KB-OPD: KL-BRIDGE ON-POLICY DISTILLATION WITH BEHAVIOR-CORRECTED PREFIX GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) provides dense token-level supervision on student-generated trajectories, while generalized OPD (G-OPD) extends this framework through reward extrapolation. However, both depend on the prefixes visited by the student, which can limit access to productive reasoning trajectories early in training. Teacher guidance can improve these prefixes, but incorporating teacher-guided tokens into policy updates requires accounting for the mismatch between the sampling and student distributions. We propose **KB-OPD**, which combines constrained prefix sampling with reward extrapolation to learn from both guided prefixes and student-generated continuations. During a short warm-up phase, KB-OPD geometrically blends student and teacher distributions, selecting the largest teacher influence permitted by local and cumulative KL budgets. These constraints control both token-level and sequence-level deviation from the student, while a stopping rule hands generation back to the student for the remaining response. Clipped behavior importance weights and a weighted prefix–suffix objective incorporate bridge tokens into policy updates alongside student-generated tokens. After warm-up, sampling returns to pure student rollouts while retaining the G-OPD supervision signal. Experiments on mathematical reasoning and code generation show improved average performance over OPD and ExOPD across single-teacher, multi-teacher, and strong-to-weak distillation settings.

1 INTRODUCTION

On-policy distillation (OPD) has emerged as a powerful paradigm for capability transfer in autoregressive language models, providing dense, token-level supervision directly on student-generated rollouts (Agarwal et al., 2024; Gu et al., 2024). By evaluating teacher feedback along the student's own trajectory, OPD inherently mitigates training-inference distribution shifts (Bengio et al., 2015; Ross et al., 2011). Recent advances, notably generalized OPD (G-OPD) with reward extrapolation, demonstrate that dense KL-regularized signals even enable students to outperform their domain teachers (Yang et al., 2026). Consequently, OPD has become central to merging specialized RL experts (Xiaomi LLM-Core Team et al., 2026; Liang et al., 2026) and compressing frontier models into compact architectures (DeepSeek-AI et al., 2025; Yang et al., 2025). Nevertheless, the fundamental efficacy of OPD remains strictly constrained by the support of trajectories the student can self-generate.

This generative dependency poses a critical bottleneck during early training. When tackling complex reasoning, a sub-optimal student frequently initiates responses with unpromising prefixes (Wei et al., 2022; Ye et al., 2025b), thereby failing to visit informative state spaces from which productive behaviors can be reinforced. While teacher-guided prefix interventions can steer subsequent generation toward successful outcomes (Plyusov et al., 2026), prefix tokens themselves encapsulate crucial reasoning steps that the student must assimilate. Crucially, simply enforcing teacher-generated tokens does not guarantee learnability: substantial distributional discrepancies between teacher and student render naive prefix updates prone to catastrophic training collapse and severe off-policy mismatch (Tang et al., 2024; Liu et al., 2025). The core challenge therefore lies in curating high-quality reasoning prefixes that stay within the student's reachable vicinity, while enabling principled policy optimization over both guided prefixes and autonomous continuations.

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.

Addressing this question requires coordinating prefix sampling with policy optimization. During sampling, teacher guidance should improve the prefix while controlling its deviation from the student distribution (Schulman et al., 2015; Kakade & Langford, 2002). This control must account for both individual token decisions and the deviation accumulated across the prefix. During optimization, guided tokens should receive weights that account for their behavior distribution (Precup et al., 2000; Munos et al., 2016). Together, these controls make guided prefixes accessible for exploration and usable for learning.

To make guided prefixes part of student learning, we introduce KB-OPD (**KL-Bridge On-Policy Distillation**), a unified framework that operationalizes adaptive, trust-region prefix exploration. During a brief initial warm-up, a geometric bridge policy interpolates student and teacher distributions, greedily selecting the maximal teacher influence permitted under dual local and cumulative KL budgets.

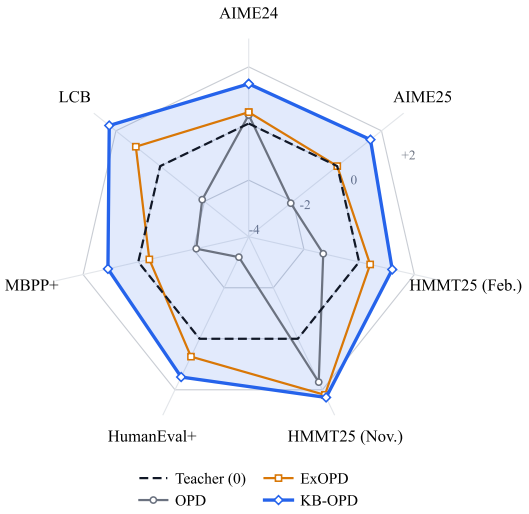


Figure 1: Gain over the domain teacher in percentage points under single-teacher distillation. The dashed ring marks the teacher's own accuracy, so a curve outside it beats the teacher (see Table 1).

Figure 1 previews the effect. Each axis gives the gain over the domain teacher on one benchmark in the single-teacher setting, so the dashed ring marks the teacher's own accuracy and a curve outside the ring means the distilled student beats its teacher. KB-OPD clears the ring on all seven benchmarks, whereas OPD stays inside it on five and ExOPD dips below it on one.

Our main contributions are:

- We introduce KB-OPD, which trains on both teacher-guided prefixes and student-generated continuations through constrained bridge sampling and behavior-corrected, reward-extrapolated updates.
- We develop local and cumulative KL constraints for adaptive prefix guidance and establish a sequence-level KL bound on the resulting rollout distribution.
- Extensive experiments demonstrate improvements over OPD and ExOPD across three distillation settings, with ablations examining prefix learning, behavior correction, and the guidance constraints.

2 Related Work

On-Policy Distillation and Extrapolation. Knowledge distillation traditionally trains students to match teacher predictions or imitate teacher-generated sequences (Hinton et al., 2015; Kim & Rush, 2016; Sanh et al., 2019). For autoregressive models, training on fixed sequences creates a mismatch with student-generated contexts at inference time (Ross et al., 2011; Bengio et al., 2015). Generalized Knowledge Distillation (GKD) addresses this mismatch by evaluating teacher feedback on student-generated trajectories (Agarwal et al., 2024), while MiniLLM develops an optimization framework for reverse-KL distillation (Gu et al., 2024). Other approaches explore alternative distillation objectives, including f-divergence minimization and DistiLLM (Wen et al., 2023; Ko et al., 2024). Building on the connection between OPD and reinforcement learning (Xiao et al., 2025), G-OPD formulates distillation through dense KL-regularized RL and introduces reward extrapolation (Yang et al., 2026). This formulation allows students to improve beyond direct imitation and, in the evaluated settings, surpass their teachers. Recent analyses further examine the conditions supporting effective on-policy learning (Lu & Thinking Machines Lab, 2025; Li et al., 2026). Despite these advances, supervision remains tied to the trajectories the student can generate. KB-OPD retains the dense extrapolated supervision signal while improving prefix exploration during early training.

