INTERACTIVE-POLICY DISTILLATION WITH BIDIRECTIONAL PROPOSE-AND-VERIFY

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) trains a compact student model on its self-generated trajectories with dense token-level teacher feedback. However, naive OPD may suffer from teacher unanchoring, where the student's reasoning trajectory drifts outside the teacher's acceptance region, causing the teacher to be queried on states it would hardly visit and thus provide unreliable supervision. We propose **Interactive-Policy Distillation (IPD)**, which dynamically corrects training trajectories within adaptive teacher intervention. Under a bidirectional propose-and-verify state machine, the student and teacher alternately exchange their roles as proposer and verifier, and collaboratively generate mixed-source trajectories. Then different supervisions are applied according to the source of each token. This bidirectional propose-and-verify mechanism and the source-split loss make IPD not only a more performant distillation method, but also a unified bridge between on-policy and off-policy paradigms. To make the interleaved dual-model rollouts more efficient, we also design a dedicated fused inference engine that co-hosts both models in one serving instance with separate KV caches and instantiates the state machine model to distribute, collect, and process requests. On math reasoning tasks and across multiple teacher-student model pairs, student models trained with IPD not only outperform those trained with OPD, but also demonstrate higher data efficiency. Specifically, when distilling Qwen3-30B-A3B (non-thinking) into Qwen3-1.7B-Base, IPD brings a $+3.28$ mean@8 and a $+3.28$ best@8 benchmark-averaged accuracy improvement compared with OPD. Besides, IPD only consumes about $1/4$ of the training examples and steps to outperform OPD trained on the whole training dataset for one epoch. We also investigate the impact of different loss variants and takeover / handback configurations, and demonstrate the robustness of IPD on different training data.

1 INTRODUCTION

Large language models (LLMs) have advanced at a remarkable pace over the past few years, and current frontier models attain expert-level performance in mathematics, programming, and scientific problems, and drive increasingly capable agents (OpenAI, 2026; Anthropic, 2026; Z.ai, 2026; Xu et al., 2026). These extraordinary capabilities, however, are usually obtained by the very largest models, which pair over hundreds of billions of parameters and long chains of thought. The high memory and computation cost makes them a suboptimal fit for latency-sensitive, on-device, or high-throughput deployments. Knowledge distillation (Hinton et al., 2015) offers a way out by transferring as much of the large model's capability as possible into a more compact and cheaper student model. In the LLM era, the dominant paradigm used to be off-policy distillation via supervised fine-tuning (SFT), where the teacher generates high-quality reasoning traces, and the student learns to imitate them token by token (Kim & Rush, 2016; Ho et al., 2023; Magister et al., 2023; Hsieh et al., 2023). The distilled variants of DeepSeek-R1, trained by SFT on roughly 800K teacher-generated samples, are a prominent recent example of how far this simple recipe scales (Guo et al., 2025).

Despite its scalability, off-policy distillation faces a structural challenge: the student is trained under the teacher's demonstrations, and the imitation is only ever supervised on teacher-anchored states that the student is not likely to visit. Therefore, the student never learns to continue from its own partial generations, and small early deviations can compound at inference time, as discussed in the classic exposure-bias problem (Bengio et al., 2015; Ranzato et al., 2015; Arora et al., 2022). Long

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.

and stylized teacher traces can exceed what a small student is able to absorb (Mirzadeh et al., 2020; Shing et al., 2025; Li et al., 2025), and the forward KL pushes the student to spread probability mass over every teacher mode rather than reliably on modes it can actually realize (Gu et al., 2024). On-policy distillation (OPD) overcomes these challenges by moving the training distribution to the student. The student generates its own responses, and the teacher supervises every generated token with dense probabilistic feedback (Agarwal et al., 2024; Gu et al., 2024; Lu & Lab, 2025). Training on self-generated states removes the training-inference mismatch by construction. The student learns from states it actually visits (Ross et al., 2011), while the teacher's per-token distributions provide denser learning signals than scalar outcome rewards of reinforcement learning with verifiable rewards (RLVR). With those advantages, OPD can match RLVR-based training with much less compute (Lu & Lab, 2025), and has quickly moved into production practice as a core stage of post-training pipelines for building reasoning models (Yang et al., 2025; Ma et al., 2026), accompanied by a growing line of refinements on divergence choice and data mixing (Ko et al., 2024).

However, student-generated trajectories can deviate far from the teacher when there is a large reasoning gap between them, and then the teacher is queried by states it would hardly visit and may provide unreliable learning signals (Ko et al., 2024; Xu et al., 2025). We call this failure mode *teacher unanchoring*. To overcome this challenge, we introduce **Interactive-Policy Distillation (IPD)**, which couples a bidirectional propose-and-verify state machine with different supervisions on tokens from different sources. As shown in Figure 1, IPD allows the teacher to dynamically correct a drifting rollout and returns control once the corrected continuation falls within the student's acceptance region where student can actually resume. As the resulting trajectory interleaves the student's own decisions with teacher-provided corrective demonstrations, IPD assigns different supervisions by token sources, *i.e.*, reverse KL at student-generated positions and forward KL at teacher-generated positions. Our contributions are summarized as follows:

- We introduce IPD, a novel distillation algorithm with adaptive teacher intervention through a bidirectional propose-and-verify state machine, in which student and teacher dynamically exchange proposer and verifier roles. Its source-split objective applies reverse KL to student-generated positions and forward KL to teacher-generated positions, bridging on-policy and off-policy distillation paradigms.
- We design a fused inference engine, which co-hosts both models in one serving instance, maintains separate KV caches, and schedules requests according to their current proposer and verifier roles, to achieve more efficient dual-model scheduling during the rollout stage.
- Empirically, across four teacher-student pairs on mathematical reasoning, IPD achieves the highest benchmark-averaged mean@8 and best@8 accuracy and leads on most individual benchmark metrics. Besides, IPD shows higher data efficiency. When distilling Qwen3-30B-A3B to Qwen3-1.7B-Base, IPD surpasses the mean@8 of OPD trained for a full epoch (279 steps) after only 60 training steps. We also demonstrate the robustness of IPD across different training datasets.

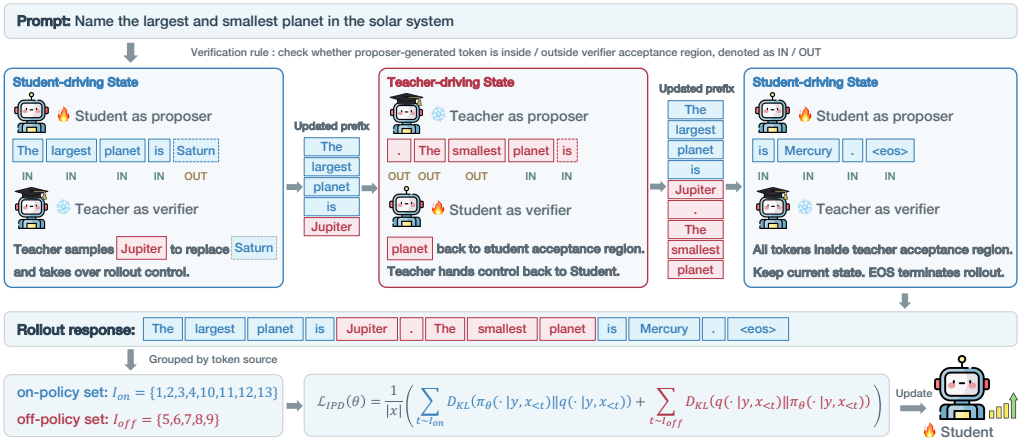


Figure 1: The rollout and training demonstration of IPD. The student and teacher collaboratively generate the rollout under a bidirectional propose-and-verify state machine. Then tokens are grouped by their sources, and the IPD source-split loss is applied to update the student weights.

