Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe

Anonymous authors
Paper under double-blind review

Abstract

On-policy distillation (OPD) has become a core technique in the post-training of large language models, yet its training dynamics remain poorly understood. This paper provides a systematic investigation of OPD dynamics and mechanisms. We first identify that two conditions govern whether OPD succeeds or fails: (i) the student and teacher should share compatible thinking patterns; and (ii) even with consistent thinking patterns and higher scores, the teacher must offer genuinely new capabilities beyond what the student has seen during training. We validate these findings through weak-to-strong reverse distillation, showing that same-family 1.5B and 7B teachers are distributionally indistinguishable from the student's perspective. Probing into the token-level mechanism, we show that successful OPD is characterized by progressive alignment on high-probability tokens at student-visited states, a small shared token set that concentrates most of the probability mass (97%–99%). We further propose two practical strategies to recover failing OPD: off-policy cold start and teacher-aligned prompt selection. Finally, we show that OPD's apparent free lunch of dense token-level reward comes at a cost, raising the question of whether OPD can scale to long-horizon distillation.

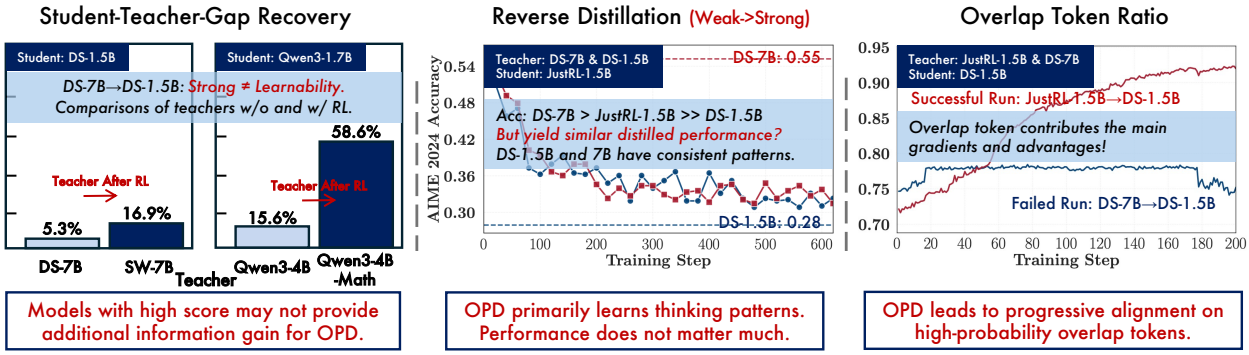


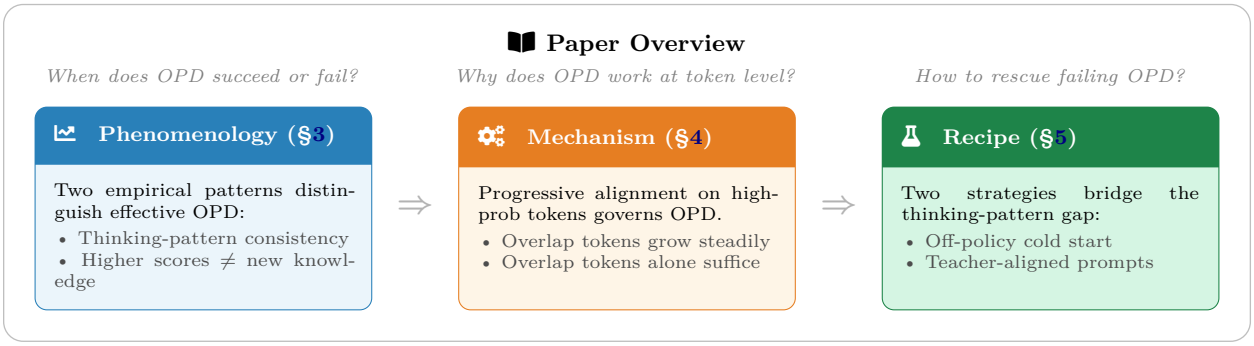
Figure 1: Overview of our paper. JustRL-1.5B is obtained by applying RL to DeepSeek-Distill-1.5B (DS-1.5B), and Skywork-OR1-Math-7B (SW-7B) by applying RL to DeepSeek-Distill-7B (DS-7B).

1 Introduction

On-policy distillation (OPD) has rapidly emerged as a core technique for large language model (LLM) post-training. Recent industry efforts, including Qwen3 (Yang et al., 2025), MiMo (Xiao et al., 2026) and GLM-5 (Zeng et al., 2026), all adopt OPD in their post-training pipelines and report substantial gains, establishing it as a competitive complement to conventional supervised fine-tuning (SFT) and outcome-reward reinforcement learning (RL). Thinking Machines Lab (Lu & Lab, 2025) replicates the Qwen3 OPD recipe at a fraction of the RL compute cost, independently confirming that on-policy, dense supervision is a practically efficient alternative.

Unlike off-policy distillation, which trains the student on fixed teacher-generated sequences and suffers from exposure bias (Bengio et al., 2015), OPD has the student generate its own rollouts and leverages the teacher's per-token log-probabilities as a dense reward signal to refine behavior on states the student actually visits. Recently, this has been extended to self-distillation settings where a single model serves as its own teacher given privileged information, demonstrating that the framework can drive continual self-improvement (Hübotter et al., 2026; Zhao et al., 2026b; Shenfeld et al., 2026).

However, despite these successes, OPD remains poorly understood and fragile in practice. We observe a striking failure mode: a stronger teacher can completely fail to improve a student, even when a weaker teacher succeeds from lower initial alignment. Yet few studies have investigated why the teacher's token-level signal steers the student distribution in the desired direction, or the conditions under which it fails.



We present a systematic study of OPD training dynamics, progressing from empirical conditions through token-level mechanism to practical recipe.

Phenomenology (§3). We investigate the empirical patterns that distinguish effective from ineffective OPD, and identify two governing factors. *(i) Thinking-pattern consistency*: the student and teacher should share consistent thinking patterns (e.g. higher overlap ratio in their top-k token distributions). Even when the teacher achieves higher benchmark scores, mismatched thinking patterns produce low initial overlap that training cannot fully recover. *(ii) Higher scores $\neq$ new knowledge*: even with consistent thinking patterns and higher benchmark scores, the teacher should offer knowledge that the student has not already acquired. When both models are trained on the same data and recipe, they converge to similar distributions at their respective scales, leaving the teacher with little transferable signal. Only when the teacher carries knowledge beyond what the student has already seen can OPD yield substantial gains. We validate both conditions through reverse distillation experiments, which further reveal that OPD fundamentally learns thinking patterns rather than merely benefiting from pattern consistency, and that training dynamics can be entirely decoupled from benchmark scores.

Mechanism (§4). We then investigate the token-level mechanism based on these conditions. Across all settings studied, effective OPD exhibits a consistent signature where the student and teacher distributions become progressively more similar on student-visited states. The high-probability tokens increasingly coincide (overlap ratio rising from 72% to 91%), the entropy gap between the two distributions narrows, and the shared top-k tokens concentrate 97–99% of the combined probability mass. By contrast, failing runs show stagnant overlap and persistent entropy mismatch from the outset. We further show that restricting supervision to overlap tokens alone matches full top-k performance, confirming that the overlap set is the principal locus of OPD's gradient signal.

Recipe (§5). Guided by the phenomenological analysis, we propose two complementary strategies that recover OPD in otherwise failing configurations: *(i) off-policy cold start*, a warmup SFT phase on teacher-generated rollouts before OPD that bridges the thinking-pattern gap by raising initial overlap ratio; *(ii) teacher-aligned prompt selection*, which uses prompts drawn from the teacher's post-training data to sharpen alignment on high-probability tokens, though at the cost of substantially lower student entropy that necessitates mixing with out-of-distribution prompts. In both cases, the recovered runs exhibit the same dynamic signature as naturally successful ones as shown in §4: steadily rising overlap ratio, improving token-level advantage, and a narrowing entropy gap.

Finally, we examine the cost of OPD's dense supervision (Appendix E). We show that reward quality degrades systematically with trajectory depth and that instability originates at later tokens before propagating backward through the trajectory. Surprisingly, even failing teachers provide reward that is globally correlated with rollout correctness, suggesting that the failure is not one of signal quality but of local optimization geometry. A larger teacher may induce a reward landscape that is locally flat around the student's policy, making token-level gradients ineffective despite an informative global signal. These findings reveal a fundamental tension between supervision density and supervision reliability, and point to the limitations of current OPD for long-horizon reasoning and agentic settings.

2 Preliminaries

2.1 Notation

Let $x = (x_1, \ldots, x_n)$ denote an input prompt and $y = (y_1, \ldots, y_m)$ a response. We write $y_{<t} \triangleq (y_1, \ldots, y_{t-1})$ for the prefix up to step t. We consider two LLMs: a student π_θ and a teacher π_T, each defining a next-token distribution $\pi(\cdot \mid x, y_{<t})$ over a vocabulary $\mathcal{V}$. We write $y \sim \pi_\theta(\cdot \mid x)$ for a response sampled autoregressively from the student. $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$ denotes a fixed dataset of prompt–response pairs with teacher-generated outputs, and $\mathcal{D}_x \triangleq \{x^{(i)}\}_{i=1}^N$ the corresponding prompt set. Knowledge distillation (KD) transfers knowledge from π_T to π_θ by minimizing the divergence between the two distributions. A standard choice is the Kullback-Leibler (KL) divergence, defined for two distributions P and Q over $\mathcal{V}$ as $D_{\mathrm{KL}}(P \| Q) = \sum_{v \in \mathcal{V}} P(v) \log \frac{P(v)}{Q(v)}$.

2.2 On-Policy Distillation

On-Policy Distillation (OPD) computes supervision on trajectories sampled from the current student π_θ. Given a prompt $x \sim \mathcal{D}_x$, the student samples a response $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T) \sim \pi_\theta(\cdot \mid x)$. At each student-generated prefix $\hat{y}_{<t}$, we write $p_t(v) \triangleq \pi_\theta(v \mid x, \hat{y}_{<t})$ and $q_t(v) \triangleq \pi_T(v \mid x, \hat{y}_{<t})$ for the student and teacher next-token distributions. The canonical OPD objective aligns the two policies on these student-visited states:

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x,\, \hat{y} \sim \pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} D_{\mathrm{KL}}(p_t \| q_t) \right]. \tag{1}$$

In practice, implementations trade off supervision density and computation through full-vocabulary, sampled-token, or top-k variants. In this paper, unless otherwise specified, we instantiate OPD with student top-k supervision, where the loss is evaluated on $S_t = \mathrm{TopK}(p_t, k)$. We give the formal variants in Appendix A.1.

2.3 Dynamic Metrics

We define the student's and teacher's top-k sets at step t as $S_t^{(p)} = \mathrm{TopK}(p_t, k)$ and $S_t^{(q)} = \mathrm{TopK}(q_t, k)$, respectively. The following metrics are monitored throughout OPD training in later experiments.

Overlap Ratio. This is the average fraction of student top-k tokens that also appear in teacher top-k set:

$$\mathcal{M}_{\mathrm{overlap}} \triangleq \mathbb{E}_t \left[\frac{|S_t^{(p)} \cap S_t^{(q)}|}{k} \right]. \tag{2}$$

Higher values indicate stronger support alignment between student and teacher.

