KL for a KL: On-Policy Distillation with Control Variate Baseline

Minjae Oh [1] **Sangjun Song** [1] **Gyubin Choi** [1] **Yunho Choi** [1] **Yohan Jo** [1]

Abstract

On-Policy Distillation (OPD) has emerged as a dominant post-training paradigm for large language models, especially for reasoning domains like mathematics. However, OPD remains unstable in practice due to the high gradient variance of its single-sample Monte Carlo estimator, and recipes for stable training are still immature. We propose *v*OPD (**O**n-**P**olicy **D**istillation with a control ***v***ariate baseline), which casts OPD as policy-gradient RL and stabilizes it by introducing a control variate baseline—canonically a value function—from the RL literature. We show that the OPD value function admits a closed form as the per-token negative reverse KL divergence between the student and the teacher, available directly from the already-computed forward pass with no additional critic or inference. Existing stabilization methods either compute the full token-level reverse KL over the entire vocabulary, adding significant overhead, or restrict it to a top-k support, biasing the objective. *v*OPD instead preserves the lightweight single-sample estimator, subtracting the value function as a detached baseline to keep the gradient unbiased while reducing variance. Furthermore, we show that a top-k approximation of the baseline further lowers cost without compromising performance. Across mathematical and scientific benchmarks, *v*OPD consistently outperforms OPD and matches the most expensive full-vocabulary baseline, offering an efficient stabilization of On-Policy Distillation through principled RL variance reduction.

1. Introduction

Large Language Models have made remarkable advances in mathematical reasoning, accompanied by improvements in post-training recipes (Jaech et al., 2024; Yang et al., 2025; Team et al., 2025). A key factor has been Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert et al., 2024; Guo et al., 2025), which trains LLMs directly against easily verifiable rewards—answer correctness, code execution—sidestepping the noise and reward hacking introduced by learned reward models. RLVR has been successful thanks to its simple recipe, but this simplicity comes at a cost: LLMs generate thousands of intermediate tokens during reasoning before receiving a single scalar reward for the final answer. RLVR methods must perform credit assignment over long chains of thought from a single sparse scalar signal. This sparse supervision demands large rollouts and prolonged training, making training progress painfully slow.

On-Policy Distillation (OPD) (Gu et al., 2024; Agarwal et al., 2024) has emerged as an attractive alternative to RLVR when a strong teacher is available. Rather than relying on a sparse terminal reward, OPD minimizes the reverse KL divergence between the student and the teacher via dense, token-level signals, enabling faster training (Lightman et al., 2024). Because it is on-policy and reward-driven, OPD can naturally be implemented using standard RL pipelines with a single-sample Monte Carlo estimator (Lu & Lab, 2025), and empirically matches RLVR accuracy with a fraction of the compute (Patiño et al., 2025). Its effectiveness has been demonstrated in industrial-level post-training such as Qwen3, GLM-5, Nemotron-Cascade2, and DeepSeek-V4 (Yang et al., 2025; Zeng et al., 2026; Yang et al., 2026; DeepSeek-AI, 2026). Despite this success, OPD's optimization recipe remains underdeveloped: training is unstable in practice, and stabilization techniques are still immature relative to the recipes that drive successful RLVR training (Yu et al., 2025; Khatri et al., 2025; Zheng et al., 2025; Chen et al., 2025; Liu et al., 2026). The most widely adopted fix replaces the single-sample estimator with a full-vocabulary token-level KL, incurring additional compute overhead (Agarwal et al., 2024); a lighter-weight variant restricts the KL to a top-k support, which biases the gradient away from the true objective and still adds compute, yet yields only marginal gains (Li et al., 2026). In contrast, we turn to the RL interpretation of OPD and propose a principled, low-compute method that controls variance while preserving the efficient single-sample Monte Carlo estimator.

[1]Graduate School of Data Science, Seoul National University, Seoul, South Korea. Correspondence to: Yohan Jo <yohan.jo@snu.ac.kr>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

We propose ***v*OPD** (**O**n-**P**olicy **D**istillation with a control ***v***ariate baseline), which leverages a standard tool from policy-gradient RL to reduce gradient variance: subtracting a control variate baseline (Williams, 1992; Sutton et al., 1999; Greensmith et al., 2004). Baseline subtraction for variance reduction underlies actor-critic methods such as PPO, and more recently GRPO and RLOO (Schulman et al., 2017; Mnih et al., 2016; Ahmadian et al., 2024; Shao et al., 2024) (see § 3.1). *v*OPD reduces variance without biasing the gradient in expectation. The standard choice of baseline is the value function, and we show that for OPD this quantity admits a computable closed form: the per-token negative reverse KL between the student and the teacher at each token. The baseline is therefore available from the same forward pass that already computes the OPD objective—without an additional critic model, extra rollouts, or additional backward passes. We show that the baseline can be approximated using only the top-k student tokens at a lower cost; crucially, because this approximation does not depend on the sampled token, it preserves the unbiasedness of the gradient regardless of k (see § 3.2). Furthermore, we find empirically that the choice of k has little effect on performance (see § 4.3).

We evaluate *v*OPD on four models from the Qwen3 (Yang et al., 2025) and Olmo-3 (Olmo et al., 2025) families across six reasoning benchmarks spanning mathematics and science—MATH500 (Hendrycks et al., 2021; Lightman et al., 2024), Minerva Math (Lewkowycz et al., 2022), AMC23 (Mathematical Association of America, 2026b), AIME24/25 (Mathematical Association of America, 2026a), SciKnowEval (Feng et al., 2024), and GPQA-Diamond (Rein et al., 2024)—demonstrating consistent improvements over baseline methods. *v*OPD delivers an absolute average accuracy gain of up to +3% on average over base OPD, with improvements of up to +6.2% on MATH500 (see §§ 4.2 and 4.4). Against the two stabilization variants, *v*OPD substantially outperforms top-k OPD and matches full-vocabulary OPD while reducing wall-clock time up to 57.7%. We further validate the stability of *v*OPD through consistently lower gradient norms, and show that it acts as a regularizer on destabilizing reward tokens (see § 4.3). Overall, *v*OPD bridges RL and knowledge distillation, providing a principled, efficient approach to stable On-Policy Distillation.

2. Preliminaries

2.1. On-Policy Distillation

On-Policy Distillation (OPD) (Agarwal et al., 2024; Gu et al., 2024) trains the student by minimizing the reverse KL divergence between the student (π_θ) and the teacher (π_T):

$$\mathbb{D}_{\mathrm{KL}}(\pi_\theta \,\|\, \pi_T) = \mathbb{E}_{\substack{x\sim\mathcal{D} \\ y\sim\pi_\theta(\cdot|x)}}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_T(y \mid x)}\right]. \quad (1)$$

where x is a prompt drawn from a dataset $\mathcal{D}$ and $y = (y_1, \ldots, y_{|y|})$ is a response of length $|y|$. Importantly, OPD samples from the student during generation to obtain an unbiased estimator of the KL. On-policy learning mitigates exposure bias (Ranzato et al., 2016)—the train-test discrepancy in off-policy training, where the model is trained on static data but conditions on its own outputs at test time. This has enabled effective training on long Chain-of-Thought reasoning tasks such as mathematics (Wei et al., 2022; Yang et al., 2025; Lu & Lab, 2025). In practice, Eq. (1) is commonly optimized via a single-sample Monte Carlo estimate, by maximizing the following token-level objective, where $c_t = (x, y_{<t})$ denotes the context at step t (Lu & Lab, 2025):

$$\mathcal{J}_{\mathrm{OPD}}(\theta) \;=\; \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{|y|} \log \frac{\pi_T(y_t \mid c_t)}{\pi_\theta(y_t \mid c_t)}\right]. \quad (2)$$

Following recent practice (Lu & Lab, 2025; Zeng et al., 2026; Ko et al., 2026), Eq. (2) is optimized as policy-gradient RL (Williams, 1992; Sutton et al., 1999) by defining the per-token reward $r_t(c_t, y_t) = \log \pi_T(y_t \mid c_t) - \log \pi_\theta(y_t \mid c_t)$ as a fixed scalar with no gradient flowing through it. This yields the gradient:

$$\nabla_\theta \mathcal{J}_{\mathrm{OPD}}(\theta) = \mathbb{E}\left[\sum_{t=1}^{|y|} r_t(c_t, y_t) \nabla_\theta \log \pi_\theta(y_t \mid c_t)\right]. \quad (3)$$

We refer to this base formulation as **OPD** throughout the paper. Its backward pass touches only $\nabla_\theta \log \pi_\theta(y_t \mid c_t)$ at the single sampled token, making it the most computationally efficient variant. However, the single-sample Monte Carlo estimator carries high variance, leading to training instability. We next discuss two variants that aim to stabilize OPD, along with the drawbacks of each.