2 Related Works

On-policy distillation for LLMs. LLM distillation used to be dominated by off-policy distillation on teacher-generated traces (Hinton et al., 2015; Kim & Rush, 2016; Guo et al., 2025). The main weakness of off-policy distillation is that the student trains on teacher-authored prefixes but deploys on its own, and is vulnerable to exposure bias (Bengio et al., 2015; Arora et al., 2022). GKD (Agarwal et al., 2024) applies purely on-policy student sequences for training, and MiniLLM (Gu et al., 2024) studies sequence-level reverse KL on those sequences. DistiLLM (Ko et al., 2024) introduces skew divergences with an adaptive off-policy scheduler, and DistiLLM-2 (Ko et al., 2025) contrasts teacher and student responses with skew forward and reverse KL. Recently, OPD has already been incorporated as a core stage into reasoning-model post-training pipelines (Yang et al., 2025; Lu & Lab, 2025; Xiao et al., 2026).

Challenges in on-policy distillation. Recent analyses characterize unreliable teacher feedback on student rollouts (Fu et al., 2026; Armandpour et al., 2026; Li et al., 2026). EOPD (Jin et al., 2026) adjusts the loss while retaining student-generated rollouts. TrOPD (Xing et al., 2026) uses a teacher-generated prefix followed by a student continuation and applies region-specific losses, but without dynamic teacher intervention during the continuation. SKD (Xu et al., 2025) adapts speculative propose-and-verify (Leviathan et al., 2023; Chen et al., 2023) to rollout and implements dynamic teacher corrections. The student and teacher remain fixed roles as proposer and verifier. When a student token is rejected, the teacher resamples that position, and the student resumes generation from the corrected prefix. Interactive imitation learning studies intervention and return of control through human judgment in HG-DAgger (Kelly et al., 2019) or uncertainty-based gating in EnsembleDAgger (Menda et al., 2019). In contrast, IPD build a state machine model that dynamically swaps proposer and verifier roles through teacher takeover on rejected student tokens and handback on accepted teacher tokens, and then trains the resulting mixed-source trajectory with a source-split objective.

3 Methodology

3.1 Preliminaries

Let $y \sim \mathcal{D}$ be a prompt from a dataset $\mathcal{D}$, $x = (x_1, \ldots, x_T)$ a response trajectory, and $c_t = (y, x_{<t})$ the prefix context at step t. The student and teacher next-token distributions are denoted as $\pi_\theta(\cdot \mid c_t)$ and $q(\cdot \mid c_t)$, and their complete-response probabilities factorize as $\pi_\theta(x \mid y) = \prod_{t=1}^{|x|} \pi_\theta(x_t \mid c_t)$ and $q(x \mid y) = \prod_{t=1}^{|x|} q(x_t \mid c_t)$.

Off-Policy Distillation. Traditional knowledge distillation trains on prefixes sampled from a distribution μ, which is typically a dataset of fixed reference responses or trajectories sampled from a teacher model (Hinton et al., 2015; Kim & Rush, 2016). Its standard objective is minimizing the forward KL,

$$\mathcal{L}_{\text{off}}(\theta) = \mathbb{E}_{y\sim\mathcal{D}, x\sim\mu(\cdot|y)} \left[\frac{1}{|x|} \sum_{t=1}^{|x|} D_{\text{KL}}(q(\cdot \mid c_t) \,\|\, \pi_\theta(\cdot \mid c_t)) \right]. \tag{1}$$

The fixed prefixes make optimization stable and permit dense supervision over the vocabulary, but they also induce a state-distribution mismatch: at inference time, the student must condition on its own previous tokens, including mistakes that are absent from teacher-generated trajectories.

On-Policy Distillation. Compared with off-policy distillation, on-policy distillation (OPD) evaluates teacher feedback on responses sampled from the student itself (Agarwal et al., 2024; Gu et al., 2024). Under the notations above, its theoretical objective is the sequence-level reverse KL

$$\mathcal{L}_{\text{on}}^{\text{seq}}(\theta) = \mathbb{E}_{y\sim\mathcal{D}} \left[D_{\text{KL}}(\pi_\theta(\cdot \mid y) \,\|\, q(\cdot \mid y)) \right] = \mathbb{E}_{y\sim\mathcal{D}, x\sim\pi_\theta(\cdot|y)} \left[\sum_{t=1}^{|x|} D_{\text{KL}}(\pi_\theta(\cdot \mid c_t) \,\|\, q(\cdot \mid c_t)) \right], \tag{2}$$

The outer expectation samples prefixes from the student-induced state distribution, while each summand is a token-level conditional reverse KL at one such prefix. At a fixed prefix c_t, the exact

token-level conditional reverse KL is computed over the entire vocabulary $\mathcal{V}$ as

$$\ell_t^{\text{full}}(\theta) = D_{\text{KL}}(\pi_\theta(\cdot \mid c_t) \,\|\, q(\cdot \mid c_t)) = \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) \log \frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)}. \tag{3}$$

Optimizing Eq. (3) requires forward and backward passes over all $|\mathcal{V}|$ indices at every position, which is computationally expensive (Li et al., 2026; Oh et al., 2026).

Sampled-Token and Top-K OPD. As a less expensive alternative to the full-vocabulary KL in Eq. (3), sampled-token OPD evaluates only the student-sampled token $x_t \sim \pi_\theta(\cdot \mid c_t)$. The resulting single-sample Monte Carlo estimate of the token-level KL value at the fixed prefix c_t is

$$\widehat{\ell}_t^{\text{sample}} = \log \pi_\theta(x_t \mid c_t) - \log q(x_t \mid c_t). \tag{4}$$

Although Eq. (4) is an unbiased estimate for the full-vocabulary KL, the variance introduced by sampling a single token can make updates noisy and unstable (Oh et al., 2026). Top-K OPD reaches a sweet spot between full-vocabulary and sampled-token OPD. It retains multi-token supervision on a small high-probability support while avoiding a full-vocabulary computation (Li et al., 2026; Fu et al., 2026). Let

$$S_t^\pi = \text{Top}_K(\pi_\theta(\cdot \mid c_t)), \qquad S_t^q = \text{Top}_K(q(\cdot \mid c_t)), \tag{5}$$

denote the student and teacher top-K supports, respectively, where Top_K returns the K highest-probability indices, and denote $S_t \subseteq \mathcal{V}$ as the support for distillation, which can usually be S_t^π, S_t^q, $S_t^\pi \cup S_t^q$, or $S_t^\pi \cap S_t^q$. As the two models generally assign different total probability mass to S_t, each distribution should be normalized by

$$\bar{\pi}_\theta^{S_t}(v \mid c_t) = \frac{\pi_\theta(v \mid c_t)\mathbb{1}[v \in S_t]}{\sum_{u \in S_t} \pi_\theta(u \mid c_t)}, \quad \bar{q}^{S_t}(v \mid c_t) = \frac{q(v \mid c_t)\mathbb{1}[v \in S_t]}{\sum_{u \in S_t} q(u \mid c_t)}. \tag{6}$$

Then, the token-level top-K KL and its rollout-level aggregated loss are given by

$$\ell_t^{\text{top-}K}(\theta; S_t) = D_{\text{KL}}\left(\bar{\pi}_\theta^{S_t}(\cdot \mid c_t) \,\|\, \bar{q}^{S_t}(\cdot \mid c_t)\right) = \sum_{v \in S_t} \bar{\pi}_\theta^{S_t}(v \mid c_t) \log \frac{\bar{\pi}_\theta^{S_t}(v \mid c_t)}{\bar{q}^{S_t}(v \mid c_t)}, \tag{7}$$

$$\mathcal{L}_{\text{on}}^{\text{top-}K}(\theta) = \mathbb{E}_{y \sim \mathcal{D}, x \sim \pi_\theta(\cdot|y)} \left[\sum_{t=1}^{|x|} \ell_t^{\text{top-}K}(\theta; S_t) \right]. \tag{8}$$

Although top-K OPD is a biased approximation to full-vocabulary OPD as it discards probability mass outside S_t, it replaces the high-variance single-token estimate of sampled-token OPD with deterministic multi-token supervision over the selected high-probability support for each fixed prefix c_t, and therefore provides a practical middle ground with denser and more stable supervision than sampled-token OPD (Fu et al., 2026), while at the same time reducing the per-token computation and backward-pass cost of full-vocabulary OPD (Li et al., 2026; Oh et al., 2026).

