0]$, so J is L-smooth with $L = \max_t |A_t|/2$.

Proposition B.1 (Stationarity and One-Step Ascent)**.** *Let $\mathbf{z}_k$ be the current logits and write $\mathbf{g}_t = \nabla_{\mathbf{z}_t} J(\mathbf{z}_k)$ and $u_t = \mathbf{d}_t^\top \mathbf{g}_t$ for the alignment coefficient, with $\|\mathbf{d}_t\|_2 = 1$ and $\lambda_t \geq 0$ fixed for this step. With step size $\eta > 0$, set*

$$\widetilde{\mathbf{g}}_t = (\mathbf{I} + \lambda_t \mathbf{d}_t \mathbf{d}_t^\top)\mathbf{g}_t, \qquad \mathbf{z}_{k+1,t} = \mathbf{z}_{k,t} + \eta\, \widetilde{\mathbf{g}}_t. \tag{B.2}$$

Then $\widetilde{\mathbf{g}}_t = \mathbf{0}$ for every t if and only if $\nabla_{\mathbf{z}} J(\mathbf{z}_k) = \mathbf{0}$. If in addition $\eta L(1 + \bar{\lambda}) \leq 1$ with $\bar{\lambda} = \max_t \lambda_t$,

$$J(\mathbf{z}_{k+1}) - J(\mathbf{z}_k) \;\geq\; \frac{\eta}{2} \|\nabla_{\mathbf{z}} J(\mathbf{z}_k)\|_2^2 + \frac{\eta}{2} \sum_t \lambda_t u_t^2. \tag{B.3}$$

Proof. Let $\mathbf{g} = \nabla_{\mathbf{z}} J(\mathbf{z}_k)$ and let $\mathbf{P}$ be the block-diagonal matrix with blocks $\mathbf{I} + \lambda_t \mathbf{d}_t \mathbf{d}_t^\top$, so that $\mathbf{z}_{k+1} = \mathbf{z}_k + \eta \mathbf{P}\mathbf{g}$. Each block has eigenvalue $1 + \lambda_t$ along $\mathbf{d}_t$ and 1 on the orthogonal complement. Hence $\mathbf{P}$ is positive definite and therefore invertible, which gives the first claim, and

$$\mathbf{g}^\top \mathbf{P} \mathbf{g} = \|\mathbf{g}\|_2^2 + \sum_t \lambda_t u_t^2, \qquad \mathbf{P}^2 \preceq (1 + \bar{\lambda})\mathbf{P}.$$

By L-smoothness,

$$\begin{aligned} J(\mathbf{z}_{k+1}) &\geq J(\mathbf{z}_k) + \eta\, \mathbf{g}^\top \mathbf{P} \mathbf{g} - \frac{L\eta^2}{2}\, \mathbf{g}^\top \mathbf{P}^2 \mathbf{g} \\ &\geq J(\mathbf{z}_k) + \eta \Big(1 - \frac{L\eta(1 + \bar{\lambda})}{2}\Big) \mathbf{g}^\top \mathbf{P} \mathbf{g} \\ &\geq J(\mathbf{z}_k) + \frac{\eta}{2}\, \mathbf{g}^\top \mathbf{P} \mathbf{g} \\ &= J(\mathbf{z}_k) + \frac{\eta}{2} \|\mathbf{g}\|_2^2 + \frac{\eta}{2} \sum_t \lambda_t u_t^2. \end{aligned}$$

□

Setting $\lambda_t = 0$ in Eq. B.3 recovers the bound $\frac{\eta}{2}\|\nabla_{\mathbf{z}} J(\mathbf{z}_k)\|_2^2$ of an unscaled step, so the second term is what scaling adds. It grows with the component of the verifier-driven policy gradient along the teacher direction and disappears when the two are orthogonal at every token. Scaling therefore adds to the progress guaranteed at each step without changing where the update vanishes, and the price is the tighter condition $\eta L(1 + \bar{\lambda}) \leq 1$ on the step size, since the scaled update is longer. The gain depends on u_t^2, so alignments of equal magnitude contribute equally whether the student follows or opposes the teacher. Appendix C.1 analyzes what changes when the two branches use different scales.

The stationarity and ascent statements concern the fixed-response objective $J(\mathbf{z})$ in token logit space. Note that they do not assert preservation of stationary points or the same ascent bound in neural network parameter space. For a parameterization $\mathbf{z}_\theta$ with Jacobian $\mathbf{K}_\theta$, the backward signals are $\mathbf{K}_\theta^\top \mathbf{g}$ and $\mathbf{K}_\theta^\top \mathbf{P}\mathbf{g}$. Although $\mathbf{P}$ is invertible, cancellations after multiplication by $\mathbf{K}_\theta^\top$ need not be preserved.

C Analysis of Asymmetric Alignment Scaling

C.1 One-Sided Amplification under Positive-Only Scaling

Positive-only amplification is locally well motivated. When $u_t > 0$, the component of the sampled student gradient $\mathbf{g}_t$ along the teacher-derived direction $\mathbf{d}_t$ follows the teacher's post-training shift. Amplifying this component therefore reinforces an update supported by both the teacher shift and the verifier-driven student gradient. A related positive-gating rule is used by Du et al. (2019), who weight auxiliary updates by the positive part of their gradient cosine similarity.

However, applying different scales to the two alignment signs introduces a one-sided effect. Let λ_+ and λ_- denote the scales applied when $u_t \geq 0$ and $u_t < 0$, respectively. The coefficient multiplying $\mathbf{d}_t$ in the added correction is

$$\begin{aligned} c_t &:= \lambda_+ u_t \mathbf{1}\{u_t \geq 0\} + \lambda_- u_t \mathbf{1}\{u_t < 0\} \\ &= \tfrac{\lambda_+ + \lambda_-}{2}\, u_t + \tfrac{\lambda_+ - \lambda_-}{2}\, |u_t|. \end{aligned} \tag{C.1}$$

This decomposition separates sign-symmetric scaling from the asymmetry introduced by using different scales for the two branches. The first term symmetrically scales u_t by the average of the two branch scales. Because it preserves the sign of u_t, positive and negative contributions can cancel across tokens. The second term depends on $|u_t|$ and appears only when the branch scales differ. In particular, when $\lambda_+ > \lambda_-$, this term remains nonnegative for either sign of u_t. It therefore cannot be canceled by changes in the alignment sign, leaving a one-sided coefficient on the local teacher direction.

Under positive-only scaling, $\lambda_- = 0$. If positive and negative alignments nearly balance across sampled tokens and rollouts, such that $\mathbb{E}[u_t] \approx 0$, then

$$\begin{aligned} \mathbb{E}[c_t] &= \tfrac{\lambda_+}{2}\, (\mathbb{E}[u_t] + \mathbb{E}[|u_t|]) \\ &\approx \tfrac{\lambda_+}{2}\, \mathbb{E}[|u_t|] > 0. \end{aligned} \tag{C.2}$$

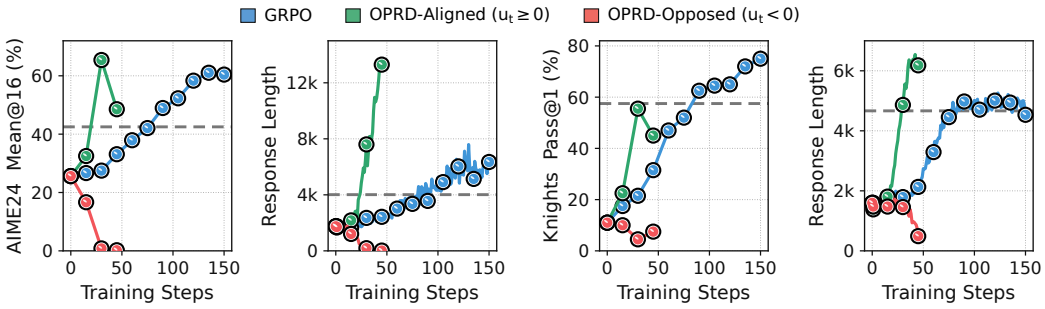
Thus, even when the signed alignments cancel on average, the scalar coefficient c_t remains positive on average under positive-only scaling. This residual coefficient is governed by the mean alignment magnitude $\mathbb{E}[|u_t|]$, rather than the small signed mean $\mathbb{E}[u_t]$.

Importantly, this residual amplification need not reflect only reward-relevant teacher progress. The sign of u_t reveals whether $\mathbf{d}_t$ and $\mathbf{g}_t$ agree, but not why they agree. At an individual sampled token, we write $\mathbf{g}_t = \mathbf{g}_t^\star + \boldsymbol{\epsilon}_t$, where $\mathbf{g}_t^\star$ denotes the underlying reward-improving signal and $\boldsymbol{\epsilon}_t$ aggregates incidental or misattributed components arising from coarse response-level credit assignment, rollout and mini-batch sampling, and estimator-specific effects that need not correspond to actions responsible for higher reward. Similarly, $\mathbf{d}_t$ captures all changes induced by teacher post-training, including both reward-relevant progress and incidental behavioral changes. Positive alignment may therefore arise from either useful teacher-acquired progress or an incidental tendency shared by the two vectors. Positive-only scaling cannot distinguish between these cases and amplifies the aligned component regardless of its source.

Response length provides one concrete example of such a shared tendency. In reasoning tasks, higher verifier rewards are often associated with longer reasoning traces, so the student gradient $\mathbf{g}_t$ may favor token-level changes that prolong generation. The teacher direction $\mathbf{d}_t$ may encode a similar tendency acquired during teacher post-training. This tendency may represent useful additional reasoning, but it may also reflect length-dependent effects in the policy-gradient estimate (Liu et al., 2025). When it is shared by both signals, positive-only scaling amplifies it whenever it produces positive alignment.

C.2 Isolating the Positive and Negative Alignment Branches

To examine the branch-specific effects, we isolate the two alignment branches by activating gradient scaling only when $u_t \geq 0$ (*positive-only*) or only when $u_t < 0$ (*negative-only*), while holding all other training settings fixed within each task. Figure 5 reports evaluation performance and response length during training on mathematics and Knights & Knaves tasks. Across both tasks, positive-only scaling produces rapid early gains accompanied by a sharp increase in response length. Performance then begins to decline as responses grow toward the generation limit. Negative-only scaling exhibits the opposite pattern: responses become shorter, while performance quickly falls to near zero.