Full-vocabulary OPD (OPD$_{\text{full-V}}$). To mitigate the variance of the single-sample estimator, one variant computes the full per-token KL over the entire vocabulary $\mathcal{V}$ (Agarwal et al., 2024):

$$\begin{aligned}\mathcal{J}_{\mathrm{OPD}_{\text{full-V}}}(\theta) &= -\mathbb{E}\left[\sum_{t=1}^{|y|} \mathbb{D}_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid c_t) \,\|\, \pi_T(\cdot \mid c_t)\big)\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{|y|} \sum_{v\in\mathcal{V}} \pi_\theta(v \mid c_t) r_t(c_t, v)\right].\end{aligned} \quad (4)$$

where $r_t(c_t, v) = \log \pi_T(v \mid c_t) - \log \pi_\theta(v \mid c_t)$ extends the per-token reward to any vocabulary entry. Similar to Eq. (3), Eq. (4) can be optimized by the corresponding

gradient:

$$\nabla_\theta \mathcal{J}_{\text{OPD}_{\text{full-V}}}(\theta) = \mathbb{E}\left[\sum_{t=1}^{|y|}\sum_{v\in\mathcal{V}} \pi_\theta(v \mid c_t) r_t(c_t, v) \nabla_\theta \log \pi_\theta(v \mid c_t)\right]. \tag{5}$$

which is the exact expectation of Eq. (3) under $v \sim \pi_\theta$, and is therefore zero-variance for a given c_t. The cost, however, is substantial as it requires a backward pass against the full vocabulary at every token (e.g., $|\mathcal{V}| \approx 150\text{k}$ for Qwen3 (Yang et al., 2025)).

Top-k OPD (OPD$_{\text{top-}k}$). A lightweight variant of OPD$_{\text{full-V}}$ computes the per-token KL against only the top-k tokens, with $k \ll |\mathcal{V}|$ (Li et al., 2026). We consider the student top-k version, restricting the KL to the support S_t of the student's k most likely tokens:

$$\mathcal{J}_{\text{OPD}_{\text{top-}k}}(\theta) = -\mathbb{E}\left[\sum_{t=1}^{|y|} \mathbb{D}_{\text{KL}}\big(\bar{\pi}_\theta(\cdot \mid c_t) \,\|\, \bar{\pi}_T(\cdot \mid c_t)\big)\right],$$
$$\bar{\pi}(v \mid c_t) = \frac{\pi(v \mid c_t)\mathbf{1}[v \in S_t]}{\sum_{u\in S_t} \pi(u \mid c_t)}. \tag{6}$$

Eq. (6) is optimized by a gradient of the same shape as Eq. (5), but restricted to S_t and acting on the renormalized distributions:

$$\nabla_\theta \mathcal{J}_{\text{OPD}_{\text{top-}k}}(\theta) = \mathbb{E}\left[\sum_{t=1}^{|y|}\sum_{v\in S_t} \bar{\pi}_\theta(v \mid c_t) \log \frac{\bar{\pi}_T(v \mid c_t)}{\bar{\pi}_\theta(v \mid c_t)} \nabla_\theta \log \bar{\pi}_\theta(v \mid c_t)\right]. \tag{7}$$

The backward pass now flows through k tokens per position, rather than the single sampled token in OPD—substantially more lightweight than the full vocabulary, but heavier than base OPD. More importantly, this comes at the cost of *bias*: $\nabla_\theta \mathcal{J}_{\text{OPD}_{\text{top-}k}} \neq \nabla_\theta \mathcal{J}_{\text{OPD}}$, since restricting to S_t omits out-of-support mass. In practice, despite this added compute, OPD$_{\text{top-}k}$ has been reported to yield only marginal gains over base OPD (Li et al., 2026), an observation we confirm in § 4.2.

2.2. Control Variate Baseline in Reinforcement Learning

As OPD in Eq. (3) is a form of RL, we now introduce the standard variance-reduction tool used in policy-gradient RL: subtracting a baseline $b_t(c_t)$ from the per-step reward, yielding the *advantage* $a_t(c_t, y_t) = r_t(c_t, y_t) - b_t(c_t)$ (Williams, 1992; Sutton et al., 1999):

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}\left[\sum_t \big(r_t(c_t, y_t) - b_t(c_t)\big) \nabla_\theta \log \pi_\theta(y_t \mid c_t)\right]. \tag{8}$$

This has two properties that make it successful in modern RL. **(i) Unbiasedness:** for *any* b_t that is independent of the sampled action y_t, $\mathbb{E}[b_t(c_t)\nabla_\theta \log \pi_\theta(y_t \mid c_t)] = 0$, so the expected gradient is unchanged, and the loss remains unbiased (see § A.1). **(ii) Variance reduction:** a well-defined baseline reduces the gradient variance, the canonical choice being the value function $V^{\pi_\theta}(c_t) = \mathbb{E}_{y_t\sim\pi_\theta(\cdot|c_t)}[r_t(c_t, y_t)]$ (see § A.2). Baseline subtraction underlies the success of essentially every modern policy-gradient algorithm, from classical actor-critic methods with a learned value baseline (Mnih et al., 2016; Schulman et al., 2017; 2016) to the group-relative baseline in GRPO (Shao et al., 2024).

3. Control Variate Baseline for OPD

We introduce *v*OPD (**O**n-**P**olicy **D**istillation with a control ***v***ariate baseline), which addresses the high variance of OPD (§ 2.1) by exploiting its RL interpretation and subtracting a control variate baseline (§ 2.2). *v*OPD is an unbiased, lower-variance version of OPD that requires no additional backward passes, making it computationally efficient. We first show that the value function of OPD is available in closed form as the per-step reverse KL, and discuss the loss formulation of *v*OPD (see § 3.1). We then propose an even more computationally efficient version using a top-k KL estimate (see § 3.2), and compare our methods with the various variants of OPD (see § 3.3).

3.1. The Value Function of OPD

As discussed in § 2.2, the standard choice of baseline is the value function. Recall the OPD per-token reward $r_t(c_t, y_t) = \log \pi_T(y_t \mid c_t) - \log \pi_\theta(y_t \mid c_t)$ from Eq. (3). By definition, taking the expectation of r_t under the student distribution (π_θ) gives the per-step value function:

$$\begin{aligned} V^{\pi_\theta}(c_t) &= \mathbb{E}_{y_t\sim\pi_\theta(\cdot|c_t)}\left[r_t(c_t, y_t)\right] \\ &= -\mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot \mid c_t) \,\|\, \pi_T(\cdot \mid c_t)\big). \end{aligned} \tag{9}$$

The value function is exactly the negative per-step reverse KL (Tang & Munos, 2025), computable in closed form using the already-computed student (π_θ) and teacher (π_T) distributions at context c_t without a learned value network or an additional forward pass. Substituting Eq. (9) as the baseline in Eq. (8) gives the *v*OPD gradient estimator with advantage $a_t(c_t, y_t)$:

$$a_t(c_t, y_t) = r_t(c_t, y_t) + \mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot \mid c_t) \,\|\, \pi_T(\cdot \mid c_t)\big),$$
$$\nabla_\theta \mathcal{J}_{v\text{OPD}_{\text{full-V}}}(\theta) = \mathbb{E}\left[\sum_{t=1}^{|y|} a_t(c_t, y_t) \nabla_\theta \log \pi_\theta(y_t \mid c_t)\right]. \tag{10}$$

which is denoted by *v*OPD$_{\text{full-V}}$; the subscript indicates the expectation over the full vocabulary to compute the KL baseline. As discussed in § 2.2, this estimator has the same

expected gradient as OPD: $\mathbb{E}[\nabla \mathcal{J}_{v\text{OPD}_{\text{full-V}}}] = \mathbb{E}[\nabla \mathcal{J}_{\text{OPD}}]$. Importantly, the baseline KL is computed only in the forward pass and does not propagate gradients through the vocabulary, so the backward pass flows only through $\nabla_\theta \log \pi_\theta(y_t \mid c_t)$ at the single sampled token, identical to base OPD.