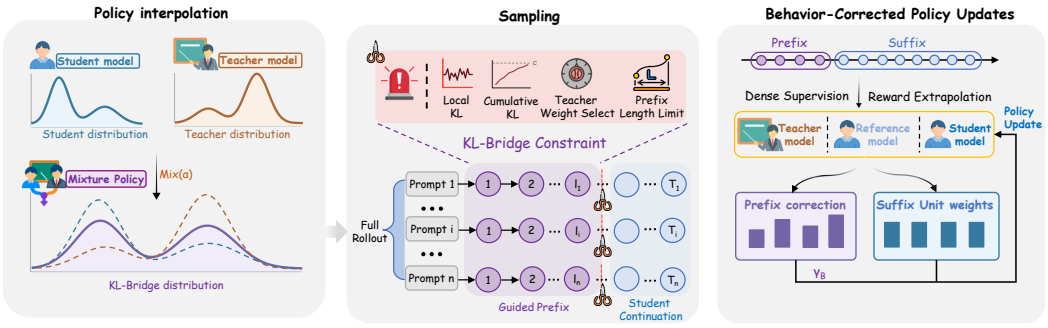


Figure 2: Overview of KB-OPD during warm-up. Left: student–teacher policy interpolation. Middle: constrained sampling produces guided prefixes (purple) of varying lengths, followed by student continuations (blue). Right: both segments receive dense supervision; prefix tokens use clipped behavior weights, while suffix tokens use unit weights.

Supervisory Targets and Token Selection. Another line of work improves distillation by refining teacher supervision or selecting informative tokens. Veto constructs an intermediate student–teacher target to stabilize distillation (Jang et al., 2026), while entropy-aware OPD supplements reverse KL with forward KL at states with high teacher uncertainty (Jin et al., 2026). TIP uses student entropy and teacher–student disagreement to identify informative token positions (Xu et al., 2026). Related approaches construct supervision through privileged information (Penaloza et al., 2026) or self-distillation (Hübotter et al., 2026; Shenfeld et al., 2026; Zhao et al., 2026), while token pruning removes uninformative positions (Guo et al., 2025). These methods primarily improve the learning signal at visited states. KB-OPD addresses the complementary problem of which prefixes are visited during data collection. It guides early rollouts toward teacher-supported prefixes and incorporates their tokens into student updates through explicit behavior weighting.

Interactive and Guided Sampling. Teacher guidance can improve training trajectories while retaining student participation. Speculative Knowledge Distillation (SKD) combines student token proposals with teacher-based replacement (Xu et al., 2025), while TRB explores geometric behavior blending under a local KL constraint (Plyusov et al., 2026). Other approaches address different barriers to knowledge transfer: MiCoTA uses intermediate assistants in offline distillation (Ding et al., 2025), and cross-family methods study settings where teacher logits are unavailable (Patiño et al., 2025; Ye et al., 2025a). KB-OPD focuses on jointly controlling prefix generation and learning from the resulting guided tokens. Local and cumulative KL budgets regulate teacher influence within an adaptive prefix, followed by student-generated continuation. Clipped behavior importance weights account for the sampling mismatch when prefix tokens participate in reward-extrapolated updates. The guided prefix thus serves both as context for continuation and as training data, with its contribution balanced against the student suffix.

3 KB-OPD

Figure 2 illustrates the three components of KB-OPD: policy interpolation, constrained prefix sampling, and behavior-corrected policy updates. During warm-up, the bridge policy blends student and teacher distributions to generate prefixes under local and cumulative KL budgets. Guidance ends when no mixing coefficient is feasible or the prefix length limit is reached, and the student completes the response. Both segments contribute to reward-extrapolated updates, with behavior correction applied to guided tokens.

3.1 Preliminaries

Given an input prompt x, an autoregressive language model generates a response $y = (y_1, \ldots, y_T)$ sequentially, where $s_t = (x, y_{<t})$ denotes the generation state at step t. We denote the student policy being optimized by π_θ, its rollout snapshot frozen at the start of each training iteration by π_S, and the domain expert teacher by π_T. The reference policy π_{ref} corresponds to the frozen base student

prior to distillation. Both teacher and student policies operate over a shared vocabulary $\mathcal{V}$ with full support under temperature 1.

Dense extrapolated supervision. Standard OPD optimizes reverse-KL divergence on student-generated rollouts (Agarwal et al., 2024; Gu et al., 2024). Generalized OPD (G-OPD) extends this formulation to dense KL-regularized RL by introducing an extrapolation coefficient $\lambda \geq 1$ (Yang et al., 2026), yielding the parameterized token-level target:

$$\widetilde{\pi}_\lambda(a \mid s_t) = \frac{\pi_T(a \mid s_t)^\lambda \pi_{\text{ref}}(a \mid s_t)^{1-\lambda}}{Z_\lambda(s_t)}, \tag{1}$$

where $Z_\lambda(s_t) = \sum_{a' \in \mathcal{V}} \pi_T(a' \mid s_t)^\lambda \pi_{\text{ref}}(a' \mid s_t)^{1-\lambda}$ is the partition function. For $\lambda = 1$, the target recovers the teacher distribution; when $\lambda > 1$, it extrapolates the log-likelihood displacement between the expert and base models (see Appendix A.1 for full derivation).

For conciseness, we abbreviate token probabilities as $p_t^k = \pi_k(y_t \mid s_t)$ for $k \in \{T, S, \text{ref}, \theta\}$. The resulting dense token advantage is defined as:

$$A_t = \text{clip}\big(\lambda \log p_t^T - \log p_t^S + (1-\lambda) \log p_t^{\text{ref}},\, -c_A,\, c_A\big)\,, \tag{2}$$

where $c_A = 2$ truncates advantage variance and is held detached during backpropagation. This dense signal assigns credits to all generated tokens. While λ governs the optimality target during policy updates, the mixing parameter α below governs the generative behavior during trajectory sampling.

3.2 KL-Bridge Prefix Sampling

Geometric policy interpolation. To systematically bridge the gap between student capability and teacher guidance, we define a geometric log-linear mixture policy parameterized by a mixing coefficient $\alpha \in [0, 1]$:

$$q_\alpha(a \mid s_t) = \frac{\pi_S(a \mid s_t)^{1-\alpha} \pi_T(a \mid s_t)^\alpha}{Z_\alpha(s_t)}, \tag{3}$$

which is equivalent to applying softmax to interpolated logits $z_\alpha = (1-\alpha) z_S + \alpha z_T$. Here, α smoothly modulates teacher injection, interpolating smoothly between the student $q_0 = \pi_S$ and the teacher $q_1 = \pi_T$.

Adaptive trust-region guidance under dual budgets. A viable bridge policy must maximize teacher expertise while preserving student proximity. At state s_t, the divergence incurred by candidate coefficient α relative to the rollout student is quantified by:

$$d_\alpha(s_t) = \mathrm{D}_{\text{KL}}(q_\alpha(\cdot \mid s_t) \,\|\, \pi_S(\cdot \mid s_t))\,. \tag{4}$$

Letting $h_t(a) = \log[\pi_T(a \mid s_t)/\pi_S(a \mid s_t)]$, the differential bridge divergence satisfies:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} d_\alpha(s_t) = \alpha \ \text{Var}_{q_\alpha}[h_t] \geq 0. \tag{5}$$

Monotonicity guarantees that larger mixing values induce strictly non-decreasing divergence (Appendix A.2). Given a discrete candidate set $\mathcal{A}_+ \subseteq [\alpha_{\min}, \alpha_{\max}]$, KB-OPD greedily selects the maximal teacher intervention bounded by both a local threshold η and a cumulative prefix budget C:

$$\alpha_t^\star = \max\{\alpha \in \mathcal{A}_+ : d_\alpha(s_t) \leq \min(\eta,\, C - C_{t-1})\}\,, \tag{6}$$

where $C_0 = 0$ and the accumulator tracks consumed divergence via $C_t = C_{t-1} + d_{\alpha_t^\star}(s_t)$ upon selecting $\alpha_t^\star$. The local constraint η bounds single-step distribution distortion, while the cumulative constraint C prevents multi-step trajectory drift. By default, $\mathcal{A}_+ = \{0.1, 0.2, \ldots, 0.5\}$, $\eta = 0.05$, and $C = 1.5$.

