S^3-OPSD: SPARSE, STABLE AND SPEEDY ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

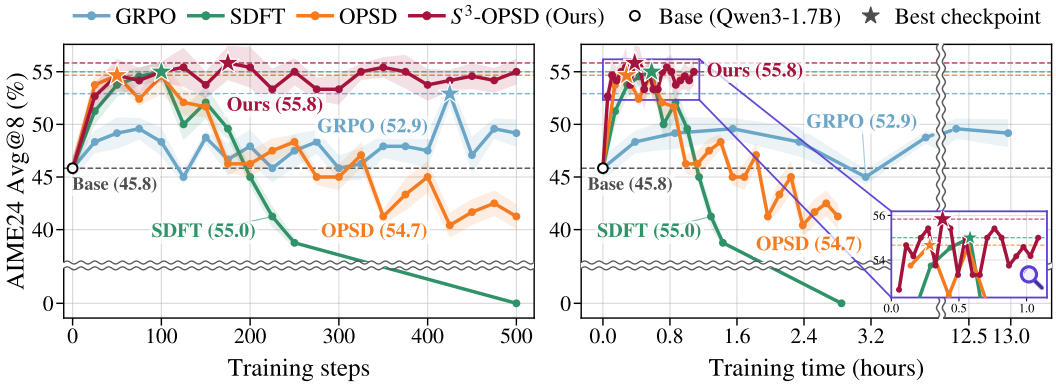


Figure 1: **S^3-OPSD trains stably and fast.** AIME24 Avg@8 of Qwen3-1.7B, plotted against training steps (left) and wall-clock time (right); stars mark the best checkpoint of each run. S^3-OPSD peaks at 55.8 ($+10.0$ over the base model, $+2.9$ over GRPO) and stays stably near point of its peak for the rest of training, while SDFT collapses to 0 and OPSD decays below the base model. Its lower memory footprint also makes each step $\sim 3\times$ faster than SDFT and OPSD and $\sim 13\times$ faster than GRPO on the same hardware at matched effective batch size.

ABSTRACT

Improving LLM reasoning often requires costly demonstrations, rollouts, or strong external teachers. On-policy self-distillation (OPSD) uses the same model conditioned on privileged context as its teacher, but dense supervision is costly and OPSD sometimes exhibit unstable training. We propose S^3-OPSD, a sparse, stable, and speedy framework combining state selection, stable optimization, and target smoothing. Viewing the teacher as an implicit reward-guided Doob-h transform, we define state importance as action-wise advantage variance and derive a low-cost proxy bounded above and below by this variance. We show that SGD enables sparse weight updates, reducing parameter displacement and improving stability over momentum-based optimizers. At selected states, we smooth targets by mixing the teacher distribution with the initial student distribution as a fixed reference policy. Across model families and sizes, mathematical reasoning benchmarks, and multi-task learning, S^3-OPSD improves performance at equal training steps and remains stable over extended training. It also enables memory-efficient training and, at matched effective batch size, reduces training time by up to $13\times$ relative to GRPO and $3\times$ relative to OPSD baselines.

1 INTRODUCTION

Large language models (LLMs) can improve reasoning through supervised fine-tuning (SFT) on chain-of-thought (CoT) (Wei et al., 2022) traces (Zelikman et al., 2022; Kim et al., 2023; Guo et al., 2025), reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; Lambert et al., 2024; Guo et al., 2025; Yu et al., 2025; Hu et al., 2025), or distillation from reasoning-capable teachers (Ho et al., 2023; Hsieh et al., 2023; Agarwal et al., 2024; Guo et al., 2025; Zhao et al.,

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.

2026a). However, these approaches require extensive high-quality demonstrations, costly rollouts and computation, or separate strong teachers, motivating alternatives with lower resource demands.

Among these methods, on-policy self-distillation (OPSD) (Zhao et al., 2026b; Shenfeld et al., 2026), provides dense supervision for every action (*i.e.,* next token) without requiring a separate strong teacher. OPSD constructs a teacher by providing privileged context (*e.g.,* solution), to the same model and distills this distribution into the student's on-policy rollouts. It has been shown to enable effective learning within only a hundred steps. However, the cost of dense distribution matching at every step remains substantial, and training suffers from instability as reported from prior on-policy distillation (OPD) methods (Luo et al., 2026; Xing et al., 2026; Xu et al., 2026; Rang et al., 2026).

To address these limitations, we propose S^3-OPSD, a sparse, stable, and speedy OPSD framework. First, we establish a local connection between OPSD and policy optimization to identify which states (tokens) should receive supervision. Interpreting the teacher as an implicit reward-guided Doob-h transform of the student, we show that forward-KL distillation recovers an all-action policy-gradient update to first order under weak tilting. Prior OPD methods (Xu et al., 2026; Liu et al., 2026; Xiao et al., 2026) use heuristic selection criteria or rely on strong assumptions, such as negligible cross-token gradient covariance. Building on our interpretation and prior observations in RL linking learning signals to uncertain future rewards (Wang et al., 2025; Ni et al., 2026; Huang et al., 2026), we define state importance as the variance of next-token advantages and derive a tractable proxy with upper and lower bounds relative to this importance. This criterion concentrates distillation on consequential decisions, reducing training costs and distribution shift from unnecessary updates.

Second, we identify optimizer choice as an overlooked source of OPSD instability. Compared with momentum-based optimizers, SGD produces substantially sparser parameter changes, stronger alignment between consecutive gradients, and less displacement from initialization, while mitigating performance degradation during extended training. Consistent with prior RL findings (Mukherjee et al., 2026), these observations support our policy-gradient interpretation of OPSD and highlight the stability benefits of sparse updates in self-distillation. Finally, teacher-only supervision at selected states can create large teacher–student discrepancies and destabilize training. We therefore use the initial student distribution as a fixed reference policy and geometrically mix it with the teacher distribution to smooth targets, reduce their discrepancy from the student, and improve stability.

We evaluate S^3-OPSD across multiple instruction-tuned model families and model sizes (Yang et al., 2025; Team et al., 2026) on mathematical reasoning benchmarks (Zhang & Math-AI, 2024; 2025; Dekoninck et al., 2026) and in a multi-task learning setting consists of seven distinct domains. We compare S^3-OPSD against a diverse set of baselines (Shao et al., 2024; Hübotter et al., 2026; Shenfeld et al., 2026; Zhao et al., 2026b; Xu et al., 2026; Xing et al., 2026). At the same number of training steps, S^3-OPSD achieves comparable or better performance and remains stable under extended training. Moreover, it is memory-efficient and, under a constrained memory budget, reduces wall-clock training time by up to 13× relative to GRPO and 3× relative to OPSD and SDFT.

Our contributions can be summarized as follows.

- We propose S^3-OPSD, a sparse, stable, and speedy self-distillation framework. We define state importance as the action-wise advantage variance and derive an inexpensive proxy with two-sided bounds, providing a principled criterion for sparse supervision.
- Building on the RL formulation of OPSD, we show that SGD improves training stability over momentum-based optimizers. We further stabilize training using a geometric mixture of the teacher and initial student distributions as a smoothed supervision target.
- Across diverse models, benchmarks, and a multi-task learning setting, S^3-OPSD improves performance at equal steps, remains stable under extended training, and is memory-efficient, achieving up to 13× speedups over baselines under matched effective batch size.

2 Related Works

2.1 On-Policy Self-Distillation

On-policy distillation (OPD) is a distillation method in which the student generates its own rollouts and is trained using supervision from a teacher (Ross et al., 2011; Schulman et al., 2017; Agarwal

et al., 2024; Zhao et al., 2026a). It has been widely adopted as a post-training approach because it can provide supervision for every action. Because OPD requires a teacher that outperforms the student, on-policy self-distillation (OPSD) (Zhao et al., 2026b; Shenfeld et al., 2026) was introduced. In OPSD, the student serves as its own teacher for OPD when provided with privileged context (*i.e.,* the solution and/or answer). Although OPSD eliminates the need for an intrinsically superior teacher, the training instability (Luo et al., 2026; Xing et al., 2026) and inefficiency (Xu et al., 2026; Rang et al., 2026) of OPD remain unresolved challenges.