Variance reduction. We now examine where and why *v*OPD reduces variance, showing it dampens gradients most strongly on the most destabilizing cases. Recent works have identified *high-mismatch tokens*—where the student and the teacher distributions strongly disagree—as the dominant source of OPD's gradient instability: at these tokens, the per-token reward (r_t) takes large negative values, producing heavy-tailed gradients that dominate training (Ko et al., 2026; Li et al., 2026). *v*OPD's baseline directly counteracts this. Since $-V^{\pi_\theta}(c_t) = \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_T)$ becomes a large positive value precisely when the student and the teacher strongly disagree, the *v*OPD advantage ($a_t = r_t + \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_T)$) stays bounded even on such heavy-tailed tokens, acting as a regularizer. This token-level reward damping translates directly into a reduction in gradient variance. We show that the per-token variance reduction of *v*OPD is approximately:

$$\begin{aligned} &\operatorname{tr}(\operatorname{Var}[g_{\text{OPD}}]) - \operatorname{tr}(\operatorname{Var}[g_{v\text{OPD}_{\text{full-V}}}]) \\ &\quad \approx \mathbb{D}_{\text{KL}}^2\big(\pi_\theta(\cdot \mid c_t) \,\|\, \pi_T(\cdot \mid c_t)\big) \\ &\qquad \cdot \mathbb{E}_{\pi_\theta}\left[\|\nabla_\theta \log \pi_\theta(y_t \mid c_t)\|^2\right]. \end{aligned} \tag{11}$$

where g_{OPD} and $g_{v\text{OPD}_{\text{full-V}}}$ are the per-step gradient estimators of OPD (Eq. (3)) and *v*OPD$_{\text{full-V}}$ (Eq. (10)), and $\operatorname{tr}(\cdot)$ denotes the matrix trace. We provide a detailed derivation in § A.3. From Eq. (11), the variance reduction is largest when the squared $\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_T)$ is large at the high-mismatch tokens, which matches our token-level reward damping view. Overall, *v*OPD dampens the noisy negative long-tail gradients destabilizing OPD, which we further validate empirically in § 4.3.

Connection to OPD$_{\text{full-V}}$. A natural question is whether the same baseline could also help OPD$_{\text{full-V}}$. The answer is no, and the reason illuminates the relationship between the two methods. Subtracting the value baseline from the OPD$_{\text{full-V}}$ gradient (Eq. (5)) gives

$$\begin{aligned} &\nabla_\theta \mathcal{J}^{+\text{baseline}}_{\text{OPD}_{\text{full-V}}} = \\ &\mathbb{E}\left[\sum_{t=1}^{|y|} \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t)\, a_t(c_t, v)\, \nabla_\theta \log \pi_\theta(v \mid c_t)\right], \end{aligned} \tag{12}$$

which is identical to the original gradient because the baseline contribution vanishes:

$$\begin{aligned} &V^{\pi_\theta}(c_t) \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) \nabla_\theta \log \pi_\theta(v \mid c_t) \\ &\quad = V^{\pi_\theta}(c_t) \nabla_\theta \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) = 0. \end{aligned} \tag{13}$$

Because OPD$_{\text{full-V}}$ computes the full KL, its gradient already has zero variance at c_t, leaving nothing for the baseline to reduce. The baseline becomes useful only once we replace the full-vocabulary expectation with a Monte Carlo estimate, as is done in *v*OPD$_{\text{full-V}}$.

3.2. Top-k Approximation

While *v*OPD$_{\text{full-V}}$ adds no additional backward-pass cost, it still requires the exact KL computation at $O(|\mathcal{V}|)$ cost. Similar to OPD$_{\text{top-}k}$ (Eq. (6)), we can approximate the baseline KL on the student's top-k support to further reduce compute:

$$\hat{b}_t(c_t) = -\mathbb{D}_{\text{KL}}\big(\bar{\pi}_\theta(\cdot \mid c_t) \,\big\|\, \bar{\pi}_T(\cdot \mid c_t)\big), \tag{14}$$

where $\bar{\pi}$ is the renormalized distribution on the student's top-k support S_t with $k \ll |\mathcal{V}|$. Substituting $\hat{b}_t$ into Eq. (8) gives the *v*OPD$_{\text{top-}k}$ gradient estimator:

$$\begin{aligned} &\hat{a}_t(c_t, y_t) = r_t(c_t, y_t) + \mathbb{D}_{\text{KL}}\big(\bar{\pi}_\theta(\cdot \mid c_t) \,\|\, \bar{\pi}_T(\cdot \mid c_t)\big), \\ &\nabla_\theta \mathcal{J}_{v\text{OPD}_{\text{top-k}}}(\theta) = \mathbb{E}\left[\sum_{t=1}^{|y|} \hat{a}_t(c_t, y_t) \nabla_\theta \log \pi_\theta(y_t \mid c_t)\right]. \end{aligned} \tag{15}$$

The crucial distinction from OPD$_{\text{top-}k}$. While both methods compute KL with a top-k approximation, they place it in different positions of the estimator. OPD$_{\text{top-}k}$ uses it as the *loss*, replacing $\text{KL}(\pi_\theta \| \pi_T)$ with $\text{KL}(\bar{\pi}_\theta \| \bar{\pi}_T)$, thus changing the optimization target and biasing the gradient. *v*OPD$_{\text{top-}k}$ uses it as a *detached baseline* subtracted from the reward. As discussed in § 2.2, because $\hat{b}_t$ depends only on π_θ, π_T, and S_t but not on the sampled token y_t, the gradient remains unbiased. Furthermore, since $\hat{b}_t$ still approximates the value function $V^{\pi_\theta}(c_t)$, it can still reduce variance. The same approximation in different positions thus has completely different consequences, which we further confirm empirically in § 4.2: *v*OPD$_{\text{top-}k}$ allows substantial gains in practice compared to OPD, while OPD$_{\text{top-}k}$ does not.

3.3. Summary: Algorithm Comparison

Table 1 compares the discussed algorithms along the key axes of bias, variance, and compute. Base OPD is the computationally lightest but suffers from high gradient variance due to its single-sample Monte Carlo estimator. OPD$_{\text{full-V}}$

Table 1. Comparison of OPD variants along key algorithmic axes. Our methods are highlighted.

Method	Gradient unbiased?	Gradient variance at c_t	Backward token count	Per-token KL cost	Total additional compute
OPD	✓	High	1	—	None
$\text{OPD}_{\text{full-V}}$	✓	None	$\lvert\mathcal{V}\rvert$	$O(\lvert\mathcal{V}\rvert)$	High
$\text{OPD}_{\text{top-}k}$	✗	None	k	$O(k)$	Medium
$v\text{OPD}_{\text{full-V}}$	✓	Low	1	$O(\lvert\mathcal{V}\rvert)$	Low
$v\text{OPD}_{\text{top-}k}$	✓	Low	1	$O(k)$	Very Low

eliminates this variance by computing the per-token KL at c_t over the full vocabulary, but requires $O(|\mathcal{V}|)$ cost for both the per-token KL computation and the backward pass. $\text{OPD}_{\text{top-}k}$ reduces both costs to $O(k)$, but changes the objective by restricting the KL to a truncated support, thereby biasing the gradient. In contrast, $v\text{OPD}_{\text{full-V}}$ preserves base OPD's unbiased single-token estimator while reducing variance via the value baseline, adding only an additional per-token KL computation in the forward pass. $v\text{OPD}_{\text{top-}k}$ further approximates this baseline on the student's top-k support, preserving unbiasedness while achieving variance reduction at the lowest compute.

4. Experiments

4.1. Experimental Setup

Models and methods. Our primary setting distills Qwen3-1.7B into Qwen3-1.7B-Base (Yang et al., 2025), mirroring a common industrial OPD configuration where a post-trained checkpoint is distilled back into its base model (Zeng et al., 2026; Yang et al., 2026; DeepSeek-AI, 2026). We additionally evaluate three axes: (i) scale, Qwen3-4B into Qwen3-4B-Base; (ii) size mismatch, Qwen3-1.7B into Qwen3-0.6B-Base; and (iii) model family, Olmo-3-7B-Think into Olmo-3-7B-Base (Olmo et al., 2025). We compare *v*OPD against the three OPD variants from § 2.1: base OPD, $\text{OPD}_{\text{full-V}}$, and $\text{OPD}_{\text{top-}k}$, with both $v\text{OPD}_{\text{full-V}}$ and $v\text{OPD}_{\text{top-}k}$. For $\text{OPD}_{\text{top-}k}$ we set $k = 20$ following Li et al. (2026), who show that gains saturate beyond k=16. For $v\text{OPD}_{\text{top-}k}$ we likewise default to $k = 20$ and verify robustness in § 4.3.

