MITIGATING BLENDING DRIFT: ADAPTIVE BLENDED POLICY DISTILLATION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) knowledge distillation (KD) with reinforcement learning (RL) has recently shown promising results in enhancing the reasoning capabilities of student models by transferring knowledge from stronger teachers. Existing approaches predominantly fall into two pure policy paradigms. Off-policy distillation uses teacher-generated rollouts for high-quality supervision, but may exceed the student's learning capacity and induce serve train–test mismatch. On-policy distillation preserves learnability by using student-generated rollouts, but suffers from *prefix failure*, where erroneous student prefixes hinder effective teacher supervision. Policy blending offers a natural compromise by combining teacher and student policies for rollout generation. However, we identify a previously overlooked failure mode, termed *blending drift*, in which the blended policy assigns probability mass to regions unsupported by either the teacher or the student, a deviation that compounds autoregressively along the rollout. Through theoretical analysis, we provide a theoretical guarantee that reducing the sampling temperature effectively mitigates this drift. Motivated by this finding, we introduce **Adaptive Blended Policy Distillation (ABPD)**, a novel and simple *sharpen-then-blend* strategy that generates high-quality rollouts while remaining within the student's on-policy support. Extensive experiments on mathematical reasoning and code generation benchmarks demonstrate that ABPD consistently achieves the best overall performance among competing methods.

1 INTRODUCTION

Recent RL-based knowledge distillation (KD) methods largely follow two pure-policy paradigms, differing in where rollouts are sampled from. Off-policy distillation samples rollouts exclusively from the teacher (Hinton et al., 2015; Guo et al., 2025), providing high *rollout quality*, but such supervision may exceed the student's learning capacity and lead to train–test mismatch, thereby limiting *rollout learnability*. In contrast, on-policy distillation samples rollouts from the student (Agarwal et al., 2024; Gu et al., 2024), improving *rollout learnability*, but the student model itself is typically weaker and error-prone. These errors can further induce *prefix failure* (Jiang et al., 2026), where the teacher struggles to provide effective supervision on such wrong prefixes. Thus, the two paradigms face a clear trade-off between *rollout quality* and *rollout learnability*.

Recent work has advanced both directions. In off-policy distillation, efforts aim to improve *rollout learnability*, e.g., by filtering teacher-generated rollouts based on student acceptability (Yu et al., 2026b). In on-policy distillation, many works instead aim to strengthen teacher supervision on student-generated rollouts, e.g., by directly modifying supervision signals (Jang et al., 2026; Kim & Baek, 2026), moving from token-level to chunk-level semantic supervision (Zhou et al., 2026), or correcting student-generated rollouts (Jiang et al., 2026). Despite these improvements, existing approaches largely retain the underlying pure-policy rollout mechanism. Due to the autoregressive nature of LLM sampling, early-step deviations propagate through subsequent predictions, progressively reinforcing teacher or

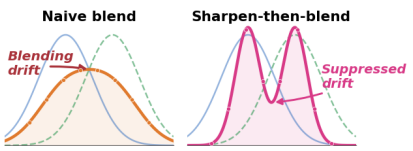


Figure 1: Illustration of *blending drift*. Naive blending (orange) allocates probability mass to low-confidence regions, whereas sharpen-then-blend suppresses such drift.

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.

	Rollout Policy	Rollout Accuracy (↑)	Rollout Learnability (↑)	Supervision Signal	Supervision Entropy (↓)
On-policy (Agarwal et al., 2024)	$\mathcal{P}_s$	29.2	94.6	$\mathcal{P}_t$	2.153
Off-policy	$\mathcal{P}_t$	47.1	65.2	$\mathcal{P}_t$	0.134
MiniLLM (Gu et al., 2024)	$\mathrm{MoE}(\mathcal{P}_s, \mathcal{P}_t)$	33.6	68.6	$\mathcal{P}_t$	1.879
Veto (Jang et al., 2026)	$\mathcal{P}_s$	29.2	94.6	$\mathcal{Q} \propto \mathcal{P}_t \cdot \mathcal{P}_s^{\beta}$	0.543
TRB (Plyusov et al., 2026)	$\mathrm{PoE}(\mathcal{P}_s, \mathcal{P}_t)$	31.0	77.8	$\mathcal{P}_t$	1.479
ABPD (Ours)	$\mathrm{MoE}(\mathcal{P}_s^{\tau}, \mathcal{P}_t^{\tau})$	37.1	86.6	$\mathrm{MoE}(\mathcal{P}_s^{\tau}, \mathcal{P}_t^{\tau})$	0.172

Table 1: Comparison of rollout policies on DAPO-17K (Yu et al., 2026a) using Qwen3-1.7B (non-thinking) as student and Qwen3-4B-GRPO (non-thinking) as teacher. $\mathcal{P}_t, \mathcal{P}_s$: teacher/student distributions; $\mathcal{P}_t^{\tau}, \mathcal{P}_s^{\tau}$: temperature-sharpened teacher/student distributions; $\mathrm{MoE}(\cdot,\cdot)$: weighted mixture of distributions; $\mathrm{PoE}(\cdot,\cdot)$: weighted product of distributions. We measure *Rollout Learnability* as the average probability that the student policy assigns to rollout tokens, and measure *Rollout Quality* using *Rollout Accuracy* and *Supervision Entropy*.

student preferred generation patterns. Consequently, we argue that pure-policy rollouts struggle to jointly preserve *rollout quality* and *rollout learnability* as shown in Table 1.