2.2 Sparse and Stable On-Policy Distillation

Prior work has improved the training efficiency of OPD by reusing past rollouts (Rang et al., 2026) or performing distillation with offline rollouts (Wu et al., 2026). Another line of work is motivated by the observation that OPD remains effective when applied to only a subset of tokens (Liu et al., 2026; Xu et al., 2026), and improves efficiency by restricting distillation to selected tokens. A separate research direction focuses on stabilizing OPD training (Luo et al., 2026; Xing et al., 2026). For example, TrOPD (Xing et al., 2026) constrains student distributional drift by imposing a KL divergence penalty relative to a reference policy, thereby enabling more stable training. Although sparse training signals are generally effective, the selection of tokens receiving them has largely relied on heuristics, such as entropy. Moreover, the effect of sparsity on training stability, as well as the role of commonly used optimizer choices, remains insufficiently understood. Most importantly, how these factors interact in OPSD remains underexplored.

3 Method

Let s_t be the prefix before generating token a, and c be privileged context (*e.g.,* a solution or answer). The student and privileged-context-conditioned teacher distributions are

$$p_t(a) = p_\theta(a \mid s_t), \qquad q_t(a) = p_{\theta'}(a \mid s_t, c), \tag{1}$$

where θ and θ' are the student and teacher parameters. The θ' is initialized from the θ and is either fixed or updated by an exponential moving average. We optimize θ using

$$\mathcal{L}_{\text{OPSD}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{a\in\mathcal{V}} q_t(a)\log\frac{q_t(a)}{p_t(a)}, \tag{2}$$

where $\mathcal{V}$ is the vocabulary set of student and teacher. This forward-KL objective is standard implementation in OPSD and achieves strong performance within a few hundred steps, but dense distillation can become unstable and collapse during extended training. We address this through sparse state selection from $\sum_{t=1}^{T}(\cdot)$ to limit excessive supervision and distribution shift, SGD in place of AdamW for sparse weight update, and smoothing of $q_t(a)$ to reduce large teacher–student gaps.

3.1 Sparse token selection and supervision

This section develops a principled approach to sparse token selection. Interpreting the privileged-context teacher as a reward-guided Doob-h transform of the student, we first connect forward-KL self-distillation to policy optimization locally under weak tilting. We then define state importance through the variance of next-token advantages and derive a tractable proxy with upper and lower bounds in terms of this importance. Finally, we motivate and introduce blockwise selection to prioritize important tokens while maintaining supervision coverage throughout each trajectory.

Interpreting teacher as Doob-h transform. In OPSD, the teacher is the student augmented with privileged information about the correct answer or solution (Zhao et al., 2026b). Although this conditioning promotes trajectories with correct answers, its effect on the student's generative distribution remains uncharacterized. We view it as reweighting responses toward higher terminal rewards.

Let R denote the bounded reward of a completed response, where c is the information that determines the reward, and let $\gamma > 0$ control the strength of the reweighting. For a prefix s, we define the potential under the student's continuation distribution without privileged context as

$$h_\gamma(s; c) = \mathbb{E}p_\theta\left[\exp(\gamma R) \mid s\right]. \tag{3}$$

For any candidate token a, let $s_t a$ denote the prefix obtained by appending a to s_t.

Teacher as a Doob-h transform. We interpret the teacher's next-token distribution as

$$q_t(a) \approx p_t(a)\frac{h_\gamma(s_t a; c)}{h_\gamma(s_t; c)}. \tag{4}$$

The teacher upweights a token when it increases the potential of the resulting prefix.

This formulation is consistent with perspectives that view conditioning on a reward or desired outcome as reweighting the distribution (Yang & Klein, 2021; Rafailov et al., 2023; Zhao et al., 2024). Moreover, privileged conditioning increasing the likelihood of high-reward rollouts suggests that the teacher distribution may reflect an implicit reward function aligned with this preference, motivating the Doob-h interpretation. The derivation is provided in Appendix Sec. B.1.

We examine the implications of this modeling approach under the forward-KL objective. We define

$$V_\theta(s) = \mathbb{E}_{p_\theta}[R \mid s], \qquad A_t(a) = V_\theta(s_t a) - V_\theta(s_t). \tag{5}$$

Here, $A_t(a)$ is the standard advantage, since token generation deterministically transitions from s_t to $s_t a$. Under the above Doob-h model, weak tilting yields $q_t(a)/p_t(a) = 1 + \gamma A_t(a) + O(\gamma^2)$.

OPSD as policy optimization. With teacher targets and sampled prefixes held fixed, forward-KL distillation locally maximizes the following policy-gradient surrogate, where sg denotes stop-gradient:

$$\mathcal{J}_{\mathrm{RL}} = \frac{1}{T}\sum_{t=1}^{T}\sum_{a\in\mathcal{V}} \mathrm{sg}[p_t(a)A_t(a)]\log p_t(a), \qquad -\nabla_\theta \mathcal{L}_{\mathrm{OPSD}} = \gamma \nabla_\theta \mathcal{J}_{\mathrm{RL}} + O(\gamma^2), \tag{6}$$

Thus, weakly tilted OPSD recovers an all-action policy-gradient update without a learned critic. The equivalence is local and holds to first order in γ. The derivation is provided in Appendix B.2.

Selecting important tokens. Prior sparse OPD approaches (Liu et al., 2026; Xu et al., 2026) select tokens using heuristics such as entropy or KL divergence. Our policy-gradient interpretation of OPSD instead prioritize states where the next-token choice most strongly affects expected reward. We quantify this dependence through the variation in action advantages.

Token importance. Since $\mathbb{E}_{a\sim p_t}[A_t(a)] = 0$, we define the importance of state s_t as

$$C_t = \mathrm{Var}_{a\sim p_t}[A_t(a)] = \mathbb{E}_{a\sim p_t}[A_t(a)^2]. \tag{7}$$

By the law of total variance,

$$\mathrm{Var}_{p_\theta}(R \mid s_t) = C_t + \mathbb{E}_{a\sim p_t}\left[\mathrm{Var}_{p_\theta}(R \mid s_t a)\right]. \tag{8}$$

Thus, C_t captures the component of future reward uncertainty explained by the current action. Selecting states with large C_t concentrates the distillation budget on consequential token decisions.

Directly evaluating C_t requires estimating the future reward for every candidate token. We instead use the Doob-h relation in Eq. 4 to construct a proxy from the student and teacher distributions.

Tractable importance proxy. We define

$$\phi_t(a) = \frac{q_t(a) - p_t(a)}{q_t(a) + p_t(a)}, \qquad S_t = \mathrm{Var}_{a\sim p_t}[\phi_t(a)]. \tag{9}$$

Under a Doob teacher with $R \in [r_{\min}, r_{\max}]$ and $\Delta = r_{\max} - r_{\min}$,

$$m_\gamma^2[\sqrt{C_t} - \eta_\gamma]_+^2 \le S_t \le M_\gamma^2(\sqrt{C_t} + \eta_\gamma)^2, \tag{10}$$

where $m_\gamma = \frac{\gamma}{2}\,\mathrm{sech}^2(\gamma\Delta/2)$, $M_\gamma = \gamma/2$, $\eta_\gamma = \gamma\Delta^2/16$, and $[x]_+ = \max\{x, 0\}$.

