WHY ON-POLICY DISTILLATION SOMETIMES FAILS: VANISHING LEARNING SIGNALS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) enables effective capability transfer between language models, yet the mechanisms underlying its failures are not fully understood. Across code generation and mathematical reasoning, OPD with larger-scale teachers exhibits early loss plateaus, with an average final loss reduction of 25.1% after 200 updates, compared with 96.2% for self-RL teachers, obtained by further reinforcement learning (RL) training of the initial student. To understand this difference, we analyze OPD as an idealized continuous-time dynamical system in the small-learning-rate limit. Our diagnostics link these plateaus to premature learning-signal collapse, where the gradient becomes weak relative to the remaining loss. We further prove a local recovery guarantee for teachers sufficiently close to the initial student in a shared parameterization under regularity conditions, offering a conditional explanation for the success of self-RL teachers in our experiments. Across runs with and without loss plateaus, we observe small relative parameter changes (0.025–0.098%) and high similarity between the student's representations before and after OPD (linear CKA > 0.98 across layers). These observations suggest that limited representation adaptation may contribute to learning-signal collapse, a hypothesis that remains to be tested.

1 INTRODUCTION

On-policy distillation (OPD) has become an important component of language-model post-training (Yang et al., 2025; Xiao et al., 2026; Lu & Thinking Machines Lab, 2025). In OPD, the student generates its own responses and is trained to match the teacher's next-token distributions along these responses (Agarwal et al., 2024). This provides dense supervision on the prefixes the student actually visits, reducing the train–inference distribution mismatch that motivates on-policy imitation learning (Ross et al., 2011; Agarwal et al., 2024). This approach has shown promise for transferring capabilities between language models (Agarwal et al., 2024; Gu et al., 2024; Yang et al., 2025; Xiao et al., 2026). Despite these successes, recent studies have observed that OPD does not consistently improve student performance, even when the teacher substantially outperforms the student (Li et al., 2026). These findings suggest that teacher capability alone is insufficient to ensure effective transfer, motivating a closer examination of when and why OPD fails.

To investigate these failures, we examine OPD training dynamics across code generation and mathematical reasoning (Section 4). We observe that *self-RL teachers*, obtained by further reinforcement learning (RL) training of the initial student, support sustained loss reduction and substantial improvements in student performance. By contrast, OPD with the larger-scale teachers in our experiments exhibits early loss plateaus and more limited performance gains (Figure 1). In these runs, the distillation loss stops decreasing while substantial teacher–student mismatch remains. Related observations have also been reported elsewhere: Li et al. (2026) observe weak gradients and persistent token-probability discrepancies in a failing run, while Jiang et al. (2026) report persistent high-loss tokens after the aggregate OPD objective saturates. Together with our results, these reports indicate that limited optimization progress despite unresolved disagreement occurs across multiple OPD settings. This raises a central question:

Why does OPD stop reducing its loss before the student matches the teacher?

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.

Prior work has proposed several explanations for why OPD does not always improve student performance (Li et al., 2026; Fu et al., 2026a; Wang et al., 2026a). These include teacher–student incompatibility (Li et al., 2026; Wang et al., 2026b), unreliable teacher guidance on student-generated prefixes (Fu et al., 2026a; Zhu et al., 2026), limitations of sampled-token supervision (Fu et al., 2026a; Jia et al., 2026), and imperfect alignment between distillation updates and task success (Wang et al., 2026a; Armandpour et al., 2026). These studies show that a strong teacher does not always provide feedback that helps the student learn. However, these explanations do not fully clarify why the OPD loss keeps decreasing with some teachers but plateaus early with others.

We address this problem in the small-learning-rate limit (Borkar & Meyn, 2000). In this setting, we model OPD as an idealized continuous-time dynamical system and analyze its loss dynamics (Section 5.1). Our main contributions are as follows:

- We identify a recurring failure mode of OPD across models and tasks: training stalls while much of the distillation loss remains. This pattern occurs with the larger-scale teachers in our experiments (Section 4).
- Our diagnostics link the observed loss plateaus to *premature learning-signal collapse*, where the raw gradient weakens relative to the remaining loss before teacher–student disagreement is resolved (Section 5.2).
- We prove a local recovery guarantee for OPD with teachers sufficiently close to the initial student in a shared parameterization (Section 5.3). This result offers a conditional theoretical explanation for the success of OPD in our experiments when self-RL teachers are used.
- Across runs with and without loss plateaus, we observe small relative parameter changes and high similarity between the student's representations before and after OPD (Section 6). These observations may help explain why OPD succeeds with nearby teachers but stalls when matching the teacher requires changes that its local updates do not make.

2 Related Work

On-policy distillation and its failures. OPD trains students on their own rollouts, extending on-policy imitation learning (Ross et al., 2011) to language models (Lin et al., 2020; Agarwal et al., 2024; Gu et al., 2024; Ko et al., 2024). GKD matches teacher and student next-token distributions on student-generated responses (Agarwal et al., 2024), while MiniLLM minimizes sequence-level reverse KL using policy gradients (Gu et al., 2024). OPD is used in LLM post-training (Yang et al., 2025; Lu & Thinking Machines Lab, 2025) and has been interpreted as dense KL-constrained RL (Yang et al., 2026) and online imitation learning (Zhang et al., 2026). However, a stronger model is not always a better teacher. In classical distillation, students can learn less from much larger teachers (Cho & Hariharan, 2019; Mirzadeh et al., 2020) and fail to match them despite sufficient capacity (Stanton et al., 2021). In OPD, Li et al. (2026) emphasize alignment between teacher and student thinking patterns and observe small gradients in failing runs; Wang et al. (2026b) find that teacher supervision is learnable mainly where the teacher assigns probability mass to the student's top candidate tokens. Other studies link OPD failures to imbalanced token-level supervision (Fu et al., 2026a; Jia et al., 2026), teacher signals misaligned with task success (Wang et al., 2026a; Armandpour et al., 2026), and progressively slower learning from supervision (Fu et al., 2026b). In RL fine-tuning, low reward variance can lead to small expected gradients (Razin et al., 2024), and a more accurate reward model need not be a better teacher (Razin et al., 2025). These findings motivate our population-level analysis of how the OPD loss evolves as the student and its rollout distribution change (Section 5.1).

Training dynamics and geometry of post-training. Continuous-time dynamics can approximate gradient descent with small step sizes (Elkabetz & Cohen, 2021), while Polyak–Łojasiewicz-type conditions yield convergence guarantees for nonconvex optimization, including policy gradient methods (Karimi et al., 2016; Mei et al., 2020). Post-training often induces limited model changes. Language-model fine-tuning can be well described by kernel dynamics (Malladi et al., 2023). Studies of RL post-training report proximity to the base model in KL divergence (Shenfeld et al., 2026), updates that avoid principal weight directions (Zhu et al., 2025), preserved hidden representations (Runwal et al., 2026), and many unchanged weights at BF16 precision (Mukherjee et al., 2025). In OPD, updates quickly concentrate in a low-dimensional subspace (Shen et al., 2026). Cai et al.

(2026) explain this early low-rank structure using a local quadratic model around the initial student with fixed contexts. We study how teacher proximity shapes OPD loss by expanding around the teacher under a changing rollout distribution (Section 5.3) and compare parameter and representation changes under OPD and RL post-training (Section 6).

3 PRELIMINARIES

3.1 ON-POLICY DISTILLATION

Notation. A prompt $x \sim \mathcal{D}$ is followed by a response $y = (y_1, \ldots, y_{|y|})$ of tokens from a vocabulary $\mathcal{V}$. The prefix state at position t is $s_t = (x, y_{<t})$ and the action is the next token $a_t = y_t$, so a policy π factorizes as $\pi(y \mid x) = \prod_{t=1}^{|y|} \pi(a_t \mid s_t)$. The student policy π_θ has parameters θ; the teacher policy is π_T.

On-policy distillation. OPD minimizes the reverse KL between the student's and the teacher's next-token distributions on the prefixes of the student's own sampled responses,

$$\widehat{\mathcal{L}}(\theta; \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s) \,\|\, \pi_T(\cdot \mid s)\big), \quad \mathcal{L}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta} D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s) \,\|\, \pi_T(\cdot \mid s)\big), \tag{1}$$

where $\mathcal{S}$ is a set of prefixes, and a prefix $s = (x, y_{<t})$ is distributed by drawing $x \sim \mathcal{D}$, sampling $y \sim \pi_\theta(\cdot \mid x)$, and taking the position t uniformly from $1, \ldots, |y|$. In $\widehat{\mathcal{L}}$ the set $\mathcal{S}$ collects the prefixes of the responses sampled in a minibatch.

In practice, the sum over the full vocabulary in the KL is replaced by a top-K version. At each prefix s, we retain only the K tokens with the highest student probabilities, denoted by $\mathcal{V}_K(s)$, and renormalize the student distribution over this set to obtain $\tilde{\pi}_\theta$. We then average the log-ratio under this renormalized distribution:

$$\sum_{a \in \mathcal{V}_K(s)} \tilde{\pi}_\theta(a \mid s) \big[\log \pi_\theta(a \mid s) - \log \pi_T(a \mid s)\big]. \tag{2}$$

The full KL needs the teacher's log-probabilities for all $|\mathcal{V}|$ tokens at every prefix, whereas Eq (2) needs them for top K tokens and drops the tail of the distribution. We use a top-K approximation to the reverse KL loss with $K = 16$ during training.

4 EARLY LOSS PLATEAUS IN ON-POLICY DISTILLATION

We compare OPD with different teachers across code generation and mathematical reasoning. In our experiments, OPD with larger-scale teachers exhibits early loss plateaus while substantial teacher–student performance gaps remain. By contrast, self-RL teachers support sustained loss reduction and larger performance gains.

Table 1: Student–teacher pairs in the seven main OPD runs. Full model names and training provenance are given in Appendix E.

Task	Student	Teacher
Code	Qwen3-4B (Non-thinking)	Qwen3-4B-RL-Code
		Qwen3-14B (Non-thinking)
Math	Qwen3-1.7B (Non-thinking)	Qwen3-4B-RL-Math
		Qwen3-4B (Non-thinking)
Math	R1-Distill-1.5B	JustRL-1.5B
		Skywork-7B
		R1-Distill-7B

4.1 Experimental Setup

Models. We study three student settings spanning code generation and mathematical reasoning, with seven student–teacher pairs listed in Table 1. Details of the models are given in Appendix E. We use *self-RL teachers* to refer specifically to teachers obtained by further RL training of the initial student, namely Qwen3-4B-RL-Code and JustRL-1.5B, and *larger-scale teachers* to refer to teachers with more parameters than the student, namely Qwen3-14B, Qwen3-4B-RL-Math, Qwen3-4B, Skywork-7B, and R1-Distill-7B.

Datasets. For training, we use Eurus-2-RL-Data (Cui et al., 2025) for code generation and DAPO-Math-17K (Yu et al., 2025) for mathematical reasoning.

Training. We implement OPD in veRL (Sheng et al., 2025) and optimize the top-K objective in Eq (2) with $K = 16$ for the seven main runs. These runs use one rollout per prompt and 64 prompts per update. We train for 200 steps on both code generation and mathematical reasoning. We use AdamW with a constant learning rate of 10^{-6}, a sampling temperature of 1 for both the student and teacher, and a gradient clipping norm of 1. The remaining hyperparameters are reported in Table E.2.

Evaluation. We evaluate code on HumanEval+, MBPP+ (Liu et al., 2023), and LiveCodeBench (Jain et al., 2025), and mathematics on AIME 2024, AIME 2025, and AMC 2023. For each benchmark, we sample k responses per problem and report $\mathrm{avg@}k$, the fraction of correct responses across all sampled responses and problems, using $k = 4$ for code and $k = 8$ for mathematics. We cap responses at 16,384 tokens for code and 31,744 tokens for mathematics. Task-level validation accuracy is the unweighted mean of the three benchmark accuracies, ensuring that each benchmark contributes equally regardless of its number of problems.

4.2 Evaluation Metrics

Accuracy gain and gap recovery. Let $\mathrm{Acc}_{\mathrm{start}}$ be the validation accuracy of the base student model before any OPD updates (step 0), $\mathrm{Acc}_{\mathrm{end}}$ its accuracy at the final comparison step, and $\mathrm{Acc}_{\mathrm{teacher}}$ the teacher's validation accuracy. All accuracies are expressed as percentages (%). We define the student's accuracy gain as $\mathrm{Gain} = \mathrm{Acc}_{\mathrm{end}} - \mathrm{Acc}_{\mathrm{start}}$, reported in percentage points (pp). Gap recovery is the percentage of the initial teacher–student accuracy gap closed by OPD:

$$\mathrm{GapRecovery} = \frac{\mathrm{Acc}_{\mathrm{end}} - \mathrm{Acc}_{\mathrm{start}}}{\mathrm{Acc}_{\mathrm{teacher}} - \mathrm{Acc}_{\mathrm{start}}} \times 100\%.$$

4.3 Experimental Results

The OPD loss plateaus with larger-scale teachers while the gap remains large. As Figure 1 shows, with larger-scale teachers, the OPD loss stops decreasing early in training, then stays roughly flat in the code run and rises again in the math runs. Table 2 quantifies this plateau: the maximum observed loss reduction in these five runs is only 30.4%–65.1%, compared with 96.3%–99.5% for self-RL teachers. The corresponding final reductions average 25.1% and 96.2%, respectively (Appendix E). The student's validation accuracy plateaus at the same time. After 200 updates, it has recovered only 13.0%–47.5% of the teacher–student performance gap, leaving at least half of the gap unclosed. With a larger-scale teacher, OPD can stop improving the student while much of the accuracy gap remains and the reverse-KL distance to the teacher is still large.

Self-RL teachers drive the OPD loss to near zero and recover most of the gap. Figure 1 shows that OPD follows a different training dynamic with self-RL teachers than with larger-scale teachers. With self-RL teachers, the OPD loss continues to decrease rather than plateau. As shown in Table 2, the maximum observed loss reduction reaches 99.5% for Qwen3-4B-RL-Code and 96.3% for JustRL-1.5B. In the same runs, the student closes 77.4% and 70.7% of the accuracy gap to the teacher, respectively. These results suggest that students can approach self-RL teachers in reverse-KL distance during training and learn transferable abilities from them.

SFT does not resolve the failure, even when followed by OPD. Li et al. (2026) use SFT on teacher responses before OPD to recover failing OPD. In our experiments, SFT alone does no better

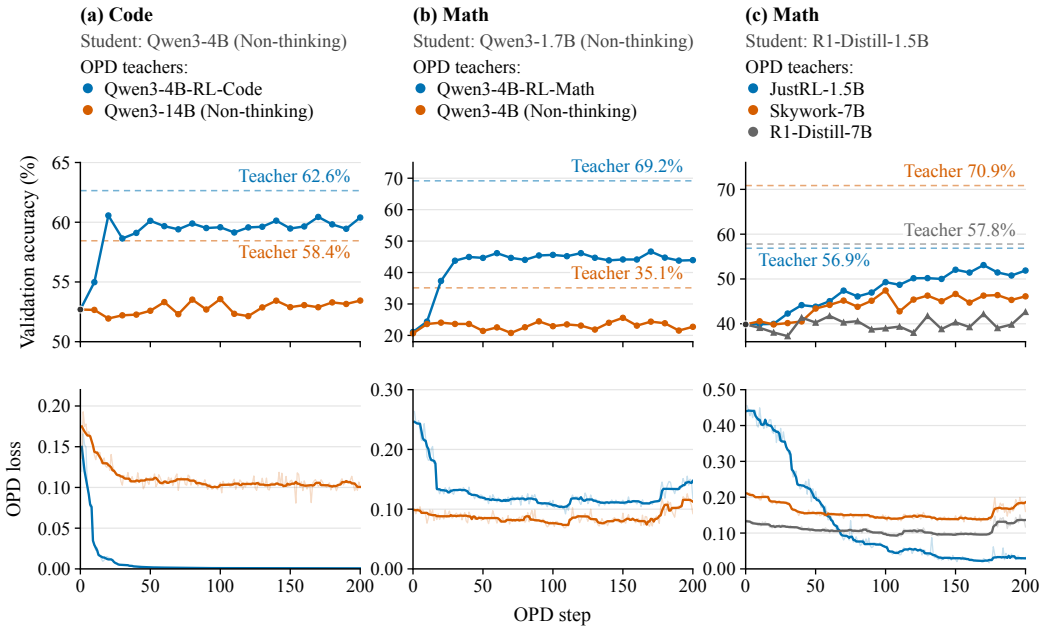


Figure 1: Validation accuracy (top) and OPD loss (bottom) over 200 updates. The black point in panels (a) and (c) is the shared initial-student evaluation. Dashed lines mark teacher accuracies. Loss curves are centered 9-step rolling medians; raw losses are shown faintly. Qwen3 models use non-thinking mode.

Table 2: OPD outcomes over 200 updates: step-1 and minimum loss, loss reduction $(1 - \widehat{\mathcal{L}}_{\min}/\widehat{\mathcal{L}}_{\text{start}}) \times 100\%$ from unrounded losses, accuracy gain from the base student at step 0, and gap recovery (Section 4.2). Bold marks within-group maxima; † marks self-RL teachers.

Student	Teacher	$\widehat{\mathcal{L}}_{\text{start}}$	$\widehat{\mathcal{L}}_{\min}$	Loss reduction	Gain (pp)	Gap recovery
Code Qwen3-4B (Non-thinking)	Qwen3-4B-RL-Code†	0.120	0.001	**99.5%**	**+7.70**	**77.4%**
	Qwen3-14B (Non-thinking)	0.136	0.081	40.5%	+0.75	13.0%
Math Qwen3-1.7B (Non-thinking)	Qwen3-4B-RL-Math	0.263	0.092	**65.1%**	**+22.83**	**47.5%**
	Qwen3-4B (Non-thinking)	0.099	0.069	30.4%	+2.07	14.3%
Math R1-Distill-1.5B	JustRL-1.5B†	0.455	0.017	**96.3%**	**+12.03**	**70.7%**
	Skywork-7B	0.210	0.131	37.7%	+6.27	20.2%
	R1-Distill-7B	0.136	0.089	34.3%	+2.80	15.6%

than direct OPD with any of the three larger-scale teachers tested. We further continue OPD for 200 updates from the SFT checkpoints for R1-Distill-7B and Qwen3-4B (Non-thinking). The final accuracies are 40.2% and 22.6%, respectively, compared with 42.7% and 22.7% for direct OPD. The maximum observed loss reductions during these subsequent OPD runs are only 37.1% and 23.0%. Thus, SFT initialization does not resolve the limited loss reduction or improve the final accuracy over direct OPD in these two settings (Appendix H.1).

Ablations of vocabulary support and rollout count. For the Qwen3-1.7B (Non-thinking) math student, ablations comparing top-16 with full-logit OPD and one with different training rollouts per prompt show similar training behavior and closely matched final results (Appendix E).

5 VANISHING LEARNING SIGNALS: LOSS DYNAMICS OF OPD

Section 4.3 raises an optimization question: why does OPD with larger-scale teachers stop reducing its loss, while OPD with self-RL teachers continues to do so? Section 5.1 introduces a continuous-time framework for OPD and uses it to analyze the loss dynamics. Section 5.2 shows that the learning signal collapses prematurely with larger-scale teachers. Section 5.3 explains theoretically why OPD works with self-RL teachers.

