HARD PREFERENCE CALIBRATION FOR ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) has emerged as an effective paradigm for post-training large language models, yet teacher preferences can conflict with verifier outcomes. In practice, a teacher may prefer an incorrect trajectory over a correct one, making token-level guidance unreliable on *critical examples*. We thus propose Hard Preference Calibration (HPC), which augments OPD with an adaptive contrastive objective that calibrates verifier-derived *hard preferences* using the teacher's pairwise reliability. For each correct-incorrect trajectory pair, HPC applies stronger correction as the teacher's preference becomes less trustworthy, while preserving OPD's dense supervision. Our theoretical analysis further shows that this calibration yields a strictly tighter generation-error bound than OPD and improves Pass@k. Across multiple mathematical reasoning and code generation benchmarks, it outperforms OPD and related variants across student scales and teacher regimes, with absolute Pass@16 gains of up to 5.3% on HMMT25 and gains particularly evident when teacher preferences are less well aligned with verifier outcomes.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as an effective approach to improving reasoning, but sparse trajectory-level feedback limits credit assignment to intermediate reasoning steps (Shao et al., 2024; DeepSeek-AI, 2025; Hu et al., 2025; Cui et al., 2026; Yuan et al., 2025). On-policy distillation (OPD) complements RLVR by providing dense, token-level teacher guidance on trajectories generated by the student, aligning training with the contexts that student encounters during generation. OPD has demonstrated strong reasoning gains at lower training cost than methods such as Group Relative Policy Optimization (GRPO). Recent work has explored its use in strong-to-weak distillation (Qwen Team, 2025; LLM-Core Xiaomi, 2026) and cross-family transfer (Patiño et al., 2025), as well as its connections to reinforcement learning (Yang et al., 2026).

Despite recent progress, the reliability of teacher supervision on trajectories sampled from an evolving student policy remains unclear in practice. Existing work mitigates this issue by filtering unreliable trajectories or tokens (Li et al., 2026), adjusting the contribution of each sample through adaptive weights (Zheng et al., 2026), restricting optimization to verifiable teacher signals (Xing et al., 2026; Plyusov et al., 2026), or analyzing when token-level guidance aligns with outcome rewards (Hou et al., 2026; Xu et al., 2026; Akhondzadeh et al., 2026). These approaches suppress unreliable supervision, but do not fully exploit the binary correctness signal available in verifiable tasks. A correct–incorrect trajectory pair induces a deterministic trajectory-level *hard preference*, raising a complementary question: how should the teacher's pairwise reliability regulate the use of this preference?

To assess the teacher's pairwise reliability during OPD training, we track the fraction of prompts for which the lowest teacher score among correct rollouts falls below the highest score among incorrect rollouts, with correctness determined by the verifier. Each score averages teacher's log probabilities across tokens. As shown in Fig. 1, this score overlap remains common throughout training, with no clear downward trend. These results indicate that teacher preferences remain imperfectly aligned with verifier outcomes on student-generated trajectories. This mismatch motivates augmenting the dense teacher supervision in on-policy distillation with explicit calibration guided by the verifier.

We therefore propose Hard Preference Calibration (HPC) for on-policy distillation, which estimates the teacher's pairwise reliability from relative sequence likelihoods and uses it to calibrate the verifier-derived hard preference. When teacher and verifier preferences agree, no correction is applied; when

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

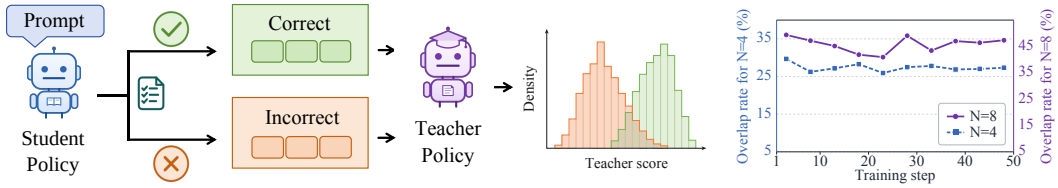


Figure 1: Prompt-level teacher-score overlap during OPD training. The left panel shows how verifier outcomes partition the rollouts into correct and incorrect sets, how teacher scores are summarized within them, and how an overlap indicator is formed for each prompt before averaging across prompts in a batch. The right panel also shows that this overlap remains common throughout training and reports the five-step means for 4 or 8 rollouts per prompt, with N denoting the number of rollouts.

they conflict, HPC introduces a trajectory-level corrective objective whose strength increases with teacher unreliability. This preserves OPD's dense token supervision while turning unreliable teacher preferences into targeted learning signals rather than discarding them, using verifier outcomes and model probabilities available during post-training. Our contributions are summarized as follows.

- We propose Hard Preference Calibration (HPC), which augments OPD with explicit and selective correction based on verifier outcomes at the trajectory level while retaining dense token supervision. For each prompt, HPC forms pairs with one correct and one incorrect trajectory, then uses the teacher's pairwise reliability to determine whether correction is applied and how strongly.
- We develop a theoretical analysis showing that every interior stationary point of the HPC objective admits a strictly tighter generation-error bound than OPD, and that this reduction yields a quantitative improvement in Pass@k, with the gain quantified by an explicit correction certificate.
- We conduct extensive experiments across mathematical reasoning and code generation benchmarks, and HPC consistently outperforms OPD and related baselines across student scales and teacher regimes, raising Pass@16 by up to 5.3 percentage points on hard benchmarks such as HMMT25.

2 Related Work

On-policy distillation. On-policy distillation (OPD) aligns the student with the teacher on student-generated trajectories, preserving dense token-level supervision while reducing the distribution mismatch of off-policy distillation (Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025). It has been extended to strong-to-weak distillation (Qwen Team, 2025; LLM-Core Xiaomi, 2026), cross-family transfer (Patiño et al., 2025), black-box settings where teacher logits are not accessible (Ye et al., 2025), self-distillation (Hübotter et al., 2026; Shenfeld et al., 2026; Zhao et al., 2026; Yang et al., 2025), training agent guided by verifier (Wang et al., 2026), and dense-RL formulations (Yang et al., 2026). Yet these variants still treat teacher token supervision as uniformly reliable across student trajectories, thereby leaving unreliable teacher signals insufficiently addressed.

Beyond uniform teacher supervision. A growing body of work explores alternatives to uniform teacher supervision. One line modulates OPD by filtering or down-weighting unreliable trajectories (Li et al., 2026), controlling updates or rollout distributions through trust regions (Xing et al., 2026; Plyusov et al., 2026), calibrating guidance for consistency with outcome rewards (Hou et al., 2026), or adapting sample weights to regulate supervision (Zheng et al., 2026). Another line uses preference or reward feedback through offline preference optimization (Rafailov et al., 2023; Meng et al., 2024; Azar et al., 2024), sample selection based on rewards (Dong et al., 2023), or reinforcement learning for self-correction (Kumar et al., 2025). Our method combines both: it retains dense OPD supervision while introducing correction based on verifier-derived hard preferences, with the teacher's pairwise reliability determining when and how strongly the gate activates this correction.

3 Method

3.1 Preliminaries

Setup. Let π_θ be the trainable student and π_T a frozen teacher with accessible token log-probabilities. For each prompt $\boldsymbol{x}_i \sim \mathcal{D}$, the student samples K trajectories $\{\boldsymbol{y}_{i,k}\}_{k=1}^{K}$, each assigned a binary verifier

reward $r_{i,k} \in \{0, 1\}$. This partitions the samples into correct and incorrect sets, $\mathcal{P}_i = \{k : r_{i,k} = 1\}$ and $\mathcal{N}_i = \{k : r_{i,k} = 0\}$. Our method operates on pairs in $\mathcal{P}_i \times \mathcal{N}_i$ when both set are non-empty.

On-policy distillation objective. We consider the reverse-KL formulation of on-policy distillation (OPD) (Agarwal et al., 2024; Gu et al., 2024; Lu & Thinking Machines Lab, 2025), which minimises the divergence between the student and the teacher distributions using token log-probabilities from both models along the same trajectories generated by the current student policy during training:

$$\mathcal{L}_{\mathrm{OPD}}(\theta) \;=\; \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\big[\mathcal{D}_{\mathrm{KL}}\big(\pi_\theta(\boldsymbol{y} \mid \boldsymbol{x}) \,\big\|\, \pi_T(\boldsymbol{y} \mid \boldsymbol{x})\big)\big]\,. \tag{1}$$

At each token y_t, the log-probability gap $\log \pi_\theta(y_t \mid \cdot) - \log \pi_T(y_t \mid \cdot)$ serves as a distillation signal. Dense teacher supervision along trajectories generated by the student motivates on-policy distillation.

3.2 Hard Preference Calibration

On-policy distillation provides dense teacher supervision, but the teacher may rank an incorrect trajectory above a correct one for the same prompt. Our proposed Hard Preference Calibration (HPC) addresses this mismatch through a pairwise corrective loss based on verifier-derived hard preferences at the trajectory level. The key intuition is to strengthen the student's preference for the correct trajectory when the teacher favours the incorrect one or scores both similarly. Accordingly, HPC uses the teacher's pairwise reliability to select and scale the pairwise corrective terms. The resulting loss is added to the OPD objective, combining verifier-guided correction with dense token-level teacher supervision throughout training. The overall framework of our proposed method is shown in Figure 2.

Sequence score. For each trajectory $\boldsymbol{y}_{i,k}$, we measure the teacher's and student's mean per-token log-probability as the scores of the k-th trajectory under the teacher model and the student model,

$$\begin{cases} T_{i,k} \;=\; \frac{1}{|\boldsymbol{y}_{i,k}|}\sum_{t=1}^{|\boldsymbol{y}_{i,k}|} \log \pi_T\big(y_{i,k,t} \,\big|\, \boldsymbol{x}_i, \boldsymbol{y}_{i,k,<t}\big), \\ S^\theta_{i,k} \;=\; \frac{1}{|\boldsymbol{y}_{i,k}|}\sum_{t=1}^{|\boldsymbol{y}_{i,k}|} \log \pi_\theta\big(y_{i,k,t} \,\big|\, \boldsymbol{x}_i, \boldsymbol{y}_{i,k,<t}\big). \end{cases} \tag{2}$$

A high $T_{i,k}$ or $S^\theta_{i,k}$ indicates that the trajectory lies in a relatively familiar region of the corresponding model policy, whereas a low $T_{i,k}$ or $S^\theta_{i,k}$ instead suggests that it lies in a rarely visited region.