(a) Math (b) Knights & Knaves

Figure 5: **Evaluation performance and response length over training for OPRD variants with only the positive- or negative-alignment branch active.** For Math and Knights & Knaves, we use Qwen3-4B and Qwen3-4B-Base teacher checkpoints obtained after 75 and 105 RL training steps, respectively. The two single-branch OPRD variants are trained for 45 steps with $\lambda = 0.5$, while GRPO is shown through 150 steps for reference. The maximum generation lengths are 20K and 8K tokens for the two settings, respectively. The gray dashed lines denote the performance or response length of the corresponding weak teachers. All other training settings follow the dataset-specific default configurations described in Appendix D.

The rapid gains under positive-only scaling suggest that teacher-aligned components provide effective early transfer of the progress acquired during teacher post-training. By contrast, the collapse under negative-only scaling suggests that teacher-opposing components are less reliable early in training, when the student's rollouts remain weak. Because the verifier provides only response-level feedback, even a rewarded trajectory may contain locally unhelpful token choices whose gradients are negatively aligned with the teacher shift. Applying negative-branch scaling at full strength from the outset can therefore reinforce unreliable token-level updates.

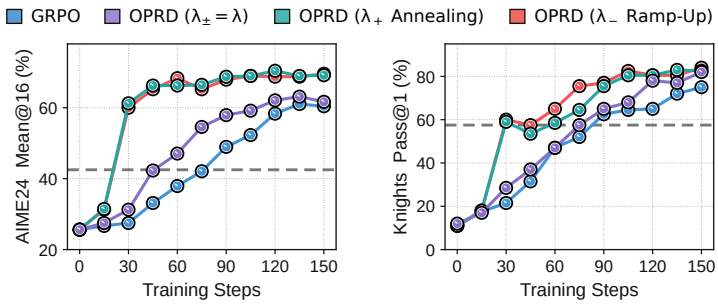
The response-length dynamics are also consistent with the shared tendency discussed in Appendix C.1. In both tasks, performance improvements under GRPO are accompanied by longer reasoning traces, suggesting that the student gradient $\mathbf{g}_t$ favors token-level changes that prolong generation. The teacher develops a similar tendency during RL post-training, which may be encoded in $\mathbf{d}_t$. Positive-only scaling reinforces this shared tendency and rapidly drives responses toward the generation limit. Negative-only scaling instead amplifies student-gradient components that oppose $\mathbf{d}_t$, counteracting the length-increasing tendency and producing shorter responses.

C.3 Mitigating One-Sided Amplification through Branch Scheduling

The isolated-branch results suggest that the positive and negative branches play complementary roles over training. The positive branch amplifies components supported by both the teacher shift and the verifier-driven student gradient, thereby providing rapid early transfer. The negative branch instead amplifies verifier-supported departures from the teacher direction, which may help the stronger student move beyond the weak teacher. However, these departures are less reliable early in training, when the student's on-policy rollouts remain weak. This difference motivates controlling the relative strengths of the two branches over training.

We compare three strategies to avoid persistent one-sided amplification. In default OPRD, $\lambda_+ = \lambda$ while λ_- rises from 0 to λ. The larger positive scale provides an early teacher-aligned bootstrap, while increasing λ_- progressively amplifies verifier-supported departures. Once $\lambda_- = \lambda_+ = \lambda$, the asymmetric term proportional to $|u_t|$ vanishes and $c_t = \lambda u_t$. Thus the schedule preserves early transfer while strengthening departures from the weak teacher.

Alternatively, we keep $\lambda_- = 0$ and gradually decrease λ_+ from λ to 0. This schedule likewise uses positive-branch scaling as an early bootstrap but progressively removes the added teacher-direction correction. Once $\lambda_+ = 0$, both branch scales are zero, so the transformed gradient reduces to the original verifier-driven policy gradient and training returns to GRPO. As a schedule-free alternative, we also consider fixed symmetric scaling, which sets $\lambda_+ = \lambda_- = \lambda$ throughout training. This removes one-sided amplification from the outset but activates the distinct effects of both branches simultaneously.



(a) Math (b) Knights & Knaves

Figure 6: **Comparison of three branch-scheduling strategies.** We compare the default λ_- ramp-up ($\lambda_+ = 0.5$, $\lambda_- : 0 \to 0.5$), λ_+ annealing ($\lambda_+ : 0.5 \to 0$, $\lambda_- = 0$), and fixed symmetric scaling ($\lambda_+ = \lambda_- = 1.0$) against GRPO. The two scheduled variants use horizons of 30 updates for Math and 75 updates for Knights & Knaves. We use Qwen3-4B and Qwen3-4B-Base teacher checkpoints obtained after 75 and 105 RL training steps, respectively. Gray dashed lines denote teacher performance. All other training configurations follow Appendix D.

As shown in Figure 6, the two scheduled variants begin with positive-only amplification and achieve rapid early gains, whereas fixed symmetric scaling improves much more slowly despite using $\lambda = 1.0$: it only gradually breaks through on Math and yields limited early gains on Knights & Knaves. Because response-level feedback can reward trajectories containing locally incorrect or incidental steps, the resulting $u_t < 0$ components are less reliable on weak early rollouts and can dampen the positive-branch bootstrap when amplified from the outset. Activating only λ_+ is therefore the more reliable default for early acceleration.

The later acceleration of fixed symmetric scaling on Knights & Knaves suggests that λ_- becomes useful once stronger student rollouts provide more informative signals. It can then amplify verifier-supported departures that move beyond the weak teacher. Annealing only λ_+ shows that returning to verifier-only optimization after the bootstrap is viable, but forgoes explicit amplification of these departures. We therefore use λ_- ramp-up to retain early λ_+ acceleration while later reducing one-sided amplification and supporting progress beyond the teacher.

D TRAINING DETAILS

For mathematics, we train on DAPO-Math-17K (Yu et al., 2025). For Reasoning Gym (Stojanovski et al., 2026), we train on 19,800 examples per task from Knights & Knaves, Quantum Lock, String Manipulation, and Countdown. For code generation, we train on 24,287 problems from the DeepCoder Preview Dataset: 16,252 from PrimeIntellect, 7,436 from TACO, and 599 from the LCBv5 training split.

Table 4 summarizes the Math and Reasoning Gym training settings, Table 5 gives the code-generation settings, and Table 6 reports method-specific configurations for Math and Reasoning Gym. Scenario-specific settings appear in their respective Appendix sections. The mathematics and Reasoning Gym experiments use Qwen3-family models (Qwen Team, 2025b) and run on four NVIDIA B200 GPUs using Fully Sharded Data Parallel (FSDP) (Zhao et al., 2023). The code OPRD, OPD, and KDRL profiles also use four B200 GPUs.

OPD can also be implemented as token-level policy optimization on student-sampled tokens using the teacher-to-student log-probability ratio as the advantage, with negligible empirical differences from direct reverse-KL optimization. We adopt KDRL (Xu et al., 2025) as the representative baseline combining distillation with RLVR (see also GKD (Agarwal et al., 2024) and dGRPO (Ramos et al., 2026)). For a fair comparison, its distillation coefficient is annealed to 0.

Table 4: **Default training settings for Math and Reasoning Gym.** These configurations are shared across all methods. Method-specific settings are provided in Table 6.

Settings	Math	Reasoning Gym
Data and Models		
Training data	DAPO-Math-17K	Knights & Knaves, Quantum Lock, String Manipulation, and Countdown (19,800 examples per task)
Prompt format	Chat template with a system prompt	Chat template without a system prompt
Student policy	Qwen3-8B (non-thinking)	Qwen3-8B-Base (non-thinking)
Teacher policy	Qwen3-4B (step 75, non-thinking)	Qwen3-4B-Base (step 75 for String task, step 105 for the other tasks, non-thinking)
Optimization		
Training horizon	150 policy updates	150 policy updates
Prompt batch / mini-batch	64 / 64	64 / 32
Rollouts per prompt	8	8
Optimizer	AdamW, $\beta = (0.9, 0.999)$, weight decay 0.01, gradient clipping 1.0	AdamW, $\beta = (0.9, 0.999)$, weight decay 0.01, gradient clipping 1.0
Learning rate	1×10^{-6} (constant schedule with 10 warm-up updates)	1×10^{-6} (constant schedule with 10 warm-up updates)
Policy optimization	PPO clipping range $[0.20, 0.28]$, no standard-deviation normalization, no KL or entropy regularization	PPO clipping range $[0.20, 0.28]$, no standard-deviation normalization, no KL or entropy regularization
Generation		
Training-time decoding	Temperature 1.0, top-p 1.0, no top-k	Temperature 1.0, top-p 1.0, no top-k
Maximum prompt length	2,048 tokens	2,048 tokens
Maximum response length	20,480 tokens	8,192 tokens
Length-based reward	No penalty up to 16,384 tokens, then linear penalty reaching -1 at 20,480 tokens	–

E EVALUATION DETAILS

For mathematics, we evaluate on AIME'24/25 (MAA, 2024–2025), HMMT'25 (Dekoninck et al., 2026), and OlympiadBench (He et al., 2024). For Reasoning Gym, we use 200 held-out examples per task for Knights & Knaves, Quantum Lock, String Manipulation, and Countdown. For code

Table 5: **Default training settings for code generation.** These settings follow the Qwen3-14B-Base code-transfer profiles using the step-435 Qwen3-8B-Base teacher.