5.1 MEAN-FLOW FORMULATION AND LOSS DYNAMICS

Write θ_0 for the student parameters at the start of OPD. By differentiating $\mathcal{L}(\theta)$, we get:

$$\nabla\mathcal{L}(\theta) = \underbrace{\mathbb{E}_{\tau\sim\pi_\theta}\nabla_\theta\widehat{\mathcal{L}}(\theta;\tau)}_{=:\,g(\theta)} + \underbrace{\mathbb{E}_{\tau\sim\pi_\theta}\big[\widehat{\mathcal{L}}(\theta;\tau)\,\nabla_\theta\log\pi_\theta(y\mid x)\big]}_{=:\,h(\theta)}, \tag{3}$$

Here, the *detached gradient* g is the KL gradient evaluated at prefixes currently visited by the student, whereas the *occupancy gradient* h captures how changes in θ alter the prefixes the student visits (Mohamed et al., 2020).

At step m, OPD updates the parameters according to $\theta_{m+1} = \theta_m - \eta\,\widehat{g}_m$, where η is the learning rate and $\widehat{g}_m$ is an unbiased stochastic gradient of $\widehat{\mathcal{L}}$, satisfying $\mathbb{E}[\widehat{g}_m \mid \theta_m] = g(\theta_m)$. In the small-learning-rate limit, the sampling noise averages out. With the rescaled time $t = m\eta$, the iterates are then described by the ODE (Kushner & Yin, 2003; Borkar & Meyn, 2000)

$$\dot\theta(t) = -g(\theta(t)), \qquad \theta(0) = \theta_0. \tag{4}$$

See Appendix A.3 for the derivation and a discussion of the idealizations used here. Under this framework, we obtain the following exact loss dynamics, together with a factorization of the decay rate.

Proposition 5.1 (OPD loss dynamics)**.** *Under the regularity conditions of Appendix A.4, while the loss is positive, the flow in Eq* (4) *satisfies*

$$\frac{d}{dt}\mathcal{L}(\theta(t)) = -2\gamma(\theta(t))\,\mathcal{L}(\theta(t)), \qquad \gamma := \frac{\langle\nabla\mathcal{L}, g\rangle}{2\mathcal{L}} = \mu(1+\alpha), \tag{5}$$

where $\mu := \|g\|^2/(2\mathcal{L})$ *and* $\alpha := \langle h, g\rangle / \|g\|^2$*, the last equality holding wherever* $g \neq 0$*.*

The proof is given in Appendix A.4.

Proposition 5.1 describes how fast the loss falls. The rate γ is the fraction of the remaining loss removed per unit time, and its average over $[0,t]$ is $\overline{\gamma}(t) := \frac{1}{t}\int_0^t \gamma(\theta(u))\,du$. Integrating Eq. (5) gives $\overline{\gamma}(t) = \frac{1}{2t}\log\big(\mathcal{L}(\theta_0)/\mathcal{L}(\theta(t))\big)$, so $\overline{\gamma}(t)$ can be computed directly from the loss curve. A loss plateau, such as those observed in Section 4.3, corresponds to a small average of γ over the plateau interval. The factorization $\gamma = \mu(1+\alpha)$ splits this rate into two factors, so a small rate can arise in only two ways, through a small μ or through α near or below -1.

Here, $\mu = \|g\|^2/(2\mathcal{L})$ quantifies the learning signal relative to the remaining loss; a small μ indicates a weak signal. The coefficient $\alpha = \langle h, g\rangle / \|g\|^2$ measures how much of the occupancy gradient h points along the update g. So it indicates whether changes in the rollout distribution reinforce the update ($\alpha > 0$) or oppose it ($\alpha < 0$). When $\alpha \le -1$, the loss no longer decreases, even if $g \neq 0$.

5.2 PREMATURE LEARNING-SIGNAL COLLAPSE WITH LARGER-SCALE TEACHERS

Proposition 5.1 suggests two possible explanations for the loss plateau in Section 4.3: the learning signal may be weak (μ is small), or the occupancy gradient may oppose the update ($\alpha \le -1$). We measure both quantities in the code setting with student Qwen3-4B (Non-thinking) and in the math setting with student R1-Distill-1.5B. At every training step m, we estimate μ from the training logs as $\widehat{\mu}_m = \widehat{G}_m^2/(2\widehat{\mathcal{L}}_m)$, where $\widehat{G}_m$ is the gradient norm and $\widehat{\mathcal{L}}_m$ is the loss. We estimate α using $\widehat{\alpha}_m$ at checkpoints saved every 10 training steps. At each checkpoint, we use sampled responses to cross-fit g and h. See Appendix E for details. The measured $\widehat{\mu}_m$ uses a noisy, pre-clipping minibatch gradient; it does not measure the size of an AdamW parameter update.

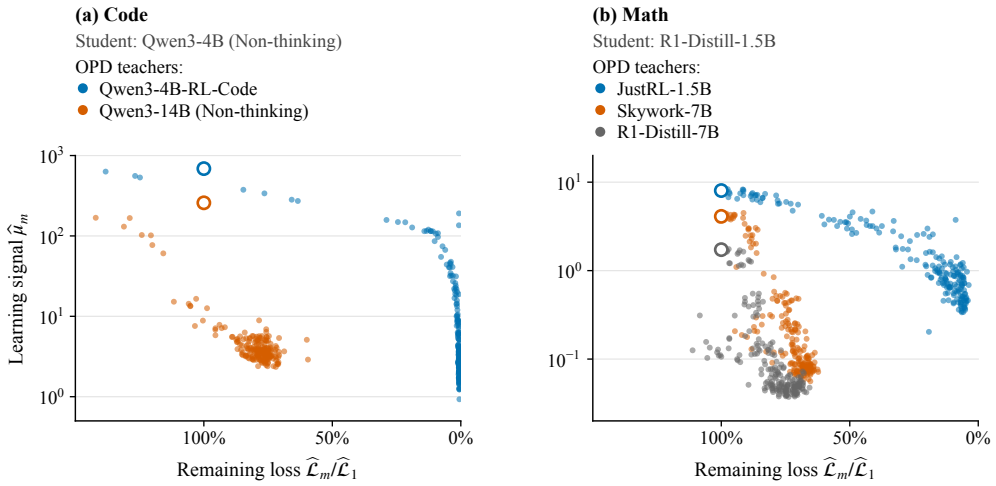


Figure 2: Learning signal $\widehat{\mu}_m$ versus remaining loss $\widehat{\mathcal{L}}_m/\widehat{\mathcal{L}}_1$ over 200 updates. Remaining loss decreases from left to right; the learning-signal axis is logarithmic. Each dot is one training step; open circles mark step 1. Colors and teachers match Figure 1. Appendix E gives the estimators and sources, and shows the corresponding time series in Figure 3.

With larger-scale teachers, the learning signal collapses while much of the loss remains. Figure 2 plots the estimated learning signal $\widehat{\mu}_m$ against the remaining loss at each training step. With self-RL teachers, the loss decreases by more than 90% as the learning signal declines gradually. By contrast, with larger-scale teachers, the learning signal drops by more than an order of magnitude while over 60% of the loss remains. After this drop, the loss changes little. We call this pattern *premature learning-signal collapse*: the logged gradient weakens before the teacher–student mismatch has been substantially reduced.

The occupancy gradient does not consistently explain the loss plateaus. Table 3 shows that, in the code setting, $1+\widehat{\alpha}_m$ is positive at 19 of 20 checkpoints for Qwen3-14B. These estimates provide little evidence for persistent occupancy cancellation in this run. In the math setting, $1+\widehat{\alpha}_m$ is positive at 12 of the 20 checkpoints for Skywork-7B and 13 of the 20 checkpoints for R1-Distill-7B. The other estimates are consistent with the occupancy gradient offsetting or reversing loss reduction in the idealized flow. This may slow training further, alongside the collapse of μ.

Across the code and math settings, an early drop in $\widehat{\mu}_m$ is the common pattern. Cross-fitted estimates of $\|g\|^2$ decline from early to middle training in Math but remain roughly stable for Qwen3-14B after step 20. The occupancy estimates remain tentative, however, because α is measured at only 20 checkpoints per teacher in each setting.

Table 3: Occupancy factor $1+\widehat{\alpha}_m$ for larger-scale teachers: median, interquartile range (IQR), and counts above zero across 20 checkpoints (steps 10–200, every 10 steps). The estimator is given in Appendix E.

		$1+\widehat{\alpha}_m$		
Task	**Teacher**	**Median**	**IQR**	**Positive**
Code	Qwen3-14B	1.10	[0.67, 1.38]	19/20
Math	Skywork-7B	0.13	[−2.31, 1.70]	12/20
	R1-Distill-7B	0.41	[−1.16, 1.69]	13/20

5.3 Local Convergence with Self-RL Teachers

We now explain why OPD keeps reducing its loss with self-RL teachers. Since each such teacher is obtained by further RL training of the student (Section 4.1), it shares the parameter space of the student, and we write it as $\theta_T = \theta_0 + \Delta$. We also find that the teacher is close to the student in this space. As Table 4 shows, the relative distance $\|\Delta\| / \|\theta_0\|$ is 0.030% for Qwen3-4B-RL-Code and 0.50% for JustRL-1.5B. As a supplementary run, we also apply OPD with the self-RL instruction-following teacher UltraData-IF-1.5B, whose distance is 0.037% (Appendix G). If the teacher is close enough, the OPD loss around it is approximately quadratic (Proposition D.1). On such a quadratic, the OPD flow keeps reducing the loss. Theorem 5.2 makes this precise and shows how the remaining loss depends on the distance.

Setting. We analyze OPD in the continuous-time ODE framework of Eq (4). We also use three regularity conditions that are standard in policy-gradient analysis, namely (i) bounded rollout length (Razin et al., 2025), (ii) a bounded and Lipschitz policy score (Fatkhullin et al., 2023; Barakat et al., 2022; Chakraborty et al., 2024), and (iii) a Lipschitz Hessian of $\mathcal{L}$ (Jaiswal et al., 2024). Appendix C states them in full. Our OPD experiments with self-RL teachers in Section 4.3 motivate this analysis but do not directly test its local, spectrum-dependent bound.

Slow directions and residual loss. The recovery bound depends on both the teacher–student distance $\rho = \|\Delta\|$ and the directions of Δ. We characterize these directions using the initial student's Fisher matrix $H_0 = H(\theta_0)$, which locally approximates the loss Hessian for nearby teachers (Appendix B). In the quadratic approximation, loss along an eigenvector u_i of H_0 with eigenvalue λ_i decays as $e^{-2\lambda_i t}$, so directions with small eigenvalues are learned slowly (Appendix D). For a cutoff $\delta > 0$, their contribution to the initial loss is

$$E_\Delta(\delta) := \frac{1}{2} \sum_{0<\lambda_i \le \delta} \lambda_i (u_i^\top \Delta)^2 \le \frac{\delta}{2} \rho^2. \tag{6}$$

Theorem 5.2 (Abridged version of Theorem C.1)**.** *Let θ_T be a self-RL teacher, with $\rho = \|\Delta\| > 0$. Under the conditions of Appendix C, the OPD flow in Eq* (4) *stays within distance 2ρ of the teacher up to a time $T_{\mathrm{loc}} = \Omega(1/\rho)$. For every $\delta > 0$ and $0 \le t \le T_{\mathrm{loc}}$,*

$$\mathcal{L}(\theta(t)) \le e^{-\delta t} \mathcal{L}(\theta_0) + E_\Delta(\delta) + O\big(\rho^3 + \rho^4/\delta\big), \tag{7}$$

where $\Omega(\cdot)$ and $O(\cdot)$ hide constants depending on the policy-score bounds, the rollout length and the Lipschitz constant of the Hessian of $\mathcal{L}$.

The proof is given in Appendix C. Theorem 5.2 bounds the loss by an exponentially decaying term plus a residual determined by $E_\Delta(\delta)$ and higher-order errors in ρ. This is consistent with Section 4.3, where the minimum loss reaches 0.5% and 3.7% of its starting value with the two self-RL teachers.

We emphasize that Theorem 5.2 does not suggest that only self-RL teachers are easy to learn from. Its proof does not use how the teacher was trained, so OPD can reach any teacher in the student's parameter space that is close enough to the student and differs from it mostly along high-curvature directions. Rather, it explains why the self-RL teachers in our experiments, which lie close to their students, can be reached by OPD, as shown in Section 4.3. The larger-scale teachers are not covered, because their parameter dimensions differ from those of their students.

6 Parameter and Representation Changes in OPD

Section 5 shows that OPD can reach a teacher close to the student, but its learning signal collapses early with larger-scale teachers. A possible reason is that OPD only makes local changes to the student, which may not suffice when matching the teacher requires changes these updates do not make. We therefore measure how much OPD changes the student, both in its parameters and in its hidden representations.

Metrics. We use the relative parameter change $d_{\mathrm{par}} = \|\theta_{\mathrm{end}} - \theta_{\mathrm{start}}\| / \|\theta_{\mathrm{start}}\|$ to measure how far OPD moves the student in parameter space. Because parameter distances say little about what the

model computes, we also use linear Centered Kernel Alignment (CKA) (Kornblith et al., 2019) to measure how much its hidden representations change. For a set of prompts, let X and Y be the hidden states of the starting and final checkpoints at one layer, with one row per prompt, averaged over its tokens and centered across prompts. Then $\mathrm{CKA}(X, Y) = \left\|Y^\top X\right\|_F^2 / (\left\|X^\top X\right\|_F \left\|Y^\top Y\right\|_F)$, which equals 1 when the two representations agree up to rotation and scaling. We use the prompts of the evaluation benchmarks, namely AIME24, AIME25 and AMC23 in the math setting (143 prompts) and HumanEval+, MBPP+ and LiveCodeBench in the code setting (673 prompts). See Appendix F for details.

OPD changes the student's parameters and representations only slightly. Table 4 reports both metrics for eight OPD runs after 200 updates. In every run, d_{par} is between 0.025% and 0.098%, and CKA is at least 0.985 at every layer. OPD therefore changes the student only slightly. The displacement magnitude alone does not separate successful from stalled runs. Whether the direction of these small changes limits teacher matching remains a hypothesis for controlled interventions.

Table 4: Parameter and representation changes from the student's initialization. *OPD student* is measured after 200 updates; *Teacher* compares the self-RL teacher (†). For teachers with a different parameterization, d_{par} is undefined and teacher–student CKA was not computed (–). d_{par} denotes relative parameter change; Min. CKA is the minimum layerwise CKA. IF details: Appendix G.

			OPD student		**Teacher**	
Task	**Student**	**Teacher**	d_{par}	Min. CKA	d_{par}	Min. CKA
Code	Qwen3-4B	Qwen3-4B-RL-Code†	0.032%	0.9979	0.030%	0.9993
		Qwen3-14B	0.063%	0.9850	–	–
Math	Qwen3-1.7B	Qwen3-4B-RL-Math	0.031%	0.9939	–	–
		Qwen3-4B	0.025%	0.9973	–	–
Math	R1-Distill-1.5B	JustRL-1.5B†	0.098%	0.9904	0.497%	0.9674
		Skywork-7B	0.063%	0.9988	–	–
		R1-Distill-7B	0.050%	0.9986	–	–
IF	R1-Distill-1.5B	UltraData-IF-1.5B†	0.026%	0.9999	0.037%	0.9998

Self-RL teachers remain close to the initial student. The Teacher columns of Table 4 compare each self-RL teacher with the base model it was trained from. RL post-training moves the base model by 0.030% to 0.50%, and its minimum CKA is between 0.967 and 0.9998. These changes are comparable to or moderately larger than those made by OPD (up to about $5\times$, for JustRL). These measurements are consistent with the local recovery picture of Section 5.3.

7 Conclusion

We study why OPD can plateau while substantial teacher–student disagreement remains. Across code generation and mathematical reasoning, our diagnostics link these plateaus to premature learning-signal collapse, while self-RL teachers support sustained loss reduction. Our continuous-time analysis separates the effects of the learning signal and the changing rollout distribution, and provides a local recovery guarantee for sufficiently nearby teachers under regularity conditions. Small parameter changes and largely preserved representations further suggest that limited adaptation may constrain learning from other teachers, although displacement magnitude alone does not separate successful from stalled runs. Our local recovery guarantee assumes an idealized continuous-time flow and a shared parameterization, leaving extensions to practical OPD for future work. The link between limited representation adaptation and learning-signal collapse remains a hypothesis to be tested through controlled interventions.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript organization, literature and citation checks, and figure preparation during this work. They also provided routine programming support for plotting and analysis utilities, including code drafting and debugging. In addition, these tools were used to flag possible inconsistencies in notation, presentation, proofs, and interpretations of results. Such suggestions support, rather than replace, the authors' assessment of the scientific claims. Teacher-generated responses were also used to construct the SFT datasets for the ablations described in Appendix H.1. The experimental observations and numerical summaries reported in the paper are based on recorded training and evaluation outputs, not on AI-generated assertions. The authors are responsible for validating AI-assisted text, code, and proof suggestions, and for the correctness of the final manuscript, experiments, results, and conclusions.

ETHICS STATEMENT

This work studies the optimization behavior and limitations of on-policy distillation in language models. The experiments use existing model checkpoints and research datasets for code generation, mathematical reasoning, and instruction following. We do not conduct new studies involving human subjects or collect personally identifiable information for this study. Student-generated responses are used as training rollouts and evaluation outputs, with performance assessed using the corresponding benchmark procedures. Distillation can transfer undesirable behaviors, biases, or incorrect reasoning along with useful capabilities, and better benchmark performance does not establish the safety of a distilled model. Our theoretical guarantees concern local loss reduction under specified assumptions; they do not certify factual correctness, fairness, or safe behavior in deployment. Improving the efficiency of capability transfer may also facilitate harmful applications. Deployment therefore requires application-specific evaluation and safeguards beyond the optimization and task-performance measurements studied here.

REPRODUCIBILITY STATEMENT

We describe the models, training datasets, OPD settings, and evaluation metrics in Sections 4.1–4.2. Appendix E provides checkpoint identities, training configurations, decoding settings, response-length limits, benchmark aggregation, and the construction of the reported loss and learning-signal statistics. It also documents run provenance, the fixed-seed subsampling used to obtain code $\mathrm{avg@4}$ scores, and checkpoint-level occupancy diagnostics. The continuous-time formulation, regularity assumptions, and proofs underlying our loss-dynamics identity and local recovery guarantee are given in Appendices A–D. Appendix F specifies parameter-distance and representation-similarity measurements, including representation extraction, linear CKA, and prompt-resampling checks. Appendices G and H provide the instruction-following experiments and additional ablations, including top-K versus full-logit supervision and training rollout counts. The experimental appendices distinguish individual training runs from checkpoint and resampling diagnostics and state the available seed information. We will publicly release our code upon publication, including the experiment and analysis utilities used to obtain the reported results and figures, to support independent verification and follow-up work.

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.

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc NM Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why. *arXiv preprint arXiv:2605.10889*, 2026.