The bounded correction prevents extreme teacher-student ratios from dominating the score. As $\gamma \to 0$ (weakly tilted model), the bounds become $S_t = \gamma^2 C_t/4 + O(\gamma^3)$, connecting the proxy to our advantage-based importance criterion. We therefore prioritize states with large S_t without estimating continuation values or generating additional rollouts. Appendix B.3 provides the derivation.

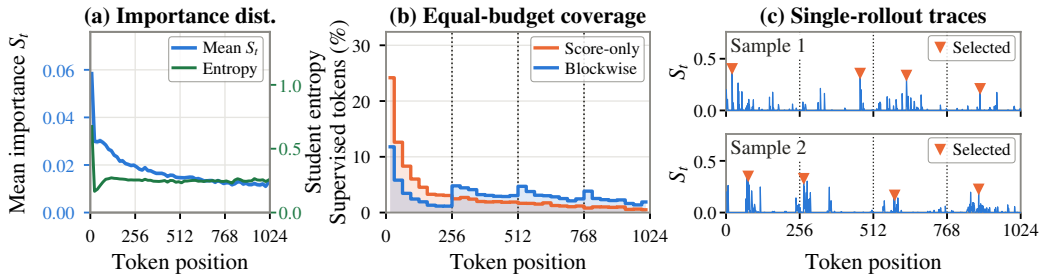


Figure 2: **Token importance and blockwise supervision.** Results from 2,048 on-policy rollouts of Qwen3-1.7B. **(a)** Mean importance score S_t within 16-token bins and student entropy across token positions. **(b)** Selected token positions under score-only and blockwise selection ($B = 256$), each with a budget of 7,054 supervised tokens. **(c)** Importance scores for two individual rollouts, with selected tokens marked by triangles. Dotted vertical lines in **(b)** and **(c)** denote block boundaries.

Blockwise importance selection. To allocate sparse supervision along each trajectory, we examine the positional distribution of S_t. As shown in Fig. 2, earlier tokens consistently exhibit higher mean S_t scores than later tokens whereas student entropy remains relatively constant over the same span. This concentration is consistent with the intuition that early decisions can influence a broader range of future outcomes, while the distinct entropy profile suggests that S_t captures information beyond student uncertainty. However, selecting tokens solely by score concentrates supervision near the beginning of a response (Fig. 2b), potentially overlooking consequential decisions later in the trajectory. Indeed, individual rollouts exhibit localized importance peaks even at later positions (Fig. 2c). We therefore partition each trajectory into contiguous blocks of B tokens and supervise the token with the highest S_t within each block. For a trajectory of length T, this selects $\lceil T/B \rceil$ tokens, preserving coverage throughout the trajectory while prioritizing locally informative states.

3.2 Optimizer for on-policy self-distillation

While sparse token selection can enhance training stability by removing redundant learning signals for model updates, sparse token supervision alone does not ensure sparse parameter changes. Since constraining parameter changes is known to help preserve existing knowledge (Mallya & Lazebnik, 2018; Yang et al., 2024; Kirkpatrick et al., 2017), this motivates examining optimizer as a core means of stabilizing OPSD. Moreover, Mukherjee et al. (2026) showed that SGD in RLVR produces substantially sparser parameter updates than AdamW Loshchilov & Hutter (2019) while maintaining competitive performance. We explore whether similar behavior extends to OPSD by comparing SGD and AdamW under OPSD and SDFT, as our interpretation in Eq. 6 provides an RL perspective.

Fig. 3 reports gradient dynamics and training stability under different optimizers. Fig. 3a shows that parameter sparsity follows a pattern consistent with the RL findings. After training, SGD leaves 88.2% (OPSD) and 84.1% (SDFT) of the trainable parameters within 10^{-5} of their initial values, compared with 2.7% and 1.6% under AdamW. Fig. 3b further reveals differences in gradient alignment accompanying this sparsity: under SGD, gradients at consecutive steps remain aligned, with a mean cosine similarity of 0.61 for OPSD and 0.78 for SDFT after step 100, whereas they are less aligned under AdamW, at 0.04 and 0.32. As less aligned steps may accumulate along a broader range of directions, it can contribute to the reduced parameter-update sparsity observed under AdamW.

This difference in alignment is also accompanied by differences in parameter displacement during training. As shown in Fig. 3c, AdamW produces substantially larger parameter displacement from initialization than SGD under both OPSD and SDFT. Fig. 3d shows the resulting AIME24 accuracy throughout training with AdamW or SGD. Under AdamW, accuracy declines substantially from its peak while SGD mitigates this degradation. Therefore, we combine sparse token selection and parameter updates in S^3-OPSD to concentrate supervision on few informative tokens and parameters.

3.3 Smoothing the teacher distribution

In Sec. 3.1, we introduce sparse supervision to limit excessive student drift and improve training efficiency. In Sec. 3.2, we introduce an SGD optimizer for sparse weight updates to prevent excessive

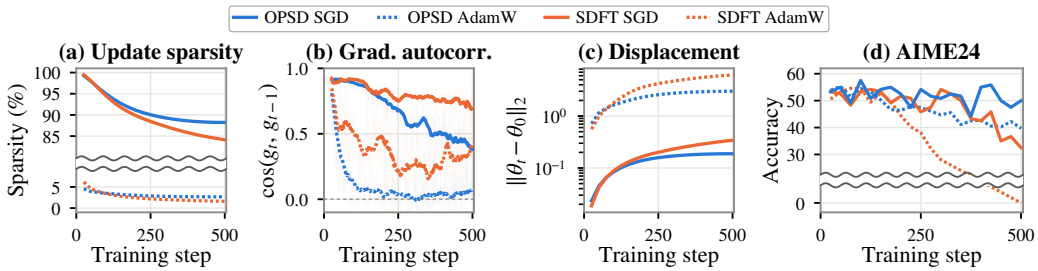


Figure 3: **SGD enables sparse parameter changes and stable performance.** OPSD and SDFT on Qwen3-1.7B with SGD (solid; LR $= 10^{-2}$) and AdamW (dotted; LR $= 5 \times 10^{-6}$), where their learning rates are adopted from the settings of Mukherjee et al. (2026) and Zhao et al. (2026b): (a) parameter sparsity ($|\theta_{t,i} - \theta_{0,i}| \leq 10^{-5}$), (b) moving average of cosine similarity for consecutive-gradient (g_t and g_{t-1}), (c) displacement $\|\theta_t - \theta_0\|_2$, and (d) AIME24 accuracy, respectively.

parameter displacement. However, sparsity alone does not control the magnitude of distributional changes at supervised states, particularly when the teacher parameters θ' are updated during training.

Related approaches in RL and OPD, such as TRPO (Schulman et al., 2015) and TrOPD (Xing et al., 2026), use KL-based controls to limit policy changes. In OPSD, we instead stabilize the training by anchoring the teacher distribution to its initialization. Let θ_{ref} denote the shared initial parameters of the student and teacher. We replace the teacher target with the normalized geometric interpolation

$$q_t(a) = \frac{p_{\theta_{\text{ref}}}(a \mid s_t, c)^{\beta} p_{\theta'}(a \mid s_t, c)^{1-\beta}}{\sum_{b \in \mathcal{V}} p_{\theta_{\text{ref}}}(b \mid s_t, c)^{\beta} p_{\theta'}(b \mid s_t, c)^{1-\beta}}, \tag{11}$$

where $\beta \in [0, 1]$ controls the strength of anchoring to the fixed reference distribution. Setting $\beta = 0$ recovers the current teacher, while $\beta = 1$ uses the initial teacher. We find that this target smoothing mitigates excessive distributional changes and enables more stable OPSD training.