3.2 Interactive-Policy Distillation

Vanilla OPD samples trajectories from the student, and the teacher feedback is evaluated on states that may be far from where it would actually visit. We name it as *teacher unanchoring*. Figure 2 illustrates the average entropy and reverse KL over position intervals on the student (Qwen3-1.7B-Base) responses for 512 randomly selected DAPO-MATH-17K (Yu et al., 2025) prompts, split by the final answer correctness. The teacher (Qwen3-30B-A3B non-thinking) queried by incorrect trajectories yields high uncertainty, even as the reverse KL decays along the position. We propose Interactive-Policy Distillation (IPD), which formulates generation as a state machine in which the student and teacher alternate as proposer and verifier. to address this challenge by dynamic adaptive teacher intervention during rollout.

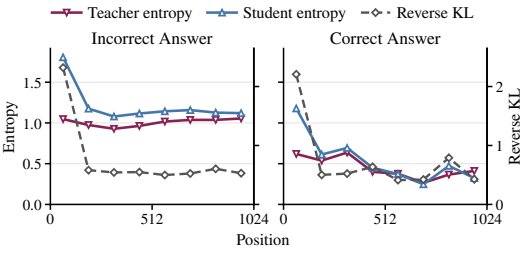


Figure 2: Average entropy and reverse KL over position intervals on the student responses split by the final answer correctness.

3.2.1 Adaptive Teacher Intervention with Bidirectional Propose-and-Verify

Let $m_t \in \{\mathsf{S}, \mathsf{T}\}$ denote the driver model state before generating token at position t, where S and T indicate student driving and teacher driving, respectively, and we initialize every rollout with $m_1 = \mathsf{S}$. Let $z_t \in \{\mathsf{S}, \mathsf{T}\}$ record the source of the token actually emitted at position t. We use top-K coverage as the takeover and handback rules, with S_t^π and S_t^q defined in Eq. (5).

Student driving. When $m_t = \mathsf{S}$, the student proposes $\widetilde{x}_t^\pi \sim \pi_\theta(\cdot \mid c_t)$ and the teacher verifies whether the student-generated token is covered by the teacher's top-K support, *i.e.,* $\widetilde{x}_t^\pi \in S_t^q$. If covered, which indicates that the student's proposal within the teacher's acceptance region, IPD accepts the student proposal, assigns $z_t = \mathsf{S}$, and remains in student driving state. Otherwise, the proposal is discarded, and the teacher samples a replacement $\widetilde{x}_t^q \sim q(\cdot \mid c_t)$. IPD emits this teacher-sampled token with $z_t = \mathsf{T}$, and the rollout control switches to the teacher from position $t + 1$.

Teacher driving. When $m_t = \mathsf{T}$, the teacher proposes $\widetilde{x}_t^q \sim q(\cdot \mid c_t)$ and the student verifies whether the teacher-sampled token satisfies $\widetilde{x}_t^q \in S_t^\pi$. If it is not covered by the student's top-K support, the teacher retains control. Otherwise, as the teacher's proposal is back to the student's acceptance region, which indicates that the student is possible to generate is on its own, it is kept and serves as the handback trigger and the rollout control is returned to student rom position $t + 1$.

The complete per-token transition can be written as

$$(x_t, z_t, m_{t+1}) = \begin{cases} (\widetilde{x}_t^\pi, \mathsf{S}, \mathsf{S}), & m_t = \mathsf{S},\ \widetilde{x}_t^\pi \in S_t^q, \\ (\widetilde{x}_t^q, \mathsf{T}, \mathsf{T}), & m_t = \mathsf{S},\ \widetilde{x}_t^\pi \notin S_t^q, \\ (\widetilde{x}_t^q, \mathsf{T}, \mathsf{S}), & m_t = \mathsf{T},\ \widetilde{x}_t^q \in S_t^\pi, \\ (\widetilde{x}_t^q, \mathsf{T}, \mathsf{T}), & m_t = \mathsf{T},\ \widetilde{x}_t^q \notin S_t^\pi. \end{cases} \tag{9}$$

The state machine iterates until an EOS token is emitted or the maximum rollout length is reached. Eq. (9) is written token by token for clarity. Following Leviathan et al. (2023) and Xu et al. (2025), for a faster rollout, the implementation usually consecutively proposes a chunk of γ tokens and verify them in parallel, instead of token-by-token verification immediately after decoded.

Note that in this paper we use the top-K coverage as the takeover and handback rules. As a unified framework, IPD is also compatible with other rules, and the takeover and handback rules can also be asymmetric. For example, we can instead use a threshold on the entropy of the student or the teacher distributions to determine whether to take over or hand back control. SKD (Xu et al., 2025) can be included as a special case of IPD with a top-K coverage takeover rule and a fixed-one-token handback rule, and in our experiments, SKD is also implemented under our customized fused engine for more efficient rollout.

3.2.2 Source-Split KL Objective

The state machine model formulated by Eq. (9) produces a response trajectory x together consisting of student-generated and teacher-generated tokens with recorded source labels $z = (z_1, \ldots, z_T)$. Denote $I_{on}(z) = \{t : z_t = \mathsf{S}\}$ and $I_{off}(z) = \{t : z_t = \mathsf{T}\}$ as the on-policy (student-generated) and off-policy (teacher-generated) position sets, respectively. Since the trajectory x is a mixure of student and teacher tokens, it is natural provide the two groups of positions with different supervisions. We therefore design a source-split KL objective that applies different KL divergences to student-generated and teacher-generated positions.

Specifically, the token-level KL divergences applied to student-generated and teacher-generated positions are respectively denoted as

$$\ell_t^{\mathrm{on}}(\theta) = D_{\mathrm{KL}}\Big(\bar{\pi}_\theta^{S_t}(\cdot \mid c_t) \,\|\, \bar{q}^{S_t}(\cdot \mid c_t)\Big), \quad \ell_t^{\mathrm{off}}(\theta) = D_{\mathrm{KL}}\Big(\bar{q}^{S_t}(\cdot \mid c_t) \,\|\, \bar{\pi}_\theta^{S_t}(\cdot \mid c_t)\Big). \tag{10}$$

Student-generated positions therefore receive the token-level reverse KL used by on-policy distillation, while teacher-generated tokens receive the token-level forward KL used by off-policy distillation. Both losses are evaluated at the realized mixed-source prefix c_t, regardless of which model generated earlier tokens. Let $\rho_\theta^{\mathrm{IPD}}(x, z \mid y)$ denote the joint distribution over emitted responses and source labels induced by the state machine in Eq. (9). The final length-normalized objective is

minimizing the following source-split KL

$$\mathcal{L}_{\mathrm{IPD}}(\theta) = \mathbb{E}_{y\sim\mathcal{D},\,(x,z)\sim\rho_\theta^{\mathrm{IPD}}(\cdot,\cdot|y)}\left[\frac{1}{|x|}\left(\sum_{t\in I_{on}(z)}\ell_t^{\mathrm{on}}(\theta)+\sum_{t\in I_{off}(z)}\ell_t^{\mathrm{off}}(\theta)\right)\right]. \tag{11}$$

With the bidirectional propose-and-verify mechanism and the source-split KL, IPD naturally becomes a unified bridge between on-policy and off-policy distillations. Specifically, never transferring control to the teacher yields length-normalized on-policy distillation with reverse KL supervision on pure student-generated trajectories, while always rejecting the first student proposal and never handing control back yields off-policy distillation with forward KL supervision on pure teacher-generated trajectories.

3.2.3 Fused Inference Engine for Efficient Dual-Model Scheduling

To integrate IPD with verl (Sheng et al., 2025), built on vLLM (Kwon et al., 2023), we design a fused inference engine for more efficient dual-model scheduling. A naive implementation that hosts two models on separate vLLM serving instances will have to re-encode the whole prefix for every verification round, as an off-the-shelf vLLM engine bypasses its prefix cache for requests that return prompt log-probabilities and cannot append externally generated tokens to the KV cache of a live request. Besides, there are also communication overheads between two engines. As shown in Figure 3, with Qwen3-4B-Base (non-thinking) as the teacher and Qwen3-1.7B-Base as the student, executing the IPD rollout with bidirectional propose-and-verify on the same 512 randomly selected DAPO-MATH-17K prompts under identical settings and memory budget, fused engine is 21.4× faster than the naive implementation on the initial checkpoint and 71.4× faster on the final one.