Anas Barakat, Pascal Bianchi, and Julien Lehmann. Analysis of a target-based actor-critic algorithm with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, pp. 991–1040. PMLR, 2022.

Vivek S Borkar and Sean P Meyn. The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.

Yuchen Cai, Ding Cao, Liang Lin, Chunxi Luo, Xin Xu, Kai Yang, Weijie Liu, Saiyong Yang, Tianxiang Zhao, Guangzhong Sun, et al. Learning to foresee: Unveiling the unlocking efficiency of on-policy distillation. *arXiv preprint arXiv:2605.11739*, 2026.

Souradip Chakraborty, Amrit Bedi, Alec Koppel, Huazheng Wang, Dinesh Manocha, Mengdi Wang, and Furong Huang. Parl: A unified framework for policy alignment in reinforcement learning from human feedback. In *International Conference on Learning Representations*, volume 2024, pp. 24410–24449, 2024.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4793–4801. IEEE, 2019.

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

Wojciech M Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1331–1340. PMLR, 2019.

Omer Elkabetz and Nadav Cohen. Continuous vs. discrete optimization of deep neural networks. *Advances in Neural Information Processing Systems*, 34:4947–4960, 2021.

Ilyas Fatkhullin, Anas Barakat, Anastasia Kireeva, and Niao He. Stochastic policy gradient methods: Improved sample complexity for fisher-non-degenerate policies. In *International Conference on Machine Learning*, pp. 9827–9869. PMLR, 2023.

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*, 2026a.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, et al. Rethinking on-policy distillation of large language models ii: One training example. *arXiv preprint arXiv:2609.04172*, 2026b.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

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.

Hrayr Harutyunyan, Ankit Singh Rawat, Aditya Krishna Menon, Seungyeon Kim, and Sanjiv Kumar. Supervision complexity and its role in knowledge distillation. *arXiv preprint arXiv:2301.12245*, 2023.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025a.

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

Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, et al. Multi-if: Benchmarking llms on multi-turn and multilingual instructions following. *arXiv preprint arXiv:2410.15553*, 2024.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 138306–138322, 2026.

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

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

Ajay Jaiswal, Yifan Wang, Lu Yin, Shiwei Liu, Runjin Chen, Jiawei Zhao, Ananth Grama, Yuandong Tian, and Zhangyang Wang. From low rank gradient subspace stabilization to low-rank weights: Observations, theories, and applications. *arXiv preprint arXiv:2407.11239*, 2024.

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.

Yuxuan Jiang, Runchao Li, Shubhashis Roy Dipta, Dawei Li, and Zhao Yang. Cornerstones or stumbling blocks? deciphering the rock tokens in on-policy distillation. *arXiv preprint arXiv:2605.09253*, 2026.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Harold J Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*. Springer, 2003.

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.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6121–6133, 2020.

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.

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

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *International Conference on Machine Learning*, pp. 23610–23641. PMLR, 2023.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International conference on machine learning*, pp. 6820–6829. PMLR, 2020.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *URL https://arxiv.org/abs/2505.11711*, 2025.

Abhishek Panigrahi, Bingbin Liu, Sadhika Malladi, Sham Kakade, and Surbhi Goel. In good graces: Principled teacher selection for knowledge distillation. In *International Conference on Learning Representations*, volume 2026, pp. 54779–54822, 2026.

Noam Razin, Hattie Zhou, Omid Saremi, Vimal Thilak, Arwen Bradley, Preetum Nakkiran, Joshua Susskind, and Etai Littwin. Vanishing gradients in reinforcement finetuning of language models. In *International Conference on Learning Representations*, volume 2024, pp. 10380–10416, 2024.

Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason Lee, and Sanjeev Arora. What makes a reward model a good teacher? an optimization perspective. *Advances in neural information processing systems*, 38:59162–59222, 2025.

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

Bharat Runwal, Ashish Agrawal, Anurag Roy, and Rameswar Panda. Prism: Demystifying retention and interaction in mid-training. *arXiv preprint arXiv:2603.17074*, 2026.

Zhennan Shen, Yanshu Li, Qingyu Yin, Chak Tou Leong, Zhilin Wang, Yanxu Chen, Rongduo Han, Sunbowen Lee, and Yi R Fung. On the geometry of on-policy distillation. *arXiv preprint arXiv:2606.07082*, 2026.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026.

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

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in neural information processing systems*, 34:6906–6919, 2021.

Rui Wang, Hongru Wang, Yi Chen, Boyang Xue, Tianqing Fang, Wenhao Yu, and Kam-Fai Wong. Demystifying on-policy distillation: Roles, pathologies, and regulations. *arXiv preprint arXiv:2607.13399*, 2026a.

Yuanyi Wang, Su Lu, Yanggan Gu, Pengkai Wang, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Not all disagreement is learnable: Token teachability in on-policy distillation. *arXiv preprint arXiv:2605.26844*, 2026b.

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

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

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2025.

Huaqing Zhang, Jingchu Gai, Juno Kim, Bingbin Liu, and Andrej Risteski. When does online imitation learning help in llm post-training? the role of (non-) realizability beyond horizon. *arXiv preprint arXiv:2606.30445*, 2026.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, et al. The path not taken: Rlvr provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026.

A OPD CALCULUS AND CONTINUOUS-TIME IDENTITY

A.1 PER-PREFIX REVERSE-KL GRADIENT

Let

$$r_\theta(a, s) := \log \frac{\pi_\theta(a \mid s)}{\pi_T(a \mid s)}. \tag{8}$$

Write $\ell(\theta; s) = D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s) \,\|\, \pi_T(\cdot \mid s)\big)$ for the per-prefix loss in Eq (1). At a fixed prefix,

$$\nabla_\theta \ell(\theta; s) = \sum_a \pi_\theta(a \mid s)\big(r_\theta(a, s) + 1\big)\psi_\theta(a, s) \tag{9}$$

$$= \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\left(r_\theta(a, s) - \mathbb{E}_{b\sim\pi_\theta} r_\theta(b, s)\right)\psi_\theta(a, s)\right]. \tag{10}$$

The second equality uses $\mathbb{E}_{a\sim\pi_\theta}\psi_\theta(a, s) = 0$; the centered baseline is optional. For the forward KL $D_{\mathrm{KL}}(\pi_T(\cdot \mid s)\|\pi_\theta(\cdot \mid s))$, the corresponding fixed-prefix gradient is $-\mathbb{E}_{a\sim\pi_T}\psi_\theta(a, s)$.

A.2 POPULATION GRADIENT SPLIT

Assume differentiation may be interchanged with expectation in Eq (1). Since the prompt distribution does not depend on θ, the score of a trajectory $\tau = (x, y)$ is $\nabla_\theta \log \pi_\theta(y \mid x)$, and the likelihood-ratio identity gives

$$\nabla\mathcal{L}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\nabla_\theta\widehat{\mathcal{L}}(\theta; \tau) + \mathbb{E}_{\tau\sim\pi_\theta}\left[\widehat{\mathcal{L}}(\theta; \tau)\nabla_\theta \log \pi_\theta(y \mid x)\right] \tag{11}$$

$$= g(\theta) + h(\theta), \tag{12}$$

which proves Eq (3). Write $\mathcal{A}(\theta) := \langle \nabla\mathcal{L}(\theta), g(\theta)\rangle$ for the loss–update alignment, so that $\mathcal{A} = 2\gamma\mathcal{L}$ in the notation of Proposition 5.1. Substituting the split gives

$$\mathcal{A}(\theta) = \|g(\theta)\|^2 + \langle h(\theta), g(\theta)\rangle. \tag{13}$$

A.3 FROM THE OPD UPDATE TO THE ODE

Write the update of Section 5.1 as

$$\theta_{m+1} = \theta_m - \eta\,\widehat{g}_m, \qquad \widehat{g}_m = g(\theta_m) + \xi_m, \tag{14}$$

where ξ_m is the sampling noise of the minibatch gradient, with $\mathbb{E}[\xi_m \mid \theta_m] = 0$. Take a short interval of length Δt that contains $n = \Delta t/\eta$ steps starting from step m. Summing Eq (14) over these steps gives

$$\theta_{m+n} - \theta_m = -\eta \sum_{k=m}^{m+n-1} g(\theta_k) - \eta \sum_{k=m}^{m+n-1} \xi_k.$$

Because the iterates move little over the interval and g is continuous, the first sum equals $\Delta t\, g(\theta_m)$ up to a term of order Δt^2. The second sum has mean zero and, for noise with bounded variance, a standard deviation of order $\sqrt{\eta\,\Delta t}$, which is negligible next to Δt when $\eta \to 0$. Writing $\theta(t) = \theta_{t/\eta}$, dividing by Δt, and letting first $\eta \to 0$ and then $\Delta t \to 0$ gives Eq (4). This is the standard ODE limit of a stochastic iteration (Kushner & Yin, 2003; Borkar & Meyn, 2000), and Proposition A.1 below bounds the loss after finitely many steps without taking the limit.

Our runs use AdamW, whereas Eq (4) idealizes detached-gradient descent. AdamW's momentum, preconditioning, and weight decay can change the update direction; gradient clipping also rescales the gradient. Razin et al. (2025) make the same idealization in their continuous-time analysis of policy gradient. Finally, the analysis uses the full-vocabulary KL in Eq (1), whereas our main runs optimize its top-16 version in Eq (2). The statistics measured in Section 5.2 are computed on the top-16 objective and serve as proxies for the corresponding full-KL quantities.

A.4 Proof of Proposition 5.1

Regularity conditions. Proposition 5.1 uses two conditions. *Differentiation:* differentiation and expectation may be interchanged in Eq (1), so that the split in Eq (3) holds. *Flow regularity:* there is an open set $\mathcal{U} \ni \theta_0$ on which $\mathcal{L}$ is C^1 and g is locally Lipschitz, and Eq (4) has a solution that stays in $\mathcal{U}$ on $[0, T_\star]$ for the horizon $T_\star$ under consideration. Since g is locally Lipschitz, this solution is unique and C^1 (Picard–Lindelöf), so $t \mapsto \mathcal{L}(\theta(t))$ is C^1 and the chain rule applies.

Proof of Proposition 5.1. Rate identity in Eq (5)*.* By the chain rule and Eq (4), for $t \in [0, T_\star]$,

$$\frac{d}{dt}\mathcal{L}(\theta(t)) = \left\langle \nabla\mathcal{L}(\theta(t)), \dot{\theta}(t) \right\rangle = -\left\langle \nabla\mathcal{L}(\theta(t)), g(\theta(t)) \right\rangle = -\mathcal{A}(\theta(t)).$$

Wherever $\mathcal{L}(\theta(t)) > 0$, the definition $\gamma = \mathcal{A}/(2\mathcal{L})$ rewrites the right-hand side as $-2\gamma(\theta(t))\,\mathcal{L}(\theta(t))$, which is Eq (5).

Integrated form. Let $t_0 := \inf\{t \in [0, T_\star] : \mathcal{L}(\theta(t)) = 0\}$, with $t_0 := T_\star$ if the loss never vanishes. On $[0, t_0)$ the loss is positive, so $\log \mathcal{L}(\theta(t))$ is C^1 with $\frac{d}{dt}\log\mathcal{L}(\theta(t)) = -2\gamma(\theta(t))$ by Eq (5). Integrating from 0 to $t < t_0$ and exponentiating gives

$$\mathcal{L}(\theta(t)) = \mathcal{L}(\theta_0)\exp\left(-2\int_0^t \gamma(\theta(u))\,du\right), \tag{15}$$

which yields the run average $\overline{\gamma}(t)$ of Section 5.1.

Factorization. Wherever $\mathcal{L}(\theta) > 0$ and $g(\theta) \neq 0$, dividing Eq (13) by $2\mathcal{L}(\theta)$ gives

$$\gamma = \frac{\|g\|^2}{2\mathcal{L}} + \frac{\langle h, g\rangle}{2\mathcal{L}} = \frac{\|g\|^2}{2\mathcal{L}}\left(1 + \frac{\langle h, g\rangle}{\|g\|^2}\right) = \mu(1+\alpha)$$

by the definitions of μ and α. □

By Cauchy–Schwarz, $|\alpha| \le \|h\| / \|g\|$, so the factorization gives $\mu(1 - \|h\| / \|g\|) \le \gamma \le \mu(1 + \|h\| / \|g\|)$.

Proposition A.1 (Finite-step stochastic decomposition)**.** *Suppose $\mathcal{L}$ is β-smooth on a region containing the iterates, $\mathbb{E}[\widehat{g}_m \mid \mathcal{F}_m] = g(\theta_m)$, and*

$$\sigma_m^2 := \mathbb{E}\left[\|\widehat{g}_m - g(\theta_m)\|^2 \mid \mathcal{F}_m\right] < \infty. \tag{16}$$

Then

$$\begin{aligned}\mathbb{E}\mathcal{L}(\theta_M) \le \mathcal{L}(\theta_0) &- \sum_{m=0}^{M-1} \eta_m \mathbb{E}\mathcal{A}(\theta_m) \\ &+ \frac{\beta}{2}\sum_{m=0}^{M-1} \eta_m^2 \mathbb{E}\left[\|g(\theta_m)\|^2 + \sigma_m^2\right].\end{aligned} \tag{17}$$

Proof. Smoothness and the update $\theta_{m+1} = \theta_m - \eta_m \widehat{g}_m$ imply

$$\mathbb{E}_m\mathcal{L}(\theta_{m+1}) \le \mathcal{L}(\theta_m) - \eta_m\mathcal{A}(\theta_m) + \frac{\beta\eta_m^2}{2}\left(\|g(\theta_m)\|^2 + \sigma_m^2\right).$$

Take total expectations and sum over m. □

B Parameter differences and the initial Fisher

Throughout this section, the teacher and student use the same architecture, parameter coordinates, tokenizer, vocabulary, temperature, and policy interface. Set $\theta_T = \theta_{\mathrm{RL}}$ and $\Delta = \theta_T - \theta_0$. All

expectations below are assumed finite, with $N(\tau) > 0$ almost surely. For a policy checkpoint ϑ, write $\psi_\vartheta(a, s) = \nabla_\vartheta \log \pi_\vartheta(a \mid s)$ and define

$$H(\vartheta) := \mathbb{E}_{\tau\sim\pi_\vartheta}\left[\frac{1}{N(\tau)}\sum_{s\in\mathcal{S}(\tau)}\mathbb{E}_{a\sim\pi_\vartheta(\cdot|s)}\psi_\vartheta(a,s)\psi_\vartheta(a,s)^\top\right]. \tag{18}$$

Thus $H_0 = H(\theta_0)$. The teacher Fisher and its difference from the initial-student Fisher are

$$H_T := H(\theta_T), \qquad \varepsilon_H := \|H_T - H_0\|_{\mathrm{op}}. \tag{19}$$

Both matrices use the expectation of the token-normalized statistic, not a ratio of separate expectations of its numerator and denominator.

Lemma B.1 (Low-curvature endpoint energy)**.** *For $H_0 \succeq 0$, let $P_{\le\delta}$ be its spectral projector onto eigenvalues in $[0, \delta]$. Then*

$$E_\Delta(\delta) = \tfrac{1}{2}\Delta^\top H_0 P_{\le\delta}\Delta, \qquad 0 \le E_\Delta(\delta) \le \tfrac{\delta}{2}\|\Delta\|^2. \tag{20}$$

No restriction of Δ to the range of H_0 is needed.

Proof. Write $d_i = u_i^\top\Delta$ in an orthonormal eigenbasis of H_0. Expanding the quadratic form gives $E_\Delta(\delta) = \frac{1}{2}\sum_{0<\lambda_i\le\delta}\lambda_i d_i^2$. Each retained eigenvalue is at most δ, so the sum is bounded by $\delta\sum_i d_i^2/2 = \delta\|\Delta\|^2/2$. Zero eigenvalues contribute no energy; nullspace components need not vanish. □

Lemma B.2 (Same-model root and Fisher identities)**.** *Assume positive conditional token probabilities and a rollout law with common support in a neighborhood of θ_T. Assume the smoothness and integrable domination needed to differentiate $\mathcal{L}$ twice and g once. Then*

$$\mathcal{L}(\theta_T) = 0, \qquad \nabla\mathcal{L}(\theta_T) = 0, \qquad g(\theta_T) = 0, \tag{21}$$

and

$$\nabla^2\mathcal{L}(\theta_T) = Dg(\theta_T) = H_T. \tag{22}$$

Proof. At θ_T, every conditional KL and its first parameter derivative vanish for every fixed prefix. Its Hessian is the conditional policy Fisher. Hence $\widehat{\mathcal{L}}(\theta_T;\tau)$ and its first parameter derivative vanish for every sampled trajectory τ. When differentiating $\mathcal{L}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\widehat{\mathcal{L}}(\theta;\tau)$ twice, all terms involving derivatives of the rollout law multiply one of these vanishing quantities. The remaining term is $\mathbb{E}_{\tau\sim\pi_{\theta_T}}\nabla_\theta^2\widehat{\mathcal{L}}(\theta_T;\tau) = H_T$. Differentiating g once gives the same term because the derivative of the rollout law multiplies $\nabla_\theta\widehat{\mathcal{L}}(\theta_T;\tau) = 0$. The denominator $N(\tau)$ is held fixed conditional on τ; its distributional dependence is already included in the law of $\tau \sim \pi_\theta$. □

Score and sampling conditions. We impose boundedness and Lipschitz continuity of the policy score,

$$\|\psi_\theta(a,s)\| \le B, \qquad \|\psi_\theta(a,s) - \psi_{\theta'}(a,s)\| \le L_s\|\theta - \theta'\|, \tag{23}$$

uniformly over the relevant prefixes and tokens and over parameters near θ_T. These conditions are standard in policy-gradient analysis (Fatkhullin et al., 2023, Assumption 1) and are also used by Barakat et al. (2022, Assumption 3.1) and in RLHF (Chakraborty et al., 2024, Assumption 3, Eq (19)). Prompts follow a fixed distribution. Each sampled trajectory uses at most M autoregressive token draws from the full student policy, with fixed stopping and masking rules and $N(\tau) = |\mathcal{S}(\tau)| > 0$. Here M bounds all draws in τ, including every response when τ collects a batch. A finite generation horizon is also used by Razin et al. (2025, Section 3.1). Lemma B.3 derives a quadratic update remainder with

$$C_g := \tfrac{3}{2}BL_s + (M+1)B^3. \tag{24}$$

Lemma B.3 (Policy scores and the detached update)**.** *Let $\mathcal{B}_R$ be a closed ball of radius R about θ_T. Assume positive differentiable token probabilities and the uniform score bounds in Eq* (23) *on an open neighborhood of this ball. Prompts and auxiliary randomness have parameter-independent laws. Each trajectory is generated using at most M draws from π_θ; stopping, masking, and*

postprocessing are fixed functions of the generated trajectory and auxiliary randomness. Suppose $N(\tau) = |\mathcal{S}(\tau)| > 0$ *almost surely. Then* g *is locally Lipschitz near the ball and*

$$\|g(\theta_T + e) - H_T e\| \le C_g \|e\|^2, \qquad \theta_T + e \in \mathcal{B}_R, \tag{25}$$

with C_g *from Eq* (24)*. The self-Fisher map also satisfies*

$$\|H(\theta) - H(\theta')\|_{\mathrm{op}} \le L_H \|\theta - \theta'\|, \qquad L_H := 2BL_s + (M+1)B^3, \tag{26}$$

for $\theta, \theta' \in \mathcal{B}_R$*.*

Proof. Write $d = \|e\|$, $p_\theta(a) = \pi_\theta(a \mid s)$, and $b_s(\theta) = \nabla_\theta \ell(\theta; s)$ for a fixed prefix s. The score identity $\sum_a p_\theta(a)\psi_\theta(a, s) = 0$ gives

$$b_s(\theta) = \sum_a p_\theta(a)\psi_\theta(a, s) \log \frac{p_\theta(a)}{p_T(a)}.$$