Inspired by *policy blending* (Baram et al., 2021; He et al., 2026), which combines the complementary strengths of multiple RL policies, we explore its use in LLM KD to balance *rollout quality* and *rollout learnability*. However, we identify a previously overlooked failure mode of *policy blending*, which we term *blending drift* (Figure 1): blended policies assign probability mass to regions that are low-confidence for both the teacher and the student. Under autoregressive generation, such deviations compound, pushing rollouts toward increasingly uncertain regions. Consequently, *policy blending* may produce rollouts that are poor in both *rollout quality* and *rollout learnability*. Table 1 provides empirical evidence: although MiniLLM (Gu et al., 2024) and TRB (Plyusov et al., 2026) adopt different blending strategies (MoE and PoE), both exhibit severe *blending drift*, reflected in substantially degraded *rollout learnability* and elevated supervision entropy.

In this paper, we theoretically model and analyze *blending drift*, and derive a novel temperature-dependent upper bound, showing that lowering the sampling temperature effectively mitigates *blending drift*. Then we propose **A**daptive **B**lended **P**olicy **D**istillation (**ABPD**), which follows a simple *sharpen-then-blend* strategy. Instead of using a predefined blending coefficient as in MiniLLM (Gu et al., 2024), ABPD formulates *policy blending* as a KL-constrained optimization that seeks the optimal mixing coefficient between sharpened teacher and student distributions. This optimization is solved by a lightweight algorithm that incurs only about 8.7% extra time overhead relative to a predefined coefficient, with negligible memory overhead. As shown in Table 1, ABPD achieves the best balance between *rollout learnability* and *rollout quality* among existing blending policies. Extensive experiments on mathematical reasoning and code generation benchmarks demonstrate that ABPD consistently achieves the best overall performance among competing distillation methods. APBD also improves training stability, with a significant reduction in reward variance .

In summary, our main contributions are twofold:

- **Theoretical analysis of blending drift in LLM distillation.** We formally characterize *blending drift*, a previously overlooked failure mode in *policy blending* for LLM distillation, and derive a temperature-dependent upper bound.
- **Adaptive Blended Policy Distillation (ABPD).** We propose a novel framework that dynamically blends sharpened teacher and student distributions via per-token constrained optimization. Experiments on math and code benchmarks validate its effectiveness.

2 Preliminaries

2.1 LLM Distillation as Reinforcement Learning

Following MiniLLM (Gu et al., 2024), we formulate LLM distillation as a reinforcement learning problem. Let $\mathcal{X}$ denote the prompt space and $\mathcal{Y}$ the vocabulary. Given a prompt $x \in \mathcal{X}$, the teacher

and student induce autoregressive distributions $\mathcal{P}(y|x) = \prod_{t=1}^{T} \mathcal{P}(y_t|x, y_{<t})$, where $\mathcal{P}(y_t|x, y_{<t})$ denotes the probability of generating token y_t conditioned on the prompt and preceding tokens. A rollout policy π first generates trajectories $y \sim \pi(\cdot|x)$, upon which the teacher provides token-level rewards $r_t = f_{reward}(\mathcal{P}_t(y_t|x, y_{<t}), \mathcal{P}_s(y_t|x, y_{<t}))$.

2.2 Pure Policies

Off-policy Distillation. Traditional knowledge distillation samples trajectories from the teacher policy, $\pi = \mathcal{P}_t$. To recover the conventional forward-KL objective, the reward should be defined as $r_t = \log \mathcal{P}_s(y_t|x, y_{<t}) - \log \mathcal{P}_t(y_t|x, y_{<t})$. It can be readily shown (see Appendix A.1) that maximizing the expected cumulative reward under teacher-generated trajectories is equivalent to minimizing the forward KL divergence $D_{\mathrm{KL}}(\mathcal{P}_t \| \mathcal{P}_s)$.

On-policy Distillation. On-policy distillation samples trajectories from the student policy, $\pi = \mathcal{P}_s$. Following MiniLLM (Gu et al., 2024), the teacher provides rewards $r_t = \log \mathcal{P}_t(y_t|x, y_{<t}) - \log \mathcal{P}_s(y_t|x, y_{<t})$, which can be shown to be equivalent to minimizing the reverse KL divergence $D_{\mathrm{KL}}(\mathcal{P}_s \| \mathcal{P}_t)$.

2.3 Blended Policies

Pure policies, which roll out trajectories solely from the teacher or the student, exhibit a clear trade-off between *rollout quality* and *rollout learnability*. As shown in Figure 2 (a,b), Off-policy distillation achieves higher teacher preference at the cost of student preference, while on-policy distillation shows the reverse trend. Among the few existing blending methods, MiniLLM (Gu et al., 2024) adopts a mixture-based strategy at the probability level, while TRB (Plyusov et al., 2026) blends in the logit space. However, as shown in Figure 2 (a,b), both methods (Gu et al., 2024; Plyusov et al., 2026) exhibit strong *blending drift* around token indices between 1000–5000, where both teacher and student preferences are markedly low.

3 Blending Drift: A Failure Mode of Blended Policies

3.1 Definition and Decomposition

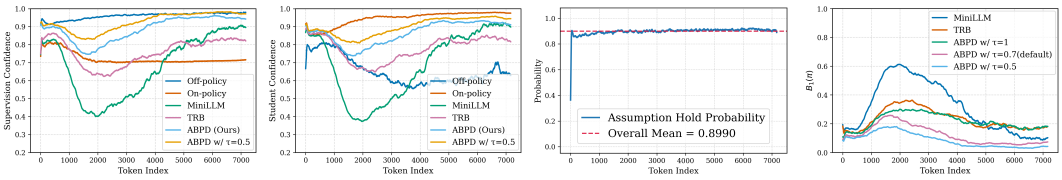
For any given prefix $y_{<t}$ and a candidate next token $y \in \mathcal{Y}$, let $\mathcal{P}_t$ and $\mathcal{P}_s$ denote the teacher and student conditional distributions, respectively. Define $\mathcal{T}_t^M = \mathrm{TopM}(\mathcal{P}_t)$ and $\mathcal{T}_s^M = \mathrm{TopM}(\mathcal{P}_s)$ as the sets of the M most likely next tokens under each distribution, where $M \in \{1, \ldots, |\mathcal{Y}|\}$ controls the size of the confident support. The common low-confidence region for this step is then $\mathcal{U}_M = \mathcal{Y} \setminus \left(\mathcal{T}_t^M \cup \mathcal{T}_s^M\right)$, which contains all candidate tokens that fall outside the confident supports of both policies. For any rollout policy π, where $\pi(y)$ denotes the probability of sampling token $y \in \mathcal{Y}$ given the current prefix. Then the **blending-drift mass** at this step is defined as the total probability assigned to tokens in $\mathcal{U}_M$:

$$B_M(\pi) = \pi(\mathcal{U}_M) = \sum_{y \in \mathcal{U}_M} \pi(y). \tag{1}$$

Theorem 3.1 (Blending Drift Upper Bound)**.** *For any given prefix and any blending ratio $\beta \in [0, 1]$, the blended policy $\pi_\beta = \beta\mathcal{P}_t + (1-\beta)\mathcal{P}_s$ satisfies (Proof shown in Appendix A.2)*

$$B_M(\pi_\beta) \leq \beta\big[1 - \mathcal{P}_t(\mathcal{T}_t^M)\big] + (1-\beta)\big[1 - \mathcal{P}_s(\mathcal{T}_s^M)\big]. \tag{2}$$

Why policy blending tends to drift in LLM KD? A policy k is *flat* when $\mathcal{P}_k(\mathcal{T}_k^M)$ is small, i.e., its top-M support covers only a small fraction of its probability mass. Theorem 3.1 then implies that blending drift is amplified when such a flat policy receives non-negligible blending weight. In LLM distillation, different policies may induce asymmetry due to their distinct output patterns. For example, in off-policy distillation, rollouts are generated by the teacher policy, and the student policy tends to be unfamiliar, resulting in a flat distribution. When blending policies, if the flat policy receives a large weight, $B_M(\pi_\beta)$ is likely to grow, i.e., the blended policy is more likely to sample tokens outside both $\mathcal{T}_s^M$ and $\mathcal{T}_t^M$. Autoregressively, such drift progressively pushes rollouts outside both policies' expertise, making subsequent $\mathcal{P}_t$ and $\mathcal{P}_s$ flatter. Overall, this self-reinforcing asymmetry makes policy blending particularly prone to drift in LLM KD.



(a) Teacher Preference (b) Student Preference (c) Assumption Validation (d) $B_M(\pi)$

Figure 2: Analysis of Rollouts across Policies. We study rollouts from Qwen3-4B-Base-GRPO (Li et al., 2026) (Teacher) and Qwen3-1.7B (Student) on DAPO-Math-17K (Yu et al., 2026a) along token positions (indexes), using 10,000 randomly sampled rollouts with a maximum length of 7,168. Rollout-token confidence $c_t^k = \mathcal{P}_k(y_t \mid x, y_{<t})$ for $k \in \{t, s\}$ serves as a per-token preference proxy, evaluated under the teacher for panel (a) and under the student for panel (b); (c) validates the common-tail assumption; (d) visualizes the blending drift, i.e., $B_M(\pi)$ where $M = 1$.

3.2 Sharpening Mitigates Blending Drift

Temperature-Dependent Upper Bound For any distribution $\mathcal{P}$, its sharpened version with temperature $\tau < 1$ is

$$\mathcal{P}^\tau(y) = \frac{\mathcal{P}(y)^{1/\tau}}{\sum_{y' \in \mathcal{Y}} \mathcal{P}(y')^{1/\tau}}. \tag{3}$$

The following bound quantifies how the *blending drift* shrinks as the temperature decreases.

Theorem 3.2 (Temperature-Dependent Bound on Blending Drift)**.** *Assume the common-tail condition holds for both teacher and student:* $\max_{y \in \mathcal{U}_M} \mathcal{P}_k(y) \le \min_{z \notin \mathcal{U}_M} \mathcal{P}_k(z)$ *for* $k \in \{t, s\}$*. For any mixing ratio* $\beta \in [0, 1]$*, define the mixture policy* $\pi_\beta = \beta\mathcal{P}_t + (1-\beta)\mathcal{P}_s$ *and the sharpen-then-blend policy* $\pi_\beta^\tau = \beta\mathcal{P}_t^\tau + (1-\beta)\mathcal{P}_s^\tau$*. Define* $q_k = \mathcal{P}_k(\mathcal{U}_M)$ *and* $\rho_k = \frac{\max_{y \in \mathcal{U}_M} \mathcal{P}_k(y)}{\min_{z \notin \mathcal{U}_M} \mathcal{P}_k(z)} \le 1$ *for* $k \in \{t, s\}$*. Then for any* $\beta \in [0, 1]$*,* $\tau \in (0, 1]$*, and* $M \in \{1, \ldots, |\mathcal{Y}|\}$*, we have*

$$B_M(\pi_\beta^\tau) \le \beta \frac{q_t\, \rho_t^{1/\tau - 1}}{q_t\, \rho_t^{1/\tau - 1} + 1 - q_t} + (1-\beta) \frac{q_s\, \rho_s^{1/\tau - 1}}{q_s\, \rho_s^{1/\tau - 1} + 1 - q_s} \le B_M(\pi_\beta). \tag{4}$$

The proof is given in Appendix A.3.

Validity of the Common-Tail Assumption. The common-tail assumption is governed by M, the size of the confident-support sets $\mathcal{T}_t^M$ and $\mathcal{T}_s^M$: it holds trivially at $M = |\mathcal{Y}|$ and is most stringent at $M = 1$, where $\mathcal{U}_M$ is largest and $B_M(\pi)$ is maximal. We verify it on DAPO-Math-17K (Yu et al., 2026a) using $\pi_\beta = \beta\mathcal{P}_t + (1-\beta)\mathcal{P}_s$ with $\beta \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. At $M = 1$, it holds for over 89% of token positions on average as shown in Figure 2(c); Appendix B provides a breakdown by β. Even if the common-tail condition fails, ABPD remains safe because it solves for the teacher-closest policy within the student's KL budget. This budget restricts ABPD's rollouts from entering regions unfamiliar to the student. As shown in Figure 2(d), even without sharpening ($\tau = 1$), ABPD also achieves lower drift than MiniLLM (Gu et al., 2024) and TRB (Plyusov et al., 2026).