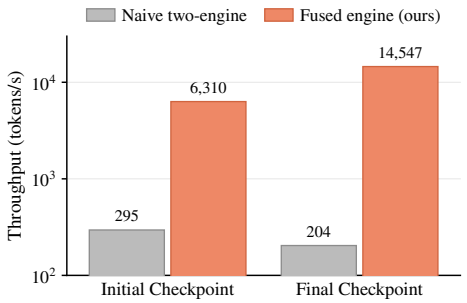


Figure 3: IPD rollout throughput of fused engine and naive two-engine implementation.

Specifically, as illustrated in Figure 4, our fused engine co-hosts the student and teacher in one serving instance, and each active request maintains a verified prefix, a chunk of unverified draft tokens, the identity of its current driver model, and separate student and teacher KV-cache states. For each request, the two models serve as symmetrical roles. Each model maintains its own cache and may act as either the proposer or the verifier, with the verifier always being the counterpart of the current driver model. One fused engine step consists of a verification, a verdict, and a proposal stage. In the verification stage, requests are grouped by their verifier, and each verifier evaluates all unverified draft tokens of its sub-batch in parallel through a single prefill pass. Only the newly appended tokens require computation as long as the cache of the verified prefix remains resident. If vLLM preempts a request under resource contention and evicts its cached blocks, the affected model, whether verifier or proposer, will instead recompute the KV states of the whole prefix. In the verdict stage, the engine determines the acceptance or rejection of each draft, detects takeover and handback transitions, updates the driver model, and aligns both KV caches with the resulting verified prefix. If no transition occurs, all drafted tokens are kept and both caches are preserved. If a student token triggers takeover, the token is rejected and will be regenerated by the teacher, so both caches are rolled back to the preceding token. If a teacher token triggers handback, the token is retained, so both caches are rolled back only to it and the subsequent draft tokens are discarded. Either way, the two caches end at the same accepted-prefix boundary. In the proposal stage, requests are regrouped by their updated driver model, and each proposer autoregressively decodes a new chunk of γ draft tokens for its sub-batch while extending only its own KV cache, leaving the counterpart model to catch up through the prefill of the next verification stage. Consequently, apart from draft tokens discarded by a transition and cache blocks evicted by preemption, every token is decoded once by its proposer and prefilled once by its verifier. The engine repeats these stages until every request has emitted an accepted EOS token or reached the maximum generation length. A student-generated EOS token terminates the request only if it passes the teacher's takeover test and otherwise triggers a teacher takeover at that position, whereas a teacher-generated EOS token is accepted directly.

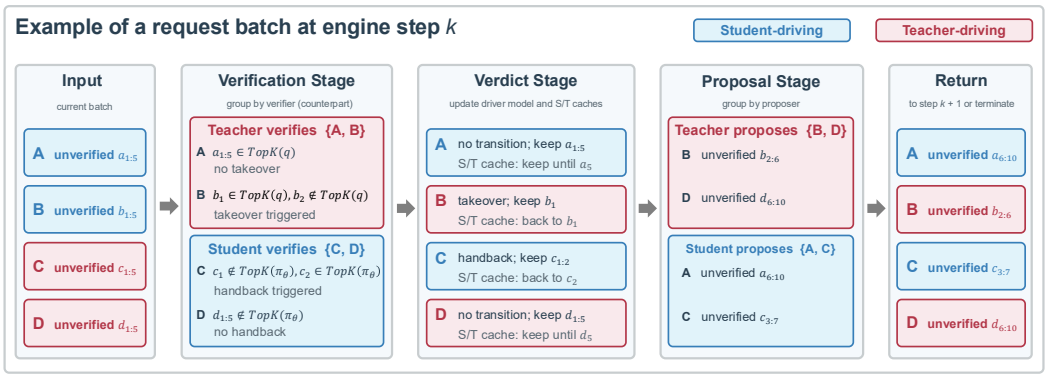


Figure 4: Example of one IPD fused engine step for a request batch. The engine distributes requests to their respective verifier models for verification, collects verification results, updates their driver models, manages their KV cache on each model, re-distributes requests to their respective proposer models for new unverified draft tokens, and return the updated requests for next step or termination.

4 Experiments

4.1 Experimental Settings

We train four Qwen3 (Yang et al., 2025) teacher-student pairs on DAPO-Math-17K (Yu et al., 2025) for one epoch, with thinking mode disabled for all teachers. We set $K = 16$: all methods apply distillation over the union support $S_t = S_t^\pi \cup S_t^q$, and IPD and SKD use top-K coverage for takeover, with IPD also using it for handback. We use the codebase from Li et al. (2026) for OPD, and additionally build our fused inference engine and implement SKD and IPD training pipelines on top of that. Both IPD and SKD use a proposal chunk length of $\gamma = 5$. We evaluate the final checkpoints on six mathematical reasoning benchmarks and report mean@8 and best@8 accuracy. Appendix A provides more detailed model, training, hardware, and evaluation configurations.

Table 1: Comparison of different distillation methods across various student and teacher model combinations. We report mean@8 and best@8 accuracy (%) on different math reasoning benchmarks, as well as the benchmark-wise average.

Teacher/ Student	Method	MATH-500		Minerva		Olympiad		AMC23		AIME24		AIME25		**Avg.**	
		mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8
Qwen3-4B/ Qwen3-1.7B-Base	OPD	66.68	84.06	26.42	44.04	31.00	48.45	35.69	**57.29**	7.92	**21.77**	2.92	8.70	28.44	44.05
	SKD	66.70	84.50	27.94	43.69	29.75	49.52	35.09	52.90	8.75	17.41	4.58	10.73	28.80	43.13
	IPD (ours)	**69.28**	**84.59**	**29.60**	**46.26**	**31.80**	**49.63**	**36.14**	57.25	**9.17**	21.69	**6.67**	**15.06**	**30.44**	**45.75**
Qwen3-30B-A3B/ Qwen3-1.7B-Base	OPD	63.35	83.54	23.25	41.43	28.14	46.88	32.68	54.39	10.42	18.84	**5.83**	13.12	27.28	43.03
	SKD	67.38	84.34	26.70	44.69	30.39	49.73	33.89	54.67	8.75	22.36	**5.83**	**14.92**	28.82	45.12
	IPD (ours)	**69.35**	**84.61**	**29.60**	**46.44**	**32.56**	**51.29**	**34.79**	**59.60**	**11.67**	**22.79**	5.42	13.10	**30.56**	**46.31**
Qwen3-8B/ Qwen3-4B-Base	OPD	79.63	90.83	36.53	52.22	44.80	60.49	44.28	66.31	15.83	25.40	13.75	29.09	39.14	54.06
	SKD	80.43	91.49	**37.91**	51.15	44.71	61.32	**51.51**	**72.01**	15.42	24.76	15.00	**29.68**	40.83	55.07
	IPD (ours)	**81.45**	**92.21**	37.87	**52.66**	**46.43**	**62.19**	48.34	67.83	**17.08**	**28.86**	**17.08**	28.66	**41.38**	**55.40**
Qwen3-30B-A3B/ Qwen3-4B-Base	OPD	81.38	90.00	39.11	53.41	46.66	61.18	48.80	67.88	15.42	24.78	12.92	26.36	40.71	53.93
	SKD	81.03	91.96	40.40	55.39	47.42	**63.95**	**49.85**	**72.82**	16.25	24.85	**16.67**	**27.60**	41.93	56.09
	IPD (ours)	**82.28**	**92.40**	**41.18**	**55.52**	**47.66**	63.45	49.55	70.64	**19.58**	**28.39**	13.75	26.22	**42.33**	**56.10**

4.2 Main Results

We compare IPD with OPD (Lu & Lab, 2025; Li et al., 2026) and SKD (Xu et al., 2025) on multiple student and teacher model combinations across different sizes. Thinking mode is disabled for all non-base teachers. In Table 1, our algorithm demonstrates consistent improvements over the baselines across all benchmarks and model combinations. Specifically, when distilling Qwen3-4B to Qwen3-1.7B-Base, IPD achieves an average mean@8 accuracy of 30.44 across all benchmarks, outperforming OPD and SKD by $+2.00$ and $+1.64$, respectively. Besides, IPD also allows the student to robustly learn from a larger teacher. According to the discussion in Li et al. (2026), a larger teacher cannot always distill a better student, and when the teacher-student reasoning gap is too large, the distillation performance can be suboptimal. In our experiments, when distilling Qwen3-30B-A3B to Qwen3-1.7B-Base, we also find this phenomenon that the distilled student achieves an average mean@8 accuracy of 27.28, which is lower than the 28.44 achieved by distilling a Qwen3-

4B teacher. Meanwhile, with the bidirectional propose-and-verify mechanism, IPD can effectively bridge the reasoning gap between the teacher and student, achieving an average mean@8 accuracy of 30.56, which is 3.28 higher than OPD and 1.74 higher than SKD.

