WHEN MIXING BEATS EITHER POLICY: RETHINKING ROLLOUT SAMPLING IN LARGE LANGUAGE MODEL DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

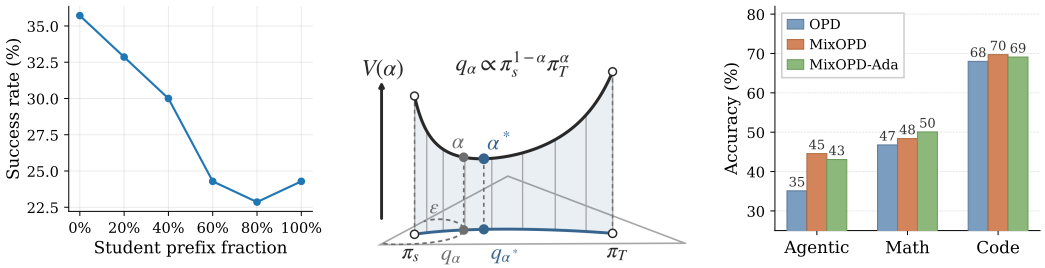
On-policy distillation (OPD) trains a student language model on its own rollouts while using a teacher's logits as supervision. However, sampling entirely from the student has two shortcomings: the teacher's guidance becomes unreliable on student-generated trajectories that deviate from the teacher's distribution, and on hard problems the student rarely produces correct paths for the teacher to reinforce. Sampling from the teacher instead incurs high variance in the gradient correction and risks catastrophic forgetting. In this paper, we prove that neither is optimal: the gradient variance of OPD is strictly convex in the student–teacher mixing strength, and its minimum lies strictly between pure student and pure teacher sampling. Building on this theoretical result, we propose MIXOPD, which replaces the on-policy sampling distribution with a geometric student–teacher mixture. This method includes two variants: a fixed mixing strength version and MIXOPD-ADA, which adapts the mixing strength at each token using the entropy gap between the two models. We implement both variants natively inside the vLLM inference engine with only 39% wall-clock overhead over vanilla OPD. Experiments on agentic, mathematical reasoning, and code generation tasks across two student sizes show that both variants consistently outperform standard OPD by an average of 4 points, indicating that the choice of rollout sampling distribution is a substantial and underexplored lever for improving distillation.

1 INTRODUCTION

Post-training a large language model typically combines supervised fine-tuning with reinforcement learning, but both approaches face fundamental limitations. Supervised fine-tuning on teacher-generated data cannot teach the student to recover from its own mistakes (Ross et al., 2011), while reinforcement learning without teacher guidance is sample-inefficient and prone to reward hacking (Agarwal et al., 2024). On-policy distillation (OPD) bridges the two by training the student on its own rollouts while using the teacher's token-level logits as supervision (Agarwal et al., 2024; Lin et al., 2020; Sun et al., 2026; Xu et al., 2025; Gu et al., 2023; Song & Zheng, 2026), which removes the train–test state mismatch that makes offline distillation brittle (Ross et al., 2011). Despite its widespread adoption, OPD has two algorithmic shortcomings that have not been adequately addressed.

The first is the unreliability of the teacher's supervision signal. When the student's output trajectory has already deviated from the teacher's normal behaviour, the teacher's token-level guidance may no longer align with the true reward signal (Liu et al., 2026; Wang et al., 2026). In extreme cases, the student can produce near-degenerate text that exploits a spurious agreement between its own distribution and the teacher's, under out-of-distribution inputs. This phenomenon is analogous to reward hacking in RLHF and can lead to training collapse (Tiapkin et al., 2025; Fu et al., 2026). The second is low sample efficiency on hard problems (Cai et al., 2026; Liao et al., 2026). On tasks where the student's success rate is low, the student-generated trajectories are predominantly incorrect, and the teacher can do little more than flag individual token errors along an already doomed path, rather than demonstrating a complete correct solution. Both issues stem from the same root cause: sampling entirely from the student confines the training signal to the student's own distribution, which is not necessarily where correction is most valuable.

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



(a) Degradation of Teacher Feedback on Prolonged Student Trajectories.

(b) Gradient Variance vs. mixing coefficient of the sampling distribution.

(c) Experimental Results.

Figure 1: **(a)** Teacher success rate on ALFWorld (Shridhar et al., 2020) as a function of how many tokens the student generates before handing off. For each problem the student (Qwen3-4B) generates the first k tokens and the teacher (Qwen3-14B) completes the rest; accuracy falls monotonically with k. **(b)** Gradient variance of OPD as the sampling distribution moves from pure student to pure teacher. The variance-minimising mixing coefficient lies strictly between the two endpoints. **(c)** Performance of MIXOPD and MIXOPD-ADA against OPD baselines on Qwen3-4B across agentic, math, and code benchmarks.

A natural alternative is to sample directly from the teacher, which would yield higher-quality trajectories, but it introduces its own problems: the importance weights required to keep the gradient unbiased have high variance (Munos et al., 2016), and the large update magnitudes can cause catastrophic forgetting of the student's existing capabilities (Agarwal et al., 2024). Neither pure student nor pure teacher sampling is satisfactory.

In this paper, we investigate this trade-off both empirically and theoretically. On the empirical side, we conduct an experiment on ALFWorld (Figure 1a): the student generates the first few tokens of each problem, then the teacher takes over. The teacher's accuracy drops with the proportion that the student generates, confirming that the teacher's supervision becomes unreliable as the student-generated prefix grows. On the theoretical side, we consider a family of sampling distributions that interpolate between the student and the teacher through a single mixing coefficient. We prove that the gradient variance of OPD is strictly convex in this coefficient (Figure 1b), and that under conditions generic in distillation the variance-minimising point lies strictly between the two endpoints. In other words, some strict mixture of the student and teacher provably achieves lower gradient variance than either endpoint alone.