Overlap-Token Advantage. To measure distributional agreement within the overlap tokens, we define $A_t(v) \triangleq \bar{p}_t(v)(\log \bar{q}_t(v) - \log \bar{p}_t(v))$ where $\bar{p}_t, \bar{q}_t$ are the renormalized student and teacher distributions over $S_t^{(p)} \cap S_t^{(q)}$. The metric averages this quantity:

$$\mathcal{M}_{\mathrm{adv}} \triangleq \mathbb{E}_t \left[\frac{1}{|S_t^{(p)} \cap S_t^{(q)}|} \sum_{v \in S_t^{(p)} \cap S_t^{(q)}} A_t(v) \right]. \tag{3}$$

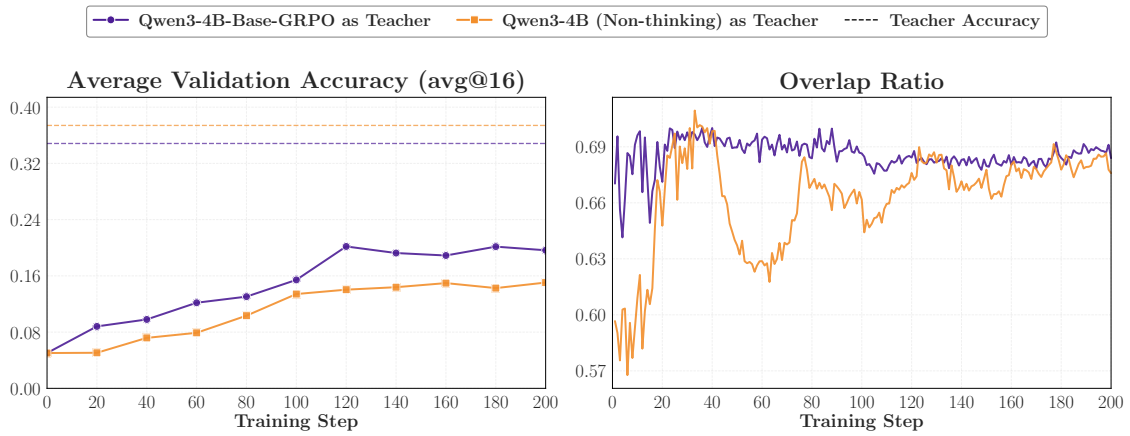


Figure 2: OPD from two teachers with different thinking patterns into the same student (Qwen3-1.7B-Base). The GRPO teacher achieves stronger performance (left) and higher initial overlap ratio (right), suggesting that thinking pattern compatibility governs OPD effectiveness.

Values closer to zero indicate better probability alignment within the shared support, while more negative values indicate student overconfidence relative to the teacher.

Entropy and Entropy Gap. We track the entropy of both policies on student rollouts and define:

$$\Delta H_t = |H(q_t) - H(p_t)| \,. \tag{4}$$

ΔH_t measures confidence mismatch on student-visited states; convergence toward zero indicates a closer match to the teacher's uncertainty profile.

3 Phenomenology of On-Policy Distillation

Before investigating the token-level mechanism of OPD, we first ask a broader question: what conditions govern the effectiveness of OPD? A natural assumption is that a stronger teacher should always yield better distillation outcomes, yet we observe configurations where this fails. We compare OPD runs under controlled settings and identify two conditions that jointly govern the outcome.

Takeaways

- **Thinking-pattern consistency.** The student and teacher should share compatible thinking patterns. Even when the teacher achieves higher benchmark scores, a large mismatch weakens the token-level distillation signal (Section 3.1).
- **Higher scores $\neq$ new knowledge.** The teacher should provide knowledge beyond what the student has seen during training. Even with consistent thinking patterns and higher scores, a teacher may offer no genuinely new knowledge, leaving OPD without a driving signal (Section 3.2).

3.1 Thinking-Pattern Consistency

We first study whether OPD requires compatible thinking patterns between the student and the teacher. A stronger teacher does not guarantee better distillation: a large mismatch in reasoning pattern can weaken the distillation signal regardless of the teacher's benchmark advantage.

Setup. We use Qwen3-1.7B-Base (Yang et al., 2025) as student and compare two teachers: Qwen3-4B (Non-thinking) (Yang et al., 2025) and Qwen3-4B-Base-GRPO, where the latter is obtained by applying zero-RL to Qwen3-4B-Base (Yang et al., 2025) using GRPO (Shao et al., 2024) (Training details are provided in Appendix B.1). Since the student is also a base model, we expect its thinking pattern to be closer to that

of the GRPO-trained teacher. We conduct two OPD experiments using the DAPO-Math-17K dataset (Yu et al., 2025), differing only in the choice of teacher model. Unless otherwise specified, all experiments use the default hyperparameters described in Appendix B.2 and are evaluated on AIME 2024 (Li et al., 2024), AIME 2025 (Balunović et al., 2025) and AMC 2023 (Li et al., 2024). Following standard practice, we sample 16 solutions per problem with temperature 0.7 and top-p 0.95, using a maximum validation response length of 31,744 tokens. We report average accuracy over 16 samples (avg@16) as the primary evaluation metric.

Results. As shown in Figure 2, distillation from Qwen3-4B-Base-GRPO consistently outperforms distillation from Qwen3-4B (Non-thinking), although the two teachers have broadly comparable performance as shown in Figure 3. Despite underperforming on benchmarks, the GRPO teacher exhibits a higher initial overlap ratio, suggesting that its thinking pattern aligns more closely with the student. Although the two overlap curves converge later in training, the performance gap persists, suggesting that early-stage thinking-pattern mismatch causes a loss of distillation benefit that cannot be recovered later. We report the validation accuracy for each benchmark individually in Appendix B.3, where the same overall trend holds across all datasets.

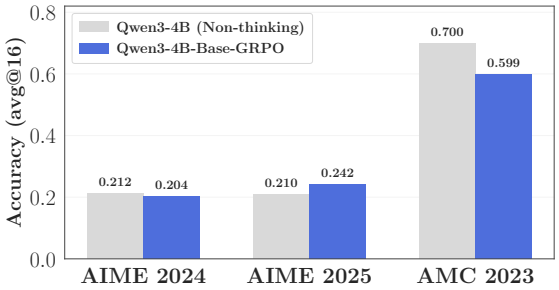


Figure 3: Validation performance of the two teachers (Qwen3-4B Non-thinking vs. Qwen3-4B-Base-GRPO).

3.2 New Knowledge, Not Just Scale

Thinking-pattern consistency alone does not explain all of our observations. Even when the teacher scores higher and shares a consistent thinking pattern with the student, OPD can still fail.

Setup. We construct two controlled comparisons across model families. In the DeepSeek family, we use DeepSeek-R1-Distill-Qwen-1.5B (R1-Distill-1.5B) (Guo et al., 2025) as the student and compare two teachers: DeepSeek-R1-Distill-Qwen-7B (R1-Distill-7B) (Guo et al., 2025) and Skywork-OR1-Math-7B (He et al., 2025b), where the latter is obtained by applying RL post-training to R1-Distill-7B. In the Qwen family, we use Qwen3-1.7B (Non-thinking) (Yang et al., 2025) as the student and compare two teachers: Qwen3-4B (Non-thinking) and Qwen3-4B-Non-Thinking-RL-Math (Yang et al., 2026b), where the latter is obtained by applying RL to Qwen3-4B (Non-thinking) on a 57K subset of DeepMath (He et al., 2025c). In both settings, the key contrast lies between a teacher from the same training pipeline and one that has acquired additional capabilities through further RL. All runs use the same dataset and training recipe as before.

Results. As shown in Figure 4, both families exhibit a consistent pattern. **Same-pipeline teachers** yield limited improvement, while the **post-trained teachers** produce substantially stronger gains. Importantly, the post-trained teachers not only achieve higher absolute performance but also recover a much larger fraction of the teacher-student gap, measured by the *gap recovery rate* $(\mathrm{Acc}_{\text{after OPD}} - \mathrm{Acc}_{\text{before OPD}})/(\mathrm{Acc}_{\text{teacher}} - \mathrm{Acc}_{\text{before OPD}})$. This indicates that the additional capabilities acquired by these teachers are more transferable through OPD. Since the post-trained teachers are derived from the same base model, their thinking patterns remain broadly aligned, which is also observed by the overlap ratio dynamic. The improvement therefore stems from new capabilities of the teacher acquired through RL.

3.3 Validation via Reverse Distillation

We design a reverse-distillation experiment as the comparison that simultaneously validates both conditions and reveals deeper insights into the nature of OPD.

Setup. JustRL-DeepSeek-1.5B (JustRL-1.5B) (He et al., 2025a) is obtained by RL from R1-Distill-1.5B. We now reverse this direction, using JustRL-1.5B as the student and distilling from R1-Distill-1.5B (its own pre-RL checkpoint). We also use R1-Distill-7B as a teacher for the comparison. Note that R1-Distill-7B achieves slightly higher benchmark scores than JustRL-1.5B, while R1-Distill-1.5B is substantially weaker.

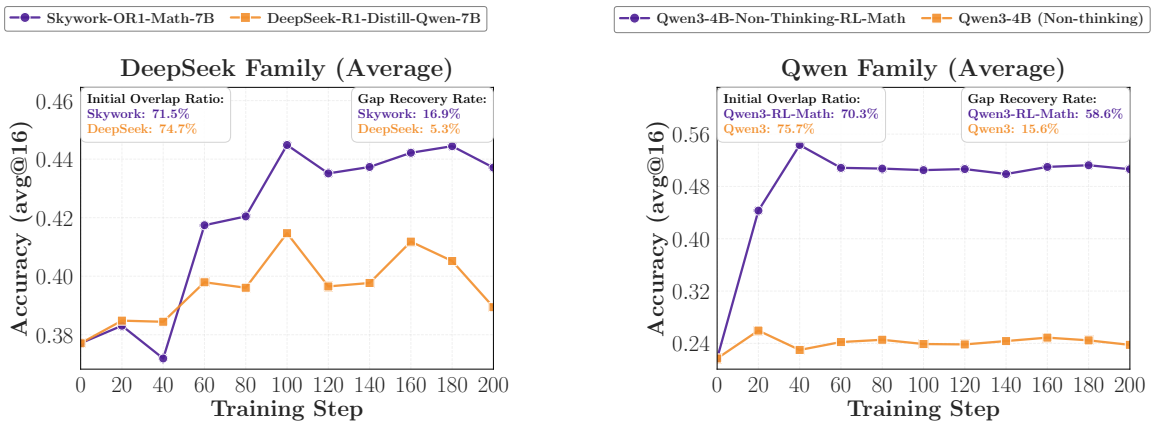


Figure 4: Comparison of OPD performance with and without additional teacher RL post-training across two model families. **Left:** DeepSeek family. **Right:** Qwen family. Post-trained teachers yield substantially stronger gains and higher teacher-student *gap recovery rate*.

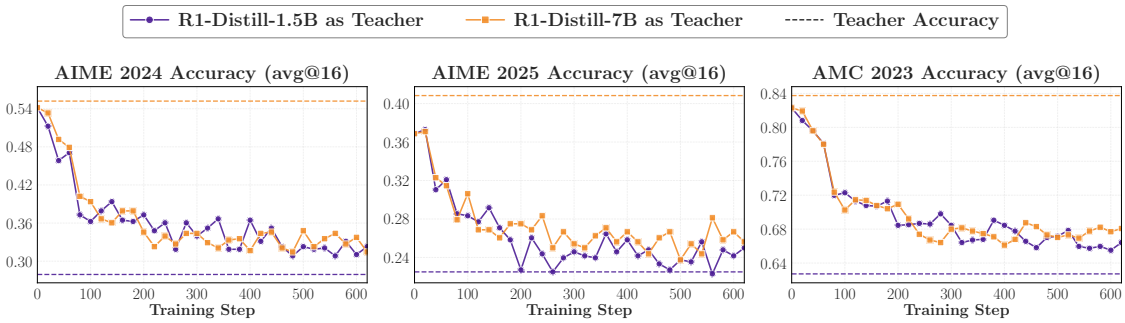


Figure 5: Reverse distillation with JustRL-1.5B as the student and two same-family teachers (R1-Distill-1.5B and R1-Distill-7B). Both runs cause the student to regress to approximately the same level despite R1-Distill-7B scoring higher than JustRL-1.5B, indicating that OPD training dynamics are governed by thinking pattern rather than benchmark performance.