Prefix termination and student handover. Sampling transitions entirely to the student policy π_S as soon as no candidate in $\mathcal{A}_+$ satisfies equation 6, or when reaching a maximum prefix length $L_{\max} = 512$ (or an end-of-sequence token). Denoting the dynamic bridge prefix length by T_b, the full rollout probability factorizes into:

$$P_{\mathrm{KB}}(y \mid x) = \prod_{t=1}^{T_b} q_{\alpha_t^\star}(y_t \mid s_t) \prod_{t=T_b+1}^{T} \pi_S(y_t \mid s_t). \tag{7}$$

Because the handover rule dynamically adapts to problem difficulty, T_b varies across instances. We denote the actual sampling behavior policy by b_t: it aligns with $q_{\alpha_t^\star}$ along the bridge prefix and reverts to π_S for subsequent tokens. Rollout probabilities $b_t(y_t \mid s_t)$ and $\pi_S(y_t \mid s_t)$ are recorded to compute off-policy corrections.

Sequence-level distribution bound. Let $P_S(y \mid x)$ denote the pure student rollout distribution under identical prompt and termination conditions. Applying the autoregressive chain rule for relative entropy yields:

$$\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{KB}} \,\|\, P_S) = \mathbb{E}_{y \sim P_{\mathrm{KB}}}\left[\sum_{t=1}^{T_b} d_{\alpha_t^\star}(s_t)\right] \le C. \tag{8}$$

Because autonomous student continuations incur zero conditional divergence relative to π_S, the total trajectory divergence from the student policy remains strictly bounded by C on every realized rollout (see Appendix A.3).

3.3 Behavior-Corrected Policy Updates

Clipped importance weighting. Since prefix tokens are sampled from the blended behavior policy b_t rather than the student snapshot π_S, naive gradient updates induce distribution mismatch. We correct for this discrepancy by introducing a truncated behavior importance weight:

$$w_t = \mathrm{clip}\left(\frac{p_t^S}{b_t(y_t \mid s_t)},\, w_{\min},\, w_{\max}\right), \tag{9}$$

with $[w_{\min}, w_{\max}] = [0.5, 2.0]$. Along student-generated suffixes, $b_t \equiv \pi_S$, yielding $w_t = 1$ automatically. Behavior weights are detached from computation graphs during optimization.

Policy optimization surrogate. Let $r_t(\theta) = p_t^\theta / p_t^S$ denote the standard PPO policy ratio. Multiplying $r_t(\theta)$ by the nominal behavior ratio $\rho_t = p_t^S / b_t(y_t \mid s_t)$ recovers the exact action-level importance ratio relative to the sampling policy:

$$r_t(\theta)\, \rho_t = \frac{\pi_\theta(y_t \mid s_t)}{b_t(y_t \mid s_t)}. \tag{10}$$

We define the clipped PPO objective (Schulman et al., 2017) with clipping margin $\epsilon = 0.2$:

$$\ell_t(\theta) = -\min\{r_t(\theta)A_t,\, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\}\,. \tag{11}$$

Crucially, the behavior weight w_t modulates the update externally: $\widetilde{\ell}_t(\theta) = w_t \ell_t(\theta)$. This decoupled structure ensures that w_t corrects for sampling deviation from π_S, while PPO clipping retains its established trust-region control on the optimizing policy π_θ relative to π_S.

Balancing prefix assimilation and autonomous learning. Let B_i and S_i denote the token sets corresponding to the bridge prefix and student suffix for instance i. For any non-empty segment $K_i \in \{B_i, S_i\}$, the segment-averaged loss is computed as:

$$\mathcal{L}_K^{(i)}(\theta) = \frac{1}{|K_i|} \sum_{t \in K_i} w_t \ell_t(\theta). \tag{12}$$

To prevent long response suffixes from overwhelming prefix updates, we combine both branches via a normalized segment weighting:

$$\mathcal{L}^{(i)}(\theta) = \frac{\gamma_B \mathcal{L}_B^{(i)}(\theta) + \mathcal{L}_S^{(i)}(\theta)}{\gamma_B + 1}, \tag{13}$$

where $\gamma_B = 0.25$ balances prefix and suffix contributions (see Appendix A.5 for edge cases where $B_i = \varnothing$). The final batch loss averages over all N trajectories:

$$\mathcal{L}_{\mathrm{KB}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}^{(i)}(\theta). \tag{14}$$

Gradients flow exclusively through π_θ; all rollout statistics, advantages, and importance weights remain detached.

3.4 Warm-Up Schedule

Bridge sampling operates strictly during an initial warm-up window of w steps. Once $g > w$, generation reverts entirely to autonomous student rollouts ($b_t \equiv \pi_S$), causing all behavior weights to collapse to $w_t \equiv 1$ and simplifying the surrogate loss to the standard G-OPD objective. The teacher is removed from the rollout loop, retaining active forward passes solely for trajectory scoring. We set $w = 5$ and $\lambda = 1.25$ by default. The complete training procedure is formalized in Algorithm 1 (Appendix C).

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate mathematical reasoning on AIME24 (AI-MO, 2024), AIME25 (OpenCompass, 2025), and HMMT25 (Feb./Nov.) (Balunović et al., 2025), and code generation on HumanEval+, MBPP+ (Liu et al., 2023), and LiveCodeBench (Jain et al., 2025). Following G-OPD (Yang et al., 2026), training uses the difficulty ≥ 6 split of DeepMath-103K (57K) (He et al., 2025) for math and Eurus-RL-Code (25K) (Cui et al., 2025) for code. For math, we report average accuracy estimated by averaging per-response correctness over 32 independent samples per problem (Math-Verify (Kydlicek, 2025)); for code, we report pass@1 estimated from 4 samples per problem (execution sandboxes; see Appendix B.1).

Baselines. We benchmark KB-OPD against standard OPD ($\lambda = 1$) (Agarwal et al., 2024; Gu et al., 2024) and ExOPD ($\lambda = 1.25$) (Yang et al., 2026), alongside the base student and domain teachers. All distillation baselines use pure student rollouts and share the identical RL backbone, prompts, reference policy, and evaluation setup across single-teacher, multi-teacher, and strong-to-weak settings.

Implementation Details. We adopt Qwen3-4B-Non-Thinking (Yang et al., 2025) as the base student and its domain RL checkpoints as teachers; Qwen3-30B-A3B-Instruct-2507 serves as the expert for 4B and 1.7B students in strong-to-weak distillation. We train via verl (Sheng et al., 2025) with GRPO (Shao et al., 2024) for 50 steps (batch size 1024, learning rate 10^{-5}, evaluated every 5 steps). KB-OPD sets $w = 5$, $\eta = 0.05$, $C = 1.5$, and $L_{\max} = 512$ (full hyperparameters in Appendix B.2). We follow the training and evaluation prompts of G-OPD (Yang et al., 2026), provided in Appendix D.

