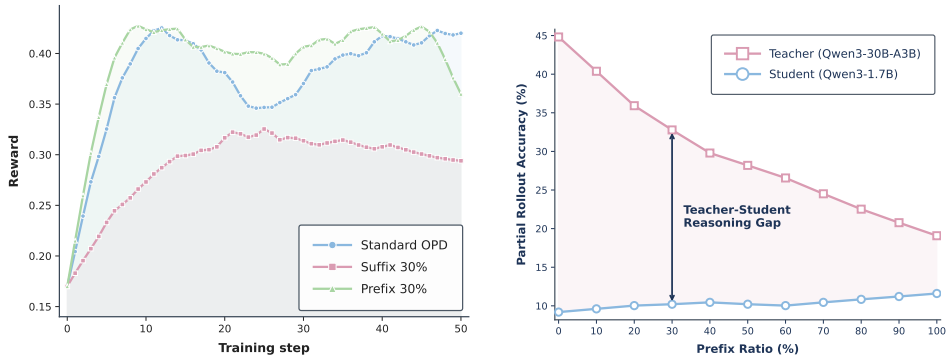
ON THE POSITION BIAS OF ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-Policy Distillation (OPD) improves the learning efficiency of standard reinforcement learning through dense, token-level supervision from teachers. In the standard KL objective of OPD, token-level losses are uniformly averaged, implying equal weights for all tokens. However, we discover that not all tokens are created equal: as student rollouts grow longer, they deviate further from the teacher's distribution, leading to degraded supervision quality at later positions. As a result, OPD using only the first 30% of tokens can perform comparably to using all tokens, whereas OPD using only the last 30% of tokens barely learns anything. In this work, we provide a principled understanding of this issue through the lens of constrained optimization. Based on these insights, we derive Importance-Weighted On-Policy Distillation (IW-OPD), in which the weight assigned to each token depends on the accumulated discrepancy between the student's and teacher's distributions, naturally upweighting earlier tokens and downweighting later ones with larger deviations. We show that IW-OPD converges significantly faster than OPD, with better learning efficiency, and achieves better final performance than standard OPD in both same-size and cross-scale settings, improving performance by 6.9 points on AIME-2025.



(a) OPD training with the same token budget but supervision applied to different token positions.

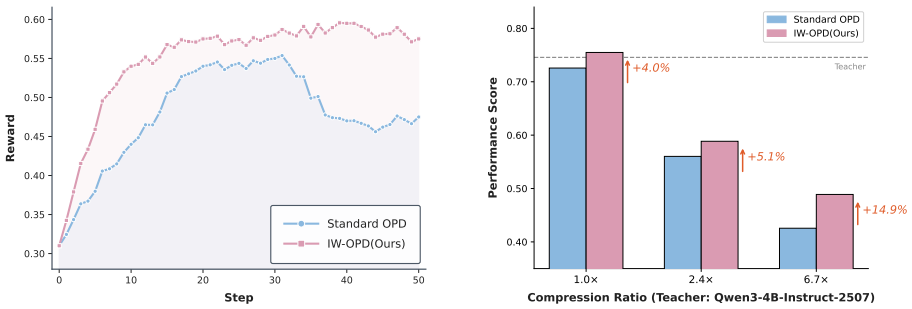
(b) Teacher–student gap in final accuracy when conditioned on student-generated prefixes.

Figure 1: **Position Bias in OPD training. (a)** With the same 30% token budget, training on the prefix part of each response matches or exceeds full token Standard OPD, whereas training on the suffix part fails to learn effectively. Student: Qwen3-0.6B, Teacher: Qwen3-4B-Instruct-2507. **(b)** Teacher and student accuracy are measured by the probability of reaching a correct answer from a given student-generated prefix. Student model maintains a low accuracy, while the teacher model's mean@32 accuracy of eventually reaching the correct answer drops rapidly toward the student level as the student-generated prefix becomes longer.

1 INTRODUCTION

On-Policy Distillation (OPD) trains a student on its own rollouts, using dense token-level teacher supervision (Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025; Yang et al., 2025; LLM-Core Xiaomi, 2026) to improve learning efficiency over sparse trajectory-level rewards in LLM post-training (Cui et al., 2025; He et al., 2026).

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.



(a) AIME25 accuracy during training. Teacher: Qwen3-4B, Student: Qwen3-1.7B.

(b) Final accuracy vs. compression ratio (teacher parameters / student parameters)

Figure 2: **IW-OPD improves both sample efficiency and final performance. (a)** AIME 2025 accuracy during training: IW-OPD converges faster and achieves better final performance than Standard OPD. **(b)** Final accuracy across student scales distilled from the same teacher; the IW-OPD advantage grows from $+4.0\%$ at $1.0\times$ compression to $+14.9\%$ at $6.7\times$.

Standard OPD uniformly aggregates per-token KL divergence, although weak students often generate prefixes outside the teacher's preferred distribution. Fig. 1b shows that the teacher's ability to reach a correct answer deteriorates as student-generated prefixes grow longer. This reveals a *position bias*: early tokens provide more useful supervision than later tokens. A controlled study confirms this pattern: with the same 30% token budget, prefix-only OPD matches or exceeds full OPD, whereas suffix-only OPD provides little benefit (Fig. 1a).

These observations suggest viewing OPD as an allocation problem under a finite local-update budget. With limited policy movement per update, more gradient budget should target prefixes where teacher supervision remains compatible with the student's trajectory. We formalize this view through a constrained local projection toward the teacher, whose closed-form optimum reweights the student distribution by a teacher-to-student likelihood ratio. As student rollouts drift from teacher-preferred reasoning paths, optimal policy naturally reduces the sampling probability of downstream tokens.

IW-OPD (**I**mportance-**W**eighted **O**n-**P**olicy **D**istillation) implements this principle by reweighting token-level distillation terms with prefix-importance weights induced by the constrained projected objective. It requires no additional teacher evaluations and recovers standard OPD when the extra weighting is removed. Experiments show faster convergence and stronger final performance (Fig. 2), with AIME25 gains of $+6.9$ points at step 10 and $+1.7$ points at convergence. Moreover, IW-OPD makes stronger teachers more sample-efficient and yields larger relative gains as students become smaller. Our contributions are:

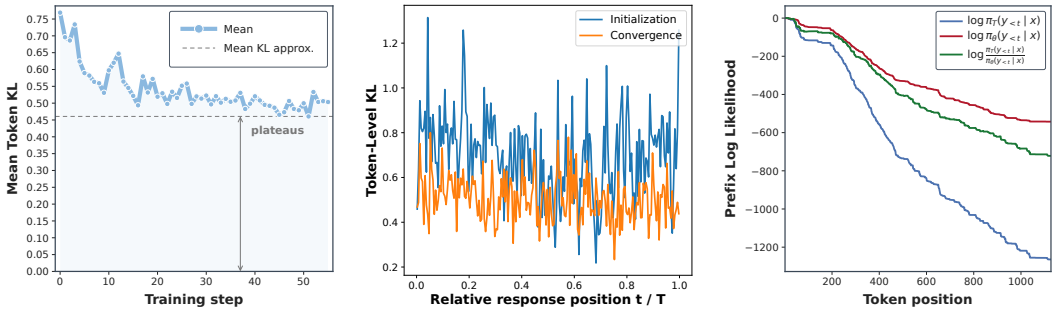
1. We identify the *position bias* phenomenon in OPD and explain it from a constrained-optimization perspective. This view shows why teacher-compatible prefixes dominate useful supervision. (Sec. 3).
2. We propose IW-OPD, an efficient objective with token-level importance estimated from teacher–student discrepancy (Sec. 4).
3. We demonstrate improved sample efficiency and final performance: stronger teachers become more sample-efficient, while smaller students obtain larger gains (Sec. 5).

2 Preliminaries

Let $\mathcal{D}$ denote the prompt distribution, π_θ the student policy, and π_T the teacher policy. For a prompt x and response $y = (y_1, \ldots, y_T)$, the trajectory-level distributions decompose autoregressively:

$$\pi_\theta(y|x) = \prod_{t=1}^{T} \pi_\theta(y_t|x, y_{<t}), \qquad \pi_T(y|x) = \prod_{t=1}^{T} \pi_T(y_t|x, y_{<t}). \tag{1}$$

On-Policy RL. RLVR samples responses from the current policy and optimizes a trajectory-level reward, assigning the resulting credit across response tokens.



(a) Mean token-level reverse KL across training steps.

(b) Token-level reverse KL before and after OPD training.

(c) Log-likelihoods gap of student-sampled prefix vs. prefix length.

Figure 3: **Position Bias phenomena in OPD. (a)** The mean token-level KL decreases during OPD training but plateaus at a non-zero residual. **(b)** Token-level reverse KL before and after OPD training. **(c)** Sequence-level log-probabilities of student-sampled prefixes under the student and teacher. Student: Qwen3-0.6B; Teacher: Qwen3-4B-Instruct.

On-Policy Distillation (OPD). OPD (Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025) replaces the sparse trajectory-level reward with dense token-level supervision from a teacher model π_T (Yang et al., 2025; LLM-Core Xiaomi, 2026):

$$\mathcal{J}_{\mathrm{OPD}}(\theta) = \max_{\theta} -D_{\mathrm{KL}}(\pi_\theta || \pi_T) = -\mathbb{E}_{x,\, y\sim\pi_\theta} \sum_{t=1}^{T} \log \frac{\pi_\theta(y_t|x, y_{<t})}{\pi_T(y_t|x, y_{<t})}, \tag{2}$$

In practice, OPD uses a *token-level* reverse-KL approximation to optimize the *sequence-level* objective, treating sampled prefixes as fixed (LLM-Core Xiaomi, 2026; Lu & Thinking Machines Lab, 2025). This yields the following *token-local* semi-gradient (App. A.3):

$$\nabla_\theta \mathcal{J}_{\mathrm{OPD}} \approx \mathbb{E}_{x,\, y\sim\pi_\theta}\left[\sum_{t=1}^{T} A_t^{\mathrm{OPD}} \nabla_\theta \log \pi_\theta(y_t|x, y_{<t})\right], \tag{3}$$

where OPD assigns a token-level advantage from the teacher–student distribution gap:

$$A_t^{\mathrm{OPD}} := -(\log \pi_\theta(y_t|x, y_{<t}) - \log \pi_T(y_t|x, y_{<t})). \tag{4}$$

3 Position Bias in On-Policy Distillation

Standard OPD provides dense token-level supervision, but it aggregates all token-level KL terms uniformly in Eq. (2). In this section, we observe that OPD actually has a clear *position bias*: its early tokens are much more valuable for learning compared to its later tokens. We will show two interesting empirical phenomena in Sec. 3.1 and provide a theoretical explanation in Sec. 3.2.