Sharpening Reduces Blending Drift. Our bound yields a key implication: when the common-tail assumption holds, blending drift decreases monotonically as the temperature decreases, so lowering the temperature effectively reduces blending drift. As shown in Figure 2(d), we vary the sharpening temperature of ABPD over $\{0.5, 0.7, 1\}$ and measure the blending drift at $M = 1$. A pronounced decrease is observed as the temperature drops, further validating our theoretical result.

Trade-off between blending drift and diversity. As established in Theorem 3.2, smaller τ tightens the upper bound on *blending drift*. However, sharpening also contracts the entropy of both distributions and thus reduces generation diversity. In the extreme case $\tau \to 0^+$, the bound vanishes but both policies collapse to one-hot vectors, eliminating the stochasticity required for diverse and exploratory generation. Consequently, τ governs a fundamental trade-off between drift suppression and diversity preservation. We set $\tau = 0.7$ by default, which proves robust across mathematical reasoning and code generation benchmarks as well as across Qwen and DeepSeek model families. A detailed ablation on τ is provided in Section 5.3.

4 ADAPTIVE BLENDED POLICY DISTILLATION

We now introduce **ABPD** (**A**daptive **B**lended **P**olicy **D**istillation), a novel distillation framework that adaptively blends sharpened teacher and student distributions to balance *rollout quality* and *rollout learnability* while mitigating *blending drift*.

4.1 MOTIVATION AND PROBLEM FORMULATION

While Theorem 3.2 shows that sharpening before blending reduces *blending drift*, a fixed mixing ratio β, e.g., 0.2 used in MiniLLM (Gu et al., 2024) cannot guarantee a balance between *rollout quality* and *rollout learnability* across the generated rollout. To address this, ABPD treats blending as a per-token constrained optimization. At each generation step, given a prefix $y_{<t}$, we have the teacher distribution $\mathcal{P}_t$ and the student distribution $\mathcal{P}_s$. We first sharpen both with a temperature $\tau \in (0, 1]$, obtaining $\mathcal{P}_t^\tau$ and $\mathcal{P}_s^\tau$. Then the blended policy is defined as a simple linear mixture: $\pi_\beta^\tau = \beta\mathcal{P}_t^\tau + (1-\beta)\mathcal{P}_s^\tau$. We seek the mixing coefficient $\beta \in [0, 1]$ that makes the blended policy as close to the teacher as possible (to maximize *rollout quality*) while keeping it within the student's confident region (to preserve *rollout learnability*). Formally, we minimize the KL divergence to the teacher subject to a KL budget, i.e., $\epsilon \geq 0$ relative to the *unsharpened* student:

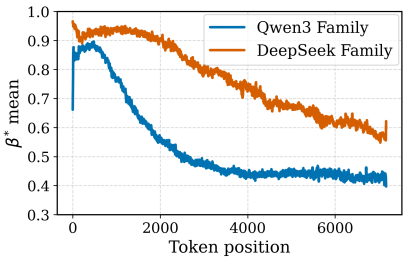
$$\beta^* = \arg\min_{\beta \in [0,1]} D_{\mathrm{KL}}(\pi_\beta^\tau \,\|\, \mathcal{P}_t) \quad \text{s.t.} \quad D_{\mathrm{KL}}(\pi_\beta^\tau \,\|\, \mathcal{P}_s) \leq \epsilon. \tag{5}$$

4.2 SOLVING THE CONSTRAINED OPTIMIZATION

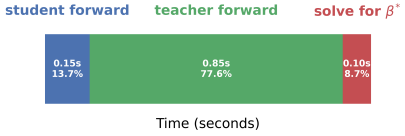
We solve equation 5 efficiently with a two-stage procedure that exploits the one-dimensional, well-behaved nature of both the objective and the constraint.

Feasibility checks. We first check whether a feasible blending coefficient β^* exists. If the teacher is already close enough to the student within the budget ϵ, i.e., $D_{\mathrm{KL}}(\mathcal{P}_t \,\|\, \mathcal{P}_s) \leq \epsilon$, we skip blending and directly adopt the off-policy teacher distribution $\mathcal{P}_t$. Otherwise, we solve for an intermediate β^*. If $\beta^* = 0$ and even the sharpened student violates the budget, i.e., $D_{\mathrm{KL}}(\mathcal{P}_s^\tau \,\|\, \mathcal{P}_s) > \epsilon$, no feasible coefficient exists, and we fall back to the on-policy raw student distribution $\mathcal{P}_s$.

Solving for β^*. For the remaining tokens that satisfy neither boundary condition, we proceed to locate the feasible interval. Since $g(\beta) := D_{\mathrm{KL}}(\pi_\beta^\tau \,\|\, \mathcal{P}_s)$ is empirically monotonic in β, the feasible set $\{\beta \mid g(\beta) \leq \epsilon\}$ is an interval $[0, R]$ with $R \in [0, 1]$; we determine R via bisection over $[0, 1]$. Within the feasible interval $[0, R]$, we minimize $f(\beta) := D_{\mathrm{KL}}(\pi_\beta^\tau \,\|\, \mathcal{P}_t)$. Since f is convex, we locate its unconstrained minimizer β_f using golden-section search over $[0, 1]$. The optimal mixing coefficient is then $\beta^* = \min(\beta_f, R)$, i.e., the best teacher-aligned point that does not exceed the budget.