Integrating along the segment from θ_T to $\theta_T + e$ yields

$$\sum_a |p_\theta(a) - p_T(a)| \le Gd, \tag{27}$$

$$\left|\log \frac{p_\theta(a)}{p_T(a)}\right| \le Gd, \tag{28}$$

$$\left|\log \frac{p_\theta(a)}{p_T(a)} - \psi_T(a, s)^\top e\right| \le \tfrac{L_s}{2} d^2. \tag{29}$$

Let $H_T(s) = \sum_a p_T(a)\psi_T(a, s)\psi_T(a, s)^\top$. Subtracting $H_T(s)e$ from $b_s(\theta)$ gives three terms: the change in the score, the remainder of the log-ratio, and the change from p_T to p_θ. Their norms are bounded by $BL_s d^2$, $BL_s d^2/2$, and $B^3 d^2$, respectively. Hence

$$\|b_s(\theta) - H_T(s)e\| \le C_{\mathrm{pre}} d^2, \qquad C_{\mathrm{pre}} := \tfrac{3}{2} BL_s + B^3. \tag{30}$$

We next control the trajectory distribution. In this proof, write Q_θ for the law of the recorded trajectory $\tau \sim \pi_\theta$. The likelihood score of the complete generated trajectory, including auxiliary randomness before postprocessing, is a sum of at most M token scores and has norm at most MG. Stopped sequences may be padded by absorbing states with zero score. Integrating the derivative of its law along the same segment, then applying the fixed postprocessing map, gives

$$\|Q_\theta - Q_{\theta_T}\|_1 \le MGd, \qquad \|Q - Q'\|_1 := \int |q - q'|. \tag{31}$$

For $A_T(\tau) := N(\tau)^{-1} \sum_{s \in \mathcal{S}(\tau)} H_T(s)$, we have $\|A_T(\tau)\|_{\mathrm{op}} \le B^2$ and $\mathbb{E}_{\tau \sim \pi_{\theta_T}} A_T(\tau) = H_T$. Averaging Eq (30) over scored prefixes and then over $\tau \sim \pi_\theta$ therefore gives

$$\|g(\theta) - H_T e\| \le C_{\mathrm{pre}} d^2 + B^2 d \|Q_\theta - Q_{\theta_T}\|_1 \le C_g d^2.$$

This averages each trajectory's normalized statistic; the random denominator is not replaced by its expectation.

For any two parameters in the ball, the conditional-policy law changes in L^1 by at most $B\|\theta - \theta'\|$. The outer product of two scores changes in operator norm by at most $2BL_s\|\theta - \theta'\|$. Combining these bounds with the trajectory-law bound gives Eq (26). Finally, the scores and log-ratios are bounded and Lipschitz on the ball. Applying the same distributional bounds to their product in b_s gives

$$\|g(\theta) - g(\theta')\| \le \left[B^2 + BL_s R + (M+1)B^3 R\right] \|\theta - \theta'\|.$$

The argument also applies on a slightly larger ball inside the open regularity neighborhood, proving the stated local Lipschitz property. □

The bounded-draw mechanism is part of this lemma's scope: conditioning on accepted trajectories or resampling empty responses requires controlling the resulting law separately unless all retries obey the stated draw bound. Parameter-dependent top-k or top-p truncation is not included.

Proposition B.4 (Local approximation from endpoint geometry)**.** *Assume Lemma B.2 and the score and sampling conditions of Lemma B.3 on a neighborhood of a ball $\mathcal{B}_R$ containing θ_0. Suppose $\mathcal{L}$ is C^2 with an $L_{\mathcal{L}}$-Lipschitz Hessian there, measured in operator norm. For $e = \theta - \theta_T$ and $\theta \in \mathcal{B}_R$,*

$$\left|\mathcal{L}(\theta) - \tfrac{1}{2} e^\top H_0 e\right| \le \tfrac{\varepsilon_H}{2} \|e\|^2 + \tfrac{L_{\mathcal{L}}}{6} \|e\|^3, \tag{32}$$

$$\|g(\theta) - H_0 e\| \le \varepsilon_H \|e\| + C_g \|e\|^2. \tag{33}$$

Consequently, uniform loss and update errors on $\mathcal{B}_R$ are bounded by

$$\varepsilon_{\mathcal{L}} = \tfrac{\varepsilon_H R^2}{2} + \tfrac{L_{\mathcal{L}} R^3}{6}, \qquad \varepsilon_g = \varepsilon_H R + C_g R^2. \tag{34}$$

Proof. The segment from θ_T to $\theta_T + e$ lies in the ball. The root identities and the Hessian condition give

$$\left|\mathcal{L}(\theta_T + e) - \tfrac{1}{2} e^\top H_T e\right| \le \tfrac{L_{\mathcal{L}}}{6} \|e\|^3.$$

Lemma B.3 gives the update remainder $\|g(\theta_T + e) - H_T e\| \le C_g \|e\|^2$. Replacing H_T by H_0 adds at most $\varepsilon_H \|e\|^2 / 2$ to the loss error and $\varepsilon_H \|e\|$ to the update error. Using $\|e\| \le R$ proves the uniform bounds. □

Lemma B.3 also yields $\varepsilon_H \le L_H \|\Delta\|$ with L_H defined in Eq (26). For $R = c\|\Delta\|$ with a fixed $c \ge 1$, the preceding constants satisfy

$$\varepsilon_{\mathcal{L}} \le \left(\tfrac{L_H c^2}{2} + \tfrac{L_{\mathcal{L}} c^3}{6}\right) \|\Delta\|^3, \tag{35}$$

$$\varepsilon_g \le \left(L_H c + C_g c^2\right) \|\Delta\|^2. \tag{36}$$

These are uniform functional estimates on the stated ball. The following lemma establishes how long the exact OPD flow remains there for the choice $R = 2\|\Delta\|$.

Lemma B.5 (A local validity interval for the OPD mean flow)**.** *Let $\rho = \|\theta_T - \theta_0\| > 0$ and suppose g is locally Lipschitz on an open neighborhood of the closed ball $\mathcal{B}_{2\rho}$ about θ_T. Assume $g(\theta_T) = 0$, $H_T \succeq 0$, and*

$$\|g(\theta_T + e) - H_T e\| \le C_g \|e\|^2 \quad \text{on } \mathcal{B}_{2\rho}.$$

The score conditions in Lemma B.3 supply this bound. Define

$$T_{\mathrm{loc}} := \begin{cases} (2C_g\rho)^{-1}, & C_g > 0, \\ +\infty, & C_g = 0. \end{cases} \tag{37}$$

The exact OPD mean flow in Eq (4) *exists uniquely and remains in $\mathcal{B}_{2\rho}$ for $0 \le t \le T_{\mathrm{loc}}$. For $C_g > 0$,*

$$\|\theta(t) - \theta_T\| \le \frac{\rho}{1 - C_g \rho t} \le 2\rho, \qquad 0 \le t \le T_{\mathrm{loc}}. \tag{38}$$

When $C_g = 0$, the distance is nonincreasing. If $\rho = 0$ and the field is locally Lipschitz at its root, the solution is stationary at θ_T.

Proof. Set $e = \theta - \theta_T$, $d = \|e\|$, and $q(e) = g(\theta_T + e) - H_T e$. For the original flow $\dot{e} = -g(\theta_T + e)$, positivity of H_T and the remainder bound imply, whenever $d > 0$,

$$\dot{d} = -\frac{e^\top H_T e + \langle e, q(e)\rangle}{d} \le C_g d^2. \tag{39}$$

If the root is reached, uniqueness makes the trajectory stationary there. Otherwise, integrate the differential inequality for $1/d$ to obtain the first bound in Eq (38) up to any first exit. For $t < T_{\mathrm{loc}}$, that bound is strictly below 2ρ, ruling out exit before T_{loc}. The solution stays in a compact ball contained in the open regularity neighborhood, so the ODE continuation theorem gives existence through T_{loc}; continuity gives the closed-ball bound at its endpoint. If $C_g = 0$, Eq (39) gives $d(t) \le \rho$, and the same compact-continuation argument applies for every finite time. The case $\rho = 0$ follows from uniqueness at $g(\theta_T) = 0$. □

C RECOVERY BOUND FROM THE PARAMETER DIFFERENCE

Write $e(t) = \theta(t) - \theta_T$, where $\theta_T = \theta_{\mathrm{RL}}$, and use the initial Fisher $H_0 = H(\theta_0)$ from Eq (18). Define the reference energy and its low-curvature part by

$$E_0(e) := \tfrac{1}{2} e^\top H_0 e, \qquad E_{\le\delta}(e) := E_0(P_{\le\delta} e). \tag{40}$$

Because $e(0) = -\Delta$, Lemma B.1 gives $E_{\le\delta}(e(0)) = E_\Delta(\delta)$ exactly. The theorem uses the error terms of Proposition B.4 with $R = 2\rho$,

$$\begin{aligned} \varepsilon_{\mathcal{L}}(\rho) &= 2\varepsilon_H \rho^2 + \tfrac{4}{3} L_{\mathcal{L}} \rho^3, \\ \varepsilon_g(\rho) &= 2\varepsilon_H \rho + 4 C_g \rho^2, \end{aligned} \tag{41}$$

where $\varepsilon_H = \|H_T - H_0\|_{\mathrm{op}}$ is defined in Eq (19).

Theorem C.1 (Local recovery for the nonlinear OPD mean flow)**.** *Consider exactly the flow in Eq* (4) *from Proposition 5.1, starting at θ_0, and let $\theta_T = \theta_{\mathrm{RL}}$ with $\rho = \|\theta_T - \theta_0\| > 0$. Assume:*

- *(S1) The teacher and student share their architecture, parameter coordinates, tokenizer, vocabulary, temperature, and policy interface. The quantities $\mathcal{L}$, g, and H_0 are well defined. The common-support, positivity and dominated-differentiation conditions of Lemma B.2 hold on a neighborhood of the closed ball $\mathcal{B}_{2\rho}$ about θ_T. Thus $g(\theta_T) = 0$ and $Dg(\theta_T) = H_T \succeq 0$.*
- *(S2) The score and sampling conditions of Lemma B.3 hold on an open neighborhood of $\mathcal{B}_{2\rho}$ with constants $B, L_s \ge 0$ and a finite total token-draw bound M. In particular, prompts have a fixed law, rollouts use the full student policy, stopping and masking rules do not depend on parameters, and $N(\tau) = |\mathcal{S}(\tau)| > 0$ almost surely.*
- *(S3) On an open neighborhood of $\mathcal{B}_{2\rho}$, the function $\mathcal{L}$ is C^2 with an $L_{\mathcal{L}}$-Lipschitz Hessian in operator norm.*

The root identities in (S1) follow from equality of student and teacher policies; they require no stationarity of the teacher's RL objective. Condition (S2) gives $C_g = \frac{3}{2} B L_s + (M+1) B^3$ and local Lipschitz continuity of the detached update. Let T_{loc} be defined by Eq (37)*, and let $\varepsilon_{\mathcal{L}}(\rho), \varepsilon_g(\rho)$ be the endpoint-dependent quantities in Eq* (41)*. The original OPD flow exists uniquely through T_{loc} and remains in $\mathcal{B}_{2\rho}$. For every finite $T_\star \le T_{\mathrm{loc}}$, every $\delta > 0$, and $0 \le t \le T_\star$,*

$$\boxed{\begin{aligned} \mathcal{L}(\theta(t)) \le e^{-\delta t} \mathcal{L}(\theta_0) + E_\Delta(\delta) \\ + 2\varepsilon_{\mathcal{L}}(\rho) + \varepsilon_g(\rho)^2 \left(\frac{t}{4} + \frac{1}{2\delta} \right). \end{aligned}} \tag{42}$$

The residual $E_\Delta(\delta)$ can be replaced by $\delta \|\theta_{\mathrm{RL}} - \theta_0\|^2 / 2$. In the exact quadratic case with zero approximation errors,

$$\mathcal{L}(\theta(t)) \le e^{-2\delta t} \mathcal{L}(\theta_0) + E_\Delta(\delta). \tag{43}$$

No lower bound on the smallest positive eigenvalue of either Fisher is required. If $\rho = 0$, local uniqueness at the root gives $\theta(t) = \theta_T$ and $\mathcal{L}(\theta(t)) = 0$.

For a target tolerance $\varepsilon > 0$, suppose some $\delta > 0$ and finite $T_\star \le T_{\mathrm{loc}}$ satisfy

$$E_\Delta(\delta) + 2\varepsilon_{\mathcal{L}}(\rho) + \varepsilon_g(\rho)^2 \left(\frac{T_\star}{4} + \frac{1}{2\delta} \right) \le \frac{\varepsilon}{2}. \tag{44}$$

Define

$$t_\varepsilon := \frac{1}{\delta} \log \max \left\{ 1, \frac{2\mathcal{L}(\theta_0)}{\varepsilon} \right\}. \tag{45}$$

If $t_\varepsilon \le T_\star$, then $\mathcal{L}(\theta(t)) \le \varepsilon$ for every $t \in [t_\varepsilon, T_\star]$.

Proof. Lemma B.5 gives existence and confinement of the original OPD mean flow through T_{loc}. Applying Proposition B.4 with $R = 2\rho$ gives

$$|\mathcal{L}(\theta(t)) - E_0(e(t))| \le \varepsilon_{\mathcal{L}}(\rho), \tag{46}$$

$$\|g(\theta(t)) - H_0 e(t)\| \le \varepsilon_g(\rho), \qquad 0 \le t \le T_\star. \tag{47}$$

For brevity, write $\varepsilon_{\mathcal{L}} = \varepsilon_{\mathcal{L}}(\rho)$ and $\varepsilon_g = \varepsilon_g(\rho)$ in the rest of the proof. Let $E(t) = E_0(e(t))$ and $r(t) = g(\theta(t)) - H_0 e(t)$. The identity $\dot{e} = -H_0 e - r$ rewrites the original update. Young's inequality and Eq (47) imply

$$\dot{E}(t) = -\|H_0 e(t)\|^2 - \langle H_0 e(t), r(t)\rangle \le -\tfrac{1}{2}\|H_0 e(t)\|^2 + \tfrac{1}{2}\varepsilon_g^2. \tag{48}$$

Write $E_{\le\delta}(t) = E_{\le\delta}(e(t))$. The spectral projector commutes with H_0. With $a = \|H_0 P_{\le\delta} e(t)\|$,

$$\dot{E}_{\le\delta}(t) = -a^2 - \langle H_0 P_{\le\delta} e(t), P_{\le\delta} r(t)\rangle \tag{49}$$

$$\le -a^2 + \varepsilon_g a \le \frac{\varepsilon_g^2}{4}. \tag{50}$$

Using the endpoint value $E_{\le\delta}(0) = E_\Delta(\delta)$ gives

$$E_{\le\delta}(t) \le E_\Delta(\delta) + \frac{\varepsilon_g^2 t}{4}. \tag{51}$$

The complementary eigenspace has eigenvalues greater than δ, so

$$\|H_0 e(t)\|^2 \ge 2\delta\big(E(t) - E_{\le\delta}(t)\big). \tag{52}$$

Combining these inequalities yields

$$\dot{E}(t) \le -\delta E(t) + \delta E_\Delta(\delta) + \frac{\delta\varepsilon_g^2 t}{4} + \frac{\varepsilon_g^2}{2}. \tag{53}$$

Multiplying by $e^{\delta t}$ and integrating gives the convenient bound

$$E(t) \le e^{-\delta t}E(0) + E_\Delta(\delta) + \varepsilon_g^2\left(\frac{t}{4} + \frac{1}{2\delta}\right). \tag{54}$$

For example, the term linear in time follows by bounding $u \le t$ in its convolution with $e^{-\delta(t-u)}$; the other forcing term contributes at most $\varepsilon_g^2/(2\delta)$. By Eq (46), $E(0) \le \mathcal{L}(\theta_0) + \varepsilon_{\mathcal{L}}$ and $\mathcal{L}(\theta(t)) \le E(t) + \varepsilon_{\mathcal{L}}$. Since $e^{-\delta t} \le 1$, substitution proves Eq (42). Lemma B.1 gives the parameter-distance bound.

In the exact quadratic limit, $\dot{e} = -H_0 e$. Expanding in an eigenbasis, with $c_i = u_i^\top e(0) = -u_i^\top \Delta$, gives $\mathcal{L}(\theta(t)) = \frac{1}{2}\sum_i \lambda_i c_i^2 e^{-2\lambda_i t}$. The terms with $\lambda_i > \delta$ are bounded by $e^{-2\delta t}\mathcal{L}(\theta_0)$ and the remaining terms by $E_\Delta(\delta)$, proving Eq (43). Finally, the error budget is bounded at $T_\star$ by Eq (44), and $e^{-\delta t}\mathcal{L}(\theta_0) \le \varepsilon/2$ for $t \ge t_\varepsilon$. The claimed interval is nonempty when $t_\varepsilon \le T_\star$, completing the proof. □

Abridged form. Lemma B.3 gives $\varepsilon_H \le L_H \rho$. Hence $\varepsilon_{\mathcal{L}}(\rho) = O(\rho^3)$ and $\varepsilon_g(\rho) = O(\rho^2)$ in Eq (41). For $t \le T_{\mathrm{loc}} = (2C_g\rho)^{-1}$, the last two terms of Eq (42) are therefore $O(\rho^3 + \rho^4/\delta)$, with constants depending only on B, L_s, M and $L_{\mathcal{L}}$. Since $T_{\mathrm{loc}} = \Omega(1/\rho)$, this gives Theorem 5.2.

D Local expansion and spectral specialization

Proposition D.1 (Local same-model Fisher expansion)**.** *Assume the conditions of Proposition B.4 on a neighborhood of θ_T. Let H_T be the teacher Fisher in Eq* (19)*. For $e = \theta - \theta_T$,*

$$\mathcal{L}(\theta_T + e) = \tfrac{1}{2}e^\top H_T e + O(\|e\|^3), \tag{55}$$

$$g(\theta_T + e) = H_T e + O(\|e\|^2), \tag{56}$$

$$h(\theta_T + e) = O(\|e\|^2), \tag{57}$$

$$\mathcal{A}(\theta_T + e) = e^\top H_T^2 e + O(\|e\|^3). \tag{58}$$

The full-logit forward KL has the same leading Hessian at the same-model root.