Results. Figure 5 reveals two striking phenomena. First, distilling JustRL-1.5B toward R1-Distill-1.5B, its own pre-RL checkpoint, causes the student to regress almost exactly to its pre-RL performance, removing all gains acquired through RL. Second, when we replace the teacher with R1-Distill-7B, a substantially larger and even slightly stronger model from the same family, the training trajectory is nearly indistinguishable: despite outscoring JustRL-1.5B on benchmarks, R1-Distill-7B drives the student to the same regressed level as the weaker 1.5B teacher. Since OPD minimizes reverse KL divergence over student-generated trajectories, this convergence implies that the two teachers induce nearly identical local target distributions on student-visited states, despite their difference in scale. These results yield several conclusions:

- **Thinking pattern matters, and OPD fundamentally learns thinking patterns.** Distilling from R1-Distill-1.5B into JustRL-1.5B causes JustRL-1.5B to regress to its pre-RL performance. This suggests OPD acquires the teacher's thinking pattern and overwrites the student's own. This explains why consistency in thinking patterns is important: if the gap is too large, the student may fail to learn effectively.
- **Benchmark performance does not predict OPD outcome.** R1-Distill-7B scores higher than JustRL-1.5B, yet the distillation produces no improvement and instead causes regression. This shows that OPD's training dynamics can be completely independent of the teacher's benchmark performance, and may even move in the opposite direction.
- **Higher scores do not imply new knowledge for OPD.** R1-Distill-7B and R1-Distill-1.5B are within the same model family and differ only in scale. The indistinguishable effects of the two models on the

student already confirm that: (i) a higher score (R1-Distill-7B) may merely reflect a different degree of fit to the same data, rather than genuinely novel capabilities. For OPD to produce gains, the teacher should possess knowledge beyond what the student has already seen during training; and (ii) despite the difference in scale, R1-Distill-7B and 1.5B exhibit the same thinking patterns.

The reverse distillation experiments and the forward comparisons in Sections 3.1 and 3.2 consolidate the two conditions. Thinking-pattern consistency is associated with higher initial overlap and stronger OPD outcomes, while new knowledge (such as from further post-training) enables larger transferable gains even when overlap is already high.

4 Mechanism of On-Policy Distillation

Section 3 identified two conditions, thinking-pattern consistency and new knowledge beyond the same model family, that govern OPD effectiveness. We now investigate the token-level mechanism through which these conditions manifest during training. By comparing successful and failing OPD runs, we show that effective distillation is driven by progressive alignment on high-probability tokens.

Takeaways

- **Progressive alignment.** The overlap between the student's and teacher's high-probability top-k tokens increases steadily throughout training at student-visited states; failing runs show stagnant overlap from the outset.
- **Overlap sufficiency.** Nearly all of the optimization's effect concentrates on the shared top-k tokens; optimizing only these overlap tokens suffices to match standard OPD, while non-overlap tokens contribute little.

4.1 Progressive Alignment of High-Probability Tokens

We compare the dynamics of a single student distilled from two different teachers under the same settings, one yielding clear improvement and the other yielding none. We find that successful OPD is essentially driven by learning the high-probability tokens shared between the student and teacher.

Setup. We choose R1-Distill-1.5B as student and compare two teachers: JustRL-1.5B and R1-Distill-7B. The two teachers exhibit comparable math performance, with the latter being slightly stronger. We still use DAPO-Math-17K dataset and training settings as before, and monitor three dynamic metrics during OPD.

Results. Figure 6 shows sharply different outcomes. Distillation from JustRL-1.5B yields consistent gains, with the final student recovering more than 80% of the performance gap to the teacher, whereas distillation from R1-Distill-7B fails to yield any improvement despite the teacher being stronger overall. The training dynamics (Figure 6, bottom) reveal the underlying divergence. **In the successful run, the overlap ratio rises steadily, the overlap-token advantage improves toward zero, and the entropy gap narrows, indicating that the student progressively locates the teacher's high-probability region, calibrates its mass within that region, and matches the teacher's local confidence. In the failing run, all three metrics stagnate.**

Two observations deserve emphasis. First, the overlap tokens carry 97%-99% of the total probability mass for both models throughout training (see Appendix C.1), so the rising overlap reflects alignment on the probabilistically dominant tokens, not merely a set-level coincidence. Second, the improvement in overlap-token advantage suggests that OPD's primary optimization signal lies in reweighting probability within the overlap region rather than in tokens outside it.

We also report auxiliary optimization metrics (policy loss, gradient norm, and extreme-advantage token probability differences) in Appendix C.2, which show consistent secondary patterns: the successful run exhibits decreasing loss and sustained gradient magnitude, while the failing run shows weak gradients and persistent probability discrepancies. We further verify that these findings generalize across different model pairs in Appendix C.3, using R1-Distill-7B as the student with two different teachers under the same settings.

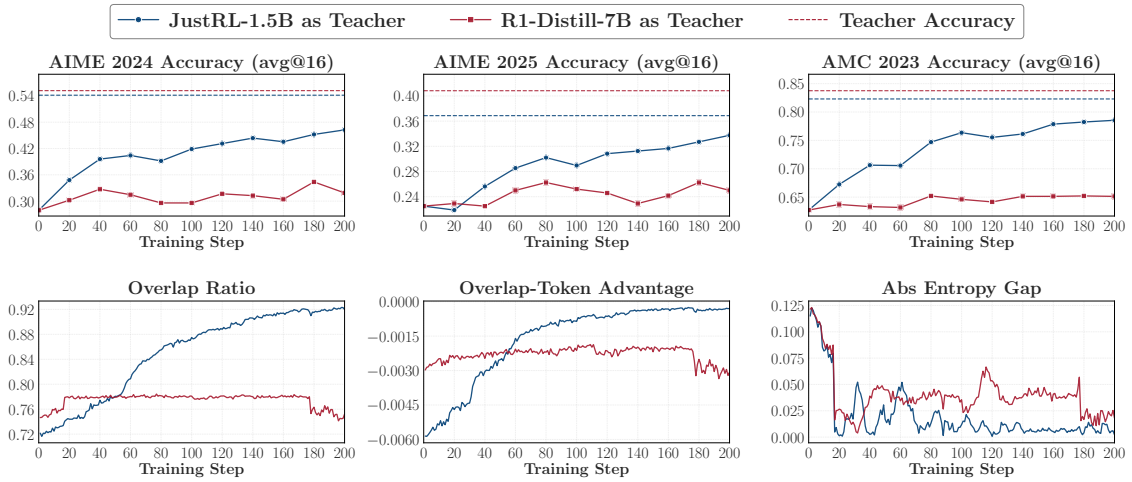


Figure 6: Successful vs. failing OPD with the same student (R1-Distill-1.5B) and two teachers. **Top:** avg@16 accuracy on three benchmarks. **Bottom:** three dynamics over training. Successful distillation (JustRL-1.5B) shows rising overlap and narrowing entropy gap, and these trends are absent in the failing run (R1-Distill-7B).

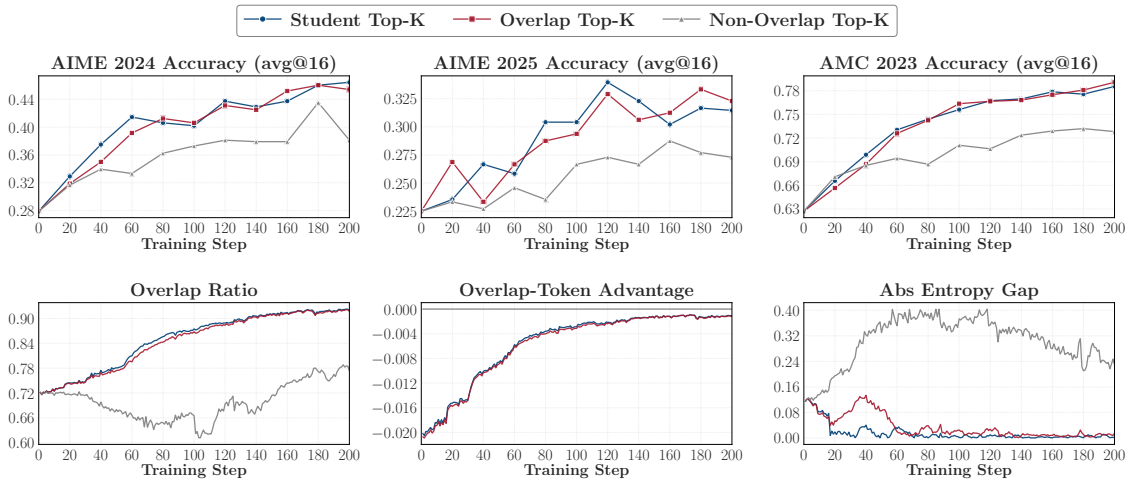


Figure 7: Ablation on the optimization support in Top-k OPD. Overlap Top-k matches Student Top-k, while Non-Overlap Top-k is substantially weaker.

4.2 Optimizing Shared Tokens Alone Suffices

The above analysis shows that high-probability token alignment correlates with OPD success. We further investigate whether this correlation is causal: whether the overlap region is not only where alignment emerges, but also the region that drives optimization. We design a targeted ablation that decomposes the top-k support into its overlap and non-overlap parts, training on each in isolation.

Setup. Using the successful OPD setting from Section 4.1 (JustRL-1.5B $\rightarrow$ R1-Distill-1.5B), we compare three variants that differ only in which tokens the distillation loss covers: (i) **Student Top-k**, which optimizes on the full student top-k support $S_t^{(p)}$; (ii) **Overlap Top-k**, which restricts optimization to the intersection of the student and teacher top-k sets $S_t^{(p)} \cap S_t^{(q)}$; and (iii) **Non-Overlap Top-k**, which restricts optimization to their symmetric difference $S_t^{(p)} \triangle S_t^{(q)}$. We set default k to 16.

Results. As shown in Figure 7, optimizing only the overlap region is sufficient to recover nearly the full benefit of standard Student Top-k OPD, while Non-Overlap Top-k remains consistently weaker. This suggests that the main gains of OPD come from gradients on the shared high-probability region, rather than non-overlap

tokens. This also explains why Student Top-k and Overlap Top-k behave so similarly. The extra tokens in the student-only support carry very little probability mass. Consistently, the overlap-token advantage curves of Student Top-k and Overlap Top-k are nearly indistinguishable, whereas Non-Overlap Top-k has much smaller magnitude, indicating a much weaker effective gradient on the overlap tokens.

Overlap optimization is self-reinforcing. Both Student Top-k and Overlap Top-k raise the overlap ratio steadily from about 72% to above 91%, while Non-Overlap Top-k first decreases and then only partially recovers (Figure 7, bottom-left). This reveals a self-reinforcing dynamic: once a token enters the shared high-probability region and is favored by the teacher, reverse-KL updates concentrate more mass on it, gradually pushing competing non-overlap tokens out of the student's top-k set. The overlap region grows because of the optimization, forming a virtuous cycle that sustains alignment throughout training.

Overall, these results support a unified mechanism for OPD: its primary effect is to progressively refine the student's distribution over teacher-supported high-probability tokens at student-visited states. This alignment is both the signature of successful OPD and its operative locus, where optimizing only the overlap tokens suffices, and non-overlap tokens contribute little. When the conditions identified in Section 3 are met, this self-reinforcing dynamic drives steady improvement; Otherwise overlap stagnates and training fails.