(a) Illustration of β^*



(b) Time overhead (Qwen3 family)

Figure 3: Illustration on DAPO-Math-17K with $\epsilon = 0.2$ and $\tau = 0.7$.

Visualization of β^*. Figure 3(a) shows that β^* decreases with token position for both backbones, confirming that ABPD adaptively reduces reliance on the teacher as generation proceeds. This phenomenon is consistent with recent findings in the on-policy setting, where teacher supervision degrades over long rollouts (Li et al., 2026; Ziheng et al., 2026), and it justifies our design of solving for β^*: ABPD naturally suppresses potentially harmful teacher signals in long rollouts.

Overhead Analysis. Policy blending inherently requires fusing the teacher and student distributions before sampling, and obtaining both requires forward passes; this cost is unavoidable for arbitrary blending methods. The blending algorithm itself, however, can be lightweight. As shown in Figure 3(b), solving for β^* incurs only about 8.7% of the runtime relative to the forward passes, and its additional memory footprint is negligible relative to the forward passes.

Algorithm 1 ABPD Training Step

Input: prompt x, teacher $\mathcal{P}_t$, student $\mathcal{P}_s$, temperature τ, budget ϵ
Output: updated student parameters θ

```
Initialize sequence y ← ()
for t = 1, ..., T do
    Obtain teacher logits l_t and student logits l_s
    Compute sharpened distributions P_t^τ, P_s^τ
    Solve for β* (Details shown in Sec. 4.2)
    Sample y_t ~ π*(· | x, y_<t)
    Compute reward r_t = log P_s(y_t) − log π*(y_t)
    Append y_t to y
end for
Compute L_ABPD(θ)
Update θ via gradient descent on L_ABPD
```

4.3 Token-Level Reward

Let $\pi^* := \pi^{\tau}_{\beta^*}$ denote the resulting optimal blended policy. In contrast to mainstream LLM distillation approaches such as on-policy (Agarwal et al., 2024), off-policy, and blending policy (Gu et al., 2024; Plyusov et al., 2026), which depend on teacher logits, our method obtains a token-wise reward directly from the adaptive blended policy π^*. At each generation step t, conditioned on the prefix $x, y_{<t}$, we define the per-token reward as the log-probability difference:

$$r_t = \log \mathcal{P}_s(y_t \mid x, y_{<t}) - \log \pi^*(y_t \mid x, y_{<t}). \tag{6}$$

The overall training objective of ABPD minimizes the expected negative cumulative reward under the blended rollout policy π^*:

$$\mathcal{L}_{\text{ABPD}}(\theta) = -\mathbb{E}_{x,\, y\sim\pi^*(\cdot|x)} \left[\sum_{t=1}^{T} r_t \right]. \tag{7}$$

Gradient estimator. We treat π^* as a stop-gradient target distribution. Under this convention, Equation equation 7 reduces to $\mathcal{L}_{\text{ABPD}}(\theta) = D_{\text{KL}}\big(\pi^* \,\|\, \mathcal{P}_s\big)$, which is the standard off-policy distillation setting, with π^* as the teacher and $\mathcal{P}_s$ as the student.

Stable Reward via ABPD. Pure-policy KD methods are prone to unstable training: when one policy assigns high probability to a token while the other assigns near-zero probability, the log-probability difference in the reward yields extremely large—or exploding—gradients whenever the near-zero term appears in the denominator. ABPD instead derives the reward from the adaptive blended policy π^*, whose deviation from the student is bounded by the per-token KL budget ϵ. This resembles Veto (Jang et al., 2026), which reshapes the supervision signal via a product of experts for stable training; unlike Veto, ABPD also blends the rollout policy, keeping rollouts and supervision consistent. As shown in Section 5.4, this consistency yields substantially lower reward variance than Veto, translating into more stable training and better overall performance.

4.4 Algorithm Summary

The ABPD framework introduces a novel adaptive blending mechanism that dynamically adjusts the mixing ratio between sharpened teacher and student distributions at each decoding step. Compared with on-policy and off-policy distillation, ABPD is easy to implement: as shown in Algorithm 1, adapting an on-policy distillation pipeline to ABPD requires only two simple fixes—replacing the rollout sampling distribution $\mathcal{P}_s$ with the blended policy π^*, and modifying the reward accordingly—both of which can be implemented with a few lines of code.

5 Experiments

In this section, we conduct comprehensive experiments to evaluate the effectiveness of ABPD on both mathematical reasoning and code generation tasks. We detail the experimental setup, present main results, and provide ablation studies to analyze the key components of our method.

