LIGHTNING OPD: EFFICIENT POST-TRAINING FOR LARGE REASONING MODELS WITH OFFLINE ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) has emerged as an efficient post-training paradigm for large language models. However, standard OPD requires a live teacher inference server throughout training, resulting in substantial infrastructure overhead. In this work, we investigate whether on-policy distillation can be performed offline. A natural approach is to precompute teacher log-probabilities once over SFT rollouts and reuse them during training. In practice, however, this offline variant fails to reliably match the performance of standard OPD. To understand this discrepancy, we identify an important design choice, which we term *teacher consistency*. This choice uses the same teacher model for both supervised fine-tuning and OPD. Across the evaluated teacher combinations, consistent pairings achieve higher distillation scores, with a larger difference for offline than online OPD. Building on this insight, we propose **Lightning OPD**, an offline on-policy distillation framework that precomputes teacher log-probabilities over student rollouts and adopts teacher consistency as a practical design guideline. This design eliminates the need for a live teacher server entirely. Our theoretical analysis bounds the online–offline gap under teacher consistency and shows that teacher mismatch can lead to a non-vanishing gap. Extensive experiments on mathematical reasoning and code generation demonstrate that Lightning OPD achieves competitive performance with significantly improved efficiency. Starting from an Qwen3-8B-SFT model, Lightning OPD reaches **69.9% on AIME 2024** in just **30 GPU hours**, achieving a **4.0**$\times$ reduction in distillation GPU cost over standard OPD and substantially lowering the barrier to entry for academic research on LLM post-training.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable progress across tasks such as mathematical reasoning, code generation, and multi-step agent planning (Guo et al., 2025; Kimi Team et al., 2026; NVIDIA et al., 2025; LLM-Core Xiaomi, 2026; Liu et al., 2024; Kimi Team, 2025; Wei et al., 2022; Yang et al., 2024). This success is underpinned by carefully designed post-training pipelines (Ouyang et al., 2022), which typically consist of supervised fine-tuning (SFT) on high-quality data (Guha et al., 2026), followed by a reinforcement learning (RL) stage to elicit stronger reasoning capabilities (Hu et al., 2025). On-policy distillation (OPD) (Song & Zheng, 2026; Liang et al., 2026b) has emerged as a particularly effective alternative to the RL stage. It trains a student model to match a stronger teacher's token-level distribution using dense per-token advantage signals. Compared to reinforcement learning from verifiable rewards (RLVR) (Schulman et al., 2017; Shao et al., 2024; Guo et al., 2025; Yu et al., 2025), OPD provides richer supervision and can achieve competitive reasoning performance at substantially lower training cost (Lu & Thinking Machines Lab, 2025; Yang et al., 2026b).

However, standard OPD requires the teacher to score every student rollout during training, which introduces a persistent infrastructure bottleneck. A dedicated multi-GPU teacher server typically runs in parallel with the training job, leading to substantial compute overhead and making large-scale experiments costly and difficult to reproduce, particularly for academic researchers without access to extensive serving infrastructure.

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

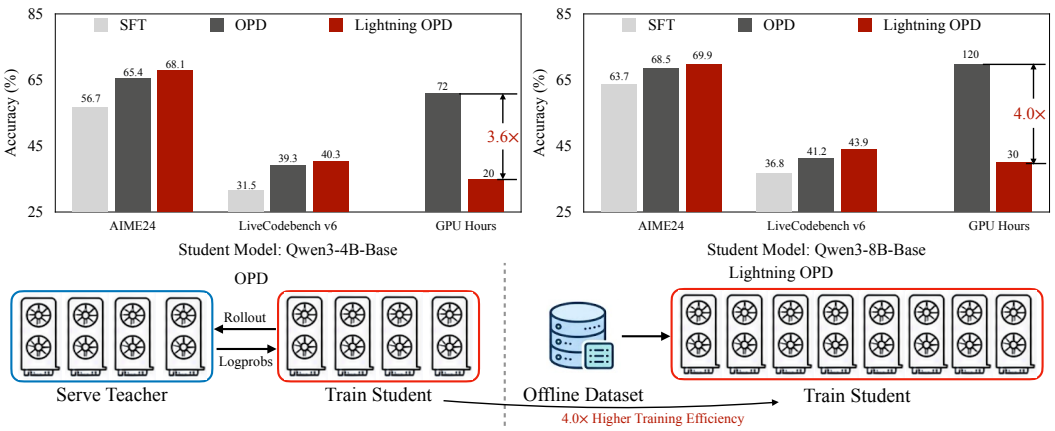


Figure 1: (Top) Performance (Pass@1, %) and training cost of Lightning OPD compared to standard OPD and the SFT baseline on Qwen3-4B-Base and Qwen3-8B-Base models. Lightning OPD achieves comparable performance to standard OPD on both math and coding benchmarks across both scales, while eliminating the need for a live teacher server during training. At the 8B scale, Lightning OPD achieves **69.9%** on AIME 2024 at a distillation cost of **30 GPU hours**, a **4.0**× reduction relative to standard OPD. (Bottom) Intuitive comparison of GPU resource allocation. Our standard OPD setup co-hosts the student and teacher on separate GPUs. Lightning OPD collects rollouts and teacher log-probabilities offline, dedicating all GPUs to student training.

A natural question is whether the benefits of on-policy supervision can be preserved while eliminating the need for a live teacher server. On-policy training follows the student's evolving rollout distribution, making the teacher appear indispensable. However, recent empirical studies suggest that RL training largely reweights reasoning behaviors already present after SFT (Yue et al., 2025a) and favors solutions close to the reference policy (Shenfeld et al., 2026b). We also observe token-level probability ratios concentrated near one when the same teacher is used across stages. This suggests a practical offline alternative (Rang et al., 2025): precompute teacher log-probabilities once over SFT rollouts and reuse them throughout training.

In practice, however, naively applying this offline precomputation fails to reliably match standard OPD. Investigating this discrepancy, we identify ***teacher consistency***: using the same teacher to generate SFT trajectories and provide the OPD reference distribution. Existing pipelines can combine different teachers across these stages. For example, Thinking Machines Lab (Lu & Thinking Machines Lab, 2025) uses OpenThoughts-3 demonstrations (Guha et al., 2026) from QwQ-32B to initialize a Qwen3-8B student before distilling from a Qwen3 teacher. Our teacher-pairing experiments show higher scores with consistent teachers for both offline and online OPD, with a larger advantage in the offline setting.

Building on this principle, we propose **Lightning OPD**. In the SFT stage, the base model is fine-tuned on teacher-generated trajectories to obtain the reference policy. In the OPD stage, rollouts are sampled from this reference, and the same teacher's log-probabilities are precomputed once over these responses, eliminating live teacher serving during training.

Our theoretical analysis bounds the online–offline gap in teacher-matching error under teacher consistency, with the bound vanishing as the student's initial fit improves. Teacher mismatch can leave a non-vanishing gap at a fixed training budget when it changes both context visitation and continuation targets. These results explain why teacher consistency is particularly valuable for offline distillation: cached supervision builds on the alignment established during SFT.

We evaluate Lightning OPD on mathematical reasoning and code generation with Qwen3 (Yang et al., 2025) and OLMo (Olmo Team et al., 2026) students, achieving performance comparable to standard OPD at substantially lower distillation GPU cost. The SFT-initialized Qwen3-8B student reaches **69.9% on AIME 2024** in **30 GPU hours**, a **4.0**× cost reduction. Improvements over SFT on math and code with OLMo demonstrate applicability across model families. Lightning OPD also enables distillation of Qwen3-30B-A3B, a 30B-parameter MoE model, on a **single 8×H100 node**,

reaching **71.0% on AIME 2024** and **60.8% on LiveCodeBench v5**. These results demonstrate the effectiveness of Lightning OPD as a practical framework that significantly lowers the barrier to academic research on LLM post-training.

2 Related Work

LLM Post-Training. LLM post-training combines learning from demonstrations with reward-driven optimization (Li et al., 2024). Supervised fine-tuning (SFT) learns from high-quality demonstrations (Dong et al., 2023; Yuan et al., 2023), while reinforcement learning (RL) uses verifiable outcomes (Liu et al., 2025b; Zheng et al., 2025; MiniMax et al., 2025) or intermediate feedback (Cui et al., 2026) to improve reasoning (Yue et al., 2025b; Yuan et al., 2025; Kazemnejad et al., 2025). Recent approaches integrate expert supervision with policy exploration through off-policy guidance (Yan et al., 2025), dynamically weighted SFT and RL objectives (Zhang et al., 2026b), and interleaved online fine-tuning (Ma et al., 2026). These methods highlight the value of coordinating supervision across post-training stages (Chen et al., 2026; Liu et al., 2025a; Huang et al., 2026). Lightning OPD brings this perspective to the choice of teachers across SFT and distillation, using teacher consistency to guide a pipeline that turns dense teacher supervision into a reusable offline resource.

On-Policy Distillation. Knowledge distillation transfers teacher behavior by matching output distributions (Hinton et al., 2015). Sequence-level distillation trains on teacher-generated responses (Kim & Rush, 2016), whereas MiniLLM and GKD supervise student-generated trajectories (Gu et al., 2024; Agarwal et al., 2024). Recent extensions develop reward extrapolation (Yang et al., 2026a), contrastive objectives for teacher- and student-generated data (Ko et al., 2025), self-distillation (Zhao et al., 2026; Hübotter et al., 2026; Shenfeld et al., 2026a; Penaloza et al., 2026), and token-level supervision allocation across multiple teachers (Sun et al., 2026). DistiLLM improves data efficiency by adaptively reusing student responses (Ko et al., 2024), while ReOPD replays teacher prefixes for multi-turn agents (Liao et al., 2026) and AsyncOPD overlaps rollout generation with learner updates (Kang et al., 2026). Rollout cost is further reduced through prefix-only distillation (Zhang et al., 2026a), turn-level budgeting (Zhou et al., 2026), and STRIDE's adaptive stopping and prefix reuse (Gui et al., 2026). DF-OPD generates training prompts with the teacher, removing the need for external prompt datasets (Li et al., 2026a). Other work examines teacher-guidance alignment (Armandpour et al., 2026), data selection (Hou et al., 2026), state coverage and supervision absorption (Fu et al., 2026b). In contrast to prior OPD work on distillation objectives (Ye et al., 2026) and rollout efficiency, we identify teacher consistency across SFT and distillation as a practical design principle for offline OPD. Lightning OPD builds on this principle to train on precomputed student rollouts and teacher scores, eliminating online rollout generation and live teacher serving during optimization.

Off-Policy Reinforcement Learning. Off-policy RL separates data collection from policy optimization, while offline RL learns from a fixed dataset (Levine et al., 2020). Classical methods enable such reuse through conservative value estimation and constraints tied to the data distribution (Kumar et al., 2020; Kostrikov et al., 2022; Fujimoto et al., 2019). For LLMs, fixed-data learning includes preference optimization (Rafailov et al., 2023) and reward-based reasoning methods such as TOPR and PCL-Reasoner-V1.5 (Le Roux et al., 2025; Lu et al., 2026). ReMix and ExGRPO combine historical rollouts with continued exploration (Liang et al., 2026a; Zhan et al., 2026), while OAPL learns from a lagged inference policy through reward-derived advantage regression (Ritter et al., 2026). Lightning OPD specializes data reuse to dense teacher supervision: cached token log-probabilities define its regression targets, enabling repeated student updates without response generation or teacher evaluation. Appendix D further develops this connection.

