ON-POLICY DISTILLATION WITH NEGATIVE-POLICY ROLLOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

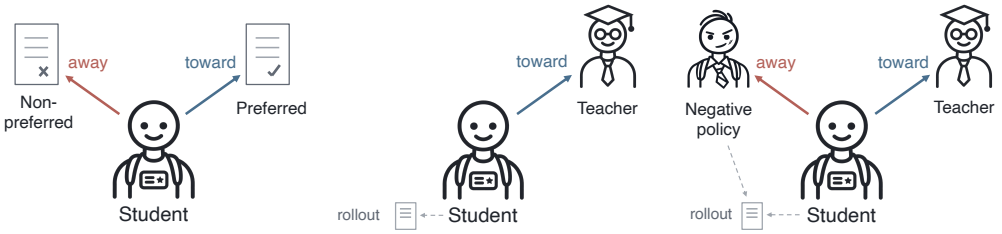
On-policy distillation (OPD) has been widely studied as a post-training method in which a student model obtains token-level supervision from a stronger teacher on its own rollouts. Recent studies have improved OPD through alternative distillation reward formulations and teacher configurations, while the objective of distillation remains centered on mimicking the teacher. However, when a stronger teacher has limited distributional overlap with the student, such positive guidance can provide insufficient learning signals. In this work, we introduce **N**egative-**P**olicy **OPD** (**NP-OPD**), which complements teacher supervision with rollouts from a lower-performing, lower-capability negative policy that serves as a negative reference for the student. Rather than modifying the distillation reward formulation, NP-OPD introduces the negative policy at the rollout stage, continuously supplying tokens preferred by the negative policy over the teacher so that they remain exposed to teacher supervision throughout training. This provides an explicit negative signal through negative-policy rollouts while preserving the positive teacher supervision used in OPD. Through extensive experiments, we show that NP-OPD improves OPD across model scales, generation modes, reasoning domains, and different OPD variants. Furthermore, our analyses show that NP-OPD effectively suppresses tokens preferred by the negative policy over the teacher and moves the student away from the negative policy. These results support our design of introducing negative signals through negative-policy rollouts and provide new insight into the role of the rollout policy in OPD.

1 INTRODUCTION

As large language models (LLMs) have rapidly improved, they have been widely utilized for diverse applications, such as daily chat, information search, and solving complex problems. While pre-training through next-token prediction remains important and allows LLMs to acquire broad knowledge, the increasingly diverse usage of LLMs has led to the development of various post-training methods to tune models toward specific goals. These methods further adapt models using additional supervision beyond pre-training. Among representative approaches, supervised fine-tuning (SFT) trains models on curated prompt-response pairs, which are often generated by stronger LLMs (Mukherjee et al., 2023; Muennighoff et al., 2025; Guha et al., 2026). Direct preference optimization (DPO) commonly uses pairs of responses with relative preferences, learning from both responses to favor one and suppress the other (Rafailov et al., 2023). On the other hand, reinforcement learning (RL) methods, such as GRPO, optimize the model's own generations according to reward signals (Ouyang et al., 2022; Shao et al., 2024). More recently, on-policy distillation (OPD) has been actively studied as an alternative to RL-based post-training with the token-level training signal obtained through distillation from a teacher model (Agarwal et al., 2024; Lu & Lab, 2025; Yang et al., 2025; Xiao et al., 2026).

Knowledge distillation (Hinton et al., 2015) has been applied to LLM post-training in various ways, particularly by varying how training sequences are obtained. SeqKD (Kim & Rush, 2016) utilizes teacher-generated sequences for distillation, commonly referred to as supervised fine-tuning (SFT) in LLM post-training. Although SeqKD plays an important role in transferring teacher knowledge, it has limitations in preserving the student's on-policy capabilities. To address this issue, OPD performs distillation on student-generated sequences, similar to RL, to preserve the student's capabilities during distillation. Recent studies have mainly focused on improving the token-level distillation signal through alternative loss functions or teacher configurations (Yang et al., 2026; Heo et al., 2026;

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



(a) Preference optimization (DPO) (b) On-policy distillation (OPD) (c) Negative-Policy OPD (NP-OPD)

Figure 1: **Conceptual comparison of preference optimization, on-policy distillation (OPD), and Negative-Policy OPD (NP-OPD).** (a) Preference optimization provides both toward and away directions through preferred and non-preferred responses, whereas (b) OPD provides an explicit positive direction through teacher supervision but no explicit negative reference indicating what the student should move away from. (c) NP-OPD complements the positive signal from teacher supervision with rollouts from a lower-performing, lower-capability negative policy that serves as a negative reference. Here, "negative" refers to the role of the negative policy as a reference from which the student is intended to move away. In our main setting, the negative policy is selected from the same model family as the student with lower model capacity and overall reasoning performance.

Feng et al., 2026; Wu et al., 2026b). Despite these advances, the objective of distillation has remained unchanged: *mimicking the teacher model*.

The teacher model serves as the only positive role model in distillation. When the student's distribution sufficiently overlaps with that of the teacher, the positive-only guidance provides enough signal for on-policy distillation. However, distillation often employs a substantially stronger teacher whose distribution can differ from that of the student. As a result, limited overlap between the teacher and student distributions provides insufficient supervision for OPD, as observed in previous studies (Fu et al., 2026; Li et al., 2026). In DPO (Rafailov et al., 2023), negative samples complement positive-sample supervision and are frequently sampled from a weaker model (Geng et al., 2025; Team Olmo et al., 2025). Motivated by DPO, we introduce a negative policy as a complementary learning signal for OPD. The negative policy provides an additional source of supervision for undesirable outputs, potentially complementing the limited teacher supervision in low-overlap regions.

We propose Negative-Policy OPD (NP-OPD) to introduce a negative policy into an on-policy distillation framework (see Fig. 1). Existing OPD variants often modify the reward design to achieve their respective objectives. Thus, introducing the negative policy through the reward limits its compatibility with recent advances in OPD (Yang et al., 2026; Heo et al., 2026). Instead, we introduce the negative policy to the rollout stage. In NP-OPD, some tokens are sampled from the negative policy and repeatedly optimized using the distillation reward. Consequently, suppression is more concentrated on negative tokens, defined as tokens preferred by the negative policy over the teacher, than under on-policy rollouts. In this way, NP-OPD enhances the suppression of these tokens while preserving the original reward direction of OPD. Therefore, NP-OPD can be combined with recent advances in OPD to further improve their performance.

In our main setting, we choose the negative policy from the same model family as the student, with lower model capacity and lower overall reasoning performance. To validate the effectiveness of NP-OPD, we conduct extensive experiments using different student scales and both thinking and non-thinking modes, and further evaluate it on 13 benchmarks from three domains, math, science, and code. NP-OPD consistently improves OPD and remains effective when combined with recent OPD variants such as ExOPD (Yang et al., 2026) and OPD^2 (Heo et al., 2026). This demonstrates that the benefit of NP-OPD is not tied to a specific OPD formulation and can complement existing OPD methods without modifying their reward formulations. Beyond empirical validation, our analyses show that NP-OPD operates as intended by concentrating suppression on negative tokens and reducing the measured overlap with the negative policy. These effects are associated with larger performance gains, explaining why NP-OPD improves OPD. As an additional practical benefit, NP-OPD can reuse pre-generated negative-policy rollouts, reducing the cost of rollout generation during training. Furthermore, we discuss a new perspective on the success of recent OPD variants. We also show that degrading student rollouts improves OPD but is less effective than using a separate negative policy.

2 PRELIMINARY

2.1 ON-POLICY DISTILLATION

On-policy distillation (OPD) trains a student model π_θ using trajectories generated by the student itself and token-level supervision from a stronger teacher π^* (Agarwal et al., 2024; Lu & Lab, 2025; Yang et al., 2026; Heo et al., 2026). Given a question $x \sim \mathcal{D}$, the student generates a rollout

$$y \sim \pi_\theta(\cdot \mid x). \tag{1}$$

At each token position t, the teacher and student distributions are conditioned on the same prefix $(x, y_{<t})$. As an alternative to reinforcement learning (RL)-based post-training, OPD can be implemented in an RL-style framework using sampled-token distillation signals (Heo et al., 2026). For a sampled token y_t, the token-level signal is given by

$$R_t^{\mathrm{OPD}} = \log \pi^*(y_t \mid x, y_{<t}) - \log \pi_\theta(y_t \mid x, y_{<t}). \tag{2}$$

This token-level signal adjusts the student toward the teacher distribution. Although individual token probabilities can increase or decrease, the teacher remains the sole reference that determines the update direction. The resulting RL-style update direction is written as

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

Recent OPD variants modify this token-level reward in different ways (Yang et al., 2026; Heo et al., 2026). We provide their detailed formulations in Appendix B.

2.2 LEARNING FROM POSITIVE AND NEGATIVE SIGNALS

Several post-training methods train models using not only outputs that should be preferred (or correct), but also those that should be avoided (or incorrect) (Rafailov et al., 2023; Shao et al., 2024; Meng et al., 2024; Geng et al., 2025). For example, Direct Preference Optimization (DPO) (Rafailov et al., 2023) uses a pair of preferred and rejected responses (y^+, y^-) with the following objective:

$$\mathcal{L}_{\mathrm{DPO}} = -\mathbb{E}\left[\log \sigma \left(\beta \left[\log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\mathrm{ref}}(y^+ \mid x)} - \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\mathrm{ref}}(y^- \mid x)} \right] \right) \right]. \tag{4}$$

The preferred response provides a direction to move toward, while the rejected response explicitly identifies a direction to move away from.

A similar structure appears in Reinforcement Learning (RL). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) uses group-relative advantages, with the policy-gradient component

$$\nabla_\theta J(\theta) \propto \mathbb{E}\left[\sum_i A_i \nabla_\theta \log \pi_\theta(y_i \mid x) \right], \tag{5}$$

where responses with positive advantage are reinforced, while those with negative advantage are suppressed. In correctness-based reasoning tasks, this typically corresponds to reinforcing correct responses and suppressing incorrect ones.

OPD follows a different structure. As illustrated in Fig. 1, the teacher provides a single direction that the student should move toward, but does not explicitly indicate what the student should move away from. This contrast raises a natural question: *can such a negative signal benefit OPD without changing its original teacher supervision?*

3 METHOD

This section introduces Negative-Policy OPD (NP-OPD), which injects negative signals into OPD via the rollout policy and addresses the question raised in Sec. 2. NP-OPD augments student-generated trajectories with trajectories from an additional rollout source and applies teacher supervision to both. We present the formulation of NP-OPD and explain how changing the rollout policy provides a direction for the student to move away from.

3.1 DISTILLATION WITH NEGATIVE-POLICY ROLLOUTS

As motivated in Sec. 2, we introduce **N**egative-**P**olicy **OPD** (**NP-OPD**), which incorporates negative signal into OPD through the rollout distribution. Specifically, we modify the rollout distribution to include both student-generated trajectories and trajectories sampled from an additional negative policy, π_n. The *negative policy* refers to a policy that the student needs to move away from during training, similar to non-preferred responses in DPO. In our main setting, we choose π_n from the same model family as the student, with lower model capacity and lower overall reasoning performance. Each training batch consists of trajectories sampled from the student policy and the negative policy, with proportions $1-\alpha$ and α, respectively:

$$\pi_{z_x} = \begin{cases} \pi_\theta, & z_x = 0, \\ \pi_n, & z_x = 1, \end{cases} \tag{6}$$

where $\alpha \in [0, 1]$ controls the proportion of negative-policy trajectories.