Settings	Code
Data and Models	
Training data	DeepCoder Preview Dataset: PrimeIntellect (16,252), TACO (7,436), and LCBv5 train (599); 24,287 problems in total
Prompt format	Structured chat template without a system prompt; thinking disabled
Student policy	Qwen3-14B-Base (non-thinking)
Teacher policy	Qwen3-8B-Base (GRPO step 435, non-thinking)
Teacher reference	Original Qwen3-8B-Base (OPRD only)
Optimization	
Training horizon	150 policy updates
Prompt batch / mini-batch	64 / 32
Rollouts per prompt	8
Optimizer	AdamW, $\beta = (0.9, 0.999)$, weight decay 0.01, gradient clipping 1.0
Learning rate	1×10^{-6} (constant schedule with 10 warm-up updates)
Policy optimization	For verifier-based methods: PPO clipping range $[0.20, 0.28]$, token-mean loss, no standard-deviation normalization, no KL or entropy regularization
Distillation schedules	OPD: sampled reverse KL with importance-ratio clipping $[0.5, 2.0]$; KDRL: $\beta_k : 0.005 \to 0$ over 45 updates; OPRD: $\lambda_+ = 0.5$, $\lambda_- : 0 \to 0.5$ over 45 updates
OPRD token support	Sampled token $\cup$ student top-10
Generation	
Training-time decoding	Temperature 1.0, top-p 1.0, no top-k
Maximum prompt length	2,048 tokens
Maximum response length	16,384 tokens
Length-based reward	For verifier-based methods, no penalty up to 12,288 tokens, then a linear penalty reaching -1 at 16,384 tokens

Table 6: **Method-specific training settings for Math and Reasoning Gym.** All distillation-based methods use the same task-specific frozen teacher checkpoint specified in Table 4.

Settings	GRPO	OPD	KDRL	OPRD
Optimization				
Frozen teacher	–	Task-specific	Task-specific	Task-specific
Teacher reference	–	–	–	Raw Qwen3-4B family
Teacher signal	–	Teacher–student log-probability ratio	K2 signal	Teacher-shift direction $\mathbf{d}_t$
Teacher temperature	–	1.0	1.0	1.0
Token support	–	Sampled tokens	Sampled tokens	Sampled $\cup$ student top-10
Coefficient schedule	–	Fixed at 1.0	$\beta_k : 0.005 \to 0$ over Math: 30 updates Reasoning Gym: 75 updates	$\lambda_+ = 0.5$, $\lambda_- : 0 \to 0.5$ over Math: 30 updates Reasoning Gym: 75 updates

generation, we evaluate on LCBv5 (Jain et al., 2025), HumanEval+ (Liu et al., 2023), CodeContests (Li et al., 2022), and APPS (Hendrycks et al., 2021).

We report Mean@16 for mathematics, Pass@1 for Reasoning Gym, and Mean@4 for code generation. For N questions and m sampled responses per question, Mean@m averages the binary correctness scores over all Nm responses. With binary scores, Mean@m is the empirical Pass@1 estimate from m samples per question, rather than the probability that any of the m attempts succeeds. All reported scores are multiplied by 100. The Avg. column is the unweighted mean of the four benchmark scores. Response averaging is distinct from checkpoint averaging: mathematics and Reasoning Gym results in Table 1, and code-generation results in Table 10, average trained-policy evaluations at steps 30, 60, 90, 120, and 150, while teacher and initial-student rows use fixed checkpoints.

Table 7 summarizes the default evaluation settings for Math and Reasoning Gym; Table 8 gives the code-generation settings. Across methods, all trained policies use the same task-specific evaluation settings.

Table 7: **Default evaluation settings for Math and Reasoning Gym.** Teacher and initial-student checkpoints use the same decoding and scoring protocols as trained student checkpoints.

Settings	Math	Reasoning Gym
Benchmarks and Metrics		
Reported benchmarks	AIME'24, AIME'25, HMMT'25, OlympiadBench	Knights & Knaves, Quantum Lock, String Manipulation, Countdown (200 examples per task)
Evaluation metric	Mean@16	Pass@1
Rollouts per problem	16	1
Decoding		
Decoding parameters	Temperature 0.7, top-p 0.8, top-k 20	Temperature 0.6, top-p 0.95, top-k 20
Maximum prompt length	2,048 tokens	2,048 tokens
Maximum response length	38,912 tokens	8,192 tokens
Scoring and Reporting		
Scoring	Exact match after answer normalization	Nonempty boxed answer required, K&K: exact match after normalization, Quantum Lock: 1.0 for a reference-length valid path, 0.5 for any other valid path, 0 otherwise, String Manipulation: case-sensitive exact match, Countdown: valid expression using each given number exactly once and reaching the target
Checkpoint averaging	5-checkpoint mean (30-update intervals)	5-checkpoint mean (30-update intervals)

Table 8: **Default evaluation settings for code generation.** Teacher and initial-student checkpoints use the same decoding and scoring protocols as trained student checkpoints.

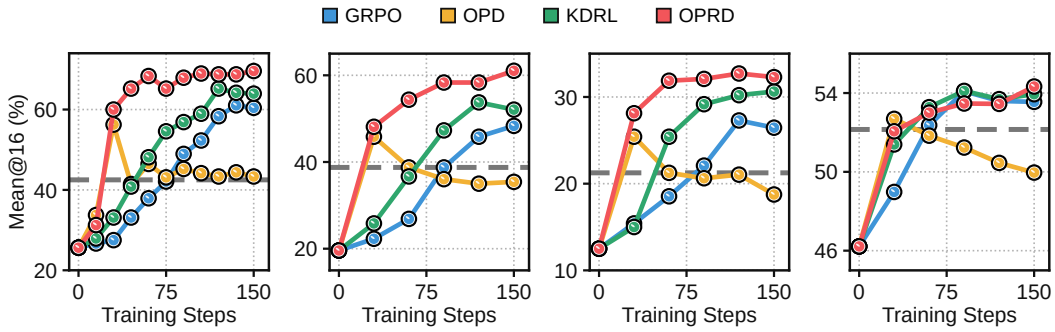
Settings	Code
Benchmarks and Metrics	
Reported benchmarks	LCBv5 (279 problems), HumanEval+ (163; excluding HumanEval/32), CodeContests (106), APPS Competition (693)
Evaluation metric	Mean@4
Rollouts per problem	4
Decoding	
Decoding parameters	Temperature 0.7, top-p 0.8, top-k 20
Prompt format	User-only chat template without a system prompt; thinking disabled; Python solution in a code block
Maximum prompt length	2,048 tokens for LCBv5; task-dependent within a 32,768-token context for independent benchmarks
Maximum response length	30,720 tokens
Scoring and Reporting	
Scoring	Binary correctness: extracted Python code must pass all tests; HumanEval+ requires both base and plus tests
Checkpoint averaging	5-checkpoint mean (steps 30, 60, 90, 120, 150); fixed teacher and initial-student checkpoints

F Detailed Results for Weak-to-Strong Model Transfer

F.1 Detailed Learning Curves

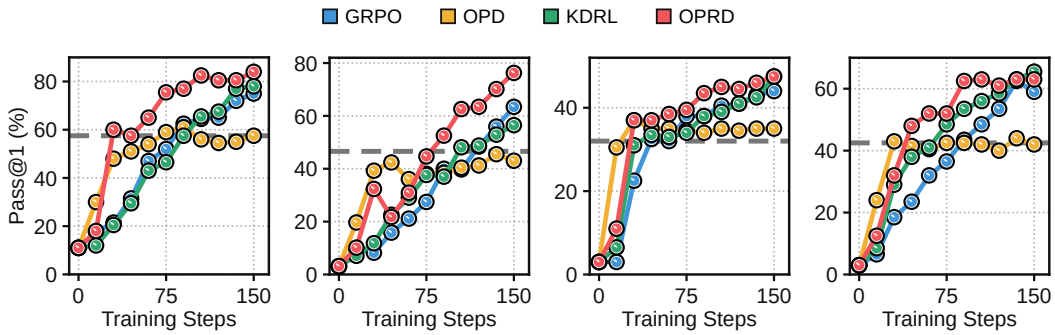
We study successive model transfer within the Qwen3 family (Qwen Team, 2025b), using GRPO-trained 4B-scale models as teachers to accelerate the post-training of larger 8B-scale students. We select intermediate teacher checkpoints whose the performance exceeds that of the initial student but remains below the student's end-of-training GRPO performance. We also considered cross-generation transfer from Qwen2.5 (Qwen Team, 2025a) to Qwen3. In preliminary experiments, however, the Qwen2.5-3B and 7B checkpoints remain below this target range (around 14% on AIME'24), while obtaining suitable post-trained checkpoints and the corresponding teacher-shift signals would require substantially more compute. We therefore focus on controlled within-family transfer.

We compare OPRD with GRPO, OPD, and KDRL on four mathematics benchmarks and four Reasoning Gym tasks, reporting Mean@16 and Pass@1, respectively. Figure 1 aggregates performance across benchmarks, whereas Table 1 averages each trained policy over five checkpoints. Figure 7 and Figure 8 show the corresponding benchmark-level learning curves. Across these benchmarks, OPRD generally retains OPD's rapid early improvement. Unlike OPD, which plateaus near the weak teacher, OPRD continues to improve beyond it and reaches GRPO's end-of-training performance substantially earlier.



(a) AIME'24 (b) AIME'25 (c) HMMT'25 (d) OlympiadBench

Figure 7: **Learning curves for successive model transfer on individual math benchmarks.** We use Qwen3-4B as the teacher and Qwen3-8B as the student. The gray dashed line denotes the performance of the weak teacher. All training settings follow the default Math configuration described in Appendix D.

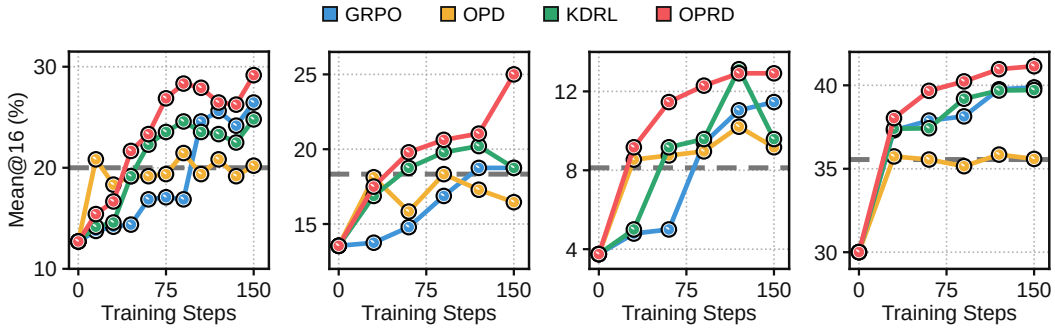


(a) Knights & Knaves (b) Quantum Lock (c) String Manipulation (d) Countdown

Figure 8: **Learning curves for successive model transfer on individual reasoning benchmarks.** We use Qwen3-4B-Base as the teacher and Qwen3-8B-Base as the student. The gray dashed line denotes the performance of the weak teacher. All training settings follow the default Reasoning Gym configuration described in Appendix D.