4.2 Single-Teacher and Multi-Teacher Distillation

As shown in Table 1, KB-OPD achieves the highest domain averages in both single-teacher (49.5% math, 62.3% code) and multi-teacher settings (48.8% math, 61.9% code). Crucially, single-teacher KB-OPD surpasses domain teachers across all seven individual benchmarks. In multi-teacher distillation, it leads on six of seven benchmarks, confirming that behavior-corrected prefix exploration generalizes effectively to multi-expert consolidation without compromising asymptotic performance.

4.3 Strong-to-Weak Distillation

Table 2 compares both extrapolated methods with OPD under the same strong-to-weak setting. KB-OPD improves the math average over OPD by 2.2 percentage points for the 4B student and 1.1

Table 1: Main distillation results on same-size Qwen3-4B models. Teacher denotes the corresponding domain teacher; Student denotes the initial student. Bold marks the best distillation score within each setting and benchmark. Superscripts show increases (↑) or decreases (↓) relative to the domain teacher, in percentage points. Math results report average accuracy; code results report pass@1. Scores are percentages.

Method	Math Reasoning					Code Generation			
	AIME24	AIME25	HMMT25 (Feb.)	HMMT25 (Nov.)	Avg.	HumanEval+	MBPP+	LCB	Avg.
Teacher	62.9	54.5	35.4	38.8	47.9	85.2	69.8	26.9	60.6
Student	21.9	22.7	10.9	9.0	16.1	76.2	64.6	17.1	52.6
Single-Teacher Distillation									
OPD	63.2$^{\uparrow 0.3}$	52.4$^{\downarrow 2.1}$	34.1$^{\downarrow 1.3}$	40.5$^{\uparrow 1.7}$	47.6$^{\downarrow 0.3}$	82.0$^{\downarrow 3.2}$	67.7$^{\downarrow 2.1}$	25.0$^{\downarrow 1.9}$	58.2$^{\downarrow 2.4}$
ExOPD	63.3$^{\uparrow 0.4}$	54.5$^{\uparrow 0.0}$	35.8$^{\uparrow 0.4}$	41.0$^{\uparrow 2.2}$	48.7$^{\uparrow 0.8}$	85.9$^{\uparrow 0.7}$	69.4$^{\downarrow 0.4}$	28.0$^{\uparrow 1.1}$	61.1$^{\uparrow 0.5}$
KB-OPD	**64.3**$^{\uparrow 1.4}$	**56.0**$^{\uparrow 1.5}$	**36.6**$^{\uparrow 1.2}$	**41.1**$^{\uparrow 2.3}$	**49.5**$^{\uparrow 1.6}$	**86.7**$^{\uparrow 1.5}$	**70.9**$^{\uparrow 1.1}$	**29.2**$^{\uparrow 2.3}$	**62.3**$^{\uparrow 1.7}$
Multi-Teacher Distillation									
OPD	62.3$^{\downarrow 0.6}$	52.1$^{\downarrow 2.4}$	**36.9**$^{\uparrow 1.5}$	39.6$^{\uparrow 0.8}$	47.7$^{\downarrow 0.2}$	85.5$^{\uparrow 0.3}$	69.8$^{\uparrow 0.0}$	25.0$^{\downarrow 1.9}$	60.1$^{\downarrow 0.5}$
ExOPD	62.8$^{\downarrow 0.1}$	54.3$^{\downarrow 0.2}$	35.1$^{\downarrow 0.3}$	40.2$^{\uparrow 1.4}$	48.1$^{\uparrow 0.2}$	85.6$^{\uparrow 0.4}$	70.0$^{\uparrow 0.2}$	27.7$^{\uparrow 0.8}$	61.1$^{\uparrow 0.5}$
KB-OPD	**63.1**$^{\uparrow 0.2}$	**55.7**$^{\uparrow 1.2}$	35.2$^{\downarrow 0.2}$	**41.1**$^{\uparrow 2.3}$	**48.8**$^{\uparrow 0.9}$	**87.0**$^{\uparrow 1.8}$	**70.2**$^{\uparrow 0.4}$	**28.4**$^{\uparrow 1.5}$	**61.9**$^{\uparrow 1.3}$

Table 2: Strong-to-weak distillation on four math benchmarks. The teacher is Qwen3-30B-A3B-Instruct-2507; Base denotes the initial student. Scores are average accuracy (%). Superscripts show percentage-point changes relative to OPD for the same student size. Bold marks the best distillation score in each column.

Method	AIME24	AIME25	HMMT25 (Feb.)	HMMT25 (Nov.)	Avg.
Teacher	74.5	61.8	43.2	57.2	59.2
Student: Qwen3-4B-Non-Thinking					
Base	21.9	22.7	10.9	9.0	16.1
OPD	57.5	49.1	30.7	36.9	43.6
ExOPD	58.8$^{\uparrow 1.3}$	49.6$^{\uparrow 0.5}$	**31.6**$^{\uparrow 0.9}$	39.6$^{\uparrow 2.7}$	44.9$^{\uparrow 1.3}$
KB-OPD	**59.0**$^{\uparrow 1.5}$	**53.1**$^{\uparrow 4.0}$	31.1$^{\uparrow 0.4}$	**40.1**$^{\uparrow 3.2}$	**45.8**$^{\uparrow 2.2}$
Student: Qwen3-1.7B-Non-Thinking					
Base	13.0	12.2	5.4	4.7	8.8
OPD	36.7	29.4	15.9	15.5	24.4
ExOPD	37.4$^{\uparrow 0.7}$	30.9$^{\uparrow 1.5}$	15.6$^{\downarrow 0.3}$	**16.3**$^{\uparrow 0.8}$	25.1$^{\uparrow 0.7}$
KB-OPD	**37.6**$^{\uparrow 0.9}$	**31.7**$^{\uparrow 2.3}$	**16.4**$^{\uparrow 0.5}$	16.2$^{\uparrow 0.7}$	**25.5**$^{\uparrow 1.1}$

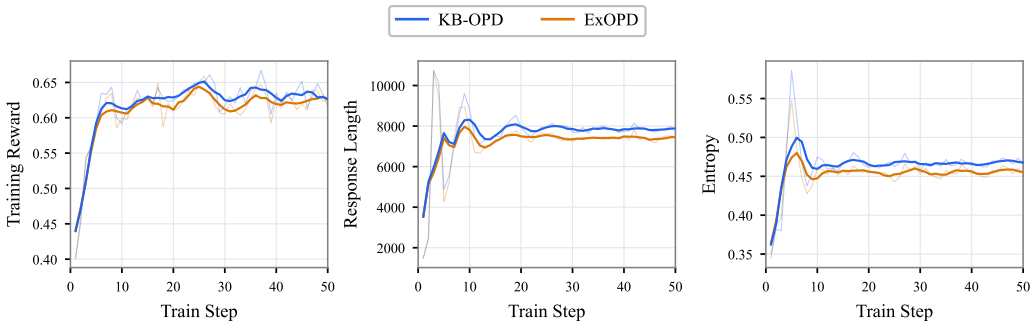
points for the 1.7B student, exceeding the corresponding ExOPD gains of 1.3 and 0.7 points. It also improves on OPD across all four benchmarks for both student sizes.