Building on this analysis, we propose MIXOPD, which replaces the standard on-policy sampling distribution with a geometric mixture of the student and the teacher, simultaneously addressing low-quality generation on hard problems and the high variance of off-policy correction. We instantiate this idea in two variants. MIXOPD uses a fixed mixing ratio across all positions, which is simple to tune and already captures most of the benefit. MIXOPD-ADA further adapts the mixing strength at the token level by using the entropy gap between the student and the teacher as a proxy for where mixing is most valuable, concentrating the mixing budget on positions where the student is uncertain but the teacher is confident. Empirically, we evaluate both variants on agentic, mathematical reasoning, and code generation tasks, and find that they consistently outperform standard OPD by an average of 4 points across all domains (Figure 1c). We modify the vLLM (Kwon et al., 2023) inference engine to run both models within a single rollout pass, reusing the teacher logits computed during sampling to eliminate the separate reference forward pass that standard OPD requires. The resulting implementation adds only 39% wall-clock overhead over vanilla OPD, which is practically acceptable. Together, these results suggest that the choice of rollout sampling distribution is a substantial and underexplored lever for improving distillation.

Contributions.

- For theoretical insights, we consider sampling from a mixture of teacher and student policies, and prove that the gradient variance of the importance-weighted estimator in OPD is strictly convex in the mixing strength. Under realistic conditions in LLM distillation, we further show that the variance-minimizing sampling distribution is a strict mixture of the student and teacher rather than either one alone.

- Algorithmically, we propose MIXOPD, which replaces the on-policy sampling distribution with a geometric student–teacher mixture. We provide two instantiations: a fixed mixing strength variant and MIXOPD-ADA, which adapts the mixing strength at each token using the entropy gap between the two models.

- Last but not least, we present a practical implementation of geometric mixture sampling inside the vLLM inference engine, running both models within a single rollout pass with only 39% wall-clock overhead. We validate both variants on agentic, mathematical reasoning, and programming benchmarks across two student sizes, showing consistent improvements over standard OPD.

2 RELATED WORK

We position our work in the field of on-policy distillation for large language models, and it is especially related to studies that treat the rollout distribution rather than the training loss as the object of design. Our core construction is a geometric mixture of the student and the teacher, which places us alongside prior attempts at mixed and off-policy sampling. We therefore briefly review the two research areas below.

On-policy distillation for LLMs. Knowledge distillation trains a smaller student to mimic a larger teacher (Hinton et al., 2015). Early LLM distillation was offline: the teacher generates a fixed dataset of completions and the student is fine-tuned on them (Kim & Rush, 2016; Gu et al., 2024). This approach suffers from a train–test distribution mismatch, since the student never learns to recover from its own errors. On-policy distillation addresses this by sampling rollouts from the student itself and using the teacher's token-level logits as the supervision signal. Agarwal et al. (2024) introduced Generalized Knowledge Distillation (GKD), which trains the student on its own samples under a reverse-KL objective. Subsequent work has refined the on-policy paradigm in several directions: Lin et al. (2020) propose interleaving imitation and distillation phases, Sun et al. (2026) introduce step-level control over the distillation signal, and Xu et al. (2025) use speculative decoding to accelerate the teacher's forward pass during on-policy rollout. Across all of these methods, the rollout distribution is either the student alone or the teacher alone, treated as a binary design choice rather than a quantity to optimise. Recent analyses have begun to expose failure modes of this binary paradigm: Fu et al. (2026) document three empirical pathologies including unreliable teacher guidance on drifted student prefixes, Wang et al. (2026) show that a large teacher–student distributional gap can steer exploration in counterproductive directions, and Li et al. (2026) find that successful OPD requires compatible thinking patterns between the two models.

Mixed and off-policy sampling in distillation. Several recent works have begun to explore alternatives to pure student sampling. GKD itself includes a variant that samples each rollout from the teacher with a fixed probability (Agarwal et al., 2024), but this sequence-level coin flip does not adapt to the local difficulty of individual tokens. Wang et al. (2025) observe that a small fraction of pivotal tokens along the reasoning path contribute disproportionately to downstream accuracy, suggesting that the sampling budget should be concentrated rather than spread uniformly. Importance-weighted OPD (IW-OPD) samples from the teacher and corrects the gradient with importance weights, which provides unbiased gradients but can suffer from high variance when the student and teacher diverge substantially (Xie et al., 2026a). Yang et al. (2026) take a different route by having the teacher rewrite student-generated trajectories into more concise reasoning traces, combining on-policy exploration with off-policy compression. Two concurrent works address the sampling problem from a trust-region perspective: Plyusov et al. (2026) blend the student and teacher rollout policies during a warmup phase and then transition to pure student sampling, while Xie et al. (2026b) construct a proximal teacher that controls gradient variance with formal convergence guarantees. Our work differs from these approaches in where the mixing happens. Rather than choosing between the student and the teacher at the sequence level or blending only during a warmup phase, we mix the two distributions at the token level through a geometric mixture, and the mixing strength follows from the variance analysis rather than from a hand-designed schedule.

3 MIXED SAMPLING FOR ON-POLICY DISTILLATION

We proceed by first recalling the on-policy distillation setup, and then deriving the optimal sampling distribution that stays as close to the teacher as possible while deviating from the student within a fixed budget. Next, we refine this budget from the sequence level to one that varies across token positions. We will analyze the gradient variance of the importance-weighted estimator along the resulting geometric path and show that it is strictly convex in the mixing strength, so that a strict mixture of student and teacher achieves lower variance than either endpoint. Finally, we propose MIXOPD, which sets the per-position mixing weight from the entropy gap between the two models, and describe its implementation inside the vLLM inference engine.

3.1 BACKGROUND

Let π_S denote the student policy and π_T the teacher policy. Given a prompt, on-policy distillation samples a rollout from the student and uses the teacher's predictive distribution as the supervision signal (Agarwal et al., 2024; Lin et al., 2020). This removes the train–test state mismatch that makes offline distillation brittle: the student is corrected on the states its own sampling actually reaches, rather than on trajectories the teacher produced (Ross et al., 2011).