Besides, IPD is also a more data-efficient distillation algorithm. As shown in Figure 5, IPD shows a much faster convergence than other baselines. Specifically, trained with only fewer than 1/4 training examples for 60 training steps, IPD can outperform OPD and matches SKD trained for a whole epoch with 279 training steps. Besides, IPD shows a stably increasing top-K overlap ratio (defined by $|S_t^\pi \cap S_t^q|/K$), indicating a healthy training under large a teacher-student gap (Li et al., 2026).

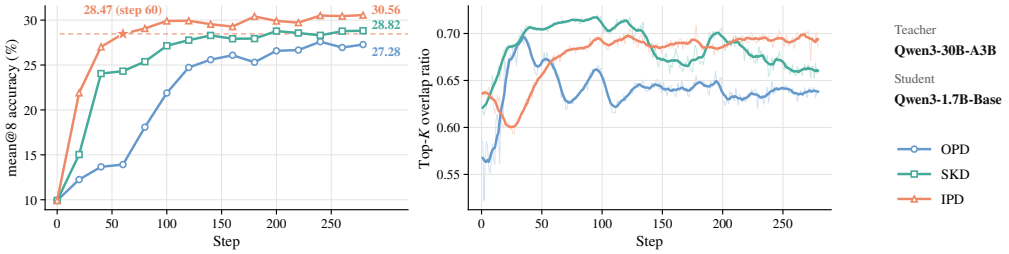


Figure 5: The mean@8 accuracy over math reasoning benchmarks and top-K overlap ratio during training on DAPO-MATH-17K. In addition to better final checkpoint performance, IPD also shows a much higher data efficiency than other baselines.

4.3 Ablation Study

KL direction. To investigate the impact of different KL directions for on-policy and off-policy positions, and the resonability of our source-split KL design, we conduct an ablation study on the IPD KL objective. We compare the following configurations: (i) Forward KL for all positions (Pure Forward KL), (ii) Reverse KL for all positions (Pure Reverse KL), (iii) Forward KL for on-policy and Reverse KL for off-policy positions (Inverse-IPD-Split KL), and (iv) Reverse KL for on-policy and Forward KL for off-policy positions (IPD-Split KL, default). As shown in Figure 6, we observe that the IPD Default configuration, which uses Reverse KL for on-policy positions and Forward KL for off-policy positions, achieves the best mean@8 accuracy. Meanwhile, Pure Reverse KL turns out to be the runner-up, with the second-best mean@8 accuracy and a 0.12 higher best@8 accuracy than the default source-split KL. The teacher intervention ratio, which is calculated by the proportion of teacher-generated tokens among all tokens, gradually decreases during training, indicating that the student is learning to generate more tokens that are covered by the teacher's top-K support, and leaning more towards on-policy paradigm.

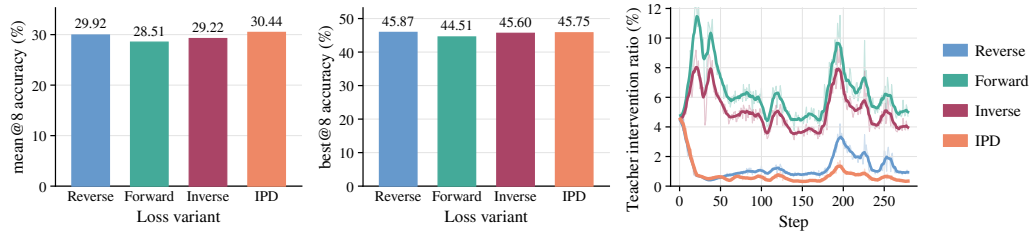


Figure 6: The mean@8 and best@8 average accuracy over benchmarks and the teacher intervention ratio with different losses during IPD training.

Coverage width. In our main experiments, we set $K = 16$ for the top-K coverage takeover and handback rules in the IPD framework. Under a smaller K, there will be more teacher takeovers and less frequent teacher handbacks, which leans more towards off-policy training, results in a higher teacher invervention ratio during the rollout stage, and requires more computational resources. While under a larger K, the teacher will invervene less, leaning more towards an on-policy training. To study the effect of different coverage widths, we further conduct experiments with $K = 4$, $K = 8$, and $K = 32$. As shown in Figure 7, IPD under $K = 4$ achieves the best performance with 30.46 mean@8 and 45.80 best@8. The default $K = 16$ appears to be a close runner-up with only -0.02 mean@8 and -0.05 best@8 gap, while getting much lower teacher intervention ratio, which provides a good balance between performance and computational efficiency.

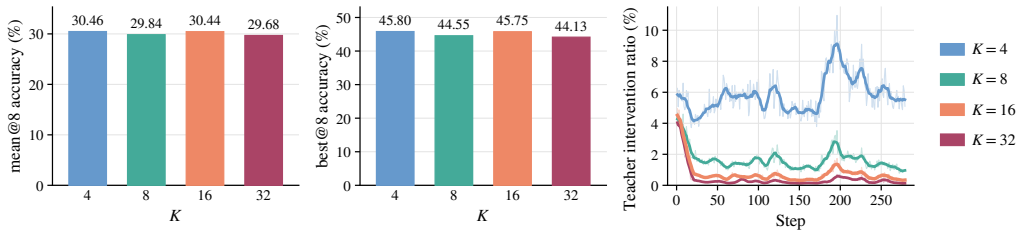


Figure 7: The mean@8 and best@8 average accuracy over benchmarks and the teacher intervention ratio with different coverage widths K during IPD training. A smaller K leads to more frequent teacher takeovers and less frequent teacher handbacks, resulting in a higher intervention ratio.

4.4 Robustness on Different Training Data

We further explore the robustness of IPD and other distillation algorithms under different training data. In addition to DAPO-MATH-17K, we select examples from DeepMath-103K (He et al., 2026) with difficulty levels greater than or equal to 8 and get 17,085 training examples, denoted as DeepMath-GE8-17K. As shown in Figure 8, we compare different algorithms on these two training datasets. Specifically, we find that algorithms with teacher intervention during rollout, including IPD and SKD, demonstrate a good resistance to the change of training data. Meanwhile, OPD shows a non-negligible mean@8 accuracy drop, which indicates that OPD is more sensitive to the training data. On the Qwen3-30B-A3B to Qwen3-4B-Base group, the mean@8 accuracy of OPD even gets a significant drop of -6.75. This indicates that without teacher intervention, OPD is more sensitive to the training data, and teacher intervention applied by IPD and SKD can alleviate this sensitivity, which requires less effort on data preparation and makes them more practical paradigms.

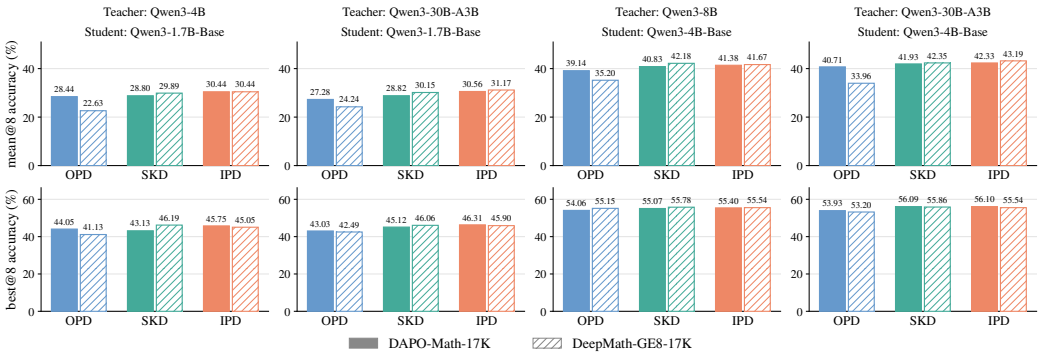


Figure 8: Comparison of different distillation algorithms on DAPO-MATH-17K and DeepMath-GE8-17K. Intervention-based distillation algorithms including IPD and SKD demonstrate consistent robustness, while OPD apprears to be more sensitive to the change of training data.