Mathematical reasoning. We train on the English subset of DAPO-Math-17K (Yu et al., 2025), consisting of 14K training samples, for a single epoch. We evaluate on MATH500 (Hendrycks et al., 2021; Lightman et al., 2024), Minerva Math (Lewkowycz et al., 2022), AMC23 (Mathematical Association of America, 2026b), and AIME24/25 (Mathematical Association of America, 2026a), reporting avg@n and pass@n with n=8 for MATH500 and Minerva Math and n=32 for the smaller AMC and AIME benchmarks (see § B for details).

Scientific reasoning. To test generalization beyond mathematics, we train Qwen3-1.7B into Qwen3-1.7B-Base on scientific reasoning. Specifically, we use the chemistry subset of SciKnowEval (Feng et al., 2024), partitioned into train/eval/test splits of 75/5/20, following recent practice (Kim et al., 2026; Hübotter et al., 2026; Zhang et al., 2026b; Shenfeld et al., 2026). We evaluate on the test set and on GPQA-Diamond (Rein et al., 2024) (see § B for details).

4.2. Mathematical Reasoning Results

Table 2 summarizes our primary results. Across the main model configurations, *v*OPD consistently improves over base OPD by a substantial margin. In the Qwen3-1.7B-Base setting, $v\text{OPD}_{\text{top-}k}$ and $v\text{OPD}_{\text{full-V}}$ achieve absolute gains of up to +6.2% on MATH500 and above +3% on average. These improvements extend to the 4B scale, where both *v*OPD variants gain around +4% on MATH500 and around +2.5% on average over base OPD. Crucially, across all settings the two *v*OPD variants, $v\text{OPD}_{\text{full-V}}$ and $v\text{OPD}_{\text{top-}k}$, achieve nearly identical performance, confirming that the top-k baseline approximation captures the essential variance reduction without loss of accuracy. In contrast, $\text{OPD}_{\text{top-}k}$ yields only marginal gains over base OPD, for example +0.4% average at 1.7B, consistent with the finding of Li et al. (2026), potentially attributable to the bias in the objective. Overall, *v*OPD performs competitively with $\text{OPD}_{\text{full-V}}$, while adding only a lightweight forward-pass computation.

These patterns are further supported by the Olmo-3-7B and Qwen3-0.6B-Base experiments reported in Table 5. On the Olmo setting, $v\text{OPD}_{\text{top-}k}$ reaches an average of 33.1% compared to 29.9% for base OPD, again on par with $\text{OPD}_{\text{full-V}}$. The same pattern holds in the size-mismatched Qwen3-0.6B-Base setting, where $v\text{OPD}_{\text{full-V}}$ achieves the highest average of 21.1% and $v\text{OPD}_{\text{top-}k}$ follows closely at 20.0%, while $\text{OPD}_{\text{top-}k}$ provides only modest benefits and still trails behind. The consistency of these gains across model scales, size-mismatched teacher-student pairs, and model families supports the claim that a control variate baseline provides a general and robust mechanism for stabilizing OPD.

Table 2. Performance on mathematical reasoning benchmarks. Best performance is **bolded**, and second best performance is underlined.

Method	MATH500		MINERVA		AMC23		AIME24/25		Avg.
	Avg@8	Pass@8	Avg@8	Pass@8	Avg@32	Pass@32	Avg@32	Pass@32	
Qwen3-1.7B → Qwen3-1.7B-Base									
Student	42.3	79.8	13.8	37.1	23.2	85.0	3.0	20.0	20.6
OPD	58.7	82.6	22.2	40.8	33.4	77.5	4.8	28.3	29.8
$\text{OPD}_{\text{top-}k}$	58.0	84.0	23.4	44.5	35.5	85.0	4.0	**30.0**	30.2
$\text{OPD}_{\text{full-V}}$	64.6	**85.4**	25.0	44.5	**36.5**	**87.5**	5.8	25.0	33.0
$v\text{OPD}_{\text{top-}k}$	**64.9**	84.8	25.2	44.5	36.1	82.5	5.6	26.7	33.0
$v\text{OPD}_{\text{full-V}}$	64.0	84.6	**26.2**	**48.9**	36.1	80.0	**6.2**	25.0	**33.1**
Qwen3-4B → Qwen3-4B-Base									
Student	51.1	86.4	15.9	45.2	38.7	90.0	7.8	26.8	28.4
OPD	75.2	90.8	35.3	54.0	50.5	90.0	10.4	38.3	42.9
$\text{OPD}_{\text{top-}k}$	75.0	91.1	35.3	52.2	50.7	**95.0**	10.1	**41.7**	42.8
$\text{OPD}_{\text{full-V}}$	78.6	91.8	36.3	53.3	51.0	87.5	**14.3**	40.0	45.1
$v\text{OPD}_{\text{top-}k}$	**79.3**	**93.0**	**37.6**	**54.4**	51.2	87.5	13.2	**41.7**	45.3
$v\text{OPD}_{\text{full-V}}$	78.9	92.4	37.2	54.0	**52.0**	92.5	13.5	40.0	**45.4**

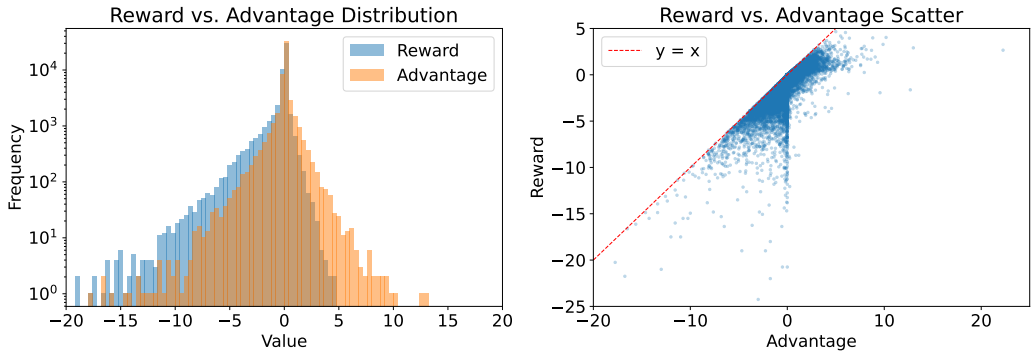

Figure 1. Token-level reward and advantage distributions. **Left:** The marginal distributions. **Right:** Per-token scatter plot (x: advantage, y: reward).

4.3. Further Analysis

Advantage vs. Reward. To further understand the effect of *v*OPD, we examine how the transformation from per-token reward r_t to advantage $a_t = r_t + \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_T)$ reshapes the training signal. Specifically, we log all token-level r_t and a_t from the first batch of 64 prompts (approximately 55k tokens) in the Qwen3-1.7B into Qwen3-1.7B-Base setting. Figure 1 (left) shows the frequency distributions. The OPD reward distribution exhibits a pronounced negative long tail, consistent with recent reports (Ko et al., 2026) and our discussion in § 3.1. The *v*OPD advantage distribution is visibly shifted rightward with the long tail compressed toward zero, which follows directly from the baseline: since $\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_T) \geq 0$, the shift is always non-negative.

We further analyze the token-level effect of this shift in Figure 1 (right), which plots the per-token advantage (x-axis) against reward (y-axis). All points lie on or to the right of $y = x$, confirming that the baseline can only shift rewards positively. Notably, positive-reward tokens are largely unchanged, whereas among tokens with similarly negative rewards, some are dampened almost entirely to zero while others retain advantages close to their original values. This follows directly from the baseline's definition: because the subtracted quantity is the token-level KL divergence at context c_t, tokens at high-KL contexts receive a large positive shift that absorbs most of the negative reward, while tokens at low-KL contexts are left nearly intact.