Method	MATH500		Olympiad		Minerva		AMC23		AIME24		AIME25		Overall	
	Avg	Pass	Avg	Pass	Avg	Pass	Avg	Pass	Avg	Pass	Avg	Pass	Avg	Pass
Teacher: Qwen3-4B-Base-GRPO (non-thinking) Student: Qwen3-1.7B (non-thinking)														
Teacher	83.0	90.6	49.6	55.6	32.1	44.9	68.1	87.5	25.0	33.3	19.2	40.0	46.2	58.7
Base	71.5	83.8	36.1	52.6	19.8	28.3	42.5	47.5	10.2	16.7	10.0	20.0	31.7	41.5
Off-policy	71.8	85.2	36.9	51.9	21.4	28.7	41.2	62.5	10.0	23.3	11.7	20.0	32.2	45.3
On-policy (Agarwal et al., 2024)	69.5	83.8	34.6	50.0	20.3	30.1	38.8	65.0	13.3	23.3	10.0	13.3	31.1	44.3
Veto (Jang et al., 2026)	70.9	85.4	37.1	54.0	20.4	29.0	48.8	67.5	12.5	23.3	7.5	13.3	32.9	45.4
MiniLLM (Gu et al., 2024)	71.3	85.0	37.1	49.8	20.1	29.8	43.1	62.5	**15.0**	23.3	11.6	23.3	33.0	45.6
TRB (Plyusov et al., 2026)	70.3	83.4	37.0	51.9	**22.8**	**31.9**	42.5	65.0	12.5	23.3	12.5	23.3	32.9	46.5
ABPD (Ours)	**72.5**	**86.4**	**37.4**	52.4	20.7	30.1	43.8	**72.5**	**15.0**	**30.0**	**15.0**	**26.7**	**34.1**	**49.7**
Teacher: DeepSeek-R1-Distill-Qwen-7B Student: DeepSeek-R1-Distill-Qwen-1.5B														
Teacher	91.0	95.4	57.4	68.4	37.7	47.8	90.3	95.0	58.8	60.0	38.3	43.3	62.3	68.3
Base	83.2	92.8	44.4	57.8	**29.0**	39.7	63.0	87.5	28.0	40.0	23.3	36.7	45.2	59.1
Off-policy	82.6	**94.2**	44.4	58.6	27.7	**44.4**	70.9	87.5	24.6	**56.7**	20.4	36.7	45.1	63.0
On-policy (Agarwal et al., 2024)	83.2	91.8	**45.4**	59.2	26.2	37.1	66.9	85.0	24.1	50.0	20.8	30.0	44.4	58.9
Veto (Jang et al., 2026)	83.1	91.6	45.1	58.8	28.7	39.0	68.1	87.5	28.3	50.0	26.7	33.3	46.7	60.0
MiniLLM (Gu et al., 2024)	84.1	92.0	44.5	58.3	27.8	36.8	69.4	87.5	28.3	50.0	23.3	33.3	46.2	59.7
TRB (Plyusov et al., 2026)	**84.2**	93.0	45.2	60.2	27.0	39.3	68.1	87.5	21.7	46.7	24.2	**43.3**	45.1	61.7
ABPD (Ours)	83.0	92.2	44.9	**62.8**	**29.0**	40.0	**71.2**	**95.0**	**31.6**	**56.7**	**26.7**	36.7	**47.7**	**63.9**

Table 2: Results on mathematical reasoning benchmarks. "Avg" and "Pass" denote Avg@4 (mean correctness over four samples) and Pass@4 (at least one correct among four samples).

5.1 Experiment Setup

Model Configurations. For mathematical reasoning, we use Qwen3-1.7B (non-thinking) as the student and Qwen3-4B-Base-GRPO (Shao et al., 2024) (non-thinking) as the teacher, following Li et al. (2026); as a second pair, we use DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Qwen-1.5B. For code generation, we use Qwen3-4B and Qwen3-1.7B (both non-thinking) as the teacher and student, respectively. For brevity, we refer to the three settings as Qwen-math, DeepSeek-math, and Qwen-code, respectively.

Datasets and Evaluation. We adopt a unified evaluation protocol for both tasks: for each problem, we sample 4 responses with temperature 0.7 and top-p 0.95, and report **Avg@4** and **Pass@4**. For mathematical reasoning, we train on DAPO-Math-17K (Yu et al., 2026a) and evaluate on MATH500 (Hendrycks et al., 2021), OlympiadBench (He et al., 2024), Minerva (Lewkowycz et al., 2022), AMC23, AIME24, and AIME25. For code generation, we train on Magicoder-Evol-Instruct-110K (Wei et al., 2023) and evaluate on MBPP (Austin et al., 2021), MBPP+ (Liu et al., 2023), HumanEval (Chen et al., 2021), and HumanEval+ (Liu et al., 2023). Details are in Appendix C.

Baselines. We compare ABPD against the student model without distillation (Base); on-policy distillation (Agarwal et al., 2024), where the student samples rollouts with teacher-provided token-level rewards; off-policy distillation, where the teacher generates rollouts; Veto (Jang et al., 2026), which keeps student rollouts but reshapes the supervision signal via a product of experts; MiniLLM (Gu et al., 2024), which blends teacher and student distributions in probability space with a fixed ratio and a length-normalized reweighting; and TRB (Plyusov et al., 2026), which blends in logit space with an optimized parameter.

Implementation Details. We implement all methods on the verl framework (Sheng et al., 2025), following the on-policy implementation of Li et al. (2026). For ABPD, we set the sharpening temperature $\tau = 0.7$ and the KL budget $\epsilon = 0.2$ by default, and also conduct ablation studies on these key hyperparameters. During training, we generate 4 responses per prompt for rollout collection. All models are trained with the same hyperparameters (learning rate 1×10^{-6} and warmup ratio 0.1) and the same number of gradient steps for fair comparison. All experiments are conducted on NVIDIA A800 GPUs.

Method	HumanEval		HumanEval+		MBPP		MBPP+		Overall	
	Avg	Pass	Avg	Pass	Avg	Pass	Avg	Pass	Avg	Pass
Teacher	63.7	78.7	58.4	73.2	77.6	87.0	65.0	74.1	66.2	78.3
Teacher: Qwen3-4B (non-thinking) *Student: Qwen3-1.7B (non-thinking)*										
Base	38.6	57.9	34.3	54.3	52.3	67.4	45.4	58.8	42.7	59.6
Off-policy	40.2	**63.4**	36.0	56.7	55.6	73.3	46.6	61.1	44.6	63.6
On-policy (Agarwal et al., 2024)	39.8	60.4	36.0	55.5	55.7	73.0	46.7	61.6	44.6	62.6
Veto (Jang et al., 2026)	41.5	61.6	37.4	58.6	56.0	**74.3**	45.4	63.2	45.1	64.4
MiniLLM (Gu et al., 2024)	41.0	61.0	36.3	53.7	**56.2**	73.5	46.8	61.6	45.1	62.5
TRB (Plyusov et al., 2026)	40.2	58.5	35.5	53.1	55.0	72.0	46.4	60.8	44.3	61.1
ABPD (Ours)	**43.2**	**63.4**	**38.2**	**59.2**	55.4	73.3	**48.4**	**63.9**	**46.3**	**65.0**

Table 3: Results on code benchmarks. "Avg" and "Pass" denote Avg@4 and Pass@4.