The token-level distillation signal defined in Sec. 2.1 is applied to trajectories sampled from both rollout policies. The resulting RL-style update direction can be written as

$$\nabla_\theta J_{\mathrm{NP}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, z_x\sim\mathrm{Bernoulli}(\alpha),\, y\sim\pi_{z_x}(\cdot|x)} \left[\sum_{t=1}^{|y|} R_t^{\mathrm{OPD}} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}) \right]. \tag{7}$$

In each term, the sampled token y_t and its prefix $y_{<t}$ are drawn from the corresponding rollout policy. Thus, NP-OPD changes the trajectory distribution exposed to teacher supervision, while the token-level teacher-student supervision remains unchanged. When $\alpha = 0$, NP-OPD reduces to OPD with student-generated trajectories, while $\alpha = 1$ uses only negative-policy trajectories.

3.2 HOW NEGATIVE-POLICY ROLLOUTS PROVIDE NEGATIVE SIGNALS

NP-OPD introduces the negative policy into the rollout model. In addition to the on-policy model, the negative model is selected as the rollout model with probability α. This rollout implementation is designed to improve applicability to reward variants R_t^{OPD} for various recent advances in OPD studies (Yang et al., 2026; Heo et al., 2026). However, it is not straightforward how the negative policy on the rollouts makes the student model move away from the negative policy. In this section, we explain this mathematically.

We expand the expectation in Eq. 7 and consider the expected gradient contribution of the t-th token in a trajectory $y = (y_1, \ldots, y_T)$. The original OPD and NP-OPD have different sampling probabilities for y. Thus, the gradients of its t-th token with respect to the expected OPD are given by

$$g_{\mathrm{OPD}}(y, t) = \underbrace{\prod_{i=1}^{t} \pi_\theta(y_i \mid x, y_{<i})}_{\text{sampling probability}} R_t^{\mathrm{OPD}} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}) \tag{8}$$

for the original OPD. On the other hand, NP-OPD has a mixed sampling probability between the on-policy and negative policy models, which is denoted as

$$g_{\mathrm{NP}}(y, t) = \underbrace{\left[(1-\alpha) \prod_{i=1}^{t} \pi_\theta(y_i \mid x, y_{<i}) + \alpha \prod_{i=1}^{t} \pi_n(y_i \mid x, y_{<i}) \right]}_{\text{sampling probability}} R_t^{\mathrm{OPD}} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}). \tag{9}$$

The difference between these two gradient contributions quantifies the gradient changes induced by sampling from the negative policy:

$$g_{\mathrm{NP}} - g_{\mathrm{OPD}} = \alpha \left[\prod_{i=1}^{t} \pi_n(y_i \mid x, y_{<i}) - \prod_{i=1}^{t} \pi_\theta(y_i \mid x, y_{<i}) \right] R_t^{\mathrm{OPD}} \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}). \tag{10}$$

Compared to OPD, NP-OPD enhances the learning signal for negative policy trajectories, and the degree of enhancement is proportional to the probability α. The enhanced sampling of NP-OPD keeps the on-policy model away from the negative policy. This is valid under the following assumption:

$$D_{\mathrm{KL}}(\pi_n \| \pi^*) > D_{\mathrm{KL}}(\pi_n \| \pi_\theta). \tag{11}$$

The Kullback-Leibler (KL) divergence between the teacher π^* and the negative policy π_n is greater than that between the student π_θ and the negative policy. Since we select a model with weaker performance than the student as a negative policy, this is a reasonable assumption in practice. Under this assumption, the expected value of OPD reward in Eq. 2 is presented as

$$\begin{aligned}\mathbb{E}_{y_t\sim\pi_n}\left[R_t^{\mathrm{OPD}}\right] &= \mathbb{E}_{y_t\sim\pi_n}\left[\log\pi^*(y_t) - \log\pi_\theta(y_t)\right] \\ &= \mathbb{E}_{y_t\sim\pi_n}\left[\big(\log\pi^*(y_t) - \log\pi_n(y_t)\big) - \big(\log\pi_\theta(y_t) - \log\pi_n(y_t)\big)\right] \\ &= -D_{\mathrm{KL}}(\pi_n\|\pi^*) + D_{\mathrm{KL}}(\pi_n\|\pi_\theta) < 0.\end{aligned} \tag{12}$$

In short, the tokens generated by the negative policy have negative expected OPD rewards. This suggests that the OPD reward trains the student not to generate tokens from the negative policy. Combined with Eq. (10), NP-OPD increases the contribution of trajectories from the negative policy and then suppresses these tokens from being generated by the student on average. Thus, NP-OPD pushes the student away from the negative policy solely through rollout modification.

There are two notable aspects in interpreting NP-OPD under this formulation. First, NP-OPD selectively suppresses tokens generated by the negative policy according to the teacher's preference. While the formulation shows that NP-OPD statistically suppresses the negative policy's rollouts, this doesn't mean that every token sampled from the negative policy is suppressed. At the token level, tokens preferred by the teacher are still enhanced even when they have high probability under the negative policy. Since R^{OPD} in Eq. 7 remains the OPD reward, NP-OPD preserves the direction of OPD signal rather than explicitly reversing or overriding it. Although NP-OPD statistically trains the student away from the negative policy, it preserves the original training dynamics of OPD. Second, this formulation naturally extends to advanced distillation rewards (Yang et al., 2026; Heo et al., 2026). Although these rewards are not directly formulated by KL divergence, tokens sampled from the negative policy are generally expected to receive lower rewards than those from the student. Therefore, the negative expected reward in Eq. 12 is expected to hold for these reward variants, allowing NP-OPD to be applied to advanced OPD methods without modifying their reward functions.

4 Experiment and Analysis

We evaluate NP-OPD across various student scales, generation modes, and OPD reward formulations, and further validate its effectiveness on diverse reasoning benchmarks, examining whether negative-policy rollouts consistently improve OPD. We also examine the practical training efficiency of reusing pre-generated negative-policy rollouts. Finally, to validate that NP-OPD follows the mechanism described in Sec. 3.2, we analyze whether NP-OPD selectively suppresses tokens favored by the negative policy over the teacher and reduces the student's overlap with the negative policy.

4.1 Experimental Setup

Models. We use Qwen3-1.7B, Qwen3-4B, and Qwen3-8B (Yang et al., 2025) as student models, and conduct experiments with both thinking and non-thinking modes. Teachers are selected according to the scale of the student model and generation mode. For thinking mode, we use Qwen3-8B as the teacher for Qwen3-1.7B, and Qwen3-30B-A3B-Thinking for Qwen3-4B and Qwen3-8B. For non-thinking mode, we use Qwen3-4B-Instruct for Qwen3-1.7B and Qwen3-30B-A3B-Instruct for Qwen3-4B and Qwen3-8B. We additionally evaluate NP-OPD on the Gemma-4 family (Gemma Team et al., 2026), using Gemma-4-E4B-it as the student and Gemma-4-12B-it as the teacher, to examine whether the observed gains extend beyond the Qwen3 family.

Training and evaluation. The training set contains Math (Moshkov et al., 2025), Science (NVIDIA Corporation, 2025), and Code (Ahmad et al., 2025) questions mixed at a 1:1:1 ratio, and a total of 30k questions are used. In all experiments, we use a fixed batch size of 256, a learning rate of 5×10^{-6}, a maximum rollout length of 16k tokens, and 100 training steps unless otherwise specified. We vary the rollout mixing ratio as $\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ for OPD. Note that $\alpha = 0.0$ uses only on-policy rollouts (OPD), while $\alpha = 1.0$ uses only negative-policy rollouts. All rollouts are sampled with a temperature of 1.0. For evaluation, we use a temperature of 0.7 and a maximum generation length of 32k tokens. We evaluate reasoning performance on seven math benchmarks, three code benchmarks, and three science benchmarks. Further details are provided in Appendix C.1.

Table 1: **Results of Qwen3 models on math, code, and science benchmarks (non-thinking mode).** The best and second-best results are shown in bold and underlined, respectively.

	Math							
Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-1.7B	11.5	10.0	45.9	5.4	69.0	25.9	82.5	35.7
+ OPD	31.7	21.7	65.3	13.3	80.4	36.0	89.9	48.3
+ NP-OPD	**43.1**	**33.3**	**78.4**	**20.0**	**86.0**	**40.6**	**92.3**	**56.3**
Qwen3-4B	24.2	20.0	69.4	11.7	79.0	34.7	88.9	46.8
+ OPD	53.8	42.3	90.0	26.2	88.2	43.7	90.9	62.2
+ NP-OPD	**67.9**	**58.8**	**95.3**	**43.8**	**90.6**	**45.5**	**91.7**	**70.5**

	Code				Science			
Model	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-1.7B	9.2	8.3	19.5	12.3	40.9	23.8	27.6	30.8
+ OPD	**19.1**	29.9	32.7	27.2	48.9	26.2	36.5	37.2
+ NP-OPD	18.5	**33.6**	**41.3**	**31.1**	**54.9**	**28.5**	**41.2**	**41.5**
Qwen3-4B	16.1	26.6	32.0	24.9	43.1	31.5	46.6	40.4
+ OPD	30.5	42.0	48.2	40.2	56.3	36.5	50.5	47.8
+ NP-OPD	**36.4**	**56.1**	**60.7**	**51.1**	**60.5**	**38.4**	**54.8**	**51.2**

Rollout policy. For our main experiments, we use Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B as negative-policy models for Qwen3-1.7B, Qwen3-4B, and Qwen3-8B students, respectively. For Gemma-4-E4B-it, we utilize Gemma-4-E2B-it as the negative-policy model. We also vary the scale of the rollout model to examine the effect of rollout policy capability.

4.2 Main Results

In non-thinking mode, NP-OPD consistently improves over OPD across model scales and reasoning domains. For Qwen3-1.7B with $\alpha = 1.0$, NP-OPD increases the average performance from 48.3 to 56.3 on Math, from 27.2 to 31.1 on Code, and from 37.2 to 41.5 on Science. Similar gains are observed for Qwen3-4B, where the averages improve from 62.2 to 70.5 on Math, from 40.2 to 51.1 on Code, and from 47.8 to 51.2 on Science. Across individual benchmarks, NP-OPD improves performance in most settings, with particularly large gains on AIME, HMMT25, LCBv5, and RGAlgo. The same trend extends to Qwen3-8B and to thinking-mode generation, with thinking-mode results summarized in Tab. 2 and full benchmark-wise results provided in Appendix C.2.