Let $y = (y_1, \ldots, y_T)$ denote the response sequence. The training objective minimises the reverse KL divergence from the student to the teacher,

$$\mathrm{KL}(\pi_S \,\|\, \pi_T) \;=\; \sum_{t=1}^{T} \mathbb{E}_{y_{<t}\sim\pi_S}\big[\mathrm{KL}(\pi_S(\cdot \mid y_{<t}) \,\|\, \pi_T(\cdot \mid y_{<t}))\big], \tag{1}$$

which decomposes into a sum of per-token KL terms. Under standard on-policy distillation the rollout is sampled entirely from π_S, so the states at which each term is evaluated are determined by the student's own distribution. As discussed in Section 1, this is not necessarily where correction is most valuable, which motivates replacing π_S with a mixed sampling distribution.

3.2 MIXED SAMPLING DISTRIBUTION IN OPD

Recent studies have shown that sampling rollouts entirely from the student leads to unreliable teacher supervision (Liu et al., 2026; Fu et al., 2026) and low sample efficiency on hard problems (Cai et al., 2026; Liao et al., 2026). However, the KL objective in Eq. equation 1 does not require the rollout to come from the student. Since it is an expectation under π_S, importance sampling allows us to replace π_S with any distribution q,

$$\mathbb{E}_{y\sim\pi_S}[f(y)] \;=\; \mathbb{E}_{y\sim q}\left[\frac{\pi_S(y)}{q(y)}\, f(y)\right]. \tag{2}$$

By sampling from q and reweighting by $\pi_S(y)/q(y)$, we obtain an unbiased estimator of the same $\mathrm{KL}(\pi_S \,\|\, \pi_T)$. The choice of sampling distribution determines which tokens are realised but does not alter the training objective. This turns the sampling distribution into a free design parameter.

A natural goal is to choose q as close to the teacher as possible, so that the sampled trajectories are of high quality, while keeping it close enough to the student that the importance weights remain well-behaved. Since autoregressive generation factorises $q(y) = \prod_t q(y_t \mid y_{<t})$, the choice of sampling distribution reduces to choosing a conditional $q_t = q(y_t \mid y_{<t})$ at each position independently. We therefore analyse a single position and drop the subscript t for brevity, writing $\pi_S(\cdot), \pi_T(\cdot)$ for the conditional distributions at that position. We want the sampling distribution $q(\cdot)$ to minimise its divergence from the teacher, subject to a budget on how far it may drift from the student,

$$\min_{q}\ \mathrm{KL}(q \,\|\, \pi_T) \quad \text{s.t.} \quad \mathrm{KL}(q \,\|\, \pi_S) \le \epsilon, \tag{3}$$

where the constraint caps how far the sampling distribution may drift from the student at that position.

Theorem 1 (The optimal sampling distribution)**.** *The closed-form solution of equation 3 is*

$$q_\alpha(\cdot) \;\propto\; \pi_S(\cdot)^{1-\alpha}\, \pi_T(\cdot)^{\alpha}, \qquad \alpha \in [0, 1], \tag{4}$$

where α is determined by the budget ϵ.

The proof is in Appendix A. By Theorem 1 the optimal sampling distribution is a geometric average of the student and the teacher, equivalently a linear interpolation of their log-probabilities, and its mixing strength α is set by how much the sampling distribution is allowed to drift from the student. We now turn to how this strength should be chosen.

3.3 Mixed sampling minimises gradient variance

For any choice of α, importance weighting yields an unbiased gradient estimator, so α does not alter the training objective but only the variance of that estimator. We now prove that, under conditions generic in distillation, the α that minimises this variance lies strictly in $(0, 1)$. In other words, sampling purely from the student or purely from the teacher is not optimal, which establishes that mixed sampling is not merely a design choice but a necessity.

Theorem 2 (Mixed sampling minimises gradient variance)**.** *Let the second moment of the importance-weighted gradient estimator under the geometric mixture q_α be*

$$M(\alpha) \;=\; \mathbb{E}_{y\sim q_\alpha}\left[\left\| \frac{\pi_S(y)}{q_\alpha(y)}\,\log\frac{\pi_T(y)}{\pi_S(y)}\,\nabla_\theta \log \pi_S(y)\right\|^2\right]. \tag{5}$$

Because the importance-weighted estimator is unbiased for every α, the gradient mean $g = \mathbb{E}[\hat{g}_\alpha]$ does not depend on α. Furthermore, it can be shown that $M(\alpha)$ is strictly convex with respect to α. The gradient variance $V(\alpha) = M(\alpha) - \|g\|^2$ is therefore strictly convex in α whenever $\pi_S \neq \pi_T$. Write $R(y) = \log \frac{\pi_T(y)}{\pi_S(y)}$ for the per-token distillation reward and $c(y) = R(y)^2\|\nabla_\theta \log \pi_S(y)\|^2$ for the gradient energy of token y. If (i) $\mathrm{Cov}_{\pi_S}(c, R) > 0$ *and* (ii) $M'(1) > 0$*, then there exists a unique $\alpha^* \in (0, 1)$ such that $V(\alpha^*) < V(0)$ and $V(\alpha^*) < V(1)$.*

Proofs are in Appendix B. Actually, both conditions hold generically in distillation. Intuitively,

- Condition (i) requires that, under the student distribution, the distillation reward $R(y)$ and the gradient energy $c(y)$ are positively correlated, meaning that the tokens the teacher favours tend to be the same tokens on which the student's gradient is large. In distillation this is the default: the tokens the student has not yet learned are exactly those where $R(y)$ is large, and because the student's parameters have not converged on these tokens they remain sensitive to updates, so $\|\nabla_\theta \log \pi_S(y)\|$ is large as well. Both factors that compose c are therefore large on the same tokens, which gives the positive covariance.
- Condition (ii), $M'(1) > 0$, says that M is still increasing at $\alpha = 1$, so reducing α from pure teacher sampling lowers variance. At $\alpha = 1$ the importance weight of every token is $\pi_S(y)/\pi_T(y)$. On tokens where the student assigns much more mass than the teacher this ratio is large, and because M involves the square of the weight, a few such tokens dominate the sum. Reducing α slightly mixes the student into q_α, which raises q_α on exactly those tokens, shrinks their weights, and brings M down. The condition fails only when $\pi_S \approx \pi_T$, i.e. distillation is already nearly complete.