Proof. Lemma B.2 gives $\nabla^2\mathcal{L}(\theta_T) = H_T$. The Lipschitz Hessian condition gives Eq (55) and $\nabla\mathcal{L}(\theta_T + e) = H_T e + O(\|e\|^2)$. Lemma B.3 gives Eq (56) directly from the policy scores. The identity $h = \nabla\mathcal{L} - g$ cancels their linear terms, proving Eq (57). Multiplying the expansions for $\nabla\mathcal{L}$ and g gives Eq (58). At equality of the two policies, the conditional forward and reverse KL Hessians both equal the policy Fisher, yielding the final statement. □

This expansion is centered at the teacher and uses H_T. Using H_0 as the reference requires controlling $\|H_T - H_0\|_{\mathrm{op}}$ in addition to the Taylor remainder. Proposition B.4 gives that comparison explicitly on the local ball. Lemma B.5 then supplies a validity interval for these estimates along the original nonlinear OPD flow.

D.1 Fixed-Fisher specialization

Suppose the local model is exactly quadratic with the initial Fisher:

$$\mathcal{L}(\theta_T + e) = \tfrac{1}{2} e^\top H_0 e, \qquad g(\theta_T + e) = \nabla \mathcal{L}(\theta_T + e) = H_0 e. \tag{59}$$

Then $\dot{e} = -H_0 e$ and $e(0) = -\Delta$. Let $H_0 u_i = \lambda_i u_i$ and $c_i = -u_i^\top \Delta$. The exact solution and loss are

$$e(t) = \sum_i c_i e^{-\lambda_i t} u_i, \tag{60}$$

$$\mathcal{L}(\theta(t)) = \tfrac{1}{2} \sum_i \lambda_i c_i^2 e^{-2\lambda_i t}, \tag{61}$$

$$\gamma(t) = \frac{\sum_i \lambda_i^2 c_i^2 e^{-2\lambda_i t}}{\sum_i \lambda_i c_i^2 e^{-2\lambda_i t}}, \qquad \mathcal{L}(\theta(t)) > 0. \tag{62}$$

For normalized weights $\omega_i(t) \propto \lambda_i c_i^2 e^{-2\lambda_i t}$,

$$\dot{\gamma}(t) = -2 \operatorname{Var}_{\omega(t)}(\lambda) \le 0. \tag{63}$$

Higher-curvature components decay first, so this decline need not represent a failure to fit the teacher if the remaining loss is already small.

For any cutoff $\delta > 0$, splitting the spectral sum yields

$$\mathcal{L}(\theta(t)) \le E_\Delta(\delta) + e^{-2\delta t} \mathcal{L}(\theta_0). \tag{64}$$

If $E_\Delta(\delta) \le \varepsilon/2$, then loss at most ε is reached once $t \ge (2\delta)^{-1} \log \max\{1, 2\mathcal{L}(\theta_0)/\varepsilon\}$, provided the quadratic model remains valid until that time. The residual is computed directly from the initial parameter difference and also satisfies $E_\Delta(\delta) \le \delta \|\Delta\|^2 / 2$.

Equal distances can have different relative decay rates. Take $H_0 = \operatorname{diag}(1, \eta)$ with $0 < \eta < 1$, and compare parameter differences $\Delta_1 = (d, 0)^\top$ and $\Delta_2 = (0, d)^\top$ for $d > 0$. Both have norm d, but Eq (61) gives

$$\frac{F^{(1)}(t)}{F^{(1)}(0)} = e^{-2t}, \qquad \frac{F^{(2)}(t)}{F^{(2)}(0)} = e^{-2\eta t}. \tag{65}$$

Thus even in the exact quadratic case, an arbitrarily small parameter distance does not specify a uniform relative convergence rate. The direction of the parameter difference matters alongside its magnitude.

E EXPERIMENTAL DETAILS FOR THE TEACHER COMPARISONS

Figure layout. Each column of Figure 1 fixes the student named in its heading and identifies the teachers in its color key. The top row shows macro-averaged validation accuracy; the bottom row shows the recorded OPD training loss. Solid curves show the student runs, and colors identify the corresponding teachers. The reference lines, loss statistic, and comparison horizons are detailed below.

Models and training data. Table E.1 lists the three student/task groups, the seven teacher conditions, and the OPD training datasets used in Figure 1. For code, we distill Qwen3-4B-RL-Code (Yang et al., 2026) and Qwen3-14B (Yang et al., 2025) into Qwen3-4B. For math, we distill Qwen3-4B-RL-Math (Yang et al., 2026) and Qwen3-4B (Yang et al., 2025) into Qwen3-1.7B, and JustRL-1.5B (He et al., 2025a), Skywork-7B (He et al., 2025b), and R1-Distill-7B (Guo et al., 2025) into R1-Distill-1.5B. All Qwen3 models use non-thinking mode, and R1-Distill denotes DeepSeek-R1-Distill-Qwen. Qwen3-4B-RL-Code and Qwen3-4B-RL-Math are obtained by RL post-training Qwen3-4B (Non-thinking) on code and math tasks, respectively (Yang et al., 2026). JustRL-1.5B and Skywork-7B are obtained by RL post-training R1-Distill-1.5B and R1-Distill-7B, respectively, on math tasks (He et al., 2025a;b). For Qwen3-4B-RL-Code and Qwen3-4B-RL-Math, we use the released step-300 and step-500 checkpoints, respectively. The table separately identifies the teacher's RL training data in the note and the student's OPD prompts in the last column. In particular, Qwen3-4B-RL-Math is trained on filtered DeepMath, while both math OPD groups use DAPO-Math-17K.

The short names used in the text expand as follows: Qwen3-4B-RL-Code and Qwen3-4B-RL-Math denote Qwen3-4B-Non-Thinking-RL-Code and Qwen3-4B-Non-Thinking-RL-Math; R1-Distill-1.5B and R1-Distill-7B denote DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B; JustRL-1.5B denotes JustRL-DeepSeek-1.5B; and Skywork-7B denotes Skywork-OR1-Math-7B. Qwen3-4B-RL-Code and JustRL-1.5B are self-RL teachers for their corresponding students, whereas Qwen3-4B-RL-Math and Skywork-7B are RL-trained models larger than their students.

Table E.1: Student–teacher groups and OPD training data. Qwen3 models (Yang et al., 2025) use non-thinking mode; R1-Distill abbreviates DeepSeek-R1-Distill-Qwen (Guo et al., 2025).

Task / student	**Teachers**	**OPD data**
Code / Qwen3-4B (Non-thinking)	Qwen3-14B (Non-thinking) Qwen3-4B-RL-Code (Yang et al., 2026)	Eurus-2-RL-Data (code) (Cui et al., 2025)
Math / Qwen3-1.7B (Non-thinking)	Qwen3-4B (Non-thinking) Qwen3-4B-RL-Math (Yang et al., 2026)	DAPO-Math-17K (Yu et al., 2025)
Math / R1-Distill-1.5B	R1-Distill-7B JustRL-DeepSeek-1.5B (He et al., 2025a) Skywork-OR1-Math-7B (He et al., 2025b)	DAPO-Math-17K

RL provenance. Qwen3-4B-RL-Code and Qwen3-4B-RL-Math are GRPO checkpoints of Qwen3-4B trained on $\sim$25K Eurus coding problems and 57K DeepMath problems (He et al., 2026) (difficulty ≥ 6), respectively. JustRL-1.5B and Skywork-7B are RL-post-trained from R1-Distill-1.5B and R1-Distill-7B, respectively.

Training configurations. Table E.2 lists the settings for all seven main student–teacher runs, verified against their saved training configurations. The two Qwen3-1.7B runs share the same DAPO-Math-17K prompts and validation protocol; validation includes the initial checkpoint and every ten updates.

Validation and teacher reference lines. Code validation covers HumanEval+ and MBPP+ (Liu et al., 2023) and LiveCodeBench (Jain et al., 2025); math validation covers AIME 2024, AIME 2025, and AMC 2023. We report the unweighted mean of the three benchmark scores, using $\mathrm{avg@4}$

Table E.2: Training configurations for the seven main OPD runs. Each teacher condition is one training run; the three columns contain two, two, and three teacher conditions, respectively.

Setting	**Code** Qwen3-4B (Non-thinking)	**Math** Qwen3-1.7B (Non-thinking)	**Math** R1-Distill-1.5B
OPD objective	Reverse KL, Eq (1)		
Vocabulary approximation	Student's top 16 tokens		
Token weights	Student probabilities		
Loss normalization	Global mean over valid response tokens		
Optimizer	AdamW	AdamW	AdamW
Learning rate	10^{-6}	10^{-6}	10^{-6}
Learning-rate schedule	constant	constant	constant
Warmup fraction	0	0	0
Adam (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$
Weight decay	0.01	0.01	0.01
Gradient clipping (ℓ_2 norm)	1	1	1
Prompt batch size	64	64	64
Optimization minibatch size	64	64	64
Training rollouts per prompt	1	1	1
Optimization epochs per batch	1	1	1
Maximum prompt tokens	2,048	1,024	1,024
Maximum response tokens	16,384	7,168	21,000
Student sampling temperature	1	1	1
Teacher temperature	1	1	1
Student sampling top-p	1	1	1
Configured training updates	200	200	200
Validation interval (updates)	10	10	10
Checkpoint interval (updates)	10	10	10
Common comparison endpoint	200	200	200

Student rollouts provide the prefixes; the teacher is fixed during training. Merged entries apply to all three groups. The configured budget is 200 updates in every run, and all seven comparisons use the complete 200-update window. The Code run with Qwen3-4B-RL-Code is the completed reproduction described in the comparison-scope discussion below.

for code and avg@8 for math. These quantities average sampled outcomes; they are not the probability that at least one of k samples is correct, commonly denoted `pass@k`. The Code and R1-Distill standalone teacher references use the benchmarks, decoding parameters, and response-length limits of their corresponding student evaluations. For the completed Qwen3-4B-RL-Code run, the saved validation outputs contain eight responses per problem. We select four without replacement using a fixed seed of 0 and recompute avg@4. Selection depends only on the benchmark, prompt identity, and response index, not on correctness; each problem uses the same four response indices at every checkpoint. Color-matched dashed lines in Figure 1 show standalone teacher accuracy, with each reference line labeled directly. They provide reference levels, not upper bounds on student performance. For the Qwen3-1.7B math runs, validation uses 30 AIME 2024 problems, 30 AIME 2025 problems, and 83 AMC 2023 problems, with eight responses per problem, temperature 0.7, top-$p = 0.95$, and a response limit of 31,744 tokens. Each checkpoint therefore evaluates 1,144 responses over 143 distinct problems. The two Qwen-math teachers are evaluated under this same protocol, giving macro accuracies of 69.17% for Qwen3-4B-RL-Math (AIME 2024 63.75, AIME 2025 56.25, AMC 2023 87.50) and 35.13% for Qwen3-4B Non-thinking (25.42, 19.58, 60.39); the earlier archived scores under a 7,168-token limit are not used.

Training-loss statistic. The plotted loss is the global token-mean student-top-16 OPD surrogate: the negative mean of the top-K advantage summed over K and averaged over valid response tokens. We use the logged statistic `critic/advantages/mean_sum_over_k` with its sign reversed, which removes the packed-microbatch scaling present in `actor/pg_loss`. This implementation-level surrogate is distinct from the full-vocabulary KL objective analyzed in Section 5. The training signal $\widehat{\mu}_m$ of Section 5.2 uses the same $\widehat{\mathcal{L}}_m$ together with the logged pre-clip gradient norm `actor/grad_norm`. In Table E.3, the signal before the split step s is the median of $\widehat{\mu}_m$ over steps 1 to s, the signal after s is the median over steps $s+1$ to $s+30$, the last-50-steps column is the median over the run's last 50 steps, and the loss left at s is the centered 9-step rolling median of $\widehat{\mathcal{L}}_m/\widehat{\mathcal{L}}_1$ (the smoothed curve of Figure 3) at step s. The signal gaps quoted in Section 5.2 are ratios of the after-s medians of the runs with self-RL and larger-scale teachers. Figure 1 draws the recorded $\widehat{\mathcal{L}}_m$ faintly and overlays its centered 9-step rolling median, without normalization; all loss statistics use the raw records. Every loss panel displays its own numerical tick labels; vertical limits vary by student/task group. Fractional loss reduction is reported separately in Table 2.

Learning signal over training steps. Figure 3 shows the data of Figure 2 against the training step, and Table E.3 summarizes the learning signal before and after a split step s.

Table E.3: Learning signal $\widehat{\mu}_m$ around a split step s. For each run of Figure 3 we report the median learning signal $\widehat{\mu}_m$ before the split step s (steps 1 to s), in the 30 steps after s, and in the last 50 steps of the run; the drop is the ratio of the first two; loss left is the smoothed remaining loss $\widehat{\mathcal{L}}_m/\widehat{\mathcal{L}}_1$ at step s. Here $s = 15$ for Code and $s = 30$ for Math. † marks self-RL teachers.

Teacher	Before s	After s	Drop	Loss left at s	Last 50 steps
Code, $s = 15$					
Qwen3-4B-RL-Code†	272	40.9	6.7×	12.6%	2.37
Qwen3-14B	60.7	5.05	12.0×	98.8%	3.78
Math, $s = 30$					
JustRL-1.5B†	7.11	3.37	2×	73.0%	0.56
Skywork-7B	3.73	0.38	10×	86.5%	0.090
R1-Distill-7B	1.14	0.121	9×	86.5%	0.056

Cross-fitted estimator of α. At each checkpoint m of Section 5.2, the eight rollouts per prompt are split into two folds. Let $g_m^{(0)}$ and $g_m^{(1)}$ be the detached top-16 loss gradients estimated from the two folds, and $h_m^{(0)}$ and $h_m^{(1)}$ the likelihood-ratio estimates of the occupancy gradient, each using the opposite fold to construct its per-prompt baseline. The estimators are

$$\widehat{g}_m^2 := \left\langle g_m^{(0)}, g_m^{(1)} \right\rangle, \qquad \widehat{X}_m := \tfrac{1}{2}\left(\left\langle h_m^{(0)}, g_m^{(1)} \right\rangle + \left\langle h_m^{(1)}, g_m^{(0)} \right\rangle\right), \qquad \widehat{\alpha}_m := \widehat{X}_m/\widehat{g}_m^2. \tag{66}$$

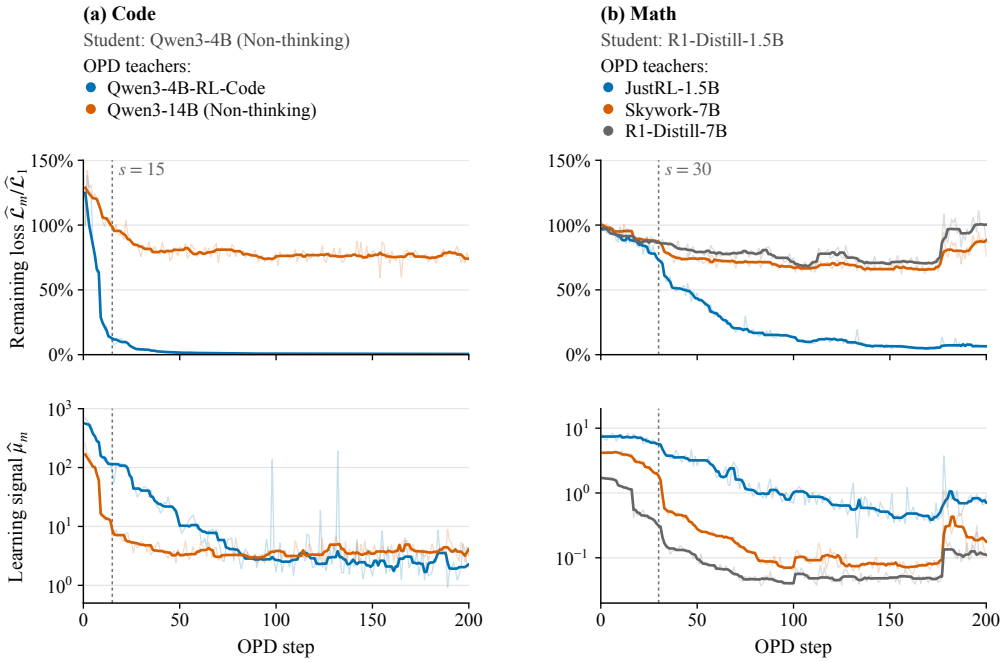


Figure 3: Learning signal and remaining loss over training steps. Left column: the Code comparison; right column: the Math comparison. Top: remaining OPD loss $\widehat{\mathcal{L}}_m/\widehat{\mathcal{L}}_1$. Bottom: learning signal $\widehat{\mu}_m$ on a log scale. Both settings cover all 200 updates. Lines are centered 9-step rolling medians of the per-step values, which are drawn faintly behind them. The dotted vertical line marks the split step s of Table E.3.

Conditional on the sampled prompts, the rollout folds are independent. Their cross-product removes the same-fold rollout-noise bias from $\widehat{g}_m^2$, but not uncertainty from prompt sampling. The baseline from the opposite fold does not bias $\widehat{X}_m$. Code uses 16 prompts and Math 64 prompts per checkpoint; Table 3 uses all 20 diagnostic checkpoints at steps $10, 20, \ldots, 200$ in both settings. The Code learning-signal curves use all 200 updates of the same runs as Figure 1: the completed Qwen3-4B-RL-Code reproduction and the original Qwen3-14B run. The Qwen3-14B occupancy statistics use all 20 checkpoints of that original run.

Metric reporting. Accuracy gain and teacher–student gap recovery follow the definitions in Section 4.2. Loss minima in Table 2 use raw updates 1–200 and need not be sustained. For the five larger-scale runs and two self-RL runs, mean maximum reductions are 41.6% and 97.9%, respectively; mean final reductions $(1 - \widehat{\mathcal{L}}_{200}/\widehat{\mathcal{L}}_1) \times 100\%$ are 25.1% and 96.2%.

Comparison scope. All seven main comparisons in Figure 1 and Table 2 use 200 training updates. For Code/Qwen3-4B, the Qwen3-4B-RL-Code curve comes from an independent reproduction on two H200 GPUs, with the same OPD hyperparameters and native checkpoint resumes. It replaces the earlier run that stopped at update 149; the two runs are not concatenated. Qwen3-14B uses its complete original 200-update run. Code accuracy uses four responses per problem, including the fixed-seed subsampling described above. Validation is recorded at steps $0, 10, \ldots, 200$ for the Qwen3-4B-RL-Code reproduction and the two Math/Qwen3-1.7B runs. The Qwen3-14B and three Math/R1-Distill-1.5B runs originally begin validation at step 10. We supplement their curves with evaluations of the same initial student under the corresponding validation protocol: Code shares the step 0 evaluation from the Qwen3-4B-RL-Code reproduction, and R1-Distill-1.5B shares a before-training evaluation from a separate full-logit OPD run with JustRL-1.5B. These are shared measurements of the initial model, not additional evaluations within each training run. The two Math/Qwen3-1.7B curves retain their own before-training evaluations of the same base checkpoint; their step 0 estimates differ slightly because of sampling. In all seven comparisons, accuracy gains and gap recovery use the base student model at step 0 as $\mathrm{Acc}_{\mathrm{start}}$. The loss and validation panels within each group both extend to step 200. Each panel uses a linear scale that contains all its plotted observations. Validation accuracy and loss use column-specific vertical limits, so their absolute magnitudes are compared within the corresponding student/task group. Each teacher condition comprises one training run; the curves therefore do not provide repeated-seed uncertainty estimates.