Preference reliability gate. For each verifier-defined correct–incorrect pair $(p, n) \in \mathcal{P}_i \times \mathcal{N}_i$, we first quantify how strongly the teacher supports the same preference ordering by comparing its sequence scores for the two trajectories. Specifically, we define the teacher's soft pairwise preference as

$$P_T(p \succ n) = \sigma\big(T_{i,p} - T_{i,n}\big), \tag{3}$$

where $\sigma(z) = 1/(1 + \exp(-z))$ is the sigmoid function, and $P_T(p \succ n) > 0.5$ indicates that the teacher assigns a higher sequence score to the correct trajectory than to the incorrect one. Values below 0.5 indicate the opposite preference. We use this preference to define the reliability gate:

$$w_{i,p,n} = \big[\rho_T - P_T(p \succ n)\big]_+, \tag{4}$$

where $[x]_+ = \max(0, x)$ and $\rho_T \in (0.5, 1)$ is the teacher-reliability threshold, specifying the minimum teacher support required to deactivate correction. This gate controls pairwise correction. When $P_T(p \succ n) \geq \rho_T$, the teacher supports the verifier ordering and no correction is applied. Otherwise, the gate increases with the shortfall $\rho_T - P_T(p \succ n)$, assigning stronger supervision to pairs with weaker teacher support. Thus, the gate preserves OPD when teacher and verifier preferences are aligned, while strengthening corrective supervision as their disagreement increases.

Weighted contrastive learning. We use the gate $w_{i,p,n}$ to scale a pairwise loss that encourages the student to favour correct trajectories over incorrect ones for each sampled prompt. For each pair $(p, n) \in \mathcal{P}_i \times \mathcal{N}_i$, we define the student's soft pairwise preference as $P_\theta(p \succ n) = \sigma(S^\theta_{i,p} - S^\theta_{i,n})$. We seek to satisfy $P_\theta(p \succ n) \geq \rho_S$ for a target preference level $\rho_S \in (0.5, 1)$ on each pair during training. This is equivalent to requiring $S^\theta_{i,p} - S^\theta_{i,n} \geq \mathrm{logit}(\rho_S)$, where $\mathrm{logit}(\rho_S) = \log\big(\rho_S/(1 - \rho_S)\big)$ is the target score margin. We use a pairwise preference loss to strengthen the student's preference for the correct trajectory when it falls below the target preference level, normalizing the gate across pairs:

$$\mathcal{L}_{\mathrm{HPC}} \;=\; \sum_{(p,n)\in\mathcal{P}_i\times\mathcal{N}_i} \frac{w_{i,p,n}}{\sum_{(p',n')\in\mathcal{P}_i\times\mathcal{N}_i} w_{i,p',n'}} \cdot \big[\,\mathrm{logit}(\rho_S) - \big(S^\theta_{i,p} - S^\theta_{i,n}\big)\,\big]_+. \tag{5}$$

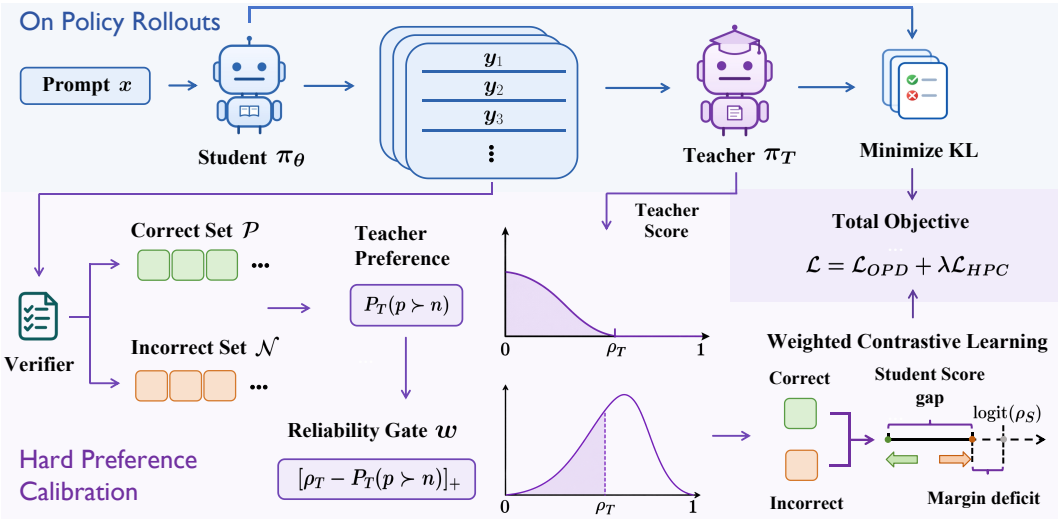


Figure 2: Overview of HPC. Verified student rollouts form correct–incorrect pairs per prompt, and the teacher's preference determines the gate for the hard-preference correction. The correction strengthens student preferences for correct trajectories, emphasising pairs where teacher's preferences conflict with verifier outcomes. It augments the OPD objective and preserves dense teacher supervision.

This objective focuses corrective supervision on pairs for which the teacher provides insufficient support for the correct trajectory and the student's score gap remains below the target margin. Once the student achieves the required separation, the pair contributes no further corrective loss. This allows the method to address weak student preferences without penalising pairs that already satisfy the target. For prompts with identical verifier outcomes across sampled trajectories or zero corrective weights across all pairs, the corrective term is set to zero, and training relies solely on OPD supervision.

The Final Objective. With Equation (5), HPC can be directly used to augment OPD as regularization:

$$\mathcal{L} \;=\; \mathcal{L}_{\text{OPD}} \;+\; \lambda \cdot \mathcal{L}_{\text{HPC}}, \tag{6}$$

where $\lambda \geq 0$ controls the strength of the corrective term relative to the original OPD objective. This formulation combines dense token-level teacher supervision with selective, adaptive trajectory-level correction based on verifier-derived hard preferences, and recovers standard OPD when $\lambda = 0$.

3.3 Gradient Analysis and Complexity

Gradient interpretation. To understand how the weighted corrective loss $\mathcal{L}_{\text{HPC}}$ shapes the student's preferences, we examine the gradient contribution of a single pair $(p, n) \in \mathcal{P}_i \times \mathcal{N}_i$. Let $\mathcal{L}^{\text{HPC}}_{i,p,n}$ denote its contribution to Eq. (5). For fixed sampled trajectories, this term depends on θ only through the student scores $S^\theta_{i,p}$ and $S^\theta_{i,n}$. For an active pair with $w_{i,p,n} > 0$ and $S^\theta_{i,p} - S^\theta_{i,n} < \text{logit}(\rho_S)$, the correction encourages a larger student score gap between the two trajectories, with gradient given by

$$\nabla_\theta \mathcal{L}^{\text{HPC}}_{i,p,n} \;=\; -\frac{w_{i,p,n}}{\sum_{(p',n')\in\mathcal{P}_i\times\mathcal{N}_i} w_{i,p',n'}} \cdot \big(\nabla_\theta S^\theta_{i,p} - \nabla_\theta S^\theta_{i,n}\big), \tag{7}$$

and zero otherwise. For an active pair, the negative loss gradient points in the direction of increasing the student's score gap $S^\theta_{i,p} - S^\theta_{i,n}$. Gradient descent on this term therefore encourages the student to favour the correct trajectory over the incorrect one, promoting alignment with the verifier-defined ordering. The normalised gate adaptively scales each pair's gradient contribution according to the teacher's pairwise reliability. When the teacher's preference meets the reliability threshold, the gate is zero and the pair contributes no corrective gradient. Among active pairs within a prompt, weaker teacher support for the verifier-defined ordering yields a larger gate value in the corrective update.

Computational cost. HPC reuses the teacher and student token log-probabilities computed for OPD to obtain the sequence scores $T_{i,k}$ and $S^\theta_{i,k}$. For each prompt, the number of pairs is $|\mathcal{P}_i||\mathcal{N}_i| \leq K^2/4$, since $|\mathcal{P}_i| + |\mathcal{N}_i| = K$. Given these scores, computing the pairwise objective costs $\mathcal{O}(|\mathcal{P}_i||\mathcal{N}_i|)$,

with a bound of $\mathcal{O}(K^2)$ in the worst case. The added pairwise computation incurs little overhead in practice relative to the cost of OPD's forward passes. Full pseudocode is provided in Appendix A.

3.4 Theoretical Analysis

We now examine how adding the HPC loss $\mathcal{L}_{\text{HPC}}$ to the OPD objective affects the probability of generating a trajectory that the verifier marks as incorrect. For a fixed prompt $\boldsymbol{x}_i$, define this probability as $\text{Err}_i(\pi) = \Pr_{\boldsymbol{y}\sim\pi(\cdot|\boldsymbol{x}_i)}[r(\boldsymbol{x}_i, \boldsymbol{y}) = 0]$, where $r(\boldsymbol{x}_i, \boldsymbol{y}) = 0$ indicates that the verifier marks the sampled trajectory as incorrect. Let $\beta_i^T = \text{Err}_i(\pi_T)$ denote the teacher's generation error probability. We study stationary points of the HPC objective over the complete trajectory support.

Theorem 1 (A tighter generation error bound)**.** *Under the finite complete support and interior stationarity conditions in Appendix B.1, let $\lambda > 0$ and suppose the teacher leaves a positive weighted margin deficit. Every interior stationary distribution π^λ of the HPC objective admits a correction certificate $\kappa_i^\lambda > 0$ that summarizes the corrective effect and is defined in Eq. (25), such that*

$$\text{Err}_i(\pi^\lambda) \le U_i^\lambda := \frac{\beta_i^T}{\beta_i^T + (1-\beta_i^T)e^{\kappa_i^\lambda}} < \beta_i^T = \text{Err}_i(\pi^0), \tag{8}$$

where $\pi^0 = \pi_T$ is the OPD solution.

Remark. The bound is strictly smaller than the generation error of the OPD solution, with larger correction certificates giving tighter error bounds. In particular, for any contributing pair, the teacher-ranking condition $T_{i,p} \le T_{i,n}$ guarantees the required positive margin deficit. This establishes a direct connection between correcting unreliable preferences and reducing generation error. The proof and further discussion, including the resulting Pass@k improvement, are provided in Appendix B.

4 Experiments

4.1 Experimental Setup

We use Qwen3-1.7B-Non-Thinking and Qwen3-4B-Non-Thinking (Qwen Team, 2025) as student models. We consider Qwen3-4B-Non-Thinking-RL-Math/Code as domain-specific teachers, obtained by applying GRPO (Shao et al., 2024) to Qwen3-4B-Non-Thinking, and Qwen3-30B-A3B-Instruct-2507 (Qwen Team, 2025) for strong-to-weak distillation. For training, we use 57K difficulty-≥ 6 prompts from DeepMath-103K (He et al., 2025) and 25K prompts from Eurus-RL-Code (Cui et al., 2026); the distillation prompts match the RL prompts used to construct the corresponding 4B teachers. Math outputs are verified by symbolic answer matching and code outputs by executing the associated test suites. We evaluate on AIME24, AIME25, HMMT25 (February/November), HumanEval+, MBPP+, and LiveCodeBench v6, reporting Pass@$\{16, 8, 4\}$ for math and Pass@$\{4, 2\}$ for code.