This selectivity has a natural interpretation. A large negative reward arises when the student assigns high probability to a token the teacher considers unlikely. Suppressing this token is likely to shift its mass toward the student's other high-probability candidates. In low-KL contexts, the teacher's density for these alternative tokens is also likely to be high, yielding an informative gradient with minimal influence from *v*OPD. In high-KL contexts, however, these tokens may be less probable for the teacher, resulting in a harmful

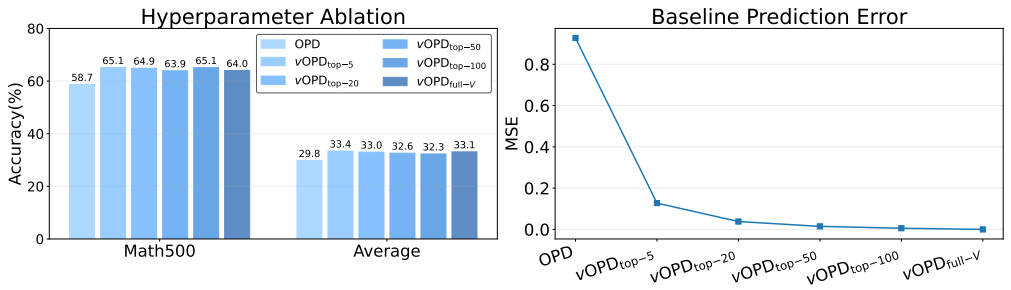

Figure 2. **Left:** Average accuracy for various k on vOPD$_{\text{top-}k}$. **Right:** Mean squared error of the top-k KL baseline relative to the full-vocabulary KL.

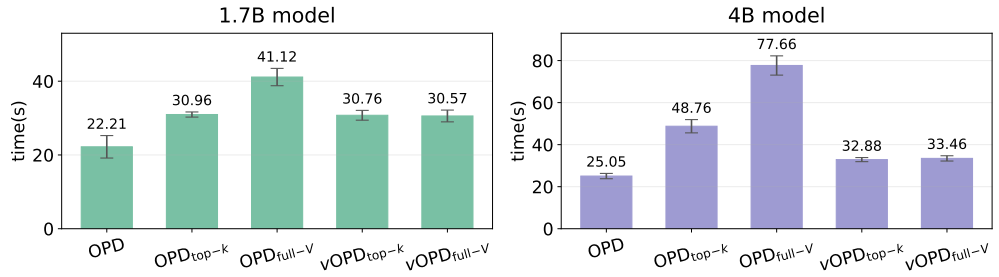

Figure 3. Per-step wall-clock time for OPD variants at 1.7B and 4B scale. The error bars denote variance.

gradient that can be mitigated by the high baseline from vOPD. This is consistent with Eq. (11), where variance reduction scales with $\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_T)^2$, largest at exactly the contexts where updates are least informative. Prior work has identified these high-mismatch tokens as the dominant source of gradient instability in OPD (Li et al., 2026), and the fact that vOPD's selective suppression improves rather than degrades accuracy (Table 2) confirms that what is removed is noise rather than signal—an effect that simple gradient clipping cannot replicate.

Hyperparameter Sensitivity. We ablate the top-k hyperparameter in vOPD$_{\text{top-}k}$ using the Qwen3-1.7B into Qwen3-1.7B-Base setting. Figure 2 (left) shows that average accuracy is stable across $k \in \{5, 20, 50, 100\}$ and the full-vocabulary baseline, with all values substantially outperforming OPD, which is interpretable as the k=0 case where no baseline is used. The key finding is that any nonzero k suffices: even the coarsest approximation at k=5 provides enough variance reduction to stabilize training. Further detailed results are reported in Table 5 (bottom). Figure 2 (right) plots the mean squared error of the top-k KL estimate relative to the full-vocabulary KL baseline; for base OPD, this equals the squared full-vocabulary KL itself, since no baseline is subtracted. The approximation error is low across all tested values of k and decreases monotonically, dropping to near zero beyond k=20. Notably, the k=5 estimate carries non-trivial approximation error yet still matches the accuracy of the full-vocabulary baseline, suggesting that a coarse approximation of the value function is sufficient for stable training, even without a precise estimate.

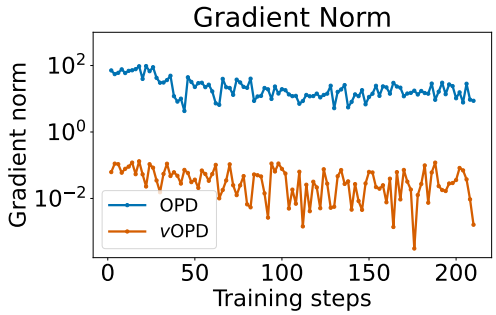

Figure 4. Gradient norm in training.

Wall-Clock Time. We compare per-step wall-clock time across all five methods at Qwen3-1.7B and 4B scales using a single NVIDIA H200 GPU in Figure 3. At 1.7B, base OPD is the fastest, followed by vOPD$_{\text{top-}k}$, vOPD$_{\text{full-V}}$, and OPD$_{\text{top-}k}$, which cluster at a modest overhead; OPD$_{\text{full-V}}$ is the most expensive due to its full-vocabulary backward pass. At 4B, the gaps widen: OPD$_{\text{top-}k}$ becomes considerably more expensive compared to both vOPD variants, and vOPD$_{\text{top-}k}$ pulls slightly ahead of vOPD$_{\text{full-V}}$ in speed, while the overall ordering remains the same. Combined with the results in Table 2, vOPD$_{\text{top-}k}$ offers the best accuracy-to-compute tradeoff among all compared methods, and the widening gap at 4B scale highlights the scalability advantage of vOPD.

Gradient Norm. To further understand the empirical stabilization effect, we plot gradient norms throughout training

Table 3. Ablation results on scientific reasoning benchmarks. Best performance is **bolded**, and second best performance is underlined.

Method	SciKnowEval	GPQA-D
Student	26.1	20.4
OPD	29.3	24.7
$\text{OPD}_{\text{top-}k}$	29.7	24.1
$\text{OPD}_{\text{full-V}}$	**35.1**	**28.7**
$v\text{OPD}_{\text{top-}k}$	33.2	28.6
$v\text{OPD}_{\text{full-V}}$	34.7	28.4

for OPD and $v\text{OPD}_{\text{full-V}}$ in the Qwen3-1.7B setting in Figure 4. Notably, *v*OPD maintains gradient norms 1–2 orders of magnitude lower than base OPD. Despite the substantially smaller gradients, *v*OPD trains stably and reaches higher accuracy (Table 2), confirming that the large gradients in base OPD are dominated by variance rather than useful signal, and that *v*OPD successfully suppresses this instability.

4.4. Scientific Reasoning Results

Table 3 demonstrates that *v*OPD's gains generalize beyond mathematics to SciKnowEval (chemistry) and GPQA-Diamond. The overall findings are consistent with the mathematical reasoning results, as $\text{OPD}_{\text{full-V}}$ and both *v*OPD variants lead in performance, gaining around +4% over base OPD on both benchmarks. Similarly, $\text{OPD}_{\text{top-}k}$ shows little gain over base OPD. The consistency of these results with the mathematical reasoning setting supports the view that *v*OPD's gains stem from a general, principled control variate mechanism.

5. Related Work

5.1. On-Policy Distillation

OPD has become an important component of LLM post-training, especially for long Chain-of-Thought reasoning tasks (Wei et al., 2022), where dense token-level teacher signals offer a compute-efficient alternative to sparse RLVR rewards. Early works such as GKD (Agarwal et al., 2024) and MiniLLM (Gu et al., 2024) established OPD as an effective alternative to standard distillation. Recent work has popularized token-level Monte Carlo OPD (Lu & Lab, 2025), studied practical recipes (Patiño et al., 2025; Li et al., 2026), and incorporated OPD into large-scale post-training systems (Yang et al., 2025; Zeng et al., 2026; Yang et al., 2026; DeepSeek-AI, 2026). Despite these successes, OPD remains unstable in practice. Follow-up studies have explored top-k support restrictions (Li et al., 2026), entropy-aware training (Jin et al., 2026), and prefix-only variants (Zhang et al., 2026a). Our work complements these efforts by treating OPD instability as an estimator-variance problem and addressing it with an unbiased control variate baseline.