NP-OPD with various OPD reward formulations. We also test NP-OPD with recent OPD variants that modify the token-level reward, including ExOPD (Yang et al., 2026) and OPD^2 (Heo et al., 2026). As shown in Tab. 2, NP-OPD improves overall performance across all three formulations for Qwen3-1.7B and Qwen3-4B in thinking mode. For Qwen3-1.7B, NP-OPD improves the overall average by 1.98 points with OPD and 1.76 points with OPD^2, while maintaining comparable performance with ExOPD. For Qwen3-4B, NP-OPD improves all three formulations, with gains of 1.11, 1.86, and 2.10 points for OPD, ExOPD, and OPD^2, respectively. This suggests that the benefit of NP-OPD is not limited to a specific student scale or OPD reward formulation. We report representative results in Tab. 2, with full results provided in Appendix C.2.

NP-OPD on Gemma. We additionally evaluate NP-OPD on the Gemma family to examine whether its effectiveness extends beyond Qwen3. As shown in Tab. 3, NP-OPD with $\alpha = 1.0$ improves the average performance of Gemma-4-E4B over OPD from 63.0 to 66.4 on Math and from 49.1 to 49.8 on Science. On Code, although both methods remain below the base model's performance of 56.5, NP-OPD still outperforms the on-policy OPD baseline, improving the average from 54.9 to 55.2. Overall, NP-OPD is also effective on Gemma, with clear gains on Math and Science.

Training efficiency. Training efficiency is a practical advantage of NP-OPD. Negative-policy rollouts can be pre-generated and reused throughout training, unlike student-generated on-policy rollouts.[1]

[1]We plan to release the pre-generated negative-policy rollouts with our code.

Table 2: **Effect of NP-OPD across OPD variants (thinking mode).** Results where NP-OPD improves over the on-policy baseline are shown in bold. "All" denotes the average across all 13 benchmarks.

	Math			Code			Science			All		
	-	NP-OPD	Δ	-	NP-OPD	Δ	-	NP-OPD	Δ	-	NP-OPD	Δ
Qwen3-1.7B												
OPD	60.0	**61.8**	**+1.83**	34.3	**37.6**	**+3.24**	40.9	**41.9**	**+1.06**	49.7	**51.6**	**+1.98**
ExOPD	63.1	**63.4**	**+0.27**	38.2	**39.0**	**+0.77**	43.4	43.2	-0.12	52.8	**53.1**	**+0.30**
OPD^2	64.3	**65.3**	**+1.00**	38.5	**43.0**	**+4.48**	44.2	**45.0**	**+0.83**	53.7	**55.5**	**+1.76**
Qwen3-4B												
OPD	70.7	**72.2**	**+1.51**	52.9	**53.4**	**+0.49**	49.8	**50.6**	**+0.80**	61.7	**62.9**	**+1.11**
ExOPD	72.8	**76.0**	**+3.20**	58.7	57.6	-1.10	51.5	**53.2**	**+1.70**	64.6	**66.5**	**+1.86**
OPD^2	74.0	**76.8**	**+2.85**	58.4	**60.6**	**+2.17**	52.3	**52.6**	**+0.28**	65.4	**67.5**	**+2.10**

Table 3: **Results of Gemma-4-E4B models on math, code, and science benchmarks (thinking mode).** The best and second-best results are shown in bold and underlined, respectively.

	Math							
Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Gemma-4-E4B	55.0	35.0	88.8	28.7	88.0	42.6	89.7	61.1
+ OPD	57.9	42.1	81.2	35.8	87.4	42.1	**94.1**	63.0
+ NP-OPD	**61.9**	**45.6**	**91.6**	**37.5**	**89.8**	**44.7**	93.9	**66.4**

	Code				Science			
Model	Code Forces	LCBv5	RG Algo	Avg	GPQA	Super GPQA	Sci Bench	Avg
Gemma-4-E4B	**44.1**	**61.2**	64.1	**56.5**	59.5	36.8	47.8	48.0
+ OPD	41.9	59.1	63.7	54.9	60.4	37.6	49.4	49.1
+ NP-OPD	40.8	58.4	**66.4**	55.2	**62.0**	**37.9**	**49.6**	**49.8**

For Qwen3-1.7B on a single 8×H100 node, NP-OPD with $\alpha = 1$ reduces standard OPD training time from 9h19m to 3h34m in thinking mode, excluding rollout pre-generation. Additional details for different values of α are provided in Appendix C.3.

4.3 Understanding the Effectiveness of NP-OPD

In the previous section, we empirically showed that NP-OPD consistently improves OPD across different student models, generation modes, and OPD variants. We now examine whether these gains are consistent with our hypothesis that supplying negative-policy rollouts provides a negative signal during distillation. Then, we consider two complementary questions. First, does training suppress the intended tokens, i.e., those preferred by the negative policy over the teacher? Second, does this targeted suppression actually reduce the student's overlap with the negative policy? For both analyses, we compare rollout policies with different capabilities.

Suppressing the intended tokens. We first test whether training selectively suppresses tokens associated with the negative policy. We define a metric, *negative-policy suppression precision* (NSP), as the fraction of suppressed tokens that are preferred by the negative policy over the teacher. We evaluate each trained student on a fixed set of trajectories generated by the initial student. For this analysis, we use Qwen3-1.7B as the student and Qwen3-8B as the teacher in thinking mode, with Qwen3-0.6B as the default negative policy. We define NSP as

$$\mathrm{NSP} = P(\log \pi_n(y_t \mid s_t) > \log \pi^*(y_t \mid s_t) \,|\, t \in \mathcal{D}_q)\,, \tag{13}$$

where $\mathcal{D}_q$ contains positions with decreased log-probability among the top $q\%$ by absolute change from the initial student. A higher NSP indicates that the largest probability reductions are more selectively concentrated on tokens preferred by the negative policy over the teacher.

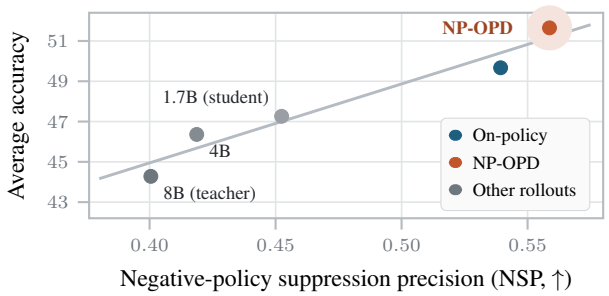


Figure 2: **Negative-policy suppression precision (NSP) for OPD and different rollout policies.** NP-OPD with a lower-capability rollout policy than the student yields higher NSP and higher reasoning performance, while using higher-capability rollouts decreases accuracy.

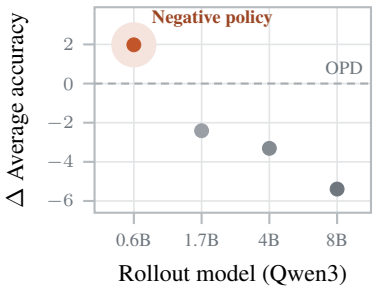


Figure 3: **Role of lower-capability rollouts.** When changing the rollout policy, only Qwen3-0.6B, a negative policy under our definition, improves over OPD.

As shown in Fig. 2, NP-OPD using Qwen3-0.6B as the negative policy achieves higher NSP than OPD using only on-policy rollouts. This indicates that suppression during distillation is more concentrated on tokens preferred by the negative policy over the teacher, supporting the role of these rollouts as a negative reference and remaining consistent with the intended mechanism described in Sec. 3.2.

Furthermore, NSP is highly correlated with average accuracy across OPD, NP-OPD, and alternative rollout settings using Qwen3-1.7B, Qwen3-4B, and Qwen3-8B. Using larger rollout models with higher capability and performance than the student, including the teacher, yields lower NSP, indicating that suppression is less concentrated on negative tokens. As shown in Fig. 3, only the lower-capability negative policy improves over on-policy OPD, consistent with our intended use of its rollouts as a negative reference. This also suggests that changing the rollout source alone does not explain the gains of NP-OPD.

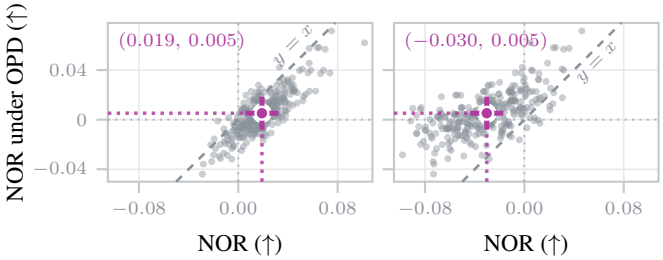
Reducing overlap with the negative policy. While NSP measures where suppression is concentrated, we next examine whether NP-OPD reduces the student's overlap with the negative policy. We define *negative-policy overlap reduction* (NOR) as

$$\mathrm{NOR}(i;\pi_A) = \frac{1}{|\mathcal{X}^{(i)}|} \sum_{t \in \mathcal{X}^{(i)}} \left[\Delta\pi(v_T^* \mid s_t) - \Delta\pi(v_n^* \mid s_t)\right], \tag{14}$$

where i indexes trajectories, $s_t = (x, y_{<t})$, and $\Delta\pi = \pi_A - \pi_{S_0}$ denotes the probability change after training. The tokens v_n^* and v_T^* are the top predictions of the negative policy and teacher at s_t, respectively, and $\mathcal{X}^{(i)}$ contains positions where they differ. Here, π_{S_0} and π_A denote the student before and after training, respectively, and overlap reduction refers to a teacher-relative shift in top-token probabilities, rather than an absolute reduction in distributional overlap. A larger NOR indicates a greater shift in the student's probability gap toward the teacher's top prediction relative to the negative policy's at disagreement positions.

For the main analysis, we use a fixed set of trajectories generated by the initial student from prompts used during training. Fig. 4 (a) compares NOR under OPD and NP-OPD for these trajectories. Each point represents one trajectory, with the horizontal and vertical axes corresponding to NP-OPD and OPD, respectively. Most points lie below the diagonal, with a mean NOR of 0.019 under NP-OPD compared with 0.005 under OPD, indicating a greater relative shift toward the teacher. In contrast, Fig. 4 (b) shows that using teacher rollouts during training yields a mean NOR of -0.030, below that of OPD. Additional results on held-out prompts and other rollout models show similar trends and are provided in Appendix C.4. These results support our design of using negative-policy rollouts to induce the intended shift away from the negative policy.

These analyses address the two questions of whether NP-OPD suppresses tokens preferred by the negative policy and whether it reduces the student's overlap with the negative policy. In both cases, NP-OPD operates as intended, consistent with the mechanism described in Sec. 3.2. These results show that NP-OPD is supported not only by empirical performance gains but also by direct analyses of the effects induced by negative-policy rollouts.



(a) Negative rollouts (b) Teacher rollouts

Figure 4: **Reduction in overlap with the negative policy.** Each point compares trajectory-level NOR under OPD (y-axis) and (a) negative-policy or (b) teacher rollouts (x-axis). Points below the diagonal indicate greater overlap reduction.

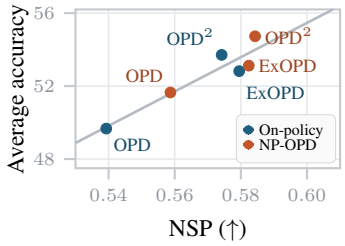


Figure 5: **NSP across OPD variants.** Each pair compares on-policy training with NP-OPD for three formulations; OPD, ExOPD, and OPD^2.