F.2 Additional Results Across Model Variants and Tasks

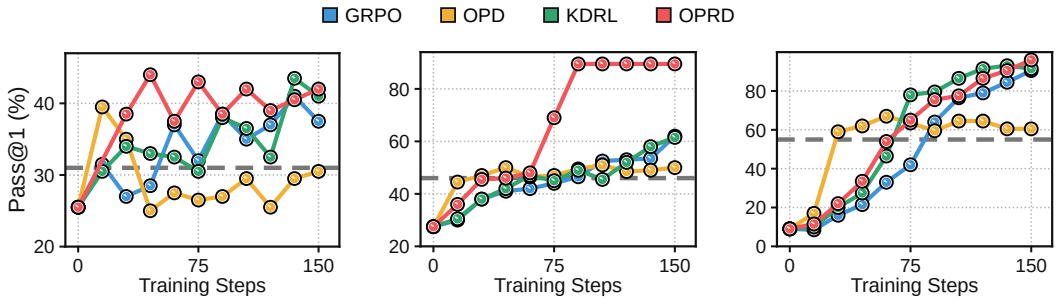
The instruction-tuned Qwen3 results reported in Section F.1 exhibit a potential response-length confound. Although thinking mode is disabled, longer responses may implicitly elicit some of the reasoning behavior associated with that mode, leading to abrupt, transient score gains. In Figure 7, for example, OPD briefly surpasses the teacher on both AIME'24 and AIME'25 at step 30 before returning toward a teacher-level plateau. Such behavior can confound comparisons of early learning speed. We therefore evaluate Qwen3-Base models, for which this effect is less pronounced, in Figure 9. OPRD again substantially accelerates weak-to-strong generalization, achieving high performance much earlier than the baselines on all four benchmarks.



(a) AIME'24 (b) AIME'25 (c) HMMT'25 (d) OlympiadBench

Figure 9: **Learning curves on individual math benchmarks using Qwen3-Base models.** We use Qwen3-4B-Base as the teacher and Qwen3-8B-Base as the student. The gray dashed line denotes the performance of the weak teacher. All other settings follow the default Math configuration in Appendix D, but we omit the system prompt and reduce the mini-batch size to 32, yielding two optimizer steps per training batch.

Reward gains often conincide with longer responses. For instruction-tuned Qwen3, this makes a potential confound: distillation gains may simply reflect longer responses eliciting latent thinking behavior. To test whether OPRD depends on this effect, we evaluate three more reasoning tasks in Figure 10, where, as in String Manipulation, post-training shortens responses by a factor of three to four relative to the raw checkpoints. OPRD still improves substantially faster than the baselines, quickly reaching GRPO's eventual plateau while reducing response length. This opposite trend shows that its gains are not tied to response-length growth. OPRD's gradient scaling can nevertheless magnify length bias in the teacher-shift signal, as discussed in Appendix C and Appendix K.2. We resolve this by using a later teacher checkpoint, rather than the raw model, as the reference policy.



(a) Zebra Puzzles (b) Color Cube Rotation (c) Binary Matrix

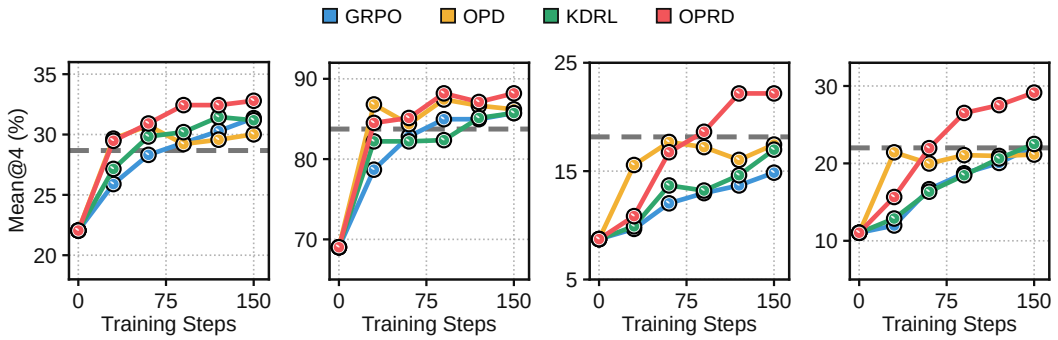
Figure 10: **Learning curves on additional three Reasoning Gym tasks.** We use Qwen3-4B-Base as the teacher and Qwen3-8B-Base as the student. The gray dashed line denotes the performance of the weak teacher. For Color Cube Rotation and Binary Matrix, we use the teacher checkpoints from steps 30 and 45, respectively, as the reference policies instead of the raw step-0 models to mitigate length bias (see Appendix K.2 for details).

Table 9 summarizes the checkpoint-averaged results: OPRD leads across these additional weak-to-strong settings.

Table 9: **Additional Qwen3-Base weak-to-strong results on Math and Reasoning Gym.** Qwen3-4B-Base GRPO checkpoints teach Qwen3-8B-Base students. We report Mean@16 for Math and Pass@1 for Reasoning Gym. Teacher and initial-student rows use fixed checkpoints; trained rows average steps 30, 60, 90, 120, and 150. Corresponding curves appear in Figure 9 and Figure 10; bold marks the best trained result.

	Math Reasoning					Reasoning Gym			
Policy	AIME'24	AIME'25	HMMT'25	Olympiad	**Avg.**	Zebra	Color	Binary	**Avg.**
	Qwen3-4B-Base (Teacher) → Qwen3-8B-Base (Student)					**Qwen3-4B-Base (Teacher) → Qwen3-8B-Base (Student)**			
Teacher	20.00	18.33	8.13	35.55	20.50	31.00	46.00	55.00	44.00
Student	12.71	13.54	3.75	30.01	15.00	25.50	27.50	9.00	20.67
+ GRPO	20.00	16.58	8.33	38.59	20.88	35.40	48.30	56.50	46.73
+ OPD	20.00	17.21	9.13	35.58	20.48	29.10	48.30	62.10	46.50
+ KDRL	21.92	18.88	9.29	38.69	22.19	35.60	49.50	61.40	48.83
+ OPRD	**24.79**	**20.79**	**11.75**	**40.01**	**24.34**	**39.10**	**72.40**	**66.80**	**59.43**

For code transfer from a Qwen3-8B-Base teacher to a Qwen3-14B-Base student, Figure 11 shows learning curves and Table 10 gives checkpoint averages. OPRD reaches 40.13 Mean@4, 1.67 points above OPD. By step 90, it leads every trained baseline on all four benchmarks; its checkpoint average also exceeds the weak teacher on three.



(a) LCBv5 (b) HumanEval+ (c) CodeContests (d) APPS

Figure 11: **Learning curves on individual code-generation benchmarks.** We use a Qwen3-8B-Base teacher from GRPO step 435 and a Qwen3-14B-Base student. The gray dashed line marks the fixed teacher's Mean@4, and all methods start from the same initial student. Other settings follow the Code configurations in Appendix D and Appendix E.

Table 10: **Code generation results for successive model transfer.** A Qwen3-8B-Base teacher guides a Qwen3-14B-Base student. Scores are Mean@4 (%); trained rows average steps 30, 60, 90, 120, and 150, while teacher and initial-student rows use fixed checkpoints. Avg. is unweighted; bold marks the best result.

Policy	LCBv5	HumanEval+	CodeContests	APPS	**Avg.**
	Qwen3-8B-Base (Teacher) → Qwen3-14B-Base (Student)				
Teacher	28.67	83.74	**18.16**	22.01	38.15
Student	22.04	69.02	8.73	11.04	27.71
+ GRPO	29.03	83.44	12.64	17.93	35.76
+ OPD	29.84	86.29	16.79	20.92	38.46
+ KDRL	29.96	83.53	13.68	18.17	36.33
+ OPRD	**31.61**	**86.63**	18.11	**24.16**	**40.13**

F.3 Evaluation Results with Standard Deviations

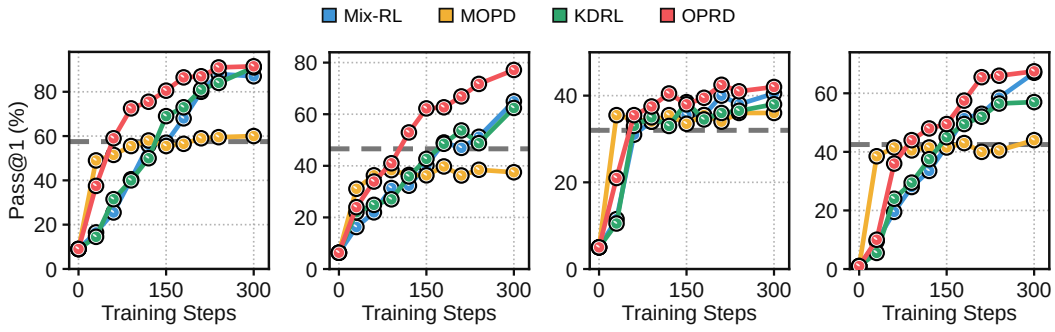
To assess the evaluation-time robustness of the comparisons in Table 1, we report response-resampling variability in Table 11. Because multi-seed post-training is prohibitively expensive, we hold the benchmark problems and trained checkpoints fixed and resample only their responses. Each of 1,000 replicates draws 16 responses with replacement from a pool of 32 per Math problem and one from a pool of eight per Reasoning Gym problem. Trained-policy results are averaged over five checkpoints within each replicate, and we report the resulting mean and sample standard deviation. OPRD still surpasses the strongest baseline by approximately 8.0 points on Math and 9.9 points on Reasoning Gym, margins far exceeding the observed response-resampling variability.

Table 11: **Evaluation results with standard deviations for successive model transfer on mathematics and reasoning tasks.** We use intermediate GRPO checkpoints of 4B-scale Qwen3 models as teachers and report Mean@16 for mathematics and Pass@1 for Reasoning Gym as the bootstrap mean ± sample standard deviation over 1,000 response-resampled evaluations, with benchmark items held fixed. In each bootstrap replicate, we sample 16 responses with replacement from a pool of 32 for each mathematics problem and one response from a pool of eight for each Reasoning Gym problem.