Table E.4: Completed 200-update Qwen3-1.7B (Non-thinking) math runs. n is the number of training rollouts per prompt; validation always uses eight responses per problem. Accuracies are percentages and gain is in percentage points. Final loss reduction is $(1 - \widehat{\mathcal{L}}_{200}/\widehat{\mathcal{L}}_1) \times 100\%$, using each run's own objective. Each row is one training run.

Teacher	Support	n	Acc_0	Acc_{200}	Gain	Final loss reduction
Qwen3-4B-RL-Math	Top-16	1	21.10	43.94	22.83	55.37%
Qwen3-4B (Non-thinking)	Top-16	1	20.66	22.74	2.07	6.24%
Qwen3-4B-RL-Math	Full logits	1	20.55	46.36	25.81	54.85%
Qwen3-4B (Non-thinking)	Full logits	1	21.66	22.88	1.22	11.60%
Qwen3-4B (Non-thinking)	Top-16	8	21.85	24.87	3.02	9.89%

Additional completed Qwen-math comparisons. Table E.4 includes the two main top-16 runs, two full-logit runs with one training rollout per prompt, and a top-16 run with eight training rollouts per prompt for the Qwen3-4B teacher. All rows use the Qwen-math training and validation limits described above. The full-logit runs replace top-16 support with all 151,936 student vocabulary entries. Each loss reduction is computed using that run's own objective at updates 1 and 200; top-16 and full-logit loss magnitudes are not identical objectives. With full logits, Qwen3-4B-RL-Math yields a 25.81-point accuracy gain and a 54.85% loss reduction, while Qwen3-4B yields a 1.22-point gain and an 11.60% loss reduction. The teacher ordering observed with top-16 OPD therefore also appears with full logits in these runs. For the Qwen3-4B teacher, the eight-rollout run gains 3.02 accuracy points and reduces its loss by 9.89%. This additional single run does not establish a

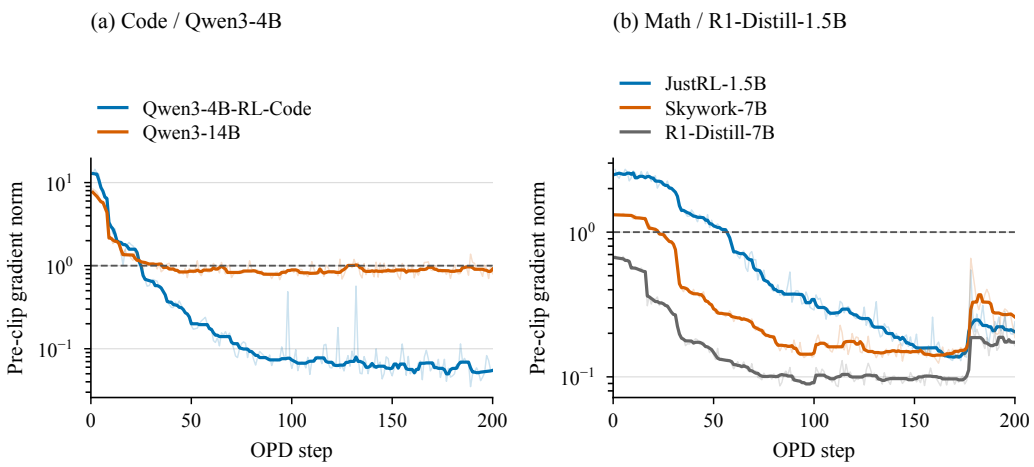


Figure 4: Pre-clipping gradient norms in the five runs used for the learning-signal analysis. Curves use a logarithmic vertical axis. Solid lines are centered 9-step rolling medians; faint lines show all 200 recorded values. The dashed line marks the clipping threshold of 1. These are norms of the training gradients, not norms of the AdamW parameter updates.

general effect of rollout count across teachers. The additional runs are excluded from the seven-run averages in Section 4.3.

Scope of the additional experiments. The seven main OPD runs are supplemented by eight Math runs that vary vocabulary support or the number of training rollouts (Appendix H.2). Three of these eight runs already appear in Table E.4; they are not additional replications. Appendix H.1 reports four SFT runs and two subsequent OPD runs initialized from SFT checkpoints. Appendix G reports two instruction-following OPD runs. These comparisons use the specified single-run configurations; they do not constitute repeated-seed estimates. Checkpoint diagnostics and standalone model evaluations are reported separately from training runs.

E.1 Additional Training-Dynamics Diagnostics

Gradient norms and clipping. Figure 4 separates the gradient norm from the loss-normalized learning signal in Figure 3. We use the logged pre-clipping norm at each of the 200 optimizer updates. The norm exceeds the clipping threshold of 1 at 24 updates with Qwen3-4B-RL-Code, 64 with Qwen3-14B, 56 with JustRL-1.5B, 22 with Skywork-7B, and none with R1-Distill-7B. These counts describe when clipping is active; they do not identify the effective AdamW update, which also depends on the optimizer's moment estimates and weight decay. Accordingly, the ratio $\widehat{\mu}_m = \widehat{G}_m^2/(2\widehat{\mathcal{L}}_m)$ is an empirical diagnostic rather than a direct measurement of the population continuous-time loss derivative. A small gradient norm alone does not distinguish successful matching from a plateau; the remaining loss must also be considered. All recorded losses in these five diagnostic runs are positive. The truncated objective is not a nonnegative full-vocabulary KL in general; Appendix G reports a run in which it becomes negative and does not interpret the corresponding ratio as a loss-decay rate.

Diagnostic sampling. The occupancy measurements are separate evaluations of saved checkpoints, not additional training runs. The prompt subset is fixed across the 20 checkpoints at steps $10, 20, \ldots, 200$, with eight sampled responses per prompt at each checkpoint. The Code subset contains 16 Eurus coding prompts; the Math subset contains the first 64 DAPO-Math-17K prompts after deterministic overlength filtering, without shuffling. The Math diagnostic uses sampling seed 42 and FP32 model computations. Table E.5 summarizes the sampling budget. Four responses per prompt enter each fold of Eq (66); the opposite fold supplies the per-prompt baseline for the occupancy gradient.

Table E.5: Sampling settings for checkpoint-level occupancy diagnostics. These rollout counts differ from the one-rollout training setting.

Setting	Code	Math
Student	Qwen3-4B	R1-Distill-1.5B
Prompt source	Eurus code	DAPO-Math-17K
Prompts per checkpoint	16	64
Responses per prompt	8	8
Responses per fold	64	256
Maximum response length	16,384	21,000
Student vocabulary support	Top 16	Top 16
Checkpoints per teacher	20	20

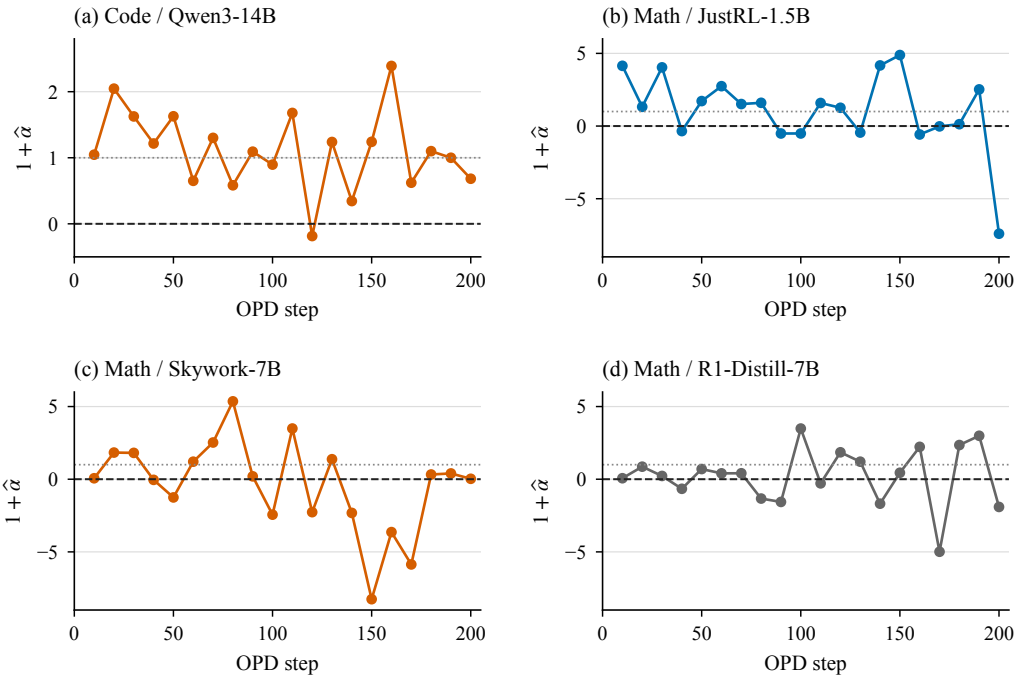


Figure 5: Checkpoint-level occupancy factor $1 + \widehat{\alpha}_m$. Each point is one cross-fitted estimate, and lines connect consecutive checkpoints without smoothing. The dashed line is zero, the cancellation threshold of the idealized flow; the dotted line is one, corresponding to zero occupancy contribution. The Code panel uses a different vertical range from the three Math panels. No confidence intervals are implied.

Occupancy factor over training. Figure 5 shows every measured $1 + \widehat{\alpha}_m$, including the self-RL teacher JustRL-1.5B as a reference. All 80 cross-fold estimates of $\widehat{g}_m^2$ are positive, so no checkpoint is removed on account of a nonpositive denominator. For the larger-scale teachers, $1 + \widehat{\alpha}_m$ is positive at 19/20 checkpoints with Qwen3-14B, 12/20 with Skywork-7B, and 13/20 with R1-Distill-7B, reproducing Table 3. It is also positive at 13/20 checkpoints with JustRL-1.5B. The checkpoint estimates fluctuate and can become negative even in the self-RL run. They therefore do not establish the sign of the population loss derivative at every update. In particular, these finite-sample ratios should not be read as confidence intervals or as deterministic predictions of the subsequent AdamW trajectory. Their role is to test whether persistent occupancy cancellation is shared by the observed plateaus.

F Parameter and Representation Measurement Details

Checkpoint scope and parameter statistics. The eight OPD endpoints in Table 4 are full-parameter, student-top-16 reverse-OPD checkpoints at step 200. The three R1-Distill-1.5B runs use the same clean single-rollout runs as Figure 1: JustRL-1.5B, Skywork-7B, and R1-Distill-7B. The Qwen3-4B endpoints use Qwen3-14B Non-thinking and Qwen3-4B-RL-Code on Code. The Qwen3-4B-RL-Code endpoint comes from a 200-step reproduction on two H200 GPUs; the original run stopped at update 149. The two Qwen3-1.7B endpoints come from the Math runs of Figure 1, with teachers Qwen3-4B-RL-Math and Qwen3-4B. The IF endpoint comes from the supplementary run of Appendix G with the self-RL teacher UltraData-IF-1.5B. All parameter statistics compare exported model checkpoints, compute tensor differences in FP32, and count tied embedding/head weights once. The global displacement aggregates the squared differences and squared starting weights before taking their ratio and square root; it is not an unweighted average of per-layer relative changes.

Teacher–student parameter distance. The Teacher columns of Table 4 report the relative parameter distance $\|\theta_T - \theta_0\|_2 / \|\theta_0\|_2$ between each self-RL teacher and its student, which Section 5.3 uses. We compute it with the same protocol as d_{par} above, comparing the released teacher checkpoint with the student checkpoint used in our runs. Most parameter entries are identical in BF16, namely 87.7% for Qwen3-4B-RL-Code, 34.4% for JustRL-1.5B and 83.4% for UltraData-IF-1.5B. In the JustRL-1.5B run, the OPD student moves by only 0.098%, much less than the teacher's distance of 0.50%. OPD thus reduces the loss to 3.7% of its starting value without moving the student to the teacher's parameters.

Representation extraction and CKA. Each checkpoint pair uses identical tokenizer vocabulary and prompt token IDs. The Math bank contains AIME24 (30 prompts), AIME25 (30), and AMC23 (83); the Code bank contains HumanEval+ (164), MBPP+ (378), and LiveCodeBench (131). A BF16 forward pass is performed separately for each prompt, without response generation. At each hidden-state index, token states are mean-pooled and stored in FP32. For the resulting representation matrices X and Y, centered across prompts, we compute

$$\mathrm{CKA}(X, Y) = \frac{\left\|Y^\top X\right\|_F^2}{\|X^\top X\|_F \, \|Y^\top Y\|_F} \tag{67}$$

using FP64 accumulation (Kornblith et al., 2019). The Teacher columns of Table 4 use the same protocol and the prompt bank of the corresponding base model, so UltraData-IF-1.5B is also compared on the Math bank. The IF endpoint is likewise compared on the R1-Distill-1.5B Math bank. The Qwen3-1.7B endpoints use the Qwen3 Math bank, since Qwen3-1.7B and Qwen3-4B share the same tokenizer. The node that holds the Qwen3-4B-RL-Math endpoint had no free GPU, so its CKA is computed on CPU with standard attention instead of FlashAttention. On the Qwen3-4B endpoint, the two implementations differ by at most 3×10^{-4} in layerwise CKA. The figures report full-bank estimates. Existing stability checks use 20 bootstrap resamples, each drawing 100 prompts from the pooled bank; they quantify prompt-resampling variation rather than training-seed uncertainty. Subset variation is appreciable in Code: the minimum CKA is 0.9587 on HumanEval+, 0.9940 on MBPP+, and 0.8808 on LiveCodeBench. The LiveCodeBench minimum has bootstrap standard deviation 0.0459, so the pooled and subset estimates should be interpreted at their respective levels of aggregation. High CKA concerns the geometry of the pooled prompt representations rather than equality of individual activations, since CKA is invariant to orthogonal transformations and isotropic scaling (Kornblith et al., 2019). It also does not establish a fixed neural tangent kernel, whose use for language-model fine-tuning requires additional conditions (Malladi et al., 2023).

Layerwise profiles. Figure 6 expands the minimum-CKA summary in Table 4 into the complete layerwise profiles. Panels (a)–(c) contain the same eight step-200 OPD endpoints, and panel (d) compares the three self-RL teachers with their respective initial students. The horizontal coordinate divides the hidden-state index by its maximum, including the embedding and final normalized states. Qwen3-4B provides 37 hidden-state entries and the two smaller student architectures provide 29. We use the same pooled prompt banks and checkpoint comparisons described above, without smoothing or resampling the measured hidden-state indices.

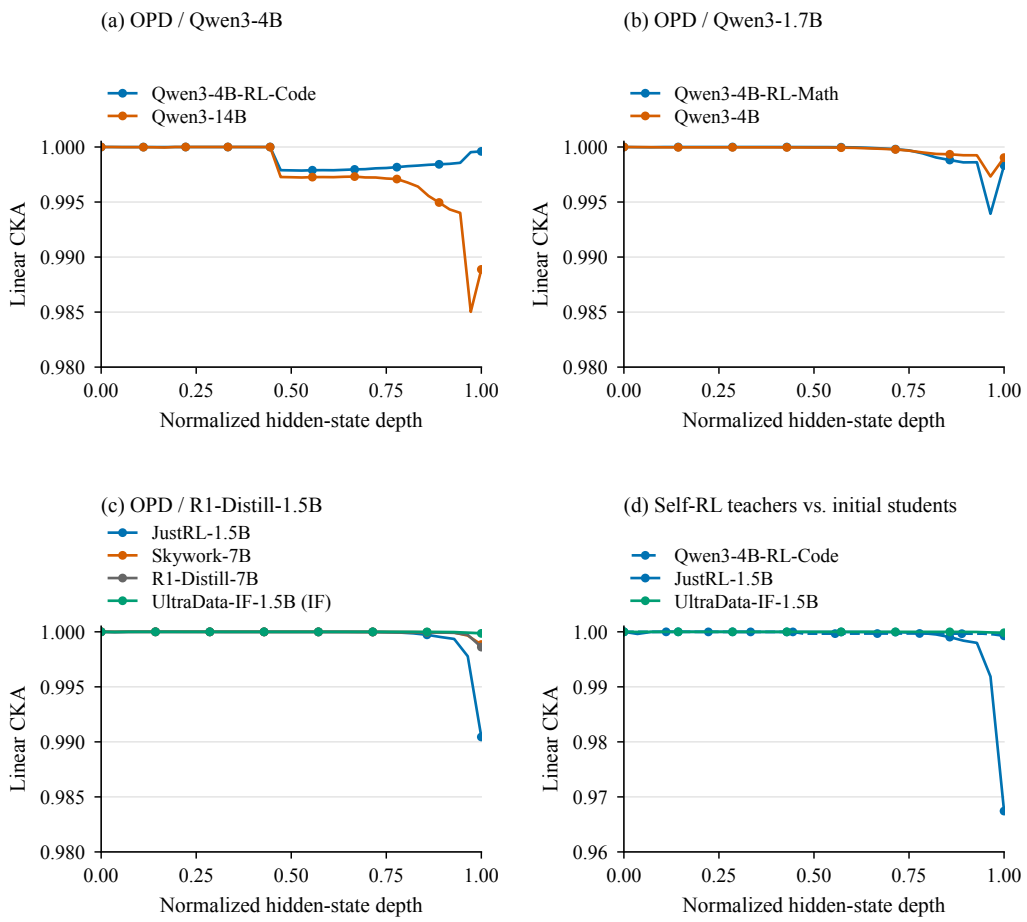


Figure 6: Layerwise linear CKA for the checkpoints in Table 4. Panels (a)–(c) compare each OPD student after 200 updates with its initialization; the legend names the teacher. Panel (d) compares self-RL teachers with their initial students, using a wider vertical range. All R1-Distill-1.5B comparisons, including the IF endpoint and UltraData-IF teacher, use the same pooled Math prompt bank. Lines connect measured hidden-state entries; markers are placed every four entries for readability.

The plots expose changes that a single minimum cannot localize: for example, the minimum CKA of the Qwen3-14B Code run is 0.9850, whereas the Qwen3-4B-RL-Code OPD endpoint stays above 0.9978. The self-RL teacher JustRL-1.5B reaches a minimum of 0.9674 relative to its initial student, while its OPD endpoint has minimum 0.9904. These are endpoint comparisons, not a time series of representation learning, and high CKA does not show that the teacher's behavior has been reproduced.

G OPD ON INSTRUCTION FOLLOWING

Models and data. We compare two teachers for R1-Distill-1.5B: the self-RL teacher UltraData-IF-1.5B, obtained by further RL training of the student on instruction-following data (Fu et al., 2026b), and the larger R1-Distill-7B. The former is the instruction-following pair included in Table 4; the latter is an additional comparison reported here. Both OPD runs use the first-turn prompts of all 4,501 conversations in Multi-IF (He et al., 2024). All prompts remain after the 4,096-token length filter. Evaluation uses the 4,445 conversations with all three turns: 896 English, 518 French, 529 Hindi, 515 Russian, 454 Chinese, 493 Italian, 524 Portuguese, and 516 Spanish conversations. All evaluation conversations' first-turn prompts occur in the training set. These results measure transfer on overlapping inputs, not held-out generalization; later-turn prompts are not used for OPD training.