4.4 Training Dynamics

Figure 3 compares 50 steps of multi-teacher training. Switching to student-only sampling after step 5 causes no visible discontinuity in these runs. Over steps 11–50, KB-OPD has higher reward on 31 of 40 steps and maintains longer responses and higher policy entropy than ExOPD.

5 Ablation Studies and Analysis

We study warm-up duration, individual components, teacher mixing, and reward extrapolation in the single-teacher math setting.



(a) Training Reward (b) Response length (c) Entropy

Figure 3: Multi-teacher training dynamics: (a) reward, (b) response length, and (c) policy entropy. Thin curves show raw values; thick curves show five-step moving averages.

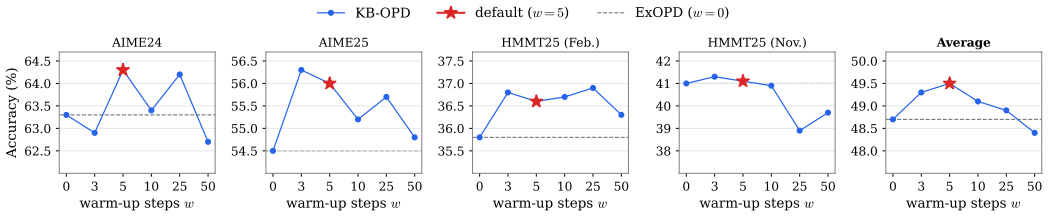


Figure 4: Warm-up duration on four math benchmarks and their average. $w = 0$ denotes pure student sampling throughout training.

5.1 Impact of Warm-Up Duration

Figure 4 evaluates distillation performance across warm-up durations $w \in \{0, 3, 5, 10, 25, 50\}$. A transient warm-up achieves the strongest overall transfer, peaking at $w = 5$ (49.5%) closely followed by $w = 3$ (49.3%), whereas unguided distillation ($w = 0$) reaches only 48.7%. Conversely, extending teacher intervention across all 50 steps lowers the average to 48.4%. While optimal duration slightly varies across individual tasks (e.g., $w = 25$ on HMMT25-Feb), a brief initial warm-up provides the best global balance between seeding high-quality prefix trajectories and preserving subsequent autonomous policy optimization.

5.2 Component-Wise Mechanism Ablation

Table 3 evaluates individual components with $w = 5$. Removing prefix updates ($\gamma_B = 0$) lowers the average from 49.5 to 48.2, even though the prefix still provides context for student continuation. This supports directly learning from guided tokens rather than using them only as context. Disabling behavior correction reduces the average to 47.8, showing the benefit of accounting for the bridge sampling distribution.

Both KL constraints contribute to performance. Removing the local constraint causes the largest drop among these ablations, reducing the average to 46.3. Removing the cumulative budget lowers it to 48.0. Replacing the blended prefix with a pure-teacher prefix ($\alpha = 1$) yields 46.8, supporting the use of student–teacher blending during guided sampling.

5.3 Sensitivity to Teacher Mixing Limit

The coefficient $\alpha_{\max}$ bounds the allowable teacher influence during bridge sampling. Table 4 evaluates $\alpha_{\max} \in \{0.50, 0.75, 0.90, 1.00\}$. The default limit $\alpha_{\max} = 0.50$ consistently performs best across all four benchmarks. As the ceiling increases, the domain average monotonically declines from 49.5% to 48.2%, 47.5%, and 46.9%. These results confirm that excessive teacher influence

Table 3: Component ablations for Qwen3-4B on math ($w = 5$; average accuracy, %). The pure-teacher variant fixes $\alpha_{\min} = \alpha_{\max} = 1$. Bold marks the best score per column.

Method	AIME24	AIME25	HMMT25 (Feb.)	HMMT25 (Nov.)	Avg.
KB-OPD (Default)	**64.3**	**56.0**	**36.6**	**41.1**	**49.5**
w/o Prefix update ($\gamma_B = 0$)	62.7	54.1	35.2	40.9	48.2
w/o Behavior correction ($w_t \equiv 1$)	61.1	54.6	35.2	40.4	47.8
w/o Local KL constraint (η)	60.1	53.6	33.0	38.4	46.3
w/o Cumulative KL budget (C)	63.0	53.0	35.3	40.6	48.0
Pure-teacher prefix ($\alpha = 1.0$)	62.1	52.2	33.7	39.0	46.8

Table 4: Sensitivity to the teacher mixing limit $\alpha_{\max}$ (average accuracy, %). Bold marks the best score per column.

$\alpha_{\max}$	AIME24	AIME25	HMMT25 (Feb.)	HMMT25 (Nov.)	Avg.
0.50 (Default)	**64.3**	**56.0**	**36.6**	**41.1**	**49.5**
0.75	62.4	55.1	35.2	40.1	48.2
0.90	62.1	53.8	35.7	38.4	47.5
1.00	61.9	52.3	34.7	38.8	46.9

drives rollouts away from the student's support, favoring a moderate mixing ceiling alongside the KL constraints.

5.4 Sensitivity to Reward Extrapolation

Figure 5 profiles performance across reward-scaling coefficients $\lambda \in [1.0, 1.75]$. The domain accuracy displays an inverted-U trend, rising from 47.6% at $\lambda = 1.0$ to a peak of 49.5% at $\lambda = 1.25$, before declining under excessive scaling ($\lambda \geq 1.50$). This indicates that moderate extrapolation provides more informative corrective signals, whereas overly aggressive scaling injects pseudo-reward noise. Importantly, KB-OPD outperforms standard OPD across $\lambda \in [1.10, 1.50]$, demonstrating that constrained prefix exploration reliably broadens the effective extrapolation window.

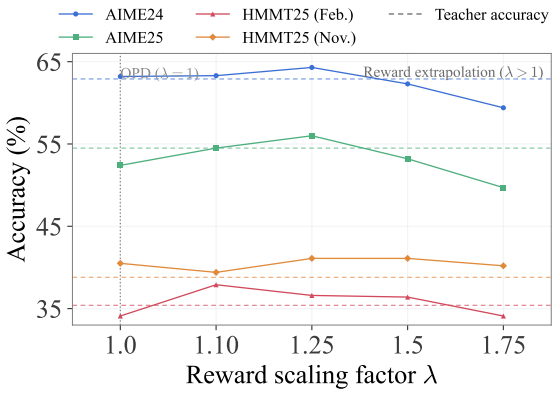


Figure 5: Math performance across reward-scaling coefficients λ.

6 Conclusion

We introduced KB-OPD, an on-policy distillation framework that mitigates the early exploration bottleneck by coupling dual-budget bridge prefix sampling with behavior-corrected policy optimization and reward extrapolation. Across rigorous mathematical reasoning and code generation benchmarks, KB-OPD consistently improves over standard and extrapolated OPD baselines across diverse distillation regimes, confirming the value of principled prefix intervention.

Limitations and Future Work. Our study primarily examines reasoning tasks within the Qwen3 architecture under static warm-up schedules. Future work includes extending this paradigm to multimodal domains and developing adaptive, student-dependent KL budget schedules to further reduce teacher inference overhead.

AI USE STATEMENT

Generative AI tools, including Gemini 3.8 Flash and GPT-5.6 sol, assisted with polishing, literature checks, and the presentation and checking of mathematical arguments. The authors are responsible for verifying all AI-assisted content and for the accuracy and integrity of the final paper.