3 Methodology

3.1 Preliminaries

Let π_T denote a fixed teacher model and π_θ a trainable student. Given a prompt q, a response $x = (a_1, \ldots, a_T)$ is generated autoregressively as

$$\pi_\theta(x \mid q) = \prod_{t=1}^{T} \pi_\theta(a_t \mid s_t), \tag{1}$$

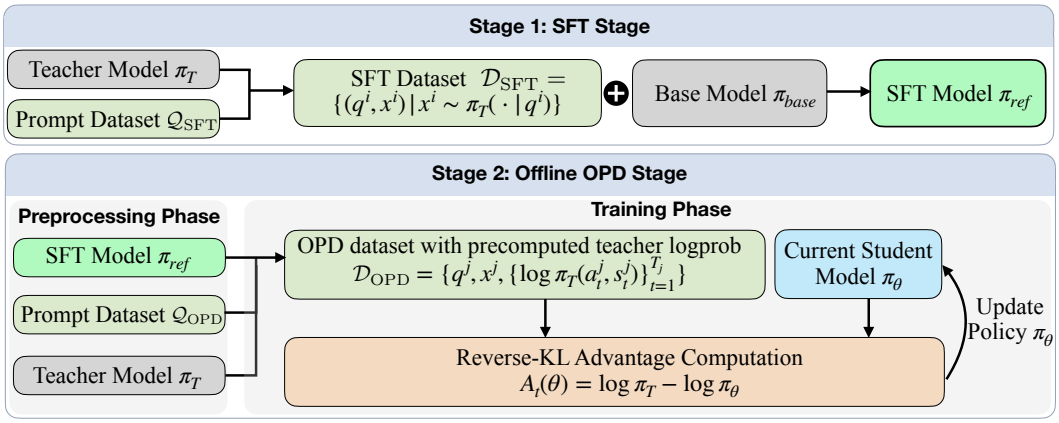


Figure 2: Overview of Lightning OPD. In the SFT stage, the teacher π_T generates trajectories on $\mathcal{Q}_{\text{SFT}}$ then the base model π_{base} is fine-tuned on these trajectories to obtain the SFT model π_{ref}. OPD stage proceeds in two phases. In the preprocessing phase, rollouts are sampled from the fixed student collection law B on $\mathcal{Q}_{\text{OPD}}$ and the teacher is queried once to compute and store per-token log-probabilities $\log \pi_T(a_t \mid s_t)$, forming the offline dataset $\mathcal{D}_{\text{OPD}}$. In the training phase, the student π_θ is initialized from π_{ref} for math (from the math-trained checkpoint for code) and trained on $\mathcal{D}_{\text{OPD}}$. The per-token advantage $A_t = \log \pi_T(a_t \mid s_t) - \log \pi_\theta(a_t \mid s_t)$ is computed by reading $\log \pi_T$ from $\mathcal{D}_{\text{OPD}}$ while computing $\log \pi_\theta$ online. During this process, we do not need a live teacher model. In contrast, standard OPD requires a live teacher server throughout training to compute advantages at every gradient step. By eliminating this requirement, Lightning OPD reduces distillation-stage cost, including offline preprocessing and excluding SFT, by **4.0**$\times$ at the 8B scale (from 120 to 30 GPU hours).

where $s_t = (q, a_1, \ldots, a_{t-1})$. Let π_{ref} denote the SFT-initialized student. The per-token OPD advantage is

$$A_t(\theta) = \log \pi_T(a_t \mid s_t) - \log \pi_\theta(a_t \mid s_t), \tag{2}$$

which is positive where the teacher is more confident than the student and negative otherwise. Standard OPD (Agarwal et al., 2024; Lu & Thinking Machines Lab, 2025) repeatedly samples responses from the current student and uses the teacher's token-level feedback to update the student. In the token-local update considered here, A_t is treated as a detached weight. For a nonempty response, the response-normalized data-update contribution is

$$\widehat{g}(\theta; q, x) = \frac{1}{T} \sum_{t=1}^{T} A_t(\theta) \nabla_\theta \log \pi_\theta(a_t \mid s_t). \tag{3}$$

Lightning OPD instead reuses responses collected from a fixed student checkpoint. Both methods use the same per-token advantage, while the student log-probabilities are recomputed as training proceeds. Their distinction lies in whether the response collection is repeated during training or completed in advance. Section 3.3 formalizes the sampling and normalization conventions for the online–offline comparison.

3.2 Lightning On-Policy Distillation

Following the common practice of LLM post-training (Ouyang et al., 2022), Lightning OPD consists of two stages. Figure 2 depicts an overview of the framework. We describe each stage in detail below and highlight where Lightning OPD departs from standard OPD.

Stage 1: Supervised Fine-Tuning. Given a base model π_{base}, a teacher π_T, and a prompt dataset $\mathcal{Q}_{\text{SFT}}$, we collect a supervised dataset from teacher-generated trajectories:

$$\mathcal{D}_{\text{SFT}} = \left\{ (q^i, x^i) \;\middle|\; x^i \sim \pi_T(\cdot \mid q^i),\ q^i \in \mathcal{Q}_{\text{SFT}} \right\}. \tag{4}$$

The base model is then fine-tuned on $\mathcal{D}_{\mathrm{SFT}}$ via maximum likelihood estimation to obtain the reference policy π_{ref}, using the objective

$$J_{\mathrm{SFT}}(\theta) = \mathbb{E}_{(q,x)\sim\mathcal{D}_{\mathrm{SFT}}}\left[\sum_{t=1}^{T} \log \pi_\theta(a_t \mid s_t)\right]. \tag{5}$$

We call reuse of the same teacher for SFT data generation and OPD scoring *teacher consistency*, and adopt it as a practical design guideline supported by the teacher-pairing experiments in Section 4.4.

Stage 2: Offline On-Policy Distillation. Starting from a task-specific prompt dataset $\mathcal{Q}_{\mathrm{OPD}}$, the OPD stage proceeds in two phases. In the *preprocessing phase*, rollouts are sampled once from a fixed student checkpoint, and the teacher is queried to precompute and store per-token log-probabilities, forming the offline dataset:

$$\mathcal{D}_{\mathrm{OPD}} = \left\{ \left(q^j, x^j, \{\log \pi_T(a_t^j \mid s_t^j)\}_{t=1}^{T_j}\right) \right\}_{j=1}^{N}, \qquad q^j \in \mathcal{Q}_{\mathrm{OPD}}. \tag{6}$$

Standard OPD requires a live teacher server throughout training, querying it for log-probabilities on freshly sampled rollouts from the current student. In the *training phase*, Lightning OPD trains the student on $\mathcal{D}_{\mathrm{OPD}}$ with no teacher server required. At each step, the advantage is computed as $A_t(\theta) = \log \pi_T(a_t \mid s_t) - \log \pi_\theta(a_t \mid s_t)$, where the teacher term is read directly from $\mathcal{D}_{\mathrm{OPD}}$ and the student term is computed online. This eliminates the concurrent teacher-serving requirement and makes distillation accessible with lower infrastructure overhead. The full procedure is given in Algorithm 1, with initialization, rollout, and training settings in Appendix B.

3.3 Theoretical Analysis

We analyze why teacher consistency reduces the cost of freezing rollouts. Our first result shows that accurate fit to the SFT teacher suppresses the offline–online error under token-score SGD. Our second result identifies how a teacher change can create an additional need for online supervision. All proofs are deferred to Appendix A.

Let U denote the SFT teacher, $B = \pi_{\theta_0}$ the initialized student, and V the distillation teacher. The token-local update in Equation (3), averaged over a fixed cache D, is the negative gradient of

$$\widehat{\mathcal{L}}_D^V(\theta) = \frac{1}{2}\,\mathrm{Avg}_D\left(\log \pi_\theta(a \mid s) - \log V(a \mid s)\right)^2, \tag{7}$$

where averaging is within responses and then across responses. Thus the teacher specifies the targets and the rollouts specify where those targets are fitted. Compare two learners starting from B: offline training collects N responses from the initial student; online training collects one response from the current student before each update. Both process each response once, using the same update rule and teacher-feedback budget. For their final policies P_{off}^V and P_{on}^V, define

$$D_B(V) = \mathbb{E}\,\mathrm{KL}(P_{\mathrm{off}}^V \| V) - \mathbb{E}\,\mathrm{KL}(P_{\mathrm{on}}^V \| V), \tag{8}$$

where KL is over complete responses and expectations are over training samples. Positive $D_B(V)$ measures the additional error from freezing rollouts.

Theorem 3.1 (Teacher Consistency under Score SGD)**.** Consider a finite response space with length H and positive conditional probabilities. Both learners take N SGD updates on the per-response loss in Equation (7), with the same fixed step sizes. Suppose the initial student satisfies $\max_{s,a} |\log B(a \mid s) - \log U(a \mid s)| \le h$. If score gradients are bounded and the response-collection law is Lipschitz in total variation on neighborhoods of uniform radius, with constants uniform across initializations, then for sufficiently small h,

$$|D_B(U)| \le C_N h^3. \tag{9}$$

The constant C_N is independent of h, and the student may share parameters across all tokens.

Remark 3.2. Teacher consistency allows distillation to build on the fit achieved during SFT. Over a fixed number of updates, score residuals of size $O(h)$ keep parameter movement and sampling-law changes at $O(h)$, while the response-KL risk is $O(h^2)$. Coupling the two sampling procedures gives the cubic bound: their sample paths differ with probability $O(h)$, and their risks differ by at most $O(h^2)$. Appendix A.2 gives the explicit constants.

Changing the teacher introduces a second term in the initial residual: $\log T - \log B = (\log U - \log B) + (\log T - \log U)$. Improving fit to U reduces the first term but leaves the second. The resulting updates can increase visitation of contexts whose continuations still require supervision.

To isolate this coverage effect, consider a two-token model with independent parameters: the first token chooses L or R, and the second chooses a binary continuation. Write $U = (\alpha, u)$ and $T = (\beta, r)$ for the teachers' probabilities of visiting R and choosing continuation 1 there; they agree within L. Each response moves the root probability a fraction ρ toward its teacher target, and moves the R continuation a fraction γ when R is visited. The full model is in Appendix A.3.

Theorem 3.3 (Teacher Changes and Context Coverage)**.** In this model, fix positive teacher probabilities, $N \geq 2$, and $\rho, \gamma \in (0, 1]$. For $B_h = (\varepsilon_h, q_h)$ with $|\varepsilon_h - \alpha|, |q_h - u| \leq h$, a changed teacher satisfying $\beta > \alpha$ and $r \neq u$ gives

$$|D_{B_h}(U)| \leq Ch^3, \qquad \lim_{h \to 0} D_{B_h}(T) = D_0 > 0. \tag{10}$$

Here C is independent of h. Both learners use N updates and $2N$ teacher-token scores. Hence $D_{B_h}(T) > D_{B_h}(U)$ for sufficiently small h.

Remark 3.4. The changed teacher favors both more visits to R and a different continuation there. Online collection supplies more supervision as visitation increases; the cache retains its original coverage. This yields a positive offline penalty even as the initial fit to U becomes exact. Both learners have the same final root probability, so their risk difference comes from fitting the continuation. At $B = U$, changing only visitation leaves that continuation fitted; changing only the continuation leaves the sampling frequencies unchanged. Either case gives $D_U(T) = 0$. The positive gap therefore isolates an interaction: the new teacher requires additional supervision precisely where the student learns to visit more often.

Appendix A.4 also constructs a positive separation under shared-parameter score SGD. Together, these results explain the value of teacher consistency for Lightning OPD: preserving the SFT target reduces the need to adapt supervision during distillation. Section 4.4 tests this design choice through teacher pairings and comparisons at matched feedback budgets.

4 Experiments

4.1 Experimental Setup

Models. Our Qwen3 experiments pair Qwen3-4B-Base with Qwen3-8B, Qwen3-8B-Base with Qwen3-32B, and the mixture-of-experts (MoE) student Qwen3-30B-A3B-Base with Qwen3-30B-A3B-Thinking-2507 (Yang et al., 2025), following the two-stage pipeline in Section 3.2. We also apply our method to the OLMo family, initializing from Olmo-3-7B-Think-SFT with Olmo-3-32B-Think as the teacher (Olmo Team et al., 2026).

Training Data. For SFT, we fine-tune base models using OpenThoughts-3 prompts (Guha et al., 2026) paired with responses generated by the respective teacher model. For OPD, we use DAPO-Math-17k (Yu et al., 2025) for mathematical reasoning and a sampled 30K subset of EpiCoder-func-380k (Wang et al., 2025) for code generation. Offline preprocessing samples one student response per prompt and caches the teacher's token-level log-probabilities before training.

Benchmarks. For mathematical reasoning, we evaluate on AIME 2024 (AI-MO, 2024), AIME 2025 (OpenCompass, 2025), and HMMT 2025 (Balunović et al., 2025). For code generation, we evaluate on LiveCodeBench v5 and v6 (Jain et al., 2025). In our evaluations, we set the temperature to 0.6, top-p to 0.95, and the maximum generation length to 32,768 for math benchmarks and 40,960 for coding benchmarks. We sample 32 solutions per problem for math benchmarks and 4 solutions per problem for code benchmarks, reporting the average pass@1.

Training Settings. The SFT stage is implemented using LlamaFactory (Zheng et al., 2024), and the OPD stage is implemented using slime (Zhu et al., 2025). Standard OPD samples rollouts online from the current student, while Lightning OPD reuses student rollouts and teacher log-probabilities precomputed before training begins. Full hyperparameter details are provided in Appendix B.

