DR. OPD: LEARNING WHAT TO FOLLOW FOR OPTIMAL ON-POLICY DISTILLATION OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) trains a student on its own generated responses using dense, token-level supervision from a stronger teacher. Vanilla OPD treats all teacher signals equally, assuming that the teacher's supervision is equally important for every token. However, teacher signals at different tokens may have very different effects on the student's performance: some correct important reasoning errors, while others have little effect on the final answer. Motivated by this observation, we introduce **Dr. OPD** (*OPD Done Right*), which defines the optimal weighted OPD to maximize the student's performance. We formulate Dr. OPD as a bilevel optimization problem in which the student learns from weighted teacher supervision, while the weights are selected to maximize the expected reward of the resulting student. To solve Dr. OPD, we develop an efficient iterative solver that updates the token weights and student policy alternatively. At each round, it updates weights in closed form and then takes one gradient step on the resulting weighted OPD objective. Under regularity conditions, we show that this weighted update achieves a higher expected reward than a vanilla OPD update. Empirically, across strong-to-weak and same-size distillation on math and code, Dr. OPD consistently outperforms all evaluated baselines. In particular, in the strong-to-weak distillation setting, Dr. OPD improves average math performance by 9.7 points over vanilla OPD, and enables the smaller student to surpass its larger teacher.

1 INTRODUCTION

On-policy distillation (OPD) has become an important post-training approach for transferring capabilities from a stronger teacher to a student model that is cheaper to deploy or has undergone less training (Agarwal et al., 2024; Xu et al., 2026; Team et al., 2026). Unlike outcome-reward reinforcement learning methods like RLOO and GRPO (Ahmadian et al., 2024; Shao et al., 2024), which update the model based on a scalar reward for an entire generated trajectory, OPD leverages the teacher model to provide dense, token-level supervision along with the student's generated responses (Gu et al., 2024). Specifically, OPD trains the student by minimizing the divergence between the next-token distributions of the student and the teacher at each token visited by the student. Therefore, it provides a learning signal at every generated token rather than only at the end of a response.

Vanilla OPD assigns equal importance to the teacher's supervisions across tokens. However, given a long response generated by the student, should every teacher signal be treated equally? Consider a student solving a math problem. The teacher could either correct a crucial reasoning step or simply prefer different wording under the same mathematical meaning. These two signals are supposed to enter the distillation loss with different weights, as their effects on the final answer can be very different. Indeed, recent work has likewise shown that not every signal from the teacher across each token is equally useful to the student (Armandpour et al., 2026; Xie et al., 2026; Xing et al., 2026).

To study how the student should trust teacher signals across tokens, we revisit the goal of OPD. In particular, note that the purpose of distillation is ultimately to improve the student's task performance. Minimizing the divergence towards the stronger teacher policy is only an approach used to achieve this goal. Indeed, the teacher signal only tells us what the teacher prefers at each token, but not whether following the preference improves the student's performance. This motivates us to adaptively

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.

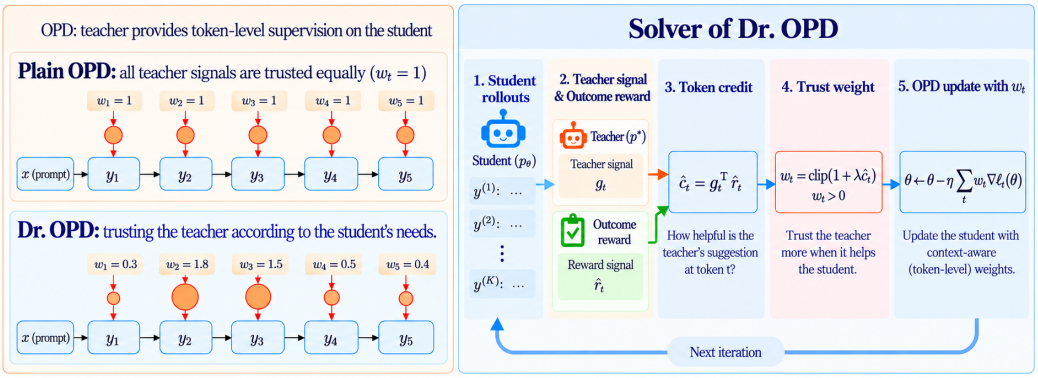


Figure 1: **Illustration of Dr. OPD.** Left: vanilla OPD assigns the same weight to every teacher signal, while Dr. OPD adapts the weights according to their benefit to the student. Right: student rollouts receive teacher and outcome-reward signals, whose gradient alignment determines token credits. These credits are converted into weights for distillation, where λ controls the weighting strength.

choose the weight of each teacher signal according to its usefulness for the student. In this paper, we therefore study how to weight token-level teacher signals so that the student achieves the best performance.

Following this principle, we introduce **Dr. OPD** (*OPD Done Right*), which assigns weights $w(s, v)$ to each teacher signal on the tokens v and context $s = (x, y)$ according to its usefulness for improving the student, as illustrated in Figure 1. A weight $w_t > 1$ places more trust in the teacher at token t, while $0 < w_t < 1$ places less trust. Vanilla OPD sets equal weights of $w_t \equiv 1$ across all tokens. We note that for any positive weight w, the corresponding w-weighted OPD remains a valid distillation objective that pushes the student toward the teacher policy, just as vanilla OPD does. Dr. OPD then selects, among these valid weights, the one that gives the best resulting student:

$$\max_{w} \Big\{ \text{Student performance after training with } w\text{-weighted OPD} \Big\}. \tag{1}$$

The performance is measured by an outcome reward based on its generated responses (Ouyang et al., 2022; Shao et al., 2024). We formally formulate Dr. OPD as a bilevel optimization in Section 3.

Solving (1) exactly would require repeatedly training the student under different weight functions and then selecting the best weight, which is computationally infeasible. We therefore develop an iterative solver, illustrated in Figure 1. At each round, we build a surrogate objective of (1) to update the weights, leading to a simple closed-form update instead of learning a complex weight function. Then the student performs a weighted OPD update. Specifically, the student generates its own responses, from which we obtain the teacher's token-level supervision and the outcome reward signal. We use these two signals to assess whether following the teacher more closely at each token would improve the student's performance. Under regularity conditions, we show that the student, after the weighted update, achieves a higher reward than the corresponding vanilla OPD update.

To summarize, we make the following contributions:

- **Bilevel Formulation of Distillation.** We introduce the concept of learning useful signals from the teacher via weighted OPD and propose Dr. OPD, a bilevel optimization formulation that defines the optimal weighted OPD to maximize the student's performance. The lower-level problem trains the student under a given weight function, while the upper-level problem selects the weights according to the performance of the resulting student.
- **Efficient Algorithm.** Despite the challenge of solving Dr. OPD directly, we develop an efficient iterative solver with a closed-form weight update followed by a gradient update of the weighted OPD. To incorporate outcome rewards into the weight updates, it requires computing an inner product between two parameter-dimensional gradients for every token, which is prohibitively expensive for large models. We instead express each inner product as the product of two scalars, and then these products for all tokens can be computed altogether using a single Jacobian–vector product (JVP); see Section 4 for details.
- **Significant Empirical Gains.** We evaluate Dr. OPD across strong-to-weak and same-size distillation on math and code tasks, where it achieves the best performance across all evaluated

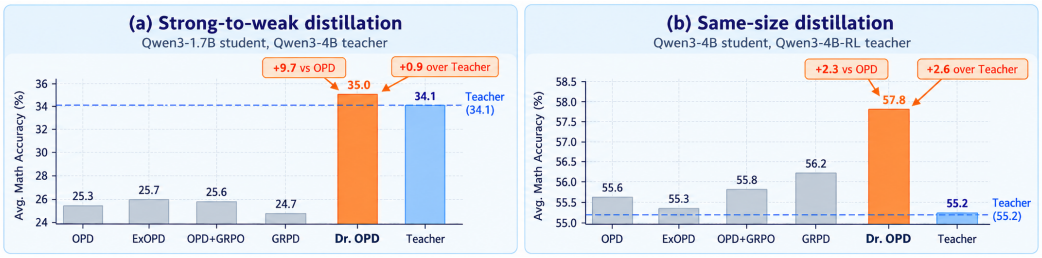


Figure 2: **Empirical gains.** Across both distillation setups, Dr. OPD consistently outperforms existing baselines and even produces a student that surpasses its teacher (Full results in Section 5.2).

settings and consistently outperforms all baselines. As shown in Figure 2, Dr. OPD improves Qwen3-1.7B over vanilla OPD by $+9.7$ points on math and even surpasses the larger Qwen3-4B teacher (35.0 vs. 34.1). In same-size distillation, it also surpasses both RL-trained teachers.

2 Weighted On-Policy Distillation

OPD trains a student model on its generated responses under the supervision from a stronger frozen teacher model. Let x be a prompt, and $y = (y_1, \ldots, y_T)$ a response generated by the student. At step t, the context $s_t = (x, y_{<t})$ contains the prompt and all the preceding tokens generated before t. We denote $p_\theta(v \mid s_t)$ as the student's probability of generating the next token $v \in \mathcal{V}$, and $p^*(v \mid s_t)$ as the teacher's probability in the same context, where $\mathcal{V}$ is the vocabulary of tokens.