5.2. Control Variate Baseline for RL

The control variate baseline is a core tool in on-policy policy-gradient reinforcement learning. This principle underlies actor-critic methods, advantage estimation, and modern policy-gradient algorithms such as A3C and PPO with GAE (Williams, 1992; Sutton et al., 1999; Mnih et al., 2016; Schulman et al., 2017; 2016). The same idea remains central in LLM reinforcement learning. Early RLHF pipelines used PPO (Ouyang et al., 2022; Schulman et al., 2017) with a learned value model to estimate advantages, while recent reasoning-oriented RLVR methods (Lambert et al., 2024) often replace the learned critic with a simpler relative baseline. GRPO and RLOO, for example, use rewards from multiple sampled responses to construct a relative baseline, and have become standard recipes for RLVR (Ahmadian et al., 2024; Shao et al., 2024; Guo et al., 2025; Yu et al., 2025). Follow-up methods such as SPO explore alternative single-stream baseline estimators (Xu & Ding, 2026), and POISE integrates mechanistic interpretability techniques for baseline estimation (Choi et al., 2026). Despite the central role of the baseline in RL, it has not been systematically explored for OPD, even though OPD admits a policy-gradient interpretation (Lu & Lab, 2025; Ko et al., 2026; Jin et al., 2026). Our work fills this gap by deriving a closed-form OPD value baseline and using it as an unbiased control variate.

6. Conclusion, Limitations, and Future Work

We introduced *v*OPD, a control variate formulation of On-Policy Distillation that reduces the variance of the single-sample Monte Carlo estimator without changing the original OPD objective. By using the negative reverse KL between the student and the teacher as a detached value baseline, *v*OPD preserves an unbiased policy-gradient estimator while retaining the single-token backward pass of base OPD. Experiments on reasoning benchmarks spanning mathematics and science show that *v*OPD improves training stability and performance over base OPD, acting as a principled regularizer on destabilizing negative reward tokens.

Several directions remain open for future work. Experiments in this work are limited to models up to 7B parameters, and validating *v*OPD at a larger scale is a next step. Wall-clock results reflect our implementation and are not definitive. Future work could aim to optimize $v\text{OPD}_{\text{top-}k}$ to be faster than $v\text{OPD}_{\text{full-V}}$. As a distillation method, *v*OPD requires access to a stronger teacher; extending it to self-distillation settings is an interesting direction. Finally, this work focuses on the token-level KL in OPD, and considering sequence-level KL objectives is another potential extension.

References

Agarwal, R., Vieillard, N., Zhou, Y., Stanczyk, P., Garea, S. R., Geist, M., and Bachem, O. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://aclanthology.org/2024.acl-long.662/.

Chen, A., Li, A., Gong, B., Jiang, B., Fei, B., Yang, B., Shan, B., Yu, C., Wang, C., Zhu, C., et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025. URL https://arxiv.org/abs/2506.13585.

Choi, Y., Lim, J., Ahn, W., Oh, M., Shim, J., and Jo, Y. Your language model is its own critic: Reinforcement learning with value estimation from actor's internal states. *arXiv preprint arXiv:2605.07579*, 2026. URL https://arxiv.org/abs/2605.07579.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf.

Feng, K., Shen, X., Wang, W., Zhuang, X., Tang, Y., Zhang, Q., and Ding, K. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024. URL https://arxiv.org/abs/2406.09098.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 2004. URL https://www.jmlr.org/papers/volume5/greensmith04a/greensmith04a.pdf.

Gu, Y., Dong, L., Wei, F., and Huang, M. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2103.03874.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Hübotter, J., Lübeck, F., Behric, L., Baumann, A., Bagatella, M., Marta, D., Hakimi, I., Shenfeld, I., Buening, T. K., Guestrin, C., et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026. URL https://arxiv.org/abs/2601.20802.

Jaech, A., Kalai, A., Lerer, A., Richardson, A., El-Kishky, A., Low, A., Helyar, A., Madry, A., Beutel, A., Carney, A., et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. URL https://arxiv.org/abs/2412.16720.

Jin, W., Min, T., Yang, Y., Kadhe, S. R., Zhou, Y., Wei, D., Baracaldo, N., and Lee, K. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026. URL https://arxiv.org/abs/2603.07079.

Khatri, D., Madaan, L., Tiwari, R., Bansal, R., Duvvuri, S. S., Zaheer, M., Dhillon, I. S., Brandfonbrener, D., and Agarwal, R. The art of scaling reinforcement learning compute for llms. *arXiv preprint arXiv:2510.13786*, 2025. URL https://arxiv.org/abs/2510.13786.

Kim, J., Luo, X., Kim, M., Lee, S., Kim, D., Jeon, J., Li, D., and Yang, Y. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026. URL https://arxiv.org/abs/2603.24472.

Ko, J., Abdali, S., Kim, Y. J., Chen, T., and Cameron, P. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026. URL https://arxiv.org/abs/2603.11137.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, 2023. URL https://arxiv.org/pdf/2309.06180.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S.,

et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. URL https://arxiv.org/abs/2411.15124.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2206.14858.

Li, Y., Zuo, Y., He, B., Zhang, J., Xiao, C., Qian, C., Yu, T., Gao, H.-a., Yang, W., Liu, Z., et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026. URL https://arxiv.org/abs/2604.13016.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

Liu, Z., Liu, J., He, Y., Wang, W., Liu, J., Pan, L., Hu, X., Xiong, S., Huang, J., Hu, J., Huang, S., Yang, S., Wang, J., Su, W., and Zheng, B. Tricks or traps? a deep dive into RL for LLM reasoning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2508.08221.

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

Mathematical Association of America. American Invitational Mathematics Examination, 2026a. URL https://maa.org/maa-invitational-competitions/.

Mathematical Association of America. American Mathematics Competitions, 2026b. URL https://maa.org/student-programs/amc/.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 2016. URL https://arxiv.org/abs/1602.01783.

Olmo, T., Ettinger, A., Bertsch, A., Kuehl, B., Graham, D., Heineman, D., Groeneveld, D., Brahman, F., Timbers, F., Ivison, H., et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025. URL https://arxiv.org/abs/2512.13961.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract.html.

Patiño, C. M., Rasul, K., Gallouédec, Q., Burtenshaw, B., Paniego, S., Srivastav, V., Frere, T., Beeching, E., Tunstall, L., von Werra, L., and Wolf, T. Unlocking on-policy distillation for any model family, 2025. URL https://huggingface.co/spaces/HuggingFaceH4/on-policy-distillation.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06732.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2311.12022.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1506.02438.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Shenfeld, I., Damani, M., Hübotter, J., and Agrawal, P. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026. URL https://arxiv.org/abs/2601.19897.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/

464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Tang, Y. and Munos, R. On a few pitfalls in kl divergence gradient estimation for rl. *arXiv preprint arXiv:2506.09477*, 2025. URL https://arxiv.org/abs/2506.09477.

Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., et al. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. URL https://arxiv.org/abs/2501.12599.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2201.11903.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992. URL https://link.springer.com/content/pdf/10.1007/BF00992696.pdf.

Xu, Z. and Ding, Z. Single-stream policy optimization. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.13232.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Yang, Z., Liu, Z., Chen, Y., Dai, W., Wang, B., Lin, S.-C., Lee, C., Chen, Y., Jiang, D., He, J., et al. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026. URL https://arxiv.org/abs/2603.19220.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., YuYue, Dai, W., Fan, T., Liu, G., Liu, J., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, R., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W.-Y., Zhang, Y.-Q., Yan, L., Wu, Y., and Wang, M. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2503.14476.

Zeng, A., Lv, X., Hou, Z., Du, Z., Zheng, Q., Chen, B., Yin, D., Ge, C., Huang, C., Xie, C., et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026. URL https://arxiv.org/abs/2602.15763.

Zhang, D., Yang, Z., Janghorbani, S., Han, J., Ressler II, A., Qian, Q., Lyng, G. D., Batra, S. S., and Tillman, R. E. Fast and effective on-policy distillation from reasoning prefixes. *arXiv preprint arXiv:2602.15260*, 2026a. URL https://arxiv.org/abs/2602.15260.

Zhang, J., Peng, X., Chen, Q., Ye, Q., Xiong, C., and Wu, C.-S. The illusion of certainty: Decoupling capability and calibration in on-policy distillation. *arXiv preprint arXiv:2604.16830*, 2026b. URL https://arxiv.org/abs/2604.16830.

Zheng, C., Liu, S., Li, M., Chen, X.-H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. URL https://arxiv.org/abs/2507.18071.

