REVISITING ON-POLICY SELF-DISTILLATION FROM AN RL PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) gives a long reasoning trace only a sparse outcome reward. On-Policy Self-Distillation densifies this signal by conditioning the model on one of its own successful rollouts and distilling the resulting self-teacher into the unprompted policy, but its gains on thinking models are fragile. In this paper, we show that, even with an ideal teacher, SDPO performs a one-step behavior-regularized improvement on the value of the behavior policy, rather than maximizing the reward of the policy being improved. To correct this, we derive the exact optimum of behavior-regularized reward maximization and show that, for binary reward, it is a *tilted mixture*, i.e., an arithmetic mixture of the behavior policy and its success posterior. Based on this result, we propose Tilted-SDPO (T-SDPO), which distills this mixture using SDPO's prompted teacher in place of the posterior and reduces to SDPO when the mixture weight is one. Across math and science reasoning with three thinking models, T-SDPO outperforms SDPO and GRPO at an equal number of optimizer updates, improving math avg@12 over SDPO by up to 13.0 points. It also keeps longer responses and higher or tied pass@12 than SDPO.

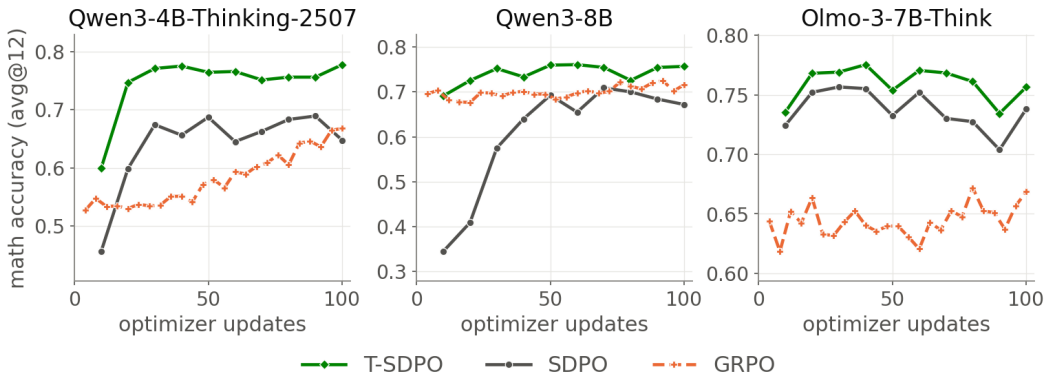


Figure 1: T-SDPO outperforms baselines on different tasks across three different (thinking) models.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become the standard recipe for post-training reasoning models (Shao et al., 2024; Guo et al., 2025). Its learning signal is sparse: the model generates thousands of tokens but receives a single scalar outcome reward, applied uniformly to every token. This creates a credit-assignment bottleneck (Hübotter et al., 2026; Zhao et al., 2026). On-policy self-distillation (OPSD) densifies this signal without an external teacher (Zhao et al., 2026; Shenfeld et al., 2026). The model, re-prompted with privileged information such as a verified solution, acts as a self-teacher, and every token of the policy's own rollout is distilled toward it. Self-Distillation Policy Optimization (SDPO) brings this idea to RLVR by using a successful rollout from the current group as the privileged context (Hübotter et al., 2026). Recent studies, however, find that these gains are fragile on reasoning. On thinking models, self-distillation shortens reasoning and

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.

suppresses the expression of uncertainty (Kim et al., 2026; Peng et al., 2026; Kaur et al., 2026), reduces diversity (Nicolicioiu et al., 2026), and degrades under prolonged training while GRPO keeps improving (Li et al., 2026a; Yang et al., 2026). Existing remedies either route part of the update back to the verifier's reward or modify the distillation loss around the teacher, which leaves a more basic question open:

Which objective does on-policy self-distillation optimize, and is it the right one?

We show that SDPO does not maximize the reward, even with an ideal teacher, . Distilling a teacher that conditions on success rollouts is equivalent to a behavior-regularized improvement step on Q^{μ}. Here μ is the behavior policy, i.e., the exponential-moving-average policy from which the teacher is built (Section 4.1). SDPO therefore optimizes a value estimated under μ, not the reward of the policy being improved, and this biases its update. The policy improvement is bounded by what μ's own continuations achieve. Moreover, because μ lags behind the current policy, each update improves on a past policy rather than the current one. In the extreme case of a frozen teacher, μ is the initial policy, and every update is a one-step improvement over the base model in the sense of one-step offline RL (Brandfonbrener et al., 2021). These biases are consistent with the failure modes reported above.

Based on this finding, we instead derive the distillation target from first principles. Starting from behavior-regularized reward maximization, we show that for binary reward its optimum is a *tilted mixture*, i.e., an arithmetic mixture of the behavior policy and the success posterior. Our method, Tilted-SDPO (T-SDPO), distills this mixture, using SDPO's prompted teacher in place of the posterior. T-SDPO thus differs from SDPO only in the distillation target, and it reduces to SDPO when the mixture weight λ equals 1. The tilted target also differs from adding a KL penalty toward μ, which yields a geometric mixture instead (Xu et al., 2026). The arithmetic form keeps the support of both components and stays within μ at every state. A toy example illustrates why this matters in practice: when the teacher answers better but reasons for fewer steps than μ, the success rate peaks at an interior λ, above both μ and SDPO (Figure 2). We evaluate T-SDPO on math and science reasoning with three thinking models, Qwen3-4B-Thinking-2507, Qwen3-8B and Olmo-3-7B-Think. Under the same environment updates, T-SDPO outperforms GRPO and SDPO in all 12 math and science settings. On math, it improves avg@12 over SDPO by 7.8 points on average across models and by up to 13.0 points on Qwen3-4B-Thinking-2507 (Tables 1 and 2).

2 Related work

Reinforcement learning with verified reward. In RLVR, the policy samples responses to prompts with verifiable answers and is updated by policy gradient on the verifier's outcome (Guo et al., 2025). GRPO replaces the learned critic of PPO with advantages normalized within a group of rollouts (Schulman et al., 2017; Shao et al., 2024), and DAPO further adds decoupled clipping and dynamic sampling to stabilize training at scale (Yu et al., 2025). In all of these methods, however, the reward is a single bit per response. Every token in a response receives the same advantage, so credit assignment within a long reasoning trace is left to sampling. Moreover, a group whose rollouts are all correct or all wrong yields no gradient (Yu et al., 2025; Zhao et al., 2026). This sparsity motivates dense, token-level supervision from a teacher, which is the line of work we build on.

On-policy self-distillation. On-policy distillation trains a student on its own sampled trajectories while a teacher provides dense, token-level supervision on those trajectories (Agarwal et al., 2024). Off-policy distillation instead imitates teacher-generated text, so the prefixes seen in training differ from those the student generates at inference. On-policy distillation removes this mismatch, which is the source of compounding error in imitation learningn (Ross et al., 2011; Song & Zheng, 2026). On-policy self-distillation (OPSD) removes the separate teacher as well by conditioning the same model on privileged context (Snell et al., 2022): a system prompt (Wu et al., 2026), a demonstration (Shenfeld et al., 2026), a reference solution (Zhao et al., 2026; Penaloza et al., 2026), or environment feedback and successful rollouts in RLVR (Hübotter et al., 2026). However, its gains on reasoning are fragile. Conditioning the teacher on the answer suppresses the model's expression of uncertainty and shortens its reasoning (Kim et al., 2026; Peng et al., 2026), degrades thinking models at long rollout budgets (Kaur et al., 2026), biases the target toward the particular reference solution (Harne et al., 2026), reduces diversity (Nicolicioiu et al., 2026), and causes forgetting (Wang et al., 2026);

on thinking models it acts more as compression than as correction (He et al., 2026; Lee et al., 2026). Our analysis offers one explanation: distilling the teacher alone takes the largest step the regularized objective allows.