Training. Both runs start from the same base checkpoint and use the settings in Table G.1. They use the student's top-16 tokens with renormalized student weights and the original teacher–student log-ratio, as in Eq (2), averaged over valid response tokens. The instruction-checking reward is logged as a diagnostic and is not mixed into the distillation update. Each training update uses 64 prompts and eight rollouts per prompt, so the optimizer processes 512 responses in one update. The separate three-turn evaluation uses one response per turn.

Table G.1: Shared configuration of the two instruction-following OPD runs. Both runs use full-parameter updates on two H200 GPUs.

Setting	**Value**
Optimizer	AdamW, $\beta = (0.9, 0.999)$
Learning rate / schedule	10^{-6} / constant, no warmup
Weight decay / gradient clip	0.01 / ℓ_2 norm 1
Prompt batch / configured minibatch (prompts)	64 / 64
Rollouts per prompt / optimization epochs	8 / 1
Training temperature / top-p	1 / 1
Teacher temperature	1
Maximum prompt / response tokens	4,096 / 7,168
Prompt order	Fixed, no shuffling
Updates / checkpoint interval	200 / 10
Evaluation responses per turn	1 (seed 0)
Evaluation temperature / top-p	0.6 / 0.95
Maximum evaluation response / context	16,384 / 32,768 tokens

Evaluation and score aggregation. We evaluate the base student, both standalone teachers, and every tenth OPD checkpoint on the same three-turn conversations. Each subsequent turn includes the model's preceding answers with completed reasoning blocks removed; scoring also uses the text after the final `</think>` tag when present. Responses without this closing tag are scored as generated. Overlong dialogue contexts receive an empty response rather than a truncated history. Responses that reach the output limit are retained for scoring.

For each language and turn, the Multi-IF instruction checkers provide four rates: strict and loose prompt-level success, and strict and loose instruction-level success. Prompt-level success requires all instructions to pass; instruction-level success divides the total passed instructions by the total number of instructions. Loose checks additionally allow variants removing boundary lines or asterisks. The reported *three-turn composite score* is the unweighted mean of these four rates over all eight languages and all three turns, expressed as a percentage. Thus, it averages 96 language–turn–metric rates rather than pooling conversations across languages. This is a composite benchmark score, not a single accuracy rate or an avg@8 estimate. Gain is the change from the common base-student score at step 0; gap recovery divides this gain by the corresponding teacher–base score gap. The configured single-turn diagnostic validation loop is disabled; all reported evaluation scores come from this separate three-turn protocol.

Table G.2: Instruction-following outcomes after 200 OPD updates. Top: raw signed top-16 objective at update 1, its minimum over updates 1–200, and its final value. Bottom: three-turn composite scores (%), gain from the shared base student (pp), and score-gap recovery (%). † marks the self-RL teacher.

Teacher	Initial objective	Minimum	Final
UltraData-IF-1.5B†	0.00852	−0.00134	0.00124
R1-Distill-7B	0.78025	0.09804	0.11119

Teacher	Base	OPD	Teacher score	Gain	Recovery
UltraData-IF-1.5B†	30.97%	40.86%	41.38%	+9.88	95.0%
R1-Distill-7B	30.97%	27.43%	47.72%	−3.54	−21.2%

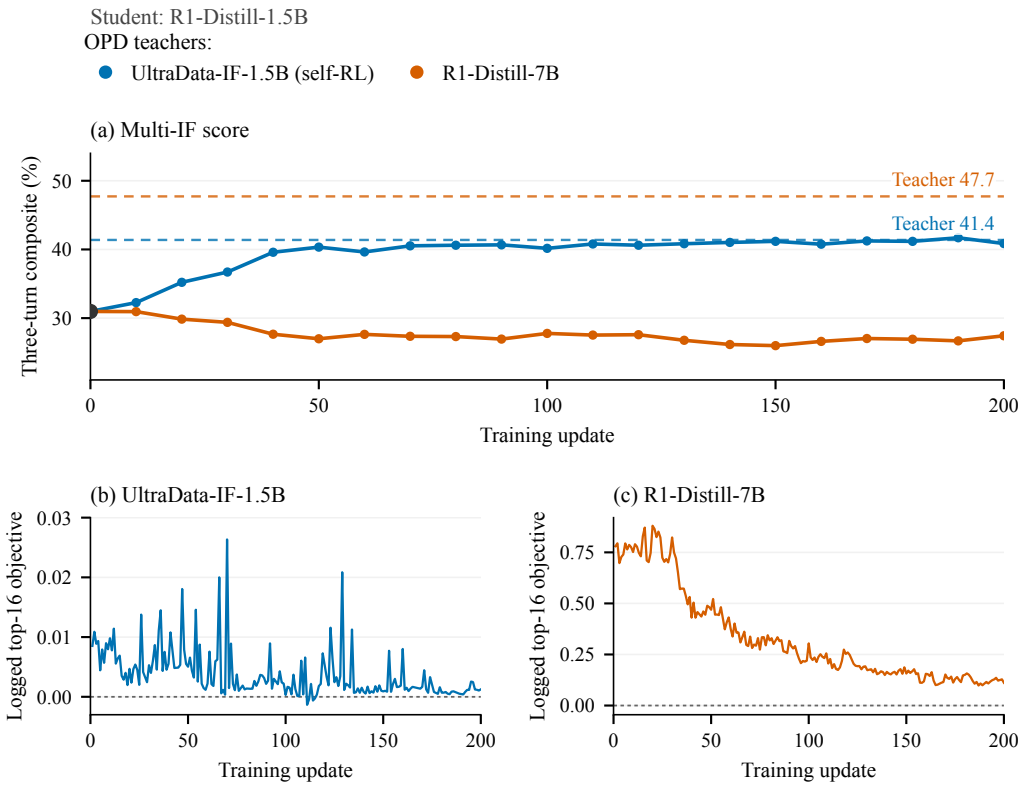


Figure 7: Instruction-following dynamics over 200 updates. (a) Three-turn composite scores; the black point is the shared initial student, and dashed lines are the corresponding standalone teachers. (b,c) Raw signed top-16 objectives, without smoothing, on separate linear axes that include zero. Colors identify the teacher used for OPD.

Results and interpretation. With the self-RL teacher, the three-turn composite score rises from 30.97 at initialization to 40.86 at update 200, compared with the teacher's 41.38, closing 95.0% of the score gap. With R1-Distill-7B, the score falls to 27.43 despite the teacher's stronger standalone score of 47.72 and a substantial decrease in the logged objective (Table G.2, Figure 7). The third-turn composite score follows the same pattern: 22.31 initially, 29.02 with UltraData-IF-1.5B, and 17.62 with R1-Distill-7B; the teachers score 29.45 and 35.84, respectively.

The top-16 surrogate in Eq (2) need not be nonnegative: its weights are renormalized on the retained support, but its log-ratio uses the original distributions. The UltraData run has four negative observations, with minimum -0.00134 at update 111. We therefore report its raw objective values rather than interpreting a percentage reduction above 100% as recovery of the full-vocabulary KL. The two runs illustrate that reducing this surrogate does not by itself ensure improvement in instruction following. Each curve is a single training run with one sampled dialogue per evaluation conversation, and the overlapping training and evaluation inputs limit conclusions about generalization.

H Ablations

H.1 SFT on Teacher Responses and Subsequent OPD

We evaluate SFT for four Math student–teacher pairs from Table 1: R1-Distill-1.5B learns from JustRL-1.5B or R1-Distill-7B, and Qwen3-1.7B (Non-thinking) learns from Qwen3-4B-RL-Math or Qwen3-4B (Non-thinking). We then continue OPD for 200 updates from the final SFT checkpoints associated with R1-Distill-7B and Qwen3-4B (Non-thinking), using the same respective teachers. Each setting is one training run. The Qwen3-4B-RL-Math teacher is its RL update-500 checkpoint. SFT, direct OPD, and SFT→OPD have different training budgets; the last includes the entire SFT stage before its 200 OPD updates.

Teacher-response collection. We use the project's DAPO-Math-17K training file, containing 17,917 rows and 17,398 distinct question texts. We preserve each question body, replace the source file's answer-format instructions with *Please reason step by step, and put your final answer within* `\boxed{}`., and apply the teacher's chat template. Each teacher contributes 20,000 accepted responses. Since there are fewer than 20,000 distinct questions, collection traverses deterministically shuffled cycles of the source rows, with shuffle seed $42 +$ cycle and fresh generations on repeated questions. The candidate pool contains four cycles (71,668 slots). We draw one response per attempt, retry a candidate slot at most three times, and continue through the candidate pool until the accepted-response quota is reached.

Teacher generation uses temperature 0.7, top-p 0.95, no top-k truncation, repetition penalty 1, a 12,288-token response cap, and a 32,768-token context limit. Two vLLM workers use seeds 42 and 43, with generation batches of 128. The Qwen teachers generate in non-thinking mode; the R1 teachers retain their reasoning traces. We require the output to contain the `\boxed` marker and terminate normally, reject detected repetitive text, and additionally require a closing `</think>` tag for R1 responses. The repetition filters detect a stripped line of at least 20 characters appearing five times, a 100-character substring appearing three times when sampled every ten characters, or three consecutive repetitions of a 50-character block for outputs longer than 5,000 characters. We do not filter for answer correctness. A final student-tokenization check rejects examples longer than 14,336 tokens including prompt and response; all accepted examples fit this limit, so the admitted targets need not be truncated.

SFT objective and implementation. We perform full-parameter SFT with LLaMA-Factory, using cross-entropy on the teacher response and masking prompt and padding tokens. We use source revision `ac26e38` of LLaMA-Factory. Sequence packing and label smoothing are disabled. The recorded templates are `deepseekr1` for R1 and `qwen3` for Qwen, with `enable_thinking=True` in the SFT preprocessing configuration. For R1, a missing opening `<think>` tag is restored before training, preserving the generated reasoning and answer. For Qwen, the reasoning template prepends empty thinking tags to a response that has neither thinking tag; those tags are included in the supervised target. This preprocessing flag is distinct from the non-thinking teacher-generation and Qwen benchmark settings.

Table H.1: SFT response collection and row-split overlap. Every teacher supplies 20,000 accepted responses, split into 19,000 training and 1,000 held-out rows. *Distinct questions* counts the accepted pool. *Shared held-out rows* counts held-out examples whose question text also appears in the training split.

Teacher	Attempts	Distinct questions	Shared held-out rows
JustRL-1.5B	27,706	16,401	344
R1-Distill-7B	30,565	15,855	387
Qwen3-4B-RL-Math	32,034	15,481	424
Qwen3-4B (Non-thinking)	22,931	17,123	268

We use BF16 training, DeepSpeed ZeRO-2, FlashAttention-2, the Liger kernel, and gradient checkpointing on two H200 GPUs. Table H.2 gives the optimization settings. The response rows are split with seed 42 before preprocessing and shuffled during SFT. This is a row-level split: repeated questions can occur on both sides (Table H.1). Consequently, held-out cross-entropy is a diagnostic of fitting these teacher responses, not an independent test of mathematical generalization. We measure mathematical capability separately on the benchmarks below.

Table H.2: Optimization settings for all four SFT runs and the two subsequent OPD runs. OPD response limits differ by student; the other listed settings are shared within each stage.

Setting	SFT	Subsequent OPD
Training examples	19,000 response rows	DAPO training prompts
Optimizer updates	297 (one epoch)	200
Global batch	64 examples	64 prompts, one rollout each
Per-GPU SFT batch	8, accumulated 4 times	Dynamic token batching
Trainable weights	All	All
Optimizer	Fused AdamW	AdamW
Learning rate	Peak 10^{-5}	Constant 10^{-6}
Schedule / warmup	Cosine / 5%	Constant / none
Adam (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$
Adam ϵ	10^{-8}	Library default (not explicitly set)
Weight decay	0	0.01
Gradient clipping norm	1	1
Prompt + response limit	14,336 tokens total	1,024 prompt tokens; 21,000 response tokens (R1), 7,168 (Qwen)
Random / data seed	42 / 42	Not explicitly set in saved configuration
Loss recording	Every 5 updates	Every update
Evaluation during training	Row-heldout CE at 100, 200, 297	Math benchmarks at $0, 10, \ldots, 200$

SFT loss dynamics. Figure 8 shows all four runs. The first logged training losses at update 5 are 0.4108, 0.3711, 0.4262, and 0.2764 for JustRL-1.5B, R1-Distill-7B, Qwen3-4B-RL-Math, and Qwen3-4B (Non-thinking), respectively. The last logged training losses, at update 295, are 0.2951, 0.3459, 0.3229, and 0.2368. Training finishes at update 297; the corresponding final held-out losses are 0.2942, 0.3537, 0.3252, and 0.2332. These cross-entropies concern different teacher-response datasets and are not the OPD objective. The final checkpoint is used for benchmarking and, where applicable, subsequent OPD; checkpoints are not selected by benchmark accuracy.

Mathematical benchmark evaluation. We evaluate each final SFT checkpoint on 30 AIME 2024, 30 AIME 2025, and 83 AMC 2023 problems. We sample eight responses per problem

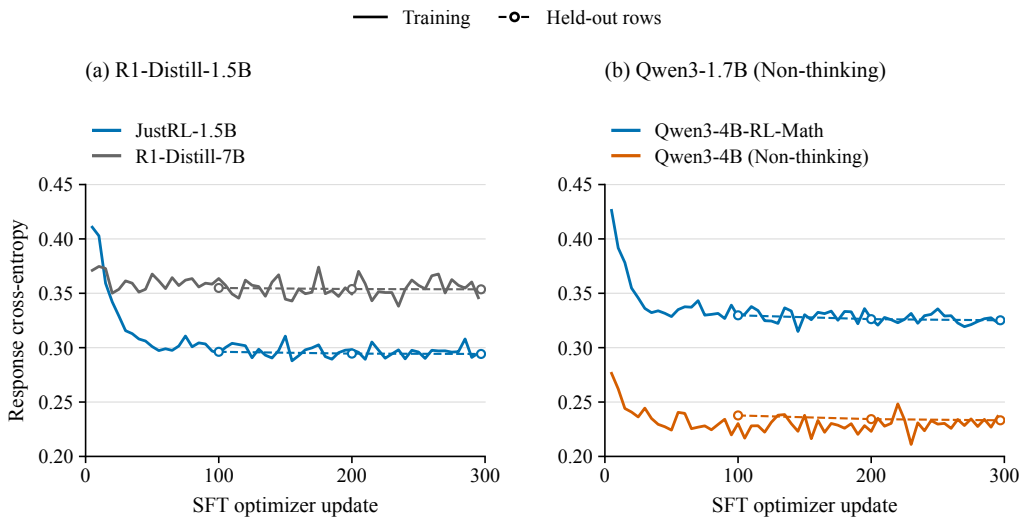


Figure 8: SFT loss dynamics for all four teacher-response datasets. Solid lines join recorded training cross-entropies at updates $5, 10, \ldots, 295$; no update-0 loss is available. Dashed lines and hollow circles mark held-out response-row cross-entropy at updates 100, 200, and 297. The held-out split has question overlap with training and is not a mathematical benchmark. Colors identify teachers as in Figure 1. Each teacher contributes one run; no averaging across seeds is shown.

(1,144 responses total), with temperature 0.7, top-p 0.95, and a 31,744-token response cap, 1,024-token prompt limit, and 32,768-token context limit. We score answer correctness and report the unweighted mean of the three benchmark accuracies, using the same avg@8 convention as Section 4.1. R1 evaluation uses vLLM 0.11.0 and the archived R1/Modal benchmark prompts; Qwen uses SGLang, its non-thinking chat template, and the Qwen benchmark prompts. These benchmarks are separate from the 1,000-row SFT loss holdout. No benchmark-accuracy trajectory was recorded during the SFT stage; Figure 8 therefore reports only SFT cross-entropy.

Table H.3: Standalone evaluation of the four final SFT checkpoints. Every entry is a percentage; Mean is the unweighted mean of the three benchmark columns. These are the original SFT evaluations, not the subsequent OPD runs' independent step-0 evaluations.

Teacher	AIME24	AIME25	AMC23	Mean
JustRL-1.5B	46.67%	35.83%	79.67%	54.06%
R1-Distill-7B	21.67%	20.42%	55.57%	32.55%
Qwen3-4B-RL-Math	31.25%	27.92%	64.01%	41.06%
Qwen3-4B (Non-thinking)	13.33%	12.92%	41.42%	22.56%

Subsequent OPD. The two continuation runs use the final update-297 SFT models as their students and retain the corresponding teachers. We optimize the student-top-16 reverse-KL surrogate in Eq (2), with student-probability weighting and a global mean over valid response tokens. Both student and teacher temperatures are 1; training uses top-$p = 1$, no top-k sampling truncation, and repetition penalty 1. Each update uses 64 prompts, one student rollout per prompt, and one optimization pass. The training data are not shuffled. Optimizer and length settings are listed in Table H.2. Benchmark decoding and aggregation follow the preceding paragraph, with an evaluation before training and every ten updates thereafter.

Both continuation runs use two H200 GPUs. The Qwen continuation uses SGLang, as does its direct-OPD comparison. The R1 continuation uses vLLM 0.11.0 in a different runtime from the four-GPU direct-OPD run; the global batch remains 64. Its validation sampling uses vLLM's Py-

Torch sampler rather than the original FlashInfer sampler, so the random draws need not coincide. Checkpoints are saved every ten updates, with at most two retained, and the comparison uses the update-200 endpoint. The Qwen log includes an abandoned initial attempt at steps 0 and 1; the completed trajectory supersedes those records. Figures and loss summaries use the last recorded value at each step, yielding one evaluation at each scheduled checkpoint and 200 training-loss observations per continuation.

Separate post-SFT starting evaluations. The continuation runs independently evaluate their SFT starting checkpoints, obtaining 36.57% for R1 and 21.63% for Qwen, compared with the original standalone SFT scores of 32.55% and 22.56%. For Qwen, all 143 rendered inputs and ground truths match the original SFT benchmark in order; the responses are newly sampled. For R1, the archived standalone benchmark asks *Let's think step by step and output the final answer within* `\boxed{}`., whereas the continuation uses *Please reason step by step, and put your final answer within* `\boxed{}`. The R1 scores therefore come from different prompt formats as well as different sampled responses and runtimes. They must not be treated as repeated evaluations under identical conditions or as the base student's pre-SFT accuracy.

Accuracy comparison. Table H.4 compares the final outcomes. With the three larger-scale teachers tested, SFT alone does no better than direct OPD. For the self-RL teacher JustRL-1.5B, SFT reaches 54.1%, slightly above direct OPD. Continuing OPD after SFT reaches 40.2% with R1-Distill-7B and 22.6% with Qwen3-4B (Non-thinking), compared with 42.7% and 22.7% for direct OPD. Relative to their own post-SFT starting evaluations, the continuations gain 3.60 and 1.01 percentage points. Neither surpasses direct OPD at the final update. These single-run comparisons do not establish statistical significance for the accuracy differences.