ETHICS STATEMENT

This work uses public benchmarks and open-weight models (Yang et al., 2025) and involves no human-subject studies or collection of personal data. Distillation may transfer teacher biases and harmful capabilities alongside task performance (Ouyang et al., 2022). Our evaluation measures reasoning and coding accuracy and does not establish model safety or alignment.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 describe KB-OPD, with derivations, training settings, and data processing provided in Appendices A, B.2, and B.1. We use publicly available models and datasets and report the evaluation protocol in the experimental setup. We plan to release the implementation, based on verl (Sheng et al., 2025), together with training and evaluation scripts.

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 (ICLR)*, 2024.

AI-MO. AI-MO aimo validation aime (2022–2024). `https://huggingface.co/datasets/AI-MO/aimo-validation-aime`, 2024.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. doi: 10.52202/085713-0679. URL `https://matharena.ai/`. Platform includes HMMT February and November 2025.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dongyi Ding, Tiannan Wang, Chenghao Zhu, Meiling Tao, Yuchen Eleanor Jiang, and Wangchunshu Zhou. MiCoTA: Bridging the learnability gap with intermediate CoT and teacher assistants. *arXiv preprint arXiv:2507.01887*, 2025.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Yiju Guo, Wenkai Yang, Zexu Sun, Ning Ding, Zhiyuan Liu, and Yankai Lin. Learning to focus: Causal attention distillation via gradient-guided token pruning. *arXiv preprint arXiv:2506.07851*, 2025.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

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

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/94074dd5a072d28ff75a76dabed43767-Abstract-Conference.html.

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. doi: 10.18653/v1/2026.findings-acl.2094. URL https://aclanthology.org/2026.findings-acl.2094/.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, pp. 267–274, 2002.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1317–1327, 2016.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pp. 24872–24895, 2024. URL https://proceedings.mlr.press/v235/ko24c.html.

Hynek Kydlicek. Math-Verify: Math verification library, 2025. URL https://github.com/huggingface/Math-Verify.

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

Kun Liang, Clive Bai, Xin Xu, Chenming Tang, Sanwoo Lee, Weijie Liu, Saiyong Yang, and Yunfang Wu. ORBIT: On-policy exploration-exploitation for controllable multi-budget reasoning. *arXiv preprint arXiv:2601.08310*, 2026.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Yu Shen. When speed kills stability: Demystifying RL collapse from the training-inference mismatch, 2025. URL https://richardli.xyz/rl-collapse.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

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

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. *Advances in Neural Information Processing Systems*, 29, 2016.

OpenCompass. OpenCompass aime2025 dataset. https://huggingface.co/datasets/opencompass/AIME2025, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family, 2025. URL https://huggingfaceh4-on-policy-distillation.hf.space/.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

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.

Doina Precup, Richard S. Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pp. 759–766, 2000.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 627–635, 2011.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, pp. 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

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 20th European Conference on Computer Systems (EuroSys)*, pp. 1279–1297, 2025. URL https://arxiv.org/abs/2409.19256.

Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, et al. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 10817–10834, 2023. doi: 10.18653/v1/2023.acl-long.605.

Teng Xiao, Yige Yuan, Mingxiao Li, Zhengyu Chen, and Vasant G. Honavar. On a connection between imitation learning and RLHF. In *International Conference on Learning Representations (ICLR)*, 2025.

Xiaomi LLM-Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, et al. MiMo-v2-Flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

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

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026.

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.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025a.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025b.

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.

A Mathematical Derivations

A.1 From G-OPD to the Dense Advantage

At a fixed state s, consider a candidate next-token policy $p(\cdot \mid s)$ and the local KL-regularized objective:

$$J_s(p) = \lambda\, \mathbb{E}_{a\sim p}\left[\log \frac{\pi_T(a \mid s)}{\pi_{\mathrm{ref}}(a \mid s)}\right] - \mathrm{D}_{\mathrm{KL}}\left(p \,\|\, \pi_{\mathrm{ref}}(\cdot \mid s)\right). \tag{15}$$

We define the λ-extrapolated target distribution as:

$$\widetilde{\pi}_\lambda(a \mid s) = \frac{\pi_T(a \mid s)^\lambda \pi_{\mathrm{ref}}(a \mid s)^{1-\lambda}}{Z_\lambda(s)}, \quad \text{where} \quad Z_\lambda(s) = \sum_{v\in\mathcal{V}} \pi_T(v \mid s)^\lambda \pi_{\mathrm{ref}}(v \mid s)^{1-\lambda}. \tag{16}$$

Expanding the KL divergence in equation 15 algebraically yields:

$$\begin{aligned} J_s(p) &= \sum_{a\in\mathcal{V}} p(a \mid s)\left[\log\left(\pi_T(a \mid s)^\lambda \pi_{\mathrm{ref}}(a \mid s)^{1-\lambda}\right) - \log p(a \mid s)\right] \\ &= \sum_{a\in\mathcal{V}} p(a \mid s)\left[\log \widetilde{\pi}_\lambda(a \mid s) + \log Z_\lambda(s) - \log p(a \mid s)\right] \\ &= -\,\mathrm{D}_{\mathrm{KL}}\left(p \,\|\, \widetilde{\pi}_\lambda(\cdot \mid s)\right) + \log Z_\lambda(s). \end{aligned} \tag{17}$$

Thus, $\widetilde{\pi}_\lambda$ is the unique, unconstrained maximizer of $J_s(p)$. When parameterizing p via $\pi_\theta(\cdot \mid s)$, we define the unclipped advantage signal:

$$u_\theta(a, s) = \lambda \log \pi_T(a \mid s) - \log \pi_\theta(a \mid s) + (1-\lambda)\log \pi_{\mathrm{ref}}(a \mid s). \tag{18}$$

Taking the gradient with respect to θ gives:

$$\begin{aligned} \nabla_\theta J_s(\pi_\theta) &= \nabla_\theta \sum_{a\in\mathcal{V}} \pi_\theta(a \mid s)\left[\lambda \log \frac{\pi_T(a \mid s)}{\pi_{\mathrm{ref}}(a \mid s)} - \log \frac{\pi_\theta(a \mid s)}{\pi_{\mathrm{ref}}(a \mid s)}\right] \\ &= \sum_{a\in\mathcal{V}} \nabla_\theta \pi_\theta(a \mid s)\cdot u_\theta(a, s) - \sum_{a\in\mathcal{V}} \pi_\theta(a \mid s)\nabla_\theta \log \pi_\theta(a \mid s) \\ &= \mathbb{E}_{a\sim\pi_\theta}\left[u_\theta(a, s)\nabla_\theta \log \pi_\theta(a \mid s)\right], \end{aligned} \tag{19}$$

where the baseline term vanishes since $\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta] = \sum_a \nabla_\theta \pi_\theta(a \mid s) = \nabla_\theta(1) = 0$. Following the next-token approximation in G-OPD (Yang et al., 2026), evaluating u_θ under the rollout snapshot π_S and applying the threshold $[-c_A, c_A]$ yields the practical advantage A_t in equation 2.