Combining self-distillation with RL. A second line keeps the verifier's reward in the loop. Hybrid methods route correct and failed samples to GRPO and SDPO respectively (Li et al., 2026a), use self-distillation to set update magnitudes while RLVR sets their directions (Yang et al., 2026), add it to the policy gradient as an auxiliary loss (Liu et al., 2026), or apply it only on prompts where every rollout fails (Ding, 2026). Other methods reverse its sign (Shen et al., 2026a) or refine the teacher by variational EM (Li et al., 2026b). Closer analyses show that the self-distillation token score measures the mutual information between response and feedback rather than an action value (Shen et al., 2026b; Nguyen et al., 2026). These methods are designed around an observed failure mode and modify the loss around a fixed teacher target, T-SDPO is instead derived from first principles: we start from the behavior-regularized RL objective on the verifier's reward and take its closed-form optimum as the distillation target.

Self-distillation as regularized policy optimization. Several works relate the self-distillation target to a KL-regularized objective. β-OPSD (Xu et al., 2026) shows that vanilla OPSD is the $\beta = 1$ member of a family that adds a KL penalty toward a reference policy, and its optimum is a *geometric* interpolation of the reference and the teacher, implemented by mixing logits. HDPO (Ding, 2026) proves that success-filtered privileged generation recovers the KL-regularized optimum in the hard-threshold limit; PBSD (Yu et al., 2026) derives a reward-regularized optimum that reweights the teacher; and the power distribution is the KL-regularized optimum under a self-reward (Tomihari & Sato, 2026). Our starting point is different: we regularize the RL objective on the verifier's reward toward the behavior policy, rather than the distillation objective toward the teacher. For binary reward its optimum is an *arithmetic* mixture of the behavior policy and its success posterior, which keeps the mass of both components and bounds the deviation from μ at every state (Section 4.3), and it contains the success-posterior target of SDPO and filtered generation as its zero-temperature limit. Appendix A covers outcome-conditioned learning and KL-regularized RL.

3 Preliminaries: On-Policy Self-Distillation

Compared with on-policy RL such as GRPO (Shao et al., 2024), OPSD differs in what each sampled response is trained against. GRPO applies one scalar reward uniformly to every token of a response, and its group-normalized advantage vanishes when all responses are correct or all are wrong (Zhao et al., 2026). OPSD instead gives every token its own target distribution, which provides a dense learning signal whenever the privileged context is available, and it needs neither a reward model nor an external teacher (Zhao et al., 2026; Hübotter et al., 2026). Self-Distillation Policy Optimization (SDPO) (Hübotter et al., 2026) brings this to RLVR: its privileged context is the environment's feedback on the policy's own rollouts, and, in binary-reward environments, a successful rollout from the same group. We review its components below and introduce the notation used throughout the paper.

On-policy interaction. Let x be a prompt and $\pi = \pi_\theta$ the current policy. SDPO samples a group of responses $y = (y_1, \ldots, y_T) \sim \pi(\cdot \mid x)$, and an environment evaluates each y and returns feedback f. We write $s_t = (x, y_{<t})$ for the unaugmented token state.

Self-teacher. SDPO does not evaluate its teacher with the current parameters θ. It keeps a separate copy $\bar{\theta}$, initialized to the initial policy parameters, that tracks θ through an exponential moving average (EMA),

$$\bar{\theta} \leftarrow (1 - \eta)\,\bar{\theta} + \eta\,\theta, \qquad \mu := \pi_{\bar{\theta}}, \tag{1}$$

where $\eta \in [0, 1]$ is the teacher update rate. We call μ the *behavior policy*: it is the policy the teacher is built from and, in Section 4, the policy that behavior regularization anchors to. The teacher is μ evaluated under a feedback-augmented input,

$$q_f(a \mid s_t) := \mu(a \mid x \oplus f,\, y_{<t})\,, \tag{2}$$

where $x \oplus f$ denotes the serialized SDPO input obtained by appending the feedback to the original prompt. Both the student and teacher are conditioned on the same original response prefix $y_{<t}$,

allowing their next-token distributions to be compared at aligned token positions. In the binary-reward setting studied here, $f = y^\star$ and

$$q_{y^\star}(a \mid s_t) := \mu(a \mid x \oplus y^\star,\, y_{<t})\,. \tag{3}$$

The SDPO template appends `Correct solution:` and the successful rollout before the original problem (Hübotter et al., 2026), so $q_{y^\star}$ is simply the same network given a longer prompt, with $y^\star$ entering only as context tokens.

Note that μ coincides with the current policy π only when $\eta = 1$: at $\eta = 0$ it is the initial policy (i.e., the base model's distribution), and for $0 < \eta < 1$ a lagged average of past policies. The teacher therefore conditions a stale version of the policy on a $y^\star$ that π generated.

Distillation objective. For a frozen on-policy batch sampled from π and detached teacher logits, the core SDPO objective is

$$\mathcal{L}_{\mathrm{SDPO}}(\pi;\mu) = \mathbb{E}_{\substack{y\sim\pi(\cdot|x)\\ f\sim P_{\mathrm{env}}(\cdot|x,y)}}\left[\sum_{t=1}^{T} D_{\mathrm{KL}}(\pi(\cdot \mid s_t) \,\|\, \mathrm{stopgrad}\, q_f(\cdot \mid s_t))\right]. \tag{4}$$

We focus on the reverse-KL form of Eq. (4) in the rest of the analysis.

4 Revisiting Self-Distillation Policy Optimization

In this section, we provide a detailed revisit to SDPO and introduce our proposed method, Tilted-SDPO (T-SDPO), to address its limitations. We begin by showing that, with an ideal teacher, SDPO performs a one-step behavior-regularized improvement (Section 4.1). We then derive the principled target that behavior-regularized reward maximization yields, show that it is a tilted mixture of the behavior policy and the success posterior, and illustrate its benefit with a toy example (Section 4.2). Finally, we turn this target into a practical algorithm (Section 4.3).

4.1 Connecting SDPO to Reinforcement Learning

In binary-reward settings that give no rich feedback, SDPO uses a verified response to construct a self-teacher that guides policy updates. This use of successful examples is reminiscent of example-based RL, where an agent learns to achieve desired outcomes specified through examples (Fu et al., 2018; Eysenbach et al., 2021). In our setting, these examples are successful rollouts of the current policy, identified by a binary verifier. Conditioning these rollouts on success induces a natural policy target: the distribution of next actions at a given prefix among trajectories that eventually succeed.

Formally, for a token state $s = (x, y_{<t})$ reachable under μ with positive probability of eventual success, define

$$q_+(a \mid s) := P_\mu(A = a \mid S = s, R = 1). \tag{5}$$

To relate this target to reinforcement learning, define

$$Q^\mu(s,a) := P_\mu(R = 1 \mid s, a), \qquad V^\mu(s) := P_\mu(R = 1 \mid s), \tag{6}$$

where subsequent actions are sampled from μ. By Bayes' rule,

$$q_+(a \mid s) = \mu(a \mid s)\frac{Q^\mu(s,a)}{V^\mu(s)}. \tag{7}$$

Thus, the action distribution induced by examples (i.e., successful rollouts) is a value-reweighted version of the behavior policy: actions become more likely in proportion to their probability of eventual success under continuation with μ. This identity provides a reference target for interpreting SDPO's prompted self-teacher and the policy update obtained by distilling it.