3.1 The Position Bias Phenomenon in OPD

Early-token supervision drives OPD performance. To evaluate the influence of token positions in OPD learning, we fix the supervision budget and vary only the supervised segment. Prefix-30 applies OPD to the first 30% response tokens, while Suffix-30 applies OPD to the last 30%; Standard OPD uses all valid tokens. All other training settings are unchanged (details in App. C).

Fig. 1a shows a strong asymmetry. Supervising only the prefix 30% of tokens achieves performance comparable to standard OPD and consistently outperforms supervising only the suffix 30%. In contrast, suffix-only supervision yields substantially lower rewards throughout training. These results indicate that OPD benefits primarily from early-token supervision, while supervision on later tokens alone provides limited gains.

Teacher–student gap largely persists after OPD. Meanwhile, we also observe that even if the OPD objective tries to minimize the KL divergence between the teacher and student, the actual

divergence between these two only decreases by 20% even if training converges and the student performance saturates (shown in Fig. 3a and Fig. 3b). It suggests that OPD training effectively only optimizes the student within a small local region. This could be because the OPD only optimizes on student-generated samples and this kind of RL training is known to produce minimal weight update(Wang et al., 2025; Bigelow et al., 2025; Lin et al., 2025; Vassoyan et al., 2025; Ye et al., 2025).

The two phenomena combined suggest an interesting learning landscape in OPD learning: *it only optimizes the student distribution locally and most of the gains come from early prefix tokens*. Why does OPD have such a position bias and what does it imply for learning efficiency? We provide a theoretical explanation of this phenomenon in the next section.

3.2 Understanding Position Bias from a Finite-Budget Allocation Perspective

As the empirical result in Sec. 3.1 indicates that OPD only moves the student distribution within a small range, we can think of the actual OPD training as a constrained optimization problem, where the student distribution stays in a local region during training:

$$\min_{q} \; D_{\mathrm{KL}}(q\|\pi_T) \quad \text{s.t.} \quad D_{\mathrm{KL}}(q\|\pi_\theta) \le \rho, \tag{5}$$

where π_θ denotes the student distribution, and ρ denotes the effective local update budget measured by KL divergence. It can also be viewed as a trust-region objective where the policy is only updated within a trust region of radius ρ. In fact, this constrained objective admits a **closed-form** solution $q^\star$, as revealed in the following proposition. The proof can be found at App. A.1.

Proposition 1 (Optimal Policy)**.** *Given π_θ and π_T with common support. In the local-update regime $0 < \rho < D_{\mathrm{KL}}(\pi_T\|\pi_\theta)$, the trust-region constraint in Eq.* (5) *is active and the unique solution is*

$$q^\star_\theta(y) = \frac{\pi_\theta(y)}{Z_\alpha(\theta)} \left(\frac{\pi_T(y)}{\pi_\theta(y)}\right)^\alpha, \tag{6}$$

which reweights the student policy π_θ by the likelihood ratio $r_\theta(y) = \pi_T(y)/\pi_\theta(y)$. Here, $Z_\alpha(\theta) = \mathbb{E}_{y\sim\pi_\theta}\left[r_\theta(y)^\alpha\right]$ denotes the normalizing factor, and $\alpha \in (0,1)$ is a constant depends on ρ.

The optimal policy $q^\star$ is proportional to likelihood ratio. This likelihood-based reweighting view in Prop. 1 explains why the position bias in Fig. 1a arises. Along student rollouts, the prefix probability under the student, $\pi_\theta(y_{<t})$, usually decreases smoothly because the sequence is sampled from π_θ. In contrast, the same prefix probability under the teacher, $\pi_T(y_{<t})$, can drop much faster once early student decisions move the reasoning path away from the teacher-preferred region, as illustrated in Fig. 3c. Therefore the prefix ratio $r_\theta(y_{<t})^\alpha$ tends to become smaller at later positions. The constrained optimum would allocate less mass to such low-ratio prefixes, while uniform OPD continues to spend the same update budget on their suffix tokens. Thus, position bias is a consequence of applying uniform token-level updates to a local projection problem whose optimal solution is inherently ratio-weighted.

4 Importance-Weighted On-Policy Distillation

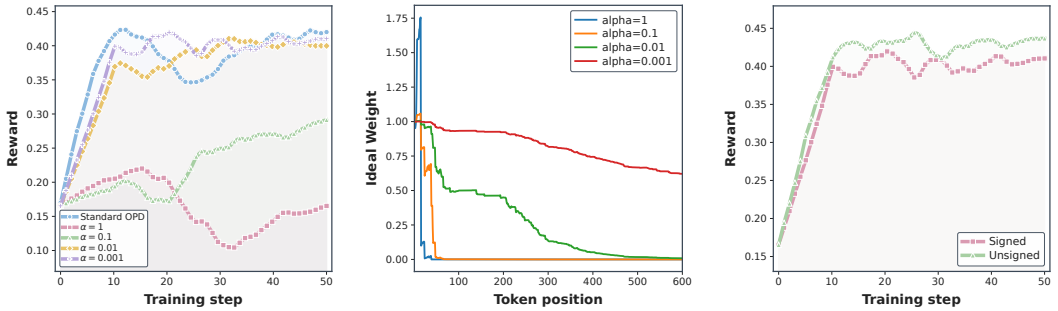
Based on the insights from Sec. 3, in this section, we propose a more efficient OPD objective, Importance-Weighted OPD, that leverages the position bias to learn more efficiently.

4.1 Importance-Weighted OPD

As discussed in Sec. 3.1, OPD can only optimize within a local region and the optimal policy it could attain is $q^\star_\theta$ (Eq. 6), which reweights the base student policy π_θ with teacher–student gap measured by the likelihood ratio. As discussed in Sec. 3.2, only tokens with relatively high likelihood ratios provide meaningful learning signals in OPD since the others have very low probability of being sampled.

Motivated by this observation, we propose to directly optimize the divergence between the optimal policy $q^\star$ and the teacher, which directly emphasizes high-probability samples:

$$\max_{\theta} \; \mathcal{J}_{q^\star_\theta} = -D_{\mathrm{KL}}(q^\star_\theta\|\pi_T). \tag{7}$$



(a) α in prefix-importance weights ablation.

(b) Prefix-importance weights visualization.

(c) Signed vs. Unsigned prefix-importance weights ablation.

Figure 4: **From signed prefix ratio to unsigned prefix discrepancy. (a)** Directly using the ideal prefix ratio is sensitive to α. **(b)** Token-wise visualization shows the desired overall downward trend, but also local rebounds caused by signed cancellation. **(c)** Replacing signed accumulation with the unsigned weight gives a more stable weighting signal.

Since $q_\theta^\star$ is a reparameterized distribution induced by the student and teacher, we can further reparameterize this new learning objective by sampling from π_θ instead. More specifically, it will be equivalent to an Importance-Weighted OPD (IW-OPD) objective, as revealed in the next proposition.

Importance-weighted form of the projected objective. For clarity, we first fix the prompt x and omit it from the notation. We consider the non-trivial local-update regime $0 < \rho < D_{\mathrm{KL}}(\pi_T \| \pi_\theta)$ from Prop. 1. In this regime α is induced by the effective step-size budget ρ, and the trust-region constraint is active as $D_{\mathrm{KL}}(q_\theta^\star \| \pi_\theta) = \rho$. Thus we have:

$$\begin{aligned} D_{\mathrm{KL}}(q_\theta^\star \| \pi_T) &= \mathbb{E}_{q_\theta^\star}\left[\log \frac{q_\theta^\star(y)}{\pi_T(y)}\right] \\ &= \mathbb{E}_{q_\theta^\star}\left[\log \frac{q_\theta^\star(y)}{\pi_\theta(y)}\right] + \mathbb{E}_{q_\theta^\star}\left[\log \frac{\pi_\theta(y)}{\pi_T(y)}\right] \\ &= \rho + \mathbb{E}_{q_\theta^\star}\left[\log \frac{\pi_\theta(y)}{\pi_T(y)}\right]. \end{aligned} \tag{8}$$

Proposition 2 (Importance-Weighted OPD Objective)**.** *By applying a change of measure from $q_\theta^\star$ to π_θ and substituting Eq.* (6)*, we obtain an importance-weighted trajectory-level objective and define the following token-level surrogate (proof in App. A.2):*

$$\mathcal{J}_{\mathrm{IW}}^\star(\theta) = -\mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{T} \mathrm{sg}\left[\tilde{r}_t\right] \log \frac{\pi_\theta(y_t \mid y_{<t})}{\pi_T(y_t \mid y_{<t})}\right], \quad \tilde{r}_t = \frac{r_t}{Z_{\alpha,t}}. \tag{9}$$

where $r_t := r_\theta(y_{<t}) = \pi_T(y_{<t})/\pi_\theta(y_{<t})$ depends on α and denotes the prefix likelihood ratio at position t, inherited from the trajectory-level ratio in Prop. 1. The position-wise normalizer is $Z_{\alpha,t} = \mathbb{E}_{y_{<t}\sim\pi_\theta}[r_t]$. $\mathrm{sg}[\cdot]$ is stop gradient operator.