A.2 Bridge Geometry and Monotonicity Proof

Let $p(a) = \pi_S(a \mid s)$ and $t(a) = \pi_T(a \mid s)$ with full support over vocabulary $\mathcal{V}$. Define the log-likelihood ratio $h(a) = \log(t(a)/p(a))$ and the cumulant generating function:

$$\psi(\alpha) = \log \sum_{a\in\mathcal{V}} p(a) e^{\alpha h(a)}. \tag{20}$$

The geometric bridge distribution can be cast as an exponential family:

$$q_\alpha(a) = p(a)\exp\left(\alpha h(a) - \psi(\alpha)\right). \tag{21}$$

Standard exponential family identities guarantee that:

$$\psi'(\alpha) = \mathbb{E}_{q_\alpha}[h], \qquad \psi''(\alpha) = \mathrm{Var}_{q_\alpha}[h]. \tag{22}$$

Evaluating the Kullback–Leibler divergence with respect to the student policy p, we have:

$$\begin{aligned} \mathrm{D}_{\mathrm{KL}}(q_\alpha \,\|\, p) &= \sum_{a\in\mathcal{V}} q_\alpha(a)\log \frac{q_\alpha(a)}{p(a)} \\ &= \sum_{a\in\mathcal{V}} q_\alpha(a)\left[\alpha h(a) - \psi(\alpha)\right] \\ &= \alpha\psi'(\alpha) - \psi(\alpha). \end{aligned} \tag{23}$$

Differentiating this divergence with respect to α yields:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\,\mathrm{D}_{\mathrm{KL}}(q_\alpha \,\|\, p) = \psi'(\alpha) + \alpha\psi''(\alpha) - \psi'(\alpha) = \alpha\,\mathrm{Var}_{q_\alpha}[h] \geq 0. \tag{24}$$

Lemma 1 (Monotonicity of the bridge divergence)**.** *Because $\alpha \geq 0$ and the variance is non-negative, $\mathrm{D}_{\mathrm{KL}}(q_\alpha \,\|\, p)$ is monotonically non-decreasing on $[0, 1]$.*

Conversely, evaluating the divergence with respect to the teacher distribution t gives:

$$\mathrm{D}_{\mathrm{KL}}(q_\alpha \,\|\, t) = \sum_{a\in\mathcal{V}} q_\alpha(a)\big[(\alpha-1)h(a) - \psi(\alpha)\big] = (\alpha-1)\psi'(\alpha) - \psi(\alpha). \tag{25}$$

Differentiating equation 25 with respect to α yields:

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\,\mathrm{D}_{\mathrm{KL}}(q_\alpha \,\|\, t) = (\alpha-1)\psi''(\alpha) = (\alpha-1)\,\mathrm{Var}_{q_\alpha}[h] \leq 0. \tag{26}$$

Because $\alpha \leq 1$ on $[0, 1]$ and the variance is non-negative, $\mathrm{D}_{\mathrm{KL}}(q_\alpha \,\|\, t)$ is monotonically non-increasing. Therefore, increasing α simultaneously maximizes teacher closeness and increases student divergence, proving that greedy selection over sorted $\mathcal{A}_+$ in equation 6 selects the largest admissible coefficient and a teacher-closest distribution within the candidate set.

A.3 Sequence-Level KL Control

Let $b_t(\cdot \mid s_t)$ denote the sampling policy: the selected bridge policy during prefix guidance and $\pi_S(\cdot \mid s_t)$ afterwards. For a fixed prompt x, the autoregressive KL chain rule gives

$$\begin{aligned}
&\mathrm{D}_{\mathrm{KL}}(P_{\mathrm{KB}}(\cdot \mid x) \,\|\, P_S(\cdot \mid x)) \\
&\quad= \mathbb{E}_{y\sim P_{\mathrm{KB}}(\cdot|x)}\left[\sum_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(b_t(\cdot \mid s_t) \,\|\, \pi_S(\cdot \mid s_t))\right] \\
&\quad= \mathbb{E}_{y\sim P_{\mathrm{KB}}(\cdot|x)}\left[\sum_{t=1}^{T_b} d_{\alpha_t^\star}(s_t)\right] \leq C.
\end{aligned} \tag{27}$$

The second equality follows because suffix tokens are sampled from π_S and therefore contribute zero conditional KL. The final inequality follows from equation 6, which bounds the accumulated prefix KL by C on every generated trajectory.

A.4 Behavior Importance Weighting and Variance Control

For any token position $t \in B$ sampled from bridge policy $b_t(a) = q_{\alpha_t^\star}(a \mid s_t)$, the nominal importance weight relative to the rollout student is $\rho_t = \pi_S(y_t \mid s_t)/b_t(y_t)$. Multiplying by the PPO ratio $r_t(\theta) = \pi_\theta(y_t \mid s_t)/\pi_S(y_t \mid s_t)$ gives:

$$r_t(\theta)\,\rho_t = \frac{\pi_\theta(y_t \mid s_t)}{\pi_S(y_t \mid s_t)} \cdot \frac{\pi_S(y_t \mid s_t)}{b_t(y_t)} = \frac{\pi_\theta(y_t \mid s_t)}{b_t(y_t)}. \tag{28}$$

For any integrable test function $f(a)$ on $\mathcal{V}$, change-of-measure gives:

$$\mathbb{E}_{a\sim b_t}\left[\rho_t(a)f(a)\right] = \sum_{a\in\mathcal{V}} b_t(a)\frac{\pi_S(a \mid s_t)}{b_t(a)}f(a) = \sum_{a\in\mathcal{V}} \pi_S(a \mid s_t)f(a) = \mathbb{E}_{a\sim\pi_S}[f(a)]. \tag{29}$$

This identity is exact for the action expectation at a fixed sampled state; it does not correct the state distribution. In KB-OPD, clipping ρ_t to $[w_{\min}, w_{\max}]$ limits the magnitude of the behavior weight. For bounded f, the resulting action-expectation bias satisfies:

$$\left|\mathbb{E}_{b_t}[w_t f] - \mathbb{E}_{\pi_S}[f]\right| \leq \|f\|_\infty\, \mathbb{E}_{b_t}\left[|w_t - \rho_t|\right]. \tag{30}$$

The KL constraint separately controls the conditional sampling deviation, while the cumulative budget controls sequence-level divergence. The clipped action correction and the KL budgets therefore act on distinct aspects of the distribution mismatch.

A.5 Loss Aggregation and Boundary Cases

The segment-weighted objective must account for responses with no bridge prefix or no student suffix. For trajectory i, let $\mathcal{L}_B^{(i)}$ and $\mathcal{L}_S^{(i)}$ be the segment means defined in equation 12, assigning zero to an empty segment. Define the active segment weights as

$$\omega_B^{(i)} = \gamma_B \, \mathbf{1}\{|B_i| > 0\}, \qquad \omega_S^{(i)} = \mathbf{1}\{|S_i| > 0\}. \tag{31}$$

Writing $Z_i = \omega_B^{(i)} + \omega_S^{(i)}$, the complete trajectory loss is

$$\mathcal{L}^{(i)} = \begin{cases} \dfrac{\omega_B^{(i)} \mathcal{L}_B^{(i)} + \omega_S^{(i)} \mathcal{L}_S^{(i)}}{Z_i}, & Z_i > 0, \\ 0, & Z_i = 0. \end{cases} \tag{32}$$

The batch objective is $\mathcal{L} = N^{-1} \sum_{i=1}^{N} \mathcal{L}^{(i)}$, where N is the number of trajectories.