Given a successful response $y^\star$, SDPO constructs a prompted self-teacher $q_{y^\star}(\cdot \mid s)$. To characterize the policy update induced by this teacher, we introduce the following idealization.

Assumption 1 (Success-posterior teacher approximation)**.** For every state s included in the distillation objective and each successful example $y^\star$ used to construct the teacher at that state,

$$q_{y^\star}(\cdot \mid s) \approx q_+(\cdot \mid s). \tag{8}$$

Assumption 1 idealizes in-context conditioning on a successful example as conditioning on the event of eventual success (Xie et al., 2022). We use it to isolate the policy-improvement objective induced by an ideal posterior teacher from errors in the prompted approximation. Under this assumption we can derive the policy-improvement objective that SDPO implicitly optimizes, which makes its connection to RL explicit.

The implicit policy-improvement objective. Consider reverse-KL distillation at a fixed state, holding μ and the teacher fixed during the update. Suppressing the dependence on $y^\star$, we have

$$D_{\mathrm{KL}}(\pi \| q_{y^\star}) = D_{\mathrm{KL}}(\pi \| \mu) - \mathbb{E}_{a\sim\pi}\left[\log \frac{q_{y^\star}(a \mid s)}{\mu(a \mid s)}\right]. \tag{9}$$

Under Assumption 1, minimizing Eq. (9) over π is therefore equivalent to

$$\max_{\pi}\ \mathbb{E}_{a\sim\pi(\cdot|s)}[\log Q^\mu(s,a)] - D_{\mathrm{KL}}(\pi(\cdot \mid s)\|\mu(\cdot \mid s))\,. \tag{10}$$

This follows by substituting Eq. (7), which gives $\log(q_+/\mu) = \log Q^\mu - \log V^\mu$, and noting that $V^\mu(s)$ does not depend on π. Eq. (10) depends on Q^μ, not Q^π. Thus, the objective evaluates a candidate action assuming that future decisions are made by a different policy (i.e., the behavior policy), μ and π are different since in practice η needs to be small to make learning stable. This distinction limits what the objective establishes. An action whose benefit depends on coordinated changes to subsequent decisions may have low value under μ but high value under an improved continuation policy. Consequently, even an exact success-posterior teacher does not provide a correct value evaluation of the evolving student policy.

Connection to one-step RL. This structure mirrors a distinction in offline RL between one-step and multi-step methods. One-step RL methods evaluate the behavior policy β in-sample using off-policy data and take a single regularized improvement step against Q^β (Peng et al., 2019; Brandfonbrener et al., 2021), whereas multi-step methods evaluate the improving policy using Q^π that learned by off-policy temporal-difference learning (Fujimoto et al., 2019; Kumar et al., 2020). One-step methods are appealing because they never query out-of-distribution actions (Brandfonbrener et al., 2021), but their improvement is capped at one step over β, which fails when a good policy must "stitch" behaviors (Kostrikov et al., 2022; Xu et al., 2023). Note that here one-step refers to policy improvement, not to the task horizon.

4.2 Optimizing the Correct RL Objective with Tilted-SDPO

The surrogate of Eq. (10) optimizes Q^μ rather than Q^π (which is doing reward maximization). To correct from the first principle, we derive the distillation target from an explicit behavior-regularized reward maximization objective. At a state s, consider

$$\pi^\star_\beta = \arg\max_{\pi} \left\{\mathbb{E}_{y\sim\pi(\cdot|x)}[R(x,y)] - \beta D_{\mathrm{KL}}(\pi(\cdot \mid x)\|\mu(\cdot \mid x))\right\}, \qquad \beta > 0. \tag{11}$$

The coefficient β controls how strongly the new policy remains anchored to the behavior policy μ.

Theorem 2 (Binary-reward behavior-regularized target)**.** *The solution to Eq. (11) is*

$$\pi^\star_\beta(y \mid x) = \frac{\mu(y \mid x)e^{R(x,y)/\beta}}{1 + (e^{1/\beta} - 1)V^\mu(x)}. \tag{12}$$

For binary R, it can equivalently be written as

$$\pi^\star_\beta = (1 - \lambda_\beta)\mu + \lambda_\beta q_+, \qquad \lambda_\beta(x) = \frac{(e^{1/\beta} - 1)V^\mu(x)}{1 + (e^{1/\beta} - 1)V^\mu(x)}. \tag{13}$$

Theorem 2 follows from the standard exponential tilt $\pi^\star_\beta \propto \mu e^{R/\beta}$ (Levine, 2018; Geist et al., 2019); Appendix B gives the derivation. Theorem 2 identifies the distillation target: a *tilted mixture* of the behavior policy and the success posterior. Directly distilling q_+ is recovered only as $\beta \to 0$, which however loses the behavior regularization; for finite β, it controls the intended strength of the behavior regularization.

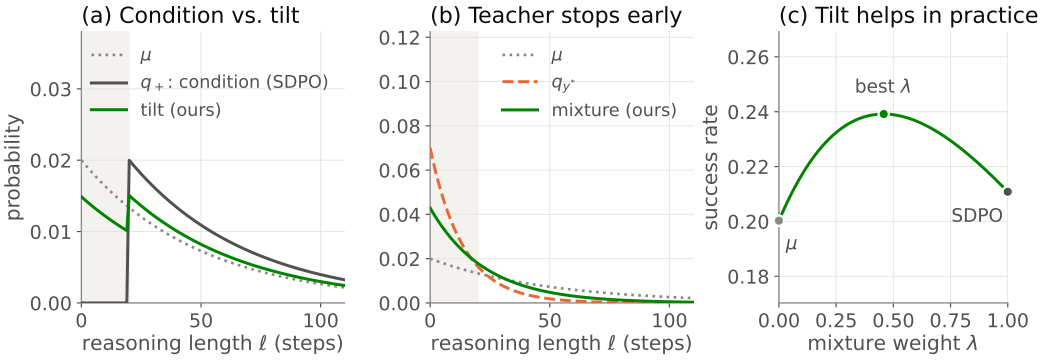


Figure 2: **A toy example of conditioning versus tilting.** Responses reason for ℓ steps and succeed only if $\ell \geq L = 20$ (shaded: always fails) and the final answer is correct. (a) SDPO's ideal target q_+ conditions μ on success, while the behavior-regularized optimum tilts it. (b) The prompted teacher answers better than μ but stops reasoning earlier. (c) The token-level mixture of μ and the teacher peaks at an interior λ, above both μ and SDPO.

A toy example. To provide more intuition about why the tilted mixture could be better than distilling the teacher alone, we consider a simple toy example. Assuming each response reasons for ℓ steps, stopping at each step with probability p, and then answers correctly with probability a; it succeeds only if $\ell \geq L = 20$ and the answer is correct. The behavior policy reasons long but answers weakly ($p_\mu = 0.02$, $a_\mu = 0.3$), so $V^\mu = 0.2$. As shown in Figure 2(a)(b), the (ideal) success posterior q_+ assigns zero probability to every response shorter than L. We also make a biased proxy of q_+, $q_{y^\star}$ that answers better ($a_q = 0.9$) but stops much earlier ($p_q = 0.07$, i.e., 14 steps on average against 50 for μ). The benefit of the tilt is shown in Figure 2(c). Mixing the two at token level makes both rates linear in λ, but the stopping term compounds over the L steps. As a result, success peaks at $\lambda = 0.46$ (0.239), above both μ (0.200) and SDPO (0.211). Intuitively, the mixture keeps the reasoning length whereas distilling the teacher alone could not, which is exactly the behavior we measure in Section 5.3.