Table H.4: SFT on teacher responses, with and without subsequent OPD. Start is the base student's step-0 accuracy before either training stage; OPD is its accuracy after 200 direct OPD updates. SFT is the original evaluation of the final SFT checkpoint; SFT→OPD is the accuracy after 200 additional OPD updates. All accuracies are percentages averaged across the three Math benchmarks. A dash indicates an untested setting. † marks the self-RL teacher. Qwen models use non-thinking mode for generation and evaluation.

Student	Teacher	Start	OPD	SFT	SFT→OPD	Teacher
R1-Distill-1.5B	JustRL-1.5B†	39.9	51.9	54.1	—	56.9
	R1-Distill-7B	39.9	42.7	32.6	40.2	57.8
Qwen3-1.7B	Qwen3-4B-RL-Math	21.1	43.9	41.1	—	69.2
	Qwen3-4B	20.7	22.7	22.6	22.6	35.1

OPD loss dynamics after SFT. Figure 9 and Table H.5 show that neither continuation sustains loss reduction toward zero. The maximum observed reductions are 37.1% and 23.0%; the Qwen final loss is above its initial OPD loss. Thus, SFT initialization does not remove the limited loss reduction or the remaining performance gap in these two settings. This finding concerns the tested data, recipe, and budgets, rather than ruling out other SFT initializations.

Table H.5: Loss during the 200 OPD updates following SFT. The initial loss is measured at the first OPD update after SFT. Loss reduction is $(1 - \widehat{\mathcal{L}}_{\min}/\widehat{\mathcal{L}}_{\text{start}}) \times 100\%$, computed from unrounded, unsmoothed losses.

Teacher	$\widehat{\mathcal{L}}_{\text{start}}$	$\widehat{\mathcal{L}}_{\min}$	$\widehat{\mathcal{L}}_{\text{end}}$	Loss reduction
R1-Distill-7B	0.1461	0.0919	0.1232	37.1%
Qwen3-4B (Non-thinking)	0.0805	0.0620	0.0861	23.0%

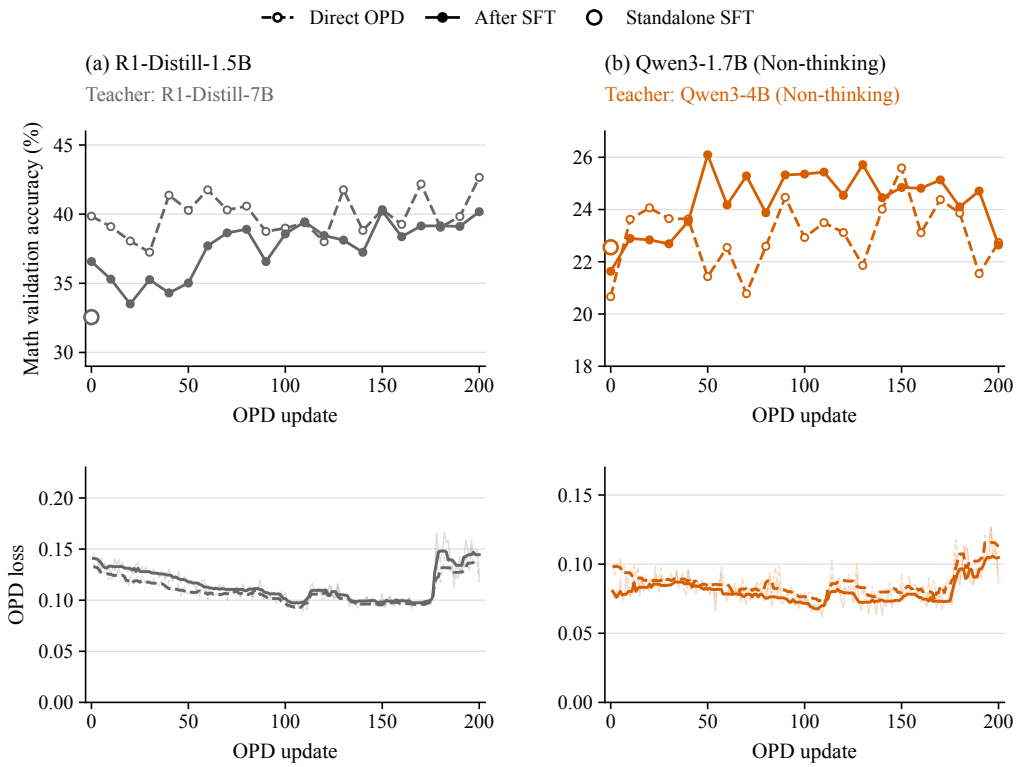


Figure 9: Direct OPD and OPD after SFT with the two tested teachers. Top: benchmark accuracy from each run's own recorded evaluations. The direct R1 curve's step-0 point is the shared base-model measurement described in Appendix E. Larger hollow circles separately show the original standalone SFT evaluations; they are not connected to either trajectory. The R1 standalone SFT evaluation uses a different prompt suffix. Bottom: raw per-update OPD loss in faint lines and centered nine-update rolling medians in stronger lines. Solid lines are continuations after SFT; dashed lines are direct OPD from the base student. Each panel uses a linear scale containing its observations. The two training stages have separate update counts, and the continuation does not include the SFT cost on its horizontal axis.

H.2 Vocabulary support and training rollouts

We compare the main top-16, one-rollout configuration with full-logit OPD at one rollout per prompt and top-16 OPD at eight rollouts per prompt. These comparisons cover Qwen3-1.7B with Qwen3-4B-RL-Math and Qwen3-4B, and R1-Distill-1.5B with JustRL-1.5B and R1-Distill-7B. Table H.6 reports all twelve conditions: four main-run baselines and eight additional completed runs. The two Qwen full-logit runs and the Qwen3-4B eight-rollout run are the same runs already reported in Table E.4, not additional replications. JustRL-1.5B is a self-RL teacher for R1-Distill-1.5B; Qwen3-4B-RL-Math is an RL-trained larger model relative to Qwen3-1.7B.

Data and optimization. All additional runs use the same DAPO-Math-17K training file as their corresponding baselines, containing 17,917 prompts before prompt-length filtering. We apply the student's chat template, filter prompts longer than 1,024 tokens, and keep the dataset order fixed. Qwen3 uses non-thinking mode. Training response limits are 7,168 tokens for Qwen3-1.7B and 21,000 for R1-Distill-1.5B. Each run has 200 optimizer updates, with 64 prompts per update, student sampling temperature 1, top-$p = 1$, no sampling top-k cutoff, and teacher temperature 1. The optimizer is AdamW with constant learning rate 10^{-6}, no warmup, $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.01, and gradient-norm clipping at 1. We use one optimization epoch per prompt batch and a global mean over valid response tokens. The teacher remains fixed; no task reward or format reward

is added to the OPD update. The vocabulary top-16 restriction is separate from the unrestricted training sampling distribution.

Increasing the rollout count from one to eight changes the number of sampled responses per update from 64 to 512. The optimization minibatch is expanded accordingly, so both settings still perform one optimizer update per batch. Thus, the total numbers of sampled training responses are 12,800 and 102,400, respectively, while both process 12,800 prompt instances. This is a comparison at fixed prompt and update budgets, not at fixed generation cost. We use dynamic microbatching with a 32,768-token per-GPU budget, FSDP, activation checkpointing, and bfloat16 rollout inference. The additional runs use two H200 GPUs, except the R1-Distill-7B top-16, eight-rollout run, which uses four. The Qwen Qwen3-4B-RL-Math top-16 baseline and both R1 top-16 baselines use four GPUs. All other Qwen conditions use two. Qwen runs use SGLang 0.5.3rc0 and Transformers 4.56.1; the additional R1 runs use vLLM 0.11.0 and Transformers 4.57.6. The additional runs use PyTorch 2.8.0 and Ray 2.49.2. No separate experiment-level seed is recorded in the saved data configuration. Each condition is a single stochastic training run; the comparisons are not paired-seed or repeated-seed estimates.

Full-logit implementation. Top-16 OPD selects the student's sixteen highest-probability vocabulary entries at each visited prefix. Student probabilities are renormalized over these entries to form detached weights, while the log-probability ratio uses each model's full-vocabulary normalization. Full-logit OPD instead includes all 151,936 student output entries. It retains the same detached student-weighted update and loss normalization, computing the output projection in chunks of 128 prefix states to limit memory use. No vocabulary truncation is applied in these full-logit runs. For R1-Distill-7B, whose output head has 152,064 entries, teacher logits are normalized over all 152,064 entries before selecting the 151,936 entries aligned with the student; the selected teacher probabilities are not renormalized. Token-ID compatibility is checked before training. Because the top-16 and full-logit objectives differ, we compare their loss reductions relative to their own update-1 losses, rather than treating their absolute losses as the same quantity.

Evaluation and run accounting. All additional runs are evaluated before training and every ten updates on AIME 2024 (30 problems), AIME 2025 (30), and AMC 2023 (83). Each evaluation contains 1,144 responses: eight per problem, sampled at temperature 0.7, top-$p = 0.95$, and a 31,744-token response limit within a 32,768-token context. We average correctness over responses within each benchmark and then average the three benchmark scores equally. The same answer-extraction and mathematical-correctness scorer is used throughout; format reward is disabled. We retain each student group's original prompt formatting: Qwen validation requests step-by-step reasoning and a boxed final answer under its non-thinking chat template, while R1 validation uses the corresponding R1 prompt and reasoning prefix. Dataset and rendered-prompt checks verify consistency within each student group. The two original R1 baselines use the shared initial-student evaluation described in Appendix E; every additional run uses its own step-0 evaluation. Variations in these initial estimates reflect sampling from the same initial checkpoint.

For each additional run, the loss record contains all 200 updates and the validation record contains all 21 evaluation checkpoints. Native optimizer and data-loader counters were checked against the saved checkpoints. We use the completed run's canonical local records when an earlier attempt failed: in particular, the JustRL eight-rollout run was restarted from the initial checkpoint after an unsaved one-update attempt, and the discarded attempt is not concatenated with the completed trajectory. These eight additional runs are excluded from the seven-run averages in the main text.

Observed behavior. Figures 10 and 11 show validation accuracy and loss over the full training window. For Qwen3-1.7B, final accuracy with Qwen3-4B-RL-Math is 43.94%, 46.36%, and 45.21% for top-16 with one rollout, full logits with one rollout, and top-16 with eight rollouts, respectively. The corresponding final loss reductions are 55.37%, 54.85%, and 51.29%. With Qwen3-4B, accuracy remains much lower (22.74%, 22.88%, and 24.87%), and final loss reductions remain small (6.24%, 11.60%, and 9.89%). Thus, removing the vocabulary restriction or increasing the rollout count preserves the broad teacher-dependent behavior in these Qwen runs.

For R1-Distill-1.5B, JustRL-1.5B also produces higher final accuracy than R1-Distill-7B under all three configurations. With JustRL, final accuracies are 51.88%, 50.46%, and 51.93%, and maximum loss reductions are all approximately 96%. However, the eight-rollout run's loss rises during its final

Table H.6: Support and rollout comparisons after 200 updates. Accuracy is macro-averaged $\mathrm{avg@8}$. Maximum and final loss reduction use $(1 - \widehat{\mathcal{L}}_{\min}/\widehat{\mathcal{L}}_1) \times 100\%$ and $(1 - \widehat{\mathcal{L}}_{200}/\widehat{\mathcal{L}}_1) \times 100\%$, respectively. Each uses its own run's objective. The four top-16, $n = 1$ rows are the main-run baselines; the other eight rows are additional runs.

Teacher	Support	n	Accuracy		Loss reduction	
			Initial	Final	Maximum	Final
Student: Qwen3-1.7B (Non-thinking)						
Qwen3-4B-RL-Math	Top-16	1	21.10%	43.94%	65.08%	55.37%
	Full	1	20.55%	46.36%	62.68%	54.85%
	Top-16	8	22.95%	45.21%	60.62%	51.29%
Qwen3-4B	Top-16	1	20.66%	22.74%	30.36%	6.24%
	Full	1	21.66%	22.88%	37.52%	11.60%
	Top-16	8	21.85%	24.87%	33.02%	9.89%
Student: R1-Distill-1.5B						
JustRL-1.5B	Top-16	1	39.85%	51.88%	96.29%	92.95%
	Full	1	39.85%	50.46%	96.22%	91.80%
	Top-16	8	39.85%	51.93%	96.18%	31.09%
R1-Distill-7B	Top-16	1	39.85%	42.66%	34.35%	13.94%
	Full	1	39.85%	40.32%	30.77%	-7.05%
	Top-16	8	40.39%	38.87%	35.03%	18.29%

updates, leaving a final loss reduction of 31.09%, compared with 92.95% and 91.80% in the one-rollout top-16 and full-logit runs. The R1-Distill-7B full-logit run ends with a loss 7.05% above its update-1 value; its maximum loss reduction is 30.77%. We therefore report both maximum and final loss reduction. These runs preserve the broad teacher ordering, but their detailed loss trajectories are not interchangeable, and a larger rollout count does not uniformly improve optimization. In the JustRL eight-rollout run, the pre-clip gradient norm and mean response length also rise during the last updates; the epoch counter remains zero throughout this interval. These logs establish the late increase but do not identify its cause.

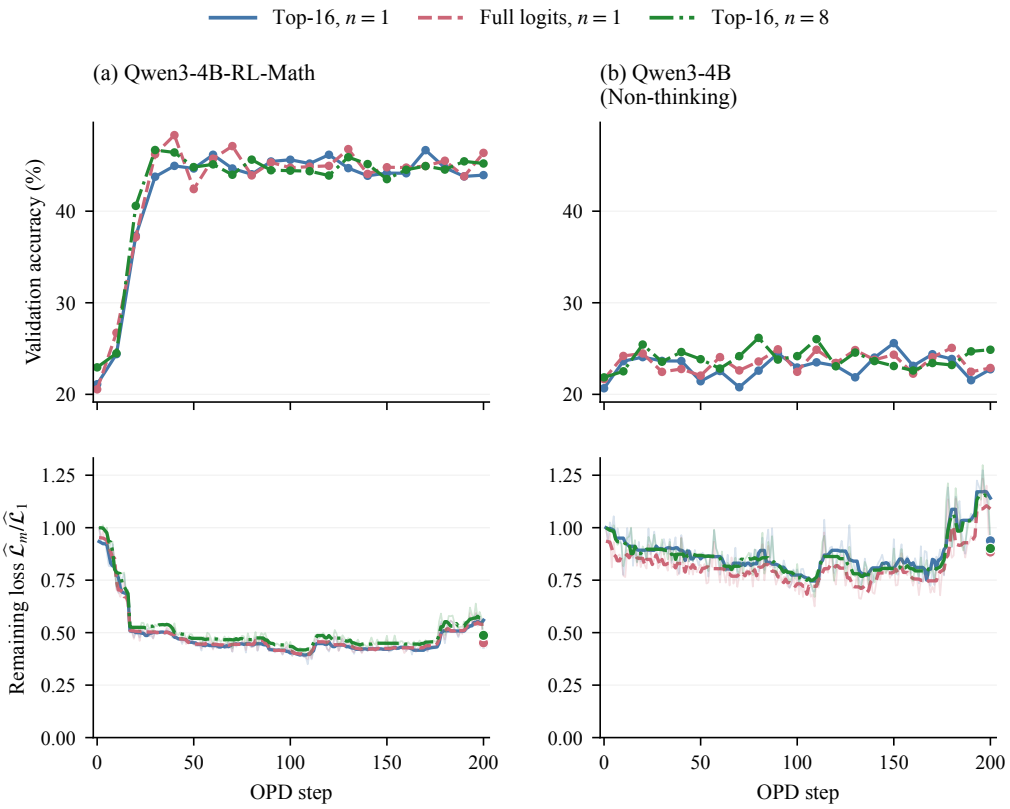


Figure 10: Vocabulary-support and rollout comparisons for Qwen3-1.7B (Non-thinking). Each column fixes the OPD teacher. Top: validation accuracy. Bottom: loss divided by that run's update-1 loss. Faint lines show all raw losses; thicker lines show centered nine-step rolling medians. Filled loss markers show the raw update-200 values. Here n is the number of training rollouts per prompt; all validation scores use avg@8.

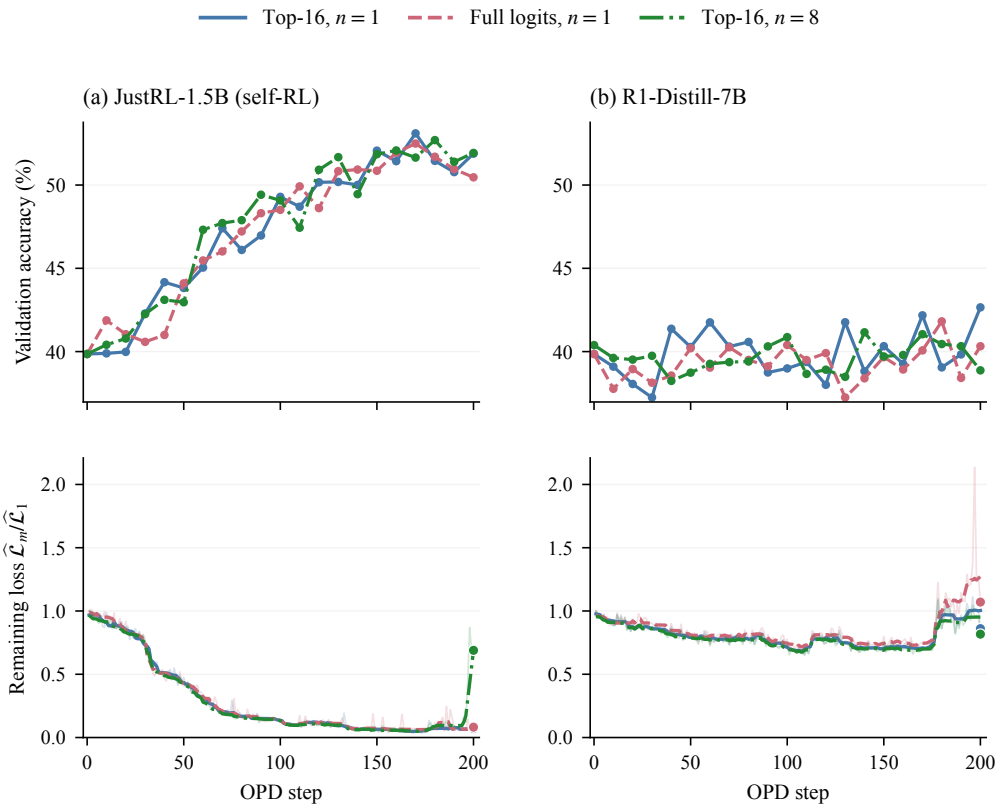


Figure 11: Vocabulary-support and rollout comparisons for R1-Distill-1.5B, with the same plotting conventions as Figure 10. JustRL-1.5B is a self-RL teacher. The late loss increase in its eight-rollout run is retained, as is the loss increase in the full-logit R1-Distill-7B run.