5 Discussion

A negative-signal perspective on recent OPD methods. As shown in Fig. 5, ExOPD and OPD^2 achieve higher NSP than standard OPD even with on-policy rollouts. This provides a different perspective on recent OPD variants that modify the distillation signal. ExOPD defines its reward relative to a reference model, the teacher's base model, while OPD^2 explicitly uses the difference between the teacher and its pretrained base model. This interpretation suggests that recent OPD methods may already benefit from such implicit negative signals even without explicitly targeting them. Even with these signals, NP-OPD further increases NSP when combined with each method, together with additional gains in average accuracy. Thus, the negative signal introduced through negative-policy rollouts can complement the implicit negative signals in these OPD methods.

Can negative rollouts be synthesized from the student? Our method uses rollouts from a separate, lower-capability negative policy as a negative reference for the student. Using Qwen3-1.7B as the student and Qwen3-8B as the teacher in thinking mode, we examine whether similar negative rollouts can be synthesized from the student by increasing the decoding temperature to 1.5, adding a persona prompt that asks it to solve the problem at an elementary-school level, or dropping three layers during rollout generation. As shown in Tab. 4, some of these modifications improve over standard on-policy rollouts, suggesting that degrading the rollout process can provide useful complementary signals. However, their gains remain substantially smaller than those obtained with a separate lower-capability negative policy. In particular, temperature scaling and persona prompting improve the average score by 0.9 and 1.0 points, respectively, whereas the negative policy improves the average by 3.4 points. These results suggest that simple degradation of the student can partially synthesize useful negative rollouts, but does not match the benefit of using a separate lower-capability policy to provide a move-away direction in OPD.

Table 4: **Performance with other degraded rollouts.**

Rollout policy	AIME24	AIME25	GPQA	LCB	AVG
On-policy	50.0	35.6	49.1	36.9	42.9
+ high temp.	52.7	37.1	48.3	37.1	43.8
+ persona prompt	52.1	36.5	51.2	35.9	43.9
+ layer drop	49.0	35.8	51.0	32.5	42.1
NP-OPD	**54.2**	**38.5**	**51.4**	**41.0**	**46.3**

6 Conclusion

In this work, we introduced Negative-Policy OPD (NP-OPD), which complemented positive teacher guidance with rollouts from a lower-capability negative policy serving as a negative reference for the student. NP-OPD introduced the negative policy at the rollout stage while preserving the original distillation reward formulation. Across model scales, generation modes, reasoning domains, and different OPD variants, NP-OPD improved performance over on-policy baselines. Our analyses showed that NP-OPD concentrated suppression on tokens preferred by the negative policy over the teacher and reduced the measured overlap with the negative policy. Reusing pre-generated negative-policy rollouts also reduced rollout generation costs during training. These findings provided a new perspective on the role of the rollout policy in OPD, showing how negative-policy rollouts complemented positive teacher guidance without modifying the distillation reward.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic training rollouts, which we plan to release publicly, and to assist with implementing methods. We also used generative AI tools to review mathematical claims, proofs, hypotheses, research methodology and experiments, and interpretations of results. For these review tasks, AI assistance was limited to reviewing author-developed content rather than generating the initial claims, proofs, hypotheses, experimental designs, or interpretations.

We did not use generative AI tools to develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, assist with translation, clean or reformat datasets, or support qualitative or thematic data analysis. Additionally, we used generative AI tools to revise author-written drafts.

We have reviewed all AI-assisted work, including the generated training rollouts, implementation assistance, manuscript revisions, and feedback on author-developed research content. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the formulation of NP-OPD in Sec. 3 and the model configurations, training data, and evaluation settings in Sec. 4.1. Additional training and evaluation details, hyperparameters, and benchmark-wise results are provided in Appendix C.1 and C.2. Sec. 4.3 defines the NSP and NOR metrics and describes the trajectories used for analysis. Appendix C.4 further details the held-out prompts used to evaluate overlap reduction beyond the training prompts. We plan to release our code and pre-generated negative-policy rollouts to facilitate reproduction of our experiments.

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*, volume 2024, 2024.

Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. Opencodereasoning: Advancing data distillation for competitive coding. *arXiv preprint arXiv:2504.01943*, 2025.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. 2026. URL https://arxiv.org/abs/2605.00674.

Shiyuan Feng, Huan-ang Gao, Haohan Chi, Hanlin Wu, Zhilong Zhang, Zheng Jiang, Bingxiang He, Wei-Ying Ma, Ya-Qin Zhang, and Hao Zhou. Weak-to-strong generalization via direct on-policy distillation. *arXiv preprint arXiv:2607.05394*, 2026.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Scott Geng, Hamish Ivison, Chun-Liang Li, Maarten Sap, Jerry Li, Ranjay Krishna, and Pang Wei Koh. The delta learning hypothesis: Preference tuning on weak data can yield strong gains. *arXiv preprint arXiv:2507.06187*, 2025.

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

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. In *International Conference on Learning Representations*, 2026.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024. URL https://arxiv.org/abs/2402.14008.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

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

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, 2025.

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

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

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. *arXiv preprint arXiv:2402.03898*, 2024.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

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.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixin Deng, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, Dehua Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tianshun Xing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, Yuelin Bai, Xingyuan Bu, Chenglin Cai, Liang Chen, Yifan Chen, Chengtuo Cheng, Tianhao Cheng, Keyi Ding, Siming Huang, Yun Huang, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Tianyang Pang, Zhongyuan Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Shanghaoran Quan, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jinyang Zhang, Xiyue Zhang, Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, Chenghua Zhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

MAA. American mathematics competition - amc. In *American Mathematics Competition - AMC*, 2023. URL https://maa.org/student-programs/amc/.

MAA. American invitational mathematics examination - aime 2024, February 2024. URL https://maa.org.

MAA. American invitational mathematics examination - aime 2025, February 2025. URL https://maa.org.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 2024.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023.

NVIDIA Corporation. Opensciencereasoning-2. https://huggingface.co/datasets/nvidia/OpenScienceReasoning-2, 2025. Hugging Face dataset.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 2022.

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. *Advances in neural information processing systems*, 2023.

Negin Raoof, Etash Kumar Guha, Ryan Marten, Jean Mercat, Eric Frankel, Sedrick Keh, Hritik Bansal, Georgios Smyrnis, Marianna Nezhurina, Trung Vu, Zayne Rea Sprague, Mike A Merrill, Liangyu Chen, Caroline Choi, Zaid Khan, Sachin Grover, Benjamin Feuer, Ashima Suvarna, Shiye Su, Wanjia Zhao, Kartik Sharma, Charlie Cheng-Jie Ji, Kushal Arora, Jeffrey Li, Aaron Gokaslan, Sarah M Pratt, Niklas Muennighoff, Jon Saad-Falcon, John Yang, Asad Aali, Shreyas Pimpalgaonkar, Alon Albalak, Achal Dave, Hadi Pouransari, Greg Durrett, Sewoong Oh, Tatsunori Hashimoto, Vaishaal Shankar, Yejin Choi, Mohit Bansal, Chinmay Hegde, Reinhard Heckel, Jenia Jitsev, Maheswaran Sathiamoorthy, Alex Dimakis, and Ludwig Schmidt. Evalchemy. 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

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

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards, 2025. URL https://arxiv.org/abs/2505.24760.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. SciBench: Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models. In *Proceedings of the Forty-First International Conference on Machine Learning*, 2024.

Yecheng Wu, Song Han, and Han Cai. Lightning opd: Efficient post-training for large reasoning models with offline on-policy distillation. *arXiv preprint arXiv:2604.13010*, 2026a.

Yecheng Wu, Song Han, and Han Cai. Lightning opd 2.0: Mitigating style bias in cross-teacher on-policy distillation for large reasoning models. *arXiv preprint arXiv:2607.28449*, 2026b.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

Haolei Xu, Xiaowen Xu, Haiwen Hong, Zixuan Ni, Hongxing Li, Yiwen Qiu, Weiming Lu, and Yongliang Shen. Pass the baton: Trajectory-relayed on-policy distillation. *arXiv preprint arXiv:2607.26057*, 2026.

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

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

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

APPENDIX

A RELATED WORK

A.1 POST-TRAINING FOR LLMS

Post-training adapts pretrained LLMs to instruction following and reasoning through additional supervision. Supervised fine-tuning (SFT) trains models on curated prompt-response pairs, often using responses generated by stronger models (Mukherjee et al., 2023; Muennighoff et al., 2025; Guha et al., 2026). Preference-based methods learn from comparisons between responses. RLHF uses a learned reward model to guide policy optimization (Ouyang et al., 2022), while DPO directly optimizes the relative likelihood of preferred and rejected responses with respect to a reference policy (Rafailov et al., 2023). For reasoning tasks, reinforcement learning with verifiable rewards uses task-specific feedback, such as answer correctness, to improve model-generated responses (Shao et al., 2024; Yu et al., 2025). These methods illustrate complementary roles for positive and negative signals, encouraging preferred behavior while discouraging less desirable outputs. Our work studies a related role for negative references in OPD, introducing them through the rollout policy while preserving the existing distillation reward.

A.2 ON-POLICY DISTILLATION

Knowledge distillation (KD) transfers the predictive behavior of a teacher model to a student (Hinton et al., 2015). For autoregressive generation, the source of training sequences distinguishes two common forms of distillation. Sequence-KD uses teacher-generated sequences (Kim & Rush, 2016), which is now commonly realized as SFT on model-generated responses (Muennighoff et al., 2025; Guha et al., 2026). OPD instead generates trajectories from the student and obtains token-level supervision from the teacher (Agarwal et al., 2024; Gu et al., 2023; Ko et al., 2024). This reduces the train-inference mismatch of teacher-generated supervision and has shown competitive performance for reasoning post-training (Lu & Lab, 2025; Yang et al., 2025; Xiao et al., 2026).

Recent studies extend OPD along several directions. A major line modifies the distillation signal, including uncertainty-aware weighting (Jin et al., 2026), extrapolated teacher signals (Yang et al., 2026), and the teacher-to-base delta used in OPD^2 (Heo et al., 2026). Other studies modify the trajectories used during distillation. Lightning OPD precomputes teacher supervision on fixed SFT rollouts to enable efficient offline training (Wu et al., 2026a), while Relay-OPD lets the teacher temporarily take over failed student prefixes to construct more reliable training trajectories (Xu et al., 2026). These approaches mainly introduce alternative trajectories to improve supervision quality or training efficiency. In contrast, we study a different role of the rollout policy: rollouts from the negative policy are intentionally supplied as a negative reference, similar in role to rejected responses in preference optimization. Our work revisits the rollout source as a means of supplying negative signals and analyzes how it shapes token-level updates under the existing distillation reward.