4.3 Practical Considerations

The exact behavior-regularized RL target requires both the success posterior q_+ and the behavior value $V^\mu(x)$, neither of which is directly available. T-SDPO replaces the first with the prompted self-teacher $q_{y^\star}$, which the SDPO pipeline already does. For the second, although we could estimate via rolling out trajectories from the behavior policy, this is computationally expensive and introduces additional variance. Instead, in practice we use a single scalar hyperparameter $\lambda \in [0, 1]$ as a constant weight to represent the mixture weight λ_β, and distill the token-level mixture

$$\widehat{\pi}_\lambda^{y^\star}(a \mid s_t) := (1 - \lambda)\, \mu(a \mid s_t) + \lambda\, q_{y^\star}(a \mid s_t). \tag{14}$$

On-policy training objective. Following SDPO, we minimize a frozen-batch, token-level on-policy divergence to this target:

$$\mathcal{L}_{\text{T-SDPO}}(\theta; \mu) = \mathbb{E}_{\substack{x \sim \mathcal{D},\ y \sim \pi(\cdot|x) \\ y^\star \sim \mathcal{S}(x)}} \left[\sum_{t=1}^{T} D_{\mathrm{KL}}\Big(\pi_\theta(\cdot \mid s_t) \,\Big\|\, \operatorname{stopgrad} \widehat{\pi}_\lambda^{y^\star}(\cdot \mid s_t) \Big) \right], \tag{15}$$

where $\mathcal{S}(x)$ is the distribution used to select a successful response from the current rollout group, and $\widehat{\pi}_\lambda$ is given by Eq. (14). We instantiate $\mathcal{S}(x)$ as a uniform draw over the group members whose sequence reward clears the verifier threshold; because rollouts within a group are exchangeable, we take the first success in group order to realize this. Prompts whose group contains no success contribute no distillation term. At $\lambda = 1$ the method is exactly SDPO, and at $\lambda = 0$ its loss is zero from the first step, so the policy cannot move. Within an update both μ and $q_{y^\star}$ are detached, so the target is a constant; Appendix C describes how the mixture is formed under SDPO's top-k truncation and its cost, which is one extra no-gradient forward pass.

Why not simply add a KL penalty? The most natural way to pull a policy toward a reference is to add $D_{\mathrm{KL}}(\pi_\theta \| \mu)$ to the loss, but that is not what Theorem 2 asks for. Penalizing divergence from μ while distilling $q_{y^\star}$ optimizes a weighted sum of two reverse KLs, which is minimized by the normalized *geometric* mixture, i.e., a weighted average of the two log-probabilities,

$$\arg\min_{\pi} \left\{(1-\lambda) D_{\mathrm{KL}}(\pi \| \mu) + \lambda D_{\mathrm{KL}}(\pi \| q_{y^\star})\right\} \;\propto\; \mu^{1-\lambda}\, q_{y^\star}^{\lambda}, \tag{16}$$

whereas Eq. (13) is an *arithmetic* mixture, a weighted average of the two probabilities themselves. A geometric mixture is supported on the intersection of its components, while an arithmetic one on their union. Note that the arithmetic form bounds the deviation from the behavior policy at every state, $\mathrm{TV}(\widehat{\pi}_\lambda, \mu) = \lambda\, \mathrm{TV}(q_{y^\star}, \mu) \le \lambda$, whereas the geometric form removes every token the teacher rules out for any $\lambda > 0$; and only the arithmetic form is the principled target distribution given Theorem 2. T-SDPO therefore builds the mixture in probability space and distills it, instead of adding a penalty. Note that Eq. (15) is deliberately *not* equal to $(1-\lambda) D_{\mathrm{KL}}(\pi_\theta \| \mu) + \lambda D_{\mathrm{KL}}(\pi_\theta \| q_{y^\star})$.

5 Experiments

In this section, we evaluate T-SDPO against SDPO and GRPO on three thinking models from two families and two RLVR domains. We first report the main results (Section 5.2), then analyze what the mixture changes in the policy (Section 5.3), and finally ablate the mixture weight and its state-dependent variant (Section 5.4). Supporting grids and protocol details are in the appendices.

5.1 Experimental setting

Models and tasks. We choose Qwen3-4B-Thinking-2507, Qwen3-8B (Yang et al., 2025) (thinking mode on) and Olmo-3-7B-Think (Ai2, 2025) models to check the generality of our approach. We consider two RLVR domains. Math trains on the DAPO dataset (Yu et al., 2025) and is evaluated on AIME24 and AMC23; Science trains on the biology and chemistry subsets of SciKnowEval (Feng et al., 2024), each evaluated on its own held-out test split. See Appendix for more details.

Baselines. We compare T-SDPO with SDPO (Hübotter et al., 2026) and GRPO (Shao et al., 2024) with the same models, prompts, budgets and evaluation decoding. For SDPO and T-SDPO, we sweep the teacher rate $\eta \in \{0, 0.01, 0.05\}$ where $\eta = 0$ freezes the teacher at initialization, for T-SDPO, we also sweep the mixture weight $\rho \in \{0.25, 0.50, 0.75, 0.90\}$. Every number is avg@12 under eval config (temperature 0.6, top-p 0.95, top-k 20), measured post hoc on saved checkpoints. Appendix D lists the remaining hyperparameters.

5.2 Main results

T-SDPO outperforms the baselines on almost all tasks. Tables 1 and 2 report avg@12 at 100 optimizer updates. T-SDPO outperforms SDPO and GRPO in all 12 cells, and the gain is largest on math, especially on the two Qwen models, where it raises the average avg@12 over SDPO by 13.0 and 8.5 points. On Olmo-3 the math gain is smaller ($+1.8$), and on science the gains range from 0.9 to 4.0 points. This suggests that the benefit of the tilt grows with the length of the reasoning it protects, since a short science answer leaves little for the teacher to shorten (Section 5.3). Figure 1 shows the same result as a trajectory: T-SDPO stays above SDPO and GRPO at every checkpoint on all three models.

5.3 Analysis of T-SDPO

Does the mixture preserve response diversity? Yes. Prior work finds that on-policy self-distillation shortens reasoning and reduces diversity, suppressing the tokens that express uncertainty (Kim et al., 2026; Peng et al., 2026; Kaur et al., 2026) and concentrating rollouts on dominant modes so that pass@k stops improving with k (Nicolicioiu et al., 2026). T-SDPO avoids both. First, it keeps longer responses than SDPO in all six panels of Figure 3, e.g., 6858 vs. 3855 tokens on 8B math, and avoids the early collapse of SDPO. Second, it achieves the highest or tied-highest pass@12 on all six math cells (Table 6), so its gain in avg@12 does not come at the cost of coverage. Intuitively, the arithmetic mixture is supported on the union of μ and the teacher, so T-SDPO retains the behavior policy's own coverage, whereas distilling the teacher alone removes whatever the teacher rules out.