Eq. (9) shows that the Eq. (7) objective can be optimized using π_θ-sampled rollouts as standard OPD, with each token-level KL term reweighted by a detached, normalized prefix-importance weight $\tilde{r}_t$. This weight, introduced by $q_\theta^\star$, becomes larger for teacher-compatible prefixes with high teacher–student likelihood ratio and smaller after accumulated teacher–student drift.

Similar to OPD, the gradient of IW-OPD can be written as a policy gradient form with importance-weighted advantage (proof in App. A.3):

$$\nabla_\theta \mathcal{J}_{\mathrm{IW}}^\star(\theta) \approx \mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{T} A_t^{\text{IW-OPD}} \nabla_\theta \log \pi_\theta(y_t \mid y_{<t})\right], \tag{10}$$

where

$$A_t^{\text{IW-OPD}} = -\mathrm{sg}\left[\tilde{r}_t\right]\left(\log \pi_\theta(y_t \mid y_{<t}) - \log \pi_T(y_t \mid y_{<t})\right). \tag{11}$$

Table 1: Evaluation results with **Qwen3-30B-A3B-Instruct-2507** as teacher (small student–teacher overlap). Math results are reported as mean@32 accuracy (%). Methods with subscript 10 are evaluated at training step 10. **Bold** indicates the best result within each student group.

Student	Method	Math			Code		Avg
		AIME24	AIME25	HMMT25	HE+	MBPP+	
Teacher Model		74.7	62.8	44.2	86.6	75.1	68.7
Qwen3-4B	Base	23.1	21.4	10.0	75.3	64.5	38.9
	OPD_{10}	51.2	42.4	23.5	76.2	66.7	52.0
	IW-OPD_{10}	$56.2_{+5.0}$	$49.3_{+6.9}$	$27.3_{+3.8}$	$76.8_{+0.6}$	$68.1_{+1.4}$	$55.5_{+3.5}$
	OPD	55.3	48.0	27.1	77.2	69.1	55.3
	IW-OPD	$\mathbf{57.5}_{+2.2}$	$\mathbf{49.7}_{+1.7}$	$\mathbf{28.7}_{+1.6}$	$\mathbf{78.7}_{+1.5}$	$\mathbf{70.9}_{+1.8}$	$\mathbf{57.1}_{+1.8}$
Qwen3-1.7B	Base	13.4	11.0	6.8	59.6	52.5	28.7
	OPD_{10}	30.5	20.2	14.4	53.0	52.6	34.1
	IW-OPD_{10}	$33.0_{+2.5}$	$23.2_{+3.0}$	$15.6_{+1.2}$	$55.5_{+2.5}$	$54.8_{+2.2}$	$36.4_{+2.3}$
	OPD	34.6	28.7	15.5	64.6	53.7	39.4
	IW-OPD	$\mathbf{35.5}_{+0.9}$	$\mathbf{29.5}_{+0.8}$	$\mathbf{16.4}_{+0.9}$	$\mathbf{65.2}_{+0.6}$	$\mathbf{55.0}_{+1.3}$	$\mathbf{40.3}_{+0.9}$
Qwen3-0.6B	Base	1.5	3.4	1.3	28.2	28.4	12.6
	OPD_{10}	6.2	14.1	6.5	26.9	23.1	15.4
	IW-OPD_{10}	$7.8_{+1.6}$	$15.8_{+1.7}$	$7.6_{+1.1}$	$28.4_{+1.5}$	$27.5_{+4.4}$	$17.4_{+2.0}$
	OPD	11.0	17.8	7.1	29.6	28.7	18.8
	IW-OPD	$\mathbf{11.5}_{+0.5}$	$\mathbf{19.3}_{+1.5}$	$\mathbf{8.0}_{+0.9}$	$\mathbf{32.5}_{+2.9}$	$\mathbf{31.9}_{+3.2}$	$\mathbf{20.2}_{+1.4}$

With stop-gradient applied, the coefficient in Eq. (10) acts as a multiplicative weight on the OPD policy-gradient signal. Thus, the weighted gradient is the operational form of the constrained-optimization view in Eq. (7): it corrects OPD's position bias by reallocating the finite update budget through prefix-importance weights.

4.2 Stable Token-level Importance Weight Estimate

Sequence-level reverse-KL optimization can produce high-variance updates (Gu et al., 2024), motivating the token-level approximation used in standard OPD (Sec. 2). The formulation in Sec. 4.1 still faces instability because its prefix importance weights accumulate discrepancies along the trajectory and can vary sharply in magnitude. A small $\alpha \to 0$ can flatten these differences, but is still insufficient. Fig. 4a shows that $\alpha = 1$ and $\alpha = 0.1$ substantially degrade training, while only very small values such as $\alpha = 0.01$ and $\alpha = 0.001$ roughly match standard OPD. Fig. 4b shows how α controls the scale and sharpness of the weights, making them overly concentrated or nearly flat. We therefore approximate the prefix weights for stable training through log scaling, unsigned accumulation, and within-sample normalization, ablated in Sec. 5.3.

I. Stabilization via log scaling. We first consider the probability ratio in log space to preserve its ordering while mitigating the exponentially accumulated gaps:

$$\tilde{r}_t^{\log} = \log r_t = \alpha \sum_{k<t} \left(\log \pi_T(y_k|y_{<k}) - \log \pi_\theta(y_k|y_{<k})\right) = \alpha \sum_{k<t} A_k^{\text{OPD}}, \tag{12}$$

This directly corresponds to the sum of token-level OPD advantages A_k^{OPD}.

II. Correcting positive-advantage tokens. Since the rollout tokens are sampled from the student, $\pi_\theta(y_k|y_{<k})$ is often relatively large on these tokens, and thus, as shown in Fig. 3c, $\log \pi_T(y_k|y_{<k}) - \log \pi_\theta(y_k|y_{<k})$ is mostly negative. However, for some tokens where the teacher assigns higher probability than the student, this term becomes positive. These terms can partially offset the accumulated negative prefix gap, as shown in Fig. 4b. In practice, we find it helpful to further reduce this cancellation effect by reverting the positive importance weights, leading to a sum of non-negative token-level discrepencies measured by $|A_k^{\text{OPD}}|$:

$$\tilde{r}_t^{\text{abs}} = \sum_{k<t} \left(\mathbb{I}[A_k^{\text{OPD}} < 0] A_k^{\text{OPD}} - \mathbb{I}[A_k^{\text{OPD}} > 0] A_k^{\text{OPD}}\right) = -\sum_{k<t} \left|A_k^{\text{OPD}}\right|. \tag{13}$$

Table 2: Evaluation results with **Qwen3-4B-Instruct-2507** as teacher (large student–teacher overlap). Math results are reported as mean@32 accuracy (%). Methods with subscript 10 are evaluated at training step 10. **Bold** indicates the best result within each student group.

Student	Method	Math			Code		Avg
		AIME24	AIME25	HMMT25	HE+	MBPP+	
Teacher Model		60.4	46.7	31.0	82.5	71.3	58.4
Qwen3-4B	Base	23.1	21.4	10.0	75.3	64.5	38.9
	OPD_{10}	56.3	45.7	23.6	76.0	66.1	54.0
	IW-OPD_{10}	$58.7_{+2.4}$	$46.7_{+1.0}$	$25.0_{+1.4}$	$77.8_{+1.8}$	$67.5_{+1.4}$	$55.1_{+1.2}$
	OPD	56.5	46.3	24.4	76.3	67.8	54.3
	IW-OPD	$\mathbf{58.7}_{+2.2}$	$\mathbf{46.7}_{+0.4}$	$\mathbf{25.0}_{+0.6}$	$\mathbf{77.9}_{+1.6}$	$\mathbf{68.2}_{+0.4}$	$\mathbf{55.3}_{+1.0}$
Qwen3-1.7B	Base	13.4	11.0	6.8	59.6	52.5	28.7
	OPD_{10}	33.4	24.7	11.3	61.1	53.4	36.8
	IW-OPD_{10}	$35.2_{+1.8}$	$25.9_{+1.2}$	$13.2_{+1.9}$	$62.0_{+0.9}$	$54.0_{+0.6}$	$38.1_{+1.3}$
	OPD	34.0	26.4	13.7	61.5	53.7	37.9
	IW-OPD	$\mathbf{35.2}_{+1.2}$	$\mathbf{27.1}_{+0.7}$	$\mathbf{15.3}_{+1.6}$	$\mathbf{62.8}_{+1.3}$	$\mathbf{54.9}_{+1.2}$	$\mathbf{39.1}_{+1.1}$
Qwen3-0.6B	Base	1.5	3.4	1.3	28.2	28.4	12.6
	OPD_{10}	11.1	17.1	6.9	26.8	31.0	18.6
	IW-OPD_{10}	$12.4_{+1.3}$	$\mathbf{19.0}_{+1.9}$	$9.4_{+2.5}$	$28.1_{+1.3}$	$33.9_{+2.9}$	$20.6_{+2.0}$
	OPD	11.8	17.1	7.7	29.8	33.3	20.0
	IW-OPD	$\mathbf{13.6}_{+1.8}$	$\mathbf{19.0}_{+1.9}$	$\mathbf{9.6}_{+0.9}$	$\mathbf{31.6}_{+1.8}$	$\mathbf{35.7}_{+2.4}$	$\mathbf{21.9}_{+1.9}$

Table 3: Evaluation results with **Qwen3-235B-A22B-Instruct-2507** as teacher. Math results are reported as mean@32 accuracy (%). **Bold** indicates the best result within each student group.