	Math Reasoning					**Reasoning Gym**				
Policy	AIME'24	AIME'25	HMMT'25	Olympiad	**Avg.**	Knights	Quantum	String	Count	**Avg.**
	Qwen3-4B (Teacher) → Qwen3-8B (Student)					**Qwen3-4B-Base (Teacher) → Qwen3-8B-Base (Student)**				
Teacher	41.77 ± 1.48	36.77 ± 1.38	21.44 ± 1.15	52.02 ± 0.23	38.00 ± 0.58	54.92 ± 2.93	41.64 ± 2.73	33.78 ± 1.17	42.13 ± 1.57	43.12 ± 1.11
Student	24.25 ± 1.09	19.84 ± 1.09	13.04 ± 0.95	46.21 ± 0.24	25.83 ± 0.45	11.71 ± 1.96	5.00 ± 1.35	3.61 ± 1.12	2.83 ± 1.05	5.79 ± 0.70
+ GRPO	46.88 ± 0.63	36.61 ± 0.55	22.47 ± 0.51	52.48 ± 0.10	39.61 ± 0.25	53.17 ± 1.21	34.70 ± 0.93	35.51 ± 0.59	41.95 ± 0.64	41.33 ± 0.45
+ OPD	46.62 ± 0.74	37.91 ± 0.59	21.40 ± 0.47	51.30 ± 0.10	39.31 ± 0.27	55.78 ± 1.31	39.91 ± 1.19	34.55 ± 0.73	42.11 ± 0.78	43.09 ± 0.51
+ KDRL	54.20 ± 0.63	42.80 ± 0.59	25.66 ± 0.55	**53.39** ± 0.10	44.01 ± 0.26	53.91 ± 1.14	37.83 ± 1.12	38.10 ± 0.58	49.54 ± 0.75	44.84 ± 0.48
+ OPRD	**67.40** ± 0.63	**55.69** ± 0.64	**31.63** ± 0.58	**53.28** ± 0.09	**52.00** ± 0.27	**72.02** ± 0.94	**49.70** ± 1.02	**42.11** ± 0.57	**55.30** ± 0.75	**54.78** ± 0.42

G Multi-Teacher Weak-to-Strong Distillation

We jointly train one student on an equal mix of four Reasoning Gym tasks, pairing each example with its specialist teacher. Each task receives roughly one-quarter of the single-teacher exposure, so we train for 300 updates with the same coefficient schedules. In Figure 12, OPRD learns fastest and reaches the highest Pass@1 on all four tasks, including String Manipulation, where rewards rise as responses shorten while the other tasks generally lengthen. MOPD shows cross-task interference on Quantum Lock, falling below its specialist teacher; OPRD transfers specialist capabilities without this degradation.

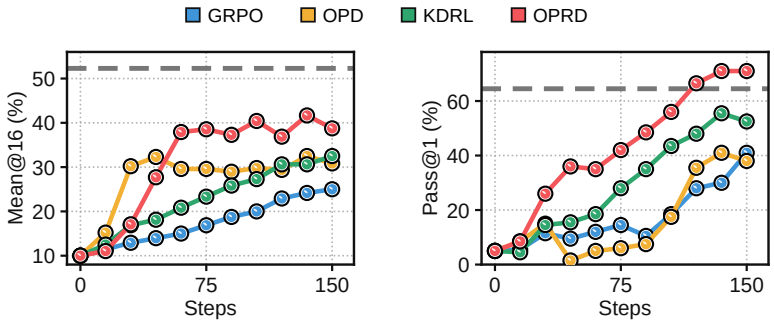


(a) Knights & Knaves (b) Quantum Lock (c) String Manipulation (d) Countdown

Figure 12: **Learning curves on individual Reasoning Gym tasks under multi-teacher distillation.** We use four task-specific Qwen3-4B-Base models as teachers and jointly train a Qwen3-8B-Base student. The gray dashed line denotes the performance of the corresponding specialist teacher. All other settings follow the configurations described in Section D.

H Detailed Results for Strong-to-Weak Distillation

Both strong-to-weak runs use prompt batch and mini-batch sizes of 64, with distillation coefficients scheduled over 45 updates; other settings follow Appendix D. Figure 13 shows OPRD raising the Qwen3-1.7B student's AIME'24 Mean@16 from 10.0 to above 41 by update 150, versus about 25 for GRPO (35 at update 240). On Knights & Knaves, OPD collapses midway and recovers below the teacher; OPRD lifts the Qwen3-0.6B student above the Qwen3-8B-Base teacher's 64.5 Pass@1.



(a) AIME'24 (b) Knights & Knaves

Figure 13: **Learning curves for strong-to-weak distillation on two tasks.** We evaluate Qwen3-8B → Qwen3-1.7B on AIME'24 and Qwen3-8B-Base → Qwen3-0.6B on Knights & Knaves, using teachers from step 105 of task-specific GRPO. Gray dashed lines mark teacher performance. All other settings follow Appendix D, with method-specific distillation coefficients scheduled over the first 45 updates.

I Detailed Results for Weak-to-Strong Method Comparisons

I.1 Baseline Implementation Details

Under the default training and evaluation configurations in Appendices D and E, all baselines use the same model pairs, teacher checkpoints, prompt formats, and evaluation protocols as OPRD unless otherwise noted. We describe only their method-specific settings below.

- **W2SR-P** (Yuan et al., 2026). We reproduce the seeded prompt stream used by the 150-update RL runs, yielding $150 \times 64 = 9{,}600$ prompt occurrences. For each occurrence, we sample eight responses from the weak teacher and select one verifier-correct, format-valid, non-truncated response, discarding occurrences with no valid candidate. We then fully fine-tune the initial student checkpoint for three epochs using next-token prediction with a global batch size of 64 and a learning rate of 2×10^{-5}.
- **S2L-PO** (Ren et al., 2026). S2L-PO linearly anneals the fraction of weak-model rollouts over the first half of GRPO training. The original method uses a smaller *base* model as its frozen explorer to exploit policy-level diversity; for this comparison, we use the post-RL weak-teacher checkpoint as the frozen explorer. While the original implementation uses 16 rollouts per prompt, we retain its 16-phase schedule with the default group size of eight. Over 150 updates, the weak/student composition transitions from $8/0$ to $0/8$ during the first eight phases (updates 1–75) and remains at $0/8$ during the remaining eight phases (updates 76–150). For each trajectory, we compute the importance ratio using its generating policy as π_{rollout}, namely π_T for weak-teacher rollouts and $\pi_{\theta_{\text{old}}}$ for student rollouts. We also retain the original KL regularization toward the initial student with a coefficient of 10^{-3}.
- **OPSD** (Zhao et al., 2026). OPSD and SDPO (Hübotter et al., 2026) use correct self-generated rollouts as privileged information. To adapt it to our weak-to-strong setting, we instead use a verifier-correct weak-teacher rollout as privileged information, falling back to a correct student rollout when the weak teacher produces none. An EMA copy of the student serves as the self-teacher, conditioning on the privileged rollout to provide distillation targets for the original student trajectories and being updated after each step with a rate of 0.05. Whenever valid privileged information is available, we apply the distillation loss to all student trajectories in the group, regardless of whether they are correct or incorrect. The weak-teacher rollout supplies privileged context only: the generalized JSD remains between the student and the EMA self-teacher, with $\alpha = 0.5$ over the top-100 student tokens and an additional tail bucket.
- **Direct-OPD** (Feng et al., 2026). Developed concurrently with OPRD, Direct-OPD optimizes the weak policy shift as a dense reward on student-generated trajectories:

$$\mathcal{J}_{\text{Direct-OPD}} = \mathbb{E}_{x,\, y\sim\pi_\theta}\left[\sum_t \left(\log \pi_T(y_t \mid s_t) - \log \pi_T^{\text{ref}}(y_t \mid s_t)\right)\right] - \alpha D_{\text{KL}}(\pi_\theta \,\|\, \pi_{S,0})\,.$$

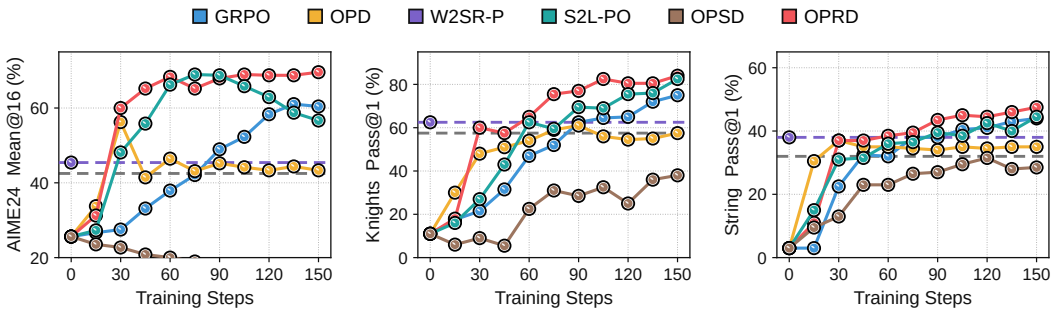
  Following the original implementation, we evaluate the dense reward over the top-16 tokens of the old student policy at each visited state and use the reported hyperparameters. The policy-shift scale is fixed at 1, while α, the coefficient of the KL anchor toward $\pi_{S,0}$, is initialized at 2.5. Before each actor update, α is multiplied by 1.01 or 0.99 depending on whether the batch-mean dense reward is positive or negative, respectively, and clipped to $[0.5, 2.5]$. This KL anchor is computed separately on the sampled response tokens using the low-variance k3 estimator.
- **W2S-OPD** (Yu et al., 2026). W2S-OPD reanchors the weak policy shift to the initial student by defining the proxy teacher as

$$\pi_{\text{proxy}}(v \mid s_t) \propto \pi_{S,0}(v \mid s_t)\left(\frac{\pi_T(v \mid s_t)}{\pi_T^{\text{ref}}(v \mid s_t)}\right)^{\gamma}.$$

  Following the original implementation, we set $\gamma = 1$ and compute the proxy scores over the full vocabulary before selecting the proxy's top-32 tokens. We normalize both the proxy and current-student distributions over this proxy-selected support and minimize the reverse KL from the current student to the proxy. This restricted-support reverse KL serves as the sole actor objective, with no additional KL anchor or adaptive coefficient.

