POSTERIOR-WEIGHTED ON-POLICY DISTILLATION: LEARNING WHEN TO TRUST THE TEACHER

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) trains a smaller student on its own rollouts, using token-level feedback from a stronger teacher. Yet it trusts the teacher equally on every rollout, and the task reward, when used, enters as a hand-tuned additive term that competes with distillation. We ask *when* teacher guidance should be trusted, and answer with a derivation rather than a heuristic: under a Boltzmann model of trajectory optimality, the principled per-trajectory weight is the group-normalized posterior $w_i = N \cdot \mathrm{softmax}(A/\beta')_i$ over group-relative advantages. Posterior-Weighted OPD (PW-OPD) thereby demotes the reward from a competing objective to a modulation of a fixed distillation budget—one new temperature, used untuned—and recovers standard OPD exactly when the reward carries no within-group information. We verify the premise directly: student–teacher divergence on high-advantage trajectories is $2.8\times$ that on low-advantage ones, in both the short- and long-context regimes. Distilling a 27B teacher into a 9B student, we find that in the long-context thinking regime the additive objective drives AIME accuracy from the untrained student's 0.90 down to 0.75–0.82; a reward-free control attributes the damage to the additive term. Posterior weighting preserves and marginally improves the student (0.90–0.92), and the same ordering—posterior above additive above uniform—replicates in a routed multi-teacher extension with a different student and five domain benchmarks.

1 INTRODUCTION

On-policy distillation (OPD) transfers a teacher's reasoning ability to a smaller student by giving token-level teacher feedback on sequences the student generates itself, removing the exposure bias of off-policy imitation (Agarwal et al., 2024; Gu et al., 2023); it is now a standard post-training recipe (Yang et al., 2025; Lu & Thinking Machines Lab, 2025; Sheng et al., 2025) with a growing literature (Song & Zheng, 2026).

Two structural choices limit what it can achieve. First, when task rewards are available, the standard way of combining them with distillation is additive: a PPO-style reward term is appended to the distillation loss with a hand-tuned global coefficient (Sheng et al., 2025; Xu et al., 2025), so the two terms compete as gradient sources on the same parameters regardless of the quality of individual rollouts. Second, distillation treats every rollout as equally deserving of guidance, although a student trajectory can drift into states the teacher rarely visits, where blind imitation reinforces failure modes. Concurrent work conditions distillation on rewards through mechanism-specific choices—a clipped likelihood-ratio objective (Zhang et al., 2026), a discrete keep-or-drop gate (Akhondzadeh et al., 2026), or domain-specific verifier scores (Ou et al., 2026)—but none grounds its weighting rule in a probabilistic account of *when* teacher guidance should be trusted.

We ask a sharper question: what is the principled form of the per-trajectory weight through which a task reward should modulate teacher guidance? Our answer is posterior weighting. Modeling trajectory optimality with a Boltzmann distribution and normalizing the resulting posterior within each prompt group, Posterior-Weighted On-Policy Distillation (PW-OPD) multiplies each trajectory's distillation loss by $w_i = N \cdot \mathrm{softmax}(A/\beta')_i$, where A is the group-relative advantage and β' a single temperature. The reward never appears as an additive term: it reallocates a fixed distillation budget across trajectories, concentrating teacher guidance where the reward indicates it will help, while degrading exactly to standard OPD whenever the reward carries no within-group information.

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.

Our contributions are:

1. We derive the per-trajectory weight from a Boltzmann model of trajectory optimality, showing that the group-softmax form emerges from the posterior and enjoys shift invariance, budget preservation, and graceful degeneration to uniform weights (§2).
2. We realize the weighting as a lightweight modification of the OPD objective—a per-step group normalization of already-computed advantages—and extend it to the token axis with a mean-preserving, regime-dependent position decay (§3).
3. We provide direct evidence for the reliability premise: student–teacher divergence on high-advantage trajectories is a stable $2.8\times$ that on low-advantage ones, in both generation regimes (§4.4).
4. Across two generation regimes and a routed multi-teacher setup, we show that the additive combination actively damages the student in the thinking regime (AIME 0.90 to 0.75–0.82) while posterior weighting preserves and marginally improves it (0.90–0.92), with a Pure OPD control attributing the damage to the additive term (§4).

2 When Should We Trust the Teacher? A Probabilistic View

We develop a probabilistic account of the question posed in the title: setup and baseline objective (§2.1), derivation of reliability weights (§2.2), the main proposition (§2.3), and the contrast with KL-constrained RL (§2.4).

2.1 Problem Setup

Consider a student policy π_θ distilling from a fixed teacher ν. For a prompt x, the student samples a trajectory $\tau = (s_1, y_1, \ldots, s_T, y_T)$ from its own policy, with $s_t = (x, y_{<t})$ and sampled token y_t, and receives an outcome reward $R(\tau) \in \mathbb{R}$ (e.g., a verifier score). The standard on-policy distillation objective minimizes the expected divergence between student and teacher distributions over student-sampled states:

$$\min_\theta \; \mathbb{E}_{\tau\sim\pi_\theta}\left[\frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s_t) \,\|\, \nu(\cdot \mid s_t))\right]. \tag{1}$$

Equation equation 1 treats every trajectory as equally deserving of teacher guidance. Our premise is that this assumption is false: trajectories can drift into states the teacher rarely visits, where ν is no longer a trustworthy target. Since drift is not directly observable, we use the outcome reward as a proxy: high-reward trajectories are those on which the teacher's guidance is most likely to still be useful.