Student	Method	Math			Code		Avg
		AIME24	AIME25	HMMT25	HE+	MBPP+	
Teacher Model		80.7	69.2	55.6	90.2	77.6	74.7
Qwen3-30B-A3B	Base	28.4	23.4	15.2	77.8	69.5	42.9
	OPD	69.5	56.7	38.4	82.1	71.3	63.6
	IW-OPD	$\mathbf{70.8}_{+1.3}$	$\mathbf{58.9}_{+2.2}$	$\mathbf{40.5}_{+2.1}$	$\mathbf{83.5}_{+1.4}$	$\mathbf{73.7}_{+2.4}$	$\mathbf{65.5}_{+1.9}$

We empirically compare both the original (Eq. 12) and unsigned versions in Fig. 4c, both variants are improve in practice, while the unsigned version yields higher p better results.

III. Normalization. Although sign correction ensures monotonicity, the absolute scale can still vary significantly across responses. For each sample, the discrepancy lies in $[d_T, 0]$, where the maximum 0 corresponds to the beginning of the sequence and the minimum d_T to the end. We map this range to $[0, 1]$ through within-sample normalization:

$$\tilde{r}_t^{\text{norm}} = \frac{d_t - d_{\min}}{d_{\max} - d_{\min}} = \frac{-\sum_{k<t} |A_k^{\text{OPD}}| - (-\sum_{k<T} |A_k^{\text{OPD}}|)}{0 - (-\sum_{k<T} |A_k^{\text{OPD}}|)} = 1 - \frac{\sum_{k<t} |A_k^{\text{OPD}}|}{\sum_{k<T} |A_k^{\text{OPD}}|}. \tag{14}$$

IV. Interpolation with OPD. After normalization, end-of-sequence tokens have low weights. We interpolate with standard OPD to retain dense supervision while allocating extra weight to compatible prefixes:

$$\tilde{r}_t^{\text{IW-OPD}} = 1 + \gamma \cdot \tilde{r}_t^{\text{norm}} = 1 + \gamma \left(1 - \frac{\sum_{k<t} |A_k^{\text{OPD}}|}{\sum_{k<T} |A_k^{\text{OPD}}|}\right), \tag{15}$$

Here $\gamma \geq 0$ controls the contribution of the teacher-informed weights; $\gamma = 0$ recovers OPD and $\gamma = 0.5$ is the default. Alg. 1 and App. C.2 provide the update procedure and training settings.

Table 4: **Ablation results on AIME25.** We isolate trajectory-adaptive prefix selection, unsigned discrepancy, and the effect of using the practical surrogate with a standard OPD blend.

Variant	AIME25	Δ vs. OPD baseline
Prefix selection strategy		
Standard OPD	43.3	0.0
Amplify fixed prefix	43.7	+0.4
Linear decay	44.1	+0.7
Manual curriculum	44.8	+1.5
Cumulative-share (ours)	48.9	+5.6
Signed vs. unsigned discrepancy		
Signed $\sum A_k$	45.9	+2.6
Unsigned $\sum \lvert A_k \rvert$ (ours)	48.9	+5.6
Weight source and OPD blend		
Ideal weight only	42.1	-1.2
Ideal weight w. OPD blend	43.9	+0.6
Surrogate weight only	46.2	+2.9
OPD blend (ours)	48.9	+5.6

5 Experiments

5.1 Setup

We evaluate IW-OPD in two teacher regimes and three student scales. The students are Qwen3-4B, Qwen3-1.7B, and Qwen3-0.6B. The first teacher is Qwen3-4B-Instruct-2507, which gives a larger overlap setting within the same model family. The second teacher is Qwen3-30B-A3B-Instruct-2507, which gives a smaller overlap setting and tests whether prefix weighting remains useful when the student often leaves the teacher's preferred trajectory region.

Training uses DeepMath with difficulty at least 6, about 57K problems, for math, and Eurus-RL-Code, about 25K problems, for code. We evaluate math on AIME 2024, AIME 2025, and HMMT 2025. We evaluate code on HumanEval+ and MBPP+. The baselines are the student base model before distillation and standard OPD. Results in the main experiments are averaged over three random seeds.

5.2 Main Results

IW-OPD improves performance and sample efficiency. Tabs. 1 and 2 show improvements over OPD across all evaluated teacher–student pairs and benchmarks, both at step 10 and at convergence. With the 30B-A3B teacher and 4B student, IW-OPD$_{10}$ raises the average from 52.0 to 55.5 and AIME25 from 42.4 to 49.3, already matching the final OPD average of 55.3. This supports allocating more of each update to teacher-compatible prefixes. The larger-scale 235B-A22B teacher / 30B-A3B student experiment also improves performance, raising the math average by 1.9 points (Tab. 3). App. E further shows that IW-OPD can be combined with reward redesign in ExOPD (Yang et al., 2026). Further LiveCodeBench, BrowseComp, and scaling results appear in App. D.

IW-OPD makes stronger teachers more sample-efficient. Fixing the 4B student isolates teacher scaling. A 30B-A3B teacher gives a higher final OPD average than the 4B teacher (55.3 versus 54.3), but a lower step-10 average (52.0 versus 54.0). IW-OPD reverses this early disadvantage: the stronger teacher reaches 55.5 versus 55.1. On AIME25, the corresponding comparison changes from 42.4 versus 45.7 with OPD$_{10}$ to 49.3 versus 46.7 with IW-OPD$_{10}$. Prefix weighting thus helps the student benefit earlier from stronger teacher supervision.

IW-OPD benefits smaller students and scales to larger models. With the 4B-Instruct teacher fixed, final average gains grow from +1.0 to +1.2 and +1.9 points as students shrink from 4B to 1.7B and 0.6B, corresponding to relative gains of approximately 1.8%, 3.2%, and 9.5%. Relative step-10 gains also grow, from 2.0% to 3.5% and 10.8%. This supports the role of prefix weighting when teacher–student mismatch increases. Its benefit extends to a capable 30B student: the 235B-teacher experiment improves AIME24, AIME25, and HMMT25 by 1.3, 2.2, and 2.1 points, respectively.

5.3 Ablations

Tab. 4 isolates three design choices: how token weights are assigned, how prefix discrepancy is measured, and whether the weighted term should be blended with standard OPD. We use Qwen3-0.6B as the student because this setting makes prefix selection most visible.

Trajectory-adaptive selection. Amplifying a fixed ratio (30%) prefix gives only $+0.4$, and a hand-designed position schedule gives $+1.5$. After unsigned correction, w_t becomes a monotonically decreasing weight. Accordingly, we test a direct linear-decay variant, which gives only $+0.7$. The cumulative-share rule gives $+5.6$. This ordering separates the benefit of early token preference from the benefit of trajectory adaptivity. A smooth preference for earlier positions is not enough. The useful boundary between reliable and unreliable prefixes changes across rollouts, so the weight must follow each trajectory's own discrepancy trace. Easy rollouts can keep high weights for longer, while hard rollouts should reduce downstream weights earlier.

Unsigned discrepancy. Using signed accumulation gives $+2.6$, while the unsigned version gives $+5.6$. Signed terms can cancel even when the prefix has passed through several model disagreements. This cancellation makes a drifted prefix appear compatible with the teacher. The absolute statistic treats each disagreement as evidence that the student has moved away from the shared prefix region.

Preserving the OPD blend. The ideal likelihood-ratio weight alone underperforms OPD, and blending it with OPD gives only a small gain. This matches the instability observed in Fig. 4: the raw ratio has high variance and is sensitive to signed cancellations along long trajectories. The practical surrogate is more robust because it preserves the desired ordering of prefixes while removing the unstable product scale. However, using the surrogate alone is still weaker than blending it with OPD. The blend keeps the standard dense OPD signal as a floor and allocates additional update budget to compatible prefixes, which is the intended role of IW-OPD.

6 Related Work

On-policy and token-selective distillation. Classical KD (Hinton et al., 2015) trains on teacher-generated data and can suffer from exposure bias (Bengio et al., 2015; Arora et al., 2022; Ross et al., 2011). OPD supervises student-sampled rollouts: GKD (Agarwal et al., 2024) unifies on/off-policy data through f-divergences, MiniLLM (Gu et al., 2024) uses reverse-KL policy gradients, and subsequent work develops related extensions (Li et al., 2026; Song & Zheng, 2026; Yang et al., 2026; Zhao et al., 2026a). Selective distillation uses sequence curricula (Wen et al., 2023; Liu & Zhang, 2025; Xu et al., 2026) or token weights based on frequency, difficulty, teacher confidence, and student learning state (Gu et al., 2020; Liang et al., 2021; Huang et al., 2025; Xie et al., 2025; Kim & Baek, 2026). IW-OPD weights tokens by on-policy prefix compatibility.

Credit assignment and reweighted policy updates. RLVR (DeepSeek-AI, 2025; Shao et al., 2024) assigns sparse outcome rewards across reasoning traces. Process supervision provides denser step-level feedback (Lightman et al., 2024; Wang et al., 2024; Cui et al., 2025); token-level analyses highlight high-entropy forking, critical, and reasoning rather than boilerplate tokens (Wang et al., 2025; Bigelow et al., 2025; Lin et al., 2025; Vassoyan et al., 2025; Ye et al., 2025; He et al., 2026). IW-OPD allocates dense teacher supervision by prefix compatibility. Its constrained-projection view connects to trust-region and proximal updates (Kakade, 2001; Schulman et al., 2015; 2017), sequence-level importance correction in GSPO and online DPO (Zheng et al., 2025; Rafailov et al., 2023; Xie et al., 2024), and geometric interpolation/Rényi midpoints (van Erven & Harremoës, 2014; Zhao et al., 2026b).

7 Discussion and Conclusion

We identify position bias as a key inefficiency in OPD and explain it through a finite-budget local projection view. IW-OPD uses cumulative prefix discrepancy to allocate extra gradient budget to teacher-compatible prefixes while preserving standard OPD as the dense supervision floor. Experiments across same-family, cross-scale, and stronger-teacher settings demonstrate improved sample efficiency and final performance. Effective on-policy supervision should therefore account for both token-level disagreement and the trajectory context in which it occurs. App. G discusses the broader relevance to methods involving two autoregressive policies.

