PRUNE-OPD: EFFICIENT AND RELIABLE ON-POLICY DISTILLATION FOR LONG-HORIZON REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) provides dense teacher feedback on student-generated responses, but its rollout budget does not account for how well the student and teacher match along those responses. We identify this unused mismatch information as an opportunity to accelerate OPD: long, weakly compatible suffixes can be curtailed while retaining the supervision needed for learning. We introduce **Prune-OPD**, which uses top-k overlap to measure local compatibility, reweight rewards after cumulative mismatches, and adaptively truncate training rollouts. Within configured length bounds, the algorithm determines and updates the truncation length from compatibility feedback, without a manually specified cutoff or length schedule for each pair. Across four student–teacher pairs, Prune-OPD reduces reported training time by 37.6%–68.0% while broadly preserving accuracy on AMC, AIME, and HMMT. In the high-compatibility DeepSeek-7B / Skywork-7B pair, Prune-OPD retains long supervision and stays close to OPD accuracy, avoiding the larger losses from fixed-4K truncation. Prune-OPD thus saves training computation where compatible supervision is short, while retaining longer rollouts where they are needed to preserve performance.

1 INTRODUCTION

On-policy distillation (OPD) trains a student on its own responses using token-level feedback from a teacher (Gu et al., 2024; Agarwal et al., 2024). The teacher evaluates the prefixes the student actually visits, reducing the trajectory mismatch associated with learning only from teacher-generated answers. It also supplies feedback throughout the response, rather than only after a final answer. Recent language-model post-training recipes use OPD to transfer reasoning capabilities (Yang et al., 2025a; Lu & Lab, 2025; Xiao et al., 2026; Zeng et al., 2026).

Dense teacher feedback does not make every part of a student rollout equally useful for distillation. As a response develops, the student's likely continuations can depart from the teacher's, leaving long suffixes with weakly compatible supervision. OPD analysis relates learning outcomes to student–teacher compatibility and reports degradation of reward quality with trajectory depth (Li et al., 2026b). Yet the standard OPD recipe evaluated here generates and scores responses under a fixed maximum budget, without using this mismatch information to decide how much computation to spend.

This unused mismatch information offers a way to accelerate OPD. When compatible supervision is concentrated in a short prefix, generating and scoring a long suffix can add substantial cost without comparable learning benefit. Our experiments show that truncating such rollouts can reduce this cost while retaining performance. The difficulty is deciding where to truncate: the useful extent of supervision differs across pairs and changes as the student learns. A fixed cutoff must be chosen before these changes are observed, and a preset growth schedule requires anticipating them. Rather than asking the user to make either choice, we use compatibility measured during training to determine and adjust the cutoff.

PRUNE-OPD uses top-k overlap between teacher and student predictions to estimate how far a response offers compatible teacher supervision. Cumulative low-overlap events define a reliability score that reweights the current rewards and yields an effective training length. The first batch is generated at the maximum budget; its average effective length sets the training cutoff and the next rollout budget, subject to a common minimum. Later batches provide feedback on whether

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

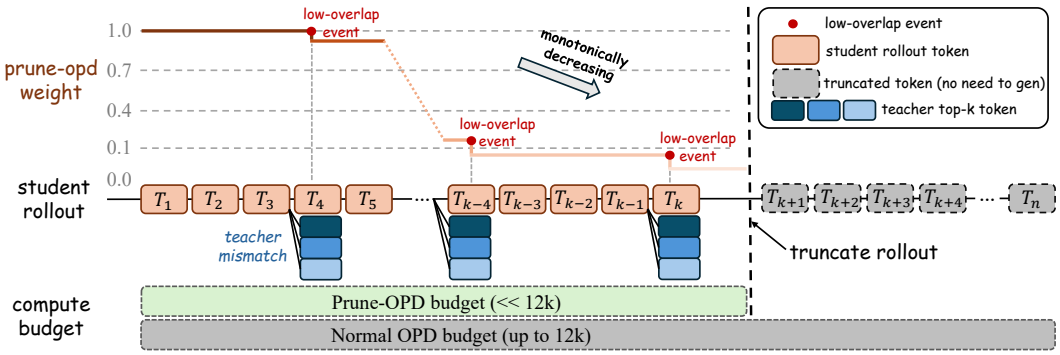

Figure 1: Conceptual overview of PRUNE-OPD. Cumulative low-overlap events reduce the reliability component and motivate a shorter rollout budget. Truncation is schematic: the first batch is truncated after generation; later budget changes apply to future rollouts, not immediate stopping at zero reliability (Algorithm 1). The actual loss weight additionally retains its base floor.

compatible supervision extends near the current cap. The controller expands the budget when it does and contracts it after repeated low-hit batches. Shorter budgets reduce generation and teacher-scoring costs; expansion allows training to retain later positions as overlap improves (Figure 1).

Across four pairs, PRUNE-OPD reduces training time by 37.6%–68.0% while broadly preserving accuracy. Effective lengths stay short in low-compatibility Qwen3 pairs, grow with compatibility in the JustRL pair, and remain long with the high-compatibility Skywork teacher.

Our contributions are:

- We identify student–teacher mismatch as a key source of efficiency gains in standard OPD: rollout computation does not adapt to the extent of compatible supervision. We show that using this information to truncate training rollouts can accelerate distillation while retaining performance.
- We introduce PRUNE-OPD, which couples overlap-guided reward reweighting with adaptive rollout truncation. Within common bounds, the algorithm determines and updates truncation lengths across pairs and training stages, without manually preset per-pair cutoffs or length schedules.
- We conduct extensive experiments across five student–teacher pairs and five reasoning benchmarks. Component and dynamic-budget ablations test reward weighting, truncation, and continued adaptation; parameter sweeps examine overlap-threshold and base-weight sensitivity. Across four pairs, training time falls by 37.6%–68.0% with broadly preserved performance. The observed lengths remain short for low-compatibility pairs, grow during training for a medium-compatibility pair, and stay long for a high-compatibility pair.

2 BACKGROUND AND PROBLEM SETUP

On-policy teacher feedback. For a prompt x, let $y = (y_1, \ldots, y_T)$ be sampled from the student policy π_θ. At a student-visited prefix $s_\tau = (x, y_{<\tau})$, define

$$p_\tau(v) = \pi_\theta(v \mid s_\tau), \qquad q_\tau(v) = \pi_{\mathrm{T}}(v \mid s_\tau), \quad v \in \mathcal{V}. \tag{1}$$

A common OPD formulation minimizes reverse KL on student-generated trajectories (Gu et al., 2024; Agarwal et al., 2024):

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{x,\, y\sim\pi_\theta(\cdot|x)} \left[\sum_{\tau=1}^{T} D_{\mathrm{KL}}(p_\tau \,\|\, q_\tau) \right]. \tag{2}$$

Equation (2) describes the full-distribution objective; our implementation uses the baseline's top-k token-level reward vector $r_{\tau,\cdot}$ on student candidate tokens. PRUNE-OPD scales this reward vector before the policy update; Appendix A.1 defines the implemented reward and loss.

Local compatibility. The teacher is conditioned on the student's prefix, which may differ from its preferred reasoning trajectory. Prior analysis uses high-probability token overlap to diagnose OPD

alignment (Li et al., 2026b). We use repeated low-overlap events to attenuate rewards and estimate how far compatibility persists within a response. This estimate supplies feedback for subsequent rollout budgets. It does not identify an optimal stopping point or establish correctness: different valid continuations can have low overlap.

3 PRUNE-OPD: COMPATIBILITY-GUIDED LENGTH ADAPTATION

PRUNE-OPD uses overlap to reweight rewards and adapt rollout budgets. Token positions and training steps are indexed by τ and t. The first rollout uses $M_1 = M_{\max}$; its average effective length, bounded below by $M_{\min} = 1024$, sets both the first batch's training cutoff and the next rollout budget.

3.1 FROM LOCAL COMPATIBILITY TO CUMULATIVE RELIABILITY

Let $\mathcal{K}_\tau^{\mathrm{S}} = \mathrm{TopK}(p_\tau, k)$ and $\mathcal{K}_\tau^{\mathrm{T}} = \mathrm{TopK}(q_\tau, k)$ be the student and teacher candidate sets at the same prefix. Their overlap ratio is

$$\mathcal{O}_\tau = \frac{|\mathcal{K}_\tau^{\mathrm{S}} \cap \mathcal{K}_\tau^{\mathrm{T}}|}{k}. \tag{3}$$

A low-overlap event is defined by a threshold γ:

$$B_\tau = \mathbf{1}[\mathcal{O}_\tau < \gamma], \qquad C_\tau = \sum_{i=1}^{\tau} B_i. \tag{4}$$

We turn the accumulated count into a reliability score,

$$R_\tau = \mathrm{clip}(1 - w_{\mathrm{drop}} C_\tau, 0, 1), \tag{5}$$

where $w_{\mathrm{drop}} \geq 0$ sets the decrement per event. This overlap-based reliability proxy reaches zero after 100 low-overlap events at the default $w_{\mathrm{drop}} = 0.01$; it does not estimate correctness probabilities.