Vanilla OPD aims to train the student policy to mimic the teacher policy using the reverse KL divergence:

$$\min_\theta D(\theta) := \mathbb{E}\left[\sum_t \mathrm{KL}\left(p_\theta(\cdot \mid s_t) \,\|\, p^*(\cdot \mid s_t)\right)\right], \tag{2}$$

where

$$\mathrm{KL}\left(p_\theta(\cdot \mid s) \,\|\, p^*(\cdot \mid s)\right) = \sum_v p_\theta(v \mid s) \log \frac{p_\theta(v \mid s)}{p^*(v \mid s)},$$

and the expectation is taken over the prompts and responses generated by the student policy. Following prior works (Agarwal et al., 2024; Shao et al., 2026), the generated contexts are treated as fixed during each student update and are refreshed as the student changes.

As shown in (2), vanilla OPD assigns equal weights to teacher signals at every token. However, not all teacher signals are equally informative (Armandpour et al., 2026; Xie et al., 2026). We introduce a weight function $w(s, v) > 0$, which controls the importance of matching the teacher's probability of token v at context s. For a fixed context s, we use $w(s, \cdot)$ to denote the corresponding vector of token weights. We then define a weighted KL divergence that generalizes the ordinary KL divergence.

Definition 1 (Weighted KL divergence)**.** At a context s, given the positive weights $w(s, \cdot)$, we define

$$\begin{aligned}&\mathrm{KL}_{w(s,\cdot)}(p_\theta(\cdot \mid s) \,\|\, p^*(\cdot \mid s)) \\ &\quad := \sum_v w(s, v)\left[p_\theta(v \mid s) \log \frac{p_\theta(v \mid s)}{p^*(v \mid s)} - p_\theta(v \mid s) + p^*(v \mid s)\right].\end{aligned} \tag{3}$$

where the additional term $-p_\theta(v \mid s) + p^*(v \mid s)$ ensures nonnegativity under token-dependent weighting, following the generalized KL divergence (also known as I-divergence) with nonnegative measures (Csiszár, 1975; Banerjee et al., 2005). This weighted KL divergence equals zero if and only if $p_\theta(v \mid s) = p^*(v \mid s)$ for every token v. Moreover, when $w(s, v) \equiv 1$ for all v, it holds $\sum_v w(s, v)\big(p^*(v \mid s) - p_\theta(v \mid s)\big) = 0$, so (3) reduces to the ordinary KL divergence.

With the weighted KL divergence, we define the weighted OPD as follows.

Definition 2 (Weighted OPD)**.** Given a weight function $w(s, \cdot) > 0$ for every context s, we define weighted OPD by

$$\min_\theta D_w(\theta) := \mathbb{E}\left[\sum_t \mathrm{KL}_{w(s_t,\cdot)}(p_\theta(\cdot \mid s_t) \,\|\, p^*(\cdot \mid s_t))\right]. \tag{4}$$

We next show that this remains a valid distillation objective. In particular, minimizing $D_w(\theta)$, for any positive w, still pushes the student to match the teacher.

Theorem 3 (Validity of weighted OPD)**.** *For any positive weight function w, we have*

$$D_w(\theta) \geq 0.$$

Equality holds if and only if $p_\theta(v \mid s) = p^*(v \mid s)$ for every token v at almost every context under the reference context distribution. When $w(s, v) \equiv 1$, weighted OPD in (4) reduces to vanilla OPD.

3 OPD Done Right: Find the Best Weight for OPD

We now study how to choose the weight function w. Recall that the ultimate goal of distillation is to improve the student's task performance, while minimizing the divergence to a stronger teacher is merely a means to achieve this goal. This principle motivates us to select the weight function according to the performance of the resulting student, formulating a bilevel optimization problem.

3.1 Formulation of Dr. OPD

We evaluate the student's performance using an outcome reward $V(x, y) \in [0, 1]$ for each prompt–response pair (x, y). The student's expected reward is then

$$R(\theta) := \mathbb{E}_{x,\, y\sim p_\theta(\cdot|x)}\big[V(x, y)\big].$$

We now introduce **DR. OPD** *(OPD done right)*, the optimal weighted OPD to maximize the student's performance:

$$\max_{w_{\min}\leq w\leq w_{\max}} R(\theta^*(w)) \qquad \text{subject to} \qquad \theta^*(w) \in \arg\min_\theta D_w(\theta). \tag{5}$$

Dr. OPD is a bilevel optimization problem, where the upper level chooses the weight function w to maximize the resulting student's reward, and the lower level trains the student θ by minimizing $D_w(\theta)$, under the given w. Here, the constraint $w \in [w_{\min}, w_{\max}]$ means pointwise that $w(s, v) \in [w_{\min}, w_{\max}]$ for every (s, v), where $0 < w_{\min} < 1 < w_{\max}$ are prespecified bounds. For simplicity, we write $\theta^*(w)$ as if the lower-level solution were unique. When multiple lower-level solutions exist, we choose the one that maximizes the upper-level reward; see the formal definition in Appendix D.3.

3.2 Iterative Solver for Dr. OPD

Solving the bilevel Dr. OPD problem in (5) exactly is computationally intractable, since evaluating each candidate weight function w would require fully optimizing the lower-level student and then evaluating its reward. In addition, the weight function $w(s, v)$ is defined over all possible context–token pairs, making the upper-level optimization infinite-dimensional.

We therefore design an iterative solver that alternates between student and weight updates. At each round, starting from the current student $\bar\theta$, we take one gradient update of weighted OPD with the weights fixed. We then update the weights according to the reward of the resulting updated student, restricting the weights update to the sampled tokens in responses. This reduces the original infinite-dimensional functional optimization into a finite-dimensional problem at each step.

Lower-level update. For a fixed weight function w, we take one gradient descent step from the current student $\bar\theta$. We generate a response $y = (y_1, \ldots, y_T)$ from $p_{\bar\theta}$ and hold the sampled trajectory fixed during this round. Here, y_t is the realized token at step t, whereas the previous $v \in \mathcal{V}$ in (2) indexes all possible tokens. We write

$$w_t := w(s_t, y_t), \qquad p_t(\bar\theta) := p_{\bar\theta}(y_t \mid s_t), \qquad p_t^* := p^*(y_t \mid s_t).$$

Differentiating the weighted OPD objective at $\bar\theta$ gives

$$-\nabla_\theta D_w(\bar\theta) = \mathbb{E}\left[\sum_t w_t g_t\right], \quad \text{where} \quad g_t := \log\frac{p_t^*}{p_t(\bar\theta)}\,\nabla_\theta \log p_\theta(y_t \mid s_t)|_{\theta=\bar\theta}\,. \tag{6}$$

The derivation is provided in Appendix D.2.2. The corresponding one-step update on the lower-level solution is therefore

$$\theta^+(w) := \bar{\theta} - \eta \nabla D_w(\bar{\theta}) = \bar{\theta} + \eta\, \mathbb{E}\left[\sum_t w_t g_t\right], \tag{7}$$

where $\eta > 0$ is the student step size. The expectation in (6)–(7) is taken over the current policy $p_{\bar{\theta}}$.

We call g_t in (6) the teacher correction at token y_t, which is interpreted as follows. If the teacher assigns token y_t a higher probability than the student, then g_t increases the student's probability of generating y_t. On the other hand, if the teacher assigns it a lower probability, then g_t decreases its probability. The weight w_t controls the strength of this teacher correction to the student update.

Upper-level update. We next update the weight function w according to the reward of the updated student $\theta^+(w)$. Since the weight function is defined over all possible context–token pairs and we do not impose a parametric form on it, a gradient update like in (7) does not apply here. Instead, we construct a tractable lower bound approximation on $R(\theta^+(w))$ and optimize this bound over the weight function. As shown below, this leads to a simple pointwise update for the weights.

We denote the one-step vanilla OPD update from the current student $\bar{\theta}$ as follows, which sets $w \equiv 1$,

$$\theta^+_{\text{vani}} := \theta^+(1).$$

To determine how the weights should be updated, we ask how changing the weight w_t of the teacher signal at token t affects the resulting student's reward. We define the **credit** at token t as

$$c_t := \frac{1}{\eta} \frac{\partial R(\theta^+(w))}{\partial w_t}\bigg|_{w\equiv 1} = g_t^\top \nabla R\big(\theta^+_{\text{vani}}\big), \tag{8}$$

where the second equality follows from the chain rule and (7). The derivative is understood per unit reference measure; Appendix D.4 gives its precise functional interpretation.

The credit c_t measures whether following the teacher more strongly at token t would locally improve the student's reward. Recall that the teacher correction g_t in (6) locally moves the sampled token's probability toward the teacher's. However, moving closer to the teacher does not necessarily improve the student's task performance. The credit c_t captures this distinction. If $c_t > 0$, placing more weight on the teacher correction is beneficial to the student's reward, suggesting that we should trust this signal more and choose $w_t > 1$. On the other hand, if $c_t < 0$, following the teacher more strongly is harmful, suggesting that the signal should be downweighted with $w_t < 1$. Thus, g_t determines the direction of the token-level teacher correction, while c_t determines how strongly the student should follow the teacher signal.

We next derive a lower bound surrogate on the reward $R(\theta^+(w))$.