AI USE STATEMENT

In preparing this paper, we used large language models to assist with literature retrieval and discovery, including identifying relevant prior work, and to draft selected sections of the manuscript. We also used AI assistance to revise and condense the text, organize the manuscript, and adapt its LaTeX layout to the conference template. The authors take responsibility for the final content of the paper, including its text, citations, claims, experimental results, and conclusions.

ETHICS STATEMENT

This study uses only the publicly available models and datasets described in the paper. It does not involve human participants, human annotation, or personal, sensitive, or non-public data.

REPRODUCIBILITY STATEMENT

The theoretical derivations are provided in App. A. Alg. 1 in App. B describes the IW-OPD training procedure. App. C reports the models and datasets, training hyperparameters, and evaluation protocols. We will publicly release the code and scripts after acceptance.

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. URL https://openreview.net/forum?id=3zKtaqxLhW.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 700–710, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 2015.

Eric J. Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. Forking paths in neural text generation. In *International Conference on Learning Representations*, 2025.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456v2*, 2025. doi: 10.48550/arXiv.2502.01456. URL https://arxiv.org/abs/2502.01456v2.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Shuhao Gu, Jinchao Zhang, Fandong Meng, Yang Feng, Wanying Xie, Jie Zhou, and Dong Yu. Token-level adaptive training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1035–1046, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.76. URL https://aclanthology.org/2020.emnlp-main.76/.

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

Yuhang He, Haodong Wu, Siyi Liu, Hongyu Ge, Hange Zhou, Keyi Wu, Zhuo Zheng, Qihong Lin, Zixin Zhong, and Yongqi Zhang. Where hindsight credit can reside: A signed-capacity view of token updates in RLVR. *arXiv preprint arXiv:2604.11056v2*, 2026. doi: 10.48550/arXiv.2604.11056. URL https://arxiv.org/abs/2604.11056v2.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. doi: 10.48550/arXiv.1503.02531. URL https://arxiv.org/abs/1503.02531.

Haiduo Huang, Jiangcheng Song, Yadong Zhang, and Pengju Ren. SelecTKD: Selective token-weighted knowledge distillation for LLMs, 2025. URL https://arxiv.org/abs/2510.24021.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. URL https://arxiv.org/abs/2403.07974.

Sham M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, pp. 1531–1538, 2001.

Minsang Kim and Seung Jun Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zph7e5JaXc.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026. doi: 10.48550/arXiv.2604.13016. URL https://arxiv.org/abs/2604.13016.

Chen Liang, Haoming Jiang, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, and Tuo Zhao. Token-wise curriculum learning for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3658–3670, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.310. URL https://aclanthology.org/2021.findings-emnlp.310/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. Critical tokens matter: Token-level contrastive estimation enhances LLM's reasoning capability. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37906–37918. PMLR, 2025. URL https://proceedings.mlr.press/v267/lin25j.html.

Lingyuan Liu and Mengxiang Zhang. Being strong progressively! enhancing knowledge distillation of large language models through a curriculum learning framework. *arXiv preprint arXiv:2506.05695*, 2025.

LLM-Core Xiaomi. MiMo-V2-Flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 627–635, 2011.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Tim van Erven and Peter Harremoës. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

Jean Vassoyan, Nathanaël Beau, and Roman Plaud. Ignore the KL penalty! boosting exploration on critical tokens to enhance RL fine-tuning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6123–6133, 2025.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems*, 2025.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025. URL https://arxiv.org/abs/2504.12516.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10817–10834, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.605. URL https://aclanthology.org/2023.acl-long.605/.

Tengyang Xie, Dylan J. Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit Q*-approximation for sample-efficient RLHF. *arXiv preprint arXiv:2405.21046*, 2024.

Xurong Xie, Zhucun Xue, Jiafu Wu, Jian Li, Yabiao Wang, Xiaobin Hu, Yong Liu, and Jiangning Zhang. LLM-oriented token-adaptive knowledge distillation, 2025. URL https://arxiv.org/abs/2510.11615.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. PACED: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*, 2026.

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

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

Ziang Ye, Zhenru Zhang, Yang Zhang, Jianxin Ma, Junyang Lin, and Fuli Feng. Disentangling reasoning tokens and boilerplate tokens for language model fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20939–20957. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1078.

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*, 2026a.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. Geometric-mean policy optimization. In *International Conference on Learning Representations*, 2026b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. doi: 10.48550/arXiv.2507.18071. URL https://arxiv.org/abs/2507.18071.

A Proofs and Derivations

The derivations below fix a prompt x unless otherwise stated and omit the conditioning on x for notational simplicity. We write π_T for the teacher next-token policy and also for its autoregressive trajectory distribution, as in the main text. The local projected distribution is denoted by $q_\theta^\star$, and the corresponding causal prefix weight is $w_t^\star$. All distributions are understood to be supported on the common support where the relevant KL divergences are finite.

A.1 Solution of the Constrained Projection

We prove Prop. 1. The constrained projection is

$$q_\theta^\star = \arg\min_q D_{\mathrm{KL}}(q\|\pi_T) \quad \text{s.t.} \quad D_{\mathrm{KL}}(q\|\pi_\theta) \le \rho, \tag{16}$$

together with the normalization constraint $\sum_y q(y) = 1$.

If the trust-region constraint were inactive, the solution would be the unconstrained minimizer $q = \pi_T$. This is infeasible when $0 < \rho < D_{\mathrm{KL}}(\pi_T\|\pi_\theta)$, so the constraint must be active in the local-update regime. Since $D_{\mathrm{KL}}(q\|\pi_T)$ is strictly convex in q on the common support and the KL ball is convex, the KKT conditions identify the unique optimum.

Introduce the Lagrangian

$$\mathcal{L}(q,\lambda,\mu) = D_{\mathrm{KL}}(q\|\pi_T) + \lambda\left(D_{\mathrm{KL}}(q\|\pi_\theta) - \rho\right) + \mu\left(\sum_y q(y) - 1\right), \tag{17}$$

where $\lambda \ge 0$ is the multiplier for the trust-region constraint and μ is the multiplier for normalization. Expanding the KL terms gives

$$\mathcal{L}(q,\lambda,\mu) = \sum_y q(y)\log\frac{q(y)}{\pi_T(y)} + \lambda\sum_y q(y)\log\frac{q(y)}{\pi_\theta(y)} - \lambda\rho + \mu\left(\sum_y q(y) - 1\right). \tag{18}$$

Taking the functional derivative with respect to $q(y)$ and setting it to zero yields

$$\begin{aligned} 0 &= \frac{\partial\mathcal{L}}{\partial q(y)} \\ &= \log q(y) - \log\pi_T(y) + 1 + \lambda\left(\log q(y) - \log\pi_\theta(y) + 1\right) + \mu. \end{aligned} \tag{19}$$

Rearranging terms,

$$(1+\lambda)\log q(y) = \log\pi_T(y) + \lambda\log\pi_\theta(y) + \text{const}, \tag{20}$$

where the constant absorbs $1 + \lambda + \mu$ and is independent of y. Therefore,

$$\log q(y) = \frac{1}{1+\lambda}\log\pi_T(y) + \frac{\lambda}{1+\lambda}\log\pi_\theta(y) + \text{const}. \tag{21}$$

Define

$$\alpha \coloneqq \frac{1}{1+\lambda} \in (0,1), \qquad 1-\alpha = \frac{\lambda}{1+\lambda}. \tag{22}$$

Exponentiating and normalizing gives

$$q_\alpha(y) = \frac{\pi_\theta(y)^{1-\alpha}\pi_T(y)^\alpha}{Z_\alpha(\theta)}, \qquad Z_\alpha(\theta) = \sum_y \pi_\theta(y)^{1-\alpha}\pi_T(y)^\alpha. \tag{23}$$

Equivalently, with

$$r_\theta(y) = \frac{\pi_T(y)}{\pi_\theta(y)}, \qquad Z_\alpha(\theta) = \mathbb{E}_{y\sim\pi_\theta}\left[r_\theta(y)^\alpha\right], \tag{24}$$

we can write

$$q_\alpha(y) = \frac{\pi_\theta(y) r_\theta(y)^\alpha}{Z_\alpha(\theta)}. \tag{25}$$

It remains to identify the value of α induced by the radius ρ. Let

$$\psi(\alpha) \coloneqq \log Z_\alpha(\theta). \tag{26}$$

Then

$$\psi'(\alpha) = \mathbb{E}_{q_\alpha}\left[\log r_\theta(y)\right]. \tag{27}$$

Therefore,

$$\begin{aligned} D_{\mathrm{KL}}(q_\alpha \| \pi_\theta) &= \mathbb{E}_{q_\alpha}\left[\log \frac{q_\alpha(y)}{\pi_\theta(y)}\right] \\ &= \mathbb{E}_{q_\alpha}\left[\alpha \log r_\theta(y) - \psi(\alpha)\right] \\ &= \alpha\psi'(\alpha) - \psi(\alpha). \end{aligned} \tag{28}$$

Since the constraint is active, α is determined by

$$\rho = D_{\mathrm{KL}}(q_\alpha \| \pi_\theta) = \alpha \frac{\partial}{\partial \alpha} \log Z_\alpha(\theta) - \log Z_\alpha(\theta), \tag{29}$$

This is the implicit relation for α. Also,

$$\begin{aligned} \frac{d}{d\alpha} D_{\mathrm{KL}}(q_\alpha \| \pi_\theta) &= \alpha\psi''(\alpha) \\ &= \alpha \operatorname{Var}_{q_\alpha}\left[\log r_\theta(y)\right] \geq 0. \end{aligned} \tag{30}$$

Thus increasing ρ increases the corresponding interpolation coefficient α whenever $\pi_T \neq \pi_\theta$. Finally, $D_{\mathrm{KL}}(q_0 \| \pi_\theta) = 0$ and $q_1 = \pi_T$, so $D_{\mathrm{KL}}(q_1 \| \pi_\theta) = D_{\mathrm{KL}}(\pi_T \| \pi_\theta)$. Hence each $0 < \rho < D_{\mathrm{KL}}(\pi_T \| \pi_\theta)$ induces $\alpha \in (0, 1)$ and

$$q_\theta^\star(y) = q_\alpha(y) = \frac{\pi_\theta(y) r_\theta(y)^\alpha}{Z_\alpha(\theta)}. \tag{31}$$

This is Eq. (6).