Table 1: Average pass@1 (%) across student backbone families. † marks external results under the source papers' settings. OOM refers to our standard OPD setup on one 8×H100 node. **Bold** marks the best listed result per group; – denotes an unreported result.

Method	Math Reasoning				Code Generation		
	AIME 2024	AIME 2025	HMMT 2025	Avg.	LCB v5	LCB v6	Avg.
Student: Qwen3-4B-SFT Teacher: Qwen3-8B							
SFT	56.7	52.1	34.0	47.6	33.8	31.5	32.6
GKD† (Agarwal et al., 2024)	58.1	50.4	29.8	46.1	–	–	–
ExOPD† (Yang et al., 2026a)	61.0	56.0	34.4	50.5	–	29.0	–
AOPD† (Jia et al., 2026)	61.0	54.4	32.1	49.2	–	–	–
OPD	65.4	57.9	**39.9**	54.4	**44.2**	39.3	**41.8**
Lightning OPD	**68.1**	**58.4**	39.8	**55.4**	42.8	**40.3**	41.5
Student: Qwen3-8B-SFT Teacher: Qwen3-32B							
SFT	63.7	51.7	36.9	50.8	44.7	36.8	40.8
GKD†	66.3	48.5	34.2	49.7	–	–	–
AOPD†	66.9	55.0	38.3	53.4	–	–	–
OPD	68.5	59.0	39.4	55.6	47.3	41.2	44.2
Lightning OPD	**69.9**	**59.2**	**41.9**	**57.0**	**49.5**	**43.9**	**46.7**
Student: Olmo-3-7B-Think-SFT Teacher: Olmo-3-32B-Think							
SFT	71.6	59.5	39.3	56.8	46.1	41.0	43.6
OPD	**74.6**	**63.4**	41.7	**59.9**	**53.9**	47.1	50.5
Lightning OPD	72.3	62.2	**43.3**	59.3	53.7	**48.3**	**51.0**
Student: Qwen3-30B-A3B-SFT Teacher: Qwen3-30B-A3B-Thinking-2507							
SFT	66.8	63.2	44.6	58.2	39.4	33.0	36.2
OPD (OOM)	✗	✗	✗	✗	✗	✗	✗
Lightning OPD	**71.0**	**66.3**	**48.3**	**61.9**	**60.8**	**54.4**	**57.6**

4.2 Main Results

Table 1 compares Qwen3 and OLMo students across five benchmarks, including externally reported distillation baselines. Lightning OPD matches standard OPD on Qwen3-4B/8B and OLMo while eliminating live teacher serving, and improves on SFT across all evaluated students in both math and code. At 4B and 8B, its math averages reach **55.4%** and **57.0%**, compared with 49.2% and 53.4% reported for AOPD. The 8B student achieves **69.9% on AIME 2024** and **49.5% on LiveCodeBench v5**, exceeding standard OPD on both benchmarks. For OLMo, Lightning OPD reaches **72.3% on AIME 2024** and a code average of **51.0%**, compared with 50.5% for standard OPD and 43.6% for SFT. With Qwen3-30B-A3B, it enables distillation on a **single 8×H100 node**, achieving **71.0% on AIME 2024** and **60.8% on LiveCodeBench v5**. These results demonstrate the effectiveness of cached teacher supervision across model families, architectures, and task domains.

4.3 Training Cost

Table 2 compares the training cost of standard OPD and Lightning OPD. At the 4B scale, Lightning OPD reduces total GPU hours from 72 to **20**, a **3.6**× reduction. At the 8B scale, cost falls from 120 to **30 GPU hours**, a **4.0**× reduction. For OLMo, the cost decreases from 195 to **42 GPU hours**, a 4.6× reduction. The per-phase breakdown shows how these savings arise. Standard OPD reserves GPUs for rollout generation and teacher scoring throughout training, accounting for 54 of its 72 GPU hours at 4B and 60 of its 120 GPU hours at 8B. Lightning OPD performs these operations once, in 12 and 14 GPU hours, followed by 8 and 16 GPU hours of student training. Separating supervision collection from optimization removes concurrent rollout and teacher serving, making OPD accessible through a standard student training job.

Table 2: Training cost (GPU hours) of standard OPD vs. Lightning OPD. Lightning OPD reduces GPU cost by 3.6× at 4B and 4.0× at 8B, requiring 20 and 30 GPU hours, respectively. For OLMo-3-7B, training costs decrease from 195 to 42 GPU hours, a 4.6× reduction. The breakdown shows the cost of rollout collection, teacher scoring, and student training.

Setting	Method	Updates	Rollout	Teacher	Student	Total
Qwen3-4B	OPD	150	36	18	18	72
	Lightning OPD	150	10	2	8	**20**
Qwen3-8B	OPD	150	30	30	60	120
	Lightning OPD	150	10	4	16	**30**
OLMo-3-7B	OPD	150	78	39	78	195
	Lightning OPD	150	13	5	24	**42**

Table 3: AIME 2024 pass@1 (%) across the evaluated teacher combinations. Rows indicate the SFT teacher and columns the OPD teacher; consistent pairings are **bold**. The recorded configurations include different SFT checkpoints and distillation durations.

Standard OPD	OPD Teacher			OPD Teacher	
SFT Teacher	Qwen3-8B	QwQ-32B	SFT Teacher	Qwen3-32B	QwQ-32B
Student: Qwen3-4B-Base			*Student: Qwen3-8B-Base*		
Qwen3-8B	**65.4**	62.4	Qwen3-32B	**68.5**	64.8
QwQ-32B	61.2	**62.8**	QwQ-32B	65.0	**66.5**

Lightning OPD	OPD Teacher			OPD Teacher	
SFT Teacher	Qwen3-8B	QwQ-32B	SFT Teacher	Qwen3-32B	QwQ-32B
Student: Qwen3-4B-Base			*Student: Qwen3-8B-Base*		
Qwen3-8B	**68.1**	62.5	Qwen3-32B	**69.9**	63.1
QwQ-32B	59.3	**63.1**	QwQ-32B	62.1	**68.7**

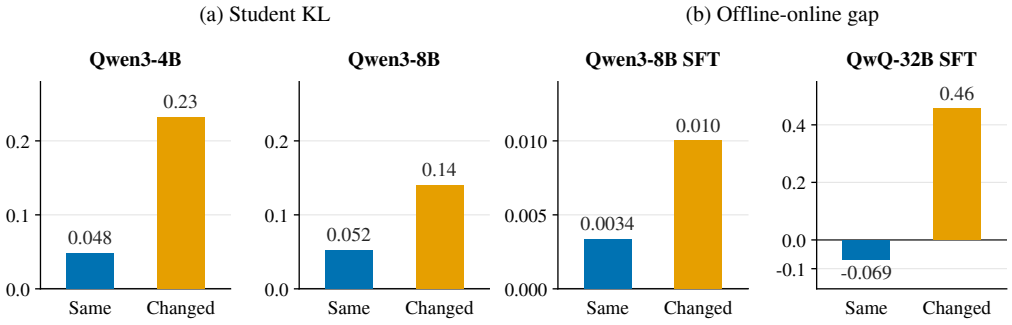
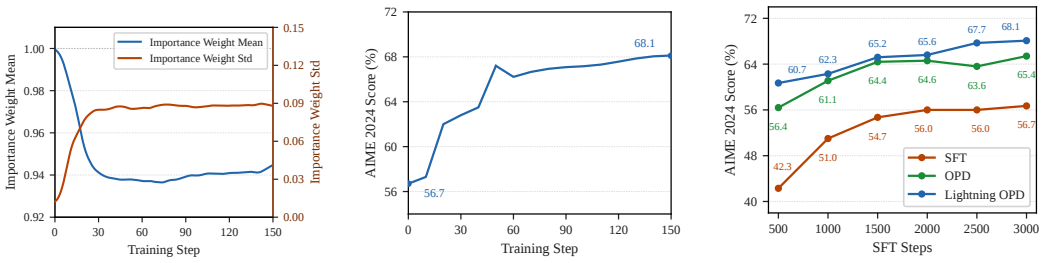


Figure 3: Teacher consistency and offline distillation. **(a)** Conditional $\mathrm{KL}(\pi_{\mathrm{ref}} \| \pi_\theta)$ on fixed cached states after offline distillation of Qwen3-4B/8B; the changed teacher is QwQ-32B. **(b)** Offline–online gap $G = L_{\mathrm{off}} - L_{\mathrm{on}}$ in a separate Qwen3-4B comparison with matched update and feedback budgets, using Qwen3-8B/QwQ-32B SFT demonstrations, respectively. Positive gaps favor online collection. Gap panels use different scales.

4.4 Ablation Study

Teacher consistency. We examine teacher consistency by crossing the SFT and OPD teacher choices, using QwQ-32B (Qwen Team, 2025) alongside Qwen3-8B at the 4B scale and Qwen3-32B at the 8B scale. Table 3 shows that consistent pairings achieve higher AIME 2024 scores for both



(a) Token probability ratios. (b) OPD learning curve. (c) SFT initialization.

Figure 4: Training dynamics of Lightning OPD with the Qwen3-4B-Base student. **(a)** Mean and standard deviation of student-to-reference probability ratios on recorded training tokens. **(b)** AIME 2024 pass@1 over 150 OPD steps. **(c)** AIME 2024 pass@1 after SFT and subsequent distillation as a function of SFT training steps.

methods at both scales, with a larger advantage for Lightning OPD. For example, with Qwen3-32B as the SFT teacher at 8B, Lightning OPD scores **69.9%** when retaining the same teacher and **63.1%** with QwQ-32B, a **6.8-point** difference; the corresponding standard OPD scores are 68.5% and 64.8%, a **3.7-point** difference. These results highlight the importance of teacher consistency for both distillation paradigms, particularly when reusing fixed rollouts.

Effect on offline distillation. Figure 3(a) shows smaller student drift under teacher consistency. In the offline Qwen-SFT runs, we measure conditional $\mathrm{KL}(\pi_{\mathrm{ref}}\|\pi_\theta)$ on fixed cached states within each scale. The QwQ-32B-target runs exhibit **4.8**$\times$ and **2.7**$\times$ the drift of the consistent-teacher runs at 4B and 8B, respectively. Figure 3(b) compares offline and online collection for two Qwen3-4B students initialized with Qwen3-8B or QwQ-32B demonstrations. Each student is distilled with both teachers under matched update and feedback budgets. We measure $G = L_{\mathrm{off}} - L_{\mathrm{on}}$ on the same held-out responses, where L is the mean half squared student–teacher log-score error against the distillation teacher. Positive G favors online collection. Changing the teacher increases G in both initialization directions; for QwQ-SFT, its sign changes from negative to positive. Together with the benchmark ablation, these observations support the role of teacher consistency in limiting policy drift and the additional error from fixed collection (Section 3.3). Experimental details are provided in Appendices C.2 and C.3.

Training dynamics. With a consistent teacher, Lightning OPD improves the SFT student using cached supervision. Figure 4(a,b) shows AIME 2024 accuracy rising from 56.7% to 68.1%, while the mean student-to-reference token probability ratio settles near 0.94 with a standard deviation below 0.10. These gains accompany modest changes in the recorded token probabilities, consistent with the smaller drift observed above. The benefit extends across SFT initializations (Figure 4(c)): Lightning OPD improves on every evaluated checkpoint, with stronger initializations generally yielding higher final scores. SFT and offline distillation thus provide complementary gains across the tested initializations.

5 Conclusion

We presented Lightning OPD, which caches teacher log-probabilities on student rollouts and removes live teacher serving from distillation. Teacher consistency coordinates SFT demonstrations with the targets used during offline training. Our analysis shows that accurate fit to a consistent teacher suppresses the offline–online error under token-score SGD, and identifies how changed context preferences and continuation targets can create an additional need for online supervision. Controlled score-fitting experiments show larger offline–online gaps after teacher switches in both initialization directions under matched feedback budgets. Qwen3-8B reaches 69.9% on AIME 2024 in 30 GPU hours, a 4.0$\times$ reduction in GPU cost. Results with OLMo and single-node Qwen3-30B-A3B distillation extend the practical value of cached supervision across model families and scales, lowering the barrier to academic research on LLM post-training.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, and assist in the writing of proofs. We have not used generative AI tools to generate synthetic data sets beyond the teacher- and student-generated trajectories described in the paper. We also did not use AI tools to propose or refine hypotheses; design or provide feedback on research methodology or experiments; implement methods; assist with translation; clean and reformat datasets; support qualitative and thematic data analysis; or interpret results. Additionally, we used generative AI tools to edit the paper to improve readability. The authors reviewed all AI-assisted analyses and language edits. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

AI-MO. AIME 2024. https://huggingface.co/datasets/AI-MO/aimo-validation-aime, 2024.