4 Experiments

4.1 Experimental Setup

Baselines. We compare S^3-OPSD with GRPO (Shao et al., 2024), SDPO (Hübotter et al., 2026), SDFT (Shenfeld et al., 2026), OPSD (Zhao et al., 2026b), TIP (Xu et al., 2026), and TrOPD (Xing et al., 2026) on instruction-tuned Qwen3 and Gemma4 models, using official baseline implementations. We train all methods for 500 steps, evaluating checkpoints at steps 100, 300, and 500. We adapt TIP and TrOPD, originally not designed for self-distillation, to use the model itself as the teacher conditioned on privileged context. We match effective batch sizes across all OPSD and OPD-based models for fair comparison. Appendix Sec. C provides baseline implementation details.

Dataset and benchmarks. We evaluate S^3-OPSD on mathematical reasoning and multi-task learning. For mathematical reasoning, we condition the teacher on reasoning traces and answers from OpenThoughts-30K, the OPSD-released mathematics subset of OpenThoughts (Guha et al., 2026), and evaluate student checkpoints on AIME24 (Zhang & Math-AI, 2024), AIME25 (Zhang & Math-AI, 2025), and HMMT25 (Dekoninck et al., 2026). We primarily report Avg@12, the mean accuracy of 12 sampled solutions per problem. For multi-task learning, we train one model on seven datasets spanning tool use (Liu et al., 2025), science (Wu et al., 2018; Ebner et al., 2026), medicine (Khandekar et al., 2024), mathematics (Cobbe et al., 2021), coding (b mc2, 2023; Zhong et al., 2017; Yu et al., 2018), finance (Chen et al., 2021), and commonsense (Lin et al., 2020), conditioning the teacher on each example's reference response. We report mean accuracy on the seven training datasets' evaluation splits (seen) and seven unseen benchmarks (Ryan Teknium & Guang, 2024; Feng et al., 2024; Jin et al., 2021; Hendrycks et al., 2021; Lightman et al., 2024; Liu et al., 2023; Chen et al., 2022; Sakaguchi et al., 2021), one per domain. Details are provided in Appendix Sec. C.

Implementation details. We followed OPSD configurations for mathematical reasoning. Specifically, all models are trained with LoRA (Hu et al., 2022) of rank 64 and each step uses 32 rollouts generated with vLLM (Kwon et al., 2023) with a budget of 1,024 new tokens. Baselines use AdamW

Table 1: **Quantitative comparison across (Top) the Qwen3 model and (Bottom) the Gemma4 family.** All methods are trained on OpenThoughts-30K for 500 steps, with Avg@12 reported at the 100-, 300-, and 500-step checkpoints. SDFT peaks near step 100 before collapsing sharply, while OPSD exhibits a more gradual collapse. In contrast, S^3-OPSD maintains stable performance across all evaluated checkpoints and achieves the highest average performance across the three benchmarks. Best performance is marked with **bold** and second-best performance is marked with underline.

Method	AIME24			AIME25			HMMT25			Average		
	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500
Qwen3-1.7B												
Base (Instruct)		46.94			34.17			23.33			34.81	
+ GRPO	47.50	46.94	48.61	35.56	36.67	34.17	23.89	22.50	22.78	35.65	35.37	35.19
+ SDPO	53.89	41.67	23.61	**39.17**	35.56	14.72	24.72	20.83	12.78	39.26	32.69	17.04
+ SDFT	54.44	28.06	0.00	38.89	20.56	0.00	**26.94**	15.00	0.00	**40.09**	21.21	0.00
+ OPSD	54.44	44.72	39.72	36.39	32.22	28.06	25.28	23.33	20.28	38.70	33.42	29.35
+ TIP	51.39	43.89	38.33	36.11	34.17	30.83	24.17	21.67	19.44	37.22	33.24	29.53
+ TrOPD	44.17	42.22	39.72	35.56	28.89	29.72	21.94	18.06	18.06	33.89	29.72	29.17
+ **S^3-OPSD**	**54.72**	**55.00**	**54.17**	38.89	**38.61**	**39.17**	26.67	**26.39**	**25.56**	**40.09**	**40.00**	**39.63**
Qwen3-4B												
Base (Instruct)		71.67			66.39			44.17			60.74	
+ GRPO	73.61	71.39	72.50	63.06	66.39	65.56	43.61	40.83	42.50	60.09	59.54	60.19
+ SDPO	73.89	62.50	14.44	64.44	55.00	10.56	43.89	36.67	8.06	60.74	51.39	11.02
+ SDFT	75.00	12.50	0.00	69.44	8.33	0.00	**46.39**	9.17	0.00	63.61	10.00	0.00
+ OPSD	72.50	63.33	38.06	67.78	53.06	28.61	41.39	38.61	24.44	60.56	51.67	30.37
+ TIP	73.33	71.11	66.67	59.17	54.17	48.61	40.00	39.17	36.39	57.50	54.82	50.56
+ TrOPD	73.61	71.39	66.11	61.67	58.33	54.72	41.39	41.39	36.11	58.89	57.04	52.31
+ **S^3-OPSD**	**76.94**	**77.22**	**77.22**	**69.72**	**69.44**	**71.11**	45.83	**46.11**	**48.06**	**64.16**	**64.26**	**65.46**
Qwen3-8B												
Base (Instruct)		77.50			66.67			42.22			62.13	
+ GRPO	77.50	77.22	76.67	69.17	67.78	67.22	44.17	45.83	45.56	63.61	63.61	63.15
+ SDPO	73.89	75.28	71.11	66.94	66.11	56.67	41.94	41.39	36.67	60.92	60.93	54.82
+ SDFT	77.50	9.17	0.00	70.56	2.78	0.00	**48.33**	5.56	0.00	65.46	5.84	0.00
+ OPSD	**78.61**	65.56	25.83	70.56	53.61	16.39	44.44	36.94	22.50	64.54	52.04	21.57
+ TIP	76.11	73.06	71.94	63.33	65.56	61.67	43.89	45.28	38.89	61.11	61.30	57.50
+ TrOPD	77.78	75.28	73.61	65.56	60.28	60.00	41.94	42.78	40.00	61.76	59.45	57.87
+ **S^3-OPSD**	78.33	**77.78**	**78.33**	**71.11**	**69.44**	**71.11**	48.06	**48.33**	**47.78**	**65.83**	**65.18**	**65.74**

Method	AIME24			AIME25			HMMT25			Average		
	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500
Gemma4-2B												
Base (Instruct)		41.94			33.06			18.89			31.30	
+ SDPO	44.44	33.33	25.28	31.11	29.44	23.33	19.72	14.72	10.83	31.76	25.83	19.81
+ SDFT	**45.28**	33.61	10.28	**33.61**	26.67	12.22	18.61	8.61	3.89	**32.50**	22.96	8.80
+ OPSD	43.89	40.56	42.78	32.78	30.83	30.00	16.67	20.28	15.56	31.11	30.56	29.45
+ TIP	42.50	43.33	41.39	28.89	30.56	27.50	19.44	17.22	13.89	30.28	30.37	27.59
+ TrOPD	43.06	39.44	38.89	**33.61**	28.33	31.94	17.78	12.50	14.17	31.48	26.76	28.33
+ **S^3-OPSD**	44.17	**43.89**	**43.33**	33.06	**34.17**	**33.61**	**20.00**	**20.56**	**19.44**	32.41	**32.87**	**32.13**
Gemma4-4B												
Base (Instruct)		52.22			38.33			24.17			38.24	
+ SDPO	53.61	52.78	49.72	39.44	**42.22**	38.61	26.67	25.83	21.67	39.91	40.28	36.67
+ SDFT	53.06	50.56	35.28	40.00	41.39	34.44	24.72	22.22	10.28	39.26	38.06	26.67
+ OPSD	52.50	51.67	51.11	41.11	**42.22**	**41.67**	26.11	24.17	23.61	39.91	39.35	38.80
+ TIP	**55.83**	51.94	50.00	41.67	38.33	37.22	28.33	24.72	26.67	41.94	38.33	37.96
+ TrOPD	54.72	50.56	50.83	39.72	39.72	40.00	**29.72**	23.89	24.72	41.39	38.06	38.52
+ **S^3-OPSD**	55.00	**54.72**	**55.28**	**42.78**	41.94	**41.67**	29.44	**28.33**	**28.61**	**42.41**	**41.66**	**41.85**

at the learning rate of 5×10^{-6} while S^3-OPSD uses SGD at 10^{-2}. For Gemma4, the baselines use AdamW at 10^{-6}, because 5×10^{-6} degrades or collapses SDFT, SDPO, and OPSD on Gemma4-2B. For multi-task learning, we fully fine-tune Qwen3-1.7B for 500 steps. We also use 32 rollouts for each training step with a budget of 1,024 new tokens. Appendix Sec. C.2 give the full configurations for baselines and the proposed method.