Table 1: **Math results**. Numbers in parentheses are gains over SDPO, and we bold the best entry of each column within a model. T-SDPO outperforms SDPO and GRPO on all six math cells, by a large margin on both Qwen models.

Method	AIME 2024 ↑	AMC 2023 ↑	Average ↑
Qwen3-4B-Thinking-2507			
Base	37.78	72.71	55.24
SDPO	44.72	84.79	64.76
GRPO	48.61	85.00	66.81
T-SDPO (ours)	**60.56** (+15.83)	**95.00** (+10.21)	**77.78** (+13.02)
Qwen3-8B			
Base	54.44	85.21	69.83
SDPO	50.00	84.38	67.19
GRPO	55.56	87.71	71.63
T-SDPO (ours)	**62.22** (+12.22)	**89.17** (+4.79)	**75.69** (+8.51)
Olmo-3-7B-Think			
Base	43.61	74.58	59.10
SDPO	58.89	88.75	73.82
GRPO	52.50	81.25	66.88
T-SDPO (ours)	**60.28** (+1.39)	**91.04** (+2.29)	**75.66** (+1.84)

Table 2: **Science results** on SciKnowEval. Numbers in parentheses are gains over SDPO, and we bold the best entry of each column within a model. T-SDPO outperforms SDPO and GRPO on all six science cells.

Method	Biology ↑	Chemistry ↑	Average ↑
Qwen3-4B-Thinking-2507			
Base	11.25	22.38	16.82
SDPO	44.20	75.95	60.08
GRPO	29.75	52.54	41.15
T-SDPO (ours)	**45.83** (+1.63)	**76.83** (+0.87)	**61.33** (+1.25)
Qwen3-8B			
Base	26.61	47.58	37.10
SDPO	48.83	77.30	63.07
GRPO	30.92	54.60	42.76
T-SDPO (ours)	**52.83** (+4.00)	**78.45** (+1.15)	**65.64** (+2.57)
Olmo-3-7B-Think			
Base	9.50	21.11	15.31
SDPO	47.50	74.84	61.17
GRPO	24.83	34.72	29.78
T-SDPO (ours)	**49.00** (+1.50)	**75.95** (+1.11)	**62.48** (+1.31)

5.4 Ablation of T-SDPO

How does the mixture weight affect performance? Figure 4 sweeps the hyperparameter λ T-SDPO adds. On math the curve rises, peaks strictly inside the interval, and falls again toward plain SDPO: 0.592 at $\lambda = 0.75$ against 0.502 at $\lambda = 1$ on 4B, and 0.597 at $\lambda = 0.50$ against 0.528 on 8B.

How does the teacher update rate affect performance? The teacher rate η controls how fast the EMA teacher tracks the policy, and Figure 5 shows that it affects the two domains in opposite directions. On math, every distillation arm degrades as η grows, but SDPO degrades much faster: at $\eta = 0.05$ it falls to 0.042 on 4B and 0.173 on 8B, while the best T-SDPO arm still holds 0.507 and 0.481. However, on science a moderately drifting teacher helps, and at $\eta = 0.05$ T-SDPO outperforms SDPO on both models while SDPO collapses to 0.290 on 8B. In both domains, the mass retained on μ is what slows the collapse under drift, which is exactly the role of the behavior anchor in Theorem 2.



Figure 3: **Response length analysis**, T-SDPO keeps longer responses than SDPO in all six panels and avoids SDPO's early collapse.

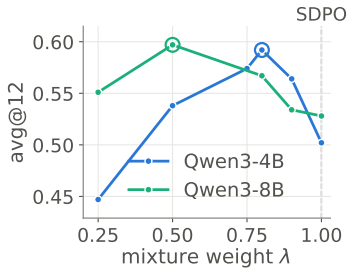


Figure 4: **Effect of** λ on math. The optimum is interior on both models.

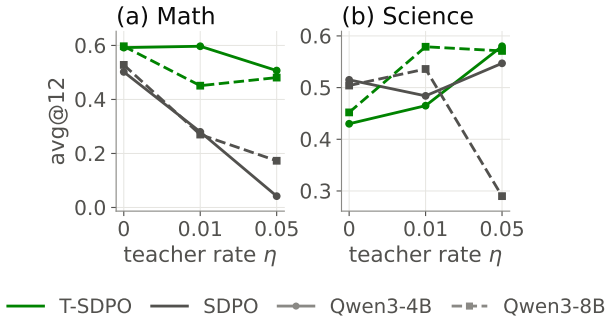


Figure 5: **Effect of** η: SDPO collapses faster under a drifting teacher.

Does T-SDPO outperform SDPO with a KL penalty? A natural alternative keeps the SDPO target and adds a penalty $D_{\mathrm{KL}}(\pi_\theta \| \mu)$ to the original distillation loss, which yields a geometric rather than an arithmetic mixture (Section 4.3). We sweep the same range as $\lambda \in \{0.5, 0.75, 0.9\}$. As shown in Table 3, a small penalty does help SDPO, but T-SDPO outperforms its best setting on both Qwen models by 8 and 5 points. On biology, T-SDPO also outperforms the best penalty on 8B and Olmo-3. This justifies mixing in probability space rather than adding a penalty.

Table 3: SDPO + KL penalty vs. T-SDPO.

Method	Qwen3-4B	Qwen3-8B
SDPO	64.76	67.19
+ KL (best λ)	69.72	70.19
T-SDPO	**77.78**	**75.69**

6 Conclusion

In this paper, we show that, with an ideal teacher, SDPO performs a one-step behavior-regularized improvement on Q^μ, the success probability of an action under the behavior policy's continuation, rather than maximizing the reward of the policy being improved. We then derive the exact optimum of behavior-regularized reward maximization, which for binary reward is a tilted mixture of the behavior policy and its success posterior and recovers the SDPO target only as $\beta \to 0$, and propose Tilted-SDPO, which distills this mixture using SDPO's prompted teacher in place of the posterior. One limitation is that λ is tuned per task; estimating the weight per prompt is a natural next step. We believe this is a concrete step toward deriving self-distillation targets from the reward rather than from the teacher.

AI USE STATEMENT

We used generative AI tools to help launch experiments, analyze and interpret results, and polish the text of this paper. All AI-assisted code, analyses, and text were reviewed by the authors, and the reported numbers were checked against the experiment logs. We take responsibility for the full content of this work.

REPRODUCIBILITY STATEMENT

We take several steps to make our results reproducible. On the theory side, the proof of Theorem 2 is given in Appendix B. On the method side, T-SDPO changes only the distillation target of SDPO, and Appendix C describes how the mixture is formed under top-k truncation. On the experiment side, all base models (Qwen3-4B-Thinking-2507, Qwen3-8B and Olmo-3-7B-Think) and datasets (DAPO, AIME24, AMC23 and SciKnowEval) are publicly available. Appendix D lists every hyperparameter (Table 4), Appendix E specifies the evaluation protocol and the source of every number in Tables 1 and 2. Our implementation builds on the open-source verl framework, and we will release the code, training scripts and evaluation results upon publication.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations (ICLR)*, 2018.

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

Ai2. Olmo 3. `https://huggingface.co/allenai/Olmo-3-7B-Think`, 2025. Model card, `allenai/Olmo-3-7B-Think`.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

David Brandfonbrener, William F. Whitney, Rajesh Ranganath, and Joan Bruna. Offline RL without off-policy evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Wojciech Marian Czarnecki, Razvan Pascanu, Simon Osindero, Siddhant M. Jayakumar, Grzegorz Swirszcz, and Max Jaderberg. Distilling policy distillation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Ken Ding. HDPO: Hybrid distillation policy optimization via privileged self-distillation. *arXiv preprint arXiv:2603.23871*, 2026.