2.2 From Reward to Reliability Weights

We formalize the premise by introducing a weight $w(\tau) \geq 0$ that modulates each trajectory's contribution to the loss:

$$\min_\theta \; \mathbb{E}_{\tau\sim\pi_\theta}\left[w(\tau)\cdot\frac{1}{T}\sum_{t=1}^{T} D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s_t) \,\|\, \nu(\cdot \mid s_t))\right]. \tag{2}$$

What should $w(\tau)$ be? We derive it rather than impose a heuristic. Following control-as-inference (Levine, 2018), we introduce a binary optimality variable O and model optimal behavior with a Boltzmann distribution

$$p(O = 1 \mid \tau) \;\propto\; \exp(R(\tau)/\beta)\,, \tag{3}$$

where β controls the confidence placed in the reward signal—the same closed-form structure that appears in KL-regularized policy optimization (Levine, 2018; Tan & Hong, 2025). Using $p(O{=}1 \mid$

τ) as the reliability weight, $w(\tau) \propto \exp(R(\tau)/\beta)$, and normalizing within the GRPO-style group of N rollouts per prompt so that the per-prompt distillation budget is preserved:

$$w_i \;=\; N \cdot \frac{\exp(R_i/\beta)}{\sum_{j=1}^{N} \exp(R_j/\beta)} \;=\; N \cdot \mathrm{softmax}(R/\beta)_i \,. \tag{4}$$

Three properties are central. *Shift invariance:* the softmax is invariant to adding a constant to all group rewards; writing the group-relative advantage $A_i = (R_i - \mu)/\sigma$ gives $w_i = N \cdot \mathrm{softmax}(A_i/\beta')$ with $\beta' = \beta/\sigma$, so the weights depend only on relative standing within the group. *Budget preservation:* $\sum_i w_i = N$, so PW-OPD reallocates a fixed distillation budget across trajectories rather than changing the overall strength of distillation—no global coefficient needs to be tuned alongside the learning rate. *Graceful degeneration:* when $N = 1$ or all group rewards coincide, every weight equals one and Equation equation 2 reduces to the standard objective equation 1, so the method degrades exactly to OPD whenever the reward carries no discriminative information.

2.3 Main Proposition

Proposition 1. *Under the Boltzmann optimality model of Equation equation 3, the reliability weights that concentrate the weighted distillation objective of Equation equation 2 on trajectories more likely to be optimal are the group-normalized softmax weights of Equation equation 4, which are invariant to group-mean reward shifts and preserve the per-group distillation budget. When all rewards within a group coincide, the objective reduces to standard on-policy distillation.*

The proposition separates PW-OPD from prior reward-aware distillation (Zhang et al., 2026; Akhondzadeh et al., 2026; Ou et al., 2026): the weight is a *continuous* function of the group-relative advantage alone, its functional form is *derived* rather than imposed, and its normalization guarantee is exact rather than approximate.

2.4 Connection to KL-Constrained Reinforcement Learning

The additive combination of task reward and distillation corresponds to the Lagrangian of a KL-constrained reward-maximization problem: the reward is the primary objective, the KL term a constraint, and the two signals interact through a single global multiplier fixed throughout training. In PW-OPD the roles are reversed: the distillation objective is primary, and the reward enters only through the posterior weight $w(\tau)$. Both formulations tilt the student toward trajectories that are high-reward and teacher-consistent, but the reward's influence enters as a competing gradient source rather than as a reallocation of the distillation gradient—a distinction §4 shows is empirically consequential.

3 Posterior-Weighted On-Policy Distillation

3.1 Overview

The probabilistic view of §2 reduces to a lightweight modification of the on-policy distillation objective: replace the uniform treatment of trajectories with the group-normalized posterior weights of Equation equation 4. Figure 1 illustrates the resulting pipeline: the student samples N trajectories per prompt; a verifier assigns outcome rewards, from which the group-relative advantage A_i is computed in the usual GRPO fashion (Shao et al., 2024); the teacher is queried for log-probabilities on the student-generated tokens, exactly as in standard OPD. The only change is how the per-token divergences are combined.

This single design choice spans a family of algorithms through three independent components, which §4 varies one at a time to isolate each contribution: *posterior weighting*, whose absence recovers standard OPD and whose presence applies Equation equation 4; an *additive task-reward term*, i.e., a PPO-style loss with coefficient 1.0 appended to the distillation objective, the combination used in common implementations (Sheng et al., 2025); and *position-aware decay* (§3.3). Configurations are named by their active components—Pure OPD, OPD+TR, PW-OPD, PW-OPD+TR,

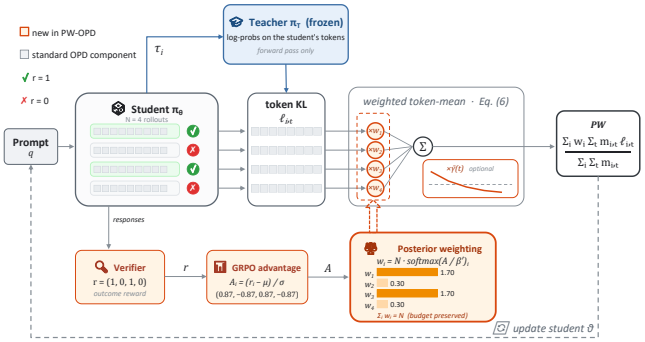

Figure 1: Overview of PW-OPD. The student samples N trajectories per prompt; a verifier scores each; posterior weights $w_i = N \cdot \mathrm{softmax}(A/\beta')_i$ are formed within the group; the teacher provides token-level log-probabilities on student-generated tokens; each trajectory's loss is scaled by its weight before token-mean aggregation. The only change relative to standard OPD is the amber posterior-weighting module.