5 Practical Recipe

In Section 3, we identified two conditions for successful OPD. While possessing new knowledge is an intrinsic property of the teacher, the thinking-pattern gap between the teacher and the student can be narrowed through training design. In this section, we present two complementary strategies that recover OPD in otherwise failing configurations by improving the overlap dynamics.

Takeaways

- **Off-policy cold start (Section 5.1).** Fine-tuning the student on teacher-generated rollouts before OPD closes the initial thinking-pattern gap, leading to higher overlap from the start and consistently stronger final performance.
- **Teacher-aligned prompts (Section 5.2).** Using prompts from the teacher's post-training data sharpens alignment on high-probability tokens, although such prompts should be mixed with out-of-distribution prompts to prevent entropy collapse.

5.1 Off-Policy Distillation from Teacher Rollouts as Cold Start

When the student and teacher have substantially different thinking patterns, pure OPD can be ineffective because the teacher's token-level supervision is difficult for the student to exploit from its initial policy. To mitigate this mismatch, we consider a two-stage framework: we first perform off-policy distillation by supervised fine-tuning (SFT) the student on teacher-generated rollouts to bring it closer to the teacher's thinking pattern, and then continue training with standard OPD.

Setup. We study this setting using Qwen3-1.7B-Base as the student and Qwen3-4B (Non-thinking) as the teacher. We use the math-domain subset of OpenThoughts3-1.2M (Guha et al., 2025) as the prompt source for SFT. The teacher generates 200K responses on a subset of this dataset, and we use these teacher rollouts to perform SFT on the student as a cold start, yielding Qwen3-1.7B-SFT. We then continue training with OPD from this SFT initialization, using the remaining prompts from OpenThoughts after deduplicating against the SFT prompt subset (approximately 30K prompts). As a control, we compare against a pure-OPD baseline that starts directly from Qwen3-1.7B-Base and uses the same teacher and OPD prompt set, but performs no cold-start distillation before OPD. Rollout and training details are provided in Appendix D.1.

Results. As shown in Figure 8, the two-stage approach substantially outperforms pure OPD. Starting from Qwen3-1.7B-SFT yields consistently better validation performance than starting directly from Qwen3-1.7B-Base. Moreover, the performance gap persists throughout training, indicating that the off-policy cold start improves not only early optimization, but also the final performance ceiling of subsequent OPD.

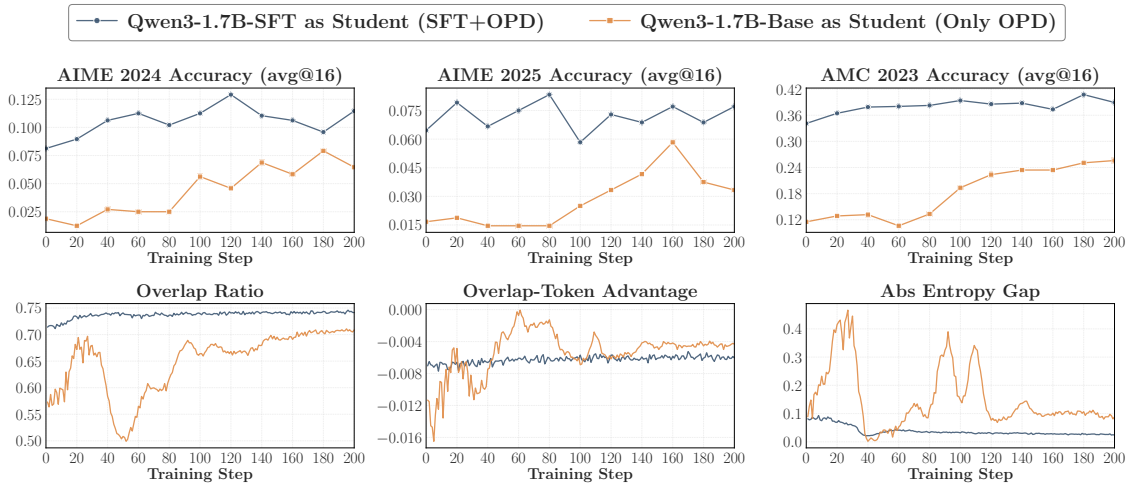


Figure 8: Effect of off-policy cold start before OPD, using fixed teacher Qwen3-4B (Non-thinking). The two curves correspond to OPD from Qwen3-1.7B-SFT and Qwen3-1.7B-Base.

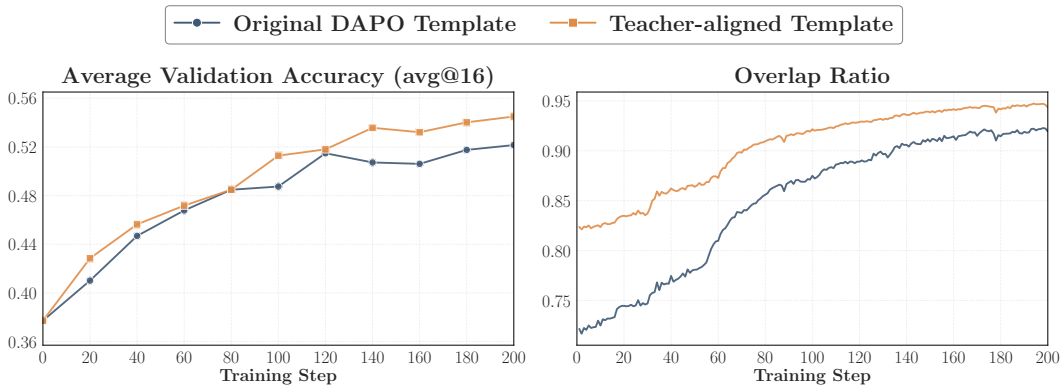


Figure 9: Effect of prompt template alignment. The teacher-aligned template yields higher accuracy and overlap growth throughout training.

The overlap dynamics support the same conclusion. The SFT-initialized student begins with a much higher overlap ratio and maintains a smooth, stable trajectory, whereas the base-initialized student starts lower and exhibits pronounced instability before gradually recovering. The entropy gap is also substantially smaller for the SFT-initialized student, indicating a closer match to the teacher's confidence profile from the outset. These observations confirm that off-policy distillation reduces the initial pattern mismatch, making the teacher's token-level supervision immediately exploitable once OPD begins. A more detailed analysis of the overlap mass dynamics is provided in Appendix D.2.

5.2 Leveraging Teacher Post-Training Prompts

The previous section narrows the thinking-pattern gap by applying SFT to the student using teacher rollouts. Another way to improve alignment is from the data side. Since the teacher's policy is shaped by the prompts seen during post-training, we find that using teacher-aligned prompts yields more effective supervision.

Setup. We conduct experiments at two granularities: whether matching the prompt *template* matters, and whether matching the prompt *content* matters.

- **Prompt template:** The teacher is JustRL-1.5B and the student is R1-Distill-1.5B. The prompt set is DAPO-Math-17K, with only the prompt template differing. The *original* template is the standard DAPO

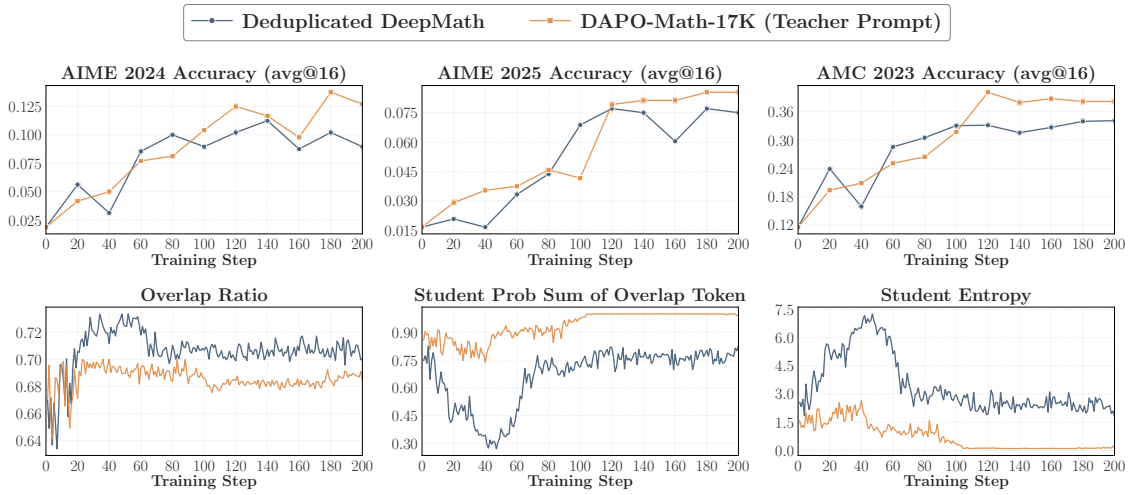


Figure 10: Effect of prompt content alignment. The teacher-aligned prompt contents yield stronger performance, with higher mass concentration on shared tokens and notably lower entropy.

format used in all previous experiments unless otherwise specified, while the *teacher-aligned* template matches the format used during JustRL post-training:

Original DAPO Template

Solve the following math problem step by step. The last line of your response should be of the form Answer: $Answer (without quotes) where $Answer is the answer to the problem. {Question} Remember to put your answer on its own line after "Answer:".

Teacher-Aligned Template

{Question} Please reason step by step, and put your final answer within \boxed{}.

Thus, the two runs use identical math problems but differ only in task presentation, isolating the effect of prompt-template alignment with the teacher while holding problem content constant.

- **Prompt content:** The teacher is Qwen3-4B-Base-GRPO introduced in Section 3.1 and the student is Qwen3-1.7B-Base. We compare two prompt sets of matched size: DAPO-Math-17K (aligned with the teacher's RL training datasets) and a subset of DeepMath, deduplicated against DAPO-Math-17K (see Appendix D.3). This design tests whether OPD benefits from using prompts that are identical to the teacher's post-training data, rather than prompts that are merely in-domain.

Results. The prompt template setting in Figure 9 shows that simply switching to the teacher-aligned template improves validation performance. The overlap dynamics support this result: the teacher-aligned template run begins with a higher overlap ratio and converges to a higher level, which indicates that even a minor change in prompt template can materially affect OPD by making the student's generated states more compatible with the teacher. The benchmark-wise breakdown in Appendix D.4 shows the same trend.

The prompt content setting in Figure 10 shows a similar downstream advantage but with a subtlety: teacher-aligned prompts produce a lower overlap ratio throughout training. However, the cumulative student probability mass on the overlap tokens is substantially higher, indicating that the student concentrates its mass on fewer but more strongly shared tokens. The effective alignment on high-probability tokens is therefore stronger, even though the overlap set is smaller.

At the same time, we observe that using teacher-aligned prompts leads to substantially lower student entropy during training. This suggests that performing OPD only on prompts seen during teacher post-training may not always be ideal, as it can overly reduce policy entropy. In practice, a more robust strategy may be to mix teacher-aligned prompts with prompts outside the teacher's post-training data in order to preserve policy entropy and maintain the student's capacity for exploration.

6 Related Work

Knowledge Distillation. Knowledge distillation (KD) (Hinton et al., 2015) transfers knowledge from a large model to a smaller one by training a student network on the soft output distributions of a teacher. For autoregressive sequence models, Kim & Rush (2016) extended this to sequence-level distillation by training students on teacher-generated outputs, establishing the dominant off-policy distillation baseline (Sanh et al., 2019; Jiao et al., 2020; Wang et al., 2020). In parallel, supervised fine-tuning (SFT) has been directly applied to improve performance on a variety of downstream tasks (Chung et al., 2024; Sanh et al., 2021; Wei et al., 2021). A fundamental limitation shared by all off-policy approaches is the train-inference distribution mismatch. The student is optimized on teacher-generated or reference sequences, but must generate from its own distribution at inference, which is an instance of the exposure bias (Bengio et al., 2015) that accumulates errors over long generations. This mismatch motivates shifting distillation to the student's own on-policy distribution, which is the central idea behind on-policy distillation.