Together the two conditions imply that neither the student nor the teacher is the variance-optimal sampling distribution, which gives the following corollary.

Corollary 1. *In typical distillation settings where the student has not yet converged to the teacher, the variance-minimising sampling distribution is a **strict** mixture of the student and teacher rather than either one alone.*

To justify that the two conditions above do hold in ordinary distillation settings, we measure the gradient variance term at a range of mixing strengths on the DeepMath mathematical reasoning dataset (He et al., 2026). As shown in Figure 2, the measured curve is indeed convex, and its minimum occurs away from both endpoints, which confirms Corollary 1.

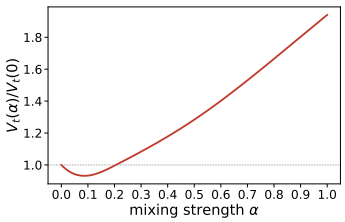


Figure 2: Gradient variance measured along the geometric mixture path, with a Qwen3-1.7B student and a Qwen3-14B teacher.

3.4 MIXOPD: GEOMETRIC MIXTURE SAMPLING FOR OPD

The preceding analysis establishes that the geometric mixture q_α from Theorem 1 is the optimal form for the sampling distribution, and Theorem 2 guarantees that the variance-minimising mixing strength lies strictly between pure student and pure teacher sampling. We now instantiate this theoretical result as a practical algorithm.

MixOPD with constant mixing strength. The simplest instantiation applies the geometric mixture with a single α shared across all positions and all problems. At each decoding step, the mixed logits are computed as

$$\log q_\alpha(y_t \mid y_{<t}) \;=\; (1-\alpha)\log \pi_S(y_t \mid y_{<t}) \;+\; \alpha \log \pi_T(y_t \mid y_{<t}) \;-\; \log Z_\alpha, \tag{6}$$

where Z_α is the normalising constant. The next token is sampled from q_α, and the importance weight $\pi_S(y)/q_\alpha(y)$ is recorded for the gradient update. Because the mixture operates in log-probability space, it amounts to a weighted average of the two models' logit vectors before the softmax, which adds negligible computation beyond the teacher forward pass itself. The mixing strength α is the sole hyperparameter and controls the trade-off between trajectory quality and importance-weight variance. A larger α shifts the sampling distribution toward the teacher, producing higher-quality rollouts but increasing the variance of the importance weights. A smaller α keeps the distribution close to the student, reducing variance but limiting the benefit of teacher guidance.

MixOPD-Ada with adaptive per-token mixing. A shared α treats every position equally, but the variance-optimal α^* derived in Theorem 2 varies from position to position as the prefix changes. A constant strength cannot distinguish positions where the student genuinely needs correction from positions where the two models differ only over interchangeable phrasings, and it allocates the deviation budget in proportion to raw divergence rather than to the value of correction (Wang et al., 2025; Lin et al., 2024). To let the mixing strength vary with position, we extend the trust-region formulation by attaching a nonnegative weight w_t to each position and pooling the per-position budgets into a single constraint,

$$\min_{q_1 \dots q_T} \sum_t w_t \mathrm{KL}(q_t \,\|\, \pi_T) \quad \text{s.t.} \quad \sum_t \mathrm{KL}(q_t \,\|\, \pi_S) \le \epsilon. \tag{7}$$

Theorem 3 (The weighted optimal sampling distribution)**.** *The closed-form solution of equation 7 is*

$$q_t(\cdot) \;\propto\; \pi_S(\cdot \mid y_{<t})^{1-\alpha_t}\, \pi_T(\cdot \mid y_{<t})^{\alpha_t}, \qquad \alpha_t \in [0,1], \tag{8}$$

where α_t is determined by w_t and the budget ϵ.

The proof is in Appendix A. We argue that mixing is most valuable at positions where the student is uncertain but the teacher is confident, because the mixture concentrates sampling onto the few tokens the teacher favours, which generally point toward the correct reasoning direction, and these tokens are already among the student's candidate outputs so the resulting update can be absorbed. The natural diagnostic is the entropy gap between the two predictive distributions. Write $H_S = H(\pi_S(\cdot \mid y_{<t}))$ and $H_T = H(\pi_T(\cdot \mid y_{<t}))$ for the entropies at position t. A large positive gap $H_S - H_T$ signals exactly the situation where mixing helps, while a negative gap means the student is already more certain than the teacher and mixing would only inject noise. We therefore scale α_t linearly with the entropy gap up to a cap $\bar{\alpha}$, and clamp it to zero when the gap is non-positive:

$$\alpha_t \;=\; \bar{\alpha} \cdot \mathrm{clamp}\left(\frac{H_S - H_T}{\tau},\; 0,\; 1 \right). \tag{9}$$

Here $\bar{\alpha} \in [0,1]$ is a global mixing cap and $\tau > 0$ sets the entropy-gap threshold.

At each decoding step, MixOPD-Ada computes α_t from the entropy gap and samples from the position-dependent geometric mixture,

$$\log q_{\alpha_t}(y_t \mid y_{<t}) \;=\; (1-\alpha_t)\log \pi_S(y_t \mid y_{<t}) \;+\; \alpha_t \log \pi_T(y_t \mid y_{<t}) \;-\; \log Z_{\alpha_t}. \tag{10}$$

Implementation. The geometric mixture requires both models' logits at every decoding step. A naive approach would run the teacher as a separate inference pass and merge logits afterwards, doubling the number of generation loops. We instead implement the mixture natively inside the vLLM inference engine (Kwon et al., 2023). Both the student and the teacher are loaded as separate weight sets within the same engine worker process. The teacher maintains its own KV cache, but reuses the same block table and slot-mapping metadata as the student, so the scheduler treats both models as a single request stream. At each decode step, the engine runs the student forward pass, then the teacher forward pass on the same input, mixes their log-probabilities according to α, and samples from the resulting distribution. As a side benefit, since the teacher's full next-token distribution is already computed at every step, we record $\log \pi_T(y_t \mid y_{<t})$ for the sampled token and pass it directly to the training loss, eliminating the separate reference forward pass that standard OPD requires after rollout generation.