PW-OPD+pos, PW-OPD+TR+pos—of which PW-OPD, posterior weighting without the additive term, is the one §4 finds most effective.

3.2 Sequence-Level Posterior Modulation

Let $\ell_{i,t}$ denote the per-token distillation divergence for trajectory i at position t, and $m_{i,t} \in \{0, 1\}$ the response mask. The posterior-weighted distillation loss is

$$\mathcal{L}_{\mathrm{PW}} = \frac{\sum_{i=1}^{N} w_i \sum_t m_{i,t}\, \ell_{i,t}}{\sum_{i=1}^{N} \sum_t m_{i,t}}, \qquad w_i = N \cdot \mathrm{softmax}(A/\beta')_i, \tag{5}$$

where the normalization by total valid tokens is the standard token-mean aggregation, so the loss scale is invariant to batch composition and gradient accumulation. Three details are worth noting. First, the weights depend only on quantities available before the forward pass: they are precomputed once per step by grouping samples by prompt identifier and carried into the loss as a per-sequence tensor—no additional communication is required inside the model forward. Second, the weight multiplies an already-aggregated divergence, so it is orthogonal to the choice of divergence and to how it is optimized. Third, β' is the only new hyperparameter: as $\beta' \to \infty$ the weights become uniform (standard OPD); as $\beta' \to 0$ the method approaches hard selection. We use $\beta' = 1$ throughout—the scale of the standardized advantages—with no task-specific tuning.

Relation to the additive form. A common implementation adds a task-reward loss to the distillation loss, so both terms contribute gradients to the same parameters additively (Sheng et al., 2025). In Equation equation 5 the reward influences the update only through w_i, reallocating a *fixed* distillation budget; the two designs are compared empirically in §4.

3.3 Token-Position-Aware Modulation

Equation equation 5 assigns one weight per trajectory. One may also express a prior about *where* within a response teacher guidance is most valuable—for instance, that early tokens shape the reasoning trajectory while late tokens are already committed. We introduce a second, independent modulation along the token axis:

$$\tilde{\gamma}(t) = \frac{\gamma(t)}{\frac{1}{L}\sum_{t'=1}^{L} \gamma(t')}, \qquad \gamma(t) = \exp(-\lambda\, t/L)\,, \tag{6}$$

Table 1: Short-context regime (8K, non-thinking): final validation accuracy and training reward; best AIME/HMMT across saved checkpoints. Best per column in bold.

Configuration	Val Acc	Reward	AIME 2026	HMMT
Pure OPD	0.60	0.497	0.68	0.50
OPD+TR	0.65	0.486	0.68	0.48
PW-OPD	**0.70**	**0.503**	0.68	0.50
PW-OPD+TR	0.60	0.469	0.68	**0.52**
PW-OPD+pos	0.55	**0.503**	**0.70**	0.47
PW-OPD+TR+pos	0.65	0.480	0.67	0.48

where L is the response length and $\lambda \geq 0$ the decay rate. The normalization is the token-axis analogue of budget preservation: the mean of $\tilde{\gamma}$ over valid tokens is exactly one, so the per-trajectory budget established by w_i is redistributed across positions but not changed in total. The full loss replaces $m_{i,t}\,\ell_{i,t}$ by $m_{i,t}\,\tilde{\gamma}(t)\,\ell_{i,t}$ in Equation equation 5; $\lambda = 0$ recovers it exactly. The two modulations act on orthogonal axes—w_i allocates the budget *across* trajectories, $\tilde{\gamma}(t)$ *within* one—and §4.4 shows that the effectiveness of position decay depends sharply on the generation regime.

3.4 Training Pipeline

PW-OPD is implemented on the verl framework (Sheng et al., 2025), using its hybrid-engine design, in which FSDP2 training workers and vLLM rollout workers share the student's GPUs while the teacher runs as a separate inference service. Training hyperparameters are given in Appendix A.

4 Experiments

Our experiments center on validating the effectiveness and superiority of posterior weighting. We compare PW-OPD against pure distillation, the additive combination, and the remaining component configurations across two generation regimes (§4.2–§4.3); probe *why* it helps by testing the reliability premise behind the derivation (§4.4); and examine robustness under an extended generation budget, a weaker teacher, and a routed multi-teacher setup in a different domain (§4.5, §4.6).

4.1 Experimental Setup

Models and training. The student is Qwen3.5-9B and the teacher Qwen3.6-27B, same-family released models with a shared tokenizer. All runs train on the same mathematics corpus, sampling each prompt $N{=}4$ times per step—the setting in which the group-relative advantage and hence the posterior weight are defined—with a single-sample reverse-KL estimator through the clipped policy-gradient path and $\beta' = 1$ in every run. Remaining hyperparameters are shared (Appendix A).