We compare HPC with OPD (Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025) and SCOPE (Zheng et al., 2026), while reporting the student before distillation and the frozen teacher for reference. All experiments are implemented with verl (Sheng et al., 2025). To isolate the effect of the objective, OPD, SCOPE, and HPC are trained for 50 epochs using the same learning rate, batch size, trajectory budget K, and optimization schedule. We set $\rho_T = \rho_S = 0.52$ and $\lambda = 1$ by default.

4.2 Results on Mathematical Reasoning

Table 1 reports Pass@$\{16, 8, 4\}$ on four math benchmarks under two teacher regimes.

Consistent gains over OPD. HPC raises average performance over OPD across student sizes and teacher regimes, with gains visible at different Pass@K levels. For instance, on AIME25, Pass@16 for the Qwen3-4B student improves from 74.6 to 76.4 with the same-size teacher and from 70.7 to 78.8 with the 30B teacher, with similar trends for Qwen3-1.7B. These gains across sampling budgets and teacher regimes suggest a systematic improvement in student performance across these settings.

Comparison with SCOPE. HPC achieves higher average scores on math benchmarks compared with SCOPE, with larger gains on harder HMMT25 tasks. For Qwen3-4B on HMMT25 (Feb.), Pass@16 improves from 51.1 to 54.2 with the same-size teacher and from 54.1 to 57.2 with the 30B teacher.

Table 1: Results on mathematical reasoning benchmarks under the same-size (Qwen3-4B-Non-Thinking-RL-Math) and strong-to-weak (Qwen3-30B-A3B-Instruct-2507) teacher settings. All values are Pass@k (%) and the best Avg within each block is in bold. Student rows are shown once per student size, as they do not depend on the teacher. Rows for our method are highlighted in gray.

Student	Method	AIME24			AIME25			HMMT25 (Feb.)			HMMT25 (Nov.)			Avg
		@16	@8	@4	@16	@8	@4	@16	@8	@4	@16	@8	@4	
Same-size setting (teacher: Qwen3-4B-Non-Thinking-RL-Math)														
Qwen3-1.7B	Student	40.0	31.6	24.5	30.8	25.4	21.2	14.6	12.4	10.0	16.6	13.3	9.6	20.8
	Teacher	82.4	80.1	75.8	75.0	71.2	67.4	56.4	49.6	44.9	60.7	56.0	50.4	64.2
	GRPO	54.0	49.6	44.6	50.2	43.8	38.1	29.6	25.3	22.3	32.4	27.0	21.9	36.6
	OPD	65.8	61.7	55.8	51.3	48.8	44.7	34.1	29.2	24.6	39.4	34.6	29.2	43.3
	SCOPE	66.6	63.0	56.9	54.1	49.8	45.7	35.9	31.3	26.6	41.0	34.1	28.7	44.5
	HPC	67.3	63.1	56.9	58.0	53.7	48.3	31.5	29.4	26.6	44.0	38.7	32.6	**45.8**
Qwen3-4B	Student	50.9	45.1	37.4	45.8	37.8	31.1	26.7	21.8	18.0	36.2	27.4	20.8	33.3
	Teacher	82.4	80.1	75.8	75.0	71.2	67.4	56.4	49.6	44.9	60.7	56.0	50.4	64.2
	GRPO	69.9	65.5	59.6	63.2	59.1	54.7	47.4	38.1	31.7	57.9	51.5	45.3	53.7
	OPD	81.3	78.6	73.8	74.6	70.9	67.5	51.1	48.3	44.7	55.8	52.2	48.2	62.3
	SCOPE	81.7	79.4	75.7	76.1	73.5	69.4	51.1	48.1	43.7	54.6	51.0	47.8	62.7
	HPC	81.6	79.9	75.9	76.4	73.0	68.7	54.2	50.1	45.7	61.1	55.9	50.0	**64.4**
Strong-to-weak setting (teacher: Qwen3-30B-A3B-Instruct-2507)														
Qwen3-1.7B	Teacher	89.9	88.5	85.9	82.9	80.3	75.6	62.4	59.4	56.2	85.6	80.7	73.7	76.8
	OPD	64.6	59.2	53.2	49.9	45.3	40.6	34.0	29.4	24.3	34.2	30.1	24.8	40.8
	SCOPE	61.9	57.2	51.3	52.5	47.7	42.4	34.6	29.5	24.7	38.4	33.5	27.5	41.8
	HPC	67.7	60.7	53.1	52.8	47.5	41.7	36.2	29.7	24.6	40.6	36.4	30.6	**43.5**
Qwen3-4B	Teacher	89.9	88.5	85.9	82.9	80.3	75.6	62.4	59.4	56.2	85.6	80.7	73.7	76.8
	OPD	81.2	78.6	74.2	70.7	67.5	63.2	55.5	50.0	43.5	62.4	58.7	53.4	63.2
	SCOPE	81.6	79.8	75.7	72.1	69.3	64.8	54.1	47.3	40.5	66.0	58.7	52.0	63.5
	HPC	83.3	80.7	76.7	78.8	72.5	65.6	57.2	52.2	45.1	63.6	58.9	53.2	**65.7**

These gains coincide with teacher–verifier disagreement in this regime, where verifier feedback guides the added correction. Unlike SCOPE, which reweights OPD supervision along separate paths, HPC adds verifier-guided correction when the teacher's pairwise preference itself is unreliable.

Across distillation regimes. With the same-size teacher, OPD already approaches teacher performance on AIME24/25, leaving limited headroom for further improvement, yet HPC still improves across all four benchmarks, with the largest gains on HMMT25 where the teacher gap remains wider. In the strong-to-weak regime, HPC likewise narrows the substantial gap to the 30B teacher for both students. These results suggest that trajectory-level contrastive supervision provides complementary structure beyond OPD, both near saturation and when the gap between the two models remains large.

4.3 Results on Code Generation

Table 2 reports Pass@$\{4, 2\}$ on the three code benchmarks. The pattern parallels the math results: HPC improves OPD across the code benchmarks under both student sizes and both teacher regimes.

Difficulty-dependent gains. HPC achieves higher average Pass@k than SCOPE on MBPP+ and the more challenging LiveCodeBench v6 across student scales and teacher regimes. In the same-size regime, Pass@4 on LiveCodeBench improves from 38.3 to 38.9 for Qwen3-1.7B and from 45.1 to 45.7 for Qwen3-4B. The observed gains on LiveCodeBench, where the teacher remains farther from ceiling, are consistent with HPC being more beneficial when teacher supervision is less reliable.

Strong-to-weak setting. With the larger Qwen3-30B-A3B-Instruct-2507 teacher, students distilled with OPD fall further behind the teacher on LiveCodeBench, suggesting that small students struggle to

Table 2: Results on code-generation benchmarks under the same-size (Qwen3-4B-Non-Thinking-RL-Code) and strong-to-weak (Qwen3-30B-A3B-Instruct-2507) teacher settings. All values are reported as Pass@k (%) and the best Avg within each block is in bold. Student rows are shown once for each student size, as they do not depend on the teacher. Rows for our method are highlighted in gray.

Student	Method	MBPP+		HumanEval+		LiveCodeBench v6		Avg
		@4	@2	@4	@2	@4	@2	
Same-size setting (teacher: Qwen3-4B-Non-Thinking-RL-Code)								
Qwen3-1.7B	Student	62.9	59.6	74.1	69.1	21.1	18.1	50.8
	Teacher	81.6	78.7	92.0	89.0	41.7	39.5	70.4
	GRPO	75.7	71.7	87.0	80.0	36.6	33.8	64.1
	OPD	79.8	73.0	88.3	83.3	36.6	35.1	66.0
	SCOPE	77.8	72.5	88.3	84.3	38.3	35.3	66.1
	HPC	78.9	73.1	90.1	84.6	38.9	36.0	**66.9**
Qwen3-4B	Student	74.6	71.1	84.0	80.7	30.3	27.6	61.4
	Teacher	81.6	78.7	92.0	89.0	41.7	39.5	70.4
	GRPO	78.9	74.5	91.4	89.5	38.9	36.8	68.3
	OPD	82.2	78.8	94.4	91.7	42.9	41.7	72.0
	SCOPE	82.5	77.3	94.4	92.7	45.1	42.3	72.4
	HPC	84.8	80.2	95.1	92.7	45.7	42.2	**73.5**
Strong-to-weak setting (teacher: Qwen3-30B-A3B-Instruct-2507)								
Qwen3-1.7B	Teacher	82.5	80.7	92.6	91.6	50.3	46.1	74.0
	OPD	76.6	71.2	83.3	78.7	32.0	27.6	61.6
	SCOPE	76.0	72.0	84.6	79.0	32.6	28.8	62.2
	HPC	78.7	72.9	85.2	81.9	34.3	31.0	**64.0**
Qwen3-4B	Teacher	82.5	80.7	92.6	91.6	50.3	46.1	74.0
	OPD	82.7	78.6	91.4	88.1	40.0	34.8	69.3
	SCOPE	82.7	78.2	93.2	88.7	44.0	39.7	71.1
	HPC	83.0	79.3	95.1	92.1	45.1	41.5	**72.7**

follow its supervision token by token. HPC narrows this gap and raises Qwen3-4B above the teacher on HumanEval+ Pass@2. These results further suggest that effective distillation depends on more than teacher capability, with verifier feedback providing complementary guidance on correctness.

4.4 Training Dynamics Analysis

Having established that HPC outperforms OPD on downstream benchmarks, we next examine whether its training dynamics match the intended mechanism. If the corrective term successfully reorders unreliable trajectory pairs, teacher–verifier disagreement should decline faster than under OPD, accompanied by a faster increase in training reward. We verify these by tracking three per-step diagnostics for Qwen3-4B distilled from Qwen3-30B-A3B-Instruct-2507 on mathematical reasoning.

Figure 3 summarizes the training dynamics. Panel (a) shows that HPC maintains higher verifier reward throughout training, yielding a persistent advantage and indicating a faster shift toward correct on-policy trajectories. Panels (b) and (c) show that HPC reduces both the number and margin of teacher inversion pairs more rapidly than OPD. Together, these results indicate that HPC reduces the frequency and severity of teacher–verifier disagreement and yields sustained gains in verifier reward.