Benjamin Eysenbach, Sergey Levine, and Ruslan Salakhutdinov. Replacing rewards with examples: Example-based policy search via recursive classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Kehua Feng, Keyan Ding, Weijie Wang, et al. SciKnowEval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Justin Fu, Avi Singh, Dibya Ghosh, Larry Yang, and Sergey Levine. Variational inverse control with events: A general framework for data-driven reward definition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning (ICML)*, 2019.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International Conference on Machine Learning (ICML)*, 2019.

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

Sarthak Harne, Chinmay Karkar, Yash Pandya, Ahmed Awadallah, and Akshay Nambi. Privileged, but biased: How PI-conditioned teachers break self-distillation. *arXiv preprint arXiv:2608.04794*, 2026.

Ran He, Yuanda Xu, Zhipeng Wang, Zhengze Zhou, et al. CRISP: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026.

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, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

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

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of LLMs? *arXiv preprint arXiv:2603.24472*, 2026.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Dongha Lee et al. OPSD compresses what RLVR teaches: A post-RL compaction stage for reasoning models. *arXiv preprint arXiv:2605.06188*, 2026.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026a.

Yang Li, Erik Nijkamp, Semih Yavuz, and Shafiq Joty. Learning from language feedback via variational policy distillation. *arXiv preprint arXiv:2605.15113*, 2026b.

Yifeng Liu, Shiyuan Zhang, Yifan Zhang, and Quanquan Gu. Self-distilled policy gradient. *arXiv preprint arXiv:2606.04036*, 2026.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*, 2026.

Xuan-Phi Nguyen, Zeyu Leo Liu, Yang Li, Shrey Pandit, Yiran Zhao, Anurag Koul, and Shafiq Joty. Privileged likelihood is not automatically value: Three checks for token credit in on-policy self-distillation. *arXiv preprint arXiv:2608.09263*, 2026.

Andrei Liviu Nicolicioiu, Mohammad Pezeshki, and Aaron Courville. On-policy self-distillation with sampled demonstrations reduces output diversity. *arXiv preprint arXiv:2606.26091*, 2026.

Mohammad Norouzi, Samy Bengio, Zhifeng Chen, Navdeep Jaitly, Mike Schuster, Yonghui Wu, and Dale Schuurmans. Reward augmented maximum likelihood for neural structured prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

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.

Keqin Peng, Chen Li, Yuanxin Ouyang, Yancheng Yuan, and Liang Ding. Diagnosing and mitigating thinking collapse in on-policy self-distillation. *arXiv preprint arXiv:2607.10805*, 2026.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *International Conference on Machine Learning (ICML)*, 2007.

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

Andrei A. Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. In *International Conference on Learning Representations (ICLR)*, 2016.

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

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

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, et al. BOND: Aligning LLMs with best-of-n distillation. *arXiv preprint arXiv:2407.14622*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, 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.

Guobin Shen, Xiang Cheng, Chenxiao Zhao, Lei Huang, Jindong Li, Dongcheng Zhao, and Xing Yu. Anti-self-distillation for reasoning RL via pointwise mutual information. *arXiv preprint arXiv:2605.11609*, 2026a.

Guobin Shen, Lei Huang, Xiang Cheng, Chenxiao Zhao, Jindong Li, Dongcheng Zhao, and Xing Yu. From generic correlation to input-specific credit in on-policy self distillation. *arXiv preprint arXiv:2605.11613*, 2026b.

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

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

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

Akiyoshi Tomihari and Issei Sato. Power distribution bridges sampling, self-reward RL, and self-distillation. *arXiv preprint arXiv:2605.04542*, 2026.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of KL regularization in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Meng Wang, Haohan Zhao, Wenzhuo Liu, Lu Yang, Geng Liu, Haiyang Guo, Guo-Sen Xie, Gaofeng Meng, Hongbin Liu, and Fei Zhu. Denser $\neq$ better: Limits of on-policy self-distillation for continual post-training. *arXiv preprint arXiv:2607.01763*, 2026.

Xun Wu, Tianzhu Ye, Li Dong, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022.

Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. Offline RL with no OOD actions: In-sample learning via implicit value regularization. In *International Conference on Learning Representations (ICLR)*, 2023.

Jiawei Xu, Minghui Liu, Juzheng Zhang, Tom Goldstein, and Furong Huang. β-OPSD: Deriving with policy optimization, training with self-distillation. *arXiv preprint arXiv:2607.28582*, 2026.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *arXiv preprint arXiv:2604.03128*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Xin Yu et al. Preference-based self-distillation: Beyond KL matching via reward regularization. *arXiv preprint arXiv:2605.05040*, 2026.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Tianjun Zhang, Fangchen Liu, Justin Wong, Pieter Abbeel, and Joseph E. Gonzalez. The wisdom of hindsight makes language models better instruction followers. *arXiv preprint arXiv:2302.05206*, 2023.

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.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Extended related work

On-policy and self-distillation for language models. Distilling a teacher on the student's own samples dates to policy distillation and on-policy imitation (Hinton et al., 2015; Rusu et al., 2016; Ross et al., 2011; Czarnecki et al., 2019) and was adapted to language models by GKD (Agarwal et al., 2024). Self-distillation removes the separate teacher: the same model, given privileged context, supervises its unprompted self, whether that context is a system prompt (Wu et al., 2026), a demonstration (Shenfeld et al., 2026), or environment feedback and successful rollouts in RLVR (Hübotter et al., 2026; Yang et al., 2026; Lu et al., 2026); other work reports that the resulting dense signal can degrade reasoning or fail to help (Kim et al., 2026; Wang et al., 2026). These works choose the teacher and study its effect. Compared with them, we fix the objective and ask which teacher it implies, which shows that SDPO's success-conditioned teacher is one limit of a family rather than the optimum.

Outcome-conditioned and example-based learning. Conditioning behavior on a desired outcome avoids reward engineering: example-based RL learns from samples of success (Fu et al., 2018; Eysenbach et al., 2021), hindsight relabeling reinterprets achieved outcomes as goals (Andrychowicz et al., 2017; Zhang et al., 2023), return-conditioned models generate conditioned on high return (Chen et al., 2021), and filtered self-training, best-of-N distillation and reward-augmented maximum likelihood fine-tune on successes, the best of N samples, or an exponentiated-reward distribution (Zelikman et al., 2022; Sessa et al., 2024; Norouzi et al., 2016). With a binary reward, filtering on success and conditioning on it both target the success posterior. Our result places that target on the temperature axis of KL-regularized RL: it is the zero-temperature limit, and every finite temperature asks for a mixture with the behavior policy instead.

KL-regularized policy improvement. The optimum of reward maximization under a KL penalty is an exponential tilt of the reference policy (Levine, 2018; Geist et al., 2019; Vieillard et al., 2020), the basis of weighted-regression methods (Peters & Schaal, 2007; Peng et al., 2019; Abdolmaleki et al., 2018) and of KL-penalized language-model fine-tuning (Ziegler et al., 2019); trust-region methods, PPO and GRPO bound the same divergence by constraint or clipping (Schulman et al., 2015; 2017; Shao et al., 2024; Guo et al., 2025). What is specific to binary reward is that the tilt is exactly an arithmetic mixture of the reference policy and its success posterior, which turns the optimum into a target a self-distillation pipeline can already form. Conservative policy iteration (Kakade & Langford, 2002) also updates toward a mixture, but chooses it as a safe step size; here the mixture is the exact regularized optimum, with a weight set by the temperature and the behavior success rate.