On-Policy Distillation. MiniLLM (Gu et al., 2023) first formalized on-policy distillation (OPD) for LLMs under a reverse KL objective optimized via policy gradient, arguing that reverse KL's mode-seeking behavior prevents the student from spreading probability mass over regions the teacher considers unlikely. GKD (Agarwal et al., 2024) introduced a unified framework interpolating between on-policy and off-policy data across multiple divergences, demonstrating consistent gains over other KD baselines. Yang et al. (2026b) later formalized OPD theoretically as a special case of dense KL-constrained RL, showing that the teacher's per-token log-ratio constitutes an implicit reward and that scaling this reward beyond its standard weight can push the student past the teacher's performance boundary. OPD has since been adopted in industry post-training pipelines (Yang et al., 2025; Lu & Lab, 2025; Zeng et al., 2026; Xiao et al., 2026; Ko et al., 2026; Jin et al., 2026; Jang et al., 2026; Fu et al., 2026; Yang et al., 2026b), and extended to scalable self-distillation (Hübotter et al., 2026; Zhao et al., 2026b; He et al., 2026; Shenfeld et al., 2026; Ye et al., 2026b; Sang et al., 2026; Kim et al., 2026; Ye et al., 2026a; Yang et al., 2026a; Li et al., 2026; Zhao et al., 2026a; Ding, 2026), where a single model acts as its own teacher by conditioning on privileged information such as ground-truth solutions or execution feedback. Despite this growing body of work, existing studies focus on demonstrating OPD's promise, such as dense rewards and mitigated exposure bias, across varied objectives, tasks, and teacher-student pairs, without systematically analyzing when or why OPD fails.

Capacity Gap and Distillability. A recurring observation in knowledge distillation is that large teacher-student capacity gaps can degrade or even reverse the distillation gains. Cho & Hariharan (2019) demonstrate that distillation can hurt student performance when the teacher is substantially more capable, and Mirzadeh et al. (2020) propose an intermediate-sized teacher assistant to bridge the gap. Busbridge et al. (2025) provide a quantitative treatment via distillation scaling laws, showing that student loss follows a power law as a function of teacher quality, student size, and data volume, identifying a U-shaped capacity regime where teacher over-capability degrades distillation efficiency. For LLM reasoning, Li et al. (2025) document a "learnability gap" showing that training small models on long chain-of-thought traces from strong reasoning teachers consistently underperforms simpler approaches, suggesting that the reasoning complexity of teacher outputs must match the student capacity. These findings caution against assuming distillation is universally effective. However, prior analyses have largely focused on off-policy knowledge distillation, leaving capacity gaps and distillability in OPD underexplored.

7 Conclusion

This work provides a systematic analysis of OPD, decomposing its success into two governing conditions: thinking-pattern consistency and the presence of genuinely new knowledge beyond what the student has seen during training. When these conditions are unmet, off-policy cold start and teacher-aligned prompt selection provide effective remedies.

Limitations. Our experiments focus on mathematical reasoning and a limited set of model families, so generalization to code generation, open-ended dialogue, and larger-scale models remains open. We use top-k overlap as a proxy for thinking-pattern consistency, and the anisotropy hypothesis in Appendix E.2 still requires direct gradient-geometry validation.

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 *The twelfth international conference on learning representations*, 2024.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

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

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4794–4802, 2019.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL http://jmlr.org/papers/v25/23-0870.html.

Ken Ding. Hdpo: Hybrid distillation policy optimization via privileged self-distillation. *arXiv preprint arXiv:2603.23871*, 2026.

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

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

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

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.

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

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

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

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

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

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

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 4163–4174, 2020.

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

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

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

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

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

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9, 2024.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25366–25394, 2025.

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.

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.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019. URL http://arxiv.org/abs/1908.10084.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation, 2026. URL https://arxiv.org/abs/2603.05433.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2021.

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

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

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

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.

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

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

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

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

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

Guoliang Zhao, Ruobing Xie, An Wang, Shuaipeng Li, Huaibing Xie, and Xingwu Sun. Self-distillation for multi-token prediction, 2026a. URL https://arxiv.org/abs/2603.23911.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026b.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL http://arxiv.org/abs/2403.13372.

A Details for Preliminaries

A.1 OPD Objective and Granularity Variants

Sequence-Level Objective. Given a prompt $x \sim \mathcal{D}_x$, let $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T) \sim \pi_\theta(\cdot \mid x)$ be a student rollout. A standard OPD formulation minimizes the sequence-level reverse KL:

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x}\Big[D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid x) \parallel \pi_T(\cdot \mid x)\big)\Big]. \tag{5}$$

For each prefix $\hat{y}_{<t}$, define $p_t(v) \triangleq \pi_\theta(v \mid x, \hat{y}_{<t})$ and $q_t(v) \triangleq \pi_T(v \mid x, \hat{y}_{<t})$. Using the autoregressive factorization, the sequence-level objective has the exact token-level decomposition in Eq. equation 1. In practice, implementations approximate or instantiate this decomposition at different granularities.

Sampled-Token OPD. The most lightweight variant evaluates only the token sampled by the student, and is also the most common implementation in prior on-policy distillation work (Lu & Lab, 2025; Xiao et al., 2026; Yang et al., 2026b). Given $\hat{y}_t \sim p_t$, the per-token loss is $\ell_t^{\mathrm{sample}} \triangleq \log p_t(\hat{y}_t) - \log q_t(\hat{y}_t)$, aggregated as:

$$\mathcal{L}_{\mathrm{OPD}}^{\mathrm{sample}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x,\, \hat{y} \sim \pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \ell_t^{\mathrm{sample}}\right]. \tag{6}$$

Since $\mathbb{E}_{\hat{y}_t \sim p_t}[\ell_t^{\mathrm{sample}}] = D_{\mathrm{KL}}(p_t \| q_t)$, each ℓ_t^{sample} is an unbiased single-sample estimator of the token-level reverse KL.

Full-Vocabulary OPD. At the other extreme, one computes the divergence over the entire vocabulary at each prefix:

$$\mathcal{L}_{\mathrm{OPD}}^{\mathrm{full}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x,\, \hat{y} \sim \pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}(p_t \| q_t)\right]. \tag{7}$$

This yields denser gradients compared to sampled-token OPD, at the cost of $O(BTM)$ memory for batch size B, sequence length T and vocabulary size $M = |\mathcal{V}|$.

Top-k OPD. Top-k OPD restricts the divergence computation to a subset $S_t \subseteq \mathcal{V}$. For the student top-$k$ variant, $S_t = \mathrm{TopK}(p_t, k)$. Define the renormalized student and teacher distributions on S_t as:

$$\bar{p}_t^{(S_t)}(v) = \frac{p_t(v)\,\mathbf{1}[v \in S_t]}{\sum_{u \in S_t} p_t(u)}, \qquad \bar{q}_t^{(S_t)}(v) = \frac{q_t(v)\,\mathbf{1}[v \in S_t]}{\sum_{u \in S_t} q_t(u)}.$$

Distillation is then performed by minimizing the subset KL divergence, yielding:

$$\mathcal{L}_{\mathrm{OPD}}^{\text{top-k}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}_x,\, \hat{y} \sim \pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}\big(\bar{p}_t^{(S_t)} \parallel \bar{q}_t^{(S_t)}\big)\right]. \tag{8}$$

This formulation discards mass outside S_t and therefore remains an approximation to the full-vocabulary reverse KL, but it substantially reduces teacher-query cost while preserving multi-token supervision on the student's high-probability region.

B Details for Section 3

B.1 GRPO Training Details

Base Model. We initialize GRPO training from Qwen3-4B-Base.

Training Dataset. We use the processed DAPO-Math-17K dataset for GRPO training. Specifically, each question is augmented with the following instruction:

GRPO dataset template

{Question} Please reason step by step, and put your final answer within \boxed{}.

Training and Evaluation Settings. We train the teacher model using GRPO. During training, we sample $n = 8$ responses for each prompt. The maximum prompt length and maximum response length are set to 1,024 and 7,168 tokens, respectively. Training is conducted for one epoch on 8 A800 80G GPUs with a learning rate of 1×10^{-6}. We set both the student sampling temperature and the teacher temperature to 1.0, use a repetition penalty of 1.0, disable KL regularization, and adopt `token-mean` loss aggregation. The main hyperparameters are summarized in Table 1.

Table 1: Training hyperparameters of GRPO for Qwen3-4B-Base-GRPO.

Hyper-parameter	**Value**
Base model	Qwen3-4B-Base
RL algorithm	GRPO
Training epochs	1
Train batch size	64
Micro batch size	64
Rollout n	8
Maximum prompt length	1,024
Maximum response length	7,168
Validation max response length	31,744
Learning rate	1×10^{-6}
Temperature	1.0
Top-p	1.0
KL regularization	0.0
Loss aggregation	`token-mean`
KL Coefficient	0.0

B.2 Experimental Setup

Unless otherwise noted, all experiments use the default OPD hyperparameters listed in Table 2.

B.3 Benchmark-wise breakdown of thinking-pattern compatibility

To further unpack the averaged result in Figure 2, Figure 11 presents a benchmark-wise breakdown. The advantage of distillation from Qwen3-4B-Base-GRPO is broadly consistent across datasets rather than being driven by a single benchmark. The gap is more pronounced on AMC 2023 and AIME 2024, and smaller but still generally present on AIME 2025. This per-benchmark view supports the interpretation that better early-stage thinking-pattern compatibility leads to better downstream distillation performance, and the loss from an early mismatch is not fully recovered later in training.

C Details for Section 4

C.1 Additional Analysis of Token Overlap Mass

To quantify how much probability mass each model assigns to the overlap top-k region, we define:

$$\mathcal{M}^{(p)}_{\text{overlap-mass}} = \mathbb{E}_t \left[\sum_{v \in S_t^{(p)} \cap S_t^{(q)}} p_t(v) \right], \tag{9}$$

Table 2: **Default hyperparameters for OPD.**

Item	Value
Training temperature	1.0
Global batch size	64
Mini batch size	64
Rollout number	4
LogProb top-K	16
Top-K strategy	Student Top-K
Top-p	1.0
Max prompt length	1024
Max response length	7168
Learning rate	1e-6
Epoch	1
KL Coefficient	0.0

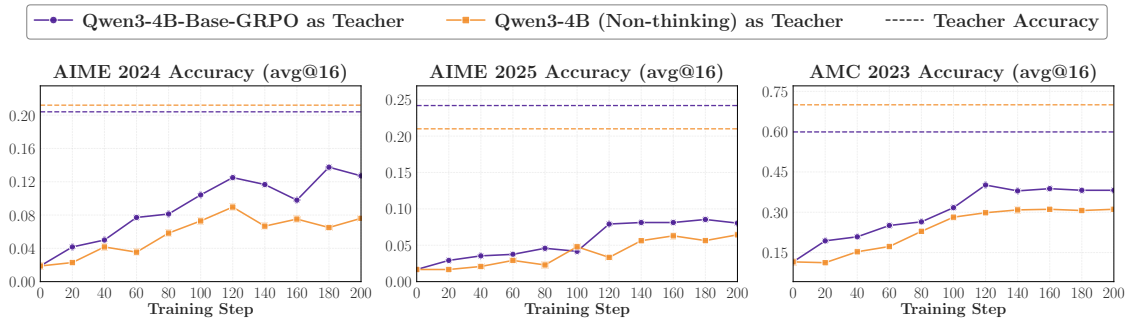


Figure 11: Benchmark-wise breakdown of the average validation accuracy shown in Figure 2. We report results on AIME 2024, AIME 2025, and AMC 2023 separately. Distillation from Qwen3-4B-Base-GRPO consistently matches or outperforms distillation from Qwen3-4B (Non-thinking) across the three benchmarks.

and

$$\mathcal{M}^{(q)}_{\text{overlap-mass}} = \mathbb{E}_t \left[\sum_{v \in S_t^{(p)} \cap S_t^{(q)}} q_t(v) \right], \tag{10}$$

which measure the fraction of total probability mass that the student and teacher, respectively, assign to the shared tokens in their top-k sets. In our experiments, the overlap tokens carry 97%–99% of the total probability mass for both models throughout training, as shown in Figure 12.