The score is non-increasing along each valid response. Local agreement after an incompatible prefix does not necessarily indicate that useful teacher supervision has resumed, so the score does not recover when overlap rises again. This choice may also suppress supervision on trajectories that genuinely recover (Appendix A.5). We compare overlap with an action-level alternative that tests whether the sampled student token belongs to the teacher's visible top-p set (Appendix A.4).

3.2 REWEIGHTING THE CURRENT UPDATE

We retain a base floor in the actual loss weight:

$$L_\tau = R_\tau + w_{\mathrm{base}}, \tag{6}$$

for valid response tokens, and set $R_\tau = L_\tau = 0$ on padding. Every candidate reward at a position receives the same multiplier:

$$\widetilde{r}_{\tau,j} = L_\tau r_{\tau,j}. \tag{7}$$

The baseline advantage estimator uses the scaled reward directly. With $w_{\mathrm{base}} = 0.5$, valid-token weights range from 1.5 to 0.5. Positions with high reliability receive larger weights than in baseline OPD, while zero-reliability positions retain a nonzero reward weight.

3.3 INITIALIZING AND ADAPTING THE ROLLOUT BUDGET

Each response's *effective training length* counts positions with reliability above ϵ:

$$E = \sum_{\tau=1}^{T} \mathbf{1}[R_\tau > \epsilon], \tag{8}$$

where ϵ is a small tolerance. Effective training length E estimates the prefix worth retaining under the compatibility criterion. It differs from response length T (generated tokens)

Algorithm 1 PRUNE-OPD: first-step truncation, reward weighting, and budget control

Require: Student π_θ, teacher π_{T}, OPD reward routine, reliability parameters $(k, \gamma, w_{\mathrm{drop}}, w_{\mathrm{base}}, \epsilon)$, controller parameters $(m, \rho, \Delta, P, M_{\min}, M_{\max})$

Initialize $M_1 \leftarrow M_{\max}$ and consecutive low-hit count $c \leftarrow 0$.
for training step $t = 1, 2, \ldots$ **do**
  Sample prompts and generate student responses with maximum length M_t.
  Score collected prefixes with the teacher and obtain raw OPD rewards r.
  Compute R, scaled rewards $\widetilde{r}$, and reliable lengths E using Eqs. (3)–(8).
  Compute h_t using Eq. (9).
  if $t = 1$ **then**
    $M_{t+1} \leftarrow \max(M_{\min}, \mathrm{round}(B^{-1} \sum_{b=1}^{B} E^{(b)}))$.
    Truncate collected responses and aligned rewards $\widetilde{r}$ at M_{t+1}.
  else if $h_t \geq \rho$ **then**
    $M_{t+1} \leftarrow \min(M_t + \Delta, M_{\max})$; $c \leftarrow 0$.
  else
    $c \leftarrow c + 1$; $M_{t+1} \leftarrow M_t$.
    if $c \geq P$ **then**
      $M_{t+1} \leftarrow \max(M_t - \Delta, M_{\min})$; $c \leftarrow 0$.
    end if
  end if
  Update the student with the baseline OPD objective using $\widetilde{r}$.
end for

and rollout budget M_t (the truncation cap). For the first batch of B responses, set $M_2 = \max(M_{\min}, \mathrm{round}(B^{-1} \sum_b E^{(b)})) \leq M_{\max}$ and truncate its responses and aligned rewards at M_2 before updating the student. Later steps use the hit ratio

$$h_t = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}[E^{(b)} \geq M_t - m], \tag{9}$$

where m is a margin. If $h_t \geq \rho$, enough responses remain reliable near the current cap to justify expanding the next budget by Δ. If $h_t < \rho$ for P consecutive steps, the budget contracts by Δ. Otherwise it stays unchanged; all changes respect $[M_{\min}, M_{\max}]$.

The first batch supplies the initial cutoff without a separate length search; feedback then adjusts it within common bounds. Early EOS can also lower the hit ratio: contraction need not indicate worsening compatibility (Appendix A.5). Responses may exceed E: zero-reliability positions retain the base weight, and the shared budget need not equal an individual effective length. Appendix A.1 gives implementation details.

4 EXPERIMENTS

We evaluate whether compatibility-guided length adaptation reduces training time while preserving performance, and whether it can also improve accuracy in low-compatibility pairs. The five student–teacher pairs include low-, medium-, and high-compatibility cases.

4.1 EXPERIMENTAL SETUP

Training and model pairs
We train on DAPO-Math-17K (Yu et al., 2025). DeepSeek-R1-Distill-Qwen-1.5B learns from JustRL-DeepSeek-1.5B or DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025; He et al., 2025). Qwen3-1.7B-Base and Qwen3-4B-Base learn from Qwen3-4B (Non-thinking) (Yang et al., 2025a), forming two low-compatibility pairs. The fifth pair, DeepSeek-R1-Distill-Qwen-7B / Skywork-OR1-7B, has high compatibility. The 1.5B student / JustRL teacher pair has medium compatibility that improves during training. These categories describe observed overlap, not inputs to the controller. Tables use student / teacher order.

Table 1: Main results for PRUNE-OPD. Accuracy is reported in percent; time reduction is relative to OPD within each pair. Bold marks the best value in each column within a pair, including ties (lower is better for time). Red. denotes the reduction of training time.

Method	AMC23	AIME24	AIME25	HMMT24	HMMT25	Time	Red.
DeepSeek-R1-Distill-Qwen-1.5B / JustRL-DeepSeek-1.5B							
OPD	77.5	43.5	32.7	**23.5**	18.1	6.9h	0%
OPD (fixed 4K)	77.8	43.1	31.0	22.7	17.9	4.8h	30.4%
PRUNE-OPD (top-p)	77.3	44.0	30.8	23.3	**19.8**	4.3h	37.7%
PRUNE-OPD (overlap)	**78.4**	**45.2**	**33.5**	22.7	18.8	**4.1h**	**40.6%**
DeepSeek-R1-Distill-Qwen-1.5B / DeepSeek-R1-Distill-Qwen-7B							
OPD	**66.4**	**33.5**	25.4	14.4	**15.0**	12.5h	0%
OPD (fixed 4K)	66.3	31.7	**25.8**	14.2	14.4	6.6h	47.2%
PRUNE-OPD (top-p)	65.5	29.6	24.2	**14.6**	14.0	4.8h	61.6%
PRUNE-OPD (overlap)	65.9	33.3	**25.8**	**14.6**	**15.0**	**4.0h**	**68.0%**
Qwen3-1.7B-Base / Qwen3-4B (Non-thinking)							
OPD	28.0	**7.3**	2.9	2.1	0.6	16.5h	0%
OPD (fixed 4K)	28.6	6.3	2.9	2.3	**1.0**	12.9h	21.8%
PRUNE-OPD (top-p)	28.4	5.6	2.9	1.9	0.6	11.5h	30.3%
PRUNE-OPD (overlap)	**29.1**	**7.3**	**4.2**	**2.9**	**1.0**	**10.3h**	**37.6%**
Qwen3-4B-Base / Qwen3-4B (Non-thinking)							
OPD	**42.7**	14.8	10.6	5.4	2.9	28.5h	0%
OPD (fixed 4K)	41.9	15.2	13.5	4.8	2.7	22.3h	21.8%
PRUNE-OPD (top-p)	40.5	13.1	12.7	4.8	4.0	20.0h	29.8%
PRUNE-OPD (overlap)	**42.7**	**16.3**	**14.6**	**6.7**	**5.6**	**13.5h**	**52.6%**

Baselines and settings
We compare OPD, fixed-4K OPD, and PRUNE-OPD with overlap or top-p acceptance. Defaults are $k = 16$, $\gamma = 0.7$, $w_{\text{drop}} = 0.01$, $w_{\text{base}} = 0.5$, and $p = 0.95$. Full PRUNE-OPD uses Algorithm 1 with $(M_{\min}, M_{\max}) = (1024, 12288)$, $m = \Delta = 100$, $\rho = 0.1$, and $P = 3$. All methods train for 200 steps and retain the OPD reward routine; further settings are in Appendix A.2.

Evaluation and comparison scope
We evaluate the final checkpoint after 200 steps at temperature 1.0. Avg@16 averages correctness over 16 responses per problem, then over problems; Mean averages the five benchmarks equally. Training times cover the full run, excluding evaluation, with matched hardware within pairs. Fixed-4K is not compute-matched or tuned per pair; parameter search is excluded. Training truncation does not restrict evaluation: the response cap is 31,744 tokens (Appendices A.2 and A.6).