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc N. M Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why, 2026. URL https://arxiv.org/abs/2605.10889.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating LLMs on uncontaminated math competitions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=y0zL9IZxZ7.

Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. Beyond two-stage training: Cooperative SFT and RL for LLM reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=xWvj03N4sJ.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=9SkkifLopZ.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=m7p5O7zblY.

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*, 2026a. URL https://openreview.net/forum?id=uZbZb058bA.

Zixuan Fu, Bingxiang He, Yuxin Zuo, Haohuan Huang, Jinqian Zhang, Ruhang Xiao, Cheng Qian, Qinyu Luo, Huan-ang Gao, Yudong Wang, Zhiyuan Liu, Ning Ding, and Chaojun Xiao. Rethinking on-policy distillation of large language models ii: One training example, 2026b. URL https://arxiv.org/abs/2609.04172.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8ac015d409635f196f9e3e9dcfb9a94e-Paper-Conference.pdf.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. OpenThoughts: Data recipes for reasoning models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7xjoTuaNmN.

Zhiyu Gui, Kexin Huang, Jia Guo, Junkang Wu, Zihao Wang, Zhiqiang Zhang, Jun Zhou, Jiancan Wu, and Xiang Wang. Know when to stop, where to restart: Accelerating multi-turn agentic on-policy distillation, 2026. URL https://arxiv.org/abs/2609.14636.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, Sep 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Zhinan Hou, Jiaqi Zhang, Xunliang Cai, and Keyou You. What matters in on-policy distillation? a perspective on data efficiency and data selection, 2026. URL https://arxiv.org/abs/2609.05198.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on

the base model. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 162239–162262. Curran Associates, Inc., 2025. doi: 10.52202/085713-5418. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/ed873d79e7c268c020c4b4db13a2812a-Paper-Conference.pdf.

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Yinghui Xu, Edoardo Ponti, and Ivan Titov. Blending supervised and reinforcement fine-tuning with prefix sampling. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=MgtyPYrC4h.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=QkfkxyRizZ.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida I. Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=chfJJYC3iL.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level, 2026. URL https://arxiv.org/abs/2605.06387.

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. URL https://openreview.net/forum?id=J5i09faOOf.

Wonjun Kang, Kevin Galim, Seunghyuk Oh, Minjun Kang, Sanghyun Park, Donghoon Kim, Minjae Lee, Minseo Kim, Rishabh Tiwari, Yuchen Zeng, Hyung Il Koo, and Kangwook Lee. Asyncopd: How stale can on-policy distillation be?, 2026. URL https://arxiv.org/abs/2606.24143.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29557–29590. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, Yanru Chen, Yanxu Chen, Yicun Chen, Yimin Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Ziwei Chen, Dazhi Cheng, Yean Cheng, Minghan Chu, Jialei Cui, Jiaqi Deng, Muxi Diao, Hao Ding, Mengfan Dong, Mengnan Dong, Yuxin Dong, Yuhao Dong, Ang'ang Du, Chenzhuang Du, Dikang Du, Lingxiao Du, Yulun Du, Yu Fan, Shengjun Fang, Qiulin Feng, Yichen Feng, Garimugai Fu, Kelin Fu, Hongcheng Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Chengyang Gong, Xiaochen Gong, Zhuoma Gongque, Qizheng Gu, Xinran Gu, Yicheng Gu, Longyu Guan, Shuhao Guan, Yuanying Guo, Xiaoru Hao, Dailan He, Tianhong He, Weiran He, Wenyang He, Yibo He, Yunjia

He, Chao Hong, Hao Hu, Jiaxi Hu, Yangyang Hu, Zhenxing Hu, Ke Huang, Ruiyuan Huang, Weixiao Huang, Zhiqi Huang, Chaobo Jia, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yu Jing, Guokun Lai, Aidi Li, C. Li, Cheng Li, Fang Li, Guanghe Li, Guanyu Li, Haitao Li, Haoyang Li, Jia Li, Jingwei Li, Junxiong Li, Lincan Li, Mo Li, Weihong Li, Wentao Li, Xinhang Li, Xinhao Li, Yang Li, Yanhao Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Weilong Liao, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zhishan Lin, Zichao Lin, Cheng Liu, Chenyu Liu, Hongzhang Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Tianyu Liu, Weizhou Liu, Xiangyan Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yuanxin Liu, Zhengying Liu, Zhongnuo Liu, Enzhe Lu, Haoyu Lu, Zhiyuan Lu, G. Luo, Junyu Luo, Tongxu Luo, Yashuo Luo, Long Ma, Shaoguang Mao, Yuan Mei, Xin Men, Fanqing Meng, Zhiyong Meng, Yibo Miao, Minqing Ni, Kun Ouyang, Siyuan Pan, Bo Pang, Yuchao Qian, Ruoyu Qin, Zeyu Qin, Jiezhong Qiu, Bowen Qu, Zeyu Shang, Youbo Shao, Tianxiao Shen, Zhennan Shen, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Feifan Song, Pengwei Song, Tianhui Song, Xiaoxi Song, Hongjin Su, Jianlin Su, Zhaochen Su, Lin Sui, Jinsong Sun, Junyao Sun, Tongyu Sun, Flood Sung, Yunpeng Tai, Chuning Tang, Heyi Tang, Xiaojuan Tang, Zhengyang Tang, Jiawen Tao, Shiyuan Teng, Chaoran Tian, Pengfei Tian, Bowen Wang, Chensi Wang, Chuang Wang, Congcong Wang, Dingkun Wang, Dinglu Wang, Dongliang Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hengzhi Wang, Huaqing Wang, Hui Wang, Jiahao Wang, Jinhong Wang, Jiuzheng Wang, Kaixin Wang, Linian Wang, Qibin Wang, Shengjie Wang, Shuyi Wang, Si Wang, Wei Wang, Xiaochen Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yipu Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhexu Wang, Zifan Wang, Zihan Wang, Zizhe Wang, Chu Wei, Ming Wei, Chuan Wen, Zichen Wen, Chengjie Wu, Haoning Wu, Junyan Wu, Rucong Wu, Wenhao Wu, Yuefeng Wu, Yuhao Wu, Yuxin Wu, Zijian Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Yuchong Xie, Bowei Xing, Boyu Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinbo Xu, Xinran Xu, Yangchuan Xu, Yichang Xu, Yuemeng Xu, Zelai Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Guangyao Yang, Hao Yang, Junwei Yang, Kai Yang, Ningyuan Yang, Xiaofei Yang, Xinlong Yang, Xinyu Yang, Ying Yang, Yi Yang, Yi Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhuorui Ye, Peng Yebo, Bohong Yin, Chengzhen Yu, Longhui Yu, Tao Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Xiaokun Yuan, Yang Yue, Weihao Zeng, Dunyuan Zha, Haobing Zhan, Dehao Zhang, Hao Zhang, Jin Zhang, Puqi Zhang, Qiao Zhang, Rui Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yushun Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Chenguang Zhao, Feifan Zhao, Jinxiang Zhao, Shuai Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Junfeng Zhong, Longguang Zhong, Weiming Zhong, M. Zhou, Runjie Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yuxuan Zhu, Zhen Zhu, Jingze Zhuang, Weiyu Zhuang, Ying Zou, and Xinxing Zu. Kimi k2.5: Visual agentic intelligence, 2026. URL https://arxiv.org/abs/2602.02276.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24872–24895. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31044–31062. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/ko25a.html.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.),

Advances in Neural Information Processing Systems, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

Nicolas Le Roux, Marc G. Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alex Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Toth, and Sam Work. Tapered off-policy REINFORCE: Stable and efficient reinforcement learning for LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=gFFgCWiXWI.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. URL https://arxiv.org/abs/2005.01643.

Gengsheng Li, Mao Zheng, Mingyang Song, Jie Sun, Zeyuan Liu, Ruiqi Liu, Qiyong Zhong, Haiyun Guo, Junfeng Fang, and Jinqiao Wang. Data-free on-policy distillation, 2026a. URL https://arxiv.org/abs/2609.14193.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe, 2026b. URL https://arxiv.org/abs/2604.13016.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 29128–29163. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/li24cd.html.

Jing Liang, Jinyi Liu, Yi Ma, Hongyao Tang, Yan Zheng, Shuyue Hu, Lei Bai, and Jianye Hao. Squeeze the soaked sponge: Efficient off-policy RFT for large language model. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=quBjNSJMrC.

Kun Liang, Clive Bai, Xin Xu, Chenming Tang, Sanwoo Lee, Weijie Liu, Saiyong Yang, and Yunfang Wu. Orbit: On-policy exploration-exploitation for controllable multi-budget reasoning, 2026b. URL https://arxiv.org/abs/2601.08310.

Baohao Liao, Hanze Dong, Christof Monz, Xinxing Xu, Li Dong, and Furu Wei. Multi-turn on-policy distillation with prefix replay, 2026. URL https://arxiv.org/abs/2607.04763.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar. Uft: Unifying supervised and reinforcement fine-tuning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 101347–101383. Curran Associates, Inc., 2025a. doi: 10.52202/085713-3394. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/92e886487a8354b03d8bf4416eae6d7d-Paper-Conference.pdf.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025b. URL https://openreview.net/forum?id=5PAF7PAY2Y.

LLM-Core Xiaomi. Mimo-v2-flash technical report, 2026. URL https://arxiv.org/abs/2601.02780.

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

Yao Lu, Dengdong Fan, Jianzheng Nie, Fan Xu, Jie Chen, Bin Zhou, and Yonghong Tian. Pcl-reasoner-v1.5: Advancing math reasoning with offline reinforcement learning, 2026. URL https://arxiv.org/abs/2601.14716.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, Bin Cui, and Wentao Zhang. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=LzCBLrNoyM.

MiniMax, Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, Chengjun Xiao, Chengyu Du, Chi Zhang, Chu Qiao, Chunhao Zhang, Chunhui Du, Congchao Guo, Da Chen, Deming Ding, Dianjun Sun, Dong Li, Enwei Jiao, Haigang Zhou, Haimo Zhang, Han Ding, Haohai Sun, Haoyu Feng, Huaiguang Cai, Haichao Zhu, Jian Sun, Jiaqi Zhuang, Jiaren Cai, Jiayuan Song, Jin Zhu, Jingyang Li, Jinhao Tian, Jinli Liu, Junhao Xu, Junjie Yan, Junteng Liu, Junxian He, Kaiyi Feng, Ke Yang, Kecheng Xiao, Le Han, Leyang Wang, Lianfei Yu, Liheng Feng, Lin Li, Lin Zheng, Linge Du, Lingyu Yang, Lunbin Zeng, Minghui Yu, Mingliang Tao, Mingyuan Chi, Mozhi Zhang, Mujie Lin, Nan Hu, Nongyu Di, Peng Gao, Pengfei Li, Pengyu Zhao, Qibing Ren, Qidi Xu, Qile Li, Qin Wang, Rong Tian, Ruitao Leng, Shaoxiang Chen, Shaoyu Chen, Shengmin Shi, Shitong Weng, Shuchang Guan, Shuqi Yu, Sichen Li, Songquan Zhu, Tengfei Li, Tianchi Cai, Tianrun Liang, Weiyu Cheng, Weize Kong, Wenkai Li, Xiancai Chen, Xiangjun Song, Xiao Luo, Xiao Su, Xiaobo Li, Xiaodong Han, Xinzhu Hou, Xuan Lu, Xun Zou, Xuyang Shen, Yan Gong, Yan Ma, Yang Wang, Yiqi Shi, Yiran Zhong, Yonghong Duan, Yongxiang Fu, Yongyi Hu, Yu Gao, Yuanxiang Fan, Yufeng Yang, Yuhao Li, Yulin Hu, Yunan Huang, Yunji Li, Yunzhi Xu, Yuxin Mao, Yuxuan Shi, Yuze Wenren, Zehan Li, Zelin Li, Zhanxu Tian, Zhengmao Zhu, Zhenhua Fan, Zhenzhen Wu, Zhichao Xu, Zhihang Yu, Zhiheng Lyu, Zhuo Jiang, Zibo Gao, Zijia Wu, Zijian Song, and Zijun Sun. Minimax-m1: Scaling test-time compute efficiently with lightning attention, 2025. URL https://arxiv.org/abs/2506.13585.