A. Theoretical Derivations

This appendix provides derivations omitted from § 2 and § 3. Following the main sections, we denote the context $c_t = (x, y_{<t})$, and the reward $r_t(c_t, y_t) = \log \pi_T(y_t \mid c_t) - \log \pi_\theta(y_t \mid c_t)$. We use the score-function identity:

$$\sum_v \pi_\theta(v \mid c_t)\, \nabla_\theta \log \pi_\theta(v \mid c_t) \;=\; \sum_v \nabla_\theta\, \pi_\theta(v \mid c_t) \;=\; \nabla_\theta \sum_v \pi_\theta(v \mid c_t) \;=\; 0, \tag{16}$$

which follows from the log trick $\nabla_\theta \log \pi_\theta(v \mid c_t) = \nabla_\theta \pi_\theta(v \mid c_t)/\pi_\theta(v \mid c_t)$.

A.1. Unbiasedness of Baseline Subtraction

We show that subtracting an action-independent baseline $b_t(c_t)$ from the per-token reward leaves the policy gradient unbiased (Williams, 1992; Sutton et al., 1999). Specifically, we show:

$$\mathbb{E}_{y_t \sim \pi_\theta(\cdot \mid c_t)}\Big[\big(r_t(y_t) - b_t(c_t)\big)\, \nabla_\theta \log \pi_\theta(y_t \mid c_t)\Big] \;=\; \mathbb{E}_{y_t \sim \pi_\theta(\cdot \mid c_t)}\Big[r_t(y_t)\, \nabla_\theta \log \pi_\theta(y_t \mid c_t)\Big]. \tag{17}$$

By linearity of expectation, the difference between the two sides equals

$$\mathbb{E}_{y_t \sim \pi_\theta(\cdot \mid c_t)}\Big[b_t(c_t)\, \nabla_\theta \log \pi_\theta(y_t \mid c_t)\Big] \;=\; b_t(c_t) \sum_v \pi_\theta(v \mid c_t)\, \nabla_\theta \log \pi_\theta(v \mid c_t), \tag{18}$$

where $b_t(c_t)$ factors out of the expectation because it is independent of y_t by assumption. The score-function identity (Eq. (16)) gives the remaining sum as zero, so the difference vanishes and Eq. (17) holds.

A.2. Variance Reduction and the Optimal Baseline

We derive the variance-reducing property of baseline subtraction and identify the optimal scalar choice (Greensmith et al., 2004). Consider the per-step gradient estimator:

$$g(b) \;=\; \big(r_t(y_t) - b\big)\, \nabla_\theta \log \pi_\theta(y_t \mid c_t), \qquad b \in \mathbb{R}. \tag{19}$$

We seek the b minimizing $\mathrm{tr}(\mathrm{Var}[g(b)])$. $\mathbb{E}[g(b)]$ does not depend on b because baseline subtraction is unbiased (§ A.1), so the variance is minimized by minimizing the second moment $\mathbb{E}[\|g(b)\|^2]$. Expanding,

$$\mathbb{E}\big[\|g(b)\|^2\big] \;=\; \mathbb{E}\Big[\big(r_t(y_t) - b\big)^2 \, \big\|\nabla_\theta \log \pi_\theta(y_t \mid c_t)\big\|^2\Big] \tag{20}$$

$$\;=\; \mathbb{E}\Big[r_t(y_t)^2 \, \big\|\nabla_\theta \log \pi_\theta(y_t \mid c_t)\big\|^2\Big] \;-\; 2b\, \mathbb{E}\Big[r_t(y_t) \, \big\|\nabla_\theta \log \pi_\theta(y_t \mid c_t)\big\|^2\Big] \tag{21}$$

$$\;+\; b^2\, \mathbb{E}\Big[\big\|\nabla_\theta \log \pi_\theta(y_t \mid c_t)\big\|^2\Big]. \tag{22}$$

This is a convex quadratic in b. Setting its derivative to zero yields the *optimal scalar baseline* $b^\star$:

$$b^\star \;=\; \frac{\mathbb{E}_{y_t \sim \pi_\theta(\cdot \mid c_t)}\Big[r_t(y_t) \, \big\|\nabla_\theta \log \pi_\theta(y_t \mid c_t)\big\|^2\Big]}{\mathbb{E}_{y_t \sim \pi_\theta(\cdot \mid c_t)}\Big[\big\|\nabla_\theta \log \pi_\theta(y_t \mid c_t)\big\|^2\Big]}. \tag{23}$$

The optimal baseline in Eq. (23) is a weighted form of the value function. In practice, the value function itself is used as the baseline, motivating the choice in § 3.1:

$$b^\star \;\approx\; \mathbb{E}_{y_t \sim \pi_\theta(\cdot \mid c_t)}\big[r_t(y_t)\big] \;=\; V^{\pi_\theta}(c_t). \tag{24}$$

A.3. Variance Reduction of *v*OPD

We now derive Eq. (11), the per-step variance reduction obtained by *v*OPD$_{\text{full-V}}$. Recall the two estimators:

$$g_{\text{OPD}} = r_t(c_t, y_t)\, \nabla \log \pi_\theta(y_t \mid c_t), \qquad g_{v\text{OPD}_{\text{full-V}}} = \big(r_t(c_t, y_t) - b_t\big)\, \nabla \log \pi_\theta(y_t \mid c_t), \tag{25}$$

where $b_t = V^{\pi_\theta}(c_t) = -\mathbb{D}_{\mathrm{KL}}(\pi_\theta(\cdot \mid c_t) \,\|\, \pi_T(\cdot \mid c_t))$. We want to show:

$$\mathrm{tr}\big(\mathrm{Var}[g_{\mathrm{OPD}}]\big) - \mathrm{tr}\big(\mathrm{Var}[g_{v\mathrm{OPD}_{\text{full-V}}}]\big) \;\approx\; \mathbb{D}_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid c_t) \,\big\|\, \pi_T(\cdot \mid c_t)\big)^2 \;\cdot\; \mathbb{E}\big[\|\nabla \log \pi(y_t \mid c_t)\|^2\big]. \tag{26}$$

We start from the second-moment expansion of § A.2. Since both estimators share the same expectation (by § A.1), the variance difference equals the difference of second moments:

$$\begin{aligned}\mathrm{tr}\big(\mathrm{Var}[g_{\mathrm{OPD}}]\big) - \mathrm{tr}\big(\mathrm{Var}[g_{v\mathrm{OPD}_{\text{full-V}}}]\big) &= \mathbb{E}\big[r_t^2 \,\|\nabla \log \pi_\theta(y_t|c_t)\|^2\big] - \mathbb{E}\big[(r_t - b_t)^2 \,\|\nabla \log \pi_\theta(y_t|c_t)\|^2\big] \\ &= 2b_t\, \mathbb{E}\big[r_t \,\|\nabla \log \pi_\theta(y_t|c_t)\|^2\big] - b_t^2\, \mathbb{E}\big[\|\nabla \log \pi_\theta(y_t|c_t)\|^2\big].\end{aligned}$$

Under a weak-correlation approximation $\mathbb{E}[r_t \,\|\nabla \log \pi_\theta(y_t|c_t)\|^2] \approx b_t\, \mathbb{E}[\|\nabla \log \pi_\theta(y_t|c_t)\|^2]$, this yields:

$$\mathrm{tr}\big(\mathrm{Var}[g_{\mathrm{OPD}}]\big) - \mathrm{tr}\big(\mathrm{Var}[g_{v\mathrm{OPD}_{\text{full-V}}}]\big) \;\approx\; b_t^2\, \mathbb{E}\big[\|\nabla \log \pi_\theta(y_t|c_t)\|^2\big]. \tag{27}$$

Substituting $b_t^2 = \mathbb{D}_{\mathrm{KL}}(\pi_\theta \| \pi_T)^2$ recovers Eq. (26).