B Proof of Theorem 2

Throughout, x is fixed and all distributions are over the response space $\mathcal{Y}$; we write $\mu(y)$ for $\mu(y \mid x)$ and $R(y)$ for $R(x, y)$. We assume R is bounded, so that the normalizer below is finite, and we optimize over distributions absolutely continuous with respect to μ — any π placing mass where μ does not has $D_{\mathrm{KL}}(\pi\|\mu) = \infty$ and hence objective $-\infty$, so it cannot be optimal.

Step 1: the optimizer is an exponential tilt of μ. Define

$$Z \;:=\; \sum_y \mu(y)\, e^{R(y)/\beta}, \qquad p(y) \;:=\; \frac{\mu(y)\, e^{R(y)/\beta}}{Z}, \tag{17}$$

so p is a well-defined distribution ($Z \in (0, \infty)$ because R is bounded and μ is a distribution). For any $\pi \ll \mu$, add and subtract $\beta \log Z$ inside the objective of Eq. (11):

$$\begin{aligned} -\beta\, D_{\mathrm{KL}}(\pi\|p) &= -\beta \sum_y \pi(y) \log \frac{\pi(y)\, Z}{\mu(y)\, e^{R(y)/\beta}} \\ &= -\beta \sum_y \pi(y) \log \frac{\pi(y)}{\mu(y)} \;+\; \sum_y \pi(y) R(y) \;-\; \beta \log Z \\ &= \mathbb{E}_{y\sim\pi}[R(y)] - \beta\, D_{\mathrm{KL}}(\pi\|\mu) - \beta \log Z. \end{aligned} \tag{18}$$

Rearranging, the objective equals $-\beta\, D_{\mathrm{KL}}(\pi\|p) + \beta \log Z$. The second term does not depend on π and $\beta > 0$, so maximizing it is equivalent to minimizing $D_{\mathrm{KL}}(\pi\|p)$. Since the Kullback–Leibler divergence is non-negative and vanishes only when its arguments agree, the maximizer exists, is unique, and equals p:

$$\pi^\star_\beta(y \mid x) \;=\; \frac{\mu(y \mid x)\, e^{R(x,y)/\beta}}{Z}. \tag{19}$$

Step 2: the normalizer for binary reward. Let $R \in \{0,1\}$ and recall $V^\mu(x) = \mathbb{E}_{y\sim\mu}[R(x,y)] = \sum_y \mu(y)R(y)$, which for binary reward is the behavior policy's success probability. Splitting the sum in Eq. (17) by the value of R,

$$Z = \sum_{y:\, R(y)=0} \mu(y)\cdot 1 \;+\; \sum_{y:\, R(y)=1} \mu(y)\, e^{1/\beta} = \big(1 - V^\mu(x)\big) + e^{1/\beta} V^\mu(x) = 1 + \big(e^{1/\beta} - 1\big) V^\mu(x). \tag{20}$$

Substituting Eq. (20) into Eq. (19) gives Eq. (12).

Step 3: the tilt is an arithmetic mixture. Assume $V^\mu(x) > 0$, where $V^\mu(x) = P_\mu(R = 1 \mid x)$ is Eq. (6) at the empty prefix, and let $q_+(y \mid x) := P_\mu(Y = y \mid x, R = 1)$ be the sequence-level success posterior — Eq. (5) with the whole response as the action. The same Bayes computation that gives Eq. (7), with deterministic binary reward so that $P_\mu(R = 1 \mid x, y) = R(x, y)$, gives

$$q_+(y \mid x) = \mu(y \mid x)\, \frac{R(x,y)}{V^\mu(x)}. \tag{21}$$

Set

$$\lambda_\beta(x) \;=\; \frac{\big(e^{1/\beta} - 1\big) V^\mu(x)}{Z}, \qquad \text{so that} \qquad 1 - \lambda_\beta(x) = \frac{Z - \big(e^{1/\beta} - 1\big) V^\mu(x)}{Z} = \frac{1}{Z}, \tag{22}$$

where the last equality is Eq. (20). We verify $\pi^\star_\beta = (1 - \lambda_\beta)\mu + \lambda_\beta q_+$ pointwise, separately on the two level sets of the reward. If $R(y) = 0$ then $q_+(y) = 0$, so

$$\big[(1 - \lambda_\beta)\mu + \lambda_\beta q_+\big](y) = \frac{\mu(y)}{Z} = \frac{\mu(y)\, e^{R(y)/\beta}}{Z} = \pi^\star_\beta(y), \tag{23}$$

using $e^0 = 1$. If $R(y) = 1$ then $q_+(y) = \mu(y)/V^\mu(x)$, so

$$\begin{aligned}\big[(1 - \lambda_\beta)\mu + \lambda_\beta q_+\big](y) &= \frac{\mu(y)}{Z} + \frac{\big(e^{1/\beta} - 1\big) V^\mu(x)}{Z} \cdot \frac{\mu(y)}{V^\mu(x)} \\ &= \frac{\mu(y)}{Z}\Big[1 + e^{1/\beta} - 1\Big] \;=\; \frac{\mu(y)\, e^{1/\beta}}{Z} \;=\; \pi^\star_\beta(y).\end{aligned} \tag{24}$$

The two cases exhaust $\mathcal{Y}$, which proves Eq. (13). If instead $V^\mu(x) = 0$ then $\lambda_\beta(x) = 0$ and $Z = 1$, and Eq. (19) gives $\pi^\star_\beta = \mu$: with no successful response in the support of μ, the objective supplies no signal and the optimizer is the behavior policy. □

The reweighting identity. For completeness, the Bayes step behind Eq. (7), whose sequence form Eq. (21) Step 3 uses. At a state s with $V^\mu(s) > 0$,

$$\begin{aligned}q_+(a \mid s) = P_\mu(A = a \mid S = s, R = 1) &= \frac{P_\mu(R = 1 \mid S = s, A = a)\, P_\mu(A = a \mid S = s)}{P_\mu(R = 1 \mid S = s)} \\ &= \mu(a \mid s) \frac{Q^\mu(s,a)}{V^\mu(s)},\end{aligned} \tag{25}$$

which is Eq. (7). At the sequence level the same computation applies with $S = x$ and $A = y$; for deterministic binary reward $P_\mu(R = 1 \mid x, y) = R(x, y)$, giving Eq. (21).

C Implementation of the mixture target

The constant-weight target of Eq. (14) needs two things done carefully.