When the budget is large enough that the teacher itself is feasible, the constraint can be inactive, $\lambda = 0$, and $\alpha = 1$, which recovers $q_\theta^\star = \pi_T$. □

A.2 Derivation of the Importance-Weighted OPD Objective

We prove Prop. 2. Fix a prompt x and let $h_t = (x, y_{<t})$. We keep the conditioning on x explicit in this appendix. We work in the non-trivial local-update regime of Prop. 1, where $0 < \rho < D_{\mathrm{KL}}(\pi_T \| \pi_\theta)$ and the trust-region constraint is active:

$$D_{\mathrm{KL}}\left(q_\theta^\star \| \pi_\theta\right) = \rho. \tag{32}$$

The projected objective in Eq. (7) is

$$\mathcal{J}_{q_\theta^\star}(\theta; x) = \max_{q_\theta^\star} -D_{\mathrm{KL}}\left(q_\theta^\star \| \pi_T\right). \tag{33}$$

By adding and subtracting $\log \pi_\theta(y \mid x)$ inside the KL, we obtain

$$\begin{aligned} D_{\mathrm{KL}}\left(q_\theta^\star \| \pi_T\right) &= \mathbb{E}_{y \sim q_\theta^\star(\cdot|x)}\left[\log \frac{q_\theta^\star(y \mid x)}{\pi_T(y \mid x)}\right] \\ &= \mathbb{E}_{y \sim q_\theta^\star(\cdot|x)}\left[\log \frac{q_\theta^\star(y \mid x)}{\pi_\theta(y \mid x)}\right] + \mathbb{E}_{y \sim q_\theta^\star(\cdot|x)}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_T(y \mid x)}\right] \\ &= \rho + \mathbb{E}_{y \sim q_\theta^\star(\cdot|x)}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_T(y \mid x)}\right]. \end{aligned} \tag{34}$$

Thus, under a fixed local-update budget ρ, minimizing the projected KL is equivalent up to the constant ρ to minimizing the second term in Eq. (34).

Following Eq. (6), write the trajectory-level likelihood ratio as

$$r_\theta(y \mid x) = \frac{\pi_T(y \mid x)}{\pi_\theta(y \mid x)}, \qquad Z_\alpha(\theta, x) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}\left[r_\theta(y \mid x)^\alpha\right]. \tag{35}$$

Then the optimal projected policy can be written as

$$q_\theta^\star(y \mid x) = \frac{\pi_\theta(y \mid x) r_\theta(y \mid x)^\alpha}{Z_\alpha(\theta, x)}, \qquad \alpha \in (0, 1). \tag{36}$$

Therefore,

$$\frac{q_\theta^\star(y \mid x)}{\pi_\theta(y \mid x)} = \frac{r_\theta(y \mid x)^\alpha}{Z_\alpha(\theta, x)}. \tag{37}$$

For any measurable function f, this gives the change-of-measure identity

$$\mathbb{E}_{y \sim q_\theta^\star(\cdot|x)}[f(y)] = \frac{\mathbb{E}_{y \sim \pi_\theta(\cdot|x)}\left[r_\theta(y \mid x)^\alpha f(y)\right]}{Z_\alpha(\theta, x)}. \tag{38}$$

By the autoregressive factorization,

$$\begin{aligned} \log \frac{\pi_\theta(y \mid x)}{\pi_T(y \mid x)} &= \log \frac{\prod_{t=1}^T \pi_\theta(y_t \mid h_t)}{\prod_{t=1}^T \pi_T(y_t \mid h_t)} \\ &= \sum_{t=1}^T \log \frac{\pi_\theta(y_t \mid h_t)}{\pi_T(y_t \mid h_t)}. \end{aligned} \tag{39}$$

Substituting Eq. (39) into Eq. (38) gives the exact trajectory-level importance-weighted form of the non-constant part of Eq. (34):

$$\widetilde{\mathcal{J}}_{q_\theta^\star}(\theta; x) \coloneqq -\left(D_{\mathrm{KL}}\left(q_\theta^\star \| \pi_T\right) - \rho\right) = -\frac{\mathbb{E}_{y \sim \pi_\theta(\cdot|x)}\left[r_\theta(y \mid x)^\alpha \sum_{t=1}^T \log \frac{\pi_\theta(y_t|h_t)}{\pi_T(y_t|h_t)}\right]}{Z_\alpha(\theta, x)}. \tag{40}$$

Eq. (40) is a sequence-level expression: every token term in the same sampled trajectory is multiplied by the full trajectory ratio $r_\theta(y \mid x)^\alpha$. However, OPD is optimized through a token-local semi-gradient on student-sampled prefixes. Hence the coefficient assigned to the token term at position t should depend only on the causal prefix $h_t = (x, y_{<t})$, rather than on the future suffix $y_{\geq t}$.

To obtain the causal token-level surrogate, decompose the trajectory ratio as

$$\begin{aligned} r_\theta(y \mid x)^\alpha &= \left(\frac{\pi_T(y \mid x)}{\pi_\theta(y \mid x)}\right)^\alpha \\ &= \left(\frac{\pi_T(y_{<t} \mid x)}{\pi_\theta(y_{<t} \mid x)}\right)^\alpha \left(\frac{\pi_T(y_{\geq t} \mid x, y_{<t})}{\pi_\theta(y_{\geq t} \mid x, y_{<t})}\right)^\alpha. \end{aligned} \tag{41}$$

The first factor is the prefix likelihood ratio inherited from the trajectory-level ratio in Prop. 1. We write

$$\begin{aligned} r_t(y_{<t} \mid x) \coloneqq r_\theta(y_{<t} \mid x) &= \frac{\pi_T(y_{<t} \mid x)}{\pi_\theta(y_{<t} \mid x)} \\ &= \prod_{k=1}^{t-1} \frac{\pi_T(y_k \mid x, y_{<k})}{\pi_\theta(y_k \mid x, y_{<k})}. \end{aligned} \tag{42}$$

The empty product is 1, so $r_1 = 1$. The position-wise normalizer is

$$Z_{\alpha,t}(\theta, x) \coloneqq \mathbb{E}_{y_{<t} \sim \pi_\theta(\cdot|x)}\left[r_t(y_{<t} \mid x)^\alpha\right], \tag{43}$$

and the normalized prefix ratio is

$$\tilde{r}_t(y_{<t} \mid x) \coloneqq \frac{r_t(y_{<t} \mid x)^\alpha}{Z_{\alpha,t}(\theta, x)}. \tag{44}$$

Replacing the full trajectory ratio in Eq. (40) with its causal prefix component yields the token-level IW-OPD surrogate for a fixed prompt:

$$\mathcal{J}_{\mathrm{IW}}^\star(\theta; x) = -\mathbb{E}_{y \sim \pi_\theta(\cdot|x)}\left[\sum_{t=1}^T \mathrm{sg}\left[\tilde{r}_t(y_{<t} \mid x)\right] \log \frac{\pi_\theta(y_t \mid h_t)}{\pi_T(y_t \mid h_t)}\right]. \tag{45}$$

Here $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. It makes the normalized prefix ratio act as a detached multiplicative coefficient on the standard token-level log-ratio term.

Averaging Eq. (45) over $x \sim \mathcal{D}$ gives

$$\mathcal{J}^{\star}_{\mathrm{IW}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \mathrm{sg}\left[\tilde{r}_t(y_{<t} \mid x)\right] \log \frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi_T(y_t \mid x, y_{<t})}\right]. \tag{46}$$

Suppressing the fixed prompt x in the notation recovers Eq. (9). □

A.3 Standard OPD Chain Rule and Single-Step Semi-Gradient

We spell out the chain-rule decomposition of standard OPD and the single-step semi-gradient used by Eq. (10). Fix a prompt x and write $h_t = (x, y_{<t})$. We keep the conditioning on x explicit in this appendix, while the main text suppresses it when no ambiguity arises.

The per-prompt reverse-KL quantity minimized by standard OPD is

$$J_{\mathrm{OPD}}(\theta; x) = \max_{\theta} -D_{\mathrm{KL}}\left(\pi_\theta \| \pi_T\right) = -\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_T(y \mid x)}\right]. \tag{47}$$

By the autoregressive factorization,

$$\log \frac{\pi_\theta(y \mid x)}{\pi_T(y \mid x)} = \sum_{t=1}^{T} \log \frac{\pi_\theta(y_t \mid h_t)}{\pi_T(y_t \mid h_t)}. \tag{48}$$

Therefore,

$$\begin{aligned} D_{\mathrm{KL}}\left(\pi_\theta \| \pi_T\right) &= \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \log \frac{\pi_\theta(y_t \mid h_t)}{\pi_T(y_t \mid h_t)}\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{y_{<t}\sim\pi_\theta(\cdot|x)}\left[\mathbb{E}_{y_t\sim\pi_\theta(\cdot|h_t)}\left[\log \frac{\pi_\theta(y_t \mid h_t)}{\pi_T(y_t \mid h_t)}\right]\right] \\ &= \sum_{t=1}^{T} \mathbb{E}_{y_{<t}\sim\pi_\theta(\cdot|x)}\left[D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid h_t) \| \pi_T(\cdot \mid h_t)\right)\right]. \end{aligned} \tag{49}$$

Next consider one local next-token KL at a fixed prefix h_t:

$$D_t(\theta; h_t) = D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid h_t) \| \pi_T(\cdot \mid h_t)\right). \tag{50}$$

Expanding over the next-token vocabulary gives

$$D_t(\theta; h_t) = \sum_{a} \pi_\theta(a \mid h_t)\left[\log \pi_\theta(a \mid h_t) - \log \pi_T(a \mid h_t)\right]. \tag{51}$$