Weak-correlation approximation. We discuss why a weak-correlation assumption is reasonable in the OPD setting. Because y_t is sampled from π_θ, sampled tokens naturally lie in the high-probability region, so $\log \pi_\theta(y_t \mid c_t)$ varies in a narrow band and $\|\nabla \log \pi\|^2$, a function of π_θ at y_t, can inherit this concentration. In contrast, the variation in $r_t = \log \pi_T(y_t \mid c_t) - \log \pi_\theta(y_t \mid c_t)$ is primarily driven by the teacher term, which has no structural dependence on $\pi_\theta(y_t \mid c_t)$. Figure 5 confirms this empirically, showing scatter plots from the same data used in § 4.3: $\log \pi_\theta(y_t)$ concentrates near zero with weak correlation to r_t, while $\log \pi_T(y_t \mid c_t)$ spans a wider range and exhibits a clear correlation with r_t. This is also consistent with the analysis in § 4.3 and recent long-tail negative reward structure report (Ko et al., 2026): since $\log \pi_\theta(y_t \mid c_t)$ stays close to zero, r_t cannot spike in the positive direction, but can extend into large negative values whenever $\log \pi_T(y_t \mid c_t)$ drops—making the teacher term the dominant driver of variation in r_t.

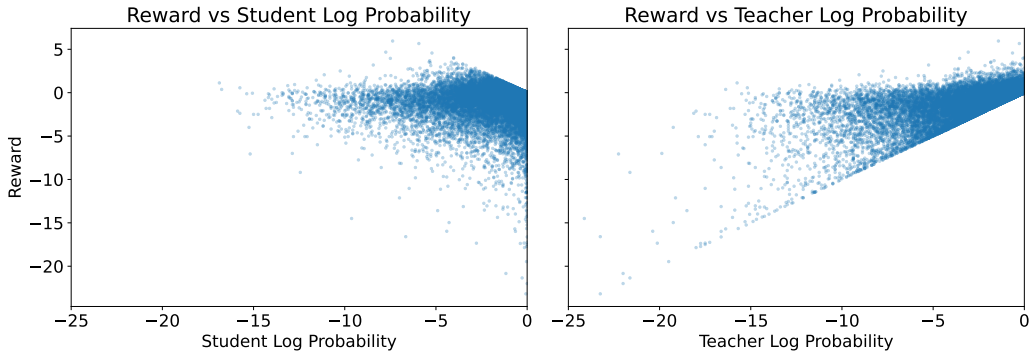

Figure 5. Scatter plot. **Left:** student log probability (x) vs. reward (y). **Right:** teacher log probability (x) vs. reward (y).

B. Experiment Settings

This section details the experimental settings from § 4.1. All experiments were conducted on a single NVIDIA H200 NVL or A100 NVL GPU, paired with an Intel(R) Xeon(R) Gold 6530 or Gold 6230 CPU @ 2.10GHz, respectively. All training runs took between 2h (1.7B on A100) and 6h (7B on H200).

B.1. Training Settings

Table 4 summarizes the hyperparameters for training and evaluation. Following recent work demonstrating that the maximum response length in OPD need not be excessively long (2–3K tokens suffice), we set the maximum response length to 2048 (Zhang et al., 2026a; Li et al., 2026). Following recent work on OPD, we set the student sampling temperature to 1.0 (Li et al., 2026; Ko et al., 2026). We adopt parameter-efficient training via LoRA (Hu et al., 2022). Our implementation builds on `TRL`'s MiniLLM[1] implementation (von Werra et al., 2020; Gu et al., 2024). When using Qwen as the teacher model, we disable its extended thinking mode. For scientific reasoning, which comprises 1,890 train samples from the chemistry subset of SciKnowEval (Feng et al., 2024), we extend training to a maximum of 10 epochs.

[1] https://huggingface.co/docs/trl/main/minillm

For evaluation, following the official guidelines for Qwen3 and Olmo-3 (Yang et al., 2025; Olmo et al., 2025), we set the sampling temperature to 0.6 and top-p sampling to 0.9. We use `vLLM` for accelerated inference (Kwon et al., 2023).

Table 4. Training and Evaluation Settings

Hyperparameter	Train	Eval	Hyperparameter	Train	Eval
Prompt length	1024	1024	Response length	2048	4096
Rollout temperature	1.0	0.6	Top-p sampling	1.0	0.9
Batch size	64	–	Optimizer	AdamW	–
Learning rate	1e-5 (Qwen3), 2e-5 (Olmo-3)	–	LoRA rank	64	–
Epochs	1 (math), 10 (science)	–	LoRA α	128	–
Engine	`TRL`	`vLLM`	Precision	`bfloat16`	`bfloat16`

B.2. Prompts

We use the following prompts for mathematical and scientific reasoning throughout training and evaluation, following the official prompts (Yang et al., 2025).

Math Reasoning Prompt Template

{`problem`}\n\nPlease reason step by step, and put your final answer within \\boxed{}.

Scientific Reasoning Prompt Template

{`problem`}\n\nReturn only one JSON object with the key "answer". The value must be exactly one choice letter, for example: {"answer": "C"}.

C. Extended Experiment Results

Table 5 presents the extended results. Specifically, it reports: (i) the accuracy of the teacher models used in our experiments; (ii) the results of distilling Olmo-3-7B-Think into Olmo-3-7B-Base and Qwen3-1.7B into Qwen3-0.6B-Base, from § 4.2; and (iii) comprehensive benchmark results for the hyperparameter ablations in Figure 2.

Table 5. Additional mathematical reasoning benchmark results. Best performance is **bolded**, and second best performance is underlined.

	MATH500		MINERVA		AMC23		AIME24/25		
Method	Avg@8	Pass@8	Avg@8	Pass@8	Avg@32	Pass@32	Avg@32	Pass@32	**Avg.**
Teacher Performance									
Qwen3-1.7B	73.0	90.8	28.6	42.6	44.8	85.0	14.4	35.0	40.2
Qwen3-4B	83.5	93.8	43.9	49.6	65.9	95.0	20.2	51.7	53.4
Olmo-3-7B-Think	87.2	95.4	39.2	52.2	70.0	97.5	37.2	68.3	58.4
Olmo-3-7B-Think → Olmo-3-7B-Base									
Student	43.1	83.4	11.6	35.3	20.6	**70.0**	5.4	**30.0**	20.2
OPD	61.2	78.6	18.4	36.0	32.5	50.0	7.4	21.7	29.9
$\text{OPD}_{\text{top-}k}$	58.8	75.4	21.5	39.0	32.3	45.0	5.5	15.0	29.5
$\text{OPD}_{\text{full-V}}$	62.8	**83.6**	21.7	**43.0**	34.0	62.5	**8.6**	23.4	31.8
$v\text{OPD}_{\text{top-}k}$	64.0	81.0	**23.9**	41.9	35.9	57.5	8.4	25.0	**33.1**
$v\text{OPD}_{\text{full-V}}$	**64.4**	83.2	22.5	40.4	**36.8**	65.0	7.9	21.7	32.9
Qwen3-1.7B → Qwen3-0.6B-Base									
Student	30.9	66.8	5.6	21.7	12.3	60.0	0.4	6.7	12.3
OPD	41.4	73.0	8.1	26.1	19.3	**80.0**	1.1	13.3	17.5
$\text{OPD}_{\text{top-}k}$	43.7	72.0	12.7	**32.4**	21.8	65.0	**1.5**	**13.4**	19.9
$\text{OPD}_{\text{full-V}}$	**47.3**	72.0	12.8	29.4	20.7	65.0	1.4	13.3	20.6
$v\text{OPD}_{\text{top-}k}$	44.4	71.0	13.7	29.0	20.9	67.5	0.9	10.0	20.0
$v\text{OPD}_{\text{full-V}}$	46.5	**73.2**	**14.3**	32.0	**22.3**	67.5	1.4	10.0	**21.1**
Qwen3-1.7B → Qwen3-1.7B-Base (Ablation)									
OPD	58.7	82.6	22.2	40.8	33.4	77.5	4.8	**28.3**	29.8
$v\text{OPD}_{\text{top-5}}$	**65.1**	83.8	26.0	45.2	35.6	85.0	**6.7**	26.7	**33.4**
$v\text{OPD}_{\text{top-20}}$	64.9	**84.8**	25.2	44.5	**36.1**	82.5	5.6	26.7	33.0
$v\text{OPD}_{\text{top-50}}$	63.9	84.6	25.6	44.1	35.0	**87.5**	6.1	23.3	32.6
$v\text{OPD}_{\text{top-100}}$	**65.1**	76.4	25.7	32.0	33.4	75.0	5.2	26.7	32.3
$v\text{OPD}_{\text{full-V}}$	64.0	84.6	**26.2**	**48.9**	**36.1**	80.0	6.2	25.0	33.1