NVIDIA, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, Aleksandr Shaposhnikov, Alex Kondratenko, Alexander Bukharin, Alexandre Milesi, Ali Taghibakhshi, Alisa Liu, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amir Klein, Amit Zuker, Amnon Geifman, Amy Shen, Anahita Bhiwandiwalla, Andrew Tao, Anjulie Agrusa, Ankur Verma, Ann Guan, Anubhav Mandarwal, Arham Mehta, Ashwath Aithal, Ashwin Poojary, Asif Ahamed, Asit Mishra, Asma Kuriparambil Thekkumpate, Ayush Dattagupta, Banghua Zhu, Bardiya Sadeghi, Barnaby Simkin, Ben Lanir, Benedikt Schifferer, Besmira Nushi, Bilal Kartal, Bita Darvish Rouhani, Boris Ginsburg, Brandon Norick, Brandon Soubasis, Branislav Kisacanin, Brian Yu, Bryan Catanzaro, Carlo del Mundo, Chantal Hwang, Charles Wang, Cheng-Ping Hsieh, Chenghao Zhang, Chenhan Yu, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christopher Parisien, Collin Neale, Cyril Meurillon, Damon Mosk-Aoyama, Dan Su, Dane Corneil, Daniel Afrimi, Daniel Lo, Daniel Rohrer, Daniel Serebrenik, Daria Gitman, Daria Levy, Darko Stosic, David Mosallanezhad, Deepak Narayanan, Dhruv Nathawani, Dima Rekesh, Dina Yared, Divyanshu Kakwani, Dong Ahn, Duncan Riach, Dusan Stosic, Edgar Minasyan, Edward Lin, Eileen Long, Eileen Peters Long, Elad Segal, Elena Lantz, Ellie Evans, Elliott Ning, Eric Chung, Eric Harper, Eric Tramel, Erick Galinkin, Erik Pounds, Evan Briones, Evelina Bakhturina, Evgeny Tsykunov, Faisal Ladhak, Fay Wang, Fei Jia, Felipe Soares, Feng Chen, Ferenc Galko, Frank Sun, Frankie Siino, Gal Hubara Agam, Ganesh Ajjanagadde, Gantavya Bhatt, Gargi Prasad, George Armstrong, Gerald Shen, Gorkem Batmaz, Grigor Nalbandyan, Haifeng Qian, Harsh Sharma, Hayley Ross, Helen Ngo, Herbert Hum, Herman Sahota, Hexin Wang, Himanshu Soni, Hiren Upadhyay, Huizi Mao, Huy C Nguyen, Huy Q Nguyen, Iain Cunningham, Ido Galil, Ido Shahaf, Igor Gitman, Ilya Loshchilov, Itamar Schen, Itay Levy, Ivan Moshkov, Izik Golan, Izzy Putterman, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jatin Mitra, Jeffrey Glick, Jenny Chen, Jesse Oliver, Jian Zhang, Jiaqi Zeng, Jie Lou, Jimmy Zhang, Jinhang Choi, Jining Huang, Joey Conway, Joey Guman, John Kamalu, Johnny Greco, Jonathan Cohen, Joseph Jennings, Joyjit Daw, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kai Xu, Kan Zhu, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kevin Shih, Kezhi Kong, Khushi Bhardwaj, Kirthi Shankar, Krishna C. Puvvada, Krzysztof Pawelec, Kumar Anik, Lawrence McAfee, Laya Sleiman, Leon Derczynski, Li Ding, Lizzie Wei, Lucas Liebenwein, Luis Vega, Maanu Grover, Maarten Van Segbroeck, Maer Rodrigues de Melo, Mahdi

Nazemi, Makesh Narsimhan Sreedhar, Manoj Kilaru, Maor Ashkenazi, Marc Romeijn, Marcin Chochowski, Mark Cai, Markus Kliegl, Maryam Moosaei, Matt Kulka, Matvei Novikov, Mehrzad Samadi, Melissa Corpuz, Mengru Wang, Meredith Price, Michael Andersch, Michael Boone, Michael Evans, Miguel Martinez, Mikail Khona, Mike Chrzanowski, Minseok Lee, Mohammad Dabbah, Mohammad Shoeybi, Mostofa Patwary, Nabin Mulepati, Najeeb Nabwani, Natalie Hereth, Nave Assaf, Negar Habibi, Neta Zmora, Netanel Haber, Nicola Sessions, Nidhi Bhatia, Nikhil Jukar, Nikki Pope, Nikolai Ludwig, Nima Tajbakhsh, Nir Ailon, Nirmal Juluru, Nishant Sharma, Oleksii Hrinchuk, Oleksii Kuchaiev, Olivier Delalleau, Oluwatobi Olabiyi, Omer Ullman Argov, Omri Puny, Oren Tropp, Ouye Xie, Parth Chadha, Pasha Shamis, Paul Gibbons, Pavlo Molchanov, Pawel Morkisz, Peter Dykas, Peter Jin, Pinky Xu, Piotr Januszewski, Pranav Prashant Thombre, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Qing Miao, Qiyu Wan, Rabeeh Karimi Mahabadi, Rachit Garg, Ran El-Yaniv, Ran Zilberstein, Rasoul Shafipour, Rich Harang, Rick Izzo, Rima Shahbazyan, Rishabh Garg, Ritika Borkar, Ritu Gala, Riyad Islam, Robert Hesse, Roger Waleffe, Rohit Watve, Roi Koren, Ruoxi Zhang, Russell Hewett, Russell J. Hewett, Ryan Prenger, Ryan Timbrook, Sadegh Mahdavi, Sahil Modi, Samuel Kriman, Sangkug Lim, Sanjay Kariyappa, Sanjeev Satheesh, Saori Kaji, Satish Pasumarthi, Saurav Muralidharan, Sean Narentharen, Sean Narenthiran, Seonmyeong Bak, Sergey Kashirsky, Seth Poulos, Shahar Mor, Shanmugam Ramasamy, Shantanu Acharya, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shiqing Fan, Shreya Gopal, Shrimai Prabhumoye, Shubham Pachori, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Simeng Sun, Smita Ithape, Somshubra Majumdar, Soumye Singhal, Stas Sergienko, Stefania Alborghetti, Stephen Ge, Sugam Dipak Devare, Sumeet Kumar Barua, Suseella Panguluri, Suyog Gupta, Sweta Priyadarshi, Syeda Nahida Akter, Tan Bui, Teodor-Dumitru Ene, Terry Kong, Thanh Do, Tijmen Blankevoort, Tim Moon, Tom Balough, Tomer Asida, Tomer Bar Natan, Tomer Ronen, Tugrul Konuk, Twinkle Vashishth, Udi Karpas, Ushnish De, Vahid Noorozi, Vahid Noroozi, Venkat Srinivasan, Venmugil Elango, Victor Cui, Vijay Korthikanti, Vinay Rao, Vitaly Kurin, Vitaly Lavrukhin, Vladimir Anisimov, Wanli Jiang, Wasi Uddin Ahmad, Wei Du, Wei Ping, Wenfei Zhou, Will Jennings, William Zhang, Wojciech Prazuch, Xiaowei Ren, Yashaswi Karnati, Yejin Choi, Yev Meyer, Yi-Fu Wu, Yian Zhang, Yigong Qin, Ying Lin, Yonatan Geifman, Yonggan Fu, Yoshi Subara, Yoshi Suhara, Yubo Gao, Zach Moshe, Zhen Dong, Zhongbo Zhu, Zihan Liu, Zijia Chen, and Zijie Yan. Nvidia nemotron 3: Efficient and open intelligence, 2025. URL https://arxiv.org/abs/2512.20856.

Olmo Team, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2026. URL https://arxiv.org/abs/2512.13961.

OpenCompass. AIME 2025. https://github.com/open-compass/opencompass, 2025.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ebZcMQImhG.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

Miao Rang, Zhenni Bi, Hang Zhou, Hanting Chen, An Xiao, Tianyu Guo, Kai Han, Xinghao Chen, and Yunhe Wang. Revealing the power of post-training for small language models via knowledge distillation, 2025. URL https://arxiv.org/abs/2509.26497.

Daniel Ritter, Owen Oertell, Bradley Guo, Jonathan D. Chang, Kianté Brantley, and Wen Sun. Llms can learn to reason via off-policy rl, 2026. URL https://arxiv.org/abs/2602.19362.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=qA6FgH0nnZ.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 59839–59864, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/618c95f4557c15b253fb0e6f548ea0c0-Paper-Conference.pdf.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models, 2026. URL https://arxiv.org/abs/2604.00626.

Jie Sun, Mao Zheng, Mingyang Song, Zeyuan Liu, Gengsheng Li, Houcheng Jiang, Yilin Cheng, Bichuan Feng, Yuchen Cai, Junfeng Fang, and Xiang Wang. Distill what you trust: Reliability-aware multi-teacher on-policy distillation, 2026. URL https://arxiv.org/abs/2609.23697.

Boxin Wang, Chankyu Lee, Nayeon Lee, Sheng-Chieh Lin, Wenliang Dai, Yang Chen, Yangyi Chen, Zhuolin Yang, Zihan Liu, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-cascade: Scaling cascaded reinforcement learning for general-purpose reasoning models, 2026. URL https://arxiv.org/abs/2512.13607.

Yaoxiang Wang, Haoling Li, Xin Zhang, Jie Wu, Xiao Liu, Wenxiang Hu, Zhongxin Guo, Yangyu Huang, Ying Xin, Yujiu Yang, Jinsong Su, Qi Chen, and Scarlett Li. EpiCoder: Encompassing diversity and complexity in code generation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 63569–63606. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wang25bi.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 117157–117186. Curran Associates, Inc., 2025. doi: 10.52202/085713-3904. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a9d5c33e38442450f9781bd7bad3c81e-Paper-Conference.pdf.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

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

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, Renjie Pi, Grace Lam, Nayeon Lee, Alexander Bukharin, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation, 2026b. URL https://arxiv.org/abs/2603.19220.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models, 2026. URL https://arxiv.org/abs/2511.10643.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret, 2025. URL https://arxiv.org/abs/2503.01491.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023. URL https://arxiv.org/abs/2308.01825.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 57654–57689. Curran Associates, Inc., 2025a. doi: 10.52202/085713-1933. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu,

Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025b. URL https://arxiv.org/abs/2504.05118.

Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F. Wong, and Yu Cheng. ExGRPO: Learning to reason from experience. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=701tjQXWVk.

Dongxu Zhang, Zhichao Yang, Sepehr Janghorbani, Jun Han, Andrew Ressler, II, Qian Qian, Gregory D. Lyng, Sanjit Singh Batra, and Robert E. Tillman. Fast and effective on-policy distillation from reasoning prefixes. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25553–25569, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1276. URL https://aclanthology.org/2026.findings-acl.1276/.

Wenhao Zhang, Yuexiang Xie, Yuchang Sun, Yanxi Chen, Guoyin Wang, Yaliang Li, Bolin Ding, and Jingren Zhou. On-policy rl meets off-policy experts: Harmonizing supervised fine-tuning and reinforcement learning via dynamic weighting. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 120693–120726, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/c44a04289beaf0a7d968a94066a1d696-Paper-Conference.pdf.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for LLM reasoning. In *Forty-third International Conference on Machine Learning*, 2026c. URL https://openreview.net/forum?id=0RBMhFule7.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Jpxfof0EaS.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In Yixin Cao, Yang Feng, and Deyi Xiong (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 400–410, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-demos.38. URL https://aclanthology.org/2024.acl-demos.38/.

Yuhang Zhou, Kai Zheng, Haoling Li, Dengyun Peng, Can Xu, and Jingjing Chen. Turnopd: Making on-policy distillation turn-aware for efficient long-horizon agent training, 2026. URL https://arxiv.org/abs/2607.05804.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An LLM post-training framework for RL scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

A Theoretical Analysis and Proofs

We establish the cached score objective, prove the teacher-consistent bound and cross-teacher separation in Section 3.3, and give a shared-parameter SGD construction of the separation.

A.1 The Cached Score Objective

Fix a finite prompt distribution p, a finite vocabulary, and a maximum response length H. Let B be the fixed response collection law, $\ell_\theta(s,a) = \log \pi_\theta(a \mid s)$ the student score, and $\ell_T(s,a)$ the teacher score. Scored probabilities are positive, student scores are differentiable, and teacher scores are fixed with respect to θ. For a response x, let $m_t(x)$ be its loss mask and

$$\lambda_t(x) = \frac{m_t(x)}{\max\{1, \sum_j m_j(x)\}}, \qquad A_t = \ell_T(s_t, a_t) - \ell_\theta(s_t, a_t).$$

Thus the update averages active tokens within each response and then averages responses. Padding and masked tokens contribute zero.