I.2 Detailed Learning Curve

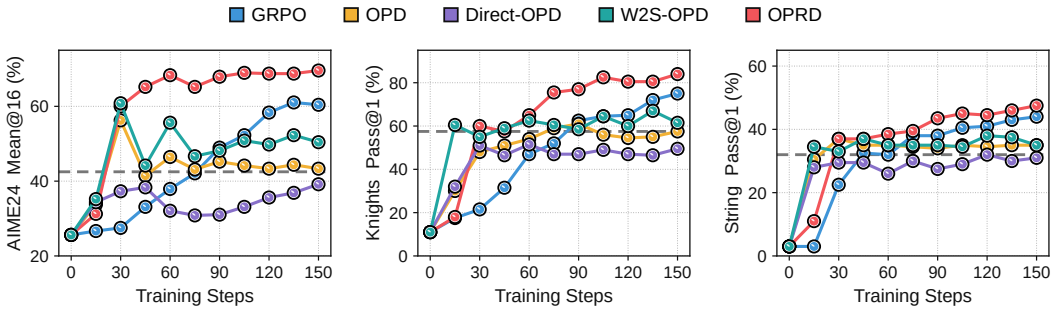
Figure 14 compares OPRD with three baselines using off-policy weak-teacher generations. OPRD has the strongest and most consistent gains. As in Yuan et al. (2026), W2SR-P's SFT on verifier-correct teacher rollouts moves the student slightly beyond weak-teacher performance. S2L-PO (Ren et al., 2026) remains competitive on both Reasoning Gym tasks, but its AIME'24 score declines after weak-teacher rollouts are fully annealed at update 75, and its checkpoint average stays below OPRD. OPSD (Zhao et al., 2026) performs poorly with either self-generated or weak-teacher privileged traces. This agrees with findings that privileged self-distillation can impair thinking models by shortening or suppressing deliberative reasoning (Kim et al., 2026; Kaur et al., 2026). OPSD provides a modest benefit only on String Manipulation, where rewards rise as reasoning shortens, and remains uncompetitive on the other tasks.



(a) AIME'24 (b) Knights & Knaves (c) String Manipulation

Figure 14: **Training dynamics of weak-to-strong methods using off-policy generations from the weak teacher.** Because W2SR-P performs SFT without subsequent RL, its final performance is shown as a horizontal dashed line. The gray dashed lines indicate weak-teacher performance. See Appendix I.1 for baseline implementation details.

Figure 15 compares OPRD with Direct-OPD (Feng et al., 2026) and W2S-OPD (Yu et al., 2026), concurrent methods using the weak policy delta. Their delta-defined objectives lack an independent verifier-driven update direction. W2S-OPD can surpass the weak teacher but plateaus near its performance because its target is limited to teacher-encoded policy changes. OPRD instead uses the delta to rescale only the verifier-gradient component aligned with the weak shift, preserving $\mathbf{g}_t^{\perp}$. The delta thus guides rather than replaces verifier-driven optimization, allowing OPRD to move beyond teacher-level saturation and lead on all three tasks.



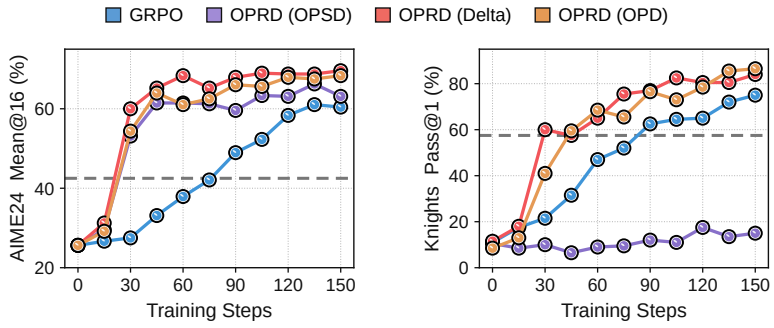
(a) AIME'24 (b) Knights & Knaves (c) String Manipulation

Figure 15: **Training dynamics of weak-to-strong methods using the weak policy delta.** The horizontal dashed lines indicate weak-teacher performance. See Appendix I.1 for baseline implementation details.

J Additional Results on Guidance-Direction Construction

OPRD requires a guidance direction that captures the reward-relevant change acquired by the teacher. Section 4.2 compares three constructions: the weak policy delta $\mathbf{\Delta}_t$ contrasts the post-trained teacher with its reference policy and isolates the change acquired during post-training; OPD contrasts the post-trained teacher with the current student, so its direction conflates the teacher's post-training update with the broader mismatch between the teacher's reference policy and the current student (i.e., $\mathbf{z}_T - \mathbf{z}_S = (\mathbf{z}_T - \mathbf{z}_T^{\text{ref}}) + (\mathbf{z}_T^{\text{ref}} - \mathbf{z}_S)$); and OPSD derives its direction from the discrepancy induced by a privileged teacher draft. The comparison uses the step-60 checkpoint from the weak teacher's GRPO run. Under this setting, the weak policy delta outperforms both alternatives by a wide margin.

However, OPD follows the gradient of a teacher-matching objective, its usefulness as a scaling direction should depend on teacher performance. We test this using the stronger teacher checkpoints adopted in our main experiments while keeping all other settings fixed (Figure 16). For mathematics, we use the step-75 teacher, which is already relatively strong. For Knights & Knaves, we use the step-105 teacher, which achieves 57.5% Pass@1 compared with 29.0% at step 60. With these teachers, the OPD direction performs well on both tasks, although it remains slightly behind the weak policy delta overall. OPSD is less consistent: it finishes above GRPO on AIME'24 but barely improves on Knights & Knaves.



(a) Math (b) Knights & Knaves

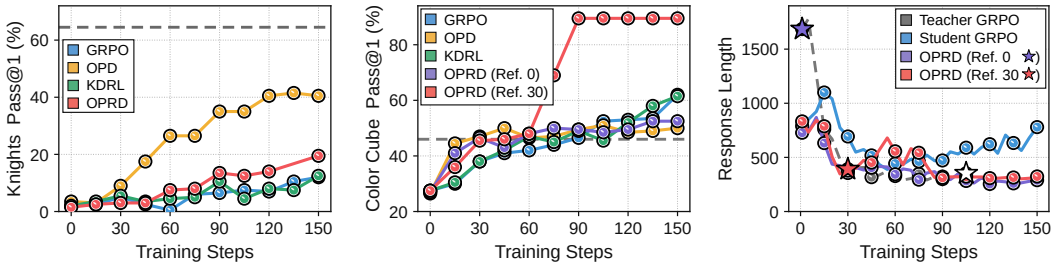
Figure 16: **Comparison of guidance-direction constructions.** The variants construct $\mathbf{d}_t$ from the teacher–reference policy shift (Delta), the teacher–student mismatch (OPD), or the privileged-context discrepancy (OPSD). Using stronger teacher checkpoints than those used in Figure 3a, we pair the step-75 Qwen3-4B teacher with a Qwen3-8B student for Math and the step-105 Qwen3-4B-Base teacher with a Qwen3-8B-Base student for Knights & Knaves. Gray dashed lines mark teacher performance. All other training configurations follow Appendix D.

These results suggest that the OPD gradient can provide a useful guidance direction when the weak teacher is sufficiently capable. In practice, however, the eventual performance gap between the weak teacher and the larger student cannot be known without fully training the student, making OPD difficult to adopt as a reliable default. OPSD is also less reliable because a privileged draft can constrain the student to a prescribed reasoning path (Kim et al., 2026; Kaur et al., 2026). Using its self-distillation gradient as $\mathbf{d}_t$ can then amplify verifier-gradient components aligned with this restrictive signal and hinder learning. The weak policy delta avoids both limitations because it compares the post-trained teacher only with its own reference policy. This isolates the change acquired during post-training without relying on either the evolving student or a privileged draft. We therefore retain the weak policy delta as our default guidance direction due to its stronger empirical performance and greater reliability in practice.

K Discussion of Key Challenges

K.1 Vanishing Policy Gradients

OPRD requires a nonzero verifier-driven policy gradient. In an additional strong-to-weak experiment pairing a Qwen3-8B-Base teacher with a Qwen3-1.7B-Base student, most Knights & Knaves responses are invalid, so most rollout groups receive identical rewards (i.e., the resulting group-relative advantages and their contributions to $\mathbf{g}_t$ therefore vanish). For these groups, the projection onto $\mathbf{d}_t$ also vanishes, leaving no component for OPRD to amplify and hence no teacher-guided correction. As shown in Figure 17a, OPRD still accelerates learning relative to GRPO and KDRL, although all three remain below 20% Pass@1, whereas OPD reaches 40.5% using dense policy-matching targets that do not depend on verifier rewards. This challenge arises from the student's initial rollout distribution rather than the absence of a useful teacher signal. Such an extreme regime is less likely in our primary weak-to-strong setting, where the student has greater capacity than the teacher, but may still arise on sufficiently difficult tasks. A short task-specific SFT or distillation warm-up could bootstrap valid on-policy behavior before switching to OPRD.



(a) Limited gradient signal (b) Task performance (c) Response Length

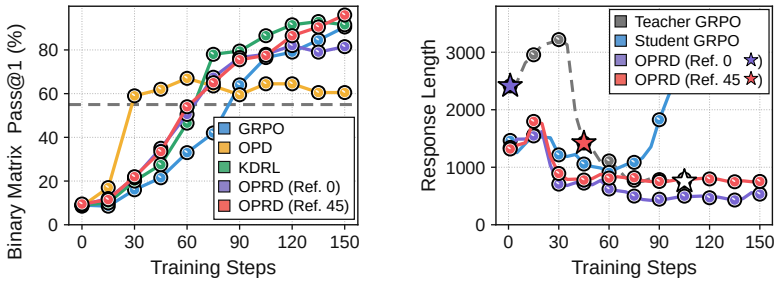
Figure 17: **(a) Results under limited policy-gradient signal.** On Knights & Knaves, we transfer a step-105 Qwen3-8B-Base teacher to a Qwen3-1.7B-Base student, with OPRD's negative-branch scale λ_t warmed up over the first 75 steps. **(b, c) Effect of reference-policy selection on length bias.** On Color Cube, we transfer a step-105 Qwen3-4B-Base teacher π_T to a Qwen3-8B-Base student, using either the step-0 or step-30 checkpoint from the same GRPO run as π_T^{ref}. OPRD's negative-branch scale λ_t is warmed up over the first 75 steps. The gray dashed line marks teacher performance, while the colored stars denote the mean response lengths of the two choices of π_T^{ref}, and the white star marks that of π_T. All other settings follow Appendix D.