Differentiating this local KL while holding the sampled prefix h_t fixed yields

$$\begin{aligned} \nabla_\theta D_t(\theta; h_t) &= \sum_{a} \nabla_\theta \pi_\theta(a \mid h_t)\left[\log \pi_\theta(a \mid h_t) - \log \pi_T(a \mid h_t) + 1\right] \\ &= \mathbb{E}_{a\sim\pi_\theta(\cdot|h_t)}\left[\left(\log \frac{\pi_\theta(a \mid h_t)}{\pi_T(a \mid h_t)} + 1\right) \nabla_\theta \log \pi_\theta(a \mid h_t)\right]. \end{aligned} \tag{52}$$

The $+1$ term vanishes because

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|h_t)}\left[\nabla_\theta \log \pi_\theta(a \mid h_t)\right] = \nabla_\theta \sum_{a} \pi_\theta(a \mid h_t) = 0. \tag{53}$$

Thus,

$$\nabla_\theta D_t(\theta; h_t) = \mathbb{E}_{a\sim\pi_\theta(\cdot|h_t)}\left[\log \frac{\pi_\theta(a \mid h_t)}{\pi_T(a \mid h_t)} \nabla_\theta \log \pi_\theta(a \mid h_t)\right]. \tag{54}$$

Using the OPD advantage notation in Eq. (4), for a sampled token $a = y_t$ we have

$$A_t^{\mathrm{OPD}} := -\left(\log \pi_\theta(y_t \mid h_t) - \log \pi_T(y_t \mid h_t)\right) = -\log \frac{\pi_\theta(y_t \mid h_t)}{\pi_T(y_t \mid h_t)}. \tag{55}$$

Therefore, the negative local KL gradient, written in policy-gradient ascent form, is

$$-\nabla_\theta D_t(\theta; h_t) = \mathbb{E}_{y_t \sim \pi_\theta(\cdot \mid h_t)} \left[A_t^{\mathrm{OPD}} \nabla_\theta \log \pi_\theta(y_t \mid h_t) \right]. \tag{56}$$

The word "semi-gradient" is important. The chain-rule decomposition in Eq. (49) is exact, but the token-local update in Eq. (56) treats the sampled prefix $y_{<t}$ as fixed context. A full score-function gradient of the sequence-level reverse KL would instead couple all positions:

$$\nabla_\theta D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid x) \| \pi_T(\cdot \mid x)\right) = \mathbb{E}_{y \sim \pi_\theta(\cdot \mid x)} \left[\left(\sum_{k=1}^{T} \log \frac{\pi_\theta(y_k \mid h_k)}{\pi_T(y_k \mid h_k)} \right) \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(y_t \mid h_t) \right) \right]. \tag{57}$$

Standard OPD practice instead uses the single-step token-local update direction. With the sign convention of Eq. (4), this gives

$$\nabla_\theta J_{\mathrm{OPD}}(\theta) \approx \mathbb{E}_{x \sim \mathcal{D},\, y \sim \pi_\theta(\cdot \mid x)} \left[\sum_{t=1}^{T} A_t^{\mathrm{OPD}} \nabla_\theta \log \pi_\theta(y_t \mid h_t) \right], \tag{58}$$

which is the standard OPD form in the main text.

We now apply the same single-step rule to the ideal IW-OPD surrogate in Prop. 2. For a fixed prompt, Eq. (9) can be written as

$$J_{\mathrm{IW}}^\star(\theta; x) = -\mathbb{E}_{y \sim \pi_\theta(\cdot \mid x)} \left[\sum_{t=1}^{T} \mathrm{sg}\left[\tilde{r}_t(y_{<t} \mid x)\right] \log \frac{\pi_\theta(y_t \mid h_t)}{\pi_T(y_t \mid h_t)} \right], \tag{59}$$

where the notation follows the main text:

$$r_t := r_\theta(y_{<t} \mid x) = \frac{\pi_T(y_{<t} \mid x)}{\pi_\theta(y_{<t} \mid x)}, \qquad Z_{\alpha,t}(\theta, x) = \mathbb{E}_{y_{<t} \sim \pi_\theta(\cdot \mid x)}\left[r_t\right], \qquad \tilde{r}_t = \frac{r_t}{Z_{\alpha,t}}. \tag{60}$$

When differentiating the local next-token term at position t, all prefix-determined quantities are treated as fixed context. This includes the sampled prefix $y_{<t}$, the prefix ratio r_t, the normalizer $Z_{\alpha,t}$, and the normalized prefix ratio $\tilde{r}_t$. The stop-gradient operator in Eq. (59) enforces exactly this convention.

Detaching the prefix ratio is necessary for preserving the single-step OPD semi-gradient. Indeed, for fixed α,

$$\begin{aligned} \log r_t &= \alpha \sum_{k<t} \left(\log \pi_T(y_k \mid h_k) - \log \pi_\theta(y_k \mid h_k)\right) \\ &= \alpha \sum_{k<t} A_k^{\mathrm{OPD}}. \end{aligned} \tag{61}$$

Since the teacher is fixed, differentiating through the ratio would introduce prefix score terms:

$$\nabla_\theta \log r_t = -\alpha \sum_{k<t} \nabla_\theta \log \pi_\theta(y_k \mid h_k). \tag{62}$$

Such terms couple the token-t update to earlier sampled actions and recover a sequence-level credit-assignment estimator rather than the token-local OPD semi-gradient. Differentiating $Z_{\alpha,t}$ would similarly require gradients through both the prefix sampling distribution and the prefix likelihood ratio. Therefore, both r_t and $Z_{\alpha,t}$ are detached in the single-step update.

With these detached prefix quantities, the local IW-OPD update is simply the standard OPD update multiplied by the normalized prefix ratio. Thus,

$$\nabla_\theta J_{\mathrm{IW}}^\star(\theta) \approx \mathbb{E}_{x \sim \mathcal{D},\, y \sim \pi_\theta(\cdot \mid x)} \left[\sum_{t=1}^{T} A_t^{\text{IW-OPD}} \nabla_\theta \log \pi_\theta(y_t \mid h_t) \right], \tag{63}$$

where

$$A_t^{\text{IW-OPD}} = \text{sg}\left[\tilde{r}_t\right] A_t^{\text{OPD}} = -\text{sg}\left[\tilde{r}_t\right] \left(\log \pi_\theta(y_t \mid h_t) - \log \pi_T(y_t \mid h_t)\right). \quad (64)$$

This is the semi-gradient form of Eq. (10), with the stop-gradient convention inherited from Eq. (9). □

B Algorithm

For clipped PPO, let π_0 be the frozen rollout policy for the current batch, and use $\eta_t(\theta) = \pi_\theta(y_t \mid x, y_{<t})/\pi_0(y_t \mid x, y_{<t})$ for the PPO ratio. The inner update minimizes

$$\mathcal{L}_{\text{IW-OPD}}(\theta) = -\mathbb{E}_{x,y,t}\left[\min\left(\eta_t(\theta) A_t^{\text{IW-OPD}}, \text{clip}(\eta_t(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})\, A_t^{\text{IW-OPD}}\right)\right], \quad (65)$$

where the expectation is over valid response tokens.

Algorithm 1 IW-OPD: Importance-Weighted On-Policy Distillation

Require: Student π_θ, teacher π_T, prompt distribution $\mathcal{D}$, amplification γ (default 0.5), PPO clip ϵ_{clip}, stabilizer ε.

Initialize rollout policy $\pi_0 \leftarrow \pi_\theta$.
for each training iteration **do**
  Sample $x \sim \mathcal{D}$ and generate $y = (y_1, \ldots, y_T)$ from $\pi_0(\cdot \mid x)$.
  Cache $\ell_{0,t} \leftarrow \log \pi_0(y_t \mid x, y_{<t})$ and $\ell_{T,t} \leftarrow \log \pi_T(y_t \mid x, y_{<t})$ for all valid tokens t.
  Set $A_t^{\text{OPD}} \leftarrow \text{sg}[\ell_{T,t} - \ell_{0,t}]$ for all valid tokens t.
  Set $w_t \leftarrow 1 - \sum_{k<t} |A_k^{\text{OPD}}| / \sum_{k<T} |A_k^{\text{OPD}}|$ for all valid tokens t.
  Set $A_t^{\text{IW-OPD}} \leftarrow A_t^{\text{OPD}} + \gamma \cdot \text{sg}[w_t] A_t^{\text{OPD}}$ for all valid tokens t.
  for several PPO inner steps **do**
    Update θ by minimizing Eq. (65), using $\eta_t(\theta) = \exp(\log \pi_\theta(y_t \mid x, y_{<t}) - \ell_{0,t})$.
  end for
  Refresh rollout policy: $\pi_0 \leftarrow \pi_\theta$.
end for

C Experimental Setup and Hyperparameters

This section reports the implementation details used for the experiments in Sec. 5. All OPD variants are implemented in the same `verl`-based PPO training pipeline. The student samples responses on-policy; student and teacher log probabilities are then evaluated on the sampled response tokens. No learned reward model is used: the token-level OPD or IW-OPD advantages are passed directly into the clipped PPO surrogate.

C.1 Models and Data

For all Qwen3 models, we use the chat template with thinking disabled. Prompts longer than the context budget are filtered rather than truncated, and response tokens beyond the generated answer mask are excluded from all OPD and IW-OPD computations.

C.2 Training Hyperparameters

Epoch budgets follow the corresponding model-scale scripts and are held fixed across OPD and IW-OPD within each setting. All reported comparisons use the same data order and seed set across methods.

Table 5: Model and data configurations used in the main experiments.