4.2 ACCURACY AND TRAINING TIME

Table 1 shows training-time reductions of 40.6% and 68.0% for the JustRL and larger DeepSeek teachers, and 37.6% and 52.6% for the two low-compatibility Qwen3 pairs. Accuracy changes relative to OPD range from -0.8 to $+1.7$ points for the two DeepSeek-1.5B student pairs. Both low-compatibility Qwen3 pairs match or exceed OPD on all five benchmarks while training faster. Figure 2 shows comparable accuracy reached earlier in many panels; Figure 5 compares progress per training step with JustRL.

4.3 THE HIGH-COMPATIBILITY PAIR

For DeepSeek-R1-Distill-Qwen-7B / Skywork-OR1-7B, the plotted aggregate overlap stays above the threshold, while individual positions may still have low overlap. The rollout budget expands to 12,288 tokens (Figure 3).

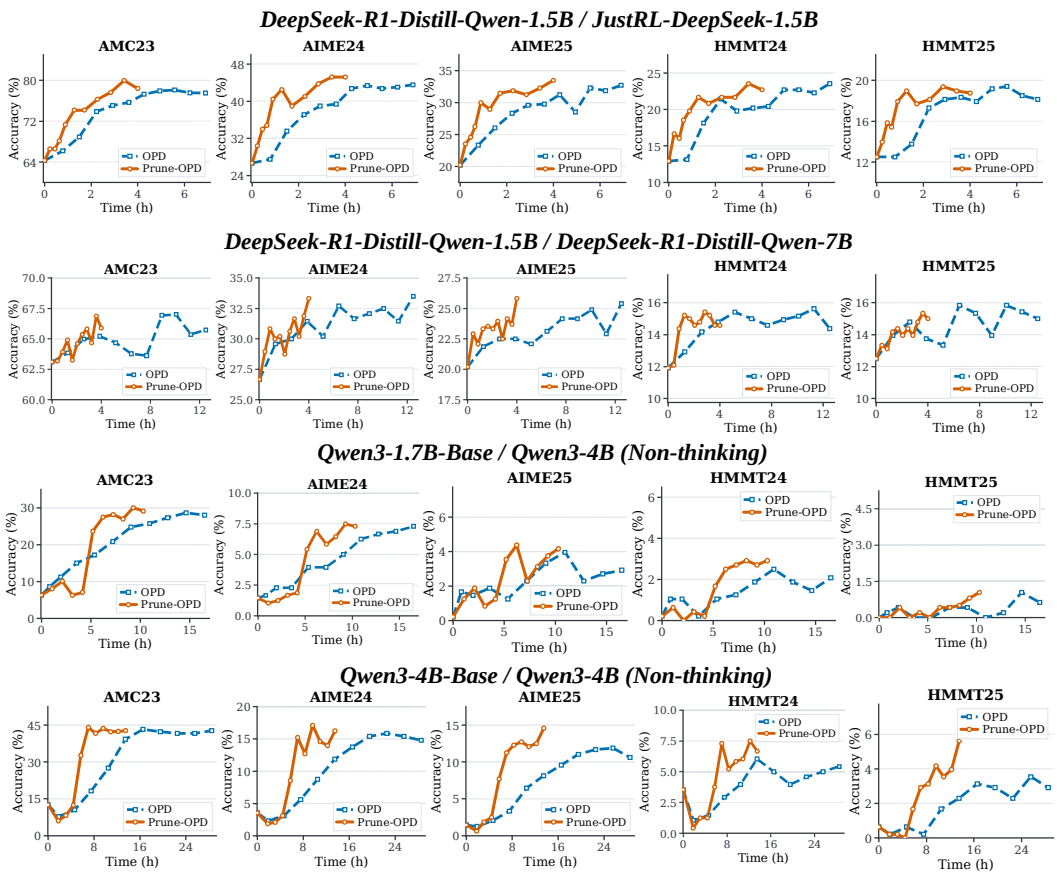

Figure 2: Accuracy versus training time for four student-teacher pairs: two DeepSeek pairs and two Qwen3 pairs. Rows correspond to pairs and columns to benchmarks; each panel compares OPD with PRUNE-OPD.

Table 2: High-compatibility behavior for DeepSeek-R1-Distill-Qwen-7B / Skywork-OR1-7B. Bold marks column-wise best values, with lower being better for time.

Method	AMC23	AIME24	AIME25	HMMT24	HMMT25	Time	Red.
DeepSeek-R1-Distill-Qwen-7B / Skywork-OR1-7B							
OPD	88.3	**67.1**	**52.5**	**33.5**	**32.1**	17.5h	0%
OPD (fixed 4K)	87.2	66.0	48.8	31.0	30.4	**11.3h**	**35.4%**
PRUNE-OPD (overlap)	**88.9**	66.7	52.1	33.1	31.5	17.0h	2.9%

For Skywork, PRUNE-OPD takes 17.0h versus OPD's 17.5h, with accuracy differences of -0.6 to $+0.6$ points. Fixed-4K OPD is faster at 11.3h but loses accuracy on every benchmark, including 3.7 points on AIME25 (Table 2). Here, the faster fixed-4K run sacrifices accuracy, whereas PRUNE-OPD retains long supervision and stays closer to OPD. Its small time reduction is consistent with the high effective lengths in Figure 3: little of the rollout is removed. This differs from the Qwen3 pairs, where the same controller keeps budgets short. No separate cutoff is supplied for Skywork.

4.4 ADAPTATION OF EFFECTIVE TRAINING LENGTH

Growing lengths with JustRL and sustained long lengths with Skywork. For the medium-compatibility JustRL pair, the budget stays near its minimum after initialization, then grows as overlap improves and reliability extends to later positions (Figure 4). With the high-compatibility

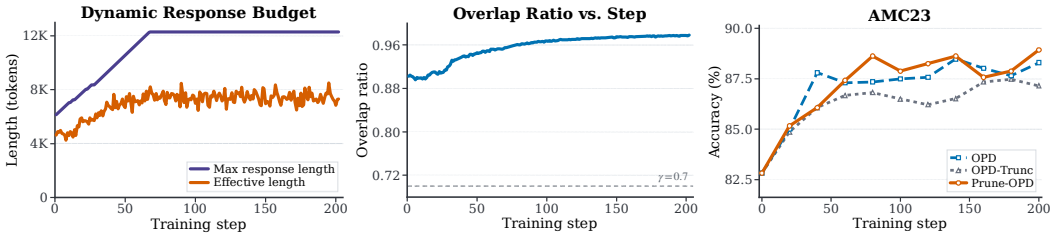

Figure 3: High-compatibility training dynamics for DeepSeek-R1-Distill-Qwen-7B / Skywork-OR1-7B. Left: effective training length and Prune-OPD rollout budget versus training step. Middle: overlap ratio versus training step. Right: AMC23 accuracy over training, comparing OPD, OPD (fixed 4K), and PRUNE-OPD.

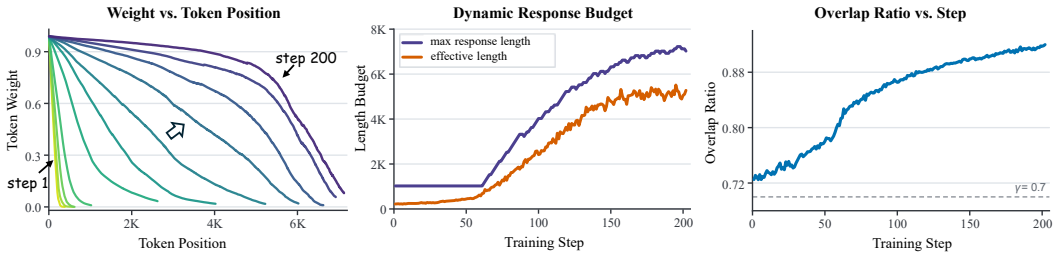

Figure 4: Training dynamics for DeepSeek-R1-Distill-Qwen-1.5B / JustRL-DeepSeek-1.5B. Left: mean loss weight minus 0.5, at step 1 and steps 20–200 in increments of 20 (yellow to purple). Each position averages valid responses reaching it; means at the base floor are omitted before subtraction. Middle: effective training length and rollout budget. Right: overlap ratio.

Skywork teacher, effective length grows early and remains high, while the budget reaches its cap (Figure 3). Appendix A.9 reports overlap at different trajectory depths.

Accuracy at matched training steps. Figure 5 shows similar accuracy trajectories for OPD and PRUNE-OPD on the JustRL pair, with checkpoint-dependent differences. Training time falls from 6.9h to 4.1h while learning progress per update remains comparable.

Persistently short effective training lengths in the low-compatibility Qwen3 pairs. Both low-compatibility Qwen3 pairs retain effective lengths of a few hundred tokens (Figure 6) while matching or improving OPD accuracy. Actual responses can exceed E: the budget is at least 1,024 and zero-reliability positions retain the base weight. Appendix A.8 discusses a possible explanation.