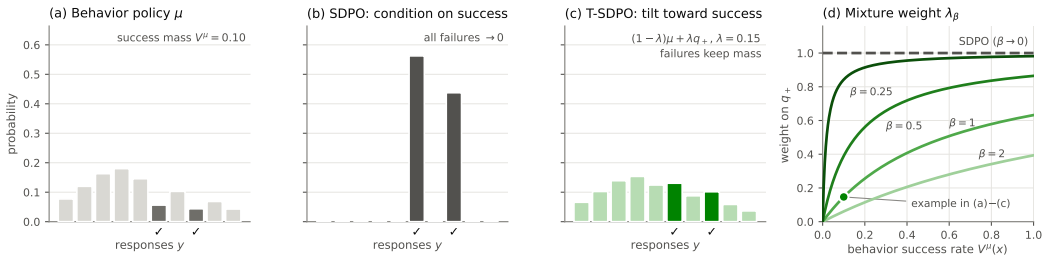


Figure 6: **Conditioning versus tilting.** (a) A behavior policy μ with success mass $V^\mu = 0.10$. (b) SDPO's target conditions on success and removes all mass from failed responses. (c) T-SDPO's target $(1-\lambda)\mu + \lambda q_+$ tilts μ toward success while failed responses keep their mass. (d) The exact mixture weight $\lambda_\beta(x)$ of Theorem 2 as a function of the behavior success rate. It is interior for every finite β, grows with $V^\mu(x)$, and reaches SDPO's $\lambda = 1$ only as $\beta \to 0$. Note that T-SDPO replaces it with a constant in practice (Section 4.3).

Mixing under top-k truncation. The mixture must be formed in probability space, $\log\left((1-\rho)e^{\log \mu} + \rho\, e^{\log q_{y^\star}}\right)$ via log-sum-exp, so that it commutes with the top-k-plus-tail bucketing SDPO uses for the full-vocabulary divergence. Mixing in log space would instead form a geometric mixture, which is not the target of Theorem 2.

Behavior logits. With $\mu = \pi_{\bar\theta}$ (Eq. (1)), the full-vocabulary behavior logits are read from the teacher module on the unaugmented input, at the cost of one extra no-gradient forward pass per micro-batch; this is exact by construction and makes μ and $q_{y^\star}$ share one parameter vector, as Eq. (2) requires. At $\eta = 1$ they could instead be read from the detached student, which equals μ whenever the update is on-policy. All experiments in Section 5 use the teacher module.

Frozen targets and numerical stability. Within one update, the behavior policy μ and the prompted teacher $q_{y^\star}$ are detached, so the target is a constant. The mixture is formed in probability space with a log-sum-exp rather than by interpolating log-probabilities, which keeps it an arithmetic mixture and makes it commute with the top-k truncation SDPO uses for the full-vocabulary divergence. The optimized target is that mixture itself (see "Why not simply add a KL penalty?" in Section 4.3).

Computational cost. Relative to standard SDPO, the only substantial addition is the full-vocabulary behavior logits. They can be cached when memory permits or obtained with one extra no-gradient behavior forward pass; token chunking avoids materializing both distributions for the full sequence at once. Forming the mixture itself is negligible, and with a constant ρ there is nothing further to estimate at run time.

Improvement condition. The arithmetic target also makes its improvement condition transparent. For any prompt teacher q and mixture $m_\lambda = (1-\lambda)\mu + \lambda q$,

$$\mathbb{E}_{m_\lambda}[R] - \mathbb{E}_\mu[R] = \lambda\left(\mathbb{E}_q[R] - \mathbb{E}_\mu[R]\right). \tag{26}$$

Thus the exact posterior always improves the binary reward when $0 < V^\mu < 1$, while the prompt-based target improves only when the prompted teacher has positive verifier advantage. Behavior regularization controls the magnitude of this update; it does not turn an inaccurate example-conditioned teacher into an improving one.

D Experimental configuration

Olmo-3-7B-Think uses the same configuration as the Qwen3 models except for the vLLM memory fraction: it has no grouped-query attention, so its KV cache per token is about $3.6\times$ that of Qwen3-8B and vLLM cannot reserve enough cache at 0.4; this changes rollout throughput and nothing else. Two further configuration details affect interpretation. The divergence uses $\alpha = 0.5$, the

Table 4: Full training configuration. Values shared by all arms and base models unless stated. Each science run trains and evaluates on a single SciKnowEval subject.

	MATH	SCIENCE
Base models	Qwen3-4B-Thinking-2507, Qwen3-8B, Olmo-3-7B-Think	
Train prompts	DAPO	SciKnowEval biology (450) or chemistry (1890)
In-distribution val	AIME24 boxed (30 q)	biology (50 q) or chemistry (210 q) test split
Max prompt / response	2048 / 14336	2048 / 8192
Max sequence (distill)	31744	19456
Max sequence (GRPO)	16896	10752
Question batch	32	
Group size	8	
Train rollout decoding	temperature 1.0, top-p 1.0, top-k disabled	
$y^\star$ selection	uniform over group members with reward ≥ 0.5	
Optimizer updates / step	1 (distillation), 4 (GRPO)	
Learning rate	10^{-5} (distillation), 10^{-6} (GRPO); 10 warmup steps	
Divergence	full-logit top-100 + tail bucket, $\alpha = 0.5$, IS clip 2.0	
Rollout correction	token-level IS, threshold 2.0	
Teacher EMA rate η	$\{0, 0.01, 0.05\}$	
λ	$\{0.25, 0.50, 0.75, 0.90, 1.00\}$	
Training steps	100	
Eval decoding	every 10 steps, avg@12, temperature 0.6 / top-p 0.95 / top-k 20	
vLLM memory fraction	0.4 (Qwen3), 0.5 (Olmo-3)	
Hardware	1 node $\times$ 8 H100 80GB	

JSD interpolation SDPO adopts in practice, rather than the pure reverse KL that Eq. (4) analyzes; T-SDPO and SDPO share it, so the comparison isolates the target. And the in-distribution math validation set must be the boxed-format AIME24: the DAPO-formatted variant asks for `Answer:` while the scorer accepts only boxed answers, so its metric decays toward zero as the policy learns to comply with the prompt, which would be indistinguishable from a training failure.

Table 5: Selected mixture weight λ and teacher EMA rate η of T-SDPO for each result in Tables 1 and 2. SDPO uses the same η with $\lambda = 1$.

	Math		Biology		Chemistry	
Model	λ	η	λ	η	λ	η
Qwen3-4B-Thinking-2507	0.75	0	0.90	0.05	0.90	0.01
Qwen3-8B	0.50	0	0.75	0.05	0.90	0.01
Olmo-3-7B-Think	0.75	0.01	0.90	0.01	0.90	0.01

E EVALUATION PROTOCOL

Benchmarks. Math is evaluated on AIME24 (30 problems) and AMC23 (40 problems), both in boxed format and both graded by the same rule-based `\boxed{}` matcher used for the training reward. The math validation set must be the boxed AIME24 rather than the DAPO-formatted variant, whose `Answer:` prompts fight a boxed-only scorer so that the metric decays toward zero as the policy learns to comply with the prompt. Science is evaluated on the SciKnowEval test split of the training subject: biology (50 questions) and chemistry (210 questions). Each subject is split once into train and test at a 9:1 ratio with a fixed seed, so the test questions never appear in training.

Metrics. **avg@12** is the mean score over the 12 samples of a question, averaged over questions. **pass@12** is the exact per-question best of all 12 samples, again averaged over questions.

F PASS@12 RESULTS

Table 6: Pass@12 on math, for the runs and checkpoints of Table 1. T-SDPO achieves the highest or tied-highest pass@12 on all six cells.

	Qwen3-4B-Thinking-2507		Qwen3-8B		Olmo-3-7B-Think	
Method	AIME24	AMC23	AIME24	AMC23	AIME24	AMC23
Base	53.33	87.50	66.67	**97.50**	66.67	95.00
SDPO	76.67	92.50	76.67	**97.50**	**83.33**	**97.50**
GRPO	66.67	95.00	70.00	95.00	70.00	92.50
T-SDPO	**80.00**	**100.00**	**83.33**	**97.50**	**83.33**	**97.50**