C.2 Auxiliary Optimization Dynamics

To complement the analysis in Section 4.1, we report several additional optimization diagnostics for the same contrastive setting. Throughout this appendix, we fix the student to R1-Distill-1.5B and compare two teachers under the same Student Top-k OPD training recipe: JustRL-1.5B, which yields a successful run, and R1-Distill-7B, which yields a failing run under otherwise matched conditions. These diagnostics are not intended as primary evidence; rather, they provide a complementary view of how the optimization signal differs between successful and failing OPD.

Diagnostics. We monitor three additional quantities. The first is the batch-averaged OPD training loss, denoted as *PG Loss* in Figure 13. The second is the gradient norm, which measures the overall magnitude of the update signal reaching the student. The third is the probability difference $p_t(v) - q_t(v)$ on the token with the largest absolute advantage, which tracks whether the student can reduce the most pronounced local disagreement with the teacher on the tokens that carry the strongest optimization signal. Together, these

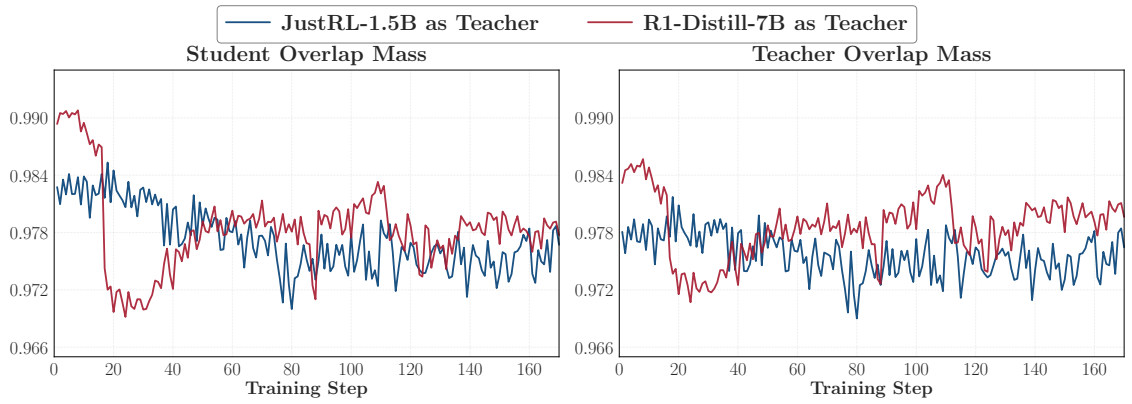


Figure 12: Probability mass assigned to overlap tokens during training. For both the student and teacher distributions, the overlap tokens consistently account for roughly 97%–99% of the total probability mass, indicating that the overlap is not only increasing at the set level but also dominates the probability distribution.

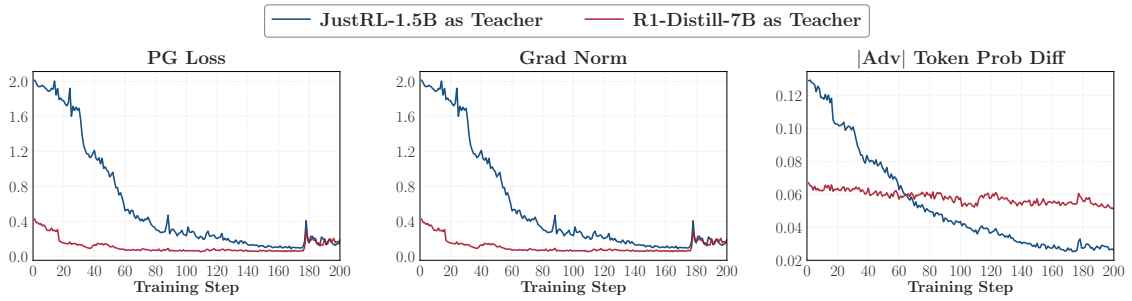


Figure 13: Auxiliary optimization diagnostics for the contrastive OPD setting in Section 4.1, using R1-Distill-1.5B as the student and comparing JustRL-1.5B against R1-Distill-7B as the teacher. **Left:** batch-averaged OPD training loss (*PG Loss*) over training. **Middle:** gradient norm over training. **Right:** probability difference $p_t(v) - q_t(v)$ measured on the token with the largest absolute advantage. The successful run exhibits a large reduction in optimization loss, sustained gradient magnitude, and a steady decrease in extreme-token probability mismatch. By contrast, the failing run starts with and maintains much weaker gradients, and its extreme-token probability discrepancy remains noticeably larger throughout training.

metrics help distinguish between successful and failing OPD: in the former, the student receives a usable signal and progressively reduces mismatch, whereas in the latter, the signal is too weak or too poorly aligned to drive substantial improvement.

Results. The trends in Figure 13 are consistent with the main conclusion of Section 4.1. First, the successful run with JustRL-1.5B shows a pronounced reduction in training loss over the course of optimization. Starting from a much larger initial mismatch, the loss decreases steadily for most of training before flattening at a low value. By contrast, the failing run with R1-Distill-7B begins with a much smaller loss and changes only modestly thereafter. This pattern suggests that the smaller loss in the failing run does not indicate better optimization. Rather, it reflects a weak teacher-induced training signal from the outset, which remains too small to drive substantial policy improvement.

Second, the gradient norm shows an even clearer separation between the two runs. In the successful run, the gradient norm is initially large and remains substantial through a long portion of training, indicating that the student continues to receive a meaningful corrective signal. In the failing run, the gradient norm is consistently much smaller, with only limited variation over time. Thus, even though optimization proceeds under the same algorithm and training budget, the student trained against R1-Distill-7B experiences a much weaker update signal. This observation is consistent with the finding that failure is associated with poor

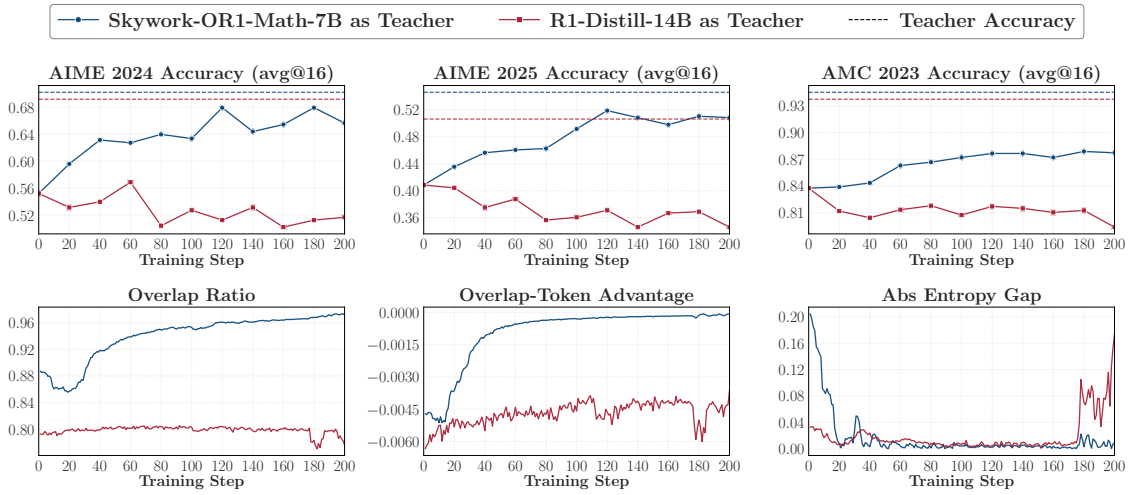


Figure 14: Cross-model validation with a fixed student (R1-Distill-7B) and two teachers. **Top:** avg@16 accuracy on AIME 2024, AIME 2025, and AMC 2023. **Bottom:** overlap ratio, overlap-token advantage, and absolute entropy gap over training. The successful run is again accompanied by increasing high-probability-token alignment, while the stagnating run is not.

alignment on high-probability tokens: when the student does not meaningfully enter the teacher-supported region, the resulting gradients remain weak.

Third, the right panel shows that the successful run steadily reduces the probability discrepancy on the token with the largest absolute advantage, whereas the failing run maintains a noticeably larger gap throughout training. In other words, when OPD succeeds, the student progressively corrects the local mistakes that matter most under the teacher-induced advantage signal. When OPD fails, these high-advantage discrepancies persist rather than being resolved. This is again consistent with the interpretation that the decisive signal in OPD lies on a small set of high-probability, high-advantage tokens, and failure occurs when the student cannot effectively exploit that signal.

Taken together, these auxiliary dynamics reinforce the interpretation developed in Section 4.1. Successful OPD is characterized not only by increasing overlap on high-probability tokens, but also by a training regime in which the student receives gradients of sufficient magnitude to reduce the most important local distributional mismatches. In contrast, failing OPD is accompanied by weak gradients, limited loss reduction, and persistent disagreement on the tokens with the strongest advantage signal. While these diagnostics are supportive rather than central, they provide an optimization-level view that is fully consistent with that the useful learning signal of OPD is concentrated on high-probability tokens at student-visited states, and training degrades when that signal is too weak or too misaligned to drive effective updates.

C.3 Cross-Model Validation of High-Probability-Token Alignment

We further test whether the phenomenon in Section 4.1 generalizes to another model pair. Here we fix the student model to R1-Distill-7B and choose Skywork-OR1-Math-7B and DeepSeek-R1-Distill-Qwen-14B (R1-Distill-14B) as teachers, using the same training and evaluation setup as in Section 4.1.

Results. Figure 14 shows the same pattern as Figure 6. With Skywork-OR1-Math-7B as the teacher, distillation improves student performance and is accompanied by steadily increasing overlap ratio, overlap-token advantage approaching zero, and a small entropy gap. In contrast, with R1-Distill-14B as the teacher, training shows little improvement and the alignment metrics remain poor or unstable. This provides additional evidence that successful OPD consistently coincides with the emergence of high-probability-token alignment at student-visited states.

Table 3: SFT hyperparameters for cold-start distillation from Qwen3-4B (Non-thinking) to Qwen3-1.7B-Base.

Hyper-parameter	**Value**
Student model	Qwen3-1.7B-Base
Training objective	Full-parameter SFT
Template	`qwen3`
Training epochs	1
Sequence length	14,336
Per-device batch size	8
Gradient accumulation steps	1
Learning rate	1×10^{-5}
LR scheduler	Cosine
Warmup ratio	0.05
Precision	BF16

D Details for Section 5

D.1 Cold-Start Distillation Details

Offline teacher rollout. To construct the cold-start SFT data, we sample 200K math prompts from the math subset of OpenThoughts3-1.2M (Guha et al., 2025) and use Qwen3-4B (Non-thinking) to generate one offline response for each prompt. For each prompt, we use the following template:

Teacher rollout template

{Question} Please reason step by step, and put your final answer within \boxed{}.

We decode with temperature 0.7, top-$p = 0.95$, top-$k = -1$, and a maximum generation length of 12,288 tokens. After generation, we filter out incomplete responses (e.g., truncated outputs that do not finish properly) and degenerate repetitive responses. The remaining prompt-response pairs are used as the supervised distillation corpus for training the student.