4.5 WHY CONTINUE ADAPTING THE ROLLOUT BUDGET?

On JustRL, the first-batch average effective length is below 1,024, so the initial cutoff is 1K. Fixed-1K retains the same initialization and reward weighting as full PRUNE-OPD, disabling only subsequent budget updates. It isolates continued adaptation; fixed-4K OPD instead tests a preset cutoff with baseline rewards (Table 3). Neither cutoff is tuned per pair. Before OPD begins, selecting a cutoff requires anticipating supervision needs that change during training; PRUNE-OPD selects and updates it from observed compatibility.

Fixed-1K takes 3.2h but gives 36.1% Mean, 3.0 points below OPD. Fixed-4K recovers Mean to 38.5% at 4.8h. Adaptation reaches 39.7% in 4.1h: it recovers 3.6 points over fixed-1K and exceeds fixed-4K by 1.2 points in less time. Neither tested cutoff matches this result. Compatible supervision extends to later positions as training proceeds (Figure 4), making the initial 1K budget restrictive. The practical benefit is obtaining suitable lengths from training feedback, without prescribing when or how far to extend rollouts (Appendix A.7.2).

4.6 ROLES OF REWEIGHTING AND ADAPTIVE TRUNCATION

We separate reward reweighting from overlap-guided budget control (Table 4). Weighting only retains the OPD rollout budget; truncation only still computes overlap and effective lengths but leaves rewards unweighted.

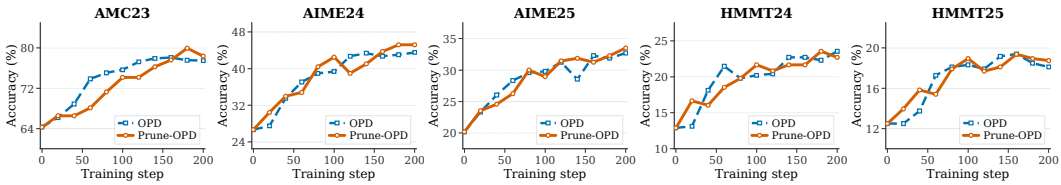

Figure 5: Training-step accuracy dynamics for DeepSeek-R1-Distill-Qwen-1.5B distilled from JustRL-DeepSeek-1.5B. The five panels report benchmark accuracy over training steps on AMC23, AIME24, AIME25, HMMT24, and HMMT25, comparing OPD and PRUNE-OPD.

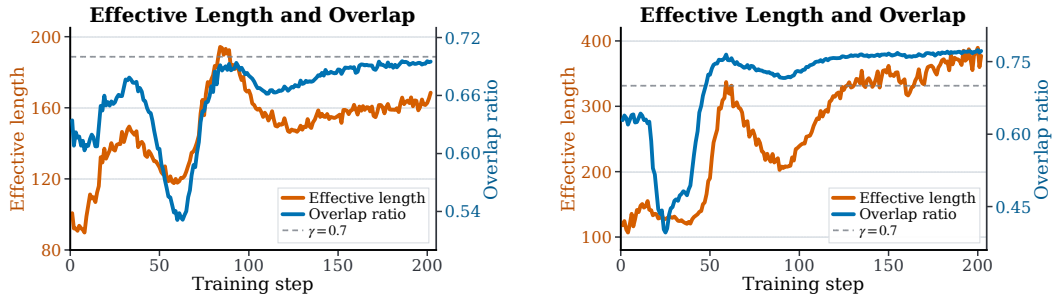

Figure 6: Persistently short effective training lengths in the two low-compatibility Qwen3 student–teacher pairs, as characterized by top-k overlap. For Qwen3-1.7B-Base / Qwen3-4B (Non-thinking) and Qwen3-4B-Base / Qwen3-4B (Non-thinking), the effective training lengths span a few hundred tokens, whereas the baseline response cap is 12,288. These lengths count positions above the reliability threshold, not all generated tokens. Average overlap above γ can coexist with short effective lengths: cumulative low-overlap events, not average overlap, determine reliability.

Table 3: Continued budget adaptation on the JustRL pair. Fixed-1K retains reward weighting but disables subsequent budget updates. Mean averages five benchmark accuracies (%). Bold marks best scores or lowest time.

Method	AMC23	AIME24	AIME25	HMMT24	HMMT25	Mean	Time
OPD	77.5	43.5	32.7	**23.5**	18.1	39.1	6.9h
PRUNE-OPD (fixed 1K)	75.0	37.9	29.6	19.8	18.1	36.1	**3.2h**
OPD (fixed 4K)	77.8	43.1	31.0	22.7	17.9	38.5	4.8h
PRUNE-OPD	**78.4**	**45.2**	**33.5**	22.7	**18.8**	**39.7**	4.1h

Truncation provides the main speedup, reducing time by 51.6% on Qwen3 and 43.5% on JustRL; weighting alone does not accelerate either pair. On low-compatibility Qwen3, weighting alone improves Mean accuracy by 0.4 points, truncation alone by 0.9, and the full method by 1.9 over OPD. Adding weighting to truncation therefore gives a further 1.0-point gain while retaining the speedup. The difference is smaller on JustRL: the full method has the highest observed Mean at 39.7%, but exceeds truncation alone by only 0.2 points and takes 0.2h longer.

Three repeated runs per method on low-compatibility Qwen3-4B average 17.0% Mean for PRUNE-OPD versus 15.6% for OPD, with 52.5% less training time. Every PRUNE-OPD run exceeds every OPD run in Mean (Appendix A.7.3). The accuracy gain therefore extends beyond the main-table run. The pattern is not confined to the five-benchmark average: on AIME24, AIME25, HMMT24, and HMMT25, every PRUNE-OPD run exceeds all OPD runs. AMC23 does not show the same advantage. These repeated experiments support average-accuracy gains on this low-compatibility pair while retaining the training speedup, rather than uniform improvement on every benchmark. The appendix reports all six rows and distinguishes run 1 from the additional runs.

Table 4: Component ablations (student / teacher). Left: Qwen3-4B-Base / Qwen3-4B (Non-thinking), a low-compatibility pair. Right: DeepSeek-R1-Distill-Qwen-1.5B / JustRL-DeepSeek-1.5B. Mean: accuracy (%); Red.: time reduction relative to OPD (negative means slower).

	Qwen3-4B pair			*JustRL-1.5B pair*		
Method	**Mean**	**Time**	**Red. (%)**	**Mean**	**Time**	**Red. (%)**
OPD	15.3	28.5h	0.0	39.1	6.9h	0.0
Weighting only	15.7	29.3h	-2.8	38.7	7.0h	-1.4
Truncation only	16.2	13.8h	51.6	39.5	**3.9h**	**43.5**
PRUNE-OPD	**17.2**	**13.5h**	**52.6**	**39.7**	4.1h	40.6

4.7 COMPATIBILITY METRIC AND PARAMETER SENSITIVITY

Overlap versus action acceptance. Overlap is faster than top-p acceptance on all four pairs and matches or exceeds its accuracy in 18 of 20 comparisons (Table 1). On low-compatibility Qwen3-4B, all five scores improve and time falls from 20.0h to 13.5h. Overlap compares candidate sets rather than a single action; these results favor it under the evaluated thresholds and top-16 compatibility measurements.

Overlap threshold. On JustRL, increasing γ from 0.6 to 0.9 reduces time from 4.8h to 2.4h, but both 0.8 and 0.9 score below the default on all five benchmarks (Appendix A.10). Figure 8 shows earlier weight decay and shorter floor-corrected maximum response lengths. At 0.9, Mean is 38.2%, below OPD's 39.1%; the default reaches 39.7% with a 40.6% time reduction. All five pairs share $\gamma = 0.7$ and controller settings, not preset lengths. The 0.6 setting leads on AMC23 and AIME24, whereas 0.7 leads on AIME25 and HMMT24; neither threshold is best on every benchmark.

Base reward weight. On JustRL, $w_{\text{base}} = 0$, 0.5, and 1.0 yield Mean accuracies of 38.9%, 39.7%, and 39.6% (Appendix A.7.4). Zero base weight retains accuracy near OPD's 39.1% without reward amplification; the default has the highest Mean. Changing the floor also changes low-reliability supervision, not just reward scale.

5 RELATED WORK

On-policy and self-distillation. MiniLLM and GKD study distillation on student-generated trajectories (Gu et al., 2024; Agarwal et al., 2024); subsequent work analyzes OPD objectives, dynamics, and failure modes (Yang et al., 2026b; Li et al., 2026b; Fu et al., 2026). Self-distillation uses privileged information or feedback to construct an on-policy teacher signal (Hubotter et al., 2026; Zhao et al., 2026b; Shenfeld et al., 2026). PRUNE-OPD retains token-level OPD feedback and uses compatibility to adjust its weights and the length of subsequent rollouts.

Teacher suitability and long reasoning. Capacity-gap studies and distillation scaling laws caution that a stronger teacher need not always be easier for a student to learn from (Cho & Hariharan, 2019; Mirzadeh et al., 2020; Busbridge et al., 2025; Li et al., 2025). OPD mechanism analysis further links outcomes to local compatibility and trajectory depth (Li et al., 2026b). We exploit this mismatch to accelerate OPD: overlap guides reward reweighting and adapts rollout budgets across pairs and training stages. Appendix A.12 provides additional context.