4.5 Ablation Study

We now examine the sensitivity of HPC to its three hyperparameters: the mixing coefficient λ, the teacher reliability threshold ρ_T, and the student margin threshold ρ_S. In each experiment we use the Qwen3-1.7B student with the Qwen3-4B-Non-Thinking-RL-Math teacher, and report Pass@8 and

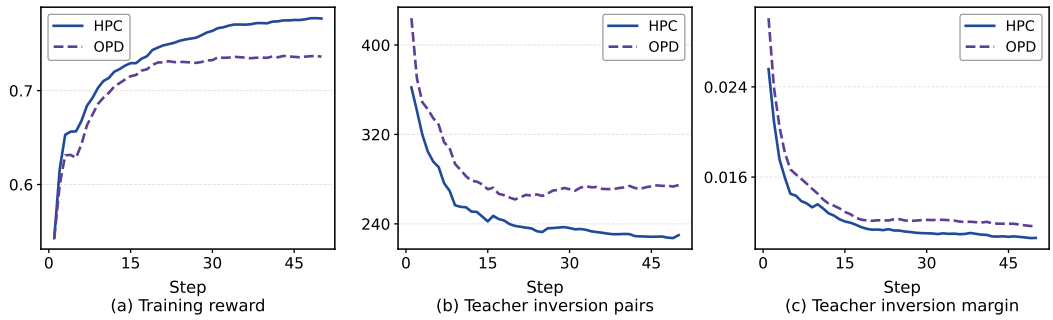
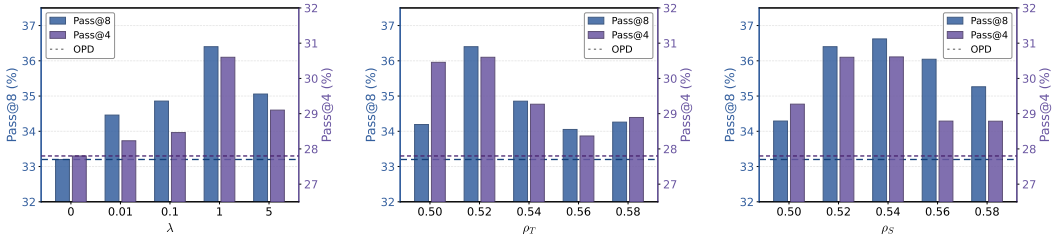


Figure 3: Training dynamics of HPC and OPD, comparing the Qwen3-4B student trained with the Qwen3-30B-A3B-Instruct-2507 teacher on mathematical reasoning. Curves use time-weighted EMA smoothing. Panel (a): mean verifier reward per step, the fraction of correct trajectories among on-policy rollouts. Panels (b) and (c): number of teacher inversion pairs, i.e., pairs for which the teacher assigns a higher score to an incorrect trajectory than to a correct one, and their mean inversion margin, i.e., the mean excess score the teacher places on the incorrect trajectory over the correct one.



(a) Varying λ. (b) Varying ρ_T. (c) Varying ρ_S.

Figure 4: Sensitivity of HPC to three hyperparameters, measured by Pass@8 and Pass@4 on HMMT25 (Nov.) (Qwen3-1.7B student, Qwen3-4B-Non-Thinking-RL-Math teacher). Dashed lines mark the OPD value for each metric; in panel (a), the leftmost bars at $\lambda = 0$ represent these values.

Pass@4 on HMMT25 (Nov.) as representatives of a coarse-grained and a strict metric. When one hyperparameter is varied, the other two are held at the default values ($\rho_T = \rho_S = 0.52$, $\lambda = 1$).

Mixing coefficient λ. Figure 4a shows that both metrics remain consistently above OPD across all tested λ values, reaching a peak at the default $\lambda = 1$. Even at $\lambda = 0.01$, our approach yields clear gains, indicating that HPC requires little corrective weighting in this range. At $\lambda = 5$, performance moves back toward OPD but remains superior, suggesting that excessive weighting weakens OPD's dense token-level guidance, while the contrastive term guided by the verifier still limits degradation.

Teacher reliability threshold ρ_T. Figure 4b shows that performance peaks at the default $\rho_T = 0.52$, improving Pass@8 from 33.2 under OPD to 36.4, while remaining above OPD across the tested range. At $\rho_T = 0.50$, the gate activates only on strictly misranked pairs, yielding a smaller but still positive correction; at larger values, it applies more broadly and may override reliable teacher supervision. The flat plateau from $\rho_T = 0.54$ to 0.58, indicates that HPC is robust to the precise choice of ρ_T.

Student margin threshold ρ_S. In Figure 4c, ρ_S controls the target confidence gap between correct and incorrect trajectories. Performance peaks at $\rho_S = 0.52$ (Pass@8 36.4) and remains above OPD throughout the tested range, including the loose setting $\rho_S = 0.5$. This suggests that the gain comes from modest corrections across many low-confidence pairs rather than aggressive updates on a few. Together with λ and ρ_T ablations, these results show the robustness of HPC *w.r.t.* hyperparameters.

Beyond numerical sensitivity, we further ablate the reliability gate, verifier noise, and pairwise preference sources. We use the Qwen3-1.7B student with the Qwen3-30B-A3B-Instruct-2507 teacher on mathematical reasoning, reporting Pass@8 and Pass@4 averaged over four benchmarks in Table 3.

Correct answer (low teacher score)

… Let $u = e^x$, so $du = e^x\,dx$. As $x \to -\infty$, $u \to 0^+$; as $x \to \infty$, $u \to \infty$. Substituting,

$$\int_{-\infty}^{\infty} \frac{e^x}{1+e^{2x}}\,dx = \int_0^{\infty} \frac{du}{1+u^2} = \frac{\pi}{2}.$$

This is the standard arctangent integral.
Final Answer: $\pi/2$.

Incorrect answer (high teacher score)

… Let $u = e^x$, so $du = e^x\,dx$. Applying this substitution to the original integral gives $\int_0^{\infty} du/(1+u^2) = \pi/2$. Since the integrand is even, we have

$$\int_{-\infty}^{\infty} \frac{e^x}{1+e^{2x}}\,dx = 2 \cdot \frac{\pi}{2} = \pi.$$

Final Answer: π.

Figure 5: A case of teacher–verifier disagreement in integral evaluation. The incorrect rollout contains a longer derivation and a familiar symmetry step, but the red factor 2 double-counts the integral because the substitution already spans the full real line. This rollout nevertheless receives a higher teacher sequence score despite the error. HPC uses verifier feedback to favour the correct rollout.

Reliability gate. The gate $w_{i,p,n} = [\rho_T - P_T]_+$ both discards pairs on which the teacher is confidently aligned with the verifier and, among the remaining pairs, assigns a larger weight when the teacher is less confident. Table 3 isolates these two effects. Dropping the gate entirely so that every pair contributes with unit weight already improves over OPD, showing that the trajectory-level contrastive term is beneficial on its own. Adding a hard binary gate that keeps only the pairs with $P_T < \rho_T$ improves further, and replacing the binary indicator with the continuous ramp of HPC lifts performance higher still. Both selection and continuous magnitude reweighting therefore contribute to the advantage of our method.

Table 3: Design ablations of HPC. Rows are grouped by the axis studied. Averages are over the four math benchmarks.

Variant	P@8	P@4	Avg.
OPD	41.00	35.73	38.36
HPC	**43.58**	**37.50**	**40.54**
Reliability gate			
None	42.08	36.73	39.41
Hard, $\mathbf{1}\{P_T < \rho_T\}$	42.54	37.33	39.94
Verifier noise			
$\varepsilon = 0.05$	43.21	37.39	40.30
$\varepsilon = 0.10$	42.94	36.98	39.96
$\varepsilon = 0.20$	42.35	36.90	39.62
Preference source			
Random	41.75	35.89	38.82
Teacher-ranked	41.65	36.64	39.14

Verifier noise. To test robustness to imperfect verifiers, we independently flip each rollout reward with probability $\varepsilon \in \{0.05, 0.10, 0.20\}$ before pair construction, so corrupted labels affect both gating and correctness supervision. As shown in Table 3, HPC degrades gracefully as verifier noise increases, retaining most of its gain at $\varepsilon = 0.05$. Even at $\varepsilon = 0.20$, it still exceeds OPD by 1.26 points on average across the benchmarks, retaining 58% of the gain achieved with clean labels. The corrective signal therefore remains broadly effective even under substantial verifier noise.

Preference source. To test the role of verifier feedback, we keep the loss unchanged but use either random splits or the teacher's sequence scores to define preferences, using unit pair weights to remove gating effects. As shown in Table 3, both variants fall at least 1.4 points below HPC with verifier labels and remain close to OPD on average. These results show that the verifier's correctness signal, rather than randomly splitting rollouts or ordering them by teacher scores, drives the improvement.

4.6 Case Study

Here, we present a case study with a representative teacher–verifier conflict from the mathematical reasoning trajectories in our experiments. The incorrect answer in Fig. 5 performs more mathematical work than the correct one, yet the reasoning still gives a wrong result and receives higher likelihood from the teacher. OPD then reinforces this preference, whereas HPC uses the verifier's ordering to favour the correct trajectory when the ranking is unreliable. More cases are provided in Appendix D.

5 Conclusion

We introduce Hard Preference Calibration (HPC), which adapts OPD to teacher reliability. HPC preserves dense guidance when teacher and verifier agree and corrects disagreements using verifier outcomes. The method reuses post-training signals without auxiliary models or annotations. Our analysis establishes a tighter generation error bound than OPD and corresponding Pass@k improvements. Across reasoning and code benchmarks, HPC outperforms OPD and related methods, especially when teacher and verifier disagree. Future work may extend this approach with learned reward models.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing and to check code for errors. We did not use generative AI tools to generate data or experimental results. We have reviewed all AI-assisted work, manually checked the final text and scientific claims, and verified the code used in the paper. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses publicly available datasets, pretrained models, and involves no human participants, personal data, or human-subject experimentation. We do not release a new dataset or collect additional user data, and we are not aware of any ethical concerns specific to the methods or experiments presented here.

REPRODUCIBILITY STATEMENT

We provide details for reproducing the experimental results in Section 4.1, including training and model details. Appendix C additionally provides the hyperparameter configuration and implementation settings, while Appendix A gives pseudocode for the proposed method. For the theoretical results, Appendix B.1 states the assumptions, and Appendices B.2 and B.3 provide complete proofs of the generation-error bound and the corresponding Pass@k improvement.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Mohammad Sadegh Akhondzadeh, Vijay Lingam, Atula Tejaswi, Chanakya Ekbote, Sujay Sanghavi, and Aleksandar Bojchevski. Reward-gated on-policy distillation. *CoRR*, abs/2607.04037, 2026. doi: 10.48550/ARXIV.2607.04037. URL https://doi.org/10.48550/arXiv.2607.04037.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li (eds.), *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4447–4455. PMLR, 2024. URL https://proceedings.mlr.press/v238/gheshlaghi-azar24a.html.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

Frank H. Clarke. *Optimization and Nonsmooth Analysis*. Society for Industrial and Applied Mathematics, 1990. doi: 10.1137/1.9781611971309. URL https://doi.org/10.1137/1.9781611971309.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006. ISBN 978-0-471-24195-9. URL http://www.elementsofinformationtheory.com/.

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. *Trans. Mach. Learn. Res.*, 2026, 2026. URL https://openreview.net/forum?id=9SkkifLopZ.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: reward ranked finetuning for generative foundation model alignment. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=m7p5O7zblY.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *CoRR*, abs/2504.11456, 2025. doi: 10.48550/ARXIV.2504.11456. URL https://doi.org/10.48550/arXiv.2504.11456.