Proposition A.1 (Cached score objective)**.** For fixed collection and response weights,

$$\mathcal{L}_B(\theta) = \frac{1}{2}\mathbb{E}_{pB} \sum_t \lambda_t (\ell_\theta - \ell_T)^2, \qquad g_{\text{off}} = \mathbb{E}_{pB} \sum_t \lambda_t A_t \nabla \ell_\theta = -\nabla_\theta \mathcal{L}_B. \tag{11}$$

The same identity holds for a realized cache D with stored teacher scores $c_{i,t}$:

$$\widehat{\mathcal{L}}_D(\theta) = \frac{1}{2|D|} \sum_{i \in D, t} \lambda_{i,t} (\ell_\theta(s_{i,t}, a_{i,t}) - c_{i,t})^2, \qquad \widehat{g}_D = -\nabla_\theta \widehat{\mathcal{L}}_D.$$

Proof. The collection law, stored scores, and response weights are fixed. Differentiating each squared residual gives $\nabla \frac{1}{2}(\ell_\theta - c)^2 = (\ell_\theta - c)\nabla \ell_\theta$. Averaging over tokens and responses proves both identities. □

The cache fixes the teacher targets, while the student's scores change throughout training. A positive teacher–student log-score difference encourages increasing the sampled token's probability; a negative difference encourages decreasing it. The same cached response therefore provides an evolving corrective signal as the student changes. This explains how fixed teacher supervision supports continued distillation without collecting new responses.

Connection to the implemented update. The Qwen configuration takes one optimizer step per 256-response batch, recomputing student scores and detached advantages before each step. At the pre-update parameter $\bar{\theta}$, the PPO ratio $r_\theta = \exp(\ell_\theta - \ell_{\bar{\theta}})$ equals one and lies inside the clipping interval. Its gradient is $\nabla r_\theta|_{\bar{\theta}} = \nabla \ell_{\bar{\theta}}$, so the advantage-weighted update is $A_{\bar{\theta}} \nabla \ell_{\bar{\theta}}$ (Williams, 1992). Averaging these contributions gives the negative gradient of the cached score objective. Adam, gradient norm clipping, and weight decay then transform this data gradient; the following results analyze score SGD.

A.2 Proof of Teacher-Consistent Stability

We prove Theorem 3.1 by controlling two quantities: the teacher-KL risk along each training trajectory and the probability that fixed and adaptive collection produce different trajectories. Accurate fit to the distillation teacher makes these quantities second and first order in the initial score error, respectively. Their product gives the cubic bound.

Score regression and collection schedules. Fix a finite vocabulary and response length H, and a strictly positive teacher U. Prompts may be fixed or drawn from a fixed finite distribution, shared by both training procedures. Write $\ell_\theta(s,a) = \log \pi_\theta(a \mid s)$ for every reachable prefix–token pair. For a sampled response y, use the loss

$$L_U(\theta; y) = \frac{1}{2H} \sum_{t=1}^{H} \big[\ell_\theta(s_t, a_t) - \log U(a_t \mid s_t)\big]^2.$$

Each update uses the H teacher scores on its sampled response. Let Q_θ denote the response-collection law, including the common prompt distribution. It may differ from the untempered model law π_θ. Offline training draws N responses independently from Q_{θ_0} and takes one gradient step on each, in collection order. Online training draws its next response from Q_{θ_i} before each step. Both procedures use the same deterministic step sizes $0 \le \eta_i \le \eta$, N updates, and NH sampled-token teacher scores. The optimizer is SGD on L_U, with no additional update terms. Their evaluation risk is the prompt-averaged response KL $R_U(\theta) = \mathbb{E}_x \mathrm{KL}(\pi_\theta(\cdot \mid x) \| U(\cdot \mid x))$.

Proposition A.2 (Quantitative teacher-consistent bound)**.** Suppose

$$\max_{s,a} \left| \ell_{\theta_0}(s,a) - \log U(a \mid s) \right| \le h. \tag{12}$$

On a parameter ball of radius r_0 around θ_0, assume ℓ_θ is differentiable, $\|\nabla_\theta \ell_\theta(s,a)\| \le G$ for some $G > 0$, and

$$\mathrm{TV}(Q_\theta, Q_{\theta'}) \le L_Q \|\theta - \theta'\|.$$

Define $A_i = (1 + \eta G^2)^i$ and $S_N = \sum_{i=0}^{N-1} (A_i - 1)$. If $h(A_N - 1)/G \le r_0$, then

$$\left| \mathbb{E} R_U(\theta_N^{\mathrm{off}}) - \mathbb{E} R_U(\theta_N^{\mathrm{on}}) \right| \le H(e^{A_N h} - 1)^2 \min\left\{ 1, \frac{L_Q S_N}{G} h \right\}. \tag{13}$$

In particular, if $A_N h \le 1$, this is at most

$$H e^2 A_N^2 \frac{L_Q S_N}{G} h^3. \tag{14}$$

Proof. Step 1: control the score residual along every trajectory. For either collection schedule, let $d_i = \|\theta_i - \theta_0\|$. When θ_i is in the stated ball, the gradient bound along the segment from θ_0 to θ_i gives

$$\max_{s,a} |\ell_{\theta_i}(s,a) - \log U(a \mid s)| \le h + G d_i.$$

Consequently, $\|\nabla L_U(\theta_i; y)\| \le G(h + G d_i)$ for every response y, and

$$d_{i+1} \le (1 + \eta G^2) d_i + \eta G h.$$

Induction gives

$$d_i \le \frac{h}{G}(A_i - 1), \qquad \max_{s,a} |\ell_{\theta_i}(s,a) - \log U(a \mid s)| \le A_i h. \tag{15}$$

The radius condition closes the induction: every subsequent iterate also lies in the ball.

Step 2: bound the final risk. For positive categorical distributions p, q with $|\log(p_a/q_a)| \le b$, the inequalities $\mathrm{KL}(p\|q) \le \chi^2(p,q)$ and $|p_a/q_a - 1| \le e^b - 1$ imply $\mathrm{KL}(p\|q) \le (e^b - 1)^2$. Apply this to every conditional distribution at the final iterate. The autoregressive KL chain rule then yields the pathwise bound

$$0 \le R_U(\theta_N) \le H(e^{A_N h} - 1)^2. \tag{16}$$

Step 3: couple the two collection schedules. Generate the offline cache sequentially using its fixed law Q_{θ_0}. At step i, couple its next response maximally with the online response drawn from $Q_{\theta_i^{\mathrm{on}}}$. This preserves the independent offline-cache marginal and the online conditional sampling law. Before their first unequal response, both procedures have identical parameters because their initial parameters, updates, and preceding responses agree. By Equation (15), the conditional probability of the first disagreement at step i is at most $L_Q h(A_i - 1)/G$. A union bound therefore gives

$$\Pr(\text{at least one unequal response}) \le \min\left\{ 1, \frac{L_Q S_N}{G} h \right\}.$$

With no disagreement, the final risks coincide. Otherwise, both risks lie in the interval in Equation (16), so the absolute difference is at most its upper endpoint. Taking expectations proves Equation (13). Finally, $e^z - 1 \le ez$ for $0 \le z \le 1$ proves Equation (14). □

From the quantitative bound to Theorem 3.1. Fix the update budget N, response length H, and step-size bound η, and consider initializations satisfying Equation (12) as $h \to 0$. With uniform local constants G, L_Q, r_0, both the radius condition and $A_N h \le 1$ hold for sufficiently small h. Equation (14) then proves the theorem with $C_N = He^2 A_N^2 L_Q S_N / G$. The argument permits arbitrary sharing of the student parameters.

The collection condition includes fixed-temperature sampling. If each conditional of Q_θ is proportional to $\pi_\theta(a \mid s)^{1/\tau}$ for fixed $\tau > 0$, its log-score gradient has norm at most $2G/\tau$. Summing over H tokens and using $\mathrm{TV}(Q_\theta, Q_{\theta'}) \le \frac{1}{2} \sup_\vartheta \mathbb{E}_{Q_\vartheta} \|\nabla \log Q_\vartheta\| \|\theta - \theta'\|$ gives $L_Q \le HG/\tau$, where the supremum is over the parameter segment. Thus the collection distribution and the untempered evaluation distribution may differ.

A.3 Proof of the Cross-Teacher Separation

We prove Theorem 3.3 in a two-token model that separates the teacher's effect on visitation from its effect on the supervision required at each visited context. The proof first expresses the offline–online gap through the number of supervised visits, then compares this gap across teachers. Both collection schedules use the same finite update and feedback budgets.

Model and updates. Each response consists of two tokens (X, Y), with $X \in \{L, R\}$ and $Y \in \{0, 1\}$. The initial student B has $B(R) = \varepsilon$, $B(Y = 1 \mid R) = q$, and $B(Y = 1 \mid L) = \tau$. A target teacher $V = (v, s)$ has $V(R) = v$, $V(Y = 1 \mid R) = s$, and $V(Y = 1 \mid L) = \tau$. All probabilities are in $(0, 1)$; the root and branch conditionals have independent tabular parameters. Generation uses these probabilities directly. Each response supplies two sampled-token teacher log-scores. Because each distribution is binary, either root score identifies v, and either second-token score on R identifies s.

Fix root and conditional fitting rates $\rho, \gamma \in (0, 1]$. After each training response, update the root probability p and the R conditional z by

$$p^+ = (1 - \rho)p + \rho v, \qquad z^+ = \begin{cases} (1 - \gamma)z + \gamma s, & X = R, \\ z, & X = L. \end{cases} \tag{17}$$

The L conditional remains at τ. Offline training first collects N independent responses from B, then processes them once in their collection order. Online training alternates one response from the current student with one update. Both procedures use N updates and $2N$ sampled-token teacher scores. Rates below one give incomplete fitting; $\rho = \gamma = 1$ fits each observed coordinate in one update. This damped probability update decreases the sampled teacher-score regression loss: each observed coordinate moves toward its teacher target, reducing the log-score residual for either binary outcome.

Let $P^V_{\mathrm{off}}, P^V_{\mathrm{on}}$ be the final students and define the additional error from fixed collection under the same teacher:

$$D_B(V) = \mathbb{E}\,\mathrm{KL}(P^V_{\mathrm{off}} \| V) - \mathbb{E}\,\mathrm{KL}(P^V_{\mathrm{on}} \| V). \tag{18}$$

The KL divergences are over complete two-token responses, with expectations over the respective training samples.

Proposition A.3 (The offline–online gap is a visitation effect)**.** Define

$$p_i^V = v + (\varepsilon - v)(1 - \rho)^i, \qquad \psi_{q,s}(k) = \mathrm{KL}\big(\mathrm{Ber}\big(s + (q - s)(1 - \gamma)^k\big) \,\big\|\, \mathrm{Ber}(s)\big)\,.$$

If $K_{\mathrm{off}} \sim \mathrm{Bin}(N, \varepsilon)$ and K_{on} is a sum of independent Bernoulli variables with success probabilities $p_0^V, \ldots, p_{N-1}^V$, then

$$D_B(V) = p_N^V \left[\mathbb{E}\psi_{q,s}(K_{\mathrm{off}}) - \mathbb{E}\psi_{q,s}(K_{\mathrm{on}})\right]. \tag{19}$$

For $N \ge 2$, $q \ne s$, and $\rho, \gamma > 0$, the sign of $D_B(V)$ is the sign of $v - \varepsilon$. When $v > \varepsilon$,

$$D_B(V) \ge p_N^V \rho\gamma(v - \varepsilon)(1 - \varepsilon)(1 - v)^{N-2} \mathrm{KL}(\mathrm{Ber}(q) \| \mathrm{Ber}(s)). \tag{20}$$

Proof. The root update is independent of the observed branch, so both procedures have the deterministic root sequence p_i^V. After k visits to R, the conditional probability is $s + (q - s)(1 - \gamma)^k$. Offline branch indicators are independent Bernoulli(ε) variables. Online root probabilities depend only on

the number of previous updates, so its branch indicators are independent Bernoulli variables with parameters p_i^V. The chain rule gives each final risk as

$$\mathrm{KL}(\mathrm{Ber}(p_N^V)\|\mathrm{Ber}(v)) + p_N^V\,\mathbb{E}\psi_{q,s}(K).$$

The common root term cancels in their difference, proving Equation (19).

For $v > \varepsilon$, couple the indicators using independent $Z_i \sim \mathrm{Unif}(0,1)$:

$$I_i^{\mathrm{off}} = \mathbf{1}\{Z_i < \varepsilon\}, \qquad I_i^{\mathrm{on}} = \mathbf{1}\{Z_i < p_{i-1}^V\}.$$

Then $K_{\mathrm{on}} \geq K_{\mathrm{off}}$ almost surely. The function $\psi_{q,s}(k)$ is nonincreasing in k, since its first argument moves along the line segment from q to s. For $q \neq s$, its first decrease is strict. When $\gamma < 1$ every decrease is strict; when $\gamma = 1$ it is zero after the first visit.