Component	Configuration
Students	Qwen3-4B, Qwen3-1.7B, and Qwen3-0.6B.
Teachers	Qwen3-4B-Instruct-2507 for the large-overlap setting; Qwen3-30B-A3B-Instruct-2507 for the small-overlap setting.
Math training data	DeepMath problems filtered to difficulty level ≥ 6 (approximately 57K prompts).
Code training data	Eurus-RL-Code (approximately 25K prompts).
Validation during training	AIME 2024 and AIME 2025, evaluated every 10 training steps.
Final evaluation	AIME 2024, AIME 2025, and HMMT 2025 for math; HumanEval+ and MBPP+ for code.

Table 6: Default training hyperparameters. Unless noted otherwise, the same values are used for OPD and IW-OPD within each model–teacher setting.

Hyperparameter	Value
Training framework	`verl` PPO trainer with vLLM rollouts
Nodes / GPUs	4 nodes, 32 GPUs
Learning rate	1×10^{-5}
Training batch size	1024 prompts
PPO mini-batch size	1024
PPO micro-batch size	1 per GPU
PPO epochs per rollout batch	1
PPO clipping range	0.2
Dual-clip constant	3.0
Loss aggregation	Token mean
Maximum prompt length	2048 tokens
Maximum response length	16384 tokens
Rollout samples per prompt	1 during training
Training decoding	Temperature 1.0, top-p 1.0
Validation decoding	Temperature 1.0, top-p 1.0, 32 samples per prompt for math validation
Optimizer warmup	0 warmup ratio
Entropy coefficient	0
KL reward penalty	Disabled
Auxiliary KL loss	Low-variance KL form, coefficient 0
Rollout importance correction	Token-level correction with threshold 5.0
Checkpoint / evaluation frequency	Save every 10 steps; validate every 10 steps

C.3 Method-Specific Parameters

IW-OPD uses w_t as a detached extra-budget multiplier on top of standard OPD:

$$A_t^{\text{IW-OPD}} = A_t^{\text{OPD}} + \gamma \cdot \text{sg}\left[w_t\right] A_t^{\text{OPD}}. \tag{66}$$

where γ control the magnitude of weighting term. We use $\gamma = 0.5$ throughout the main IW-OPD experiments. The weight is detached from the computational graph. Setting $\gamma = 0$ exactly recovers standard OPD.

C.4 Evaluation Protocol

For math benchmarks, we generate 32 responses per problem with vLLM using temperature 1.0, top-p 1.0, maximum generation length 16384, and seed-matched sampling across methods. Each prompt appends the instruction: "Please reason step by step, and put your final answer within \boxed{}." We extract the final boxed answer and evaluate it with symbolic equivalence checking. Tables report the aggregation specified in their captions, e.g., best@32 or mean@32.

For code benchmarks, we use the EvalPlus evaluation suite for HumanEval+ and MBPP+. Greedy single-sample evaluation is used for the reported pass-rate results.

Checkpoints with subscript $_{10}$ in the main tables are evaluated at training step 10. Converged OPD and IW-OPD checkpoints are selected using the same validation protocol within each model–teacher setting and are then evaluated on the held-out math and code benchmarks.

D Additional Experimental Results

D.1 Code Generation on LiveCodeBench

We evaluate on LiveCodeBench v6 (Jain et al., 2024), using the same coding setup as in the main experiments. The teachers are Qwen3-4B-Instruct-2507 and Qwen3-30B-A3B-Instruct-2507, and the students are Qwen3-4B and Qwen3-1.7B. Tab. 7 reports results at training step 10 and at the final checkpoint.

Table 7: LiveCodeBench v6 results across teacher–student configurations. Scores are reported in percent. Δ denotes the IW-OPD score minus the OPD score at the same checkpoint, in percentage points. Model sizes abbreviate the Qwen3 configurations specified above.

Teacher	Student	Step 10			Final		
		OPD	IW-OPD	Δ	OPD	IW-OPD	Δ
4B	1.7B	16.3	**17.7**	+1.4	19.6	**21.7**	+2.1
4B	4B	30.0	**32.5**	+2.5	30.2	**32.5**	+2.3
30B	1.7B	15.5	**17.6**	+2.1	22.3	**24.9**	+2.6
30B	4B	27.4	**30.1**	+2.7	31.8	**34.7**	+2.9

IW-OPD improves over OPD in all four teacher–student configurations at both checkpoints. Final gains range from 2.1 to 2.9 percentage points, extending the coding results beyond HumanEval+ and MBPP+.

D.2 Larger-Scale Distillation

We use Qwen3-30B-A3B as the student and compare Qwen3-30B-A3B-Instruct-2507 and Qwen3-235B-A22B-Instruct-2507 as teachers. The experimental setup follows the main math and code experiments. Tab. 8 reports the average across AIME24, AIME25, HMMT25, HumanEval+, and MBPP+ at training step 10 and at the final checkpoint.

Table 8: Early-checkpoint and final performance in larger-scale distillation. Scores are Math & Code Avg (%). Δ is the IW-OPD gain over OPD at the same checkpoint, in percentage points. The final 235B-to-30B results correspond to Tab. 3.

Teacher	Student	Step 10			Final		
		OPD	IW-OPD	Δ	OPD	IW-OPD	Δ
30B	30B	59.8	**63.2**	+3.4	60.7	**63.2**	+2.5
235B	30B	56.3	**62.9**	+6.6	63.6	**65.5**	+1.9

IW-OPD improves the step-10 average by 3.4 and 6.6 points with the 30B and 235B teachers, respectively, and retains final gains of 2.5 and 1.9 points. These results complement the smaller-student experiments with both early and final improvements at the 30B student scale.

D.3 Agentic Reasoning on BrowseComp

We evaluate agentic reasoning on BrowseComp (Wei et al., 2025) with Qwen3.5 and Nemotron teacher–student configurations. Results are mean accuracies over three evaluation runs.

Qwen3.5. The teacher is Qwen3.5-35B-A3B trained on agentic data using GSPO (Zheng et al., 2025), and the student is Qwen3.5-9B. The agent follows a ReAct workflow with `Search`, `LinkReader`, and `LinkSummary` tools.

Nemotron. The teacher is OpenResearcher, obtained by further supervised fine-tuning of Nemotron-3-Nano-30B-A3B. The student is Nemotron-3-Nano-4B.

Table 9: BrowseComp results with Qwen3.5 and Nemotron teacher–student configurations. Accuracy (%) is averaged over three evaluation runs. Values in parentheses indicate the gain over OPD at the same checkpoint, in percentage points. Teacher and student scores are shown before distillation.

Model family	Before distillation		Step 10		Final	
	Teacher	Student	OPD	IW-OPD	OPD	IW-OPD
Qwen3.5	43.8	24.3	30.3	**33.0** (+2.7)	35.7	**38.1** (+2.4)
Nemotron	28.4	6.1	14.4	**18.1** (+3.7)	18.4	**21.6** (+3.2)

Tab. 9 shows gains at both the early and final checkpoints in both settings. Final accuracy improves by 2.4 points with Qwen3.5 and 3.2 points with Nemotron, extending the evaluation to agentic tasks and an additional model family.

E Combination experiments with other reward design methods

Table 10: Combination results with ExOPD; both methods are distilled from Qwen3-30B-A3B. Math results are reported as mean@32 accuracy (%). IW-ExOPD denotes the combination of IW-OPD and ExOPD. **Bold** indicates the best result within each student group.

Student	Method	Math			Code		Avg
		AIME24	AIME25	HMMT25	HE+	MBPP+	
Teacher Model		74.7	62.8	44.2	86.6	75.1	68.7
Qwen3-4B	Base	23.1	21.4	10.0	75.3	64.5	38.9
	OPD	55.3	48.0	27.1	77.2	69.1	55.3
	ExOPD	57.9	50.1	31.7	78.9	70.2	57.8
	IW-ExOPD	**59.4**$_{+1.5}$	**51.7**$_{+1.6}$	**32.0**$_{+0.3}$	**80.1**$_{+1.1}$	**71.0**$_{+0.8}$	**58.8**$_{+1.0}$
Qwen3-1.7B	Base	13.4	11.0	6.8	59.6	52.5	28.7
	OPD	34.6	28.7	15.5	64.6	53.7	39.4
	ExOPD	37.6	31.8	16.8	67.2	55.0	41.7
	IW-ExOPD	**38.9**$_{+1.3}$	**33.2**$_{+1.4}$	**18.3**$_{+1.5}$	**68.5**$_{+1.3}$	**57.4**$_{+2.4}$	**43.2**$_{+1.5}$

IW-OPD is orthogonal to reward design. IW-OPD changes how an already-computed OPD signal is allocated across positions, so it is not tied to a particular advantage or reward design. As one example, ExOPD (Yang et al., 2026) reformulates OPD as a reinforcement-learning problem with a KL constraint, separates out the reward term, and improves exploration by scaling that reward with a fixed hyperparameter λ. Since IW-OPD supplies prefix-level importance weights, we can apply the same idea on top of ExOPD by making the reward scale prefix-dependent. We call this combination IW-ExOPD. As shown in Tab. 10, IW-ExOPD improves over ExOPD for both Qwen3-4B and Qwen3-1.7B students. This suggests that prefix-level importance weighting is orthogonal to reward design, rather than being tied only to vanilla OPD.

F Limitations

At convergence, w_t approaches $1 - t/T$ rather than becoming perfectly uniform—a mild residual non-uniformity. Cumulative prefix discrepancy is a conservative prefix-compatibility proxy induced by the prefix likelihood-ratio principle, not an exact density-ratio correction. Experiments are conducted at the 4B student scale; validation at larger scales remains future work.

G BROADER DISCUSSION

The key insight of this work is that OPD supervision is not uniformly reliable along a student-generated trajectory. This creates a *position bias*: teacher supervision is often more useful near the beginning of the rollout than near the end. Such bias is not unique to OPD. It may also appear in many methods involving two autoregressive sequence models, such as on-policy RL where samples are drawn from the current policy but updates move toward a new policy. The issue is especially visible in OPD since the teacher–student distribution gap is not known in advance, so we cannot predefine where their trajectories remain compatible.