K.2 Reference Policy Selection and Length Bias

As discussed in Section 2.2 and Appendix C, both $\mathbf{g}_t$ and $\mathbf{d}_t$ can contain reward-irrelevant components such as $\boldsymbol{\epsilon}_t$, which the projection-and-amplification step can magnify. Response length is one example: when it correlates with verifier reward, both signals can encode a preference for longer or shorter responses, even if changing length does not itself improve reasoning quality. As shown in Figures 17b and 17c, the step-0 reference π_T^{ref} produces substantially longer responses than the step-105 teacher π_T on Color Cube. The resulting shift $\boldsymbol{\Delta}_t$ therefore contains a strong shortening component. With this reference, OPRD rapidly shortens its responses and achieves strong early gains. It nevertheless plateaus at 52.5% Pass@1, below GRPO and KDRL, suggesting that the teacher-guided correction overemphasizes shortening at the expense of task-relevant reasoning. A simple mitigation is to move the reference to step 30, after the teacher's initial length collapse. This excludes some of the teacher's early gains from $\boldsymbol{\Delta}_t$ but substantially narrows the reference–teacher length gap and weakens the associated bias. OPRD then avoids the plateau and jumps to 89.5%, discovering a more effective reasoning strategy.

Binary Matrix provides another instance of this behavior. The teacher's mean response length falls sharply between steps 30 and 45 and then stabilizes. We therefore compare step-0 and step-45 choices of π_T^{ref}: the former includes a large shortening component in $\boldsymbol{\Delta}_t$, whereas the latter excludes most of it. Figure 18 compares both OPRD variants with GRPO, OPD, and KDRL. Both initially improve faster than GRPO but diverge after step 90. With the step-0 reference, the student's responses continue to shorten and Pass@1 plateaus at 81.5%, below GRPO and KDRL, consistent with the correction

overemphasizing length reduction. With the step-45 reference, response length does not exhibit the same continued decline and Pass@1 reaches 96.0%. As on Color Cube, placing the reference after the sharp length transition mitigates this bias while retaining the teacher's later task progress. However, changing the reference modifies $\mathbf{\Delta}_t$ as a whole rather than isolating its length-related component. Disentangling structured bias from task-relevant guidance therefore remains an open question.



(a) Task performance (b) Response Length

Figure 18: **Effect of reference-policy selection on length bias in Binary Matrix.** We transfer a step-105 Qwen3-4B-Base teacher π_T to a Qwen3-8B-Base student, using either the step-0 or step-45 checkpoint from the same GRPO run as π_T^{ref}. OPRD's negative-branch scale λ_t is warmed up over the first 75 steps. The gray dashed line marks teacher performance, while the colored stars denote the mean response lengths of the two choices of π_T^{ref}, and the white star marks that of π_T. All other settings follow Appendix D.

L Detailed Analysis of Student Behavior

L.1 Token Alignment Analysis

We analyze the correct AIME'25 response shown in Figure 4a using the OPRD-trained Qwen3-8B student at update 150 and the Qwen3-4B teacher at GRPO update 75. We compute the policy gradient $\mathbf{g}_t$ assuming a single correct rollout with $A_t = 1$, since the magnitude of a positive advantage does not affect cosine similarity. Each token is colored by $\cos(\mathbf{d}_t, \mathbf{g}_t)$, with green indicating positive alignment and red indicating negative alignment. We compare the original continuation with an alternative generated by the same student checkpoint, keeping the selected prefix fixed and forcing the next token to be **5**, the top-ranked token under $\mathbf{d}_t$.

L.2 Response Style Analysis

We evaluate Qwen3-8B students trained with OPD and OPRD on DAPO-Math-17K at updates 30, 60, 90, 120, and 150. For each checkpoint, we generate 16 responses to each of the 30 AIME'24 problems, yielding 480 responses. We use two fixed references: the Qwen3-4B teacher at GRPO update 75 and a separately GRPO-trained Qwen3-8B student at update 150. Each response is represented by 101 style features across five categories: connectives (15), modality (10), grammar (64), punctuation (8), and sentence and paragraph structure (4). The first three categories measure relative frequencies of function words, including connectives, modal and negation words, and grammatical words such as pronouns and articles. Punctuation features count occurrences per 1,000 words, while structure features capture the mean and standard deviation of words per sentence and sentences per paragraph. We standardize the features at every checkpoint of each method using a shared mean and standard deviation for each feature, computed from the 960 responses of the two reference models.



(a) Overall style similarity

(b) Style similarity by category

Figure 19: **(a) Comparing response style distributions over training.** Energy-distance differences are averaged across 30 AIME'24 problems, with positive values indicating greater similarity to the GRPO student and negative values to the teacher. Shading shows 95% confidence intervals from 2,000 bootstrap resamples of the problems. **(b) Measuring similarity to teacher and student response styles.** Normalized distance differences compare average styles in five categories, with red indicating greater similarity to the teacher and green to the GRPO student. Both panels use fixed references: the Qwen3-4B teacher at GRPO update 75 and the Qwen3-8B GRPO student at update 150.

Figure 19a compares response style distributions using all 101 standardized features. For this comparison, we combine connectives, modality, and grammar into a group of 89 function-word features and scale the function-word, punctuation, and structure coordinates by $1/\sqrt{89}$, $1/\sqrt{8}$, and $1/\sqrt{4}$, respectively, to balance the three groups' contributions. For each problem, let X_k, X_T, and X_S denote the sets of 16 response feature vectors from the evaluated checkpoint at update k, the teacher, and the GRPO student, respectively. We compute the energy-distance difference

$$\Delta_{\mathrm{ED}} = \mathrm{ED}(X_k, X_T) - \mathrm{ED}(X_k, X_S).$$

Energy distance measures differences between distributions by accounting for both between-set distances and within-set variation. We average Δ_{ED} across the 30 problems, with positive values indicating greater similarity to the GRPO student and negative values to the teacher. Shading shows

95% confidence intervals from 2,000 bootstrap resamples of the problems. OPD shifts toward the teacher over training, with its mean difference decreasing from $+0.135$ at update 30 to -0.190 at update 150. OPRD remains closer to the GRPO student at every evaluated checkpoint, with its mean difference increasing from $+0.316$ to $+0.447$ over the same period.

Figure 19b compares the same checkpoints separately across the five style categories. For each category c, we average the standardized features across all 480 responses without additional feature-group scaling. Let $\boldsymbol{\mu}_{k,c}$, $\boldsymbol{\mu}_{T,c}$, and $\boldsymbol{\mu}_{S,c}$ denote these average vectors for the evaluated checkpoint at update k, the teacher, and the GRPO student, respectively. We compute

$$s_c(k) = \frac{\|\boldsymbol{\mu}_{k,c} - \boldsymbol{\mu}_{T,c}\|_2 - \|\boldsymbol{\mu}_{k,c} - \boldsymbol{\mu}_{S,c}\|_2}{\|\boldsymbol{\mu}_{T,c} - \boldsymbol{\mu}_{S,c}\|_2}.$$

The score measures the difference in distances to the two reference averages, normalized by their separation. Scores range from -1 to $+1$, with negative values indicating greater proximity to the teacher, positive values to the GRPO student, and zero indicating equal distance. OPD is closer to the GRPO student in all five categories at update 30 but closer to the teacher in all five by update 150. OPRD remains closer to the GRPO student in every category at all evaluated checkpoints, consistent with the overall distribution comparison.

M Reverse Distillation without Policy Matching

Ironically, OPRD performs reverse distillation without a distillation objective. Knowledge still passes from a weaker teacher to a stronger student, but through guidance for the student's own policy optimization rather than imitation of the teacher's policy. OPRD uses the teacher's learned change as a guiding axis, while the student continues to optimize its own reward.

Guided ES (Maheswaranathan et al., 2019) similarly uses imperfect surrogate gradients to guide a search rather than following them directly. It allocates more search variance to the subspace spanned by these gradients and estimates updates from evaluations of the original objective at perturbed points. FOSI (Sivan et al., 2024) uses curvature information to update selected Hessian eigenvector directions while applying a base optimizer to their orthogonal complement. These methods use different mechanisms, but both exploit selected directions while retaining the underlying objective.

For OPRD, the directional information comes from learning already carried out by a weaker model. The distinction matters because the usefulness of that learning is not the same as the quality of the teacher's final policy. A teacher can acquire useful task structure even when a stronger student can ultimately find better solutions. Reverse distillation retains this source of knowledge while leaving policy improvement to the student's own reward objective. Progress acquired at smaller scale can therefore support learning at larger scale without making the smaller model's behavior the endpoint.

N FUTURE WORKS

Broader Tasks and Settings. Mathematics, logical reasoning, and code generation provide controlled settings with relatively clear verification signals. To assess the generality of OPRD, future work should examine tasks with different forms of feedback and sequential interaction. Agentic environments are particularly informative: early actions shape subsequent observations and decisions, while rewards may depend on the entire trajectory. Evaluations with longer interactions and less structured feedback would help establish how broadly OPRD transfers useful policy improvements.

Scaling to Larger Models. Our results cover models from 0.6B to 14B parameters and both weak-to-strong and strong-to-weak capacity orderings. At larger scales, OPRD may be especially useful because learning a policy shift with a smaller model could be substantially cheaper than optimizing the larger model directly from verifier feedback. Larger-scale experiments would test how students with greater capacity use the same teacher shift, how far they can improve beyond the teacher, and whether the gains in update efficiency persist as post-training costs increase.