When both segments are present, equation 32 reduces to equation 13. The default $\gamma_B = 0.25$ assigns weights 0.2 and 0.8 to the prefix and suffix mean losses, respectively. If guidance ends before the first token, or sampling occurs after warm-up, only the student segment contributes. If EOS occurs within the bridge, only the bridge segment contributes, provided $\gamma_B > 0$. Setting $\gamma_B = 0$ excludes bridge tokens from the loss; a response containing only bridge tokens then contributes zero loss.

B More Experimental Details

B.1 Dataset Details

Training datasets. For mathematical reasoning, we use the subset of DeepMath-103K (He et al., 2025) with difficulty level at least six, containing approximately 57K examples. For code generation, we use Eurus-RL-Code (Cui et al., 2025), containing approximately 25K examples. Problem statements serve as distillation prompts, and the training signal is computed from teacher, student, and reference log-probabilities. The same domain datasets are used for the corresponding single-teacher and strong-to-weak experiments.

Evaluation benchmarks. We evaluate mathematical reasoning on AIME24 (AI-MO, 2024), AIME25 (OpenCompass, 2025), and the February and November editions of HMMT25 (Balunović et al., 2025). Related benchmarks include MATH, GSM8K, and OlympiadBench. Code generation is evaluated on HumanEval+ (Liu et al., 2023), MBPP+ (Liu et al., 2023), and LiveCodeBench (Jain et al., 2025). We use the LiveCodeBench v6 subset covering February–May 2025.

Math responses are assessed using Math-Verify, while code responses are evaluated with the corresponding benchmark test suites (Liu et al., 2023). We generate 32 responses per math problem and four responses per code problem. For math, average accuracy is estimated by averaging per-response correctness over the 32 samples within each problem and then across problems; this is not pass@32 (at least one correct out of 32). For code, pass@1 is estimated by averaging correctness over the four samples. Domain averages assign equal weight to each benchmark.

B.2 Training Details

Models and optimization. Same-size distillation uses Qwen3-4B-Non-Thinking (Yang et al., 2025) as the initial student and its domain-specific RL variants as teachers. Strong-to-weak distillation uses Qwen3-30B-A3B-Instruct-2507 as the teacher and Qwen3-4B-Non-Thinking or Qwen3-1.7B-Non-Thinking as the student. The reference policy is the frozen initial student in each setting. Training uses the AdamW optimizer (Loshchilov & Hutter, 2019) with the dense advantage and PPO surrogate defined in Section 3. We train for 50 steps and save checkpoints every five steps. The reported checkpoint is selected by the highest mean score on the corresponding evaluation suite.

Sampling and supervision. Each prompt produces one response. During the first five training steps, KB-OPD samples a bridge prefix and lets the student generate the remaining tokens. After warm-up, the student generates the entire response. Teacher and reference scoring continue

Table 5: Training and generation settings.

Parameter	Value
Training steps	50
Global batch size	1024
Responses per prompt	1
Learning rate	10^{-5}
Checkpoint interval	5 steps
Temperature	1.0
Top-p	1.0
Max. prompt length	1024
Max. response length	16,384

Table 6: Dense supervision and policy updates.

Parameter	Value
Reference π_{ref}	Frozen initial student
Reward scaling λ	1.25
Advantage clipping	$[-2, 2]$
PPO clip ϵ	0.2

Table 7: Bridge sampling and weighting.

Parameter	Value
Warm-up w	5 steps
Candidate set $\mathcal{A}_+$	$\{0.1, 0.2, \ldots, 0.5\}$
Minimum $\alpha_{\min}$	0.1
Maximum $\alpha_{\max}$	0.5
Local KL η	0.05
Cumulative KL C	1.5
Prefix cap $L_{\max}$	512
Bridge weight γ_B	0.25
Behavior-weight clipping	$[0.5, 2.0]$

Table 8: Evaluation settings.

Parameter	Value
Responses per math problem	32
Responses per code problem	4
Temperature	1.0
Top-p	1.0

throughout training; the reference term is omitted when $\lambda = 1$. Rollout probabilities, advantages, and behavior weights are detached, with gradients flowing only through the optimized student policy. Tables 5–8 summarize the training, supervision, bridge, and evaluation settings. All length limits are measured in tokens. The zero mixing coefficient corresponds to student fallback and is excluded from the positive bridge candidate set.

C KB-OPD Pseudocode

Algorithm 1 summarizes KB-OPD training. During warm-up, a bridge policy generates the initial prefix under local and cumulative KL constraints. The student completes the response, and both segments contribute to the policy update. After warm-up, sampling returns entirely to the student.

Student and behavior log-probabilities are stored during sampling. For each response, B denotes bridge positions and S denotes student-generated positions. Generation always respects the maximum response length $T_{\max}$, and EOS terminates the response.

Segment losses are normalized separately before aggregation. Empty segments are omitted and the remaining weights are renormalized. A trajectory with no positive-weight segment contributes zero loss. After warm-up, B is empty and S contains the entire response, reducing the update to the student token-averaged loss.

All rollout probabilities, advantages, and behavior weights are detached. Teacher scores are evaluated throughout training, while reference scores are needed only when $\lambda \neq 1$.

D Prompt Templates

We use the same task-specific prompts as G-OPD (Yang et al., 2026) for training and evaluation. The templates below show the user message and the assistant generation prefix, where `{question}` denotes the problem statement.

Algorithm 1 KL-Bridge On-Policy Distillation

Require: Student π_θ, teacher π_T, reference π_{ref}
Require: Training steps G, warm-up w, response limit $T_{\max}$
Require: Candidates $\mathcal{A}_+$, budgets η, C, prefix cap $L_{\max}$

for $g = 1, \ldots, G$ **do**
  Freeze rollout student $\pi_S \leftarrow \pi_\theta$; sample prompt batch $\mathcal{X}$
  for each $x \in \mathcal{X}$ **do**
    Initialize $y \leftarrow ()$, $B \leftarrow \varnothing$, and $c \leftarrow 0$
    if $g \le w$ **then**
      for $t = 1, \ldots, \min(L_{\max}, T_{\max})$ **do**
        Set $s_t = (x, y_{<t})$; compute q_α and $d_\alpha(s_t)$
        $\mathcal{F} \leftarrow \{\alpha \in \mathcal{A}_+ : d_\alpha(s_t) \le \min(\eta, C - c)\}$
        if $\mathcal{F} = \varnothing$ **then**
          break
        end if
        $\alpha^\star \leftarrow \max \mathcal{F}$; sample $y_t \sim q_{\alpha^\star}(\cdot \mid s_t)$
        Append y_t; $B \leftarrow B \cup \{t\}$; $c \leftarrow c + d_{\alpha^\star}(s_t)$
        if $y_t = \text{EOS}$ **then**
          break
        end if
      end for
    end if
    Complete any unfinished response with π_S until EOS or $T_{\max}$
    $S \leftarrow \{1, \ldots, |y|\} \setminus B$
    Compute detached A_t and w_t using equation 2 and equation 9
  end for
  Compute PPO token losses using equation 11
  Average losses within segments and combine using weights γ_B and 1
  Average trajectory losses over the batch and update θ
end for

Math Reasoning: Training and Evaluation

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

Code Generation: Training and Evaluation

```
<|im_start|>user
{question}
Write Python code to solve the problem. Present the code in
```python
Your code
```
at the end.
You need to think first then write the Python code.<|im_end|>
<|im_start|>assistant
```