4 EXPERIMENTS

4.1 SETUP

Evaluation benchmarks. We evaluate mathematical reasoning on AIME 2024, AIME 2025, and HMMT-Nov 2025 (30 problems each). Code generation is evaluated on HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). Agentic performance is measured on ALFWorld (Shridhar et al., 2020) with both the seen (140 episodes) and unseen (134 episodes) validation splits, reporting episode success rate.

Training data. For math, we train on the level-6 subset of DeepMath-103K (57K problems) (He et al., 2026). For ALFWorld, we train on 3,553 episodes from the official training split.

Models. We use Qwen3-4B and 1.7B as the student across all experiments (Yang et al., 2025). For mathematical reasoning and code generation, the teacher is Qwen3-30B-A3B; for ALFWorld, the teacher is Qwen3-14B.

Training details. All methods share the same on-policy training loop and are trained with the reverse-KL distillation objective. For math, we train with a batch size of 1024, learning rate 1×10^{-5}. For ALFWorld, we use a batch size of 128, learning rate 1×10^{-6}, and multi-turn rollout with up to 30 assistant turns per episode. For MIXOPD, we set the constant mixing strength to $\alpha = 0.3$. For MIXOPD-ADA, we set the global mixing cap $\bar{\alpha} = 0.3$ and the entropy-gap threshold $\tau = 0.4$. All three values are obtained by grid search.

Baselines. We compare against four baselines: (1) the untrained student checkpoint, (2) supervised fine-tuning (SFT) on teacher-generated rollouts, (3) standard OPD with pure student sampling, and (4) importance-weighted OPD (IW-OPD), which samples from the teacher and reweights gradients with clipped importance weights (Xie et al., 2026a).

4.2 MAIN RESULTS

Table 1 summarises the results across all three task domains. Both MIXOPD variants consistently outperform the OPD baseline and all other methods we compare against.

For the Qwen3-4B student, MIXOPD and MIXOPD-ADA raise the average from 49.5 (OPD) to 53.4 and 53.5, respectively, a gain of roughly 4 points. Both also surpass IW-OPD (51.8), which reweights teacher-sampled gradients but does not control where the teacher is mixed in. The two variants exhibit complementary strengths. MIXOPD with a constant α performs best on ALFWorld (44.3/44.8, matching the teacher) and HumanEval (79.4), while MIXOPD-ADA leads on all three mathematics benchmarks, with the largest margin on AIME24 (59.5 vs. 57.0) and HMMT (39.2 vs. 37.2). This pattern is consistent with the design of the entropy-gap schedule, which concentrates mixing at positions where the student is uncertain and the teacher is confident, a situation that arises more frequently in multi-step mathematical derivations than in agentic interactions or code generation. SFT performs worst on ALFWorld, falling below even the untrained student. On mathematics it closes part of the gap but still trails OPD substantially, confirming that pure off-policy learning cannot substitute for on-policy exploration.

Table 1: Distillation results across three task domains. Math and Code report pass@1 accuracy (%); ALFWorld reports episode success rate (%). The teacher for Math/Code is Qwen3-30B-A3B; for ALFWorld, Qwen3-14B. All methods train for the same number of steps with the same hyperparameters; only the rollout sampling strategy differs.

		ALFWorld		Math			Code		
Student	Method	Seen	Unseen	AIME24	AIME25	HMMT	HumanEval	MBPP	Avg
	Teacher	44.3	41.8	77.3	61.9	56.9	89.0	73.8	63.6
Qwen3-4B	Student	32.1	28.4	22.3	21.8	8.5	76.2	60.3	35.7
	SFT	15.0	15.7	48.1	42.7	29.2	76.2	58.2	40.7
	OPD	32.9	37.3	55.3	49.3	35.7	76.2	59.8	49.5
	IW-OPD	37.9	41.8	57.0	49.8	37.6	78.5	60.0	51.8
	MIXOPD	**44.3**	**44.8**	57.0	50.9	37.2	**79.4**	60.0	53.4
	MIXOPD-ADA	42.1	44.0	**59.5**	**51.5**	**39.2**	78.0	**60.2**	**53.5**
Qwen3-1.7B	Student	12.1	19.4	12.7	11.6	4.4	65.9	50.0	25.2
	SFT	25.7	28.4	19.5	18.3	5.8	66.5	48.9	30.4
	OPD	32.9	32.8	33.3	27.7	15.1	66.5	49.5	36.8
	IW-OPD	27.1	36.6	**35.3**	29.7	16.5	67.0	50.3	37.5
	MIXOPD	**40.0**	**41.8**	34.8	30.0	16.5	**68.4**	**51.7**	**40.5**
	MIXOPD-ADA	35.0	38.1	35.2	**30.2**	**16.9**	67.6	51.2	39.2

Table 2: KL divergence and entropy on teacher-generated rollout tokens (ALFWorld). KL is measured between the student and teacher distributions at each rollout position; entropy is the student's own output entropy. Teacher columns are zero by definition for KL and serve as an upper reference for entropy.

	Student	OPD	MIXOPD	Teacher
$\mathrm{KL}(\pi_S \,\|\, \pi_T)$	2.70	2.47	2.54	0
$\mathrm{KL}(\pi_T \,\|\, \pi_S)$	3.20	3.06	2.68	0
Entropy	0.37	0.36	0.47	0.76

The same trend holds for the smaller Qwen3-1.7B student, where the student-teacher gap is wider. MIXOPD improves the average from 36.8 (OPD) to 40.5 ($+3.7$), again with the largest gains on ALFWorld ($+7.1$/$+9.0$ on seen/unseen). MIXOPD-ADA leads on the harder mathematics benchmarks (AIME25, HMMT).