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, Chengquan Zhang, Zhuotao Tian, Han Hu, Yi Yang, Fei Wu, and Hehe Fan. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *CoRR*, abs/2605.03677, 2026. doi: 10.48550/ARXIV.2605.03677. URL https://doi.org/10.48550/arXiv.2605.03677.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/ed873d79e7c268c020c4b4db13a2812a-Abstract-Conference.html.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *CoRR*, abs/2601.20802, 2026. doi: 10.48550/ARXIV.2601.20802. URL https://doi.org/10.48550/arXiv.2601.20802.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M. Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal M. P. Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=CjwERcAU7w.

Yuying Li, Leqi Zheng, Yongzi Yu, Wenrui Zhou, Xuchang Zhong, Xing Hu, Jing Jin, Hangjie Yuan, and Tao Feng. Filter, then reweight: Rethinking optimization granularity in on-policy distillation. *CoRR*, abs/2606.02684, 2026. doi: 10.48550/ARXIV.2606.02684. URL https://doi.org/10.48550/arXiv.2606.02684.

LLM-Core Xiaomi. Mimo-v2-flash technical report. *CoRR*, abs/2601.02780, 2026. doi: 10.48550/ARXIV.2601.02780. URL https://doi.org/10.48550/arXiv.2601.02780.

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

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/e099c1c9699814af0be873a175361713-Abstract-Conference.html.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5. URL https://doi.org/10.1007/978-0-387-40065-5.

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family. https://huggingface.co/spaces/HuggingFaceH4/on-policy-distillation, 2025.

Daniil Plyusov, Alexey Gorbatovski, Alexey Malakhov, Nikita Balagansky, Boris Shaposhnikov, Daria Korotyshova, and Daniil Gavrilov. Trust-region behavior blending for on-policy distillation. *CoRR*, abs/2605.31159, 2026. doi: 10.48550/ARXIV.2605.31159. URL https://doi.org/10.48550/arXiv.2605.31159.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN 978-0-47161977-2. doi: 10.1002/9780470316887. URL https://doi.org/10.1002/9780470316887.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*, volume 317 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1998. ISBN 978-3-540-62772-2. doi: 10.1007/978-3-642-02431-3. URL https://doi.org/10.1007/978-3-642-02431-3.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *CoRR*, abs/2601.19897, 2026. doi: 10.48550/ARXIV.2601.19897. URL https://doi.org/10.48550/arXiv.2601.19897.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL http://www.incompleteideas.net/book/first/the-book.html.

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. Openclaw-rl: Train any agent simply by talking. *CoRR*, abs/2603.10165, 2026. doi: 10.48550/ARXIV.2603.10165. URL https://doi.org/10.48550/arXiv.2603.10165.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *CoRR*, abs/2606.01249, 2026. doi: 10.48550/ARXIV.2606.01249. URL https://doi.org/10.48550/arXiv.2606.01249.

Haoran Xu, Hongyu Wang, Yifei Gao, Jiaze Li, Xiaofeng Zhang, and Xiaosong Yuan. SG-OPD: sign-gated on-policy distillation via sign-consistency gating and phased teacher sampling. *CoRR*, abs/2606.09304, 2026. doi: 10.48550/ARXIV.2606.09304. URL https://doi.org/10.48550/arXiv.2606.09304.

Wenkai Yang, Yankai Lin, Jie Zhou, and Ji-Rong Wen. Distilling rule-based knowledge into large language models. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pp. 913–932. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.61/.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *CoRR*, abs/2602.12125, 2026. doi: 10.48550/ARXIV.2602.12125. URL https://doi.org/10.48550/arXiv.2602.12125.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *CoRR*, abs/2511.10643, 2025. doi: 10.48550/ARXIV.2511.10643. URL https://doi.org/10.48550/arXiv.2511.10643.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. RRHF: rank responses to align language models with human feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/23e6f78bdec844a9f7b6c957de2aae91-Abstract-Conference.html.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/yuan25c.html.

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. *CoRR*, abs/2601.18734, 2026. doi: 10.48550/ARXIV.2601.18734. URL https://doi.org/10.48550/arXiv.2601.18734.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. SCOPE: signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *CoRR*, abs/2604.10688, 2026. doi: 10.48550/ARXIV.2604.10688. URL https://doi.org/10.48550/arXiv.2604.10688.

A METHOD SUPPLEMENT

We provide the pseudocode of the HPC loss assembly summarised in Section 3.3. Sequence scores $T_{i,k}$, $S^{\theta}_{i,k}$, gate weights $w_{i,p,n}$, and the per-pair loss $\mathcal{L}^{\mathrm{HPC}}_{i,p,n}$ follow the definitions in Section 3.2. No additional model evaluations are introduced beyond those already required by OPD.

Algorithm 1: HPC pre-computation and loss assembly.

Input: prompt batch $\{\boldsymbol{x}_i\}$, trajectories per prompt K, teacher π_T, student π_θ, hyperparameters ρ_T, ρ_S, λ.

Sample $\{\boldsymbol{y}_{i,k}\}_{k=1}^{K} \sim \pi_\theta(\cdot \mid \boldsymbol{x}_i)$;
Compute verifier rewards $r_{i,k} \in \{0, 1\}$;
for *each prompt* i **do**
  Form $\mathcal{P}_i = \{k : r_{i,k} = 1\}$ and $\mathcal{N}_i = \{k : r_{i,k} = 0\}$;
  if $|\mathcal{P}_i| = 0$ ***or*** $|\mathcal{N}_i| = 0$ **then**
    continue;
  end
  Compute teacher sequence scores $T_{i,k}$ via Eq. (2) (no gradient);
  for $(p, n) \in \mathcal{P}_i \times \mathcal{N}_i$ **do**
    $P_T \leftarrow \sigma(T_{i,p} - T_{i,n})$;
    $w_{i,p,n} \leftarrow [\rho_T - P_T]_+$;
  end
end
Compute student sequence scores $S^{\theta}_{i,k}$ via Eq. (2) (with gradient);
Aggregate the per-pair losses into $\mathcal{L}_{\mathrm{HPC}}$ using normalized gate weights;
$\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{OPD}} + \lambda \mathcal{L}_{\mathrm{HPC}}$;
Backpropagate $\nabla_\theta \mathcal{L}$;

B THEORETICAL DETAILS FOR HPC

This appendix complements Section 3.4. We first specify the distribution setting for the objective in Eq. (6), prove the generation error bound in Theorem 1, and then derive a quantitative Pass@k improvement over OPD.

B.1 SETTING AND ASSUMPTIONS

Fix a prompt $\boldsymbol{x}_i$. Let $\mathcal{Y}_i = \{\boldsymbol{y}_{i,j} : j \in \mathcal{K}_i\}$ denote its complete trajectory support, with $L_{i,j} = |\boldsymbol{y}_{i,j}| > 0$. Using the policies in Section 3.1, write $t_{i,j} = \pi_T(\boldsymbol{y}_{i,j} \mid \boldsymbol{x}_i)$ and $p_{i,j} = \pi_\theta(\boldsymbol{y}_{i,j} \mid \boldsymbol{x}_i)$ for the teacher and student trajectory probabilities. The index j ranges over all completed outputs; the K rollouts used by the method are sampled from the student policy. The verifier defines $\mathcal{P}_i$ and $\mathcal{N}_i$ on this support using the same reward rule as in Section 3.1.

For the distribution comparison, fix the contributing pair set $\mathcal{E}_i \subseteq \mathcal{P}_i \times \mathcal{N}_i$. The teacher is frozen, and the gate $w_{i,p,n}$ in Eq. (4) gives the normalised weights

$$Z_i = \sum_{(p,n)\in\mathcal{E}_i} w_{i,p,n} > 0, \qquad a_{i,p,n} = \begin{cases} w_{i,p,n}/Z_i, & (p,n) \in \mathcal{E}_i, \\ 0, & (p,n) \notin \mathcal{E}_i. \end{cases} \tag{9}$$

These weights reproduce Eq. (5) and are held constant with respect to the student probabilities during the stationarity calculation. With $m = \mathrm{logit}(\rho_S) > 0$ and $\Delta_{i,p,n}(p_i) = \log p_{i,p}/L_{i,p} - \log p_{i,n}/L_{i,n}$, the objective for this prompt is

$$H_i(p_i) = \sum_{p\in\mathcal{P}_i, n\in\mathcal{N}_i} a_{i,p,n}[m - \Delta_{i,p,n}(p_i)]_+, \tag{10}$$

$$F_{i,\lambda}(p_i) = D_{\mathrm{KL}}(p_i \| t_i) + \lambda H_i(p_i). \tag{11}$$

Here H_i is the HPC correction expressed in trajectory probabilities, and the reverse KL term is the sequence distribution objective underlying OPD (Agarwal et al., 2024; Gu et al., 2024). The analysis uses the following assumptions.

(A1) Finite complete trajectory support. The vocabulary and maximum generation length are finite. $\mathcal{Y}_i$ contains every completed trajectory under a common termination rule, ending at the termination token or the generation limit. Both verifier classes are nonempty, and $\pi_T(\boldsymbol{y}_{i,j} \mid \boldsymbol{x}_i) > 0$ for every $j \in \mathcal{K}_i$.

(A2) Interior stationarity. For $\lambda > 0$, p_i^λ is a constrained Clarke stationary distribution of $F_{i,\lambda}$ in the interior of the probability simplex. Thus $p_{i,j}^\lambda > 0$ for every j and $\sum_j p_{i,j}^\lambda = 1$, with stationarity taken with respect to the trajectory probabilities under this normalisation constraint.

(A3) Positive weighted margin deficit. The teacher leaves a positive weighted margin deficit, $H_i(t_i) > 0$.

Discussion of the assumptions. Assumption (A1) uses the finite horizon representation of policy models (Puterman, 1994; Sutton & Barto, 1998). The common termination rule defines a complete probability space, and positive teacher probabilities give finite logarithms in the scores and reverse KL. The probability factorisation follows the chain rule (Cover & Thomas, 2006). The score retains the mean log-probability normalisation in Eq. (2), also used in related preference objectives (Meng et al., 2024; Yuan et al., 2023).

Assumption (A2) uses the standard framework of constrained nonsmooth optimisation (Clarke, 1990; Rockafellar & Wets, 1998; Nocedal & Wright, 2006). Clarke stationarity accommodates the change in slope at the target margin and yields the equations relating the correction to the student probabilities. Assumption (A3) identifies teacher preferences that activate the correction at the OPD solution. A contributing pair with $T_{i,p} \leq T_{i,n}$ satisfies this condition under the thresholds in Eqs. (4) and (5); the verification is given in Eq. (17). Together, these conditions connect correcting the teacher's unreliable preferences to the strict generation error bound in Theorem 1.

B.2 Proof of Theorem 1

We derive the error bound by first comparing the total probabilities of the correct and incorrect classes. The stationarity equations give opposite likelihood adjustments on the two classes. A decomposition of their log-odds change and two applications of Cauchy–Schwarz then yield the certificate in Eq. (25).

Proof. We work under Assumptions (A1)–(A3).