Student SFT. Starting from Qwen3-1.7B-Base, we perform full-parameter SFT on the filtered 200K teacher-generated samples using the LLaMA-Factory framework (Zheng et al., 2024), yielding Qwen3-1.7B-SFT. We summarize the detailed hyperparameters in Table 3.

D.2 Additional Analysis of Overlap Mass

To better understand why the base-initialized student can occasionally exhibit a comparable or even slightly better Overlap-Token Advantage while still underperforming overall, we further examine the probability mass covered by the overlap set from both the student and teacher sides. As shown in Figure 15, the SFT-initialized student maintains both student overlap mass and teacher overlap mass at consistently high levels throughout training. This indicates that the overlap tokens cover most of the high-probability regions of both the student and teacher distributions, suggesting a strong and stable alignment from the beginning of OPD. In contrast, the base-initialized student exhibits substantially lower and more unstable overlap mass, especially in the early stage of training.

This analysis helps explain why Overlap-Token Advantage alone can sometimes be misleading. Since it is averaged only over overlap tokens, it can appear relatively favorable even when the overlap set itself misses substantial high-probability teacher tokens. Overlap mass complements this view by revealing whether the shared support actually covers the most important parts of the two distributions. From this perspective, the SFT cold start leads to a substantially better and more stable match between student and teacher.

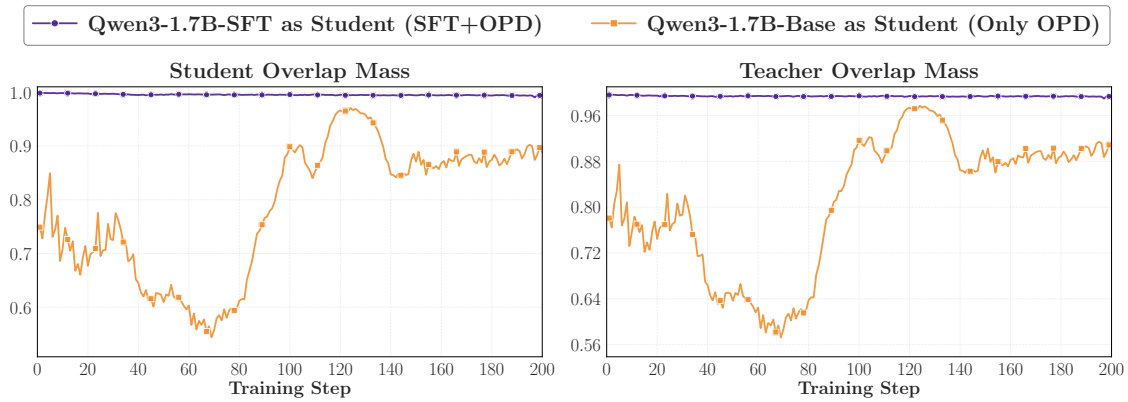


Figure 15: Student overlap mass and teacher overlap mass during training for SFT-initialized and base-initialized students.

D.3 Deduplication Details for the DeepMath Subset

For the cross-size setting, we construct a DeepMath subset deduplicated against DAPO-Math-17K in order to compare prompts aligned with the teacher's RL post-training data against prompts that are only in-domain.

Our deduplication is performed in two stages: exact-match deduplication and semantic deduplication.

Question extraction. For both DAPO-Math-17K and DeepMath, we extract the question content and remove the instruction suffix in the prompt, so that deduplication is performed based on the question text alone.

Stage 1: Exact-match deduplication. We collect all extracted DAPO-Math-17K questions into a set and remove any DeepMath example whose extracted question exactly matches one of the DAPO questions.

Stage 2: Semantic deduplication. To further remove near-duplicate prompts, we encode both DAPO-Math-17K and DeepMath questions using the sentence embedding model all-mpnet-base-v2 (Reimers & Gurevych, 2019). We L2-normalize the embeddings and build a FAISS inner-product index over the DAPO embeddings, so that the inner product corresponds to cosine similarity. For each DeepMath question, we retrieve its top-1 nearest neighbor in DAPO-Math-17K. If the cosine similarity to the nearest DAPO question is at least 0.6, we mark the DeepMath example as a semantic duplicate and remove it.

Final retained subset. We remove any DeepMath example flagged by either exact-match or semantic deduplication. The resulting subset is in-domain but deduplicated against DAPO-Math-17K, enabling a controlled comparison between prompts that overlap with the teacher's post-training data and prompts that are only in-domain.

D.4 Benchmark-wise breakdown of prompt-template alignment

To further unpack the averaged result in Figure 9, Figure 16 presents a benchmark-wise breakdown. The teacher-aligned template yields broadly consistent improvements across datasets, with larger gains on the two AIME sets and a smaller but still positive effect on AMC 2023. It also allows the student to recover a larger fraction of the teacher's performance, increasing from roughly 80% to roughly 85%. Together with the overlap-ratio result in Section 5.2, this suggests that prompt-template alignment improves OPD by making the student's generated states more compatible with the teacher.

E Discussion

The appeal of OPD lies in its dense supervision, where every token receives a reward signal from the teacher, in contrast to the sparse outcome-level reward used in RL. However, this increased supervision density comes

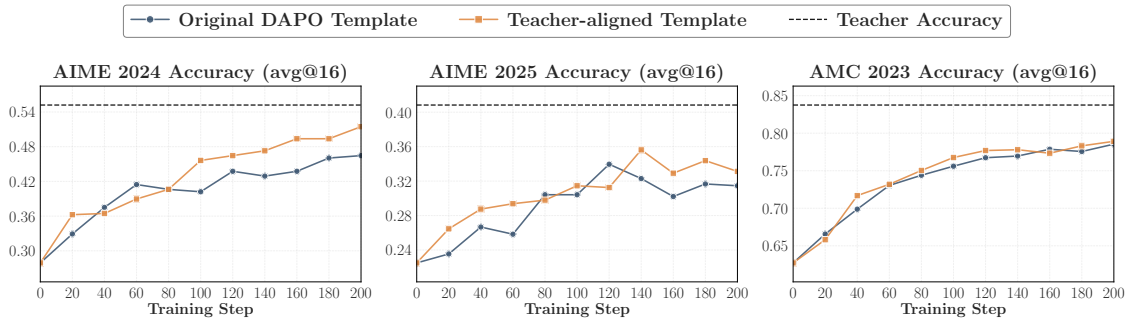


Figure 16: Benchmark-wise breakdown of the average validation accuracy shown in Figure 9. Using the teacher-aligned template consistently matches or outperforms the original DAPO template across the three benchmarks.

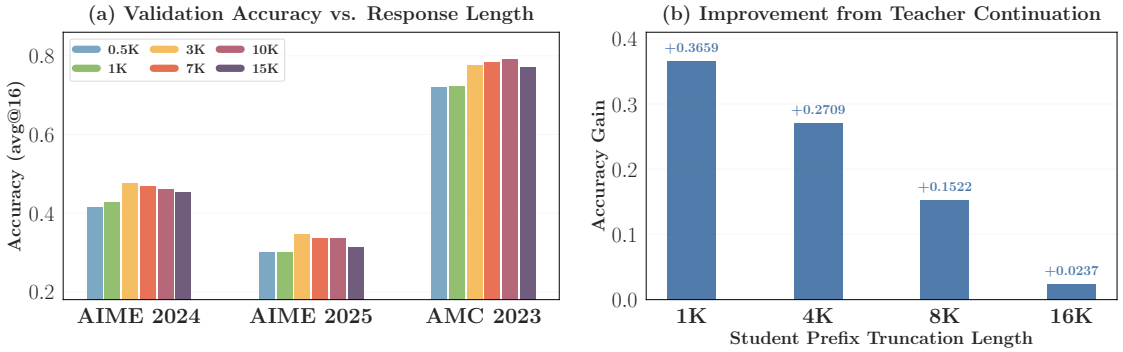


Figure 17: (a) Validation accuracy on three benchmarks under different response lengths. (b) Accuracy gain from teacher continuation under different student prefix truncation lengths.

at a cost. The above sections all implicitly depend on the teacher's token-level reward being reliable in student-visited states, yet we have seen that this assumption can break down. In this section, we investigate the reward signal itself and examine its properties and limitations.

E.1 Reward Quality Degrades with Trajectory Depth

We first investigate how the teacher's reward quality varies with response length.

Response length exhibits a sweet spot. The supervision at position t depends on the teacher's conditional $\pi_T(y_t \mid x, y_{<t})$ under a student-generated prefix $y_{<t}$, which may drift from trajectories the teacher would naturally produce. We train R1-Distill-1.5B against JustRL-1.5B across six maximum response lengths for 200 steps. As shown in Figure 17(a), very short responses (0.5K and 1K) provide too few supervised tokens for sample-efficient learning, while moderate lengths (3K and 7K) yield the strongest results. Beyond this range (10K and 15K), performance plateaus or declines. The training dynamics in Figure 18 confirm that moderate lengths produce smooth overlap growth, whereas 10K and 15K exhibit late-stage collapse, with the overlap ratio dropping sharply, accompanied by spikes in student entropy and gradient norm.

Instability originates at later tokens. Where does this collapse begin? In the 15K setting, analyzing student entropy as a function of output position across training steps reveals a clear back-to-front pattern: as shown in Figure 19, high entropy first appears at the end of the response and progressively propagates toward earlier tokens as training proceeds. Teacher entropy exhibits a similar suffix-to-prefix trend (see Appendix E.1.1), consistent with the teacher encountering increasingly unfamiliar prefixes at later positions and producing noisier reward that in turn destabilizes the student.

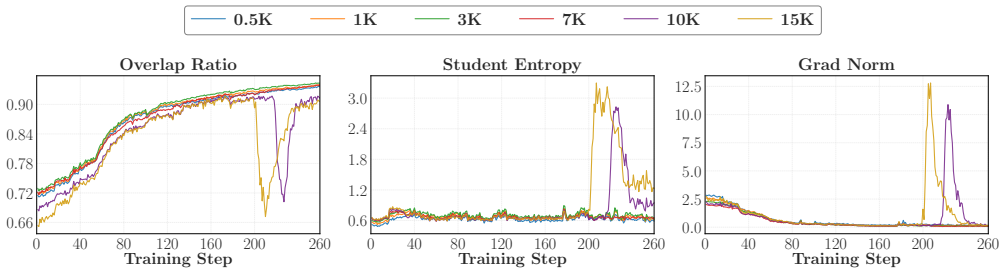


Figure 18: Training dynamics under different maximum response lengths for OPD.

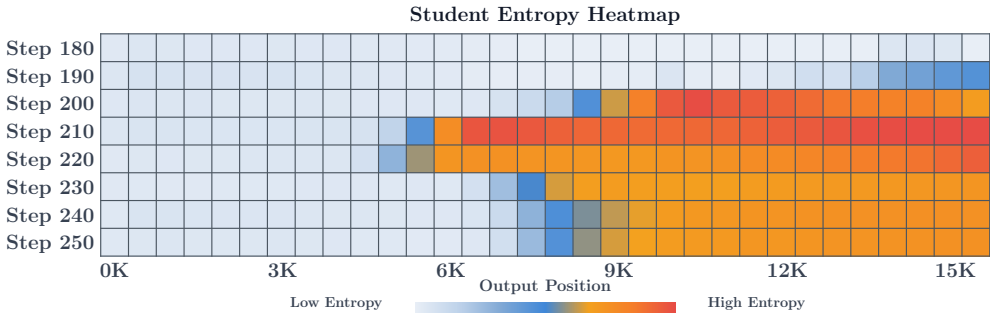


Figure 19: Average student entropy across decoding positions during OPD training with 15K max response length, measured on student-generated trajectories from Step 180 to Step 250.