Consider the event

$$Z_1 > \varepsilon, \quad \varepsilon < Z_2 < p_1^V, \quad Z_i > p_{i-1}^V \quad (3 \leq i \leq N).$$

On this event $K_{\mathrm{off}} = 0$ and $K_{\mathrm{on}} = 1$. Its probability is

$$(1-\varepsilon)\rho(v-\varepsilon)\prod_{i=2}^{N-1}(1-p_i^V) \geq (1-\varepsilon)\rho(v-\varepsilon)(1-v)^{N-2}.$$

Convexity of KL in its first argument gives $\psi_{q,s}(1) \leq (1-\gamma)\psi_{q,s}(0)$. All other coupled outcomes contribute nonnegative differences. Multiplying this event's contribution by p_N^V proves Equation (20) and strict positivity.

When $v < \varepsilon$, the same coupling reverses the ordering of the visit counts. The event $Z_1 > \varepsilon$, $p_1^V < Z_2 < \varepsilon$, and $Z_i > \varepsilon$ for $i \geq 3$ has positive probability and gives $K_{\mathrm{on}} = 0$, $K_{\mathrm{off}} = 1$. Thus the difference is strictly negative. If $v = \varepsilon$, the count distributions coincide and $D_B(V) = 0$. □

Equation (19) isolates the source of the gap. The root error is identical under the two schedules, while the conditional error depends on how often the changing branch is visited. When $v > \varepsilon$, adaptive collection supplies more supervision to that branch, giving a positive gap whenever its continuation also requires an update.

Consistent and changed teachers. Fix an SFT teacher $U = (\alpha, u)$ and a changed teacher $T = (\beta, r)$, with $\beta > \alpha$ and $r \neq u$. Fix $N \geq 2$ and $\rho, \gamma \in (0,1]$. Let the initialization $B_h = (\varepsilon_h, q_h)$ satisfy

$$|\varepsilon_h - \alpha| \leq h, \qquad |q_h - u| \leq h,$$

with the same L conditional as both teachers. Define

$$A_N(\rho) = \sum_{i=0}^{N-1}\left[1-(1-\rho)^i\right] = N - \frac{1-(1-\rho)^N}{\rho}.$$

Then the matched-teacher penalty obeys

$$|D_{B_h}(U)| \leq \frac{A_N(\rho)}{u(1-u)}h^3. \tag{21}$$

For $h < \min\{\alpha, 1-\alpha, \beta-\alpha, |r-u|\}$,

$$\begin{aligned} D_{B_h}(T) \geq\ & 2(\alpha-h)\rho\gamma(\beta-\alpha-h)(1-\alpha-h) \\ & \cdot (1-\beta)^{N-2}(|r-u|-h)^2. \end{aligned} \tag{22}$$

Consequently, for every sufficiently small positive h,

$$D_{B_h}(T) > D_{B_h}(U).$$

These are quantitative versions of the two claims in Theorem 3.3.

Proof of Theorem 3.3. Apply the visit-count coupling to the target U. The two indicator vectors differ with probability at most

$$\sum_{i=0}^{N-1} |p_i^U - \varepsilon_h| = |\alpha - \varepsilon_h| A_N(\rho) \le h A_N(\rho).$$

Since $0 \le \psi_{q_h,u}(k) \le \psi_{q_h,u}(0)$, the absolute difference of its expectations is at most $hA_N(\rho)\psi_{q_h,u}(0)$. The Bernoulli inequality

$$\mathrm{KL}(\mathrm{Ber}(q_h)\|\mathrm{Ber}(u)) \le \frac{(q_h-u)^2}{u(1-u)} \le \frac{h^2}{u(1-u)}$$

and $p_N^U \le 1$ prove Equation (21).

For the changed teacher, $\beta > \varepsilon_h$ and $p_N^T \ge \varepsilon_h \ge \alpha - h$. Furthermore, $\beta - \varepsilon_h \ge \beta - \alpha - h$ and $1 - \varepsilon_h \ge 1 - \alpha - h$. Pinsker's inequality for Bernoulli distributions gives

$$\mathrm{KL}(\mathrm{Ber}(q_h)\|\mathrm{Ber}(r)) \ge 2(q_h - r)^2 \ge 2(|r-u| - h)^2.$$

Substituting into Equation (20) proves Equation (22). Its lower bound tends to a positive constant, whereas the upper bound in Equation (21) tends to zero. This proves the strict comparison. The finite count distributions and their KL risks are continuous in the initialization, which also proves the stated positive limit. □

The role of consistency. Accurate fit to U reduces visitation changes to $O(h)$ and conditional KL to $O(h^2)$, suppressing the additional offline error. A changed teacher can retain both a visitation shift and a continuation error, giving online collection an advantage at the same feedback budget.

A.4 Teacher Switching under the Same SGD Update

The consistent-teacher bound controls the magnitude of the offline–online gap. We now characterize its sign after a teacher switch using the same score-regression update. Use the loss, collection law Q_θ, and response-KL risk from Appendix A.2, with target T. Both learners take two SGD steps of size η: offline uses two independent responses from Q_{θ_0}, while online samples its second response after the first update. Each uses two responses and $2H$ sampled-token teacher scores. Write $g_y(\theta) = \nabla L_T(\theta; y)$ and define, at initialization,

$$r = \nabla R_T(\theta_0), \qquad \bar{g} = \mathbb{E}_{Q_{\theta_0}} g_y(\theta_0), \qquad C = \mathbb{E}_{Q_{\theta_0}}\left[g_y(\theta_0) \nabla \log Q_{\theta_0}(y)^\top\right].$$

Here $\bar{g}$ is the expected loss gradient, C is the contribution of the collection law to its parameter derivative, and r is the teacher-KL gradient. Their alignment determines the leading effect of changing the collection schedule.

Proposition A.4 (The leading effect of adaptive collection)**.** On a finite response space, suppose $L_T(\cdot; y)$, $Q_\theta(y)$, and R_T are twice continuously differentiable near θ_0, with $Q_\theta(y) > 0$ there. For $D_T(\eta; \theta_0) = \mathbb{E}R_T(\theta_2^{\mathrm{off}}) - \mathbb{E}R_T(\theta_2^{\mathrm{on}})$,

$$D_T(\eta; \theta_0) = -\eta^2 r^\top C \bar{g} + O(\eta^3). \tag{23}$$

Consequently, $\kappa = -r^\top C\bar{g} > 0$ implies $D_T(\eta; \theta_0) > 0$ for every sufficiently small $\eta > 0$.

Proof. For responses x, y, put $\theta_x = \theta_0 - \eta g_x(\theta_0)$ and $\theta_{xy} = \theta_x - \eta g_y(\theta_x)$. Because the two second-response laws have unit mass,

$$D_T = \sum_{x,y} Q_{\theta_0}(x)[Q_{\theta_0}(y) - Q_{\theta_x}(y)][R_T(\theta_{xy}) - R_T(\theta_x)].$$

Uniformly over the finite response space, the two brackets equal $\eta \nabla Q_{\theta_0}(y)^\top g_x(\theta_0) + O(\eta^2)$ and $-\eta r^\top g_y(\theta_0) + O(\eta^2)$. Multiplication and summation give $-\eta^2 r^\top \big(\sum_y g_y(\theta_0) \nabla Q_{\theta_0}(y)^\top\big)\bar{g} + O(\eta^3)$, which is Equation (23).

If $\kappa > 0$, the leading term dominates the remainder for sufficiently small positive η, proving the sign claim. □

A positive cross-teacher separation. The coefficient can be positive even when both teachers belong to the student model family. Consider a two-token model with $X \in \{L, R\}$ and $Y \in \{0, 1\}$:

$$\pi_\theta(R) = \sigma(\mathrm{logit}(1/5) + \theta),$$
$$\pi_\theta(Y = 1 \mid R) = \sigma(\mathrm{logit}(7/20) + \theta), \qquad \pi_\theta(Y = 1 \mid L) = 1/2.$$

The root and right continuation share one trainable parameter. Take $Q_\theta = \pi_\theta$, $\theta_0 = 0$, $U = \pi_0$ and $T = \pi_{\log 6}$. Both teachers belong to the student family; $T(R) = 3/5$ and $T(Y = 1 \mid R) = 42/55$. For the loss $L_T = \frac{1}{4}\sum_{t=1}^{2}(\log \pi_\theta - \log T)^2$, direct evaluation gives

$$\bar{g} = -\frac{411}{4000}\log 6 < 0,$$
$$r = \frac{203}{2000}\log(11/72) + \frac{117}{2000}\log(11/12) - \frac{4}{25}\log 2 < 0,$$
$$C = -\frac{8}{125}\log 3 + \frac{2639}{80000}\log(11/24) - \frac{819}{80000}\log(11/4) + \frac{2}{125}\log 2 < 0.$$

For C, the middle terms are negative and the first plus last terms are negative since $3^4 > 2$. Hence $\kappa = -rC\bar{g} \approx 0.0054 > 0$.

For teacher U, every sampled gradient vanishes at $\theta_0 = 0$: both schedules stay at the same fixed point and $D_U(\eta; 0) = 0$. Fix a sufficiently small positive step size such that $D_T(\eta; 0) > 0$, and perturb the initial parameter to $\delta \to 0$. Continuity of the finite two-update expectations gives $D_T(\eta; \delta) \to D_T(\eta; 0) > 0$. The uniform SFT log-score error h_δ obeys $h_\delta \le |\delta|$, since all conditional log-score derivatives have magnitude at most one. Proposition A.2 applies to these same SGD updates, so $|D_U(\eta; \delta)| = O(h_\delta^3)$. Thus $D_T(\eta; \delta) > D_U(\eta; \delta)$ for all sufficiently small $|\delta|$, including nonzero SFT error. This gives a cross-teacher separation with shared parameters and a common score-SGD optimizer: improving the fit to U removes the additional offline error for U, while the changed teacher retains a positive gap at the same fixed step size and feedback budget.

B Implementation Details

B.1 Qwen Training

Algorithm 1 summarizes Lightning OPD. Tables 4 and 5 list the SFT and distillation hyperparameters for Qwen3-4B and Qwen3-8B. The MoE experiment follows the 8B distillation settings on one 8×H100 node. Math distillation starts from the SFT checkpoint; code distillation starts from the math-trained checkpoint, following the math-to-code progression explored by prior work (Wang et al., 2026).

Offline preprocessing stores response tokens and the teacher's log-probability for each sampled token, with a 4,096-token response cap. This requires one scalar teacher score per token, rather than a full-vocabulary distribution. During training, the student recomputes its log-probabilities at temperature 1 and uses cached teacher scores minus current student scores as detached token advantages. Losses are averaged within responses and then across the 256-response batch, with one optimizer step per batch. Advantages are unnormalized, and PPO ratio clipping uses $[0.8, 1.2]$.

Table 4: SFT hyperparameters for Qwen3 students.

Hyperparameter	**Qwen3-4B**	**Qwen3-8B**
Training steps	3000	3000
Global batch size	256	128
Max sequence length	16384	16384
Learning rate	8×10^{-5}	8×10^{-5}
LR schedule	cosine	cosine
Warmup ratio	0.1	0.1
Packing	✓	✓
DeepSpeed stage	ZeRO-0	ZeRO-1

Algorithm 1 Lightning OPD

Require: Base model, teacher, SFT and OPD prompts, scoring convention
1: Generate teacher SFT responses and fine-tune to obtain π_{ref}
2: Choose student initialization θ_0 and fixed collection law B
3: **for** each OPD prompt q^j **do**
4: Sample $x^j \sim B(\cdot \mid q^j)$; cache teacher scores c_t^j
5: **end for**
6: Initialize the student at θ_0
7: **for** each training batch $\mathcal{B}$ **do**
8: Compute current scores ℓ_θ and $A_t^j = c_t^j - \ell_\theta(s_t^j, a_t^j)$
9: Form $\widehat{g} = |\mathcal{B}|^{-1} \sum_{j \in \mathcal{B}} \sum_t \lambda_t(x^j)\, \mathrm{stopgrad}(A_t^j) \nabla_\theta \ell_\theta(s_t^j, a_t^j)$
10: Apply one optimizer step using $-\widehat{g}$ as the data gradient
11: **end for**

Table 5: Distillation hyperparameters for Qwen3 students.