Trajectory probabilities and the OPD solution. Under (A1), the chain rule gives

$$t_{i,j} = \pi_T(\boldsymbol{y}_{i,j} \mid \boldsymbol{x}_i) = \prod_{s=1}^{L_{i,j}} \pi_T(y_{i,j,s} \mid \boldsymbol{x}_i, \boldsymbol{y}_{i,j,<s}), \tag{12}$$

$$p_{i,j} = \pi_\theta(\boldsymbol{y}_{i,j} \mid \boldsymbol{x}_i) = \prod_{s=1}^{L_{i,j}} \pi_\theta(y_{i,j,s} \mid \boldsymbol{x}_i, \boldsymbol{y}_{i,j,<s}), \tag{13}$$

Here s indexes token positions, corresponding to the token index t in Eq. (2); $t_{i,j}$ is the teacher probability of a complete trajectory, while $y_{i,j,s}$ is a token. Each probability includes every token factor required by the termination rule. The completed outputs are mutually exclusive and exhaustive, so

$$\sum_{j \in \mathcal{K}_i} t_{i,j} = \sum_{j \in \mathcal{K}_i} p_{i,j} = 1. \tag{14}$$

The normalisation is over the complete trajectory support. Taking logarithms in Eqs. (12) and (13) and dividing by the trajectory length gives the sequence scores in Eq. (2),

$$T_{i,j} = \frac{\log t_{i,j}}{L_{i,j}}, \qquad S_{i,j}^\theta = \frac{\log p_{i,j}}{L_{i,j}}. \tag{15}$$

Thus the analysis retains the method's length normalisation while allowing unequal trajectory lengths. Substituting the trajectory factorisation into the reverse KL gives

$$D_{\mathrm{KL}}(p_i \| t_i) = \sum_j p_{i,j} \log \frac{p_{i,j}}{t_{i,j}} = \mathbb{E}_{\boldsymbol{y} \sim \pi_\theta(\cdot|\boldsymbol{x}_i)} \left[\sum_{s=1}^{|\boldsymbol{y}|} \log \frac{\pi_\theta(y_s \mid \boldsymbol{x}_i, \boldsymbol{y}_{<s})}{\pi_T(y_s \mid \boldsymbol{x}_i, \boldsymbol{y}_{<s})} \right]. \tag{16}$$

Its unique minimiser is $p_i^0 = t_i$, by nonnegativity of KL and its equality condition (Cover & Thomas, 2006).

The teacher ranking condition in the main text implies (A3). Indeed, if $(p, n) \in \mathcal{E}_i$ satisfies $T_{i,p} \leq T_{i,n}$, then

$$w_{i,p,n} \geq \rho_T - \tfrac{1}{2} > 0, \qquad H_i(t_i) \geq a_{i,p,n}[m - (T_{i,p} - T_{i,n})]_+ \geq a_{i,p,n} m > 0. \tag{17}$$

We now use $H_i(t_i) > 0$ to establish the strict improvement.

Stationarity equations. Under (A2), the locally Lipschitz objective admits constrained Clarke stationarity equations. For each pair, a hinge multiplier at p_i^λ satisfies

$$\xi_{i,p,n} \in \begin{cases} \{1\}, & \Delta_{i,p,n}(p_i^\lambda) < m, \\ \{0\}, & \Delta_{i,p,n}(p_i^\lambda) > m, \\ [0,1], & \Delta_{i,p,n}(p_i^\lambda) = m. \end{cases} \tag{18}$$

Each hinge is the maximum of zero and a smooth function. Its subdifferential therefore has this multiplier representation. Stationarity means that there exists a simultaneous choice of these multipliers and a scalar ν_i satisfying the simplex stationarity equations derived below. Fix any such choice and define

$$B_{i,p} = \sum_{n \in \mathcal{N}_i} a_{i,p,n} \xi_{i,p,n}, \qquad C_{i,n} = \sum_{p \in \mathcal{P}_i} a_{i,p,n} \xi_{i,p,n}. \tag{19}$$

The endpoint weights are nonnegative, with $\sum_p B_{i,p} = \sum_n C_{i,n} = \sum_{p,n} a_{i,p,n} \xi_{i,p,n}$.

Introduce the Lagrangian

$$\mathcal{L}_i(p_i, \nu_i) = F_{i,\lambda}(p_i) + \nu_i \left(\sum_j p_{i,j} - 1 \right).$$

For a single pair, the selected subgradient of $[m - \Delta_{i,p,n}(p_i)]_+$ has correct and incorrect coordinates $-\xi_{i,p,n}/(L_{i,p} p_{i,p})$ and $\xi_{i,p,n}/(L_{i,n} p_{i,n})$, respectively. Summing their weighted contributions gives

$$(\partial H_i)_{i,p} = -\frac{B_{i,p}}{L_{i,p} p_{i,p}^\lambda}, \qquad (\partial H_i)_{i,n} = \frac{C_{i,n}}{L_{i,n} p_{i,n}^\lambda}.$$

Since $\frac{\partial}{\partial p_{i,j}} D_{\mathrm{KL}}(p_i \| t_i) = \log(p_{i,j}/t_{i,j}) + 1$, stationarity yields

$$\log \frac{p_{i,p}^\lambda}{t_{i,p}} + 1 + \nu_i - \frac{\lambda B_{i,p}}{L_{i,p} p_{i,p}^\lambda} = 0, \qquad p \in \mathcal{P}_i, \tag{20}$$

$$\log \frac{p_{i,n}^\lambda}{t_{i,n}} + 1 + \nu_i + \frac{\lambda C_{i,n}}{L_{i,n} p_{i,n}^\lambda} = 0, \qquad n \in \mathcal{N}_i. \tag{21}$$

Let

$$\gamma_i = -1 - \nu_i, \qquad u_{i,p} = \frac{\lambda B_{i,p}}{L_{i,p} p_{i,p}^\lambda} \geq 0, \qquad v_{i,n} = \frac{\lambda C_{i,n}}{L_{i,n} p_{i,n}^\lambda} \geq 0. \tag{22}$$

Then

$$\log \frac{p_{i,p}^\lambda}{t_{i,p}} = \gamma_i + u_{i,p}, \qquad \log \frac{p_{i,n}^\lambda}{t_{i,n}} = \gamma_i - v_{i,n}. \tag{23}$$

The endpoint weights cannot all vanish. Otherwise, Eq. (23) would imply $p^{\lambda}_{i,j} = e^{\gamma_i} t_{i,j}$ for every j. Summing over the complete support would give $e^{\gamma_i} = 1$ and $p^{\lambda}_i = t_i$. But $H_i(t_i) > 0$, as a finite sum of nonnegative terms, implies the existence of a pair with $a_{i,p,n} > 0$ and $\Delta_{i,p,n}(t_i) < m$. At this pair, $\xi_{i,p,n} = 1$, which forces both corresponding endpoint weights to be positive, a contradiction. Equality of the two total endpoint weights therefore implies that at least one $u_{i,p}$ and at least one $v_{i,n}$ are strictly positive.

Class probabilities and the error bound. The coordinates of $\pi^{\lambda}(\cdot \mid \boldsymbol{x}_i)$ are denoted p^{λ}_i. In these coordinates the error already defined in Section 3.4 is $\mathrm{Err}_i(p_i) = \sum_{n \in \mathcal{N}_i} p_{i,n}$. Write

$$\begin{aligned} \alpha^T_i &= \sum_{p \in \mathcal{P}_i} t_{i,p} = 1 - \beta^T_i, \quad & \beta^T_i &= \sum_{n \in \mathcal{N}_i} t_{i,n}, \\ \alpha^{\lambda}_i &= \sum_{p \in \mathcal{P}_i} p^{\lambda}_{i,p}, & \beta^{\lambda}_i &= \sum_{n \in \mathcal{N}_i} p^{\lambda}_{i,n} = 1 - \alpha^{\lambda}_i. \end{aligned} \tag{24}$$

All four class probabilities lie in $(0, 1)$. The certificate in Theorem 1 is computed as

$$\begin{aligned} d^{\lambda}_i &= \frac{1}{\alpha^T_i} \left(\sum_{p \in \mathcal{P}_i} \sqrt{\frac{t_{i,p} B_{i,p}}{L_{i,p}}} \right)^2, \\ b^{\lambda}_i &= \sum_{n \in \mathcal{N}_i} \frac{C_{i,n}}{L_{i,n}}, \\ \kappa^{\lambda}_i &= \lambda \big(\sqrt{d^{\lambda}_i} + \sqrt{b^{\lambda}_i} \big)^2. \end{aligned} \tag{25}$$

It depends on the stationary distribution through the hinge multipliers. The factors $1/L_{i,j}$ come from differentiating the sequence scores.

Define the teacher and student distributions conditioned on each verifier class by

$$\tau^{+}_{i,p} = \frac{t_{i,p}}{\alpha^T_i}, \quad r^{+}_{i,p} = \frac{p^{\lambda}_{i,p}}{\alpha^{\lambda}_i}, \qquad \tau^{-}_{i,n} = \frac{t_{i,n}}{\beta^T_i}, \quad r^{-}_{i,n} = \frac{p^{\lambda}_{i,n}}{\beta^{\lambda}_i}. \tag{26}$$

Each distribution sums to one within its own class. Exponentiating Eq. (23) gives

$$p^{\lambda}_{i,p} = e^{\gamma_i} t_{i,p} e^{u_{i,p}}, \qquad p^{\lambda}_{i,n} = e^{\gamma_i} t_{i,n} e^{-v_{i,n}}. \tag{27}$$

Consequently, if

$$A_{i,+} = \sum_{p \in \mathcal{P}_i} \tau^{+}_{i,p} e^{u_{i,p}}, \qquad A_{i,-} = \sum_{n \in \mathcal{N}_i} \tau^{-}_{i,n} e^{-v_{i,n}}, \tag{28}$$

then summation of Eq. (27) within each class yields

$$\alpha^{\lambda}_i = e^{\gamma_i} \alpha^T_i A_{i,+}, \qquad \beta^{\lambda}_i = e^{\gamma_i} \beta^T_i A_{i,-}, \qquad \frac{\alpha^{\lambda}_i / \beta^{\lambda}_i}{\alpha^T_i / \beta^T_i} = \frac{A_{i,+}}{A_{i,-}}. \tag{29}$$

The common simplex multiplier has cancelled. Moreover, $A_{i,+} > 1$ and $0 < A_{i,-} < 1$, since both teacher conditional distributions are positive and the corresponding tilts contain strictly positive entries. Dividing each equality in Eq. (27) by its class mass in Eq. (29) gives

$$\begin{aligned} r^{+}_{i,p} &= \frac{e^{\gamma_i} t_{i,p} e^{u_{i,p}}}{e^{\gamma_i} \alpha^T_i A_{i,+}} = \frac{\tau^{+}_{i,p} e^{u_{i,p}}}{A_{i,+}}, \\ r^{-}_{i,n} &= \frac{e^{\gamma_i} t_{i,n} e^{-v_{i,n}}}{e^{\gamma_i} \beta^T_i A_{i,-}} = \frac{\tau^{-}_{i,n} e^{-v_{i,n}}}{A_{i,-}}. \end{aligned} \tag{30}$$