4.2 RESULTS

Mathematical reasoning. Fig. 1 reports AIME24 (Zhang & Math-AI, 2024) Avg@8 performance at 25-step intervals during mathematical reasoning training. GRPO improves slowly with training steps and requires more wall-clock time than OPSD-based methods. SDFT and OPSD achieve strong performance within fewer steps but subsequently deteriorate. In contrast, S^3-OPSD maintains stable performance after reaching its peak, while achieving wall-clock speedups of $13\times$ over GRPO

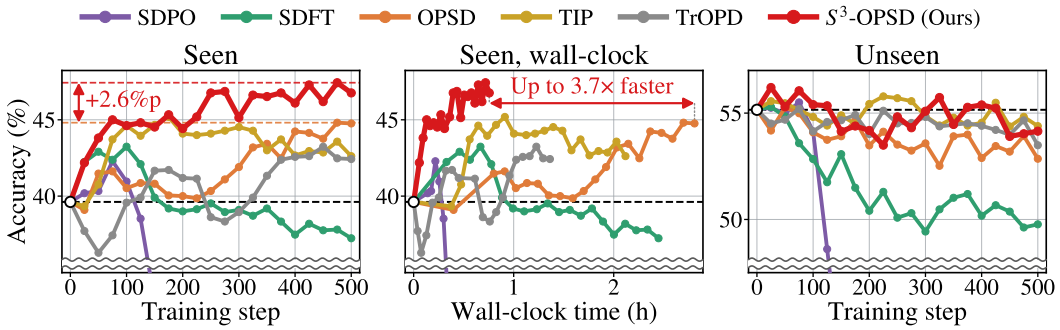


Figure 4: **Multi-task learning.** Qwen3-1.7B trained on seven domains; mean accuracy on their training tasks vs. steps (left) and wall-clock time (middle), and on unseen benchmarks of these domains (right); dashed line represents the base performance. S^3-OPSD achieves the highest seen accuracy and trains 1.3–3.7× faster than baselines, while relieving forgetting on unseen benchmarks.

and 3× over SDFT and OPSD. Tab. 1 extends this evaluation to multiple benchmarks, model families, and model sizes using Avg@12. Across these settings, S^3-OPSD consistently matches or outperforms competing methods and sustains or improves performance as training progresses. It also achieves wall-clock speedups from 1.9× to 13× over the other methods in this comparison.

Multi-task learning. Fig. 4 reports the mean accuracy on the seven seen and the seven unseen benchmarks during multi-task training. S^3-OPSD achieves the highest mean seen accuracy and S^3-OPSD is the only method that improves over the base model on all seven seen tasks at final step (Fig. 4). On the unseen benchmarks, S^3-OPSD successfully mitigates the forgetting while several baselines severely degrades on the unseen benchmarks. Furthermore, S^3-OPSD trains 0.76 h for reaching the final training step, 1.3–3.7× faster than the other on-policy methods, and surpasses the best seen accuracy of every on-policy baseline within 0.3 h (Fig. 4, middle). Per-task results on the seen and unseen benchmarks are provided in Appendix Sec. D.3.

The chosen tokens. To examine which states S_t selects from the student's rollouts, we analyze on 9,600 on-policy rollouts from the base Qwen3-1.7B, using position-matched enrichment to account for higher S_t near the response beginning (Fig. 2). Fig. 5 shows that the selected tokens are enriched for reasoning transitions, sentence and step openings, and termination, while numerical and formatting tokens are underrepresented. These positions also exhibit greater student–teacher disagreement and higher teacher confidence. These findings suggest that S_t identifies branching points where the student is uncertain about how to proceed and the privileged teacher provides a more confident alternative.

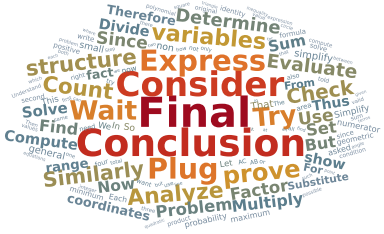


Figure 5: **Word cloud of selected tokens.** We visualize the tokens selected by our sparse supervision.

4.3 Ablation Study

We conduct an ablation study to examine how each component of S^3-OPSD contributes to performance and stability. The results in Tab. 2 compare configurations with different combinations of components. For efficiency, all experiments use Qwen3-1.7B and are evaluated on AIME24 and HMMT25. We follow the default S^3-OPSD configuration except for the study subject.

On token selection (ID 1–3). Random token selection per block (ID 1) reduces average accuracy, and selecting the lowest-S_t token (ID 2) reduces it further, indicating that S_t identifies important supervision states. Supervising every token (ID 3) matches S^3-OPSD at step 100 but degrades through step 500, making dense supervision both costlier and less stable over extended training.

On blockwise selection (ID 4–7). Global top-k selection reduces performance across training steps, suggesting that distributing supervision along the trajectory matters, consistent with high-scoring tokens clustering near response beginnings (Fig. 2b). Smaller blocks increase supervision density without improving overall performance, whereas larger blocks generally reduce accuracy.

Table 2: **Ablation study.** We evaluate token selection, blockwise selection, optimizer choice, and anchoring configurations at training steps 100, 300, and 500. Best results in each column are **bold** and second-best results are underlined.

ID	Study subject	AIME24			HMMT25			Average		
		step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500
0	S^3**-OPSD (Base)**	54.72	**55.00**	**54.17**	26.67	**26.39**	25.56	40.70	**40.70**	**39.87**
1	Random token selection	51.39	50.28	52.22	24.72	24.44	23.61	38.06	37.36	37.92
2	Low S_t token selection	48.06	46.39	47.22	23.89	23.06	23.33	35.98	34.73	35.28
3	Dense token selection	54.44	50.00	48.61	26.39	25.56	24.17	40.42	37.78	36.39
4	Without blockwise selection	52.50	52.22	53.06	25.00	25.83	25.56	38.75	39.03	39.31
5	Block size $B = 128$	53.89	54.44	**54.17**	26.11	25.56	25.28	40.00	40.00	39.73
6	Block size $B = 512$	54.72	50.00	51.67	25.56	24.44	25.00	40.14	37.22	38.34
7	Block size $B = 1024$	51.11	50.00	51.94	24.17	24.44	25.56	37.64	37.22	38.75
8	AdamW optimizer	**55.00**	51.39	49.44	**27.22**	24.17	21.67	**41.11**	37.78	35.56
9	Anchoring with fwd-KL	53.61	54.17	53.06	25.83	25.28	**26.11**	39.72	39.73	39.59
10	Anchoring with rev-KL	50.28	50.83	50.00	24.17	23.89	24.44	37.23	37.36	37.22
11	$\beta = 0.0$	54.17	54.72	51.39	26.67	25.00	23.33	40.42	39.86	37.36
12	$\beta = 0.5$	52.50	51.39	53.61	25.00	25.56	**26.11**	38.75	38.48	39.86
13	$\beta = 0.7$	51.67	52.50	53.61	25.00	24.72	25.28	38.34	38.61	39.45

On optimizer selection (ID 8). Replacing SGD with AdamW (ID 8) yields the highest average score at step 100, followed by a steady decline. This mirrors the behavior of OPSD and SDFT observed in Sec. 3.2: AdamW achieves strong early performance but fails to sustain these gains. Nevertheless, the degradation is substantially milder for S^3-OPSD than for OPSD and SDFT, suggesting that its remaining components help mitigate performance deterioration.