4.3 ANALYSIS: WHAT DOES MIXED SAMPLING CHANGE?

To understand why mixed sampling helps, we examine what changes in the trained student's distribution. Table 2 reports the KL divergence and entropy of the student evaluated on teacher-generated rollout tokens after training on ALFWorld. The reverse KL $\mathrm{KL}(\pi_T \,\|\, \pi_S)$ drops from 3.06 (OPD) to 2.68 (MixOPD), indicating that the MixOPD-trained student covers the teacher's distribution more faithfully. At the same time, the student's output entropy rises from 0.36 to 0.47, moving toward the teacher's entropy of 0.76. Standard OPD produces a student whose entropy barely changes from the untrained checkpoint, suggesting that pure on-policy training reinforces the student's existing confident predictions without broadening its distribution where the teacher would prefer more uncertainty. MixOPD, by exposing the student to teacher-influenced trajectories during training, prevents this premature narrowing and yields a student that better captures the diversity of the teacher's behaviour.

4.4 ABLATION STUDY

Table 3 sweeps the constant mixing strength α on ALFWorld with Qwen3-4B. Performance rises monotonically from α=0 (pure student, 32.9/37.3 seen/unseen) to α=0.3 (44.3/44.8), then begins to decline at α=0.5 (41.4/44.8). At the other extreme, α=1 collapses to 0.0, which is consistent with the theoretical prediction of Section 3: the gradient variance is strictly convex in α with its minimum in the interior of $(0, 1)$, so neither endpoint is optimal when the student and teacher differ

Table 3: Effect of mixing strength α on ALFWorld. α=0 corresponds to pure on-policy sampling (OPD) and α=1 to pure teacher sampling. Intermediate values use constant-α geometric mixing throughout training.

	α=0 (OPD)	α=0.1	α=0.3	α=0.5	α=1
Seen	32.9	36.4	**44.3**	41.4	0.0
Unseen	37.3	38.8	**44.8**	44.8	0.0

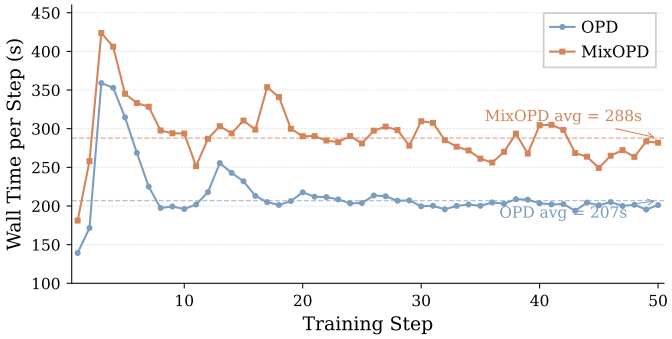


Figure 3: Per-step wall-clock time over 50 training steps. Dashed lines show the steady-state average (step 8–50). MixOPD adds 39% overhead over vanilla OPD by running both models within a single rollout pass and reusing teacher logits as reference log-probabilities.

substantially. The collapse at α=1 is expected because the importance weight $\pi_S(y)/\pi_T(y)$ can become very large on tokens that the student favours but the teacher does not, and the resulting gradient estimates are dominated by occasional high-weight spikes. The best constant value (α=0.3) lies closer to the student end, suggesting that retaining the student's own exploration is important for stable distillation in the multi-turn setting.

4.5 Training Efficiency

A practical concern with mixed sampling is the additional cost of running the teacher at every decoding step. Figure 3 compares the per-step wall-clock time of MixOPD (constant α=0.3, Qwen3-30B-A3B teacher) against vanilla OPD over 50 training steps on the same 4-node, 32-GPU cluster. OPD averages 207 s per step while MixOPD averages 288 s, which is a 1.39$\times$ overhead. This overhead is modest since the in-engine implementation reuses the teacher's logits computed during sampling to serve as the reference log-probabilities, eliminating the separate reference forward pass that standard OPD requires (52.8 s per step).

5 Conclusion

On-policy distillation faces a fundamental issue: sampling from the student exposes training to unreliable teacher guidance and poor coverage of hard problems, while sampling from the teacher introduces high-variance importance weights and risks catastrophic forgetting. We have shown that neither extreme is necessary. The gradient variance of the importance-weighted estimator is strictly convex in the student–teacher mixing coefficient, and under conditions generic in distillation its unique minimum lies strictly between the two endpoints. Building on this result, we propose MixOPD, which replaces the on-policy sampling distribution with a geometric student–teacher mixture, and its adaptive variant MixOPD-Ada, which modulates the mixing strength per token according to the entropy gap between the two models. Both variants are implemented natively inside the vLLM inference engine with only 39% wall-clock overhead. Experiments on agentic, mathematical reasoning, and code generation tasks across two student sizes confirm that mixed sampling consistently outperforms standard OPD, that the optimal mixing strength is neither zero nor one. These findings suggest that rollout distribution design is a substantial and underexplored lever for improving distillation.

AI USE STATEMENT

For manuscript preparation, generative AI assistants were used only for English translation, language polishing, and LaTeX formatting. They were not used to generate experimental results. The authors reviewed all AI-assisted manuscript edits and take responsibility for the final content and claims of the paper.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *ICLR*, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Pengxiang Cai, Tianchen Fang, Xiaohan Li, Qingyuan Zeng, Guocong Li, and Jintai Chen. Curriculum reinforcement learning can incentivize reasoning capacity in llms beyond the base model. *arXiv preprint arXiv:2606.22317*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2, 2023.

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

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

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

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *EMNLP*, 2016.

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

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

Baohao Liao, Hanze Dong, Christof Monz, Xinxing Xu, Li Dong, and Furu Wei. Multi-turn on-policy distillation with prefix replay. *arXiv preprint arXiv:2607.04763*, 2026.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *EMNLP*, 2020.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. Critical tokens matter: Token-level contrastive estimation enhances llm's reasoning capability. *arXiv preprint arXiv:2411.19943*, 2024.

Xinyu Liu, Kechen Jiao, Chunyang Xiao, Runsong Zhao, Junhao Ruan, Bei Li, Jiahao Liu, Qifan Wang, Xin Chen, Jingang Wang, et al. Teacher-guided policy optimization for on-policy reasoning distillation under large policy divergence. *arXiv preprint arXiv:2605.13230*, 2026.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *NeurIPS*, 2016.