6 CONCLUSION

PRUNE-OPD uses student–teacher mismatch to guide reward reweighting and adaptive truncation without manually preset per-pair cutoffs. It reduces training time by 37.6%–68.0% across four pairs with broadly preserved performance. Ablations support continued adaptation and identify truncation as the main source of speedup, with lengths adjusted from compatibility feedback during training. High-compatibility Skywork retains long supervision, demonstrating adaptation rather than uniform shortening.

AI USE STATEMENT

Generative AI was used in two roles. As part of the research process, it assisted in writing code, but all code underwent human review. In addition, a generative AI assistant was used to improve the language, organization, and formatting of the manuscript. Generative AI was not used to propose research ideas or formulate hypotheses; all research ideas, methodological decisions, and experimental designs were developed by the authors. All AI-assisted manuscript editing was reviewed and approved by the authors, who verified the reported results and take full responsibility for the content of the paper and all AI-assisted outputs.

ETHICS STATEMENT

This work studies the training efficiency of on-policy distillation through model training and mathematical benchmark evaluation, rather than human-subject research. Improving distillation efficiency does not by itself ensure that the resulting models are correct or safe. Downstream applications should therefore include appropriate evaluation and safeguards.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 describe reward weighting, first-batch initialization, and subsequent budget updates. Appendices A.1–A.3 specify the implemented loss, default hyperparameters, and prompt template; Section 4.1 and Appendix A.6 describe training and evaluation settings, hardware, and timing conventions.

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*, 2024.

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 Russell Webb. Distillation scaling laws. In *International Conference on Machine Learning*, pp. 5977–6045. PMLR, 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, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

Ken Ding. Hdpo: Hybrid distillation policy optimization via privileged self-distillation, 2026.

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

Naibin Gu, Chenxu Yang, Qingyi Si, Chuanyu Qin, Dingyu Yao, Peng Fu, Zheng Lin, Weiping Wang, Nan Duan, and Jiaqi Wang. Co-evolving policy distillation, 2026. URL https://arxiv.org/abs/2604.27083.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The twelfth international conference on learning representations*, 2024.

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 1.5b llm with a simple rl recipe, 2025.

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?, 2026.

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

Jonas Hubotter, Frederike Lubeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation, 2026.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation, 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, 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?, 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, 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, 2026a.

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*, 2026b.

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

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation, 2026.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter, 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, Y. K. 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 Hubotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026.

Yifan Song, Xingjian Tao, Zhicheng Yang, Yihong Luo, and Jing Tang. Ehrag: Bridging semantic gaps in lightweight graphrag via hybrid hypergraph construction and retrieval. *arXiv preprint arXiv:2604.17458*, 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, 2026.

Minrui Xu, Zilin Wang, Mengyi DENG, Zhiwei Li, Zhicheng Yang, Xiao Zhu, Yinhong Liu, Boyu Zhu, Baiyu Huang, Chao Chen, Heyuan Deng, Fei Mi, Lifeng Shang, Xingshan Zeng, and Zhijiang Guo. Envfactory: Scaling tool-use agents via executable environments synthesis and robust rl, 2026. URL https://arxiv.org/abs/2605.18703.

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, 2025a.

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

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

Zhicheng Yang, Yiwei Wang, Yinya Huang, Zhijiang Guo, Wei Shi, Xiongwei Han, Liang Feng, Linqi Song, Xiaodan Liang, and Jing Tang. Optibench meets resocratic: Measure and improve LLMs for optimization modeling. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=fsDZwS49uY.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Yongxin Wang, Wenlei Shi, Yiwei Wang, Xiaodan Liang, and Jing Tang. Accordion-thinking: Self-regulated step summaries for efficient and readable llm reasoning. In *Forty-Third International Conference on Machine Learning*, 2026c. URL https://arxiv.org/abs/2602.03249.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Yongxin Wang, Dongchun Xie, Hanhui Li, Yiwei Wang, Xiaodan Liang, and Jing Tang. Depth-breadth synergy in rlvr: Unlocking llm reasoning gains with adaptive exploration. In *Forty-Third International Conference on Machine Learning*, 2026d. URL https://arxiv.org/abs/2508.13755.

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

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

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

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

A Supplementary Details and Analyses

A.1 Implementation Details

PRUNE-OPD combines post-processing of the OPD reward tensor with feedback-based initialization and updates of the rollout budget. The baseline reward tensor has shape $[B, T, k]$, where B is batch size, T is response length, and k is the top-k token dimension. For each valid response position, the method computes the scalar loss weight L_τ and multiplies every candidate reward at that position by the same scalar:

$$\widetilde{r}_{b,\tau,j} = L_{b,\tau} r_{b,\tau,j}. \tag{10}$$

Padding positions have $R_\tau = 0$ and $L_\tau = 0$. The supplied reference implementation uses `positive_eps`=10^{-8} for the test $R_\tau > \epsilon$ in Eq. (8); this is a numerical tolerance, distinct from the overlap threshold γ.

Teacher scoring and candidate access. In the reference implementation, a teacher forward pass on each scoring microbatch produces vocabulary logits at the student-generated prefixes. The same logits supply both the teacher's own top-k candidates and its log probabilities at all student top-k IDs, including IDs outside the teacher top-k. For the latter, the implementation gathers the requested logits and subtracts the full-vocabulary log-sum-exp; teacher top-k probabilities use the same normalizer. These outputs share a forward pass, rather than requiring a separate teacher call for each candidate set. Top-k here limits the candidate tensors used by the reward and compatibility routines, not the teacher's vocabulary projection. An interface exposing only the teacher's top-k probabilities would therefore be insufficient for the implemented OPD reward. Teacher scoring and candidate extraction are included in the full-run timing described in Appendix A.6; we do not separately quantify their overhead.

Candidate rewards and advantages. Let $\bar{p}_{b,\tau}$ denote the student distribution used to compute the collected batch's rewards, and let $v_{b,\tau,j}$ be its jth top-k candidate ($k = 16$). The `only_stu`/`student_p` configuration normalizes student probabilities within this candidate set, not over the teacher's top-k set:

$$a_{b,\tau,j} = \frac{\bar{p}_{b,\tau}(v_{b,\tau,j})}{\sum_{\ell=1}^{k} \bar{p}_{b,\tau}(v_{b,\tau,\ell})}, \qquad r_{b,\tau,j} = a_{b,\tau,j}\big[\log q_{b,\tau}(v_{b,\tau,j}) - \log \bar{p}_{b,\tau}(v_{b,\tau,j})\big]. \tag{11}$$

The teacher probabilities in the log-ratio are not renormalized over the candidate set. With response mask $m_{b,\tau}$, `token_reward_direct` sets

$$A_{b,\tau,j} = \mathrm{stopgrad}\big(m_{b,\tau} L_{b,\tau} r_{b,\tau,j}\big). \tag{12}$$

There is no return accumulation or group-wise advantage normalization in this estimator. Baseline OPD uses $L_{b,\tau} = 1$ on valid positions. Gradients do not pass through the collected rewards, candidate weights, or reliability scores.

Actor loss. The actor recomputes log probabilities of the same candidate IDs rather than selecting new candidates during the update. For an index $i = (b, \tau, j)$, define

$$u_i = \exp(\mathrm{clip}(\log p_{\theta,i} - \log p_{\mathrm{ref},i}, -20, 20)), \quad g_i = \max(-A_i u_i, -A_i \mathrm{clip}(u_i, 0.8, 1.2)), \tag{13}$$

where p_{ref} is a detached student reference for the importance ratio, not the teacher. The default vanilla policy-loss path applies dual clipping:

$$\ell_i = \begin{cases} \min(g_i, -3A_i), & A_i < 0, \\ g_i, & A_i \geq 0. \end{cases} \qquad \mathcal{L}_{\mathcal{U}} = \frac{\sum_{(b,\tau)\in\mathcal{U}} m_{b,\tau} \sum_{j=1}^{k} \ell_{b,\tau,j}}{\sum_{(b,\tau)\in\mathcal{U}} m_{b,\tau}}, \tag{14}$$

for a microbatch $\mathcal{U}$. Thus the candidate dimension is summed, and `token-mean` averages over valid response positions within each microbatch. Dynamic microbatch gradients are weighted by their response count divided by the local minibatch response count, accumulated, and averaged across data-parallel workers. This follows the implementation rather than assuming a single global token mean across differently sized microbatches.

The default configuration uses one actor epoch and one minibatch per rollout batch. In this on-policy branch, $p_{\mathrm{ref}} = \mathrm{stopgrad}(p_\theta)$, so the forward ratio is one while its numerator remains differentiable;

the local gradient is that of $-\sum_j A_j \log p_{\theta,j}$. The general multi-minibatch branch instead uses stored student log probabilities. Entropy regularization and the separate reference-KL loss are disabled. Equation (2) gives the conceptual full-distribution objective; Eqs. (11)–(14) specify the top-k reward-based implementation.