5 Conclusion and Discussion

In this paper, we propose Interactive-Policy Distillation (IPD), which addresses the teacher unanchoring issue and improve the distillation performance through bidirectional takeover and handback during rollout, together with source-split supervision based on token provenance. This design dynamically corrects trajectories that leave the teacher's acceptance region, and also provides a unified bridge between on-policy and off-policy distillations. Across diverse teacher-student pairs, IPD demonstrates better distillation performance and higher data efficiency, as well as better robustness to training data. We also design a fused inference engine to support our algorithm with efficient dual-model scheduling during the rollout stage.

Currently, like most works in the OPD field, we only consider same-tokenization distillation here. Extending our method to cross-tokenization scenarios is an important problem. Besides, more different rules beyond top-K coverage can be further explored. On the infrastructure side, we only consider synchrnous scheduling for verification and proposal stages. Optimizations like asynchronous scheduling can further improve the throughput of dual-model rollout. We leave these directions for future exploration.

AI USE STATEMENT

In this work, we have used generative AI tools for training code implementation and optimization, where all the detailed designs were proposed by us. We have not used generative AI tools for fully automated codebase building. Additionally, in our manuscript, we have used generative AI tools to refine writing, search related works, and organize experimental results to tables and figures. We have not used generative AI tools to propose research idea of this paper. The research idea is originally proposed by us.

ETHICS STATEMENT

There is no significant ethical concern in this work. The datasets and models used in this work are all publicly available.

REPRODUCIBILITY STATEMENT

We have provided a clear description of our proposed method, the necessary experimental settings, and the information of the envirionment where we implement all the experiments. We will also open-source our codebase in the future.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Anthropic. Introducing Claude Fable 5.1 and Claude Mythos 5.1, 2026. URL `https://www.anthropic.com/claude-fable-and-mythos-5-1`.

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.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 700–710, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. In *Third Conference on Language Modeling*, 2026.

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: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025.

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.

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

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 14852–14882, 2023.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026.

Michael Kelly, Chelsea Sidrane, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8077–8083. IEEE, 2019.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models. In *International Conference on Machine Learning*, pp. 24872–24895. PMLR, 2024.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Forty-second International Conference on Machine Learning*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pp. 19274–19286. PMLR, 2023.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan ang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25366–25394, 2025.

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

Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1773–1781, 2023.

Kunal Menda, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5041–5048. IEEE, 2019.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. KL for a KL: On-policy distillation with control variate baseline. In *3rd AI for Math Workshop: Toward Self-Evolving Scientific Agents*, 2026.

OpenAI. GPT-6 Astra: A new generation of intelligence, 2026. URL https://openai.com/index/gpt-6-astra/.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman. Approximating KL divergence. http://joschu.net/blog/kl-approx.html, 2020.

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.

Makoto Shing, Kou Misaki, Han Bao, Sho Yokoi, and Takuya Akiba. TAID: Temporally adaptive interpolated distillation for efficient knowledge transfer in language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

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.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, volume 2025, pp. 64616–64646, 2025.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2025.

Z.ai. GLM-5.3: Frontier Coding with Emergent Cyber Capabilities, 2026. URL https://z.ai/blog/glm-5.3.

A Detailed Experimental Settings

Data and models. We conduct our main experiments on DAPO-Math-17K (Yu et al., 2025) with four teacher–student pairs: Qwen3-4B / Qwen3-1.7B-Base, Qwen3-30B-A3B / Qwen3-1.7B-Base, Qwen3-8B / Qwen3-4B-Base, and Qwen3-30B-A3B / Qwen3-4B-Base. Thinking mode is disabled for all teacher models.

Training configuration. We use the codebase from Li et al. (2026) for OPD, which is under the verl (Sheng et al., 2025) framework. On top of their codebase, we build our fused inference engine and implement SKD and IPD training pipelines. Specifically, as a special case of IPD, SKD also uses our fused inference engine for rollout. Following Li et al. (2026), unless otherwise specified, we train each model for one epoch (279 steps) with a learning rate of 1×10^{-6}. The maximum rollout sequence length is 8192, comprising a maximum prompt length of 1024 and a maximum generation length of 7168. Each prompt produces four training rollouts sampled with temperature 1.0 and top-$p = 1.0$. All experiments are conducted on two compute nodes, each equipped with 4 NVIDIA H100 SXM 80GB GPUs.

Distillation configuration. All methods use top-K distillation with $K = 16$. Each KL term is computed after separately renormalizing the student and teacher distributions over the union support $S_t = S_t^\pi \cup S_t^q$, as defined in Eq. (6). IPD and SKD both use $K = 16$ for the takeover coverage rule and a proposal length of $\gamma = 5$. IPD additionally applies the symmetric top-K coverage rule for handback.

Evaluation. We evaluate the final checkpoint of every run on MATH-500, Minerva, Olympiad-Bench, AMC23, AIME24, and AIME25. For each problem, we independently sample eight responses with temperature 0.7 and top-$p = 0.95$. Same as training, we use a maximum generation length of 7168. We report mean@8 and best@8 accuracy for each benchmark and their benchmark-wise averages. Evaluation uses the same maximum prompt and generation lengths as training.

B Further Discussions

B.1 Sampled-Token OPD

Since the sampled token follows $x_t \sim \pi_\theta(\cdot \mid c_t)$, taking the expectation of Eq. (4) over x_t recovers the full-vocabulary reverse KL in Eq. (3),

$$\mathbb{E}_{x_t \sim \pi_\theta(\cdot|c_t)}\left[\widehat{\ell}_t^{\text{sample}}\right] = \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) \log \frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)} = \ell_t^{\text{full}}. \tag{12}$$

According to Schulman (2020), $\widehat{\ell}_t^{\text{sample}}$ is therefore the k_1 estimator of the reverse KL, which is unbiased even though its individual realizations can be negative while $\ell_t^{\text{full}} \geq 0$ is guaranteed. Given the identity $\nabla_\theta \pi_\theta(v \mid c_t) = \pi_\theta(v \mid c_t) \nabla_\theta \log \pi_\theta(v \mid c_t)$, the product rule applied to Eq. (3) gives

$$\begin{aligned}
\nabla_\theta \ell_t^{\text{full}} &= \sum_{v \in \mathcal{V}} \nabla_\theta \pi_\theta(v \mid c_t)\ \log \frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)} + \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) \nabla_\theta \log \frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)} \\
&= \sum_{v \in \mathcal{V}} \nabla_\theta \pi_\theta(v \mid c_t)\ \log \frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)} + \sum_{v \in \mathcal{V}} \nabla_\theta \pi_\theta(v \mid c_t) \\
&= \sum_{v \in \mathcal{V}} \nabla_\theta \pi_\theta(v \mid c_t)\ \log \frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)} + \nabla_\theta \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) \\
&= \sum_{v \in \mathcal{V}} \pi_\theta(v \mid c_t) \log \frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)} \nabla_\theta \log \pi_\theta(v \mid c_t) + \nabla_\theta 1 \\
&= \mathbb{E}_{x_t \sim \pi_\theta(\cdot|c_t)}\left[\widehat{\ell}_t^{\text{sample}} \nabla_\theta \log \pi_\theta(x_t \mid c_t)\right].
\end{aligned} \tag{13}$$

We denote the gradient estimator as

$$\widehat{g}_t = \widehat{\ell}_t^{\text{sample}} \nabla_\theta \log \pi_\theta(x_t \mid c_t). \tag{14}$$

Under the context of gradient ascent in reinforcement learning, $-\widehat{g}_t$ is the policy gradient obtained when OPD is implemented as maximizing

$$\mathcal{J}_t = \mathbb{E}_{x_t \sim \pi_\theta(\cdot|c_t)}\Big[\mathrm{sg}\big[-\widehat{\ell}_t^{\mathrm{sample}}\big]\Big], \tag{15}$$

where the per-token reward is the stop-gradient operated $-\widehat{\ell}_t^{\mathrm{sample}} = \log q(x_t \mid c_t) - \log \pi_\theta(x_t \mid c_t)$ (Lu & Lab, 2025; Oh et al., 2026).