5.2 Overall Results

Mathematical Reasoning. Table 2 summarizes the results on six mathematical reasoning benchmarks, where ABPD consistently delivers the best overall performance across both model pairs. On the Qwen3 setting, ABPD achieves Avg@4 of 34.1 and Pass@4 of 49.7, outperforming all baselines; on the DeepSeek setting, it attains Avg@4 of 47.7 and Pass@4 of 63.9, with particularly strong gains on AMC23 (Pass@4: 95.0), AIME24 (Pass@4: 56.7), and OlympiadBench (Pass@4: 62.8). Notably, the improvement is more pronounced on Qwen3, where the teacher (Qwen3-4B-Base-GRPO) and the student (Qwen3-1.7B) differ more substantially in both scale and post-training (58.7 vs. 41.5 Pass@4, compared with 68.3 vs. 59.1 on DeepSeek). A wider gap amplifies blending drift for pure-policy rollouts, making ABPD's adaptive blending more beneficial.

Code Generation. Table 3 reports the results on four code generation benchmarks. ABPD achieves the highest average performance with Mean Avg@4 of 46.3 and Mean Pass@4 of 65.0, consistently outperforming all baselines. On HumanEval, ABPD attains 43.2 Avg@4 and 63.4 Pass@4, surpassing MiniLLM (41.0/61.0) and TRB (40.2/58.5). On the more challenging HumanEval+ and MBPP+ benchmarks, ABPD also achieves the highest Pass@4 of 59.2 and 63.9, respectively.

5.3 Ablation Studies

Effect of Sharpening Temperature τ. The temperature τ controls how sharply the teacher and student distributions are concentrated before blending, and thus governs the extent to which *blending drift* is suppressed. As shown in Figure 2(c,d), decreasing τ from 1.0 progressively reduces *blending drift*, consistent with the temperature-dependent bound in Theorem 3.2. However, sharpening also contracts the entropy of both distributions, trading off generation diversity: as quantified in Table 4, reducing τ from 1.0 to 0.7 improves Rollout accuracy (35.9 $\rightarrow$ 37.1) and Avg@4 (33.9 $\rightarrow$ 34.1) while reducing diversity (40.2 $\rightarrow$ 33.2), whereas $\tau = 0.5$ continues to suppress drift but degrades both accuracy (32.3) and diversity (27.0). Overall, $\tau = 0.7$ lies in a broad stable region, sharp enough to suppress drift while preserving diversity, and generalizes across tasks and model families.

Effect of KL Budget ϵ. The budget ϵ constrains the maximum KL divergence from the student. We sweep $\epsilon \in \{0.1, 0.2, 0.5\}$ and visualize the resulting β^* trajectories in Figure 4. With $\epsilon = 0.5$, β^* initially leans toward the teacher but drops sharply: a larger budget relaxes the constraint but does not keep the entire rollout teacher-aligned, since drift into regions unfamiliar to the student rapidly consumes the budget and forces β^* down. Conversely, $\epsilon = 0.1$ keeps β^* low across the early tokens, reducing ABPD to near on-policy distillation, which limits early teacher guidance and increases the risk of prefix failure. As shown in Table 5, this variant substantially lowers rollout accuracy from 37.1 to 31.4. The default $\epsilon = 0.2$ maintains a stable, moderate β^* throughout generation and achieves the best Avg@4 and Pass@4, generalizing across math and code tasks and across Qwen and DeepSeek families.

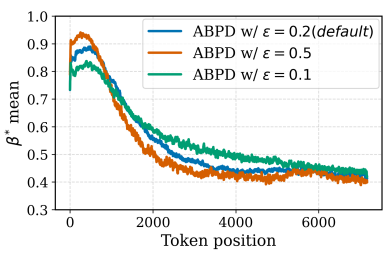


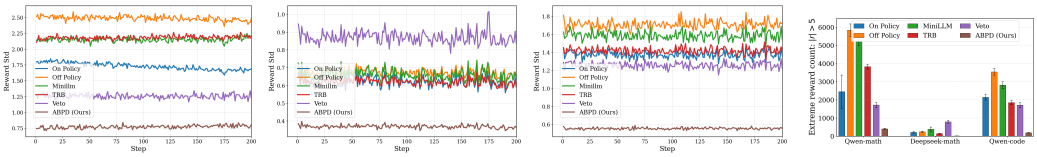
Figure 4: Visualization of β^* on the Qwen-math benchmarks.

	Rollout acc	Distinct 4-grams	Avg@4
$\tau = 1.0$	35.9	40.2	33.9
$\tau = 0.7$	37.1	33.2	34.1
$\tau = 0.5$	32.3	27.0	32.9

Table 4: Ablation of τ on Qwen-math.

	Rollout acc	Learnability	Avg@4
$\varepsilon = 0.1$	31.4	88.8	33.5
$\varepsilon = 0.2$	37.1	86.6	34.1
$\varepsilon = 0.5$	36.5	84.2	32.8

Table 5: Ablation of ε on Qwen-math.



(a) Qwen-Math (b) Deepseek-Math (c) Qwen-Code (d) Extreme |Reward|

Figure 5: Reward variance (a,b,c) and extreme reward count (d) across different model families and distillation methods. ABPD consistently reduces both the variance and the frequency of extreme rewards.

5.4 Stable Training with ABPD

On-policy distillation methods often suffer from extreme rewards and gradient explosion due to near-zero student probabilities on teacher-preferred tokens (Jang et al., 2026). ABPD addresses this from two aspects: it samples from the blended policy π^*, which accounts for both teacher and student, so rollouts rarely land on tokens with near-zero probability under either policy; and it computes the reward from the same π^* rather than from the teacher, avoiding the extreme log-ratios that arise when one policy assigns near-zero probability to a sampled token. As shown in Figure 5, this yields substantially more stable training and significantly fewer extreme rewards than prior methods.