The reference implementation keeps the baseline isolated. When `prune_opd.enable=False`, its baseline reward and budget-control paths are unchanged. When enabled, the main experiments use `overlap_ratio`; `teacher_top_p_accept` is retained as an action-level variant. The teacher top-k size is controlled by `actor_rollout_ref.rollout.log_prob_top_k` and is set to 16 in the current experiments.

A.2 Primary Configurations

Table 5 summarizes the primary hyperparameters used across our PRUNE-OPD experiments. All benchmark performance is reported as Avg@16 accuracy, averaging correctness over 16 sampled responses per problem and then over benchmark problems. The configuration field named `init` specifies the minimum budget of 1,024 tokens, not the first rollout budget. At the first training step, every pair generates responses with the maximum budget and computes their average effective training length. The average is rounded to the nearest integer, with half-integer ties rounded upward: $\mathrm{round}(x) = \lfloor x + 0.5 \rfloor$ for $x \geq 0$. The larger of this rounded value and 1,024 sets both the first-batch training cutoff and the next rollout budget. Later budgets follow Algorithm 1. No separate search over candidate starting lengths is performed.

Table 5: **Default hyperparameters for PRUNE-OPD.**

Item	Value
Training dataset	DAPO-Math-17K
Training temperature	1.0
Teacher temperature	1.0
Evaluation temperature	1.0
Evaluation checkpoint	Final (step 200)
Rollout number	4
Prompt batch size (global)	64
Response batch size (global)	256
Actor epochs per rollout batch	1
LogProb top-K	16
Top-K strategy	Student Top-K
Reward weight mode	Student probability
Max prompt length	1,024
Validation max response length	31,744
Learning rate	1×10^{-6}
Training steps	200
KL coefficient	0.0
Loss aggregation	`token-mean`
Overlap threshold γ	0.7
Top-p threshold p	0.95
w_{drop}	0.01
w_{base}	0.5
Dynamic length min/max	1,024 / 12,288
Dynamic length step	100
Hit ratio threshold	0.1
Length margin	100
Shrink patience	3

A.3 Prompt Template

We use the same DAPO-Math prompt template as the OPD dynamics study (Li et al., 2026b). For each math problem, the raw question is inserted into the following instruction.

DAPO-Math Prompt Template

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

Input serialization in the reference implementation. The student checkpoint supplies the tokenizer and chat template. Prompts are serialized with `add_generation_prompt=True`, then tokenized with `add_special_tokens=False`. Both Qwen3 launch scripts pass `enable_thinking=False` to this template. The shared OPD launcher sets `reward_model.model.input_tokenizer=null`, disabling teacher-side chat-template conversion: the teacher scores the student's input IDs and response tokens directly, rather than a separately rendered conversation. Candidate overlap compares token IDs, so this path requires matching student–teacher token-ID semantics; it does not implement cross-tokenizer alignment.

A.4 Top-p Action-Acceptance Metric

In addition to overlap ratio, we evaluate an alternative action-level compatibility metric. At position τ, the student samples the actual next token y_τ, while the teacher returns top-k tokens and probabilities under the same prefix:

$$\mathcal{K}_\tau = \{v_{\tau,1}, \ldots, v_{\tau,k}\}, \qquad q_{\tau,1} \geq q_{\tau,2} \geq \cdots \geq q_{\tau,k}. \tag{15}$$

Given a top-p threshold p, define the cumulative visible mass

$$S_{\tau,m} = \sum_{j=1}^{m} q_{\tau,j}. \tag{16}$$

If a smallest m_τ satisfies $S_{\tau,m_\tau} \geq p$, the visible nucleus is

$$\mathcal{A}_\tau(p,k) = \{v_{\tau,1}, \ldots, v_{\tau,m_\tau}\}. \tag{17}$$

If the returned top-k mass is still below p, the true nucleus extends beyond the observed set, so we conservatively use $\mathcal{A}_\tau(p,k) = \mathcal{K}_\tau$. The bad event is $B_\tau = \mathbf{1}[y_\tau \notin \mathcal{A}_\tau(p,k)]$. In the reported top-$p$ runs, $p = 0.95$. The comparison uses these configured thresholds and the same teacher top-16 candidate information for compatibility measurement; it does not match the two metrics by low-compatibility event frequency. The results compare the evaluated configurations, not fully observed nucleus acceptance or all possible metric calibrations.

A.5 Reliability Rationale

Prior OPD analysis distinguishes rewards that correlate with final correctness from rewards that provide useful local updates on student prefixes (Li et al., 2026b). We use repeated low-overlap events as a heuristic for reducing later reward weights. The count retains the history of incompatibility even if overlap subsequently increases. This is conservative: local agreement can resume on a drifted prefix, but a genuinely recovered trajectory also receives no increase in reliability. The base weight w_{base} preserves a residual OPD signal in either case. Reliability denotes this overlap-based heuristic, not a calibrated probability of correctness or a direct measurement of gradient quality.

Position-wise weight diagnostics. The reference exporter computes each position's mean weight as $\sum_b m_{b,\tau} L_{b,\tau} / (\sum_b m_{b,\tau} + 10^{-6})$. Padding therefore contributes neither weight nor count. Figure 4 subtracts the base floor from these exported means, whereas Figure 8 displays them directly. The contributing responses can differ across positions; these curves summarize responses reaching each position, not a fixed cohort followed to the longest length.

Response termination and budget feedback. Effective length is bounded by generated response length: $E \leq T \leq M_t$. If a response ends with EOS before $M_t - m$, it does not count as a budget hit even if every generated position remains compatible. Thus, h_t measures how often compatible supervision reaches near the cap, jointly reflecting overlap and response termination; a low value is not by itself evidence of increased mismatch. The controller can contract when responses finish early, as well as when reliability decays before the cap. Budget feedback therefore tracks the observed extent of compatible supervision, rather than compatibility alone.

A.6 Compute Reporting

All reported training runs use NVIDIA H200 GPUs. Training time covers the run from start to finish, excluding benchmark evaluation time; it is not a steady-state per-step estimate. It includes the first full-budget rollout and teacher scoring, compatibility computation, subsequent rollouts, and student updates. Comparisons within each pair use matched hardware and the same timing convention. These are per-run costs, not the cost of external length or parameter searches.

Training stack and batching. The reference launch configuration uses one node with eight GPUs, the bundled `verl` implementation (version marker `0.7.0.dev`), FSDP for the actor, and vLLM for rollout generation. Tensor and sequence parallel sizes default to one. The global prompt batch is 64, with four responses per prompt, giving 256 responses per training step. The configured actor minibatch of 64 prompt groups is expanded by the rollout count before distribution across workers: with eight data-parallel workers, each processes 32 responses per optimizer step. Dynamic token-budgeted microbatching partitions this local batch and accumulates gradients before one optimizer update; it does not multiply the global batch by the GPU count.

The shared reference launch scripts enable `use_dynamic_bsz` and set the actor token budget per GPU to $\max(32768, \text{max prompt length} + \text{max response length})$, giving 32,768 under the lengths in Table 5. Sequence parallelism is one, so no further factor is applied. The partitioner balances valid prompt-plus-response tokens across microbatches. With dynamic batching enabled, the configured fixed microbatch size of one does not determine the partition. These are reference-launch settings; changing the partition can change the effective gradient weighting under the microbatch-local reduction in Appendix A.1.

Evaluation. Reported scores use the final checkpoint after 200 training steps, not a checkpoint selected for its validation score. Sampling temperature is 1.0, with 16 responses per problem and a maximum response length of 31,744. The reference evaluation configuration uses top-$p = 0.95$. Training-time savings do not come from shortening these evaluation responses. The evaluation sets contain 40 AMC23 problems and 30 problems each for AIME24, AIME25, HMMT24, and HMMT25. Avg@16 averages sampled answers, not independent training runs; the reported score differences are descriptive comparisons, not statistical significance tests.

A.7 Additional Experimental Results

All benchmark scores below are percentages under the Avg@16 protocol in Section 4.1. Mean denotes the equal-weight arithmetic average of the displayed benchmark scores, not an average weighted by benchmark size. Mean values and accuracy differences are rounded to one decimal place after calculation; differences are not obtained by subtracting already rounded Mean entries. Baselines and default PRUNE-OPD rows repeated from the main tables are the same runs, not additional independent measurements.

A.7.1 Component Ablations

Table 6 reports weighting-only, truncation-only, and full-method results for a low-compatibility Qwen3 pair and the improving-compatibility JustRL pair. Weighting only retains OPD rollout budgets; truncation only uses overlap-based budget control without reward reweighting. The latter still computes reliability to determine effective lengths. Both components are active in the full method.