Regimes and evaluation. The *short* regime permits up to 8K response tokens with the chat template's thinking mode disabled; the *long* regime permits 32K with thinking enabled. We report accuracy on AIME 2026 and HMMT February 2026, each problem sampled four times and scored by exact answer matching, taking the best accuracy across saved checkpoints; the untrained student is evaluated under the same harness (0.90 AIME / 0.69 HMMT). We additionally report training-time validation accuracy on held-out data from the training distribution.

Compared configurations. We evaluate the six configurations induced by the three flags of §3.1: pure OPD, the uniform-weight member of the GKD family (Agarwal et al., 2024), as the distillation-only baseline; the additive combination (*OPD+TR*); posterior weighting alone (*PW-OPD*); and the *PW-OPD+TR*, *PW-OPD+pos*, *PW-OPD+TR+pos* counterparts. All other hyperparameters are held fixed.

4.2 Main Results: Short-Context Regime

Table 1 reports the six configurations in the 8K regime. PW-OPD attains the highest validation accuracy and ties for the highest training reward; adding the task-reward term on top (PW-OPD+TR)

Table 2: Long-context regime (32K, thinking): range of best accuracies across saved checkpoints; the reference row is the untrained student under the same harness. Best trained configuration in bold.

Configuration	AIME 2026	HMMT
Untrained student (ref.)	*0.90*	*0.69*
Pure OPD	0.87–0.91	0.67–0.70
OPD+TR	0.75–0.82	0.66–0.69
PW-OPD	**0.90–0.92**	0.67–0.70
PW-OPD+pos	**0.88–0.92**	**0.70–0.73**

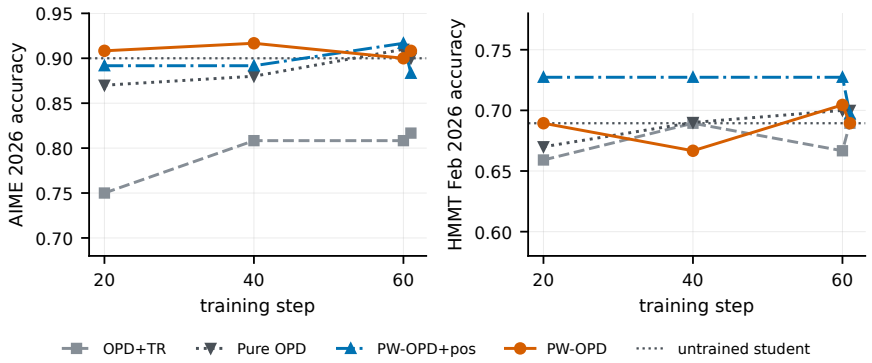

Figure 2: External benchmark accuracy over training in the 32K thinking regime. The dotted line marks the untrained student (0.90 AIME / 0.69 HMMT); the additive combination falls below it, distillation-only tracks or exceeds it.

does not improve over PW-OPD, so the additive term is redundant once the reward already enters through the weights; and position-aware decay degrades validation accuracy substantially (0.70 → 0.55). The external benchmarks do not separate the configurations here—every run lands between 0.67–0.70 on AIME and 0.47–0.52 on HMMT—so the short regime is saturated with respect to them, and validation accuracy, which averages over a far larger problem set, is what discriminates between methods.

4.3 Main Results: Long-Context Thinking Regime

Table 2 and Figure 2 report the 32K thinking regime, where the benchmarks do separate the configurations. First, the additive combination does not merely underperform—it damages the student: adding a PPO-style task-reward loss to distillation drives AIME accuracy from the untrained 0.90 down to 0.75–0.82. The Pure OPD control closes the attribution: reward-free distillation preserves the student (0.87–0.91 AIME), so the degradation is caused by the additive reward term, not by long-thinking training per se. Second, posterior weighting preserves and marginally improves the student (0.90–0.92 AIME; 0.70–0.73 HMMT with position weighting), so its advantage over the additive combination reflects damage avoided rather than capability manufactured—precisely what a modulation-only design predicts. PW-OPD also sits slightly above the Pure OPD control (0.90–0.92 versus 0.87–0.91), so the reward signal contributes a small consistent edge without incurring the additive term's damage; and it is at or above the untrained reference from the first evaluated checkpoint, whereas Pure OPD recovers only in the second half of training—posterior weighting reaches the preservation regime faster. Third, position-aware decay, harmful in the short regime, is neutral-to-beneficial here (0.88–0.92 AIME, 0.70–0.73 HMMT), an interaction analyzed in §4.4.

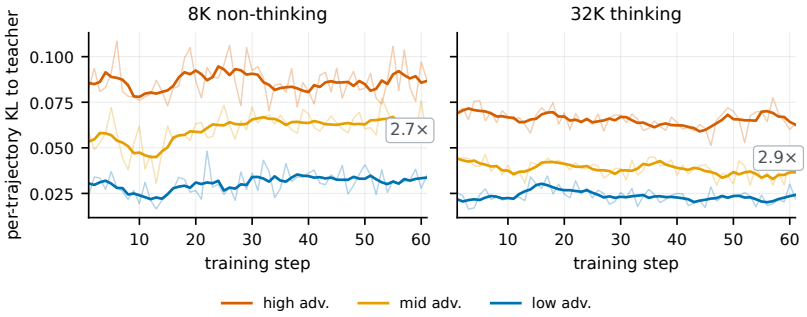

Figure 3: Per-trajectory divergence by advantage tercile (PW-OPD; thin: raw; thick: 5-step moving average). High-advantage trajectories diverge substantially more throughout training, in both regimes.