Proposition 4 (Lower bound on reward)**.** *Suppose that the reward gradient $\nabla R(\theta)$ is locally L_R-Lipschitz in θ and that the regularity conditions in Appendix D.1 hold. Then, for any weight function $w \in [w_{\min}, w_{\max}]$ and any $\lambda > 0$ satisfying $\lambda\eta L_R \mathbb{E}\big[\sum_t \|g_t\|^2\big] \le 1$.*

$$R\big(\theta^+(w)\big) \ge R\big(\theta^+_{\text{vani}}\big) + \eta\, \mathbb{E}\left[\sum_t \left((w_t - 1)c_t - \frac{(w_t-1)^2}{2\lambda}\right)\right]. \tag{9}$$

The right-hand side of (9) provides a tractable lower-bound surrogate for $R(\theta^+(w))$. The parameter $\lambda > 0$ is introduced to simplify the lower bound; see Appendix D.1 for details.

Maximizing this surrogate with respect to w yields a closed-form update

$$w_t^+ = \text{clip}\left(1 + \lambda c_t,\, w_{\min},\, w_{\max}\right). \tag{10}$$

The update implies that λ can also be interpreted as the hyperparameter to control the weighting strength, with larger values allowing stronger adjustments to token-level weights. Importantly, this closed-form update removes the need to parameterize or train a separate model for the weight function. Instead, we directly compute the weight w_t^+ for each sampled token, which substantially reduces the computational cost.

Note that the updated w^+ attains a higher reward than vanilla OPD, as long as $w_t^+ \not\equiv 1$.

Theorem 5 (Reward improvement over vanilla OPD)**.** *Under the conditions ensuring (9),*

$$R\big(\theta^+(w^+)\big) - R\big(\theta^+_{\text{vani}}\big) \ge \frac{\eta}{2\lambda}\mathbb{E}\left[\sum_t (w_t^+ - 1)^2\right] \ge 0. \tag{11}$$

4 ALGORITHM

In this section, we propose a practical algorithm to solve Dr. OPD. At each round, starting from the current student $\bar{\theta}$, we sample multiple responses per prompt and obtain the teacher signals and outcome-rewards. Such outcome rewards are readily available for verifiable tasks such as math and code through answer checking and test execution (Guo et al., 2025), and can also be provided by learned reward models for more general tasks (Ouyang et al., 2022). Given the signals from both the teacher and the outcome reward, we estimate the token credits c_t in (8) and update the weights as in (10). Lastly, we update the student with one gradient step on the resulting weighted OPD.

Computation of credits. Recall that $c_t = g_t^\top \nabla R(\theta^+_{\text{vani}})$, where the teacher correction g_t is directly computed as in (6) but estimating $\nabla R(\theta^+_{\text{vani}})$ can be computationally expensive. In particular, estimating $\nabla R(\theta^+_{\text{vani}})$ would require first applying a vanilla OPD update from the current $\bar{\theta}$ to obtain θ^+_{vani}, then generating a new set of responses from this updated student and using them to estimate its gradient of reward. This would introduce an additional round of model update and response generation at every training iteration, substantially increasing the computational cost.

To avoid the cost, we approximate $\nabla R(\theta^+_{\text{vani}})$ by $\nabla R(\bar{\theta})$ evaluated at the current student $\bar{\theta}$, which can be estimated using the responses already sampled. For a general outcome reward function, $\nabla R(\bar{\theta})$ can be estimated by weighting each response's log-probability gradient by its reward. For the verifiable math and code tasks, we can follow GRPO (Shao et al., 2024) to estimate $\nabla R(\bar{\theta})$ by

$$\widehat{r} := \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} A(x,y) \sum_t \nabla_\theta \log p_\theta(y_t \mid s_t)|_{\theta=\bar{\theta}}\,, \tag{12}$$

where $A(x,y)$ denotes the group-relative advantage with the explicit expression in Appendix C, and $\mathcal{B}$ denotes the batch of prompt-response pairs. We then approximate the token credit by $g_t^\top \widehat{r}$.

Efficient computation with Jacobian–vector product (JVP). For a response with T tokens, we shall compute credits $\widehat{c}_t = g_t^\top \widehat{r}$ for each token $t = 1, \ldots, T$. A naive implementation would require computing T parameter gradients $g_1, \ldots, g_T \in \mathbb{R}^{|\bar{\theta}|}$, one for each token. Since the number of model parameters $|\bar{\theta}|$ is very large, explicitly forming all $\{g_t\}_{t=1}^T$ is prohibitively expensive.

It turns out that we do not need to explicitly compute or store the parameter gradient g_t. Recall that

$$g_t = d_t\, \nabla_\theta \log p_\theta(y_t \mid s_t)|_{\theta=\bar{\theta}}\,, \qquad d_t := \log \frac{p^*(y_t \mid s_t)}{p_{\bar{\theta}}(y_t \mid s_t)} \in \mathbb{R}. \tag{13}$$

Substituting this expression into $\widehat{c}_t = g_t^\top \widehat{r}$ gives, for each token $t = 1, \ldots, T$,

$$\widehat{c}_t = d_t \left\langle \nabla_\theta \log p_\theta(y_t \mid s_t)|_{\theta=\bar{\theta}}\,, \widehat{r} \right\rangle = d_t \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon} \log p_{\bar{\theta}+\epsilon\widehat{r}}(y_t \mid s_t)\right|_{\epsilon=0}, \tag{14}$$

where the second equality follows from the directional-derivative identity. Thus, instead of explicitly constructing the T parameter-dimensional gradients $\{g_t\}_{t=1}^T$, we only need the T scalar directional derivatives $\left\{ \frac{\mathrm{d}}{\mathrm{d}\epsilon} \log p_{\bar{\theta}+\epsilon\widehat{r}}(y_t \mid s_t)\big|_{\epsilon=0} \right\}_{t=1}^T$, which can be computed altogether with a single JVP.

Updating weights. Following the closed-form update in (10), we update the token weights:

$$\widehat{w}_t = \mathrm{clip}\left(1 + \lambda \frac{\widehat{c}_t}{\max(\sigma, \epsilon)},\, w_{\min},\, w_{\max}\right), \tag{15}$$

where $\epsilon > 0$ prevents division by zero, and σ is the root-mean-squared (RMS) scale of the estimated credits, whose explicit expression is provided in Appendix C.3. This RMS normalization helps stabilize the weight updates.

We then perform one gradient step on the $\{\widehat{w}_t\}$-weighted OPD, as in (7), but with the population expectation replaced by an empirical average. The complete procedure is in Algorithm 1. Compared with vanilla OPD, the only additional steps are computing the gradient of the outcome rewards, the token credits, and the corresponding weights. Since the credits can be computed with a single JVP and the weights are obtained in closed form, these computations add little overhead.

Algorithm 1 Solver for DR-OPD

Require: Student p_θ, teacher p^*, outcome reward function V, weighting strength λ, bounds $w_{\min}, w_{\max}$

Set current student policy $\bar{\theta} \leftarrow \theta$
for each training round **do**
  Sample multiple responses per prompt from $\bar{\theta}$.
  Evaluate responses' outcome reward and compute the estimated gradient of reward $\widehat{r}$.
  Compute the token discrepancies d_t in (13).
  Compute credits $\widehat{c}_t$ by JVP using (14).
  Update weights $\widehat{w}_t$ using (15).
  Update $\bar{\theta}$ by gradient update of $\{\widehat{w}_t\}$-weighted OPD in (7).
end for

5 Experimental Results

We evaluate Dr. OPD across four student–teacher configurations:

$$(\text{Base}, \text{Instruct}), \quad (\text{Instruct}, \text{Instruct}), \quad (\text{Base}, \text{RL}), \quad (\text{Instruct}, \text{RL}),$$

where the first entry denotes the type of student model and the second denotes the type of teacher. These terms refer to different stages of post-training. A *Base* model is the pretrained model before instruction tuning, while an *Instruct* model has been post-trained to better follow user instructions and solve downstream tasks. An *RL* model is obtained through further reinforcement learning using outcome rewards and represents a more strongly post-trained model.

These configurations form two complementary distillation regimes. In *strong-to-weak* regime, a larger Instruct model serves as the teacher for either a Base or an Instruct student. In *same-size* regime, we use an RL-trained teacher and a base or Instruct model of the same size as the student.

5.1 Experimental setup

Models. We instantiate the four student–teacher configurations above using Qwen3 models, where models without the "Base" or "RL" suffix refer to their Instruct versions. For strong-to-weak distillation, we use Qwen3-4B (Yang et al., 2025) as the teacher and consider two students, Qwen3-1.7B-Base and Qwen3-1.7B. For same-size distillation, we consider Qwen3-4B-Base with an RL-trained Qwen3-4B-Base teacher, and Qwen3-4B with an RL-trained Qwen3-4B teacher (Yang et al., 2026). All teachers are frozen during distillation.

Training data and protocol. For mathematical reasoning, we train on DeepMath-103K (He et al., 2026). For code generation, we train on Eurus-RL-Code (Cui et al., 2025). Following prior OPD work (Yang et al., 2026; Sun et al., 2026b), we use a small number of post-training steps: 50 steps in all settings. Full training details are provided in Appendix B.

Evaluation. For math, we evaluate on AIME24, AIME25, AMC, Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024), and report accuracy averaged over 16 sampled responses per problem (avg@16). For code, we evaluate on HumanEval+ and MBPP+ from EvalPlus (Liu et al., 2023), together with LiveCodeBench (Jain et al., 2025), and report avg@4. We evaluate checkpoints every 10 optimization steps on the evaluated datasets and report the best checkpoint for each method, as adopted in works (Ding & Zhang, 2026; Zhao et al., 2026).