B OPD REWARD FORMULATIONS

In our experiments, we additionally use two variants of OPD to examine the effectiveness of NP-OPD across different reward formulations. Here, we describe the token-level reward formulations of ExOPD (Yang et al., 2026) and OPD^2 (Heo et al., 2026), following the notation introduced in Sec. 2.1. For compactness, we denote the token context by $s_t = (x, y_{<t})$. As described in Sec. 2.1, standard OPD uses

$$R_t^{\mathrm{OPD}} = \log \pi^*(y_t \mid s_t) - \log \pi_\theta(y_t \mid s_t). \tag{15}$$

ExOPD. ExOPD (Yang et al., 2026) introduces a reference policy π_{ref} and extrapolates the reward from the reference toward the teacher. Its token-level reward can be written as

$$\begin{aligned} R_t^{\mathrm{ExOPD}} = \lambda \left[\log \pi^*(y_t \mid s_t) - \log \pi_{\mathrm{ref}}(y_t \mid s_t)\right] \\ - \left[\log \pi_\theta(y_t \mid s_t) - \log \pi_{\mathrm{ref}}(y_t \mid s_t)\right], \end{aligned} \tag{16}$$

where λ controls the degree of reward extrapolation. Equivalently,

$$R_t^{\mathrm{ExOPD}} = R_t^{\mathrm{OPD}} + (\lambda - 1)\left[\log \pi^*(y_t \mid s_t) - \log \pi_{\mathrm{ref}}(y_t \mid s_t)\right]. \tag{17}$$

Thus, $\lambda = 1$ recovers standard OPD, while $\lambda > 1$ extrapolates the reward along the direction from the reference policy toward the teacher. Following the main setting of ExOPD, we use the teacher's base model as the reference policy π_{ref} in our experiments.

OPD^2. OPD^2 (Heo et al., 2026) instead uses the probability change from the teacher's base model to the post-trained teacher. Denoting the corresponding base model by π^*_{base}, the delta reward is

$$R_t^{\Delta} = \log \pi^*(y_t \mid s_t) - \log \pi^*_{\text{base}}(y_t \mid s_t). \tag{18}$$

OPD^2 centers both the original OPD reward and the delta reward under the current student distribution:

$$A_t^{\text{OPD}} = R_t^{\text{OPD}} - \mathbb{E}_{\tilde{y}_t \sim \pi_\theta(\cdot|s_t)} \left[R_t^{\text{OPD}}(\tilde{y}_t) \right], \tag{19}$$

$$A_t^{\Delta} = R_t^{\Delta} - \mathbb{E}_{\tilde{y}_t \sim \pi_\theta(\cdot|s_t)} \left[R_t^{\Delta}(\tilde{y}_t) \right]. \tag{20}$$

The final OPD^2 signal is

$$A_t^{\text{OPD}^2} = \begin{cases} A_t^{\Delta}, & A_t^{\Delta} A_t^{\text{OPD}} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

This conditioning applies the delta signal only when its update direction agrees with that of standard OPD.

C Details and Additional Results of Section 4

C.1 Training and Evaluation Details

Training data and optimization. We train on 30K reasoning prompts with the corresponding system prompts. Unless otherwise specified, all models are trained for 100 steps with a global batch size of 256. We use AdamW with a learning rate of 5×10^{-6} and cosine decay with a minimum learning-rate ratio of 0.1. We use a warmup ratio of 0.1, gradient clipping at 1.0, bf16 precision, and ZeRO-3. Rollouts are generated with a maximum length of 16,384 tokens and a sampling temperature of 1.0. For ExOPD and OPD^2, we use the base model corresponding to each teacher, following their respective formulations. We set the KL regularization coefficient β to 0 for Qwen models and 0.04 for Gemma models.

Additional filtering and regularization. In the non-thinking mode of Qwen, we observe that some rollouts terminate with very short answers. Since such responses provide little token-level supervision for distillation, we filter rollouts shorter than 20 tokens from training. For Gemma, we observe substantially less stable OPD training than for Qwen. We therefore apply KL regularization with $\beta = 0.04$ for all Gemma experiments.

Infrastructure. Qwen experiments are conducted on a single node with eight NVIDIA H100 GPUs, with vLLM used concurrently for rollout generation. For Qwen3-8B, we use tensor parallelism with a tensor-parallel size of 4. Gemma experiments are conducted on two nodes.

Evaluation. To evaluate reasoning performance, we utilize AIME24 (MAA, 2024), AIME25 (MAA, 2025), AMC23 (MAA, 2023), HMMT25 (Dekoninck et al., 2026), MATH500 (Hendrycks et al., 2021), OlympiadBench (He et al., 2024), and ReasoningGym Math (RGMath) (Stojanovski et al., 2025) for math. For coding, we use CodeForces (Guha et al., 2026), LiveCodeBench (Jain et al., 2025), and ReasoningGym Algorithm (RGAlgo) (Stojanovski et al., 2025). For science, we use GPQA (Rein et al., 2023), SuperGPQA (M-A-P Team et al., 2025), and SciBench (Wang et al., 2024). We report avg@16 for AIME24 and AIME25, avg@8 for AMC23, HMMT25, MATH500, and GPQA, and avg@3 for the others. We use Evalchemy (Raoof et al., 2025) and `lm-evaluation-harness` (Gao et al., 2024) for benchmark evaluation, depending on the benchmark implementation. We use the same evaluation implementation and decoding configuration across all compared methods.

C.2 Full Results of Section 4.2

For Tab. 1, Tab. 2, Tab. C.1, and Tab. C.8, we report a single representative α per setting, preferring the value that improves all three domains over the on-policy baseline and, among such values, the one with the highest overall average. Across the evaluated nonzero α values, NP-OPD generally improves

the overall average over the corresponding on-policy baseline in both thinking and non-thinking modes. Full results for all evaluated α values are provided below.

Full results of non-thinking mode with OPD variants. Tab. C.1 summarizes the non-thinking results, showing that NP-OPD improves domain-average performance in nearly all combinations of student scale, reasoning domain, and OPD reward formulation. Tabs. C.2 and C.3 provide benchmark-wise results for OPD, Tabs. C.4 and C.5 for ExOPD, and Tabs. C.6 and C.7 for OPD^2. Larger α generally yields stronger results, particularly for OPD and ExOPD, while mixed rollouts also achieve competitive or better performance in some OPD^2 settings. These results show that incorporating negative-policy rollouts benefits multiple OPD reward formulations, with gains observed at different values of α.

Table C.1: **Summary of Qwen3 results in non-thinking mode.** The table reports average performance across Math, Code, and Science benchmarks. The best and second-best results are shown in bold and underlined, respectively.

	Qwen3-1.7B			Qwen3-4B			Qwen3-8B		
Model	Math	Code	Sci.	Math	Code	Sci.	Math	Code	Sci.
Baseline	35.7	12.3	30.8	46.8	24.9	40.4	45.7	29.8	43.0
OPD	48.3	27.2	37.2	62.2	40.2	47.8	65.1	44.0	50.1
+ NP-OPD	**56.3**	31.1	41.5	<u>70.5</u>	51.1	<u>51.2</u>	69.4	44.3	52.6
ExOPD	53.1	30.4	38.2	65.8	44.5	48.6	68.5	45.6	51.6
+ NP-OPD	<u>55.5</u>	30.2	<u>41.5</u>	**71.3**	**51.2**	**51.5**	<u>71.7</u>	<u>47.2</u>	**54.2**
OPD^2	53.1	<u>31.7</u>	39.5	66.3	46.8	49.4	68.3	42.7	52.6
+ NP-OPD	54.7	**31.9**	**43.1**	69.9	<u>51.2</u>	51.1	**72.9**	**53.0**	<u>53.2</u>

Table C.2: **Full results of Qwen3 models on mathematical reasoning benchmarks with OPD (non-thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-1.7B	11.5	10.0	45.9	5.4	69.0	25.9	82.5	35.7
OPD	31.7	21.7	65.3	13.3	80.4	36.0	89.9	48.3
+ NP-OPD (α=0.25)	32.3	25.6	72.8	15.4	82.8	36.8	91.1	51.0
+ NP-OPD (α=0.5)	34.6	26.0	70.0	15.8	83.2	37.6	90.3	51.1
+ NP-OPD (α=0.75)	<u>38.5</u>	<u>26.7</u>	<u>74.1</u>	<u>17.9</u>	<u>83.4</u>	<u>38.6</u>	<u>92.0</u>	<u>53.0</u>
+ NP-OPD (α=1)	**43.1**	**33.3**	**78.4**	**20.0**	**86.0**	**40.6**	**92.3**	**56.3**
Qwen3-4B	24.2	20.0	69.4	11.7	79.0	34.7	88.9	46.8
OPD	53.8	42.3	90.0	26.2	88.2	43.7	90.9	62.2
+ NP-OPD (α=0.25)	59.2	45.8	90.6	28.3	89.8	44.2	91.5	64.2
+ NP-OPD (α=0.5)	59.8	45.8	<u>94.4</u>	30.8	88.6	43.9	<u>92.6</u>	65.1
+ NP-OPD (α=0.75)	<u>63.5</u>	<u>48.8</u>	94.1	<u>36.7</u>	<u>90.0</u>	<u>44.6</u>	**92.9**	<u>67.2</u>
+ NP-OPD (α=1)	**67.9**	**58.8**	**95.3**	**43.8**	**90.6**	**45.5**	91.7	**70.5**
Qwen3-8B	26.5	18.5	66.9	5.4	78.8	34.3	89.6	45.7
OPD	61.5	46.5	91.9	28.3	89.6	44.4	93.5	65.1
+ NP-OPD (α=0.25)	67.3	46.0	90.0	29.2	**91.0**	44.6	94.2	66.0
+ NP-OPD (α=0.5)	68.1	51.2	93.4	27.5	90.8	44.8	**95.5**	67.3
+ NP-OPD (α=0.75)	<u>71.7</u>	<u>53.3</u>	<u>93.4</u>	<u>37.5</u>	90.4	<u>45.2</u>	<u>94.4</u>	<u>69.4</u>
+ NP-OPD (α=1)	**73.8**	**61.7**	**94.4**	**37.9**	<u>90.8</u>	**45.7**	93.5	**71.1**

Table C.3: **Full results of Qwen3 models on code and science benchmarks with OPD (non-thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively. **Avg** is the per-domain average.