Daniil Plyusov, Alexey Gorbatovski, Alexey Malakhov, Nikita Balagansky, Boris Shaposhnikov, Daria Korotyshova, and Daniil Gavrilov. Trust-region behavior blending for on-policy distillation. *arXiv preprint arXiv:2605.31159*, 2026.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Changhui Sun, Lanbo Liu, Hang Lei, Tong Ling, Jiahang Xie, Zhiyong Zheng, Yujia Wang, Hao Liu, Feng Xiao, Lu Liu, et al. Step-level on-policy distillation: Interpolating between on-policy distillation and supervised fine-tuning. *arXiv preprint arXiv:2608.16333*, 2026.

Daniil Tiapkin, Daniele Calandriello, Johan Ferret, Sarah Perrin, Nino Vieillard, Alexandre Ramé, and Mathieu Blondel. On teacher hacking in language model distillation. *arXiv preprint arXiv:2502.02671*, 2025.

Rui Wang, Hongru Wang, Yi Chen, Boyang Xue, Tianqing Fang, Wenhao Yu, and Kam-Fai Wong. Demystifying on-policy distillation: Roles, pathologies, and regulations. *arXiv preprint arXiv:2607.13399*, 2026.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. In *NeurIPS*, 2025.

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the position bias of on-policy distillation. *arXiv preprint arXiv:2606.22600*, 2026a.

Zhengpeng Xie, Li Lyna Zhang, Zeke Xie, and Mao Yang. Trust region policy distillation. *arXiv preprint arXiv:2607.04751*, 2026b.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *ICLR*, 2025.

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

Han Yang, Mingyan Wu, Bailan He, Zeyu Cao, Sikuan Yan, Kevin Qinghong Lin, and Zifeng Ding. Reasoning compression with mixed-policy distillation. *arXiv preprint arXiv:2605.08776*, 2026.

A Derivation of the geometric mixture

We work at a single state s, so q, π_S, and π_T are distributions over the vocabulary and y ranges over tokens. Introduce a multiplier $\lambda \geq 0$ for the budget and a multiplier μ for normalisation, so the Lagrangian is $\mathrm{KL}(q \,\|\, \pi_T) + \lambda \mathrm{KL}(q \,\|\, \pi_S) + \mu\big(\sum_y q(y) - 1\big)$. Both KL terms are strictly convex in q and the feasible set is convex and non-empty ($q = \pi_S$ satisfies the budget), so the optimum is unique and characterised by stationarity. The derivative in a single coordinate is

$$\left[\log \frac{q(y)}{\pi_T(y)} + 1\right] + \lambda \left[\log \frac{q(y)}{\pi_S(y)} + 1\right] + \mu \;=\; 0. \tag{11}$$

Collecting the $\log q(y)$ terms and absorbing everything independent of y into a constant yields $\log q(y) = \frac{1}{1+\lambda} \log \pi_T(y) + \frac{\lambda}{1+\lambda} \log \pi_S(y) + \text{const}$, which is the geometric mixture equation 4 with $\alpha = 1/(1+\lambda)$. The multiplier λ is fixed by the budget $\mathrm{KL}(q_\alpha \,\|\, \pi_S) = \epsilon$, so a tighter budget means a larger λ and hence a smaller α.

Section 3.4 lets the strength vary across positions by attaching a value weight $w_t \geq 0$ to position t and posing the per-position problem $\min_{q_1 \ldots q_T} \sum_t w_t \mathrm{KL}(q_t \,\|\, \pi_T)$ under the pooled budget $\sum_t \mathrm{KL}(q_t \,\|\, \pi_S) \leq \epsilon$, each divergence taken over the vocabulary at the state $y_{<t}$. A single λ on that budget makes the Lagrangian a sum of per-position terms, so every position is solved on its own. Dividing the position-t term by w_t replaces λ with λ / w_t in equation 11 and gives $\alpha_t = w_t/(w_t + \lambda)$, the weighted strength equation 8. The uniform case $w_t \equiv 1$ returns the shared $\alpha = 1/(1+\lambda)$.

B Gradient variance along the geometric path

This appendix proves Theorem 2 and its corollary. All quantities are conditioned on a fixed state s, and we write $p = \pi_S(\cdot \mid s)$, $q = \pi_T(\cdot \mid s)$ throughout. We use the following notation from Section 3.3 and its natural extensions:

- $R(a) = \log \frac{q(a)}{p(a)}$, the per-token distillation reward.
- $S(a) = \nabla_\theta \log p(a)$, the score function of the student.
- $c(a) = R(a)^2 \|S(a)\|^2$, the gradient energy.
- $q_\alpha(a) = p(a)^{1-\alpha}\, q(a)^\alpha / Z_\alpha$, the geometric mixture from Eq. equation 4, where $Z_\alpha = \sum_a p(a)^{1-\alpha}\, q(a)^\alpha$ is the normalising constant.
- $w_\alpha(a) = p(a)/q_\alpha(a) = Z_\alpha\, e^{-\alpha R(a)}$, the importance weight.

B.1 Derivation of the second-moment factorisation

The importance-weighted gradient estimator under q_α is $\hat{g}_\alpha = w_\alpha(a)\, R(a)\, S(a)$ where $w_\alpha(a) = p(a)/q_\alpha(a)$. Its second-moment trace is

$$M(\alpha) = \mathbb{E}_{q_\alpha}\big[\|\hat{g}_\alpha\|^2\big] = \mathbb{E}_{q_\alpha}\big[w_\alpha^2\, c\big], \tag{12}$$

since $\|\hat{g}_\alpha\|^2 = w_\alpha^2\, R^2\, \|S\|^2 = w_\alpha^2\, c$.

We change measure from q_α to p. Because $q_\alpha(a) = p(a)/w_\alpha(a)$,

$$\mathbb{E}_{q_\alpha}\big[w_\alpha^2\, c\big] = \sum_a q_\alpha(a)\, w_\alpha(a)^2\, c(a) = \sum_a \frac{p(a)}{w_\alpha(a)}\, w_\alpha(a)^2\, c(a) = \sum_a p(a)\, w_\alpha(a)\, c(a) = \mathbb{E}_p\big[w_\alpha\, c\big]. \tag{13}$$

Substituting $w_\alpha(a) = Z_\alpha\, e^{-\alpha R(a)}$ gives

$$M(\alpha) = Z_\alpha\, \mathbb{E}_p\big[c\, e^{-\alpha R}\big] = \mathbb{E}_p\big[e^{\alpha R}\big]\, \mathbb{E}_p\big[c\, e^{-\alpha R}\big] = A(\alpha)\, B(\alpha), \tag{14}$$

which is the factored form $M(\alpha) = A(\alpha)\, B(\alpha)$ used throughout this appendix.