E.1.1 Teacher entropy by output position

To complement the student entropy analysis in Appendix E.1, we also visualize teacher entropy as a function of output position across training steps under the $15K$ max response length setting (see Figure 20). Similar to the student, teacher entropy first increases at later decoding positions and then progressively propagates toward earlier tokens over training.

Teacher continuation degrades with prefix depth. We further probe this by testing whether the teacher can still improve upon the student's continuation when starting from a student-generated prefix. We sample 2K prompts from DAPO-Math-17K, generate full student rollouts, and select those exceeding 16K tokens. We then truncate each rollout at multiple positions and let the teacher continue from the resulting prefix. Figure 17(b) shows that the teacher's accuracy advantage decreases monotonically, from +0.37 at a 1K prefix to just +0.02 at a 16K prefix.

Together, these results reveal a fundamental tradeoff in OPD's token-level supervision. Dense reward is effective on moderately long reasoning traces, but its reliability degrades with depth as the student prefix drifts further from the states familiar to the teacher. This suggests that OPD may not extend cleanly to longer-horizon settings such as extended chain-of-thought or agentic multi-turn interaction.

E.2 Globally Informative Reward Does Not Guarantee Local Exploitability

The previous subsection shows that reward quality degrades with trajectory depth. A natural follow-up question is whether the reward signal is fundamentally uninformative in failing OPD configurations or whether the source of failure lies elsewhere.

Setup. We revisit the controlled comparison from Section 4.1, with R1-Distill-1.5B as the student and two teachers: JustRL-1.5B (successful OPD) and R1-Distill-7B (failed OPD). For each student rollout y, we compute the sequence mean reward $\bar{r}(y) = \frac{1}{T}\sum_{t=1}^{T}\left[\log \pi_T(y_t \mid x, y_{<t}) - \log \pi_\theta(y_t \mid x, y_{<t})\right]$ under sampled-token OPD, and compare the distribution of $\bar{r}(y)$ between correct and incorrect rollouts.

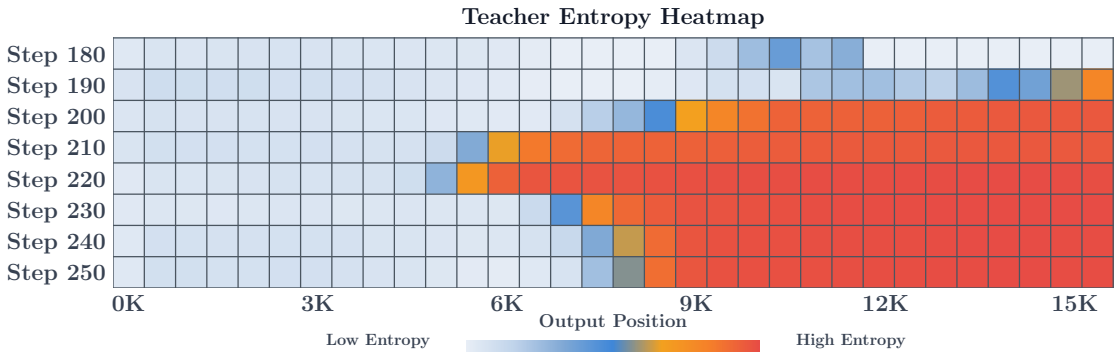


Figure 20: Average teacher entropy across decoding positions during OPD training with 15K max response length, measured on student-generated trajectories from Step 180 to Step 250. Elevated entropy first emerges in the suffix and gradually propagates toward earlier output positions over training.

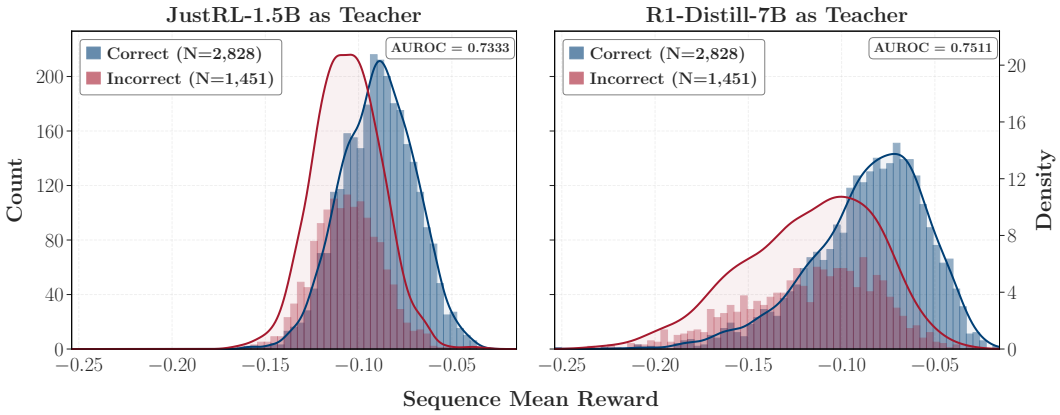


Figure 21: Sequence mean reward distributions for correct and incorrect student rollouts. Both teachers assign higher reward to correct rollouts with comparable AUROC (0.73 and 0.75).

Global reward structure is preserved in both settings. Figure 21 shows that, for both teachers, correct rollouts consistently receive higher sequence mean reward than incorrect ones, with comparable AUROC values (0.73 for JustRL-1.5B, 0.75 for R1-Distill-7B). The failing 7B teacher does not produce a weaker global signal, which is equally correlated with rollout correctness.

A hypothesis on local optimization geometry. If the reward is globally informative in both cases, why does OPD fail with the 7B teacher? The training dynamics from Section 4.1 offer a clue. As shown in Figure 6, when R1-Distill-7B serves as the teacher, the overlap-token advantage becomes larger in magnitude than in the JustRL setting during the later stages of training, yet the gradient norm remains persistently smaller (see Appendix C.2). One possible explanation is that the 7B teacher's per-token advantages, while individually large, are anisotropic across positions within each sequence. When these heterogeneous signals are aggregated into a gradient update, they partially cancel, yielding small effective gradients despite large per-token rewards. By contrast, JustRL-1.5B, which shares a compatible thinking pattern with the student, may concentrate its advantage on a more coherent subset of tokens. The resulting gradient, though composed of smaller per-token signals, points in a consistent direction that reverse KL can amplify through its mode-seeking behavior.

We have not directly verified this anisotropy hypothesis, and doing so would require analyzing the directional structure of per-token gradients, which we leave to future work. Nonetheless, the co-occurrence of high per-token advantage and low gradient norm is suggestive and points to an important distinction that a globally informative reward does not guarantee a locally exploitable one. Understanding the geometry of

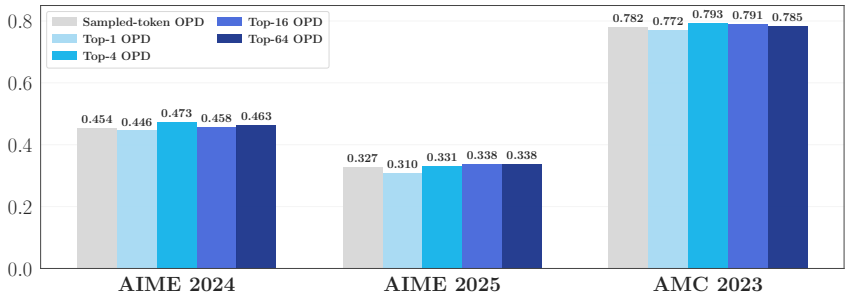


Figure 22: Effect of the support size k in Top-k OPD. All numbers are reported as avg@16.

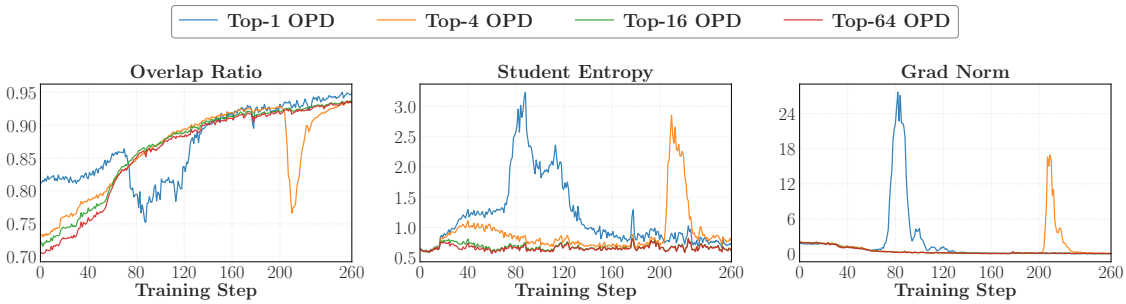


Figure 23: Training dynamics under different support sizes k for Top-k OPD.

OPD's reward landscape, and developing objectives that can exploit anisotropic reward structures, remains an open question.

E.3 Sampled-Token Reward Is Already Sufficient

A natural question about OPD's reward is how many tokens per position are needed to compute a useful gradient. Top-k OPD aggregates the reward over the k highest-probability tokens at each position, and one might expect that larger support always leads to better or more stable learning. We investigate this by varying k and comparing against the simpler sampled-token OPD, which uses only a single token drawn from the student distribution at each position.

Setup. We use R1-Distill-1.5B as the student and JustRL-1.5B as the teacher, and compare Top-k OPD with $k \in \{1, 4, 16, 64\}$ against sampled-token OPD, keeping all other hyperparameters fixed.

Results. Figure 22 shows that sampled-token OPD achieves performance comparable to that of the Top-k settings averaged on three benchmarks. The only clearly worse configuration is Top-1, which consistently underperforms. Enlarging k beyond 4 brings negligible additional gain while leading to greater computational overhead. Figure 23 shows the training dynamics and reveals where the differences arise. Top-1 exhibits unstable overlap growth, accompanied by sharp spikes in entropy and gradient norm. Top-4 is substantially more stable but still shows a late-stage dip. Top-16 and Top-64 remain smooth throughout.

Overall, these results suggest that the support size may not be a critical design choice for OPD, as long as the degenerate Top-1 setting is avoided. The reason sampled-token OPD works well despite using only one token per position is that it draws a different token at each step proportionally to the student's own distribution, providing unbiased coverage of the high-probability region across training. Top-1, by contrast, always selects the argmax token, thereby concentrating the reward on a single mode. Small policy changes can flip which token occupies rank 1, creating an unstable reward signal that does not average out over training. The failure of Top-1 is therefore not about using too few tokens, but about using a biased, mode-concentrated selection rule.

F Future Work

Building on our findings, we identify several directions for future research:

- **Beyond Mathematical Reasoning:** All experiments in this work are conducted on mathematical benchmarks. Whether the same conditions and token-level mechanisms govern OPD in other domains such as code and open-ended settings remains an important open question.
- **Impact of Pre-Training:** The "new knowledge" condition implicitly depends on differences in pre-training corpora, but isolating this factor is challenging. Current studies mainly rely on cross-family distillation, which confounds data divergence with tokenizer mismatch and architectural differences, while controlled pre-training ablations remain prohibitively expensive. As a result, measuring the effect of pre-training data on OPD remains an open problem.
- **Self-Distillation Dynamics:** Recent work increasingly adopts self-distillation, where a single model serves as its own teacher given privileged information. Extending these insights to the self-distillation regime, where thinking-pattern consistency is guaranteed but knowledge novelty arises from privileged access rather than a separate teacher, is a natural next step.
- **Long-Horizon and Agentic Settings:** The trajectory-length ceiling revealed in Appendix E motivates hybrid approaches that combine dense token-level supervision on short segments with sparse outcome-level rewards for longer horizons, as well as curriculum strategies that progressively extend the supervised horizon during training.