Baselines. We compare Dr. OPD with four OPD-based baselines. *OPD* (Gu et al., 2024) is the standard on-policy distillation baseline with uniform token weighting. *ExOPD* (Yang et al., 2026) extrapolates the teacher's distillation signal with the goal of moving the student beyond the teacher. With outcome reward signals, *GRPD* (Lin et al., 2026) considers reward-gated distillation, while *OPD+GRPO* (Xiaomi LLM-Core Team et al., 2026) adds an additional GRPO objective alongside the OPD objective. Implementation details are provided in Appendix B.

5.2 Distillation results

Strong-to-weak distillation. We distill the Qwen3-4B teacher into the Qwen3-1.7B and Qwen3-1.7B-Base students. As shown in Table 1, Dr. OPD achieves the best average performance on both

math and code for both students. Notably, vanilla OPD still leaves a substantial gap between the student and the stronger teacher, suggesting that simply mimicking a stronger, larger teacher is insufficient to bring the student close to the teacher's performance.

A smaller student can outperform its larger teacher. For the 1.7B student, Dr. OPD even surpasses the larger 4B teacher in average math performance (35.0 vs. 34.1). This improvement reflects the design of Dr. OPD, which does not merely mimic the teacher but also uses the student's outcome reward to determine which teacher signals to follow more closely. However, simply adding outcome rewards is not sufficient, since both OPD+GRPO and GRPD also use such rewards but yield only marginal improvements. This suggests that how the outcome reward is incorporated into distillation is important. For the 1.7B-Base student, Dr. OPD still improves over vanilla OPD by 1.8 points on math, while the gain is much larger on code, reaching 6.4 points.

	Math						Code			
Method	AIME24	AIME25	AMC	Minerva	Olym.	**Avg.**	HE+	MBPP+	LCB	**Avg.**
Teacher (Qwen3-4B)	22.5	16.9	60.5	27.1	43.7	34.1	77.1	64.8	16.1	52.7
Student: Qwen3-1.7B										
Initial	12.3	9.0	39.7	19.1	32.9	22.6	59.8	52.5	11.9	41.4
OPD	15.4	12.5	43.7	19.4	35.8	25.3	60.4	54.9	15.0	43.4
ExOPD	17.5	10.0	44.0	20.2	36.6	25.7	61.1	56.5	15.4	44.3
OPD+GRPO	16.0	13.1	42.9	21.0	35.1	25.6	60.7	55.4	14.9	43.7
GRPD	15.8	9.2	43.5	19.4	35.4	24.7	64.0	52.0	14.9	43.6
Dr. OPD	**25.6**$_{+10.2}$	**23.5**$_{+11.0}$	**56.9**$_{+13.2}$	**24.6**$_{+5.2}$	**44.4**$_{+8.6}$	**35.0**$_{+9.7}$	**64.2**$_{+3.8}$	**60.7**$_{+5.8}$	**20.4**$_{+5.4}$	**48.4**$_{+5.0}$
Student: Qwen3-1.7B-Base										
Initial	1.7	1.5	10.8	4.4	7.8	5.2	5.5	4.2	1.3	3.7
OPD	6.9	5.0	28.8	20.1	23.3	16.8	54.0	55.0	11.7	40.2
ExOPD	6.7	**5.6**	27.5	20.6	23.8	16.8	61.1	54.7	14.3	43.4
OPD+GRPO	8.1	5.2	29.3	21.0	23.7	17.5	63.1	55.1	15.0	44.4
GRPD	6.9	**5.6**	28.8	20.0	23.0	16.9	59.3	50.3	13.0	40.9
Dr. OPD	**9.2**$_{+2.3}$	4.8$_{-0.2}$	**31.3**$_{+2.5}$	**21.3**$_{+1.2}$	**26.5**$_{+3.2}$	**18.6**$_{+1.8}$	**64.3**$_{+10.3}$	**58.5**$_{+3.5}$	**16.9**$_{+5.2}$	**46.6**$_{+6.4}$

Table 1: Strong-to-weak distillation. We distill Qwen3-4B into Instruct and Base Qwen3-1.7B students on math and code generation. Colored subscripts report the absolute change over vanilla OPD. Bold denotes the best distillation result within each student setting.

Same-size distillation. We next consider the setting where the student and teacher have the same model size, with the teacher obtained through further RL training. As shown in Table 2, vanilla OPD already brings the student close to or even beyond the teacher, leaving limited room for further gains by simply mimicking the teacher. Despite this, Dr. OPD still achieves the best average performance for both setups, improving over vanilla OPD by 2.3 and 3.0 points, respectively.

Surpassing the RL-trained teacher. Notably, Dr. OPD surpasses the RL-trained teacher in both settings. For the Instruct student, it exceeds the teacher on all five benchmarks, a level of consistent improvement not achieved by any competing method. For the Base student, Dr. OPD exceeds the teacher on four of the five benchmarks and improves the average performance from 34.1 to 36.0.

5.3 Ablation Study: Effect of λ

The weighting strength λ is the main hyperparameter in Algorithm 1, controlling how strongly the token weights can deviate from the vanilla OPD value of 1. When $\lambda = 0$, Dr. OPD reduces to vanilla OPD. As shown in Table 3, every positive value of λ improves over OPD on all five benchmarks and in average performance, suggesting that Dr. OPD does not require precise tuning of λ. Among the values tested, $\lambda = 0.4$ performs best overall and is used in our main experiments.

6 Related Works

On-policy distillation. OPD trains a student on its own generated responses using supervision from a stronger teacher, as studied in the pioneering works GKD and MiniLLM (Agarwal et al., 2024; Gu et al., 2024). Recent work has further explored how teacher supervision should be modified or selectively applied, including reward extrapolation (Yang et al., 2026), agreement- or compatibility-based weighting (Xing et al., 2026; Xie et al., 2026; Sun et al., 2026b), and outcome-reward-guided

Method	AIME24	AIME25	AMC	Minerva	Olym.	**Avg.**
Student: Qwen3-4B		*Teacher: Qwen3-4B-RL*				
Teacher	54.0	46.5	81.0	36.9	57.7	55.2
Initial	22.5	16.9	60.5	27.1	43.7	34.1
OPD	55.8	45.8	81.0	37.3	57.9	55.6
ExOPD	56.9	43.3	82.3	37.6	56.6	55.3
OPD+GRPO	55.6	46.3	82.0	37.5	57.7	55.8
GRPD	55.8	49.2	81.6	37.6	57.0	56.2
Dr. OPD	**58.3**$_{+2.5}$	**51.7**$_{+5.8}$	**82.9**$_{+1.9}$	**37.8**$_{+0.5}$	**58.5**$_{+0.6}$	**57.8**$_{+2.3}$
Student: Qwen3-4B-Base		*Teacher: Qwen3-4B-Base-RL*				
Teacher	23.1	22.1	55.4	26.4	43.5	34.1
Initial	11.9	8.5	36.8	9.7	27.6	18.9
OPD	21.0	19.2	56.0	27.4	41.5	33.0
ExOPD	**26.9**	19.2	56.7	25.6	43.1	34.3
OPD+GRPO	21.9	**21.0**	57.5	27.2	42.3	34.0
GRPD	21.5	18.8	56.8	27.6	43.1	33.5
Dr. OPD	25.4$_{+4.4}$	20.0$_{+0.8}$	**58.4**$_{+2.4}$	**31.9**$_{+4.5}$	**44.1**$_{+2.6}$	**36.0**$_{+3.0}$

Table 2: Same-size distillation. We evaluate Instruct and Base students distilled from their corresponding RL-trained teachers on mathematical reasoning. Colored subscripts in the Dr. OPD rows report the absolute improvement over vanilla OPD. Teacher and Initial are reference points; bold denotes the best distillation result within each student–teacher setting.

λ	AIME24	AIME25	AMC	Minerva	Olym.	**Avg.**
0 (OPD)	15.4	12.5	43.7	19.4	35.8	25.3
0.2	18.5	13.5	47.8	22.8	37.7	28.1
0.3	19.4	14.0	45.0	23.2	37.5	27.8
0.4	**25.6**	**23.5**	**56.9**	24.6	**44.4**	**35.0**
0.5	24.0	19.2	54.7	**25.0**	42.6	33.1
0.6	23.8	19.8	54.1	23.3	44.0	33.0

Table 3: Effect of the weighting strength λ on mathematical reasoning. We vary λ for the Qwen3-1.7B student distilled from Qwen3-4B. $\lambda = 0$ reduces to vanilla OPD, while larger λ allows stronger token-level weight adjustments. Bold denotes the best result in each column.

selection (Lin et al., 2026; Xu & Zhang, 2026). Closely related to our motivation, Unmasking OPD studies whether teacher signals align with a success-improving direction (Armandpour et al., 2026). In contrast, Dr. OPD formulates the choice of token-level teacher weights based on the reward of the resulting student and interprets it as the optimal weighted OPD regarding the student's performance.

Bilevel optimization and reweighting. The formulation of Dr. OPD is related to bilevel optimization and data reweighting, where training weights are selected according to the performance of the resulting model (Franceschi et al., 2018; Shaban et al., 2019). In particular, Ren et al. (2018) uses a one-step learner update to select example weights from a validation objective. Related influence-based methods quantify how changing the weight of a training example affects the learned model (Koh & Liang, 2017; Pruthi et al., 2020). Dr. OPD applies this principle to token-level teacher supervision, where the outer objective is the student's outcome reward and the inner problem trains the student with weighted OPD. A more detailed discussion of related work is provided in Appendix A.