These identities yield an exact decomposition of the log-odds change. Indeed,

$$D_{\mathrm{KL}}(\tau_i^+ \| r_i^+) = \sum_p \tau_{i,p}^+ \log \frac{\tau_{i,p}^+}{r_{i,p}^+} = \sum_p \tau_{i,p}^+ (\log A_{i,+} - u_{i,p})$$
$$= \log A_{i,+} - \sum_p \tau_{i,p}^+ u_{i,p}, \tag{31}$$

$$D_{\mathrm{KL}}(r_i^- \| \tau_i^-) = \sum_n r_{i,n}^- \log \frac{r_{i,n}^-}{\tau_{i,n}^-} = \sum_n r_{i,n}^- (-v_{i,n} - \log A_{i,-})$$
$$= -\sum_n r_{i,n}^- v_{i,n} - \log A_{i,-}. \tag{32}$$

Rearranging and adding gives

$$\begin{aligned} \Gamma_i^\lambda &:= \log \frac{\alpha_i^\lambda / \beta_i^\lambda}{\alpha_i^T / \beta_i^T} = \log A_{i,+} - \log A_{i,-} \\ &= \sum_p \tau_{i,p}^+ u_{i,p} + \sum_n r_{i,n}^- v_{i,n} + D_{\mathrm{KL}}(\tau_i^+ \| r_i^+) + D_{\mathrm{KL}}(r_i^- \| \tau_i^-) \\ &= \frac{\lambda}{\alpha_i^T} \sum_p \frac{t_{i,p} B_{i,p}}{L_{i,p} p_{i,p}^\lambda} + \frac{\lambda}{\beta_i^\lambda} \sum_n \frac{C_{i,n}}{L_{i,n}} \\ &\quad + D_{\mathrm{KL}}(\tau_i^+ \| r_i^+) + D_{\mathrm{KL}}(r_i^- \| \tau_i^-). \end{aligned} \tag{33}$$

The last equality substitutes Eq. (22) and Eq. (26); in particular, $r_{i,n}^- v_{i,n} = \lambda C_{i,n} / (\beta_i^\lambda L_{i,n})$.

To bound the first sum in Eq. (33), Cauchy–Schwarz gives

$$\left(\sum_p \sqrt{\frac{t_{i,p} B_{i,p}}{L_{i,p}}} \right)^2 = \left(\sum_p \sqrt{\frac{t_{i,p} B_{i,p}}{L_{i,p} p_{i,p}^\lambda}} \sqrt{p_{i,p}^\lambda} \right)^2$$
$$\leq \left(\sum_p \frac{t_{i,p} B_{i,p}}{L_{i,p} p_{i,p}^\lambda} \right) \left(\sum_p p_{i,p}^\lambda \right)$$
$$= \alpha_i^\lambda \sum_p \frac{t_{i,p} B_{i,p}}{L_{i,p} p_{i,p}^\lambda}. \tag{34}$$

Division by $\alpha_i^T \alpha_i^\lambda > 0$, followed by the definition of d_i^λ, implies

$$\frac{1}{\alpha_i^T} \sum_p \frac{t_{i,p} B_{i,p}}{L_{i,p} p_{i,p}^\lambda} \geq \frac{1}{\alpha_i^T \alpha_i^\lambda} \left(\sum_p \sqrt{\frac{t_{i,p} B_{i,p}}{L_{i,p}}} \right)^2 = \frac{d_i^\lambda}{\alpha_i^\lambda}. \tag{35}$$

Both KL terms in Eq. (33) are nonnegative. Hence

$$\Gamma_i^\lambda \geq \lambda \left(\frac{d_i^\lambda}{\alpha_i^\lambda} + \frac{b_i^\lambda}{\beta_i^\lambda} \right). \tag{36}$$

A second application of Cauchy–Schwarz, now using $\alpha_i^\lambda + \beta_i^\lambda = 1$, gives

$$\left(\sqrt{d_i^\lambda} + \sqrt{b_i^\lambda} \right)^2 = \left(\sqrt{\frac{d_i^\lambda}{\alpha_i^\lambda}} \sqrt{\alpha_i^\lambda} + \sqrt{\frac{b_i^\lambda}{\beta_i^\lambda}} \sqrt{\beta_i^\lambda} \right)^2$$
$$\leq \left(\frac{d_i^\lambda}{\alpha_i^\lambda} + \frac{b_i^\lambda}{\beta_i^\lambda} \right) (\alpha_i^\lambda + \beta_i^\lambda)$$
$$= \frac{d_i^\lambda}{\alpha_i^\lambda} + \frac{b_i^\lambda}{\beta_i^\lambda}. \tag{37}$$

At least one endpoint weight in each class is positive. Since the teacher probabilities and trajectory lengths are positive, both d_i^λ and b_i^λ are positive. Combining Eq. (36) and Eq. (37) therefore yields

$$\Gamma_i^\lambda \geq \lambda\big(\sqrt{d_i^\lambda} + \sqrt{b_i^\lambda}\big)^2 = \kappa_i^\lambda > 0. \tag{38}$$

Because the exponential function is strictly increasing, Eq. (38) implies

$$\frac{\alpha_i^\lambda/\beta_i^\lambda}{\alpha_i^T/\beta_i^T} \geq e^{\kappa_i^\lambda}, \qquad \frac{\alpha_i^\lambda}{\beta_i^\lambda} \geq \frac{\alpha_i^T}{\beta_i^T} e^{\kappa_i^\lambda}. \tag{39}$$

Using $\alpha_i^\lambda = 1 - \beta_i^\lambda$, we obtain

$$\begin{aligned} \frac{1}{\beta_i^\lambda} - 1 &\geq \frac{\alpha_i^T}{\beta_i^T} e^{\kappa_i^\lambda}, \\ \frac{1}{\beta_i^\lambda} &\geq \frac{\beta_i^T + \alpha_i^T e^{\kappa_i^\lambda}}{\beta_i^T}, \\ \beta_i^\lambda &\leq \frac{\beta_i^T}{\beta_i^T + \alpha_i^T e^{\kappa_i^\lambda}} = U_i^\lambda. \end{aligned} \tag{40}$$

The final inequality takes reciprocals of positive quantities. Finally, $e^{\kappa_i^\lambda} > 1$ and $\alpha_i^T + \beta_i^T = 1$ imply $\beta_i^T + \alpha_i^T e^{\kappa_i^\lambda} > 1$, so $U_i^\lambda < \beta_i^T$. By Eq. (24), these are the errors in Eq. (8), with $\alpha_i^T = 1 - \beta_i^T$. The tightening is explicitly

$$\begin{aligned} g_i^\lambda &= \beta_i^T - U_i^\lambda \\ &= \frac{\beta_i^T\big(\beta_i^T + \alpha_i^T e^{\kappa_i^\lambda} - 1\big)}{\beta_i^T + \alpha_i^T e^{\kappa_i^\lambda}} \\ &= \frac{\alpha_i^T \beta_i^T (e^{\kappa_i^\lambda} - 1)}{\beta_i^T + \alpha_i^T e^{\kappa_i^\lambda}} > 0. \end{aligned} \tag{41}$$

□

B.3 Pass@k improvement over OPD

The error bound also controls the probability of obtaining a correct response through repeated sampling. For a fixed prompt and k independent draws from the same policy, define

$$\text{Pass@}\,k(\pi) = \Pr\left[\max_{1\leq \ell \leq k} r(\boldsymbol{x}_i, \boldsymbol{Y}^{(\ell)}) = 1\right], \qquad \boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(k)} \overset{\text{iid}}{\sim} \pi(\cdot \mid \boldsymbol{x}_i). \tag{42}$$

This is the population sampling probability underlying Pass@k (Chen et al., 2021). The evaluation count k is distinct from the number K of training rollouts.

Theorem 2 (Quantitative Pass@k improvement)**.** *Under the conditions of Theorem 1, for every integer $k \geq 1$ and independent evaluation draws,*

$$\text{Pass@}\,k(\pi^\lambda) - \text{Pass@}\,k(\pi^0) \geq (\beta_i^T)^k - (U_i^\lambda)^k > 0. \tag{43}$$

In particular, with $g_i^\lambda = \beta_i^T - U_i^\lambda > 0$,

$$\text{Pass@}\,k(\pi^\lambda) - \text{Pass@}\,k(\pi^0) \geq k(U_i^\lambda)^{k-1} g_i^\lambda > 0. \tag{44}$$

Proof. For any fixed policy, each draw fails with probability $\text{Err}_i(\pi)$. Independence gives

$$\begin{aligned} \Pr\left[\bigcap_{\ell=1}^{k} \{r(\boldsymbol{x}_i, \boldsymbol{Y}^{(\ell)}) = 0\}\right] &= \prod_{\ell=1}^{k} \Pr[r(\boldsymbol{x}_i, \boldsymbol{Y}^{(\ell)}) = 0] \\ &= \text{Err}_i(\pi)^k. \end{aligned} \tag{45}$$

Taking the complementary event therefore yields

$$\text{Pass@}\,k(\pi) = 1 - \text{Err}_i(\pi)^k. \tag{46}$$

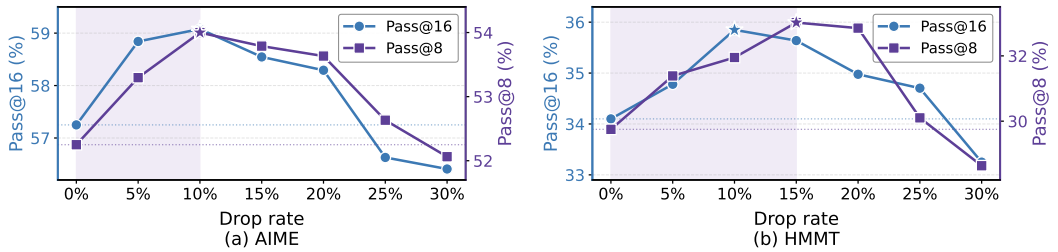


Figure 6: Diagnostic on teacher–verifier disagreement at the pair level. We train OPD (Qwen3-1.7B student, Qwen3-4B-Non-Thinking-RL-Math teacher) after removing the least-confident correct–incorrect pairs ranked by the teacher's preference for the correct trajectory. Panel (a) averages AIME24 and AIME25, while panel (b) averages HMMT25 (Feb.) and HMMT25 (Nov.). A modest drop rate improves both Pass@16 and Pass@8 across the two benchmark families, while heavier dropping reduces the gains.