6 Related Work

Knowledge distillation (KD) transfers reasoning capabilities from large teachers to smaller students. Existing RL-based KD methods largely follow two pure-policy paradigms. Off-policy distillation samples rollouts from the teacher, yielding high *rollout quality* but inducing train–test mismatch (Hinton et al., 2015; Guo et al., 2025); on-policy distillation samples from the student, improving *rollout learnability* but suffering from *prefix failure* (Agarwal et al., 2024; Gu et al., 2024; Jiang et al., 2026). Recent work mitigates this trade-off via filtering (Yu et al., 2026b) or objective redesign (Jang et al., 2026; Kim & Baek, 2026; Zhou et al., 2026; Fu et al., 2026; Zhao et al., 2026), yet the rollout mechanism remains a single pure policy.

Policy blending (Baram et al., 2021; He et al., 2026) offers a principled middle ground: MiniLLM (Gu et al., 2024) mixes teacher and student distributions in probability space, while TRB (Plyusov et al., 2026) blends in logit space. However, both suffer from blending drift. ABPD instead sharpens both distributions before blending and adaptively solves for a per-token mixing coefficient under a KL budget, yielding rollouts that stay close to the teacher while remaining within the student's confident region.

7 Conclusion

We theoretically define and analyze *blending drift*, and show that lowering the sampling temperature effectively controls it. Based on this finding, we propose ABPD, a novel sharpen-then-blend distillation framework that adaptively solves for a per-token mixing coefficient under a KL budget. Experiments on mathematical reasoning and code generation show that ABPD consistently outperforms existing methods across Qwen and DeepSeek families. However, both our analysis and our solution are built on the simplest form of policy blending, i.e., a probability-space MoE that averages the teacher and student distributions. Such a basic mixing scheme may not fully unlock the potential of blended policies, and future work will explore more expressive blending mechanisms.

AI USE STATEMENT

During the preparation of this work, the authors used GitHub Copilot for the following purposes:

- **Writing and Polishing:** GitHub Copilot was used to assist with language polishing, grammar correction, and improving the readability of selected sections of the manuscript. All AI-generated suggestions were carefully reviewed, edited, and approved by the authors.
- **Literature Search:** GitHub Copilot was used to help identify relevant prior work and suggest potential citations based on context. The final selection, verification, and curation of all references were performed manually by the authors.

We did *not* use generative AI tools for the following critical tasks:

- **Core Idea and Methodology:** The main research idea, theoretical derivations, the proposed ABPD algorithm, and the experimental design were developed entirely by the authors.
- **Data Collection and Processing:** All datasets and experimental data were collected, processed, and analyzed without the use of generative AI.
- **Result Analysis and Figure Generation:** The interpretation of experimental results and the creation of all figures and tables were performed by the authors.

We have reviewed all AI-assisted content and take full responsibility for the final content of this paper, including its accuracy, originality, and compliance with ethical guidelines. No proprietary or confidential data was shared with any AI tools during this research.

ETHICS STATEMENT

This work does not involve human subjects, private data, or sensitive applications. All datasets used are publicly available benchmarks. We do not foresee any direct negative societal impacts beyond those generally associated with large language models. We have adhered to the ICLR Code of Ethics throughout this research.

REPRODUCIBILITY STATEMENT

We provide the following to ensure reproducibility of our results:

- **Algorithm and Implementation:** The complete ABPD algorithm is detailed in Algorithm 1, with optimization procedures in Section 4.
- **Theoretical Proofs:** Full proofs of all theorems are provided in Appendix A.
- **Experimental Setup:** All hyperparameters, model configurations, and evaluation protocols are described in Section 4.1.
- **Datasets:** All datasets used (DAPO-Math-17K, MATH500, HumanEval, MBPP, etc.) are publicly available and are cited in Section 4.1.

REFERENCES

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

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.

Nir Baram, Guy Tennenholtz, and Shie Mannor. Maximum entropy reinforcement learning with mixture policies. *arXiv preprint arXiv:2103.10176*, 2021.

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. Minillm: Knowledge distillation of large language models. In *The twelfth international conference on learning representations*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

Jiamin He, Samuel Neumann, Jincheng Mei, Adam White, and Martha White. Revisiting mixture policies in entropy-regularized actor-critic. *arXiv preprint arXiv:2605.09157*, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

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

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, 2026.

Li Jiang, Haoran Xu, Yichuan Ding, and Amy Zhang. Trajectory-refined distillation. *arXiv preprint arXiv:2606.08432*, 2026.

Minsang Kim and Seung Jun Baek. Explain in your own words: Improving reasoning via token-selective dual knowledge distillation. *arXiv preprint arXiv:2603.13260*, 2026.

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

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.

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

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.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. Magicoder: Empowering code generation with oss-instruct. *arXiv preprint arXiv:2312.02120*, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026a.

Tianrun Yu, Kaixiang Zhao, Chih-Chun Chen, Amanda Hughes, Taylor W Killian, Fenglong Ma, Weitong Zhang, and Porter Jenkins. Lark: Learnability-grounded trajectory selection for efficient reasoning distillation. *arXiv preprint arXiv:2605.30651*, 2026b.

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

Yuhang Zhou, Lizhu Zhang, Yifan Wu, Mingyi Wang, Bo Peng, Jiayi Liu, Xiangjun Fan, and Zhuokai Zhao. Omniopd: Logit-free on-policy distillation via speculative verification. *arXiv preprint arXiv:2606.01476*, 2026.

Zhou Ziheng, Jiaqi Li, Huacong Tang, Ying Nian Wu, and Demetri Terzopoulos. Less is more: Early stopping rollout for on-policy distillation. *arXiv preprint arXiv:2605.27028*, 2026.