4.4 Analysis: Why Posterior Weighting Helps

4.4.1 Teacher Guidance Concentrates on High-Advantage Trajectories

Figure 3 bins trajectories by within-group advantage tercile and plots the mean per-trajectory distillation divergence over training. The separation is large and stable: averaged over training, high-advantage trajectories exhibit $2.8\times$ the divergence of low-advantage ones, in both regimes (mean 0.086 versus 0.031 short; 0.066 versus 0.024 long). This is the empirical content of the reliability premise, and its direction matters: it is the *high*-reward trajectories on which student and teacher disagree most—successful student trajectories are precisely those on which the student has found reasoning paths the teacher would not have taken, so the teacher still has something to teach there. Uniform distillation spends most of its gradient budget where it is least needed; posterior weighting redirects it.

4.4.2 Weight and Position Behavior

Weight calibration. The mean within-group weight entropy stays between 1.22 and 1.23 nats throughout training in both regimes—88–89% of the $\ln 4$ ceiling—so the weights discriminate without collapsing to uniformity or hard selection. In the long regime roughly half of all trajectories receive weight exactly one (groups whose four rollouts agree), and the weight standard deviation decreases over training ($0.686 \rightarrow 0.504$) as the improving student produces more agreeing groups: the modulation attenuates naturally as the signal saturates. Distributions are shown in Appendix C.1.

Position effects are regime-dependent. Partitioning each response into thirds, all configurations show divergence decreasing from early to late tokens, and position-aware decay amplifies early-token divergence (0.089–0.090 versus 0.075 in the short regime)—the direction its design predicts. The effect on performance, however, depends entirely on the regime: at 8K the response is too short for early and late segments to play distinct roles, so concentrating guidance on opening tokens removes it from the portion that determines the answer; at 32K, where responses occupy the full budget in extended reasoning chains, the same redistribution is neutral to beneficial (Appendix C.2).

Validation–evaluation decoupling. In the thinking regime, training-time validation accuracy falls (to $\sim$0.25 for the best run) while AIME accuracy rises above 0.90, with roughly 80% of responses reaching the 32K generation limit. The most plausible explanation is a difficulty mismatch: the validation corpus contains many problems the student can solve without thinking, on which long-reasoning behavior over-deliberates and introduces errors, while the competition benchmarks require exactly that behavior. Training-time validation on an unfiltered corpus is therefore not a reliable selection signal in long-thinking regimes (Appendix C.3).

Table 3: Multi-teacher extension: an 8B student with four routed domain teachers. Science: GPQA-Diamond; Code: LiveCodeBench-V6; IF: IFEval. Best checkpoint per run; the teacher row is the best of the four specialists per benchmark. Best trained configuration in bold.

Configuration	Science	AIME	MATH-500	Code	IF	Avg
Student before distillation	*64.65*	*70.83*	*97.40*	*65.45*	*63.77*	*72.42*
MOPD (uniform)	63.13	75.83	97.25	69.53	77.26	76.60
MOPD+TR (additive)	**66.67**	**77.08**	97.02	68.63	77.82	77.44
PW-OPD (posterior)	**66.67**	76.67	**97.08**	**69.65**	**79.11**	**77.84**
Strongest teacher	*65.15*	*75.42*	*96.97*	*70.74*	*80.80*	*77.82*

4.5 Extended Studies

Generation budget. A run at 128K generation length performs on par with the 32K runs (0.88–0.90 AIME, 0.70–0.74 HMMT) at considerably higher per-step cost; within the range tested, 32K is the cost-effective operating point.

Teacher quality. Substituting a weaker teacher with the method and student fixed degrades accuracy markedly (AIME 0.66–0.69): posterior weighting modulates the strength of teacher guidance but cannot manufacture guidance the teacher does not possess.

Training cost. The weighting introduces no additional generation; per-step wall-clock (492 s at 8K, 1800 s at 32K) is dominated by rollouts and the training update (Appendix A). A broader benchmark suite (e.g., AMC) is left to future work.

4.6 Extension: Multi-Teacher Distillation on Qwen3-8B

We further test whether the weighting rule transfers beyond the single-teacher setting. Because domain-specialist teachers are available only at the 8B scale in our setup, this extension distills into an 8B student (Qwen3-8B after a two-stage SFT (Yang et al., 2026)) from four DAPO-trained specialists (instruction following, code, mathematics, science), routing each training prompt to its domain teacher and applying equation 5 within each group unchanged, with the same $\beta' = 1$. We compare against uniform multi-teacher OPD (MOPD) and the additive task-reward combination (MOPD+TR).

Table 3 reports per-domain accuracy at the best checkpoint of each run, as in the main experiments. All variants improve substantially over the student before distillation (average 72.42 to 76.60–77.84), and the ordering among them matches the main experiments: posterior weighting attains the best average (77.84 versus 77.44 additive and 76.60 uniform), improving over uniform in four of the five domains (up to $+3.5$ on science), with saturated MATH-500 within 0.2; the additive combination wins only AIME and ties science. The student already exceeds the strongest teacher on science and AIME but remains 4–6 points below it on code and instruction following, consistent with the teacher-strength requirement of §4.5.