By Theorem 1, $0 < \beta_i^\lambda = \mathrm{Err}_i(\pi^\lambda) \le U_i^\lambda < \beta_i^T = \mathrm{Err}_i(\pi^0) < 1$. Since $z \mapsto z^k$ is strictly increasing on $(0,1)$,

$$\begin{aligned}\mathrm{Pass@}\,k(\pi^\lambda) - \mathrm{Pass@}\,k(\pi^0) &= \big[1-(\beta_i^\lambda)^k\big] - \big[1-(\beta_i^T)^k\big] \\ &= (\beta_i^T)^k - (\beta_i^\lambda)^k \\ &\ge (\beta_i^T)^k - (U_i^\lambda)^k > 0, \end{aligned} \tag{47}$$

which proves Eq. (43). To express the gain in terms of the tightening g_i^λ, factor the difference of powers,

$$\begin{aligned}(\beta_i^T)^k - (U_i^\lambda)^k &= (\beta_i^T - U_i^\lambda)\sum_{j=0}^{k-1}(\beta_i^T)^{k-1-j}(U_i^\lambda)^j \\ &\ge g_i^\lambda \sum_{j=0}^{k-1}(U_i^\lambda)^{k-1-j}(U_i^\lambda)^j \\ &= k(U_i^\lambda)^{k-1} g_i^\lambda > 0. \end{aligned} \tag{48}$$

The inequality uses $\beta_i^T > U_i^\lambda > 0$ and the nonnegative exponent $k-1-j$ in each summand. This proves Eq. (44), including $k=1$. □

Remark. Theorem 2 translates the error bound into a quantitative sampling advantage under the same distribution conditions as Theorem 1. In particular, $k=1$ gives a success probability increase of at least g_i^λ; for any finite k, the bound gives a strictly positive reduction in the probability that all draws fail.

C Experimental Supplement

General experimental setup. We use the verl framework with AdamW as the optimizer and sample eight rollouts per prompt with temperature 1.0 and top-p 1.0, and a maximum response length of 16,384 tokens. Training uses a learning rate of 1×10^{-5}, a batch size of 128, a mini-batch size of 1,024, the low-variance KL formulation with zero KL-loss coefficient. Training is performed on NVIDIA H800 80GB GPUs.

Pair-filtering diagnostic. To complement the training-dynamics evidence in Section 1, we rank correct–incorrect trajectory pairs by the teacher's preference for the correct response and remove the least-confident pairs from the OPD loss. As shown in Fig. 6, a modest drop rate improves both Pass@16 and Pass@8, whereas heavier dropping degrades performance. The least-confident pairs therefore introduce misleading supervision, but discarding too many also removes useful trajectories. This motivates retaining these pairs while correcting them with a verifier-driven pairwise signal.

D CASE STUDIES OF TEACHER MIS-RANKINGS

To illustrate the failure mode identified in Section 1, we present several cases in which the teacher assigns a higher sequence score to an incorrect answer than to a correct answer for the same prompt. These cases are drawn from student rollouts collected during OPD training of Qwen3-1.7B on mathematical reasoning, using either the Qwen3-30B-A3B-Instruct-2507 or the Qwen3-4B-Non-Thinking-RL-Math teacher.

We organize these cases into three illustrative patterns of teacher mis-ranking and present key excerpts for each. The examples show instances in which the teacher's sequence score favors a longer or more templated derivation over a shorter correct solution, despite a local mathematical error. These cases exemplify the teacher–verifier disagreement that HPC is designed to address by using its reliability gate to scale verifier-guided correction.

D.1 PATTERN A: MISCITED CANONICAL IDENTITY

Question. $X, Y \overset{\text{iid}}{\sim} \mathrm{Bin}(n, p)$; compute $\mathbb{E}[X \mid X + Y = m]$.

Correct answer (low teacher score)

… We use the property of conditional expectation for two independent and identically distributed variables. Since X and Y are exchangeable, we have $\mathbb{E}[X \mid X + Y = m] = \mathbb{E}[Y \mid X + Y = m]$. Adding the two conditional expectations gives $\mathbb{E}[X + Y \mid X + Y = m] = m$, and by symmetry each of $\mathbb{E}[X \mid X + Y = m]$ and $\mathbb{E}[Y \mid X + Y = m]$ contributes half. Hence the answer is $\frac{m}{2}$.

Final Answer: $\boxed{\frac{m}{2}}$.

Incorrect answer (high teacher score)

… We use the joint distribution of X and Y to derive the conditional pmf. For any x with $0 \le x \le m$, $\mathbb{P}(X{=}x, Y{=}m{-}x) = \binom{n}{x}\binom{n}{m-x} p^m (1-p)^{2n-m}$. Summing the numerator over x and applying Vandermonde's identity $\sum_{x=0}^{m} \binom{n}{x}\binom{n}{m-x} = \binom{2n}{m}$, the constants in p cancel and the conditional pmf simplifies to $\mathbb{P}(X{=}x \mid X{+}Y{=}m) = \binom{n}{x}\binom{n}{m-x} / \binom{2n}{m}$, so

$$\mathbb{E}[X \mid X{+}Y{=}m] = \frac{1}{\binom{2n}{m}} \sum_{x=0}^{m} x \binom{n}{x}\binom{n}{m-x}.$$

Using the identity $\sum_{x=0}^{m} x \binom{n}{x}\binom{n}{m-x} = \frac{m}{2n}\binom{2n}{m}$, the expected value simplifies to $\frac{m}{2n}$…

Final Answer: $\boxed{\frac{m}{2n}}$.

The correct answer closes in a few sentences by symmetry, without invoking any combinatorial identity. The incorrect one derives the conditional pmf through Vandermonde's identity and closes with a memorised moment identity in which a factor of n has been dropped: the true identity gives $(m/2)\binom{2n}{m}$, not $(m/(2n))\binom{2n}{m}$, and the resulting $m/(2n)$ additionally fails the elementary sanity check that $\mathbb{E}[X \mid X{+}Y{=}m]$ cannot depend on n. The chain *Vandermonde → hypergeometric conditional pmf → moment identity* is a standard closing template in expository math text, and the specific coefficient inside the memorised identity is difficult to verify from local likelihood; a plausible reading is that these two effects together dominate the teacher's sequence-level preference on this pair.

D.2 PATTERN B: LOCAL SLIP INSIDE AN ELABORATED VALID PROCEDURE

Question. Let X, Y, Z have joint density $f(x, y, z) = e^{-x-y-z}$ for $x, y, z > 0$ (and 0 otherwise); compute $P(X < Y < Z)$.

Correct answer (low teacher score)

The density factors as $e^{-x} \cdot e^{-y} \cdot e^{-z}$ over $x, y, z > 0$, so X, Y, Z are i.i.d. $\mathrm{Exp}(1)$. Because their marginals are continuous, the joint distribution is symmetric under any permutation of (X, Y, Z), and the six orderings of the triple are equiprobable events that together partition the probability-one set of distinct triples. Hence each ordering, including $X < Y < Z$, has probability $1/3! = 1/6$.

Final Answer: $\boxed{\frac{1}{6}}$.

Incorrect answer (high teacher score)

The event $X < Y < Z$ integrates f over $\{(x, y, z) : 0 < x < y < z\}$, giving $P(X<Y<Z) = \int_0^\infty \int_x^\infty \int_y^\infty e^{-x-y-z}\, dz\, dy\, dx$.

The inner integral evaluates as $\int_y^\infty e^{-z}\, dz = e^{-y}$, and combining with the e^{-y} from the density yields, at the next step,

$$\int_0^\infty \int_x^\infty e^{-y} \cdot e^{-y}\, dy\, dx \;=\; \int_0^\infty \int_x^\infty e^{-2y}\, dy\, dx.$$

Evaluating the two remaining integrals in order, $\int_x^\infty e^{-2y}\, dy = \frac{1}{2}e^{-2x}$ and $\int_0^\infty \frac{1}{2}e^{-2x}\, dx = \frac{1}{4}\cdots$

Final Answer: $\boxed{\frac{1}{4}}$.

The correct answer is a short symmetry argument. The incorrect one sets up the same triple integral correctly, evaluates the inner integral correctly, and then drops the outer e^{-x} factor at the next substitution step: the integrand should be $e^{-x} \cdot e^{-y} \cdot e^{-y} = e^{-x-2y}$, not e^{-2y}. The dropped factor propagates through and yields $1/4$ instead of $1/6$. The elaborate triple-integral setup is a common route for probability-of-ordering problems, whereas the correct answer's short symmetry proof is a shorter, less templated route; the incorrect trajectory is also markedly longer than the correct one. Among the pairs we inspected, this configuration produces the most extreme teacher–verifier disagreement, and correspondingly the largest corrective weight assigned by HPC.

D.3 Pattern C: Local sign slip in a canonical series derivation

Question. Find a closed form for $S_a = \sum_{n=1}^\infty \ln(n)/n^a$ for $a > 1$.

Correct answer (low teacher score)

We recall the Riemann zeta function

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s}, \qquad s > 1.$$

Differentiating term by term and using $\frac{d}{ds} n^{-s} = -\ln(n) n^{-s}$ gives

$$\zeta'(s) = -\sum_{n=1}^{\infty} \frac{\ln(n)}{n^s}.$$

Setting $s = a$ therefore yields $S_a = -\zeta'(a)$.

Final Answer: $\boxed{-\zeta'(a)}$.

Incorrect answer (high teacher score)

We recall the same series representation and differentiate term by term:

$$\zeta'(s) = \sum_{n=1}^{\infty} \frac{d}{ds}\left(n^{-s}\right).$$

The derivative is then written as $\frac{\ln(n)}{n^s}$, without the minus sign, giving

$$\zeta'(s) = \sum_{n=1}^{\infty} \frac{\ln(n)}{n^s}, \qquad S_a = \zeta'(a).$$

Final Answer: $\boxed{\zeta'(a)}$.

The correct answer identifies the zeta-series representation and carries the derivative sign through to the final expression. The incorrect answer follows the same canonical series derivation, but uses $\frac{d}{ds}n^{-s} = \ln(n)n^{-s}$ instead of the correct negative derivative, reversing the result. The mistake is easy to miss in a fluent derivation because all surrounding formulas remain familiar, which is exactly the kind of teacher mis-ranking that the verifier can expose and HPC is designed to address.

D.4 Additional cases

Beyond the three patterns above, the failure modes we observed recombine the same underlying tendencies: a canonical closing template invoked with a memorised but incorrect coefficient, an unnecessary transformation appended to an already complete derivation, and a single-token slip inside an otherwise long and plausible calculation. Across these cases, the correct answer is typically either markedly shorter than the incorrect one or takes a less templated route, for instance a symmetry shortcut in place of a lengthy derivation, or a divergence-theorem argument instead of integrating over each face separately, and the teacher prefers the more templated trajectory. These cases expose a mismatch between the teacher-derived OPD signal and the verifier's ordering. HPC retains the OPD signal while using its reliability gate to scale a verifier-guided correction on pairs with weak teacher support. Sequence-level teacher likelihood tends to track apparent effort and template conformity, and it is precisely on these pairs that HPC substitutes a reliable signal driven by the verifier.