	Code				Science			
Model	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-1.7B	9.2	8.3	19.5	12.3	40.9	23.8	27.6	30.8
OPD	19.1	29.9	32.7	27.2	48.9	26.2	36.5	37.2
+ NP-OPD (α=0.25)	**19.4**	30.5	34.1	28.0	49.6	26.5	37.5	37.8
+ NP-OPD (α=0.5)	16.8	30.5	34.0	27.1	51.3	27.9	36.9	38.7
+ NP-OPD (α=0.75)	18.8	31.9	37.7	29.4	51.8	27.8	39.4	39.7
+ NP-OPD (α=1)	18.5	**33.6**	**41.3**	**31.1**	**54.9**	**28.5**	**41.2**	**41.5**
Qwen3-4B	16.1	26.6	32.0	24.9	43.1	31.5	46.6	40.4
OPD	30.5	42.0	48.2	40.2	56.3	36.5	50.5	47.8
+ NP-OPD (α=0.25)	28.3	45.3	49.2	40.9	57.3	36.4	53.7	49.1
+ NP-OPD (α=0.5)	28.7	45.0	51.5	41.7	57.6	36.3	54.5	49.4
+ NP-OPD (α=0.75)	28.3	49.9	54.1	44.1	57.7	37.4	53.6	49.6
+ NP-OPD (α=1)	**36.4**	**56.1**	**60.7**	**51.1**	**60.5**	**38.4**	**54.8**	**51.2**
Qwen3-8B	21.1	28.5	39.9	29.8	52.1	31.4	45.6	43.0
OPD	31.3	**49.9**	50.8	44.0	60.4	38.3	51.6	50.1
+ NP-OPD (α=0.25)	30.2	42.5	50.1	41.0	63.9	**38.7**	52.1	51.6
+ NP-OPD (α=0.5)	31.3	46.9	52.5	43.6	63.5	38.2	52.3	51.3
+ NP-OPD (α=0.75)	**31.6**	48.0	53.4	44.3	**65.8**	38.6	53.3	**52.6**
+ NP-OPD (α=1)	30.2	47.2	**57.7**	**45.0**	56.8	36.6	**53.3**	48.9

Table C.4: **Full results of Qwen3 models on mathematical reasoning benchmarks with ExOPD (non-thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-1.7B	11.5	10.0	45.9	5.4	69.0	25.9	82.5	35.7
ExOPD	41.2	26.7	75.0	18.3	82.0	37.7	90.8	53.1
+ NP-OPD (α=0.5)	36.9	27.5	73.8	17.5	82.8	37.8	90.1	52.3
+ NP-OPD (α=1)	**41.2**	**31.5**	**80.9**	**19.6**	**83.2**	**39.6**	**92.2**	**55.5**
Qwen3-4B	24.2	20.0	69.4	11.7	79.0	34.7	88.9	46.8
ExOPD	59.6	51.0	93.8	30.4	89.6	44.9	91.4	65.8
+ NP-OPD (α=0.5)	59.2	49.2	93.1	34.2	89.8	44.1	**92.6**	66.0
+ NP-OPD (α=1)	**69.6**	**62.7**	**96.2**	**40.8**	**91.4**	**45.8**	92.5	**71.3**
Qwen3-8B	26.5	18.5	66.9	5.4	78.8	34.3	89.6	45.7
ExOPD	70.2	54.4	**95.6**	30.4	90.8	44.8	**93.4**	68.5
+ NP-OPD (α=0.5)	68.3	50.6	93.4	34.2	**91.2**	45.4	93.3	68.1
+ NP-OPD (α=1)	**74.2**	**62.9**	95.3	**41.2**	90.2	**45.7**	92.2	**71.7**

Table C.5: **Full results of Qwen3 models on code and science benchmarks with ExOPD (non-thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively. **Avg** is the per-domain average.

Model	Code				Science			
	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-1.7B	9.2	8.3	19.5	12.3	40.9	23.8	27.6	30.8
ExOPD	**21.0**	33.3	36.9	**30.4**	47.9	**29.4**	37.5	38.2
+ NP-OPD (α=0.5)	15.2	**34.7**	37.0	29.0	52.5	27.9	38.0	39.4
+ NP-OPD (α=1)	13.5	32.5	**44.6**	30.2	**56.4**	29.0	**39.2**	**41.5**
Qwen3-4B	16.1	26.6	32.0	24.9	43.1	31.5	46.6	40.4
ExOPD	31.3	46.3	55.8	44.5	56.6	37.1	52.0	48.6
+ NP-OPD (α=0.5)	29.6	45.7	56.1	43.8	57.3	36.6	**57.2**	50.4
+ NP-OPD (α=1)	**41.9**	**51.5**	**60.1**	**51.2**	**60.0**	**38.7**	55.8	**51.5**
Qwen3-8B	21.1	28.5	39.9	29.8	52.1	31.4	45.6	43.0
ExOPD	31.6	46.3	58.8	45.6	63.0	39.3	52.6	51.6
+ NP-OPD (α=0.5)	**33.8**	**49.8**	52.4	45.4	63.1	37.7	52.5	51.1
+ NP-OPD (α=1)	32.5	47.4	**61.8**	**47.2**	**67.4**	**41.3**	**53.9**	**54.2**

Table C.6: **Full results of Qwen3 models on mathematical reasoning benchmarks with OPD^2 (non-thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-1.7B	11.5	10.0	45.9	5.4	69.0	25.9	82.5	35.7
OPD^2	39.2	28.1	76.9	13.8	84.2	38.7	90.9	53.1
+ NP-OPD (α=0.5)	**39.2**	30.0	**77.2**	**19.2**	85.2	**39.8**	**92.3**	**54.7**
+ NP-OPD (α=1)	38.1	**31.0**	76.6	18.8	**85.4**	39.5	91.1	54.4
Qwen3-4B	24.2	20.0	69.4	11.7	79.0	34.7	88.9	46.8
OPD^2	60.8	50.4	92.8	31.7	91.0	45.4	92.3	66.3
+ NP-OPD (α=0.5)	65.6	**59.4**	**96.9**	38.7	90.4	**45.6**	**94.0**	**70.1**
+ NP-OPD (α=1)	**67.7**	58.1	96.2	**39.2**	**91.4**	45.3	91.5	69.9
Qwen3-8B	26.5	18.5	66.9	5.4	78.8	34.3	89.6	45.7
OPD^2	67.5	54.2	95.0	32.1	91.8	46.1	91.4	68.3
+ NP-OPD (α=0.5)	73.8	60.0	95.6	41.2	90.6	**46.6**	94.7	71.8
+ NP-OPD (α=1)	**75.6**	**62.9**	**97.2**	**41.7**	**92.2**	45.7	**94.9**	**72.9**

Table C.7: **Full results of Qwen3 models on code and science benchmarks with OPD^2 (non-thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively. **Avg** is the per-domain average.

Model	Code				Science			
	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-1.7B	9.2	8.3	19.5	12.3	40.9	23.8	27.6	30.8
OPD^2	**21.3**	33.5	40.3	31.7	51.0	**28.8**	38.6	39.5
+ NP-OPD (α=0.5)	19.6	**36.0**	40.2	**31.9**	**60.3**	28.7	**40.4**	**43.1**
+ NP-OPD (α=1)	15.3	30.9	**40.6**	28.9	55.1	28.2	38.9	40.7
Qwen3-4B	16.1	26.6	32.0	24.9	43.1	31.5	46.6	40.4
OPD^2	30.9	46.6	62.7	46.8	57.5	36.6	53.9	49.4
+ NP-OPD (α=0.5)	**35.9**	44.6	63.1	47.9	58.0	38.2	**57.9**	**51.4**
+ NP-OPD (α=1)	35.1	**50.9**	**67.5**	**51.2**	**59.2**	**39.6**	54.5	51.1
Qwen3-8B	21.1	28.5	39.9	29.8	52.1	31.4	45.6	43.0
OPD^2	33.6	44.4	50.1	42.7	64.3	40.2	**53.2**	52.6
+ NP-OPD (α=0.5)	34.1	48.6	62.7	48.5	66.3	39.5	51.4	52.4
+ NP-OPD (α=1)	**37.3**	**57.2**	**64.4**	**53.0**	**67.5**	**41.1**	51.0	**53.2**

Full results of thinking mode with OPD variants. Tab. C.8 summarizes the thinking-mode results, showing broad improvements with NP-OPD across student scales, reasoning domains, and OPD reward formulations. Tabs. C.9 and C.10 provide benchmark-wise results for OPD, Tabs. C.13 and C.14 for OPD^2, and Tabs. C.11 and C.12 for ExOPD. Gains are observed with both mixed and fully negative-policy rollouts, indicating that the benefit is not limited to a single value of α. The effect of increasing α varies across settings, with larger values often benefiting Math and Science and intermediate values sometimes yielding stronger Code results.

Table C.8: **Summary of Qwen3 results in thinking mode.** The table reports average performance across Math, Code, and Science benchmarks. The best and second-best results are shown in bold and underlined, respectively.

	Qwen3-1.7B			Qwen3-4B			Qwen3-8B		
Model	Math	Code	Sci.	Math	Code	Sci.	Math	Code	Sci.
Baseline	60.1	33.5	39.5	72.9	53.2	51.8	73.3	52.7	53.6
OPD	60.0	34.3	40.9	70.7	52.9	49.8	71.7	56.4	52.2
+ NP-OPD	61.8	37.6	41.9	72.2	53.4	50.6	74.3	54.0	53.1
ExOPD	63.1	38.2	43.4	72.8	58.7	51.5	74.2	60.3	53.6
+ NP-OPD	63.4	39.0	43.2	76.0	57.6	**53.2**	76.1	59.6	**55.9**
OPD^2	64.3	38.5	44.2	74.0	58.4	52.3	75.0	61.4	52.5
+ NP-OPD	**65.3**	**43.0**	**45.0**	**76.8**	**60.6**	52.6	**77.1**	**61.7**	55.3

Table C.9: **Full results of Qwen3 models on mathematical reasoning benchmarks with OPD (thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-1.7B	53.1	35.0	85.0	22.9	87.2	39.7	97.8	60.1
OPD	50.0	35.6	84.4	25.4	87.4	39.8	97.5	60.0
+ NP-OPD (α=0.25)	51.0	36.5	84.1	25.4	86.4	40.5	**98.1**	60.3
+ NP-OPD (α=0.5)	51.5	**39.2**	82.8	25.4	86.6	40.8	96.1	60.3
+ NP-OPD (α=0.75)	52.9	36.5	84.4	26.7	87.0	40.7	97.0	60.7
+ NP-OPD (α=1)	**54.2**	38.5	**86.6**	**27.1**	**87.4**	**41.3**	97.9	**61.8**
Qwen3-4B	72.9	61.9	95.9	44.2	91.2	45.1	99.1	72.9
OPD	67.9	54.4	97.5	39.6	91.0	46.1	98.2	70.7
+ NP-OPD (α=0.25)	69.6	60.8	96.9	39.6	90.4	45.9	98.5	71.7
+ NP-OPD (α=0.5)	70.4	57.5	97.5	43.3	91.2	46.3	**99.0**	72.2
+ NP-OPD (α=0.75)	74.2	59.0	**98.1**	45.0	91.2	46.8	97.7	73.1
+ NP-OPD (α=1)	**75.0**	**63.3**	95.9	**47.1**	**91.4**	**46.9**	98.7	**74.1**
Qwen3-8B	75.6	65.6	94.1	46.7	91.8	39.7	99.4	73.3
OPD	72.3	57.1	97.2	37.5	**91.8**	46.2	100.0	71.7
+ NP-OPD (α=0.25)	72.9	61.0	97.2	42.9	91.2	47.2	99.1	73.1
+ NP-OPD (α=0.5)	73.1	60.6	**97.8**	42.1	91.4	46.8	99.7	73.1
+ NP-OPD (α=0.75)	74.2	**62.3**	97.2	**48.8**	90.6	47.6	99.8	**74.3**
+ NP-OPD (α=1)	**74.4**	61.7	95.3	44.2	91.6	**48.0**	**100.0**	73.6