7 Conclusion

We proposed **Dr. OPD**, a bilevel optimization formulation that finds the optimal weighted OPD objective for improving students' task performance. The lower-level problem trains the student under a given weight function, while the upper-level problem selects the weights according to the reward of the resulting student. To solve Dr. OPD efficiently, we developed an iterative solver with closed-form weight updates followed by a standard gradient update on the weighted OPD. Across all evaluated experimental setups, Dr. OPD consistently outperforms existing baselines and can even enable students to surpass their teachers.

AI USE STATEMENT

In this work, we used generative AI tools as follows. Among tasks requiring disclosure, AI tools assisted with implementing and debugging code, preparing scripts for experiments, and checking and refining mathematical derivations and proofs. All AI-assisted code was inspected and executed by the authors, all reported experimental results were obtained from our own training and evaluation runs, and all AI-assisted mathematical arguments were independently verified by the authors.

We did not use generative AI tools for research ideation, developing the proposed method, designing the experiments, selecting hyperparameters, or providing methodological feedback. Generative AI tools did not autonomously run experiments or generate experimental results.

Among the tasks for which disclosure is recommended, we used generative AI tools to assist with producing scientific figures and plotting code, as well as drafting and editing the manuscript for clarity and readability. All AI-assisted text were reviewed and revised by the authors. We take full responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We are not aware of any ethical or societal consequences of this work that require specific discussion or highlighting.

REPRODUCIBILITY STATEMENT

Appendix D collects the assumptions and complete proofs for the formulations and statements in Sections 2–3.2. Appendix B collects all experimental settings, training data, and prompt templates. Appendix C specifies more details for Dr. OPD implementation, including credit normalization and weight clipping. Supplementary results appear in Appendix E. The datasets and models are publicly available and cited in the paper.

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*, pp. 21246–21263, 2024.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc NM Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why. *arXiv preprint arXiv:2605.10889*, 2026.

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749, 2005.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement, 2026.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pp. 1568–1577. PMLR, 2018.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 138306–138322, 2026.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*, 2021.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation for multilingual math reasoning. *arXiv preprint arXiv:2608.05802*, 2026.

Naman Jain, 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 *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. Pmlr, 2017.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*, 2023.

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.

Wenze Lin, Jiale Zhao, Xitai Jiang, Songde Rao, Yining Li, Shenzhi Wang, Bingxiang He, and Gao Huang. On-policy distillation with verifiable reward. *arXiv preprint arXiv:2608.24696*, 2026.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

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*, 35:27730–27744, 2022.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in neural information processing systems*, 33: 19920–19930, 2020.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343. PMLR, 2018.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd international conference on artificial intelligence and statistics*, pp. 1723–1732. PMLR, 2019.

Bing Shao, Jiazheng Zhang, Long Ma, Yujiong Shen, Senjie Jin, Xin Guo, Yuming Yang, Mingxu Chai, Zhiheng Xi, Boyang Liu, Junlin Shang, Tao Gui, Qi Zhang, and Xuanjing Huang. A token-level analysis of sampled-token reverse-kl on-policy distillation, 2026. URL https://arxiv.org/abs/2608.25643.

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.

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*, pp. 1279–1297, 2025.

Jie Sun, Mao Zheng, Mingyang Song, Zeyuan Liu, Gengsheng Li, Houcheng Jiang, Yilin Cheng, Bichuan Feng, Yuchen Cai, Junfeng Fang, and Xiang Wang. Distill What You Trust: Reliability-Aware Multi-Teacher On-Policy Distillation. arXiv preprint arXiv:2609.23697, 2026a.

Yang Sun, Lichao Ma, Houyuan Qin, Yuxin Liu, Hanyang Lu, Yao Zhu, Pinlong Cai, and Guohang Yan. Reopd: Reliability-adaptive reward extrapolation for on-policy distillation. *arXiv preprint arXiv:2608.11698*, 2026b.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Xiaomi LLM-Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, et al. MiMo-V2-Flash Technical Report. arXiv preprint arXiv:2601.02780, 2026.

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the Position Bias of On-Policy Distillation. arXiv preprint arXiv:2606.22600, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust Region On-Policy Distillation. arXiv preprint arXiv:2606.01249, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Xun Xu and Zaixi Zhang. Who Teaches Which Token? Verifier-Gated Multi-Expert On-Policy Distillation for Scientific Reasoning. arXiv preprint arXiv:2609.15404, 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.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

A Detailed Related Works

On-policy distillation. Learning from teacher feedback on student-generated sequences addresses the mismatch between training trajectories and the contexts encountered by the student at inference. GKD studies this principle with a family of divergence objectives (Agarwal et al., 2024), while MiniLLM develops reverse-KL optimization for language-model distillation (Gu et al., 2024). Subsequent work demonstrates its effectiveness for reasoning post-training (Yang et al., 2025). Multi-teacher OPD also consolidates specialist capabilities in MiMo-V2-Flash, DeepSeek-V4, and Kimi K3 (Xiaomi LLM-Core Team et al., 2026; Xu et al., 2026; Team et al., 2026). Beyond direct teacher matching, G-OPD and ExOPD introduce a reference policy and reward extrapolation (Yang et al., 2026), and OPD^2 uses the difference between a reasoning-tuned teacher and its base model as the distillation signal (Heo et al., 2026). Analyses further identify teacher–student compatibility as a factor in successful transfer (Li et al., 2026).

Trust and weighting in OPD. Several methods already adapt the strength or location of teacher supervision. REOPOLD stabilizes learning through reward clipping and entropy-based token sampling (Ko et al., 2026). TrOPD identifies trustworthy regions through teacher–student agreement and treats outlier tokens with alternative objectives (Xing et al., 2026). IW-OPD weights tokens using accumulated distribution discrepancy to address deteriorating supervision along long student trajectories (Xie et al., 2026). REOPD combines token-level compatibility weights with a batch-level budget to adapt extrapolation beyond the teacher (Sun et al., 2026b). For multiple teachers, TrustMOPD allocates token-level supervision using each specialist's displacement from a shared pre-RL reference (Sun et al., 2026a). Outcome feedback also supports selective distillation: OPDVR gates sampled-token signals according to response correctness (Lin et al., 2026), while VG-OPD verifies an expert's advantage on individual answer criteria and localizes its supervision through token disagreement (Xu & Zhang, 2026). Closely related to our motivation, Unmasking OPD diagnoses teacher signals through alignment with an estimated success-improving gradient in token-logit space (Armandpour et al., 2026). Dr. OPD uses the sensitivity of the updated student's reward to each supervision weight as an optimization criterion. This yields parameter-space credits for teacher corrections and a clipped weighting rule derived from a reward lower bound relative to vanilla OPD.

Bilevel optimization, reweighting, and influence. Selecting supervision through the performance of the learner connects to bilevel optimization and differentiation through learning dynamics (Franceschi et al., 2018; Shaban et al., 2019). A common strategy in this literature is to evaluate the outer objective through one or a few gradient updates of the inner learner, as in MAML (Finn et al., 2017). This idea is also used for data reweighting, where Ren et al. (2018) selects example weights through a one-step learner update evaluated on clean validation data, leading to a gradient-alignment criterion. Influence-function methods study how upweighting training examples affects the learned model, typically through an inverse-Hessian correction (Koh & Liang, 2017), while TracIn uses gradient inner products along optimization trajectories (Pruthi et al., 2020). Our method brings these ideas to token-level teacher supervision: we measure the reward sensitivity of a one-step OPD update, use outcome reward as the outer objective, and derive finite weight updates from a lower bound on reward improvement.

B Hyperparameters and Experimental Setup

B.1 Shared training configuration

All methods are implemented on top of `verl` (Sheng et al., 2025) and share the same trainer, rollout engine, and outcome verifier; they differ only in how the per-token learning signal is constructed. Each run uses a single node with 8×H100 GPUs.

Both RL teachers are public checkpoints that we keep frozen, without any further training: the math-RL Qwen3-4B of Yang et al. (2026) and the GRPO-trained Qwen3-4B-Base of Li et al. (2026). Within each student–teacher setting, all methods use the same frozen teacher, so their comparison isolates the choice of training objective.

Table 4 lists the hyperparameters shared by all methods and both tasks. The learning rate, batch sizes, rollout settings, and sequence limits are identical for every method. Table 5 lists the method-specific

settings. Dr. OPD reweights all valid response tokens, using a separate credit RMS for each prompt and the bounds $w_{\min} = 0.001$ and $w_{\max} = 3$; Appendix C gives the full implementation.

Table 4: Hyperparameters shared by all methods and both tasks.

Hyperparameter	Value
Learning rate	1×10^{-5}
LR schedule	constant
Gradient clipping	1.0
Batch / Mini-batch size	128
Rollouts per prompt K	8
Rollouts per step	1024
Max prompt length	1024 tokens
Max response length	12288 tokens
Temperature	1.0

Table 5: Method-specific hyperparameters. All other settings follow Table 4.