5 Related Work

Knowledge and on-policy distillation. Knowledge distillation trains a student to match a teacher (Hinton et al., 2015; Gou et al., 2020); the classic sequence-level approach trains on a fixed corpus of teacher outputs and inherits exposure bias. Two lines address this mismatch by sampling training sequences from the student itself: MiniLLM (Gu et al., 2023) minimizes a reverse KL on student-generated sequences, and GKD (Agarwal et al., 2024) unifies this into a framework interpolating on- and off-policy data with flexible divergences. On-policy distillation has since become a standard post-training recipe for reasoning models (Lu & Thinking Machines Lab, 2025; Yang et al., 2025; Sheng et al., 2025), surveyed systematically by Song & Zheng (2026) and re-examined mechanistically by Li et al. (2026).

Distillation meets reinforcement learning. Because on-policy rollouts can be scored by any reward signal, several methods combine teacher supervision with reward optimization: GKD can mix its objective with RL fine-tuning (Agarwal et al., 2024), subsequent work integrates distillation into preference-based objectives (Zhang et al., 2026) or a unified post-training pipeline (Xu et al., 2025), and open frameworks expose a switch that adds a PPO-style task-reward loss with a fixed coefficient (Sheng et al., 2025). These approaches share a common structure: the reward enters *additively*, with a global coefficient tuned by hand.

Reward-aware distillation. A concurrent line shares our motivation that teacher supervision is not uniformly reliable on student trajectories. RLAD (Zhang et al., 2026) replaces the teacher-anchored KL regularizer with a clipped likelihood-ratio objective activated only when the advantage supports it; RG-OPD (Akhondzadeh et al., 2026) keeps a trajectory only when its verifier reward and the teacher–student likelihood gap agree in sign; RWOPD (Ou et al., 2026) weights a forward-KL objective by verifier scores for hardware-design generation. Our approach differs along three axes: the weight is a *continuous* function of the group-relative advantage alone, it is normalized so the per-group distillation budget is preserved, and its functional form is *derived* from a probabilistic model of teacher reliability rather than imposed.

Group-relative advantages and weighted imitation. GRPO (Shao et al., 2024) replaces the learned value baseline with the mean of same-prompt rewards, and prior work uses group-relative advantages exclusively as a policy-gradient signal; we show the same signal is well suited to weighting distillation, being scale-invariant and available whenever rollouts are grouped. Our formulation also connects to advantage-weighted regression (Peng et al., 2019) and control-as-inference (Levine, 2018), where updates are reweighted by exponentiated advantages or derived as probabilistic inference; PW-OPD applies this weighting philosophy to the distillation objective, with the softmax form arising from the Boltzmann posterior rather than being imposed.

6 Conclusion

We asked when on-policy distillation should trust the teacher, and answered with a derivation rather than a heuristic: under a Boltzmann model of trajectory optimality, the principled per-trajectory weight is the group-normalized posterior $w_i = N \cdot \mathrm{softmax}(A/\beta')_i$. PW-OPD demotes the task reward from a competing additive objective to a modulation factor that reallocates a fixed distillation budget—translation-invariant, budget-preserving, and exactly recovering standard OPD whenever the reward carries no within-group information.

Every link of this chain survived measurement. The reliability premise is directly observable—student–teacher divergence on high-advantage trajectories is $2.8\times$ that on low-advantage ones, stably across generation regimes—and the weights remain calibrated between the uniform and degenerate extremes. In the long thinking regime, the additive combination actively damages the student (AIME 0.90 to 0.75–0.82), reward-free distillation preserves it, and posterior weighting preserves it faster and at a slightly higher level (0.90–0.92), with the Pure OPD control attributing the damage to the additive term. The same ordering replicates in a multi-teacher extension under a different student and five domain benchmarks, and the validation–evaluation decoupling we document has practical value beyond this method.

Limitations. The modulation acts only where the reward discriminates: when all outcomes in a group agree—about half the groups late in 32K training—the weights collapse to uniform and the method degenerates benignly to standard OPD. Our full factorial validation is in mathematics; the multi-teacher replication uses a smaller student and self-trained teachers. The method presupposes verifiable rewards and a teacher stronger than the student; with a weaker teacher the benefit shrinks substantially, and in thinking regimes part of what posterior weighting delivers is damage avoidance rather than active gain.

Future work. Natural next steps include continuous rewards, online per-teacher trustworthiness estimation in multi-teacher settings, and learned budget-preserving weight families. More broadly, our results suggest a design lesson for combining weak signals with strong teachers: when a signal is informative but unreliable, *modulate* with it rather than optimize against it.

AI USE STATEMENT

In this work, generative AI tools were used only for recommended-disclosure tasks—polishing parts of the paper text and assisting with literature retrieval—while no required-disclosure task (methodology design, method implementation, or synthetic data generation) involved generative AI. All AI-assisted text and retrieved references were reviewed and verified by the authors against official arXiv or publisher sources. The method, its implementation, and all experiments were developed, designed, and executed by the authors, with generative AI assisting only auxiliary programming; the authors take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

The method is fully specified in §3. Appendix A lists the complete shared training and evaluation configuration, and all compared configurations differ only in the components of §3.1. We will release the training and evaluation code upon publication.