Table C.10: **Full results of Qwen3 models on code and science benchmarks with OPD (thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

	Code				Science			
Model	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-1.7B	22.5	33.5	44.6	33.5	47.5	28.7	42.4	39.5
OPD	22.5	36.9	43.6	34.3	49.1	**29.8**	43.8	40.9
+ NP-OPD (α=0.25)	25.3	38.5	42.9	35.6	49.2	27.6	43.4	40.1
+ NP-OPD (α=0.5)	**25.6**	39.2	45.8	36.9	49.2	28.7	43.7	40.5
+ NP-OPD (α=0.75)	24.7	37.3	46.2	36.1	50.7	28.8	43.9	41.1
+ NP-OPD (α=1)	25.1	**41.0**	**46.6**	**37.6**	**51.4**	29.6	**44.8**	**41.9**
Qwen3-4B	39.7	58.9	60.8	53.2	57.0	38.5	60.0	51.8
OPD	38.9	**59.3**	60.5	52.9	52.8	36.9	59.6	49.8
+ NP-OPD (α=0.25)	39.0	58.4	**62.8**	**53.4**	53.4	38.3	58.6	50.1
+ NP-OPD (α=0.5)	**39.1**	58.2	62.8	53.4	54.4	37.5	59.8	50.6
+ NP-OPD (α=0.75)	38.6	56.8	61.1	52.2	53.2	38.7	**62.4**	51.4
+ NP-OPD (α=1)	37.6	55.8	55.4	49.6	**54.9**	**39.3**	61.9	**52.0**
Qwen3-8B	40.1	50.9	67.1	52.7	63.2	39.5	58.0	53.6
OPD	**41.6**	**59.6**	**68.0**	**56.4**	60.0	39.3	57.3	52.2
+ NP-OPD (α=0.25)	41.1	57.2	63.9	54.1	59.6	40.1	58.2	52.6
+ NP-OPD (α=0.5)	41.2	56.7	60.5	52.8	59.9	40.9	58.7	53.2
+ NP-OPD (α=0.75)	39.0	56.2	66.8	54.0	60.0	40.7	58.7	53.1
+ NP-OPD (α=1)	37.6	47.6	65.8	50.3	**60.5**	**41.8**	**60.8**	**54.4**

Table C.11: **Full results of Qwen3 models on mathematical reasoning benchmarks with ExOPD (thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-1.7B	53.1	35.0	85.0	22.9	87.2	39.7	97.8	60.1
ExOPD	**59.0**	40.6	88.4	25.8	87.6	**42.0**	98.5	63.1
+ NP-OPD (α=0.5)	56.7	40.8	87.5	**32.1**	87.4	41.6	98.2	**63.5**
+ NP-OPD (α=1)	57.9	**41.5**	**88.8**	27.5	**87.8**	41.8	**98.6**	63.4
Qwen3-4B	72.9	61.9	95.9	44.2	91.2	45.1	99.1	72.9
ExOPD	69.6	63.3	97.8	41.7	90.6	47.2	**99.7**	72.8
+ NP-OPD (α=0.5)	73.3	63.1	98.4	47.5	89.6	47.4	99.2	74.1
+ NP-OPD (α=1)	**78.8**	**68.1**	**99.4**	**48.8**	**90.8**	**47.8**	98.7	**76.0**
Qwen3-8B	75.6	65.6	94.1	46.7	91.8	39.7	99.4	73.3
ExOPD	76.5	63.3	96.6	44.2	**92.2**	47.4	99.3	74.2
+ NP-OPD (α=0.5)	76.2	66.0	97.5	49.2	91.6	47.5	**99.7**	75.4
+ NP-OPD (α=1)	**78.5**	**69.8**	**97.8**	**50.0**	90.2	**47.7**	99.0	**76.1**

Table C.12: **Full results of Qwen3 models on code and science benchmarks with ExOPD (thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

Model	Code				Science			
	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-1.7B	22.5	33.5	44.6	33.5	47.5	28.7	42.4	39.5
ExOPD	26.2	**40.6**	47.9	38.2	<u>54.5</u>	<u>30.4</u>	**45.2**	**43.4**
+ NP-OPD (α=0.5)	**27.7**	39.7	**50.8**	**39.4**	52.8	**30.6**	44.5	42.6
+ NP-OPD (α=1)	<u>27.5</u>	<u>40.3</u>	<u>49.2</u>	<u>39.0</u>	**55.7**	29.3	<u>44.7</u>	<u>43.2</u>
Qwen3-4B	39.7	58.9	60.8	53.2	57.0	38.5	60.0	51.8
ExOPD	**44.7**	**64.3**	<u>67.1</u>	**58.7**	<u>56.3</u>	**39.3**	58.9	51.5
+ NP-OPD (α=0.5)	41.2	57.0	63.4	53.9	55.2	38.7	<u>60.9</u>	<u>51.6</u>
+ NP-OPD (α=1)	<u>43.1</u>	<u>61.1</u>	**68.5**	<u>57.6</u>	**58.3**	<u>39.0</u>	**62.3**	**53.2**
Qwen3-8B	40.1	50.9	67.1	52.7	63.2	39.5	58.0	53.6
ExOPD	**47.2**	<u>63.6</u>	**70.1**	**60.3**	<u>63.9</u>	41.0	56.0	<u>53.6</u>
+ NP-OPD (α=0.5)	43.7	61.5	68.4	57.9	60.5	<u>41.5</u>	<u>58.4</u>	53.5
+ NP-OPD (α=1)	<u>44.8</u>	**63.9**	<u>70.1</u>	<u>59.6</u>	**64.1**	**42.8**	**60.7**	**55.9**

Table C.13: **Full results of Qwen3 models on mathematical reasoning benchmarks with OPD^2 (thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-1.7B	53.1	35.0	85.0	22.9	87.2	39.7	97.8	60.1
OPD^2	**62.7**	40.0	88.8	29.2	87.8	<u>42.6</u>	<u>99.0</u>	64.3
+ NP-OPD (α=0.5)	<u>61.9</u>	**43.8**	<u>88.8</u>	**32.1**	**88.6**	42.5	**99.5**	**65.3**
+ NP-OPD (α=1)	60.8	<u>42.9</u>	**90.6**	<u>30.4</u>	<u>88.2</u>	**42.8**	98.7	<u>64.9</u>
Qwen3-4B	72.9	61.9	95.9	44.2	91.2	45.1	99.1	72.9
OPD^2	70.6	64.2	96.2	48.3	<u>91.6</u>	<u>47.8</u>	99.2	74.0
+ NP-OPD (α=0.5)	<u>77.7</u>	<u>68.8</u>	<u>98.4</u>	**53.8**	**91.8**	**48.2**	<u>99.2</u>	**76.8**
+ NP-OPD (α=1)	**77.9**	**70.4**	**98.8**	<u>50.0</u>	91.4	47.0	**99.3**	<u>76.4</u>
Qwen3-8B	75.6	65.6	94.1	46.7	91.8	39.7	99.4	73.3
OPD^2	73.8	66.5	<u>97.8</u>	47.9	<u>91.8</u>	47.6	**99.7**	75.0
+ NP-OPD (α=0.5)	**81.7**	<u>70.6</u>	**98.4**	<u>50.4</u>	**91.8**	**48.7**	98.3	<u>77.1</u>
+ NP-OPD (α=1)	<u>81.5</u>	**71.2**	97.8	**52.9**	90.6	<u>48.5</u>	<u>99.3</u>	**77.4**

Table C.14: **Full results of Qwen3 models on code and science benchmarks with OPD^2 (thinking mode).** The best and second-best results among the OPD variants are shown in bold and underlined, respectively.

Model	Code				Science			
	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-1.7B	22.5	33.5	44.6	33.5	47.5	28.7	42.4	39.5
OPD^2	26.3	37.7	51.5	38.5	<u>58.5</u>	<u>29.6</u>	44.5	44.2
+ NP-OPD (α=0.5)	**30.2**	**43.9**	**54.8**	**43.0**	**60.2**	**30.2**	<u>44.7</u>	**45.0**
+ NP-OPD (α=1)	<u>28.0</u>	<u>42.5</u>	<u>52.8</u>	<u>41.1</u>	58.3	29.4	**45.9**	<u>44.6</u>
Qwen3-4B	39.7	58.9	60.8	53.2	57.0	38.5	60.0	51.8
OPD^2	<u>44.0</u>	<u>63.1</u>	68.1	58.4	**57.5**	<u>40.0</u>	59.4	52.3
+ NP-OPD (α=0.5)	**44.5**	**64.8**	<u>72.4</u>	**60.6**	56.0	**40.5**	<u>61.2</u>	<u>52.6</u>
+ NP-OPD (α=1)	43.5	62.9	**73.0**	<u>59.8</u>	<u>57.3</u>	39.3	**62.7**	**53.1**
Qwen3-8B	40.1	50.9	67.1	52.7	63.2	39.5	58.0	53.6
OPD^2	46.5	<u>65.5</u>	**72.1**	<u>61.4</u>	62.3	41.3	53.8	52.5
+ NP-OPD (α=0.5)	**48.0**	**66.0**	71.0	**61.7**	<u>63.8</u>	**42.4**	<u>59.6</u>	<u>55.3</u>
+ NP-OPD (α=1)	<u>46.7</u>	62.8	<u>71.4</u>	60.3	**64.2**	<u>42.1</u>	**60.6**	**55.6**

C.3 Additional Experiments

Training efficiency. Tab. C.15 reports the wall-clock training time of Qwen3-1.7B on a single node with 8 H100 GPUs. The negative-policy rollouts are generated in advance, whereas the student portion of each batch requires on-policy rollout generation during training. Accordingly, the training time generally decreases as α increases. For standard OPD, using only negative-policy rollouts ($\alpha = 1$) reduces the training time from 9h19m to 3h34m in thinking mode and from 5h15m to 1h16m in non-thinking mode.

Table C.15: **Wall-clock training time of Qwen3-1.7B on a single node with 8×H100 GPUs.** α denotes the proportion of negative-policy rollouts. Gray numbers show the speedup relative to α=0.

		Proportion of negative-policy rollouts (α)				
Mode	-	0	0.25	0.5	0.75	1
Thinking	OPD	9h 19m	8h 09m (1.14×)	6h 57m (1.34×)	6h 04m (1.54×)	3h 34m (2.61×)
Non-thinking	OPD	5h 15m	5h 11m (1.01×)	4h 34m (1.15×)	3h 50m (1.37×)	1h 16m (4.14×)

Generation length. As shown in Tab. C.16, on-policy training with OPD, ExOPD, and OPD^2 increases generation length relative to the base models on both AIME24 and AIME25. Compared with on-policy training, NP-OPD reduces generation length for Qwen3-4B and Qwen3-8B across all three reward formulations, while slightly increasing it for Qwen3-1.7B.

Table C.16: **Mean generation length (in thousands of tokens) in thinking mode.** Entries for OPD variants show training with on-policy rollouts (α=0.0) → NP-OPD with α=1.0.