On teacher distribution smoothing (ID 9–13). We compare the geometric mixture with additive forward- or reverse-KL anchoring toward the fixed reference distribution:

$$\mathcal{L}_{\mathrm{anc-fwd}} = \mathrm{KL}(p_{\theta_{\mathrm{ref}}}(a \mid s_t, c) \| p_\theta(a \mid s_t)), \quad \mathcal{L}_{\mathrm{anc-rev}} = \mathrm{KL}(p_\theta(a \mid s_t) \| p_{\theta_{\mathrm{ref}}}(a \mid s_t, c)). \quad (12)$$

From Tab. 2, both alternatives underperform the geometric mixture, particularly reverse-KL anchoring (ID 9–10). Removing smoothing leads to greater late-stage degradation (ID 11), whereas stronger smoothing preserves final performance but limits early gains (ID 12–13).

5 Discussion

Our analysis focuses on the forward-KL objective. Although prior OPSD studies discuss Jensen–Shannon divergence (JSD) (Zhao et al., 2026b) and reverse KL (Shenfeld et al., 2026), their implementations use forward-KL. We likewise find it most effective and connect it to policy-gradient optimization. S^3-OPSD improves training stability, reduces memory usage, and accelerates training (Appendix Tab. A5 and Tab. A7). Despite modest FLOPs reductions over existing methods (Appendix Tab. A6), speedups primarily arise from lower memory usage, enabling larger microbatches, fewer gradient accumulation steps, and fewer separate forward passes. With matched microbatch sizes and no gradient accumulation, all methods have comparable wall-clock times for equal optimization steps, yet S^3-OPSD performs best at every step (Appendix Tab. A12), suggesting gains beyond microbatch-enabled efficiency. Further FLOPs reductions remain promising future work.

6 Conclusion

We propose S^3-OPSD, a framework for stable and efficient on-policy self-distillation. By interpreting the teacher as a reward-guided Doob-h transform of the student, we connect forward-KL distillation to policy optimization locally under weak tilting. This perspective motivates advantage-based token importance and a tractable proxy for sparse supervision allocation. Complementing token selection, SGD promotes sparse parameter updates and improves stability while preserving performance, and geometric interpolation between teacher and fixed reference distributions limits excessive distributional drift. Across mathematical reasoning and multi-task benchmarks, S^3-OPSD exceeds the performance of existing OPSD, GRPO, and stability- or sparsity-focused OPD methods with up to $13\times$ speedup. S^3-OPSD offer a reinforcement learning perspective on OPSD and highlight sparsity in both token supervision and parameter updates for stable and efficient OPSD.

AI USE STATEMENT

In this work, we used generative AI tools for code generation, derivation verification, experiment managing and manuscript polishing. We have not used generative AI tools for idea generation. We have reviewed all AI-assisted work. All the generated code with generative AI tools was supervised and reviewed by authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we report mathematical reasoning performance averaged over 8 to 12 independent evaluation runs to account for sensitivity to random seeds. Our analyses use at least 2,000 samples to improve the reliability of the findings. The supplementary materials include training and evaluation code for both our method and the baselines, enabling reproduction of the results reported in Tab. 1 and Fig. 4.

REFERENCES

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

b mc2. sql-create-context dataset, 2023. URL `https://huggingface.co/datasets/b-mc2/sql-create-context`. This dataset was created by modifying data from the following sources: Zhong et al. (2017); Yu et al. (2018).

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. In *EMNLP*, pp. 3697–3711, 2021.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. In *EMNLP*, pp. 6279–6292, 2022.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *ICMLW*, 2026.

Antonia Ebner, Christoph Bartmann, Sonja Topf, Günter Klambauer, Johannes Schimunek, and Sohvi Luukkonen. Measuring ai progress in drug discovery: A reproducible leaderboard for the tox21 challenge. *Journal of Chemical Information and Modeling*, 2026.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. In *ICLR*, volume 2026, pp. 108059–108130, 2026.

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

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

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In *ACL*, pp. 14852–14882, 2023.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *ACL*, pp. 8003–8017, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *NeurIPS*, 38:162239–162262, 2025.

Guanhua Huang, Tingqiang Xu, Mingze Wang, Qi Yi, Xue Gong, Siheng Li, Ruibin Xiong, Kejiao Li, Yuhao Jiang, and Bo Zhou. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward. In *ACL Findings*, pp. 24158–24188, 2026.

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.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina S Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad W Safranek, Abid A Anwar, Andrew Zhang, et al. Medcalc-bench: Evaluating large language models for medical calculations. *NeurIPS*, 37:84730–84745, 2024.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In *EMNLP*, pp. 12685–12708, 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *SOSP*, pp. 611–626, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*, volume 2024, pp. 39578–39601, 2024.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *EMNLP Findings*, pp. 1823–1840, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.165.

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. *NeurIPS*, 36:21558–21572, 2023.

Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong WANG, et al. Toolace: Winning the points of llm function calling. In *ICLR*, volume 2025, pp. 41359–41381, 2025.

Zhishuai Liu, Xingzi Xu, Mehmet Saygin Seyfioglu, Pan Xu, and Karim Bouyarmane. Extremely sparse supervision incentivizes reasoning ability. *arXiv preprint arXiv:2609.04565*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. Demystifying opd: Length inflation and stabilization strategies for large language models. *arXiv preprint arXiv:2604.08527*, 2026.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pp. 7765–7773. IEEE, 2018.

Sagnik Mukherjee, Lifan Yuan, Pavan Jayasinha, Dilek Hakkani-Tür, and Hao Peng. Do we need adam? surprisingly strong and sparse reinforcement learning with sgd in llms. *ICML*, 2026.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, et al. The flexibility trap: Rethinking the value of arbitrary order in diffusion language models. *ICML*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 36:53728–53741, 2023.

Miao Rang, Zhenni Bi, Hang Zhou, Kai Han, Xuechun Wang, An Xiao, Xinghao Chen, Yunhe Wang, and Hanting Chen. Near-policy: Accelerating on-policy distillation via asynchronous generation and selective packing. *arXiv preprint arXiv:2605.05940*, 2026.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Jeffrey Ryan Teknium, Quesnelle and Chen Guang. Hermes 3 technical report. *arXiv preprint arXiv:2408.11857*, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, pp. 1889–1897. Pmlr, 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, YK Li, Yang Wu, et al. 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. *ICML*, 2026.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *NeurIPS*, 38:115452–115486, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35: 24824–24837, 2022.

Yecheng Wu, Song Han, and Hai Cai. Lightning opd: Efficient post-training for large reasoning models with offline on-policy distillation. *arXiv preprint arXiv:2604.13010*, 2026.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Jinwei Xiao, Zhuowen Han, Yueqing Sun, Zhengxi Lu, Yuxin Liu, Zhiyuan Yao, Wentao Chen, Qi Gu, and Xunliang Cai. Finding the evidence: Discovering decision-supporting tokens for on-policy reasoning distillation. *arXiv preprint arXiv:2606.22830*, 2026.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026.

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.

Kevin Yang and Dan Klein. Fudge: Controlled text generation with future discriminators. In *NAACL*, pp. 3511–3535, 2021.