B.2 Proof of Theorem 2

Define $\Phi(\alpha) = \log M(\alpha) = \log A(\alpha) + \log B(\alpha)$.

Second derivative of $\log A$. $A(\alpha) = \mathbb{E}_p[e^{\alpha R}]$ is the moment generating function of R under p, and $K(\alpha) = \log A(\alpha)$ is its cumulant generating function. The first two derivatives of a CGF are

$$K'(\alpha) = \frac{\mathbb{E}_p[R\, e^{\alpha R}]}{\mathbb{E}_p[e^{\alpha R}]} = \mathbb{E}_{q_\alpha}[R], \tag{15}$$

$$K''(\alpha) = \mathbb{E}_{q_\alpha}[R^2] - \big(\mathbb{E}_{q_\alpha}[R]\big)^2 = \mathrm{Var}_{q_\alpha}(R). \tag{16}$$

This is the standard identity for exponential families: q_α is the exponential tilt of p by the sufficient statistic R, and the second cumulant of the tilted distribution is the variance under that tilt.

Second derivative of $\log B$. $B(\alpha) = \mathbb{E}_p[c\, e^{-\alpha R}]$. Define the probability measure $\nu_\alpha(a) \propto p(a)\, c(a)\, e^{-\alpha R(a)}$, so that $B(\alpha) = \mathbb{E}_p[c]\, \mathbb{E}_{\tilde{p}}[e^{-\alpha R}]$ where $\tilde{p}(a) \propto p(a)\, c(a)$ is the c-reweighted base measure. Then $\log B(\alpha)$ is the CGF of $-R$ under $\tilde{p}$, and by the same exponential-family identity,

$$(\log B)'(\alpha) = \frac{\mathbb{E}_p[c\,(-R)\, e^{-\alpha R}]}{\mathbb{E}_p[c\, e^{-\alpha R}]} = -\mathbb{E}_{\nu_\alpha}[R], \tag{17}$$

$$(\log B)''(\alpha) = \mathrm{Var}_{\nu_\alpha}(R). \tag{18}$$

Combining.

$$\Phi''(\alpha) = K''(\alpha) + (\log B)''(\alpha) = \mathrm{Var}_{q_\alpha}(R) + \mathrm{Var}_{\nu_\alpha}(R) \geq 0, \tag{19}$$

with equality only if R is constant q_α-a.s. and ν_α-a.s., which under the assumption that p and q are not identical reduces to R being p-a.s. constant.

From log-convexity of M to strict convexity of the variance. Since $M = e^\Phi$, its second derivative is $M''(\alpha) = M(\alpha)\big[\Phi''(\alpha) + (\Phi'(\alpha))^2\big]$. Because $M(\alpha) > 0$ and $\Phi''(\alpha) > 0$ whenever $\pi_S \neq \pi_T$, we have $M''(\alpha) > 0$. The importance-weighted estimator is unbiased for every α, so the gradient mean $g = \mathbb{E}[\hat{g}_\alpha]$ does not depend on α. The gradient variance is therefore $V(\alpha) = M(\alpha) - \|g\|^2$, and $V''(\alpha) = M''(\alpha) > 0$, which establishes that the gradient variance itself is strictly convex. □

B.3 Proof that $\alpha^* \in (0, 1)$ under conditions (i) and (ii)

From equation 15 and equation 17 the first derivative of Φ is

$$\Phi'(\alpha) = \mathbb{E}_{q_\alpha}[R] - \mathbb{E}_{\nu_\alpha}[R]. \tag{20}$$

At $\alpha = 0$, $q_0 = p$ and $\nu_0(a) \propto p(a)\, c(a)$, so

$$\Phi'(0) = \mathbb{E}_p[R] - \frac{\mathbb{E}_p[c\, R]}{\mathbb{E}_p[c]} = -\frac{\mathrm{Cov}_p(c, R)}{\mathbb{E}_p[c]}. \tag{21}$$

If $\mathrm{Cov}_p(c, R) > 0$ then $\Phi'(0) < 0$, which means Φ is initially decreasing as α moves away from zero. Because $V' = M' = M\Phi'$, the same sign conditions hold for V: $V'(0) < 0$ and $V'(1) > 0$. Combined with the strict convexity of V established above, the intermediate value theorem gives a unique $\alpha^* \in (0, 1)$ with $V'(\alpha^*) = 0$, which is the unique global minimiser of V on $[0, 1]$. In particular $V(\alpha^*) < V(0)$ and $V(\alpha^*) < V(1)$. □

C Experimental details

Table 4 lists the full training configuration for both task families. All runs use the same codebase and the same 32-GPU cluster, and every method within a task family shares every setting other than the rollout sampling distribution, so the comparisons in Table 1 isolate the effect of sampling alone.

Table 4: Training configuration. Settings are identical across all methods within a task family, and only the rollout sampling distribution differs.

	Math and code	ALFWorld
Student	Qwen3-4B, Qwen3-1.7B	Qwen3-4B, Qwen3-1.7B
Teacher	Qwen3-30B-A3B	Qwen3-14B
Training data	DeepMath-103K level-6 subset	ALFWorld training split
Training steps	50	50
Batch size	1024	128
Minibatch size	1024	128
Learning rate	1×10^{-5}	1×10^{-6}
Optimiser	AdamW, no warmup	AdamW, no warmup
Max prompt length	2048	2048
Max response length	16384	16384
Rollouts per prompt	1	1
Sampling temperature	1.0	1.0
Assistant turns per episode	1	up to 30
Tokens per turn	n/a	2048
Observation history	n/a	2 turns
Loss	reverse KL, token-mean	reverse KL, token-mean