Systems Considerations at Scale. OPRD adds no-gradient forward passes through the frozen teacher and reference policies and a correction of the student's logit gradient. In weak-to-strong setup, both frozen policies are smaller than the student and require neither generation nor backward propagation, while the correction retains one additional dense logit-gradient tensor. As shown in Appendix O, these additions only increase wall-clock time by 11.9% and peak GPU memory by 10.2% relative to GRPO. But at frontier-model scale, keeping this overhead modest will require efficient placement, sharding, and scheduling of the frozen policies within hybrid parallelism, together with communication-efficient correction across model and vocabulary shards.

Extending OPRD to Pretraining. Although our experiments focus on verifier-driven post-training, the core gradient-scaling operation requires only a token-level student gradient and a direction extracted from a weaker model. This suggests that OPRD could also be adapted to pretraining: the policy shift between two checkpoints of a smaller model could guide a larger student's next-token prediction updates while the student continues to optimize its own language-modeling objective rather than matching the smaller model's distribution. Evaluating this extension in both initial and continued pretraining would test whether directional transfer improves training efficiency across model scales and data distributions.

Toward Recursive Self-Improvement. An important direction for future work is to connect weak-to-strong distillation with recursive self-improvement, where each model generation contributes to the development of more capable successors through training supervision, evaluation, and algorithmic improvements. These successors, in turn, use their greater capabilities to improve subsequent model development. For example, earlier models helped supervise GPT-6 Astra's training (OpenAI, 2026), while Google reports using agentic loops to recursively evaluate and refine Gemini 3.8 Flash (Gemini Team, 2026). A promising extension is to incorporate reverse distillation into these workflows, allowing earlier generations to contribute not only to training supervision and development but also directly to their successors' policy updates through their post-training policy shifts. Building such pipelines would allow us to test whether reverse distillation can consistently improve sample efficiency and accelerate training as each successor becomes a teacher for the next generation.

O Computational Cost and Memory Usage

Benchmark Setup. To isolate method-specific training overhead from response-length differences, we force every generated response to contain exactly 16,384 tokens by ignoring EOS. Both benchmarks use the mathematics setting of Section 3.1, with a Qwen3-8B student and the Qwen3-4B teacher checkpoint at update 75. OPRD additionally loads the Qwen3-4B base policy as its reference. Within each benchmark, all methods receive prompts in the same order and use the same random seed. Wall-clock timing uses 64 prompts, whereas memory profiling uses 8 prompts, with 8 rollouts per prompt in both cases. All runs execute on four NVIDIA B200 GPUs with DP4, rollout TP1, BF16, Flash Attention 2, padding removal, and gradient checkpointing. Optimization uses a global minibatch of 64 responses and one PPO epoch per training step. Dynamic token batching caps each GPU at 36,864 tokens, which yields two complete samples per microbatch under the fixed-length setting. The rollout engine uses `gpu_memory_utilization=0.60` and `max_num_seqs=128`. Sampling uses temperature 1.0, top-p 1.0, and no top-k truncation. To keep reward-side computation identical, we replace task-specific reward evaluation with deterministic alternating binary rewards within each prompt group. Validation, periodic model saving, external logging, and all non-training diagnostics are disabled.

Wall-Clock Time. We run each method in a fresh process, discard one complete warm-up step, and report the mean and sample standard deviation over the following four steps. As shown in Table 12, rollout generation is the largest component of each training step, taking roughly 602 seconds and accounting for 61.6% of the total GRPO time, with small differences across methods attributable to run-to-run variation. OPD and KDRL, each of which evaluates one frozen teacher, incur total overheads of 7.2% and 7.9% over GRPO, respectively. OPRD evaluates the teacher and its reference sequentially, increasing frozen-forward time from 60.46 seconds for OPD to 108.80 seconds. Because rollout generation dominates the step, this additional reference evaluation increases total time by only 4.4% over OPD, resulting in an overall overhead of 11.9% relative to GRPO. Student-forward time is effectively unchanged, while the update containing the teacher-direction projection and scaling increases by just 2.79 seconds over GRPO, equivalent to 0.25% of the full OPRD step. Beyond the teacher evaluation already required by OPD and KDRL, nearly all of OPRD's additional runtime therefore comes from evaluating the reference policy.

Table 12: **Wall-clock time per training step.** All methods process the same 64 prompts with 8 rollouts per prompt, with every response fixed at 16,384 tokens to equalize the number of generated tokens. Total time is reported as the mean ± sample standard deviation, while the component columns report their means. Rollout includes the complete generation call and the actor-to-rollout mode transition. Student forward is a no-gradient pass that recomputes the old log probabilities of the sampled tokens. The frozen-model column reports no-gradient teacher evaluation for OPD and KDRL and sequential teacher and reference evaluations for OPRD, while GRPO requires neither. Update includes a separate gradient-enabled student forward pass, backward propagation, and the optimizer step, excluding the separately timed frozen-model evaluations. Etc. includes reward construction, advantage computation, batch assembly and balancing, orchestration, and residual boundary costs.

	Total		Rollout	Model Forward		Optimization	
Method	Time (s/step)	Overhead	Student	Student	Teacher + Ref	Update	Etc.
Qwen3-4B (Teacher) → Qwen3-8B (Student)							
GRPO	977.72 ± 16.83	–	602.20	70.49	0.00	304.00	1.03
OPD	1048.00 ± 17.49	+7.2%	610.74	70.38	60.46	305.38	1.04
KDRL	1055.41 ± 15.94	+7.9%	618.20	70.08	60.17	305.92	1.05
OPRD	1094.44 ± 16.17	+11.9%	607.39	70.40	108.80	306.79	1.06

In our primary weak-to-strong setting, the teacher is smaller than the student, keeping the cost of maintaining and evaluating its reference policy relatively low. In strong-to-weak settings, however, retaining both a large teacher and its equally sized reference can impose substantial memory and computational costs. Since the teacher is already more capable than the student, one potential alternative is to use the OPD gradient directly as the guidance direction, as explored in Figure 16, eliminating the need for a separate reference model. Another option is to replace the teacher's reference with the student's smaller frozen reference policy. Although these alternatives no longer isolate the teacher's RL-induced policy shift or retain its corresponding implicit-reward interpretation,

they could still exploit the strong teacher's token preferences to guide verifier-driven student optimization. We leave their evaluation to future work.

Peak GPU Memory. We measure peak GPU memory during rollout generation and the actor update, while separately recording the frozen-policy evaluation performed within the update. We also report the overall maximum observed during the complete training step. As shown in Table 13, OPRD carries a nearly constant additional footprint throughout training: approximately 14 GiB per GPU relative to GRPO and 13 GiB relative to OPD and KDRL. The two largest identifiable memory requirements within OPRD are the 3.75 GiB frozen teacher and reference parameter shards and a transient 9.43 GiB dense corrected-gradient allocation within the correction hook. Of the 3.75 GiB in frozen-model parameters, 1.87 GiB is additional relative to OPD and KDRL, which already retain the teacher. The 9.43 GiB hook allocation is also specific to OPRD. Because the absolute NVML peaks additionally include shared model and optimization state, allocator caches, and CUDA and distributed runtime state, these quantities identify the main OPRD-specific allocations but do not provide an exact additive decomposition of the observed peak difference.

The overall maximum occurs during rollout for every method. OPRD reaches 151.27 GiB, exceeding GRPO by 13.98 GiB (10.2%) and OPD and KDRL by 13.21 GiB. Rollout itself increases memory by approximately 51–52 GiB for all four methods. The difference is already present before generation, where OPRD begins the measured step at 99.97 GiB, 14.93 GiB above GRPO and 13.25 GiB above OPD and KDRL. OPRD's higher rollout peak therefore results from adding essentially the same generation-time allocation to a higher starting footprint, rather than from rollout requiring more memory.

Frozen-policy evaluation is performed within the broader actor-update interval, and their maximum values coincide in our measurements. OPRD reaches 115.19 GiB during both frozen-policy evaluation and the full update, exceeding OPD and KDRL by 13.15 GiB. During the update, it also exceeds GRPO by 14.84 GiB. These differences closely match those observed before and during rollout, indicating that neither frozen-policy evaluation nor the correction introduces a separate phase-specific increase in the device-memory peak. Within the correction hook, PyTorch-allocated memory grows by 9.43 GiB, matching the largest dense BF16 corrected-gradient tensor. By comparison, the sparse support formed by the sampled action and the student's top-10 tokens occupies at most 1.38 MiB, and direct gather and sparse scatter avoid an additional 9.27 GiB response-by-vocabulary copy. The hook allocation is already contained within the 115.19 GiB update peak, which remains well below the overall maximum during rollout.

Table 13: **Peak GPU memory per training step.** We profile 8 prompts with 8 rollouts per prompt and fix every response at 16,384 tokens. Each method is evaluated in four independent trials, each launched in a fresh process on four NVIDIA B200 GPUs with DP4/TP1 and BF16. Each trial discards one complete warm-up step and measures the following step. Whole-device NVML memory is sampled every 100 ms, and each Peak entry reports the mean across trials of the maximum usage over time and across the four GPUs within the indicated interval. All values are in GiB per GPU. Because each Peak entry represents the worst-GPU peak, multiplying it by four provides only a rough upper bound on aggregate device memory. Overall is the maximum over the complete step. Under Rollout, Peak − Start is the increase from the step-start baseline to the rollout peak. Teacher + Ref. Peak reports the maximum during frozen-model evaluation, covering one teacher evaluation for OPD and KDRL and sequential sequential teacher and reference evaluations for OPRD. This evaluation is a subinterval of Actor Update. Params. reports the calculated lower bound for the total BF16 parameter shards of the resident frozen models. Actor Update Peak includes resident models, gradients, optimizer states, activations, and runtime buffers, while Hook Growth reports the additional PyTorch allocation during the OPRD correction.

	Overall		Rollout		Teacher + Ref		Actor Update	
Method	Peak	Overhead	Peak	Peak − Start	Peak	Params	Peak	Hook Growth
Qwen3-4B (Teacher) → Qwen3-8B (Student)								
GRPO	137.29	–	137.29	52.25	–	–	100.35	–
OPD	138.06	+0.6%	138.06	51.34	102.04	1.87	102.04	–
KDRL	138.06	+0.6%	138.06	51.34	102.04	1.87	102.04	–
OPRD	151.27	+10.2%	151.27	51.30	115.19	3.75	115.19	9.43