Method	Hyperparameters
OPD	None
ExOPD	$\lambda = 1.25$; the initial student as reference policy
OPD+GRPO	Equal weighting (1:1) of the distillation and outcome terms
GRPD	None (default settings of the released method)
Dr. OPD	$\lambda = 0.4$; $w_{\min} = 0.001$; $w_{\max} = 3$

B.2 Training Data

For mathematical reasoning, we train on DeepMath-103K (He et al., 2026). The dataset provides difficulty labels. We train the Qwen3-1.7B-Base student on the $[3, 4)$ band and every other student on the $[6, 10]$ band. For code generation, we train on Eurus-RL-Code (Cui et al., 2025), again with the Qwen3-1.7B-Base student as the exception: it is trained on an easier corpus drawn from APPS (introductory) (Hendrycks et al., 2021) and TACO (EASY) (Li et al., 2023). These easier subsets increase the frequency of groups containing both correct and incorrect responses for the Base student. Such groups provide nonzero group-relative advantages; groups with identical rewards have zero advantages.

B.3 Prompt Templates and Stopping Rules

Chat template For all students except Qwen3-1.7B-Base, we use the model's own chat template with thinking disabled (`enable_thinking=False`). The problem statement forms a single user message, followed by the suffix supplied with the training set. The math template is shown below:

```
[title={Zero-shot template, Qwen3-1.7B student (training and evaluation)}]
{question}
Please reason step by step, and put your final answer within \boxed{}.
```

Few-shot template Qwen3-1.7B-Base has no chat template and does not reliably follow the zero-shot instruction format. For this student, we use the fixed four-shot math prompt below.

```
[title={4-shot template, Qwen3-1.7B-Base student (training and evaluation)}]
Solve the following math problems. Reason step by step, and put your final
answer within \boxed{}.

Problem:
What is $1+2+3+\cdots+10$?

Solution:
The sum of the first $n$ positive integers is $\frac{n(n+1)}{2}$. For $n=10$
this is $\frac{10\cdot 11}{2}=55$. The final answer is $\boxed{55}$.

Problem:
Solve for $x$: $2x+3=11$.
```

```
Solution:
Subtracting $3$ from both sides gives $2x=8$. Dividing both sides by $2$ gives
$x=4$. The final answer is $\boxed{4}$.

Problem:
A rectangle has length $8$ and width $5$. What is its area?

Solution:
The area of a rectangle is length times width, so it is $8\times 5=40$. The
final answer is $\boxed{40}$.

Problem:
A fair coin is flipped twice. What is the probability that both flips are heads?

Solution:
The flips are independent, each heads with probability $\frac{1}{2}$. So both
are heads with probability $\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{4}$. The final
answer is $\boxed{\frac{1}{4}}$.

Problem:
{question}

Solution:
```

Stopping rule. With the few-shot prompt, the student may continue by generating additional problems. We therefore truncate each response at the first completed `\boxed{...}` or at the next-problem delimiter `"\nProblem:"`, whichever comes first; everywhere else generation ends at the end-of-sequence token or at the 12,288-token limit. Truncation is applied to the sampled token IDs after generation. We use the same stopping rule for every method during both training and evaluation.

C Implementation of Dr. OPD

For simplicity, Algorithm 1 uses the raw GRPO estimate of the reward gradient $\widehat{r}$ to compute the token credits c_t. In our implementation, we additionally smooth $\widehat{r}$ and apply the coordinate-wise scaling induced by the actor optimizer, matching the transformation applied to the reward gradient in standard GRPO policy optimization.

C.1 Group-relative reward signal

Index training rounds by k, prompts by b, and the K responses to a prompt by i. We suppress k on rollout-level quantities when the round is clear. Let $r_{bi} = V(x_b, y_{bi})$, and let T_{bi} be the number of valid response tokens. The group-relative advantages are

$$\overline{r}_b = \frac{1}{K}\sum_{i=1}^{K} r_{bi}, \qquad s_b = \left[\frac{1}{K-1}\sum_{i=1}^{K}(r_{bi}-\overline{r}_b)^2\right]^{1/2}, \qquad A_{bi} = \frac{r_{bi}-\overline{r}_b}{s_b+\epsilon_A}. \tag{C.1}$$

Here $K = 8$ and $\epsilon_A > 0$ stabilizes the denominator. If all rewards in a group agree, its advantages are zero. With advantages and response lengths held fixed, the length-normalized reward direction is

$$G_k = \sum_b \frac{1}{K}\sum_{i=1}^{K}\frac{A_{bi}}{T_{bi}}\sum_{t=1}^{T_{bi}} \nabla_\theta \log p_\theta(y_{bi,t} \mid s_{bi,t})\big|_{\theta=\bar{\theta}_k}, \tag{C.2}$$

up to the common distributed averaging factor. It is computed by differentiating a negative log-probability surrogate with detached coefficients $A_{bi}/(KT_{bi})$ and reversing the gradient's sign. The factor $1/T_{bi}$ normalizes each response's contribution; a response with no valid tokens contributes zero.

C.2 Optimizer-scaled reward direction

We maintain a reward first moment, separate from the actor optimizer's first moment:

$$m_k^R = \beta_1 m_{k-1}^R + (1-\beta_1)G_k, \qquad m_0^R = 0. \tag{C.3}$$

The direction used in round k is constructed from the state available before the current actor update. For $k > 1$, let $\widehat{m}^R_{k-1} = m^R_{k-1}/(1-\beta_1^{k-1})$. Let v^{actor} be the actor AdamW optimizer's stored second moment, j its number of completed optimizer steps, and $\widehat{v}^{\text{actor}} = v^{\text{actor}}/(1-\beta_2^j)$. The credit direction is

$$u_k = \frac{\widehat{m}^R_{k-1}}{\sqrt{\widehat{v}^{\text{actor}}} + \epsilon_{\text{Adam}}}, \tag{C.4}$$

where the operations are elementwise. Thus the numerator uses reward momentum, while the denominator reuses the actor's second moment. The current G_k updates the reward momentum for subsequent rounds. If the required optimizer or momentum state is unavailable, all weights are set to one. The decoupled weight-decay term is excluded from u_k.

C.3 Token credits, RMS scale, and weight clipping

For each sampled token, define the teacher log-ratio and credit by

$$d_{bi,t} = \log \frac{p^*(y_{bi,t} \mid s_{bi,t})}{p_{\bar{\theta}_k}(y_{bi,t} \mid s_{bi,t})},$$
$$\widehat{c}_{bi,t} = d_{bi,t}\, \frac{\mathrm{d}}{\mathrm{d}\alpha} \log p_{\bar{\theta}_k + \alpha u_k}(y_{bi,t} \mid s_{bi,t})\Big|_{\alpha=0} = g_{bi,t}^\top u_k. \tag{C.5}$$

A forward-mode JVP computes these directional derivatives without materializing a parameter gradient for each token. The student replica used for the JVP is synchronized with the actor, and dropout is disabled. The derivative is evaluated by automatic differentiation, without a finite-difference perturbation.

Let $\mathcal{I}_{kb}$ contain all valid response-token positions across the K rollouts for prompt x_b in round k:

$$\mathcal{I}_{kb} = \{(i,t) : 1 \le i \le K,\ 1 \le t \le T_{bi}\}, \qquad N_{kb} = |\mathcal{I}_{kb}| = \sum_{i=1}^{K} T_{bi}. \tag{C.6}$$

The explicit RMS scale used in (15) is

$$\boxed{\sigma_{kb} = \left(\frac{1}{N_{kb}} \sum_{i=1}^{K} \sum_{t=1}^{T_{bi}} \widehat{c}^{\,2}_{bi,t} \right)^{1/2}} \qquad (N_{kb} > 0). \tag{C.7}$$

This is a token-weighted RMS across the responses to one prompt, without subtracting the mean credit. Prompt and padding tokens are excluded; valid response tokens with zero credit are included. No probability-based token selection is applied. For each $(i,t) \in \mathcal{I}_{kb}$, we set

$$\widehat{w}_{bi,t} = \mathrm{clip}\left(1 + \lambda \frac{\widehat{c}_{bi,t}}{\max(\sigma_{kb}, \epsilon_\sigma)},\ 0.001,\ 3\right), \qquad \lambda = 0.4 \tag{C.8}$$

in the main experiments; the ablation varies λ. Here $\epsilon_\sigma > 0$ is the stability constant denoted by ϵ in (15). If all credits for a prompt are zero, its valid tokens receive weight one. If $N_{kb} = 0$, we set $\sigma_{kb} = 0$ and that prompt contributes no response tokens to the actor loss. The positive lower bound preserves strictly positive weights, as required by the weighted-distillation formulation.

C.4 Actor update

The detached products $\widehat{w}_{bi,t} d_{bi,t}$ are passed as token advantages to the actor loss used by vanilla OPD. The actor uses a PPO-style probability-ratio objective, including ratio clipping and dual clipping for negative advantages, token-mean aggregation, and AdamW. At the reference policy, where the probability ratio equals one, the ascent gradient is the token average of $\widehat{w}_{bi,t} g_{bi,t}$. Gradients used to construct the reward direction are cleared before the actor update. The weights and teacher signals remain detached during that update.

D Proofs and Theoretical Details

This section collects the assumptions, derivations, and proofs used in Sections 2–3.2. We first specify the reference distribution for one optimization round, then establish the weighted-distillation properties and the reward-improvement guarantee.