Hyperparameter	**Qwen3-4B**	**Qwen3-8B**
Training steps	150	150
Global batch size	256	256
Training response cap	4096	4096
Learning rate	2×10^{-6}	2×10^{-6}
LR schedule	constant	constant
Weight decay	0.1	0.1
Adam β_1 / β_2	0.9 / 0.98	0.9 / 0.98
Tensor parallel size	2	4
PPO ratio clip interval	$[0.8, 1.2]$	$[0.8, 1.2]$
Advantage clipping / normalization	none / off	none / off
Gradient norm clip	1.0	1.0
Updates per offline anchor batch	1	1

B.2 OLMo Training

We initialize from `Olmo-3-7B-Think-SFT` and use `Olmo-3-32B-Think` as the teacher (Olmo Team et al., 2026). Both math runs use 150 optimizer updates with a global batch size of 256 and a 4,096-token response cap. The student is trained with FSDP and Adam at a constant learning rate of 2×10^{-6}, $(\beta_1, \beta_2) = (0.9, 0.98)$, $\epsilon = 10^{-8}$, weight decay 0.1, and gradient norm clipping at 1.0. The loss is response-normalized, with ratio clipping $[0.8, 1.2]$ and unnormalized advantages.

Lightning OPD collects one response per DAPO-Math prompt, producing 17,398 cached responses. Standard OPD samples four responses for each of 64 prompts per update. Both use generation temperature 0.8 and top-p 1. Student scoring temperatures are 1.0 for Lightning OPD and 0.8 for standard OPD; teacher targets are untempered log-probabilities. Student and teacher inputs use their respective chat templates with aligned response tokens.

For code distillation, each method starts from its own math checkpoint. Lightning OPD caches one response for each of the 30,000 EpiCoder prompts and takes 150 optimizer updates. Standard OPD takes 150 updates with four responses per prompt. Both retain the math-stage batch size, generation settings, response cap, and optimizer settings, with an 8,192-token packed budget per GPU. Evaluation follows Section 4.1 using the checkpoint's native chat template.

B.3 Training Cost

Table 2 reports allocated H100 GPU hours, covering rollout collection, teacher scoring, and student training. SFT and benchmark evaluation are excluded. Standard OPD components apportion total cost by configured role GPU counts; Lightning OPD components correspond to sequential phases.

B.4 Evaluation Templates

The following templates are used for Qwen evaluation; OLMo uses its own checkpoint chat template as specified in Appendix B.2. For mathematical reasoning benchmarks, the prompt follows the Qwen3 chat format:

```
<|im_start|>user
Question: {problem}
Please reason step by step, and put your final answer within \boxed{}.
<|im_end|>
<|im_start|>assistant
```

For code generation benchmarks, the prompt includes a system message and a task description following LiveCodeBench conventions:

```
<|im_start|>system
You are a helpful and harmless assistant. You are Qwen developed by
↪  Alibaba.
You should think step-by-step.
<|im_end|>
<|im_start|>user
You will be given a question (problem specification) and will generate a
↪  correct
Python program that matches the specification and passes all tests.

Question: {question}

Read the inputs from stdin solve the problem and write the answer to
↪  stdout
(do not directly test on the sample inputs). Enclose your code within
↪  delimiters
as follows. Ensure that when the python program runs, it reads the inputs,
↪  runs
the algorithm and writes output to STDOUT.
# YOUR CODE HERE

<|im_end|>
<|im_start|>assistant
```

C Additional Experimental Results

The supplementary experiments address two questions: whether cached rollouts need refreshing, and how teacher consistency affects learning under fixed collection. We examine task performance under different refresh schedules, then compare initial teacher fit, student drift, and the offline–online gap.

C.1 Effect of Rollout Refresh

We examine whether periodically refreshing the rollout cache improves Lightning OPD. Starting from the same Qwen3-4B SFT checkpoint and initial cache, we replace the cached responses every 50 or 30 updates using the current student and recompute their scores with the same Qwen3-8B teacher. Both variants retain the optimizer state throughout 150 updates and use the default batch size, learning rate, and response cap. Each refresh collects one response for every DAPO-Math prompt at temperature 0.8 and top-p 1. Evaluation follows Section 4.1.

Refreshing exposes the teacher to trajectories from later student checkpoints. Holding the number of student updates fixed tests whether this additional supervision improves the final model enough to justify repeated rollout collection and teacher scoring.

Table 6 shows that the fixed cache achieves 68.1% on AIME 2024, compared with 64.7% and 63.9% when refreshing every 50 and 30 updates. These schedules add two and four full rounds of rollout collection and teacher scoring, respectively. Periodic refresh therefore provides no AIME 2024 gain in this teacher-consistent setting, supporting one-time collection and reuse of teacher supervision.

Table 6: Effect of rollout refresh on Qwen3-4B with a consistent Qwen3-8B teacher. All runs use 150 student updates. Periodic refresh replaces the entire offline response pool.

Method	Rollout refresh	AIME 2024 (%)
Standard OPD	Every update	65.4
Lightning OPD	Never	**68.1**
+ refresh	Every 50 updates	64.7
+ refresh	Every 30 updates	63.9

C.2 Initial Teacher Fit and Student Drift

Teacher consistency affects both the initial fit to the distillation target and the student's subsequent drift. At each scale, we hold the Qwen-SFT initialization fixed and compare its original teacher with QwQ-32B. Table 7 shows that changing the teacher increases the initial score error and leads to greater drift during offline training, complementing the benchmark ablation in Table 3.

Table 7: Initial teacher fit and subsequent student drift under offline distillation. Each student uses its Qwen-SFT initialization. Initial score error is measured before distillation; drift is measured after 150 updates at 4B and 50 updates at 8B. Lower values indicate closer initial teacher fit and smaller changes from the SFT reference, respectively.

Student	OPD teacher	Initial score error	Student KL
Qwen3-4B	Qwen3-8B	0.23	0.048
	QwQ-32B	0.49	0.23
Qwen3-8B	Qwen3-32B	0.35	0.052
	QwQ-32B	0.45	0.14

The initial-fit comparison changes the scoring teacher while keeping the student and cached responses fixed, directly measuring the effect of changing the supervision target. The drift comparison then measures how far the trained student moves from the same SFT reference. Together, these quantities connect teacher choice to the amount of adaptation observed during offline distillation.

Initial score error is the mean squared student–teacher log-probability difference on 2,048 cached responses, averaged first over tokens and then over responses. For the drift diagnostic in Figure 3(a), we use 128 cached responses that reach the 4,096-token cap, selecting 32 evenly spaced positions per response. We compute $\mathrm{KL}(\pi_{\mathrm{ref}} \| \pi_\theta)$ from the full-vocabulary, untempered next-token distributions on these 4,096 fixed states. The same states and SFT reference are used for both teacher choices within each scale. These measurements summarize average fit and drift; the theoretical bound in Theorem 3.1 is stated in terms of uniform initial score error.

C.3 Teacher Consistency and Collection Schedules

We examine how teacher consistency affects the difference between offline and online distillation. Two Qwen3-4B students are initialized by 3,000 SFT updates on demonstrations from Qwen3-8B or QwQ-32B. Each student is distilled with both teachers, using either a fixed cache or fresh student responses before each update. Within each SFT initialization, the four configurations share their starting weights, prompt order, optimizer settings, and teacher-feedback budget. Crossing both teacher choices allows us to examine consistency from either initialization, while the paired collection schedules measure the additional effect of freezing responses at a fixed training budget.

Each configuration starts a new Adam stage and takes 20 updates with batch size 256, learning rate 2×10^{-6}, betas $(0.9, 0.98)$, and weight decay 0.1. We traverse 512 training prompts ten times, sampling a new response for each occurrence at temperature 0.8 and top-p 1. Every response contains exactly 512 tokens, with EOS termination disabled. Offline collection generates all responses from the initial student and uses each once; online collection generates each batch from the current student. Both schedules therefore use 5,120 responses and the same number of teacher-token scores. Teacher scores are raw, untempered log-probabilities.

Evaluation uses 128 separate prompts, each with one frozen 512-token response from each of the two SFT students and two teachers. All checkpoints are evaluated on this same response mixture against their distillation teacher. We measure the per-token score loss $\frac{1}{2}(\log \pi_\theta - \log \pi_T)^2$, averaged within each response and then equally across collectors and prompts. The offline–online gap is $G = L_{\text{off}} - L_{\text{on}}$, so a positive gap indicates lower score loss with online collection.

Table 8: Teacher consistency and the offline–online gap. Initial and final score losses are evaluated against each row's distillation teacher on a shared frozen response mixture. All runs use 20 updates and equal teacher-feedback budgets. The gap is the offline loss minus the online loss, computed before rounding.

SFT teacher	OPD teacher	Initial loss	Offline	Online	Gap G
Qwen3-8B	Qwen3-8B	0.40	0.34	0.34	0.0034
	QwQ-32B	0.72	0.33	0.32	0.010
QwQ-32B	Qwen3-8B	1.2	1.6	1.2	0.46
	QwQ-32B	0.71	0.64	0.71	-0.069

Table 8 shows that changing the teacher increases the offline–online gap for both SFT initializations. For the Qwen-initialized student, the gap increases from 0.0034 to 0.010; for the QwQ-initialized student, it increases from -0.069 to 0.46. The larger change in the latter direction shows that the effect depends on the initialization–teacher pairing. With QwQ-SFT, offline collection gives the lower final score loss with the consistent teacher, while online collection gives the lower loss after changing the target to Qwen3-8B. With Qwen-SFT, both schedules improve the initial target-score loss for either teacher, and retaining the SFT teacher gives the smaller gap. Together, these results support the role of teacher consistency in limiting the additional score-fitting error from fixed collection, complementing the benchmark and drift comparisons in Section 4.4.

D Discussion

Relation to Offline RL. Lightning OPD shares the separation of data collection and policy optimization with offline RL (Levine et al., 2020). Value-based offline RL methods address errors that arise from learning and using value estimates with limited data coverage (Kumar et al., 2020; Kostrikov et al., 2022; Fujimoto et al., 2019). Lightning OPD obtains its targets directly from the teacher's log-probabilities on sampled tokens, enabling student updates without a bootstrapped value function. The relevant question is how well the fixed student trajectories support further matching of the chosen teacher. Our score-SGD analysis connects this question to SFT: accurate initial teacher fit controls the additional error from freezing collection, while a changed teacher can induce new context preferences and supervision requirements. Teacher consistency thus provides a principle for coordinating initialization with the targets used during offline learning.

Relation to Offline Knowledge Distillation. Sequence-level distillation trains on teacher-generated demonstrations (Kim & Rush, 2016), while methods such as GKD and MiniLLM (Agarwal et al., 2024; Gu et al., 2024) obtain teacher supervision on student-generated contexts (Zhang et al., 2026c; Li et al., 2026b; Fu et al., 2026a; Jin et al., 2026). Lightning OPD collects responses from the initial student and caches the teacher's log-probabilities on those responses. This supplies feedback on trajectories the student produces, including its own mistakes. The cached response determines the context and sampled token; the current teacher–student score difference determines the update. As the student changes, its advantages are recomputed against the same teacher targets. This retains supervision on student-generated contexts while allowing the response generation, teacher scoring, and student optimization stages to run separately.

Relation to Offline On-Policy Distillation. Rang et al. (2025) also precompute teacher supervision on student-generated responses. Their method stores top-k teacher logits and combines cross-entropy with a KL distillation loss. Lightning OPD stores one teacher log-probability per sampled token and uses the current teacher–student score difference for its advantage-weighted update. Our central question is how the teacher used for SFT affects the reuse of these fixed trajectories. The teacher-consistent bound and cross-teacher separation characterize how initial fit changes the additional

error from offline collection; the teacher-pairing experiments examine this relationship in reasoning models. Rang et al. (2025) also report benefits from periodically regenerating student responses in their setting. Our refresh experiment in Appendix C.1 shows that a fixed cache retains the strongest AIME 2024 result in the evaluated teacher-consistent configuration, supporting one-time supervision collection for this pipeline.

E LIMITATIONS

Our experiments cover mathematical reasoning and code generation, with teacher-pairing evidence from the evaluated Qwen combinations. Teacher consistency coordinates supervision across stages; its benefit depends on the fit achieved during SFT and the teachers and tasks involved. Adopting a new teacher may require regenerating demonstrations and repeating SFT, whose costs are separate from the reported distillation savings. Extending cached supervision to multi-turn interaction, tool use, and open-ended instruction following calls for studying the changing contexts encountered in these settings.