Xinyu Yang, Jixuan Leng, Geyang Guo, Jiawei Zhao, Ryumei Nakada, Linjun Zhang, Huaxiu Yao, and Beidi Chen. S^2ft: Efficient, scalable and generalizable llm fine-tuning by structured sparsity. *NeurIPS*, 37:59912–59947, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *NeurIPS*, 38:113222–113244, 2025.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*, 2018.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *NeurIPS*, 35:15476–15488, 2022.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Anhao Zhao, Haoran Xin, Yingqi Fan, Junlong Tong, Wenjie Li, and Xiaoyu Shen. Decoupling kl and trajectories: A unified perspective for sft, dagger, offline rl, and opd in llm distillation. *arXiv preprint arXiv:2605.16826*, 2026a.

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. *ICML*, 2026b.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential monte carlo. *ICML*, 2024.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

A CODE

We provide the training and evaluation code for S^3-OPSD to facilitate reproducibility. Detailed instructions for running the code are provided in `README.md`. The trained weights and complete codebase will be publicly released upon acceptance.

B DERIVATIONS

B.1 DERIVATION OF EQ. 4

Fix privileged context c and a prefix s_t. Let y denote a continuation completing the response, and write $R(s_t y)$ for its bounded terminal reward. Exponentially tilting the student's continuation distribution gives

$$\widetilde{p}_\gamma(y \mid s_t, c) = \frac{p_\theta(y \mid s_t)\exp(\gamma R(s_t y))}{h_\gamma(s_t; c)}, \tag{A1}$$

where the normalizing constant is

$$h_\gamma(s_t; c) = \sum_y p_\theta(y \mid s_t)\exp(\gamma R(s_t y)) = \mathbb{E}_{p_\theta}[\exp(\gamma R) \mid s_t]\,. \tag{A2}$$

Next-token distribution. Write $y = az$, where a is the first token and z is the remaining continuation. Autoregressive factorization gives

$$p_\theta(az \mid s_t) = p_t(a)\, p_\theta(z \mid s_t a). \tag{A3}$$

Marginalizing over z, the tilted next-token distribution is

$$\begin{aligned}\widetilde{q}_t(a) &= \sum_z \widetilde{p}_\gamma(az \mid s_t, c) \\ &= \frac{p_t(a)}{h_\gamma(s_t; c)} \sum_z p_\theta(z \mid s_t a)\exp(\gamma R(s_t a z)) \\ &= p_t(a)\frac{h_\gamma(s_t a; c)}{h_\gamma(s_t; c)}.\end{aligned} \tag{A4}$$

The potential satisfies

$$h_\gamma(s_t; c) = \sum_{a \in \mathcal{V}} p_t(a) h_\gamma(s_t a; c), \tag{A5}$$

so $\widetilde{q}_t$ is normalized. Moreover,

$$\frac{\widetilde{q}_t(a)}{p_t(a)} = \frac{h_\gamma(s_t a; c)}{h_\gamma(s_t; c)}. \tag{A6}$$

Thus, a token is upweighted precisely when its resulting prefix has greater potential than the current prefix.

Connection to the privileged teacher. The identity above holds exactly for the exponentially tilted student distribution. We model the privileged teacher as approximating this distribution, yielding

$$q_t(a) \approx \widetilde{q}_t(a) = p_t(a)\frac{h_\gamma(s_t a; c)}{h_\gamma(s_t; c)}. \tag{A7}$$

B.2 DERIVATION OF EQ. 6

We consider a bounded, undiscounted terminal reward R with fixed privileged context c. All value functions and teacher targets are evaluated under the current student policy. During each update, the teacher targets and sampled prefixes are held fixed.

Value and advantage. Appending token a to prefix s_t deterministically produces $s_t a$. Consequently, the action value equals $V_\theta(s_t a)$, and

$$A_t(a) = V_\theta(s_t a) - V_\theta(s_t), \qquad V_\theta(s) = \mathbb{E}_{p_\theta}[R \mid s]. \tag{A8}$$

By the law of total expectation,

$$V_\theta(s_t) = \sum_{a \in \mathcal{V}} p_t(a) V_\theta(s_t a), \qquad \sum_{a \in \mathcal{V}} p_t(a) A_t(a) = 0. \tag{A9}$$

Small-γ expansion. Boundedness of R gives

$$\begin{aligned} h_\gamma(s; c) &= \mathbb{E}_{p_\theta}[\exp(\gamma R) \mid s] \\ &= 1 + \gamma V_\theta(s) + O(\gamma^2). \end{aligned} \tag{A10}$$

Under the ideal Doob-h model, the teacher therefore satisfies

$$\begin{aligned} \frac{q_t(a)}{p_t(a)} &= \frac{h_\gamma(s_t a; c)}{h_\gamma(s_t; c)} \\ &= \frac{1 + \gamma V_\theta(s_t a) + O(\gamma^2)}{1 + \gamma V_\theta(s_t) + O(\gamma^2)} \\ &= 1 + \gamma \big[V_\theta(s_t a) - V_\theta(s_t) \big] + O(\gamma^2) \\ &= 1 + \gamma A_t(a) + O(\gamma^2). \end{aligned} \tag{A11}$$

Forward-KL gradient. With q_t held fixed, differentiating the OPSD objective gives

$$\nabla_\theta \mathcal{L}_{\mathrm{OPSD}} = -\frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} q_t(a) \nabla_\theta \log p_t(a). \tag{A12}$$

Substituting the expansion above yields

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\mathrm{OPSD}} = &- \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} p_t(a) \nabla_\theta \log p_t(a) \\ &- \frac{\gamma}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} p_t(a) A_t(a) \nabla_\theta \log p_t(a) + O(\gamma^2). \end{aligned} \tag{A13}$$

The first term vanishes because

$$\sum_{a \in \mathcal{V}} p_t(a) \nabla_\theta \log p_t(a) = \nabla_\theta \sum_{a \in \mathcal{V}} p_t(a) = 0. \tag{A14}$$

Hence,

$$-\nabla_\theta \mathcal{L}_{\mathrm{OPSD}} = \frac{\gamma}{T} \sum_{t=1}^{T} \mathbb{E}_{a \sim p_t} \left[A_t(a) \nabla_\theta \log p_t(a) \right] + O(\gamma^2). \tag{A15}$$

Policy-gradient surrogate. Define

$$\mathcal{J}_{\mathrm{RL}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} \mathrm{sg}[p_t(a) A_t(a)] \log p_t(a). \tag{A16}$$

The stopped coefficients are recomputed at the current policy for each update. Its gradient is

$$\nabla_\theta \mathcal{J}_{\mathrm{RL}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{a \in \mathcal{V}} p_t(a) A_t(a) \nabla_\theta \log p_t(a), \tag{A17}$$

which establishes

$$-\nabla_\theta \mathcal{L}_{\mathrm{OPSD}} = \gamma \nabla_\theta \mathcal{J}_{\mathrm{RL}} + O(\gamma^2). \tag{A18}$$

This is the usual advantage-weighted policy gradient with the action expectation evaluated over the full vocabulary. For on-policy prefixes and a fixed horizon T, its expectation equals $T^{-1} \nabla_\theta \mathbb{E}_{p_\theta}[R]$. The result concerns local update directions, rather than equality of the two objective values.

B.3 Derivation of the Importance Proxy

Assume an exact Doob teacher and a bounded terminal reward $R \in [r_{\min}, r_{\max}]$. Write $\Delta = r_{\max} - r_{\min}$ and fix the privileged context c throughout.

Relating the exponential potential to value. Define

$$W_\gamma(s) = \frac{1}{\gamma} \log h_\gamma(s; c), \qquad \delta_\gamma(s) = W_\gamma(s) - V_\theta(s). \tag{A19}$$

Jensen's inequality and Hoeffding's lemma give

$$0 \le \log \mathbb{E}_{p_\theta} \left[e^{\gamma(R - V_\theta(s))} \mid s \right] \le \frac{\gamma^2 \Delta^2}{8}. \tag{A20}$$

Consequently,

$$0 \le \delta_\gamma(s) \le \frac{\gamma \Delta^2}{8}. \tag{A21}$$

Thus, the exponential potential differs from the expected reward by a uniformly bounded term.