D.1 NOTATION AND REGULARITY CONDITIONS

Fix the current student $\bar{\theta}$. We assume a finite vocabulary and strictly positive student and teacher probabilities. A feasible weight function is a measurable function in

$$\mathcal{W} = \{w : w_{\min} \le w(s, v) \le w_{\max}\}, \qquad 0 < w_{\min} < 1 < w_{\max} < \infty.$$

The reference context distribution and w are held fixed when we differentiate the distillation objective. In particular, the loss for this round is

$$D_w^{\bar{\theta}}(\theta) := \mathbb{E}_{x,\, y\sim p_{\bar{\theta}}(\cdot|x)} \left[\sum_t \mathrm{KL}_{w(s_t,\cdot)}\big(p_\theta(\cdot \mid s_t) \| p^*(\cdot \mid s_t)\big) \right]. \tag{D.1}$$

We suppress the superscript $\bar{\theta}$ below and in the main text. The reference distribution is refreshed between rounds.

Define the reference measure on context–token pairs by

$$\int f(s, v)\, \mathrm{d}\mu(s, v) := \mathbb{E}_{x,\, y\sim p_{\bar{\theta}}(\cdot|x)} \left[\sum_{t=1}^{T} f(s_t, y_t) \right] \tag{D.2}$$

for nonnegative measurable f. Its total mass is $\mathbb{E}[T]$; thus μ is an unnormalized token-occupancy measure. It includes both context visitation and the conditional probability of the sampled token. We use $\langle f, h\rangle_\mu = \int fh\, \mathrm{d}\mu$ and $\|f\|_\mu^2 = \int f^2\, \mathrm{d}\mu$ for scalar functions, and write $g(s_t, y_t) = g_t$ for the teacher correction in (6).

Assumption 6 (Local regularity and integrability)**.** Let μ denote the fixed reference distribution over sampled trajectories, and let $\mathcal{W} = \{w : w_{\min} \le w \le w_{\max}\}$. Assume the following conditions hold.

(i) **Integrability.** The expected response length is finite,

$$\mathbb{E}_\mu[T] < \infty.$$

Moreover, the weighted OPD objective is finite in a neighborhood of $\bar{\theta}$, and differentiation can be interchanged with expectation. In particular,

$$\nabla_\theta D_w(\theta) = \mathbb{E}_\mu[\nabla_\theta \ell_w(\theta; Z)], \qquad w \in \mathcal{W},$$

for all θ in this neighborhood.

(ii) **Finite second moment.** The token-level teacher correction satisfies

$$M := \mathbb{E}_\mu \left[\sum_{t=1}^{T} \|g_t\|^2 \right] < \infty. \tag{D.3}$$

(iii) **Local smoothness of the reward.** There exists an open convex set U such that

$$\{\bar{\theta}\} \cup \{\theta^+(w) : w \in \mathcal{W}\} \subset U,$$

and R is continuously differentiable on U with L_R-Lipschitz gradient:

$$\|\nabla R(\theta) - \nabla R(\theta')\| \le L_R \|\theta - \theta'\|, \qquad \forall\, \theta, \theta' \in U.$$

All parameter norms are Euclidean.

Cauchy–Schwarz gives $\int \|g\|\, \mathrm{d}\mu \le \sqrt{\mathbb{E}[T]M}$, so the response in (7) satisfies

$$\|\theta^+(w) - \bar{\theta}\| \le \eta w_{\max} \sqrt{\mathbb{E}[T]M}. \tag{D.4}$$

Thus a sufficiently small student step size keeps all responses in a fixed neighborhood where the required smoothness holds. The condition $\lambda\eta L_R M \le 1$ is equivalent to $\lambda \le 1/(\eta L_R M)$ when $L_R M > 0$; when $L_R M = 0$, it imposes no further restriction on positive λ.

Unlike the fixed-reference distillation loss, the reward $R(\theta)$ is evaluated under the student's own response distribution. Under the corresponding differentiation and summability conditions,

$$\nabla R(\theta) = \mathbb{E}_{x,\, y\sim p_\theta} \left[V(x, y) \sum_t \nabla_\theta \log p_\theta(y_t \mid s_t) \right]. \tag{D.5}$$

This follows by differentiating $p_\theta(y \mid x) = \prod_t p_\theta(y_t \mid s_t)$, with termination included in the response convention.

D.2 Properties of weighted OPD

D.2.1 Proof of Theorem 3

Proof. For positive a, b, define

$$d(a,b) = a\log(a/b) - a + b = b\phi(a/b), \qquad \phi(u) = u\log u - u + 1. \tag{D.6}$$

Since $\phi'(u) = \log u$ and $\phi''(u) = 1/u > 0$, its unique minimum is $\phi(1) = 0$. Hence $d(a,b) \geq 0$, with equality if and only if $a = b$. At each context s,

$$\mathrm{KL}_{w(s,\cdot)}(p_\theta \| p^*) = \sum_v w(s,v) d\big(p_\theta(v \mid s), p^*(v \mid s)\big) \geq 0.$$

Because every weight is strictly positive, equality holds precisely when the student and teacher probabilities agree for every token. Taking the reference expectation proves $D_w(\theta) \geq 0$. The expected loss is zero if and only if the distributions agree at almost every context under the reference context measure. In particular, the claim holds at every context of positive reference mass; the objective places no restriction on contexts of zero measure. Finally, when $w \equiv 1$, the correction terms sum to zero by normalization, leaving the ordinary reverse KL and the vanilla OPD loss. □

The linear correction is needed for token-dependent weights. For example, if $p = (1/4, 3/4)$, $p^* = (1/2, 1/2)$, and $w = (4, 1)$, then $\sum_v w_v p_v \log(p_v/p_v^*) = -\log 2 + (3/4)\log(3/2) < 0$. The corrected divergence remains nonnegative term by term. When $w(s,v) = a(s)$ is independent of the token, the correction cancels and we recover $a(s)\,\mathrm{KL}(p_\theta(\cdot \mid s)\|p^*(\cdot \mid s))$.

D.2.2 Derivation of the sampled-token gradient

Derivation of (6). At a fixed context, let $p_v = p_\theta(v \mid s)$ and $q_v = p^*(v \mid s)$. Differentiating with w fixed gives

$$\begin{aligned}\nabla_\theta \,\mathrm{KL}_{w(s,\cdot)}(p_\theta \| p^*) &= \sum_v w(s,v)\big[\log(p_v/q_v) + 1 - 1\big]\nabla_\theta p_v \\ &= \sum_v p_v w(s,v)\log(p_v/q_v)\nabla_\theta \log p_v.\end{aligned} \tag{D.7}$$

The derivative of $-p_v$ cancels the additional 1 separately for each token. At $\theta = \bar\theta$, the last sum is a conditional expectation under the reference student's next-token distribution. Taking the reference expectation over contexts therefore yields

$$-\nabla D_w(\bar\theta) = \mathbb{E}\left[\sum_t w_t \log \frac{p_t^*}{p_t(\bar\theta)}\, \nabla_\theta \log p_\theta(y_t \mid s_t)|_{\theta=\bar\theta}\right] = \mathbb{E}\left[\sum_t w_t g_t\right]. \tag{D.8}$$

The interchange is justified by Assumption 6. □

The identity allows token-dependent w because the full conditional expectation is retained. Differentiating the reference sampling law, or recomputing w as a function of the optimization variable, would introduce additional terms.

An explicit unbiased sampled loss is

$$\widehat{D}_w(\theta; x, y) := \sum_t \frac{w_t}{p_t(\bar\theta)}\left[p_t(\theta)\log\frac{p_t(\theta)}{p_t^*} - p_t(\theta) + p_t^*\right], \tag{D.9}$$

where (x, y) is sampled from the reference policy and the denominator and weights are fixed. Conditional expectation over y_t recovers the vocabulary sum in (D.1). At $\bar\theta$, the factor $p_t(\theta)/p_t(\bar\theta)$ in its gradient is one, so $-\nabla\widehat{D}_w(\bar\theta; x, y) = \sum_t w_t g_t$. This explains why (6) contains no explicit importance denominator. At another parameter value, the probability ratio must be retained when using the same reference rollouts.

For the interpretation of g_t, let $z_t = \nabla_\theta \log p_\theta(y_t \mid s_t)|_{\bar\theta}$. Then

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\log p_{\bar\theta + \epsilon g_t}(y_t \mid s_t)\Big|_{\epsilon=0} = \log\frac{p_t^*}{p_t(\bar\theta)}\,\|z_t\|^2. \tag{D.10}$$

Thus the infinitesimal correction increases or decreases the sampled token's log-probability according to the teacher–student log-ratio, unless $z_t = 0$. This is a local statement about one token; descent of the full distillation loss follows from the aggregate gradient.

D.2.3 DESCENT OF THE LOWER-LEVEL OBJECTIVE

If, in addition, D_w has an L_D-Lipschitz gradient on U, uniformly over feasible w, its one-step response decreases the fixed-reference loss whenever $\eta L_D \le 1$.

Proof. The smoothness inequality and $\theta^+(w) = \bar{\theta} - \eta \nabla D_w(\bar{\theta})$ give

$$\begin{aligned} D_w(\theta^+(w)) &\le D_w(\bar{\theta}) - \eta \left(1 - \frac{\eta L_D}{2}\right) \|\nabla D_w(\bar{\theta})\|^2 \\ &\le D_w(\bar{\theta}) - \frac{\eta}{2} \|\nabla D_w(\bar{\theta})\|^2. \end{aligned} \tag{D.11}$$

□

This additional loss-smoothness assumption is only needed for the descent statement, not for the reward bounds below.