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 (ICLR)*, 2024.

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

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *arXiv preprint arXiv:2006.05525*, 2020.

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

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

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. Blog post. https://thinkingmachines.ai/blog/on-policy-distillation.

Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, et al. Reward-weighted on-policy distillation with an open property-equivalence verifier for NL-to-SVA generation. *arXiv preprint arXiv:2605.13501*, 2026.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the ACM European Conference on Computer Systems (EuroSys)*, 2025. doi: 10.1145/3689031.3696075.

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

Zhiquan Tan and Yinrong Hong. A theoretical lens for RL-tuned language models via energy-based models. *arXiv preprint arXiv:2512.18730*, 2025.

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. KDRL: Post-training reasoning LLMs via unified knowledge distillation and reinforcement learning. *arXiv preprint arXiv:2506.02208*, 2025.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, et al. Nemotron-Cascade 2: Post-training LLMs with cascade RL and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for LLM reasoning. *arXiv preprint arXiv:2602.22495*, 2026.

A Hyperparameters

Table 4 lists the training and evaluation configuration shared by all runs in §4; compared configurations differ only in the components of §3.1.

Table 4: Shared training and evaluation configuration. All runs use the same values; regimes differ only in the response budget and thinking mode.

Item	Value
Student / teacher	Qwen3.5-9B / Qwen3.6-27B (same tokenizer)
Rollouts per prompt N	4
Prompts per step	32 (128 trajectories)
Posterior temperature β'	1
Position decay rate λ (equation 6)	0.5
Distillation estimator	single-sample k-estimator, PG path
Clipping (ratio / per-token)	0.2 symmetric / ± 3
Advantage estimator	GRPO, group-std normalized
Learning rate	1×10^{-6}, cosine, 10-step warmup
Weight decay / grad clip	0.1 / 1.0
Rollout sampling	temperature 1.0, top-p 0.95
Short regime	8K response tokens, thinking off
Long regime	32K response tokens, thinking on
Per-step wall-clock	492 s (8K) / 1800 s (32K)

B Evaluation Protocol

External benchmarks are AIME 2026 and HMMT February 2026. Each problem is sampled four times per checkpoint and scored by exact answer matching; we report the best accuracy across the checkpoints saved during training, a protocol applied identically to every configuration, including the untrained-student reference (0.90 AIME / 0.69 HMMT), which is evaluated under the same harness. The multi-teacher extension is evaluated on its five domain benchmarks under the same protocol. Training-time validation accuracy is computed every 10 steps on held-out prompts from the training distribution (`mean@1`). As documented in §4.4, validation accuracy and external accuracy decouple in the thinking regime, so method comparisons in §4.3 rely on the external benchmarks.

C Additional Analysis

C.1 Weight Calibration

Figure 4 characterizes the weight distribution. The mean within-group entropy stays between 1.22 and 1.23 nats throughout training in both regimes, 88–89% of the maximum attainable $\ln 4 = 1.386$

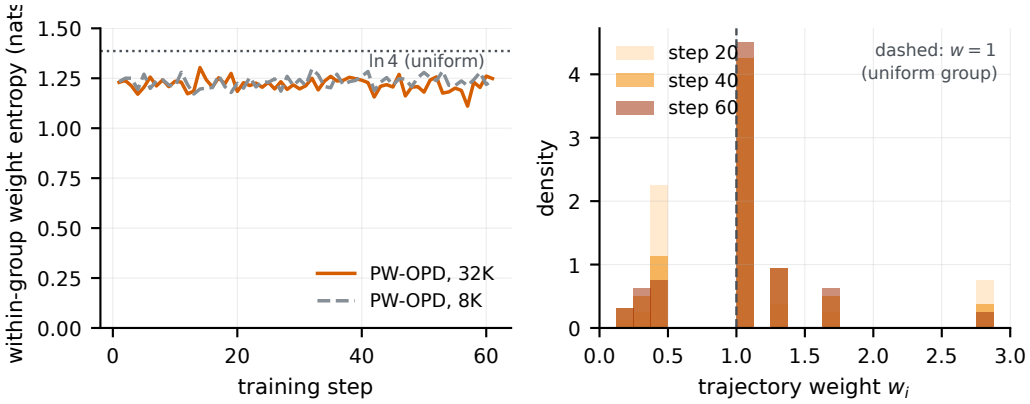

Figure 4: Weight distribution over training (PW-OPD). Left: within-group weight entropy; the dotted line marks the uniform-weight ceiling $\ln 4$. Right: histogram of trajectory weights at three checkpoints; the dashed line marks $w = 1$. Entropy stays near the upper end of its range, indicating that the weights discriminate between trajectories without collapsing onto a single sample.

for groups of four: the weights discriminate meaningfully while remaining far from both degenerate solutions—uniform collapse (which would reduce the method to standard OPD) and single-trajectory concentration (which would reduce it to hard selection). Two further observations concern the evolution of the distribution. In the long-context regime, roughly half of all trajectories receive a weight of exactly one: these belong to groups in which all four rollouts agree, where the softmax has no disagreement to exploit and the weight necessarily becomes uniform, so the modulation is active precisely on the mixed-outcome groups where the reward carries information. The standard deviation of the weights decreases over training (0.686 to 0.504 from the first to the last checkpoint in PW-OPD), a benign drift reflecting that the improving student's rollouts increasingly agree.