Given a prefix c_t, variances of the sampled-token loss and gradient are

$$\begin{aligned}
\mathrm{Var}\big[\widehat{\ell}_t^{\mathrm{sample}} \mid c_t\big] &= \sum_{v\in\mathcal{V}} \pi_\theta(v \mid c_t)\left(\log\frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)}\right)^2 - \big(\ell_t^{\mathrm{full}}\big)^2 \\
&= \sum_{v\in\mathcal{V}} \pi_\theta(v \mid c_t)\left(\log\frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)} - \ell_t^{\mathrm{full}}\right)^2, \qquad (16) \\
\mathrm{Var}\big[\widehat{g}_t \mid c_t\big] &= \mathbb{E}\big[\|\widehat{g}_t - \nabla_\theta \ell_t^{\mathrm{full}}\|^2 \mid c_t\big] \\
&= \mathbb{E}\big[(\widehat{\ell}_t^{\mathrm{sample}})^2 \|\nabla_\theta \log \pi_\theta(x_t \mid c_t)\|^2 \mid c_t\big] - \|\nabla_\theta \ell_t^{\mathrm{full}}\|^2 \\
&= \sum_{v\in\mathcal{V}} \pi_\theta(v \mid c_t)\left\|\log\frac{\pi_\theta(v \mid c_t)}{q(v \mid c_t)}\nabla_\theta \log \pi_\theta(v \mid c_t) - \nabla_\theta \ell_t^{\mathrm{full}}\right\|^2. \qquad (17)
\end{aligned}$$

Eq. (16) and Eq. (17) reveal a common source of variance in sampled-token OPD. When the teacher and student disagree strongly, the student may assign much higher probability to tokens with $q(v \mid c_t) \ll \pi_\theta(v \mid c_t)$, yielding much larger log-ratios than on tokens where the models agree (Fu et al., 2026). Sampling among these tokens makes the loss and gradient fluctuate, leading to noisy and potentially unstable updates (Oh et al., 2026).

B.2 Distillation Support

According to the GitHub issue discussion from the official repository of Li et al. (2026), different choices of support (including S_t^π, S_t^q, $S_t^\pi \cup S_t^q$, and $S_t^\pi \cap S_t^q$) do not bring significant differences for same-size distillation. From our observation, for different-size distillation, using the union of the two top-K sets $S_t^\pi \cup S_t^q$ as the support for top-K OPD is slightly better than using others. Unless specified, we use the union of the two top-K sets by default in this paper.

B.3 Loss Aggregation

Following GKD (Agarwal et al., 2024) and SKD (Xu et al., 2025), Eq. (11) uses a sequence-mean-token-mean aggregation, which assigns equal weight to each response and prevents longer reasoning traces from dominating the minibatch objective. Another aggregation commonly used in OPD implementations is the global token mean, which assigns equal weight to every valid token and consequently weights each rollout in proportion to its length. This aggregation choice is orthogonal to the rollout policy, the token source, and the direction of the source-split KL. From our practice, global token-mean aggregation and sequence-mean-token-mean aggregation produce comparable performances, while the latter yields slightly more stable optimization. We therefore use sequence-mean-token-mean aggregation by default in this paper.

B.4 Student Models

In this paper, we use Qwen3 Base models as student for all the experiments. Non-base Qwen3 models are trained to switch between thinking mode and non-thinking mode within single model, and this switch is contorlled by special ⟨think⟩ and ⟨/think⟩ tokens. Specifically, when disabling thinking mode, the tokenizer will prepend empty a ⟨think⟩⟨/think⟩ block to the prompt before decoding. If using non-base Qwen3 models as student, it can be difficult to distinguish whether the student performance gain is from successfully aligning with the teacher, or it just somehow manages to bypass the template-controlled thinking mode switch and silently activates its thinking capability obtained from previous training. Therefore, if using non-base Qwen3 models with thinking mode

disabled as teacher, we think using Qwen3 Base models as student is a cleaner setting for distillation experiments. Another clean setting is to use non-base Qwen3 models with thinking mode enabled as both teacher and student, but this setting would require a much longer response length and consume more computational resources. We leave this setting for future exploration.

C Additional Experimental Results

C.1 Additional Main Results

In addition to the main results in Table 1, we also report the mean@8 accuracy and top-K overlap ratio curves during training, as shown in Figure 9.

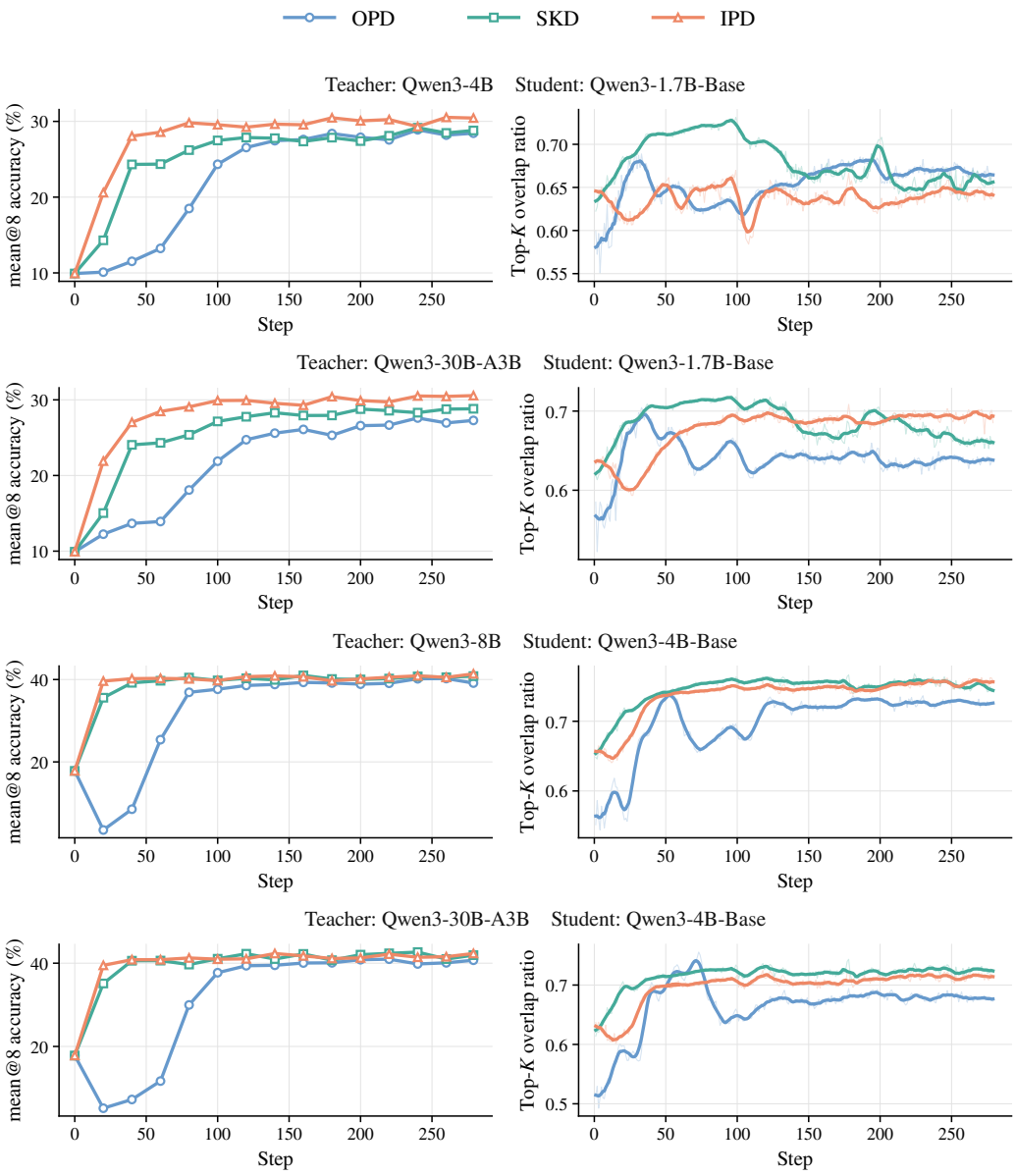


Figure 9: The mean@8 accuracy and top-K overlap ratio curve for four groups of models.