D.3 BILEVEL FORMULATION WITH MULTIPLE INNER SOLUTIONS

Let $\mathcal{P}$ denote the student parameter space. For a fixed reference distribution, define

$$\Theta(w) := \arg\min_{\theta \in \mathcal{P}} D_w(\theta). \tag{D.12}$$

The optimistic convention used in (5) selects the highest-reward member of this set:

$$\max_{w \in \mathcal{W}} \max_{\theta \in \Theta(w)} R(\theta). \tag{D.13}$$

We assume the inner solution sets are nonempty and that the displayed maxima are attained. If a reward maximum is not attained, the corresponding optimal value is written as a supremum; an empty inner solution set instead requires a different formulation.

If exact teacher matching is attainable, all feasible positive weightings share the same zero-loss minimizers by Theorem 3. Weighting can change the best attainable compromise when exact matching is infeasible and the learning trajectory before convergence. Our local analysis replaces $\Theta(w)$ by the specified response $\theta^+(w)$. It therefore does not require an attained or unique global inner optimum and does not establish convergence to (D.13).

D.4 CREDIT AS SENSITIVITY TO THE WEIGHT FUNCTION

Let $F(w) = R(\theta^+(w))$ and define $c(s,v) = g(s,v)^\top \nabla R(\theta^+_{\mathrm{vani}})$. The next calculation makes the derivative notation in (8) precise.

Derivation of the credit. For a bounded measurable perturbation h, the function $1 + \epsilon h$ is feasible for sufficiently small $|\epsilon|$. Linearity of the response gives

$$\theta^+(1 + \epsilon h) = \theta^+_{\mathrm{vani}} + \eta\epsilon \int h g \, \mathrm{d}\mu.$$

Applying the chain rule,

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon} F(1 + \epsilon h)\Big|_{\epsilon=0} = \eta \int h \, g^\top \nabla R(\theta^+_{\mathrm{vani}}) \, \mathrm{d}\mu = \eta \langle h, c \rangle_\mu. \tag{D.14}$$

Moreover, $\|c\|_\mu^2 \le M \|\nabla R(\theta^+_{\mathrm{vani}})\|^2 < \infty$. Thus the $L^2(\mu)$ gradient of F at $w = 1$ is ηc. □

Accordingly, $\partial R(\theta^+(w))/\partial w_t$ in the main text denotes sensitivity per unit reference measure, evaluated at (s_t, y_t). For an atom (s, v), the ordinary coordinate derivative instead includes its mass:

$$\left.\frac{\partial F(w)}{\partial w(s,v)}\right|_{w=1} = \eta\mu(\{(s,v)\})c(s,v). \tag{D.15}$$

A derivative with respect to a weight in a finite sample average likewise includes that average's coefficient. Changing weights only on a set of zero reference measure has no effect on F.

D.5 Proof of Proposition 4

Proof. The argument has two steps: bound the change in the student response, then apply reward smoothness.

Step 1: Control the response change. Let $h = w - 1$ and $\Delta(w) = \theta^+(w) - \theta^+_{\text{vani}} = \eta \int hg \, \mathrm{d}\mu$. By Cauchy–Schwarz,

$$\|\Delta(w)\|^2 \le \eta^2 \left(\int h^2 \, \mathrm{d}\mu\right)\left(\int \|g\|^2 \, \mathrm{d}\mu\right) = \eta^2 M \|h\|^2_\mu. \tag{D.16}$$

All terms are finite because h is bounded, μ has finite mass, and $M < \infty$. The trajectory-length contribution is already included in μ and M.

Step 2: Lower-bound the reward change. Both responses and the segment joining them lie in U. Therefore,

$$\begin{aligned} R(\theta^+(w)) - R(\theta^+_{\text{vani}}) &\ge \nabla R(\theta^+_{\text{vani}})^\top \Delta(w) - \frac{L_R}{2}\|\Delta(w)\|^2 \\ &\ge \eta\langle h, c\rangle_\mu - \frac{L_R \eta^2 M}{2}\|h\|^2_\mu. \end{aligned} \tag{D.17}$$

Using $\lambda\eta L_R M \le 1$ and the definition of μ gives

$$R(\theta^+(w)) - R(\theta^+_{\text{vani}}) \ge \eta \int \left(hc - \frac{h^2}{2\lambda}\right) \mathrm{d}\mu,$$

which is (9). The bound is exact at $w = 1$, and its first variation there agrees with (D.14). □

D.6 Clipped weights and proof of Theorem 5

Proof. We first optimize the lower bound pointwise, then quantify its gain.

Step 1: Maximize the surrogate. Completing the square yields

$$(w-1)c - \frac{(w-1)^2}{2\lambda} = \frac{\lambda c^2}{2} - \frac{(w - 1 - \lambda c)^2}{2\lambda}.$$

Its unique maximizer on $[w_{\min}, w_{\max}]$ is $w^+ = \text{clip}(1 + \lambda c, w_{\min}, w_{\max})$. This function is measurable and feasible. Since the surrogate is integrable, pointwise maximization also maximizes its expectation, uniquely up to sets of zero μ-measure.

Step 2: Bound the improvement. Write $h^+ = w^+ - 1$. The first-order optimality condition, evaluated against the feasible choice $w = 1$, gives

$$\left(c - \frac{h^+}{\lambda}\right)(1 - w^+) \le 0, \qquad h^+ c \ge \frac{(h^+)^2}{\lambda}. \tag{D.18}$$

Integrating and applying Proposition 4,

$$R(\theta^+(w^+)) - R(\theta^+_{\text{vani}}) \ge \frac{\eta}{2\lambda}\|w^+ - 1\|^2_\mu \ge 0.$$

This is (11). The gain is strictly positive when $w^+ \ne 1$ on a set of positive μ-measure. Because 1 lies strictly inside the feasible interval, this is equivalent to nonzero credit on a set of positive measure. If the credit vanishes almost everywhere, the update recovers vanilla OPD. □

E SUPPLEMENTARY EXPERIMENTAL RESULTS

This section provides additional experimental results and training dynamics. All quantities below are measured on the student's own training rollouts.

Figure 3 shows the training-reward dynamics for the experiments in Table 1.

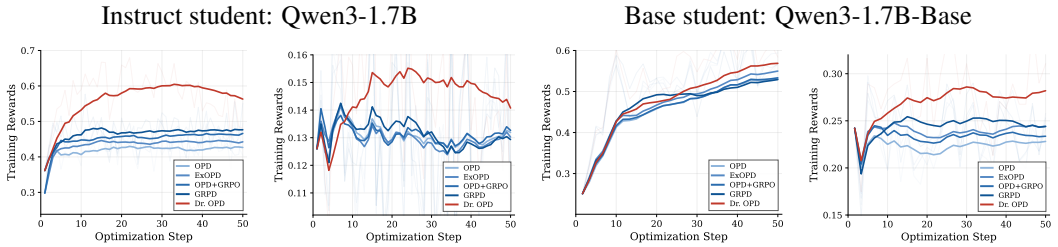
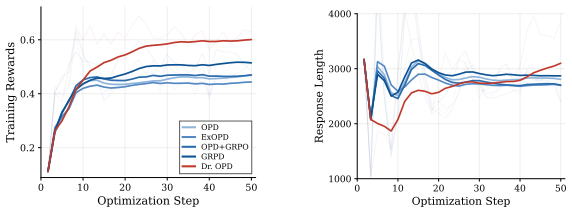


Figure 3: Dynamics of training rewards across students and tasks, computed by verifier accuracy of the student's own rollouts during training. Lines are exponential moving averages ($\alpha = 0.95$).

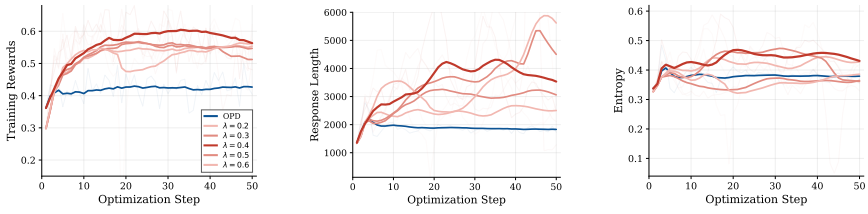
Figure 4 shows the training dynamics for the Qwen3-4B-Base student distilled from an RL-trained teacher of the same model size. The training reward is the fraction of rollouts receiving a positive outcome reward, and the response length is the average number of generated tokens per rollout. Although the student and teacher differ only in post-training, Dr. OPD maintains a clear advantage in training reward.



(a) Training reward (b) Response length

Figure 4: Training dynamics for the Qwen3-4B-Base student distilled from the Qwen3-4B-Base-RL teacher on math. Thick lines show exponential moving averages ($\alpha = 0.95$).

Figure 5 further examines the effect of the weighting strength λ, with $\lambda = 0$ corresponding to vanilla OPD. All tested positive values of λ achieve higher training rewards than vanilla OPD and exhibit broadly similar reward curves. In contrast, response length can vary substantially for larger values of λ, with differences of thousands of tokens late in training.



(a) Training reward (b) Response length (c) Entropy

Figure 5: Training dynamics under different values of λ. (a) Training reward, (b) response length, and (c) entropy over training steps for vanilla OPD ($\lambda = 0$) and Dr. OPD with $\lambda \in \{0.2, 0.3, 0.4, 0.5, 0.6\}$, using the Qwen3-1.7B student and Qwen3-4B teacher. Thick lines show exponential moving averages ($\alpha = 0.95$).