C.2 Token-Position Effects

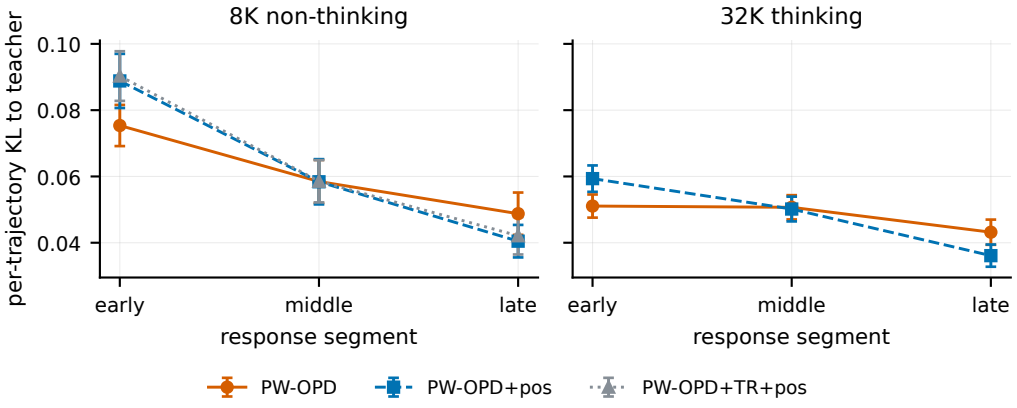

Figure 5: Per-segment divergence (mean $\pm$ 1 std over training steps). Without position weighting, divergence decreases from early to late tokens; with position weighting, early-token divergence is amplified.

Figure 5 partitions each response into thirds and reports the mean divergence in each. All configurations show the same qualitative ordering—divergence highest on early tokens and lowest on late tokens—consistent with the student drifting progressively further from the teacher as its own reasoning accumulates. Position-aware decay changes this profile in the direction its design predicts, amplifying the weight on early tokens: the early-token divergence of position-aware runs exceeds that of their position-unaware counterparts (0.089–0.090 versus 0.075 in the short-context regime).

Whether this redistribution helps depends entirely on the generation regime, as discussed in §4.4: with only 8K tokens of budget the response is too short for early and late segments to play distinct roles, whereas 32K thinking responses occupy the full budget in substantially longer reasoning chains, where the redistribution is neutral to slightly beneficial.

C.3 Validation–Evaluation Decoupling

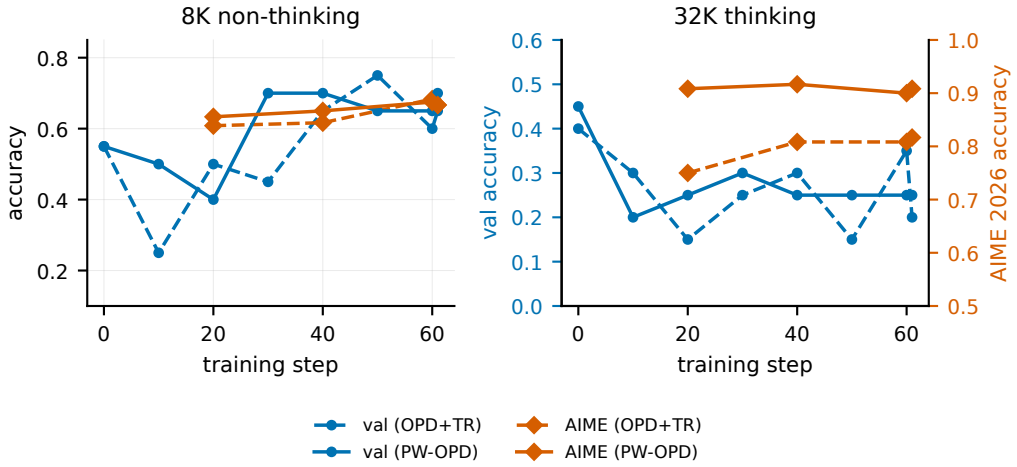

Figure 6: Decoupling of validation and evaluation in the thinking regime. Left: in the 8K non-thinking regime, validation accuracy (blue) and AIME accuracy (orange) move together. Right: in the 32K thinking regime (validation on left axis, AIME on right axis), training-time validation accuracy falls while external benchmark accuracy rises.

Figure 6 reports an observation of practical significance beyond the present method. In the short-context regime, validation accuracy and external benchmark accuracy move together: the best configuration on validation is also among the best on AIME and HMMT. In the 32K thinking regime the correspondence breaks down—the best run's validation accuracy falls to roughly 0.25 while its AIME accuracy rises above 0.90, and several runs hit the 32K generation limit on about 80% of responses. The most plausible explanation is a difficulty mismatch: the held-out validation set contains many problems the student can already solve in a short non-thinking response, and training for long thinking makes it over-deliberate on them, consuming budget and introducing opportunities for error, while the competition benchmarks require exactly the long-reasoning behavior. The methodological consequence is that in long-thinking regimes, training-time validation on an unfiltered corpus is not a reliable model selection signal, and evaluation must rely on external benchmarks whose difficulty matches the target deployment.