Benchmark	Method	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
AIME24	Base	17.0	14.2	15.1
	OPD	20.9 → 21.5	18.5 → 15.3	17.0 → 15.1
	ExOPD	20.7 → 21.1	18.7 → 16.0	18.1 → 15.9
	OPD^2	20.1 → 21.1	19.0 → 16.6	18.6 → 16.1
AIME25	Base	18.0	18.1	18.3
	OPD	22.9 → 23.5	20.5 → 18.3	19.9 → 18.0
	ExOPD	22.4 → 22.9	20.7 → 18.7	20.6 → 18.6
	OPD^2	22.4 → 22.7	21.1 → 19.2	20.7 → 18.8

Model merging. For Qwen3-8B in thinking mode, the best rollout mixing ratio varies across domains. Motivated by this observation, we merge models trained with $\alpha = 0$ and $\alpha = 1$ by averaging their parameters with equal weights. For ExOPD, Tabs. C.17 and C.18 show improvements over the on-policy baseline from 74.2 to 76.3 on Math, from 60.3 to 60.7 on Code, and from 53.6 to 54.7 on Science. For OPD^2, Tabs. C.19 and C.20 show similar improvements, from 75.0 to 77.4 on Math, from 61.4 to 62.3 on Code, and from 52.5 to 54.2 on Science. These results suggest that model merging can combine some of the benefits of using on-policy rollouts and negative rollouts.

Table C.17: **Model merging with ExOPD on Qwen3-8B (thinking mode).** *merge* averages the weights of the ExOPD model (α=0) and the NP-OPD (α=1) model with equal weights. The best and second-best results among the ExOPD variants are shown in bold and underlined, respectively.

Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-8B	75.6	65.6	94.1	46.7	91.8	39.7	99.4	73.3
ExOPD	76.5	63.3	96.6	44.2	**92.2**	47.4	<u>99.3</u>	74.2
+ NP-OPD (α=1)	<u>78.5</u>	**69.8**	**97.8**	**50.0**	90.2	<u>47.7</u>	99.0	<u>76.1</u>
+ merge (α=0 ⊕ 1)	**80.0**	<u>67.9</u>	**97.8**	<u>49.6</u>	<u>91.8</u>	**47.9**	<u>99.3</u>	**76.3**

Table C.18: **Model merging with ExOPD on Qwen3-8B: code and science benchmarks (thinking mode).** *merge* averages the weights of the ExOPD model (α=0) and the NP-OPD (α=1) model with equal weights. The best and second-best results among the ExOPD variants are shown in bold and underlined, respectively.

Model	Code				Science			
	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-8B	40.1	50.9	67.1	52.7	63.2	39.5	58.0	53.6
ExOPD	**47.2**	63.6	70.1	60.3	63.9	41.0	56.0	53.6
+ NP-OPD (α=1)	44.8	63.9	70.1	59.6	**64.1**	**42.8**	**60.7**	**55.9**
+ merge (α=0 $\oplus$ 1)	46.6	**65.0**	**70.7**	**60.7**	63.6	41.4	59.3	54.7

Table C.19: **Model merging with OPD^2 on Qwen3-8B: mathematical reasoning benchmarks (thinking mode).** *merge* averages the weights of the OPD^2 model (α=0) and the NP-OPD (α=1) model with equal weights. The best and second-best results among the OPD^2 variants are shown in bold and underlined, respectively.

Model	AIME24	AIME25	AMC23	HMMT25	MATH500	Olympiad	RGMath	**Avg**
Qwen3-8B	75.6	65.6	94.1	46.7	91.8	39.7	99.4	73.3
OPD^2	73.8	66.5	97.8	47.9	**91.8**	47.6	**99.7**	75.0
+ NP-OPD (α=1)	81.5	**71.2**	97.8	**52.9**	90.6	48.5	99.3	**77.4**
+ merge (α=0 $\oplus$ 1)	**82.1**	70.0	**98.4**	51.7	91.6	48.0	**99.7**	**77.4**

Table C.20: **Model merging with OPD^2 on Qwen3-8B: code and science benchmarks (thinking mode).** *merge* averages the weights of the OPD^2 model (α=0) and the NP-OPD (α=1) model with equal weights. The best and second-best results among the OPD^2 variants are shown in bold and underlined, respectively.

Model	Code				Science			
	Code Forces	LCBv5	RG Algo	**Avg**	GPQA	Super GPQA	Sci Bench	**Avg**
Qwen3-8B	40.1	50.9	67.1	52.7	63.2	39.5	58.0	53.6
OPD^2	46.5	65.5	72.1	61.4	62.3	41.3	53.8	52.5
+ NP-OPD (α=1)	46.7	62.8	71.4	60.3	**64.2**	42.1	**60.6**	**55.6**
+ merge (α=0 $\oplus$ 1)	47.8	65.0	**74.0**	**62.3**	63.4	41.7	57.4	54.2

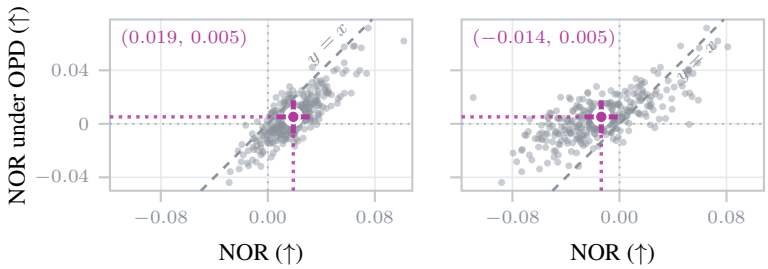
C.4 Details on Section 4.3

Details of NSP analysis. We compute NSP using changes in the student's log-probability assigned to each realized token relative to the initial student. Within each trajectory, we select the top 5% of token positions by absolute log-probability change and retain only those with negative changes. NSP is the fraction of retained positions where the negative policy (Qwen3-0.6B) assigns higher probability to the token than the teacher model (Qwen3-8B). To clearly contrast different rollout sources, we analyze NP-OPD and alternative rollout settings with $\alpha = 1$ and compare them with the on-policy baseline ($\alpha = 0$). NSP values are averaged over three repetitions of analysis-trajectory sampling, with standard errors of the mean (SEMs) ranging from 0.0004 to 0.0033 across settings. Error bars are omitted from the figures for visual clarity.

Details of NOR analysis. We use a fixed set of 300 training prompts, with 100 prompts from each of Math, Code, and Science. The negative reference and teacher are fixed to Qwen3-0.6B and Qwen3-8B, respectively, so their top predictions and the disagreement positions are shared across all settings. To clearly contrast different rollout sources, we analyze NP-OPD and alternative rollout settings with $\alpha = 1$ and compare them with the on-policy baseline ($\alpha = 0$).

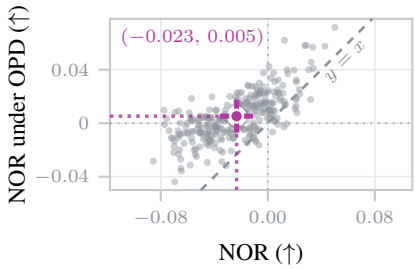
NOR analysis with different rollouts. We extend the NOR analysis to rollout models of different sizes within the Qwen3 family. As shown in Fig. C.1, the mean NOR decreases from 0.019 with

Qwen3-0.6B rollouts to -0.014, -0.023, and -0.030 with Qwen3-1.7B, Qwen3-4B, and Qwen3-8B rollouts, respectively. Only the lower-capability negative policy yields a higher mean NOR than on-policy OPD (0.005). This pattern supports our use of lower-capability rollouts as a negative reference and suggests that changing the rollout source alone does not explain the observed shift.

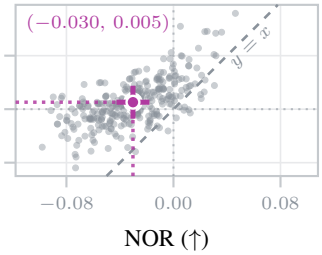


(a) Qwen3-0.6B rollouts (negative)

(b) Qwen3-1.7B rollouts



(c) Qwen3-4B rollouts



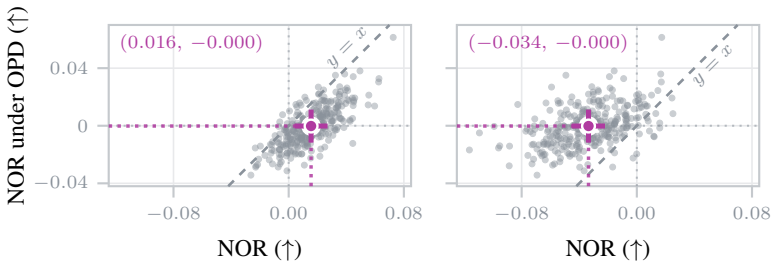
(d) Qwen3-8B rollouts (teacher)

Figure C.1: **Reduction in overlap across rollout policies.** Same quantity as Fig. 4, with the rollout policy varied over the Qwen3 family: (a) 0.6B (negative policy), (b) 1.7B, (c) 4B, and (d) 8B (teacher). Each point compares trajectory-level NOR under OPD (y-axis) with NOR under the corresponding rollouts (x-axis). Only 0.6B rollouts place most trajectories below the diagonal and the paired gap decreases monotonically with the size of the rollout policy.

NOR analysis on held-out prompts. We further evaluate NOR on held-out prompts to examine whether the reduction in overlap with the negative policy also appears beyond the prompts used during training. We use 300 prompts from nine benchmarks that are not included in the training data: 120 math prompts, with 30 each from AIME24, AIME25, HMMT, and OlympiadBench; 120 science prompts, with 45 from GPQA-Diamond, 40 from SciBench, and 35 from SuperGPQA; and 60 code prompts, with 35 from CodeForces and 25 from RGAlgorithmic. Using the same model configuration and checkpoints as in Sec. 4.3, we generate one trajectory per prompt from the initial student, yielding 300 trajectories in total. As shown in Fig. C.2, the same trend observed on training prompts also holds for these held-out examples. Most trajectories lie below the diagonal, indicating greater relative overlap reduction under NP-OPD than under OPD on held-out prompts. These results show that the effect is not limited to the prompts used during training and generalizes to unseen benchmark examples.

D Details of Section 5

Persona prompt. For the persona-prompt setting in Sec. 5, we use the following prompt to instruct the student model to respond as an elementary school student:



(a) Negative rollouts (b) Teacher rollouts

Figure C.2: **Reduction in overlap with the negative policy on held-out prompts.** Same quantity as Fig. 4, measured on 300 held-out prompts drawn from nine benchmarks spanning mathematics, science, and code, none of which appear in training. Each point is one trajectory, comparing NOR under OPD (y-axis) with (a) negative-policy or (b) teacher rollouts (x-axis).

Persona prompt

You are an elementary school student. You have only learned basic school material: simple arithmetic, fractions, basic shapes, and everyday science facts. You do not know advanced methods like algebra tricks, calculus, formal proofs, or real programming, only very simple ideas. Think and write the way a young student would: use simple words, take small easy steps, and always pick the easiest method you know. If a problem seems too hard, just make your best guess using what you know.

Layer drop. For the layer-drop setting in Sec. 5, we remove the last three transformer layers from the student model.