C.2 Additional Ablation Results

KL direction. Table 2 shows that the default source-split KL obtains the highest benchmark-averaged mean@8 accuracy (30.44), while pure reverse KL has a slightly higher best@8 (45.87) than default (45.75).

Table 2: Benchmark-wise mean@8 and best@8 accuracy for the KL-direction ablation.

	MATH-500		Minerva		Olympiad		AMC23		AIME24		AIME25		**Avg.**	
Loss Variant	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8
Pure Reverse KL	68.23	84.05	29.18	46.21	**32.13**	49.70	35.39	56.46	**10.42**	**27.24**	4.17	11.56	29.92	**45.87**
Pure Forward KL	66.80	83.64	28.77	45.15	30.46	49.69	34.19	**58.58**	6.67	18.41	4.17	11.60	28.51	44.51
Inverse-IPD-Split KL	67.35	**84.84**	27.16	44.06	31.37	**50.31**	33.58	56.63	9.17	22.66	**6.67**	**15.09**	29.22	45.60
IPD-Split KL (default)	**69.28**	84.59	**29.60**	**46.26**	31.80	49.63	**36.14**	57.25	9.17	21.69	**6.67**	15.06	**30.44**	45.75

Coverage width. Table 3 shows that $K = 4$ yields the highest average mean@8 and best@8, but only slightly exceeds the default $K = 16$ by $+0.02$ and $+0.05$, respectively.

Table 3: Benchmark-wise mean@8 and best@8 accuracy for the top-K coverage ablation.

	MATH-500		Minerva		Olympiad		AMC23		AIME24		AIME25		**Avg.**	
Coverage width	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8
$K = 4$	68.30	**85.36**	28.49	43.59	31.97	49.76	35.24	53.75	**10.00**	**24.21**	**8.75**	**18.17**	**30.46**	**45.80**
$K = 8$	69.20	84.94	28.95	45.00	32.45	**50.97**	33.43	54.60	8.75	17.68	6.25	14.10	29.84	44.55
$K = 16$ **(default)**	**69.28**	84.59	**29.60**	**46.26**	31.80	49.63	**36.14**	**57.25**	9.17	21.69	6.67	15.06	30.44	45.75
$K = 32$	68.03	83.81	28.77	44.66	**32.50**	50.50	34.64	53.96	8.75	17.58	5.42	14.30	29.68	44.13

D Same-origin Distillation

In addition to different models from the Qwen3 family, we also conduct a same-origin distillation experiment with DeepSeek-R1-Distill-Qwen-1.5B as the student and JustRL-DeepSeek-1.5B as the teacher. Specifically, JustRL-DeepSeek-1.5B is the further post-trained version of DeepSeek-R1-Distill-Qwen-1.5B via simple GRPO recipe (He et al., 2025). The results are shown in Table 4 and Figure 10. We find that intervention methods including IPD and SKD do not bring as statistically significant improvement over OPD compared with our main experiments with different-size Qwen3 models. From Figure 10, we can observe that the mean@8 accuracy and top-K overlap ratio curves of three methods are very close to each other's, which is expected as the student and teacher are same-origin models. The same-origin experiment indicates that OPD can already performs a good distillation performance when the reasoning gap between the student and teacher are not significant, which is consistent with the discussion from Li et al. (2026), and the main advantage of IPD remains in large-gap distillation scenarios.

Table 4: Same-origin distillation experiment with DeepSeek-R1-Distill-Qwen-1.5B as the student and JustRL-1.5B as the teacher on DAPO-Math-17K-Processed. Intervention methods including IPD and SKD do not bring as statistically significant improvement over OPD compared with our main experiments with different-size Qwen3 models.

Teacher/ Student	Method	MATH-500		Minerva		Olympiad		AMC23		AIME24		AIME25		**Avg.**	
		mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8	mean@8	best@8
JustRL-1.5B/ DeepSeek-R1-Distill-Qwen-1.5B	OPD	85.15	92.79	**38.01**	49.41	47.29	60.63	**64.16**	**82.92**	27.50	**52.85**	20.83	30.59	47.16	**61.53**
	SKD	85.55	**93.36**	37.13	47.80	**48.74**	**61.70**	63.55	79.67	27.92	50.43	**23.33**	**32.58**	47.70	60.92
	IPD (ours)	**85.75**	93.29	37.78	**49.99**	48.11	61.36	62.80	79.53	**30.42**	47.13	22.92	29.80	**47.96**	60.18

E Distillation on Limited Trainig Examples

We also run distillation trainings on limited training examples to evaluate the impact of data scarcity on the distillation process. Specifically, we randomly select 1,024 training examples from DAPO-MATH-17K and conduct distillation experiments using Qwen3-4B as teacher and Qwen3-1.7B-Base as student. The student model is trained for 15 epochs (with 240 steps in total), and the other training configurations remains the same as the main experiments. The results are shown in Figure 11. We find that IPD can still outperform OPD and SKD and shows a higher convergence under limited training data. Another interesting observation is that, even if the training examples only count for

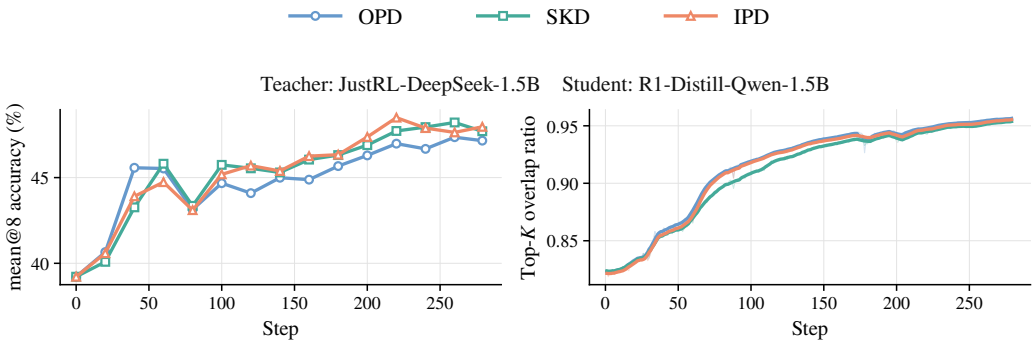


Figure 10: The mean@8 accuracy and top-K overlap ratio curve for same-origin distillation experiment with DeepSeek-R1-Distill-Qwen-1.5B as the student and JustRL-1.5B as the teacher.

about 1/17 of the original training data, all the three methods can matches the performance of the student model trained on the full training dataset. The limited training data experiment indicates that although with limited unique training prompts, as long as they can be replayed to generate diverse rollouts, the distillation can still be effective.

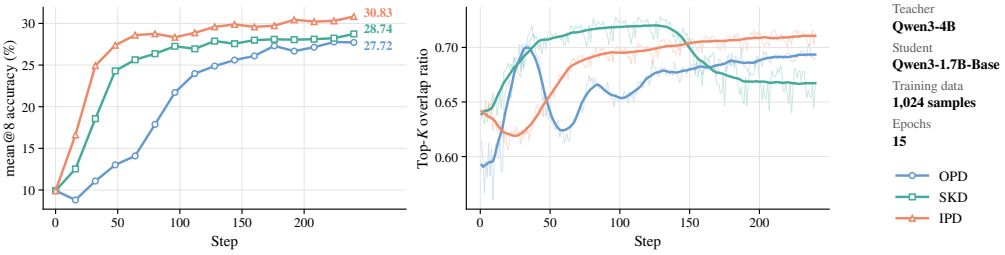


Figure 11: The mean@8 accuracy and top-K overlap ratio curve for distillation on 1,024 randomly selected DAPO-MATH-17K training examples. As long as the limited unique training prompts can be replayed to generate diverse rollouts, the distillation can still be effective.