For the Qwen3-4B pair, weighting alone increases Mean accuracy by 0.4 points and truncation alone by 0.9 points relative to OPD; the full method increases it by 1.9 points while reducing time from 28.5h to 13.5h. For JustRL, weighting alone decreases Mean accuracy by 0.3 points; truncation alone increases it by 0.4 points and the full method by 0.7 points. The full method has the highest Mean accuracy in both pairs, but is not best on every benchmark and is 0.2h slower than truncation alone on JustRL.

A.7.2 Continued Adaptation versus a Fixed Initial Budget

Table 3 reports the full benchmark scores for the comparison in Section 4.5. The fixed-1K variant uses the same first-batch procedure and reward weighting as full PRUNE-OPD. Its first-batch

Table 6: Full component-ablation results. Mean averages the five benchmark scores; time is reported in hours.

Method	AMC23	AIME24	AIME25	HMMT24	HMMT25	Mean	Time
Qwen3-4B-Base / Qwen3-4B (Non-thinking)							
OPD	42.7	14.8	10.6	5.4	2.9	15.3	28.5h
Weighting only	**43.0**	15.4	12.3	4.8	3.1	15.7	29.3h
Truncation only	41.9	15.6	13.3	6.0	4.0	16.2	13.8h
PRUNE-OPD	42.7	**16.3**	**14.6**	**6.7**	**5.6**	**17.2**	**13.5h**
DeepSeek-R1-Distill-Qwen-1.5B / JustRL-DeepSeek-1.5B							
OPD	77.5	43.5	32.7	**23.5**	18.1	39.1	6.9h
Weighting only	77.3	43.8	31.5	23.1	17.9	38.7	7.0h
Truncation only	77.8	44.6	33.1	22.3	**19.6**	39.5	**3.9h**
PRUNE-OPD	**78.4**	**45.2**	**33.5**	22.7	18.8	**39.7**	4.1h

average effective length is below 1,024, so the minimum determines the cutoff; subsequent budget updates are disabled. The first batch is still generated at the maximum budget and truncated before the update. Fixed-4K OPD instead uses the unweighted baseline reward routine. The two fixed rows therefore test different settings, with fixed-1K providing the control for continued adaptation under unchanged reward weighting.

Fixing the budget at 1K is fastest but reduces Mean accuracy by 3.0 points relative to OPD and 3.6 points relative to PRUNE-OPD. Fixed-4K recovers 2.4 points over fixed-1K, but takes 4.8h rather than 3.2h and scores below OPD on four benchmarks. Full PRUNE-OPD reaches a higher Mean than either fixed setting in 4.1h. Together with the growing effective lengths in Figure 4, these results support allowing the budget to expand beyond its initial value. The practical benefit is selecting and updating that length during OPD, when compatibility can be observed, rather than choosing a cutoff or growth schedule beforehand. This comparison does not establish superiority over every tuned schedule or measure the cost of an external length search.

A.7.3 REPEATED RUNS ON THE LOW-COMPATIBILITY QWEN3-4B PAIR

Table 7 reports three training runs per method for Qwen3-4B-Base / Qwen3-4B (Non-thinking). Run 1 is the run already reported in the main tables; runs 2 and 3 provide additional measurements. We compare the per-run scores and training times to assess whether the observed accuracy improvement and speedup extend beyond a single run.

Table 7: Repeated runs for the low-compatibility Qwen3-4B-Base / Qwen3-4B (Non-thinking) pair. Benchmark scores and Mean are percentages; time is in hours. Run indices do not imply matched random seeds across methods.

Method / run	AMC23	AIME24	AIME25	HMMT24	HMMT25	Mean	Time
Qwen3-4B-Base / Qwen3-4B (Non-thinking)							
OPD / 1	42.7	14.8	10.6	5.4	2.9	15.3	28.5h
OPD / 2	43.3	15.2	11.3	5.4	3.5	15.7	28.0h
OPD / 3	42.8	15.2	11.5	5.8	4.0	15.9	28.2h
PRUNE-OPD / 1	42.7	16.3	14.6	6.7	5.6	17.2	13.5h
PRUNE-OPD / 2	42.3	16.5	14.2	6.3	5.2	16.9	13.1h
PRUNE-OPD / 3	42.0	15.8	13.8	6.7	5.8	16.8	13.6h

Across the three runs per method, Mean accuracy ranges from 15.3% to 15.9% for OPD and from 16.8% to 17.2% for PRUNE-OPD: every PRUNE-OPD run exceeds every OPD run. Averaging the per-run benchmark means before rounding gives 15.6% for OPD and 17.0% for PRUNE-OPD, a 1.3-point difference. Average training time falls from 28.2h to 13.4h, a 52.5% reduction calculated be-

fore rounding. Each PRUNE-OPD run also exceeds all OPD runs on AIME24, AIME25, HMMT24, and HMMT25; AMC23 does not show the same advantage. These repeated experiments support the average-accuracy improvement and training speedup on this low-compatibility pair beyond the main-table run.

A.7.4 BASE-WEIGHT SENSITIVITY

Table 8 varies w_{base} on the JustRL pair. The default 0.5 gives the highest Mean accuracy, while 1.0 is 0.2 points lower and has higher scores on two individual benchmarks. Thus, a larger base weight does not monotonically improve Mean accuracy.

Table 8: Base-weight ablation on DeepSeek-R1-Distill-Qwen-1.5B / JustRL-DeepSeek-1.5B. No timing measurements are reported for the non-default settings.

Method	AMC23	AIME24	AIME25	HMMT24	HMMT25	Mean
OPD	77.5	43.5	32.7	**23.5**	18.1	39.1
$w_{\text{base}} = 0$	77.0	42.3	33.5	23.1	18.5	38.9
$w_{\text{base}} = 0.5$	**78.4**	**45.2**	33.5	22.7	**18.8**	**39.7**
$w_{\text{base}} = 1.0$	78.0	44.8	**33.8**	23.1	18.1	39.6

At $w_{\text{base}} = 0$, Eq. (6) gives weights in $[0, 1]$, so the method cannot amplify any token's reward through its multiplier. Its Mean accuracy is 38.9%, close to OPD's 39.1%. Performance retention therefore does not require multipliers above one in this setting. This is not a matched-reward-scale test: changing the base also changes the contribution of low-reliability tokens and the resulting training trajectory. These accuracy-only results do not establish efficiency differences across base weights.

A.8 EFFECTIVE TRAINING LENGTH IN THE LOW-COMPATIBILITY QWEN3 PAIRS

The Qwen3-1.7B-Base / Qwen3-4B (Non-thinking) and Qwen3-4B-Base / Qwen3-4B (Non-thinking) pairs exhibit low student–teacher compatibility as measured by top-k overlap. Both low-compatibility pairs match or improve on OPD accuracy while reducing training time (Table 1). Figure 6 shows effective-length dynamics in these two low-compatibility pairs: lengths remain within a few hundred tokens throughout the observed run, compared with the baseline's 12,288-token rollout budget. The lengths fluctuate and sometimes increase, but remain much shorter than those in the JustRL and Skywork pairs.

Long, weakly compatible suffixes may contribute many gradient terms relative to a short compatible prefix. Reweighting rewards and reducing subsequent rollout budgets could limit their influence and explain some of the accuracy gains. The component ablation in Table 6 separates their empirical effects on Qwen3-4B, but gradient quality has not been measured, so this denoising explanation remains a hypothesis. The plotted E counts positions above the reliability threshold; actual responses can be longer and still receive the base reward weight.

A.9 DEPTH-WISE OVERLAP UNDER OPD

Figure 7 isolates the compatibility dynamics of the unpruned OPD baseline. The shallow 0–1K band acts as a reference for early-prefix supervision, while the later bands test whether teacher supervision remains locally aligned after long student-generated reasoning prefixes. The deepest plotted band (6–7K) has lower and more variable overlap than the shallower bands over much of training. This illustrates depth-dependent compatibility, but does not directly measure gradient usefulness or establish a causal effect on final correctness.

A.10 THRESHOLD DIAGNOSTICS

Table 9 reports the overlap-threshold sweep on the JustRL pair. Increasing γ from 0.6 to 0.9 reduces training time from 4.8h to 2.4h. The default 0.7 gives the best AIME25 and HMMT24 scores; 0.6 leads on AMC23 and AIME24. Values 0.8 and 0.9 reduce time further but lower all five scores relative to 0.7. All five main pairs use the same default, but this sweep covers only JustRL.

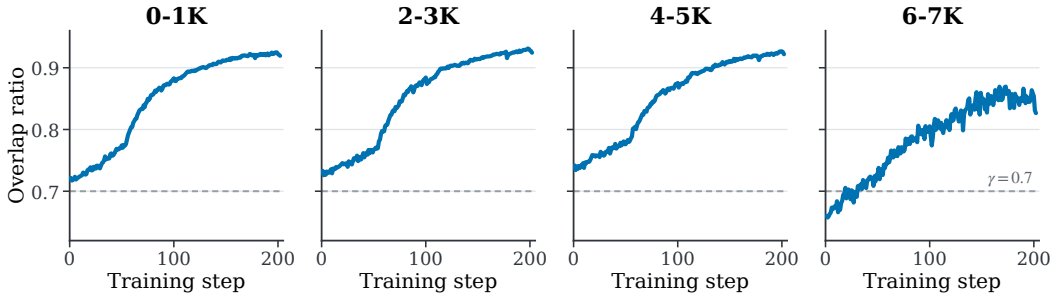

Figure 7: OPD baseline overlap-ratio training dynamics for DeepSeek-R1-Distill-Qwen-1.5B / JustRL-DeepSeek-1.5B. Each panel plots overlap ratio versus training step for a token-position band: 0–1K, 2–3K, 4–5K, and 6–7K. This diagnostic shows how local student-teacher compatibility evolves at different trajectory depths under unpruned OPD.

Table 9: Overlap-threshold ablation on DeepSeek-R1-Distill-Qwen-1.5B / JustRL-DeepSeek-1.5B. Bold marks column-wise best values (lower is better for time).

Threshold γ	AMC23	AIME24	AIME25	HMMT24	HMMT25	Time	Red.
0.6	**78.9**	**45.4**	33.1	21.9	**18.8**	4.8h	30.4%
0.7	78.4	45.2	**33.5**	**22.7**	**18.8**	4.1h	40.6%
0.8	77.8	44.4	31.5	21.9	17.9	3.1h	55.1%
0.9	76.7	43.8	30.8	21.7	17.9	**2.4h**	**65.2%**

Figure 8 plots reward weights and maximum response lengths for $\gamma = 0.6, 0.7, 0.8$. Larger thresholds produce earlier weight decay and shorter reported lengths. To correct the recorded length statistic, the right panel applies a 1,024-token floor: it displays $\max(1024, \widehat{T}_t^{\max})$, where $\widehat{T}_t^{\max}$ is the recorded maximum generated response length for the step's batch. Values above 1,024 are unchanged. These floor-corrected statistics differ from both the configured rollout budget M_t and reliability-defined effective length E. The displayed floor is a correction to this diagnostic, not a minimum generated length enforced by the algorithm.

A.11 Additional Distribution Diagnostics

The reference logger averages local candidate agreement over all valid response tokens in the batch, excluding padding. Equivalently, it divides the total overlap count by k times the valid-token count; longer responses therefore contribute more tokens, rather than receiving equal response-level weight. The expectation below is over valid token positions across the batch:

$$\mathcal{M}_{\text{overlap}} = \mathbb{E}_\tau \left[\frac{|\mathcal{K}_\tau^{\text{S}} \cap \mathcal{K}_\tau^{\text{T}}|}{k} \right]. \tag{18}$$

A separate diagnostic considered in OPD analysis (Li et al., 2026b) is the entropy gap,

$$\Delta H_\tau = |H(q_\tau) - H(p_\tau)|. \tag{19}$$

It measures a difference in uncertainty rather than candidate-set agreement. The current controller uses per-position overlap (or top-p acceptance in the alternative variant), not entropy gap or a threshold on batch-average overlap.

A.12 Additional Related Work

Knowledge distillation and off-policy distillation. Knowledge distillation transfers information from a teacher to a student by matching the teacher's soft output distribution (Hinton et al., 2015). For autoregressive sequence models, sequence-level distillation trains the student on teacher-generated outputs (Kim & Rush, 2016), and this off-policy paradigm underlies many compact language-model distillation methods (Sanh et al., 2019; Jiao et al., 2020; Wang et al., 2020; Gu et al., 2026). Supervised fine-tuning and instruction tuning similarly improve downstream behavior

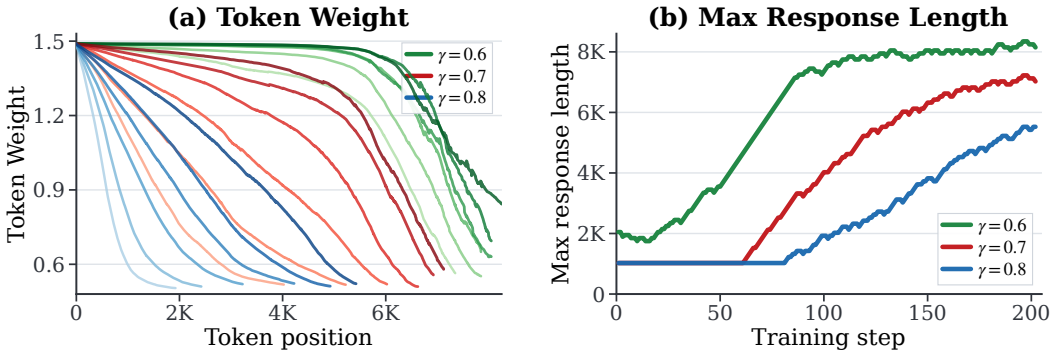

Figure 8: Prune-OPD threshold diagnostics for DeepSeek-R1-Distill-Qwen-1.5B / JustRL-DeepSeek-1.5B. Left: mean actual loss weight L_τ, including the 0.5 base floor, as a function of token position under three overlap thresholds, $\gamma = 0.6, 0.7, 0.8$; for each threshold, the curves are taken at training steps 100, 120, 140, 160, 180, and 200. Within each threshold, darker curves indicate later steps. Each position averages valid responses reaching it, excluding padding; means at the base floor are omitted. Right: maximum response length over training steps, with the recorded statistic corrected by a 1,024-token floor, $\max(1024, \widehat{T}_t^{\max})$. Values above the floor are unchanged. This floor-corrected statistic is neither the rollout budget nor effective training length.

by training on curated responses or demonstrations (Chung et al., 2024; Sanh et al., 2021; Wei et al., 2021). The common limitation is train-inference distribution mismatch: the student is optimized on teacher or reference trajectories, but at inference time it must condition on prefixes sampled from its own policy, an exposure-bias problem that compounds over long generations (Bengio et al., 2015). PRUNE-OPD retains student-generated trajectories and adjusts the reward weights and rollout budget according to local teacher-student overlap.

OPD dynamics and mechanisms. Recent OPD analysis shows that OPD success is governed by thinking-pattern consistency and by whether the teacher contributes new knowledge beyond what the student has already acquired (Li et al., 2026b). At the token level, successful OPD exhibits progressive alignment on high-probability tokens: top-k overlap rises, overlap-token advantage improves, and the entropy gap narrows. The same analysis further shows that reward quality degrades with trajectory depth, and that a reward can remain globally correlated with final correctness while failing to provide locally useful gradients around the student's current policy. PRUNE-OPD uses overlap ratio, or teacher acceptability in the alternative variant, to weight rewards and set the next rollout budget. It does not use the entropy gap.

Capacity gap and distillability. A separate line of work studies when a teacher is too strong, too different, or too complex for the student to imitate. Studies have shown that overly capable teachers can hurt student performance (Cho & Hariharan, 2019), and teacher assistants have been introduced to bridge large capacity gaps (Mirzadeh et al., 2020). Distillation scaling laws identify non-monotonic interactions among teacher quality, student size, and data volume (Busbridge et al., 2025). For reasoning models, small students can struggle to learn from strong reasoners with long chain-of-thought traces, suggesting a learnability gap in distilling complex reasoning behavior (Li et al., 2025). These results concern teacher suitability at the model level. PRUNE-OPD uses compatibility on student-generated prefixes to adjust reward weights and training length as the student learns.

Reasoning RL and long-horizon supervision. Modern reasoning models rely on verifiable RL, long chain-of-thought rollouts, and large mathematical training sets (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025; He et al., 2025; Yang et al., 2026c;d; 2025b; Xu et al., 2026). Long responses create room for search and self-correction, but they also make dense supervision expensive and harder to trust. Prior OPD results identify a response-length sweet spot: short responses provide too little dense signal, while very long responses can trigger late-stage collapse and suffix-to-prefix instability (Li et al., 2026b). PRUNE-OPD adjusts the rollout budget using the effective training lengths measured in the current batch.

Further OPD and self-distillation recipes. Related work explores entropy, stability, and co-evolving distillation policies (Ko et al., 2026; Jin et al., 2026; Jang et al., 2026; Gu et al., 2026), as well as self-distillation and online experiential learning (He 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., 2026a; Zhao et al., 2026a; Ding, 2026; Song et al., 2026). These approaches concern the construction or use of teacher feedback; our focus is adapting training length through compatibility feedback, coupled with position-wise reward weighting.

A.13 Limitations and Future Directions

Our experiments focus on mathematical reasoning with single-turn student-generated responses. Extending PRUNE-OPD to other domains and multi-turn interactions remains future work. The method uses shared overlap and controller parameters rather than manually specified per-pair cutoffs; its training-time savings depend on how much compatible supervision each pair retains. Probability-aware compatibility measures and recovery-aware weighting are natural extensions of the current top-k overlap and cumulative-decay rules.