Comparing action-wise variances. For $a \sim p_t$, let

$$X = V_\theta(s_t a), \qquad D = \delta_\gamma(s_t a), \qquad Y = W_\gamma(s_t a) = X + D. \tag{A22}$$

Since subtracting the state baseline does not change variance, $\mathrm{Var}[X] = C_t$. Moreover, the range bound on D implies

$$\sqrt{\mathrm{Var}[D]} \le \frac{\gamma \Delta^2}{16} = \eta_\gamma. \tag{A23}$$

Applying the triangle and reverse triangle inequalities to the centered variables yields

$$[\sqrt{C_t} - \eta_\gamma] + \le \sqrt{\mathrm{Var}[Y]} \le \sqrt{C_t} + \eta\gamma. \tag{A24}$$

From potential variance to the proxy. The exact Doob relation gives

$$\phi_t(a) = \frac{h_\gamma(s_t a; c) - h_\gamma(s_t; c)}{h_\gamma(s_t a; c) + h_\gamma(s_t; c)} = \tanh\left(\frac{\gamma}{2} [W_\gamma(s_t a) - W_\gamma(s_t)] \right). \tag{A25}$$

Fix $w = W_\gamma(s_t)$ and define $f_w(y) = \tanh(\gamma(y - w)/2)$. Both y and w lie in $[r_{\min}, r_{\max}]$, so

$$m_\gamma = \frac{\gamma}{2} \operatorname{sech}^2\left(\frac{\gamma \Delta}{2} \right) \le f_w'(y) \le \frac{\gamma}{2} = M_\gamma. \tag{A26}$$

For independent copies Y, Y', the pairwise variance identity and the mean value theorem give

$$S_t = \frac{1}{2} \mathbb{E}[(f_w(Y) - f_w(Y'))^2], \quad m_\gamma^2 \,\mathrm{Var}[Y] \le S_t \le M_\gamma^2 \,\mathrm{Var}[Y]. \tag{A27}$$

Combining the two variance comparisons establishes

$$m_\gamma^2 [\sqrt{C_t} - \eta_\gamma] +^2 \le S_t \le M \gamma^2 (\sqrt{C_t} + \eta_\gamma)^2. \tag{A28}$$

Weak-tilting limit. For fixed reward bounds, $\eta_\gamma = O(\gamma)$ and $m_\gamma = M_\gamma + O(\gamma^3) = \gamma/2 + O(\gamma^3)$. Since $C_t \le \Delta^2/4$, the preceding bounds imply

$$S_t = \frac{\gamma^2}{4} C_t + O(\gamma^3). \tag{A29}$$

The additive slack accounts for differences in conditional reward distributions beyond their means and vanishes under weak tilting.

C Details

C.1 Datasets and benchmarks

Training corpus for mathematical reasoning. OpenThoughts-30K is the mathematics subset of OpenThoughts (Guha et al., 2026) that the OPSD authors (Zhao et al., 2026b) released with their code. It holds 29,434 problems, most of them from olympiad collections and the MATH dataset. Each problem comes with a DeepSeek-R1 reasoning trace whose final answer matches the gold answer. The reasoning trace is the reference solution shown to the teacher; the short solution summary in the dataset is not used. The gold answer is used only to select the training prompts and to grade rollouts for GRPO and SDPO. For training, we used 16,000 prompts among them.

Evaluation protocols for matchmatical reasoning. We evaluated all benchmarks with vLLM. We sample 12 solutions per problem in thinking mode at temperature 1.0 and top-p 0.95, with up to 38,912 new tokens, which leaves 2,048 tokens of the 40,960-token context for the prompt. Then, the 360 solutions of a benchmark are generated together for AIME24, AIME25, and HMMT25. The last boxed expression of each solution is compared symbolically with the gold answer. Tab. 1 reports Avg@12 at steps 100, 300, and 500, and Pass@12 and Maj@12 are reported in Tab. A8 and Tab. A9. The curves in Fig. 1 were evaluated every 25 steps with the same protocol and reported Avg@8.

Training data for multi-task learning. The multi-task setting uses one dataset per domain: ToolACE (Liu et al., 2025) for tool use, Tox21 (Wu et al., 2018; Ebner et al., 2026) for science related to the toxicity assays, MedCalc-Bench (Khandekar et al., 2024) for medicine, GSM8K (Cobbe et al., 2021) for mathematics, sql-create-context (b mc2, 2023; Zhong et al., 2017; Yu et al., 2018) for coding, FinQA (Chen et al., 2021) for finance, and CommonGen (Lin et al., 2020) for commonsense. Each example has a reference response, a solution for GSM8K and MedCalc-Bench, and the formatted gold answer for the other datasets. The teacher is conditioned on this response with the teacher prompt style of each method, except for SDPO, whose teacher sees the feedback of its rollout group as in mathematical reasoning.

Multi-task evaluation. The unseen benchmarks are Hermes function calling (Ryan Teknium & Guang, 2024), SciKnowEval physics (Feng et al., 2024), MedQA (Jin et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), HumanEval+ (Liu et al., 2023), ConvFinQA (Chen et al., 2022), and WinoGrande (Sakaguchi et al., 2021), one per domain (Fig. A1 and Tab. A10). Every 25 steps, each checkpoint is evaluated using greedy decoding in non-thinking mode. Each benchmark uses at most its first 300 examples: 300 from the evaluation split of each training dataset, the first 300 problems of MATH-500, the 163 problems of HumanEval+ in the release we use 299 problems of SciKnowEval physics, and 300 examples of each remaining unseen benchmark.

Prompt templates. Tab. A1 lists the prompt templates; the student prompt is the same for every method and contains no privileged text. For Qwen3 and Gemma4, training rollouts use the non-thinking mode and are therefore direct solutions of at most 1,024 tokens, whereas evaluation uses the thinking mode with the budget given below. The teacher prompt uses the thinking mode (Tab. A1), the student's rollout follows it as the teacher's reply without a reasoning segment, and the teacher scores this reply token by token. This pairing of a thinking-mode teacher with a non-thinking rollout follows the released OPSD code. OPSD, TIP, and TrOPD use the OPSD teacher prompt, which asks the model to derive the same answer without copying the reference solution. SDFT and S^3-OPSD use its own teacher prompt and the SDPO teacher provides a correct rollout of the group and a fixed sentence of feedback for an incorrect attempt.

C.2 Hyperparameters

Table A2 summarizes the shared experimental settings for the baselines and S^3-OPSD. All methods use rank-64 LoRA adapters with bf16 model weights and fp32 adapter parameters for mathematical reasoning. Each training step generates 32 rollouts: one per prompt for 32 prompts by default, four per prompt for eight prompts in SDPO, and eight per prompt for four prompts in GRPO. Rollouts are generated in non-thinking mode with a temperature of 1.1, top-p of 0.95, top-k of 20, and a

Table A1: **Student and teacher prompts.** {question} is the problem, prefixed by "Return your final response within \boxed{}." as in the dataset; {reference solution} is the DeepSeek-R1 trace; {rollout} is a rollout of the student. Each prompt is wrapped in the model's chat template. We followed each prompts in their official implementation.

Prompt	Text
Student (all methods)	`Problem: {question}` `Please reason step by step, and put your final answer within \boxed{}.`
Teacher, OPSD style (OPSD, TIP, TrOPD)	`Problem: {question}` `Here is a reference solution to this problem:` `=== Reference Solution Begin ===` `{reference solution}` `=== Reference Solution End ===` `After reading the reference solution above, make sure you truly understand the reasoning behind each step --- do not copy or paraphrase it. Now, using your own words and independent reasoning, derive the same final answer to the problem above. Think step by step, explore different approaches, and don't be afraid to backtrack or reconsider if something doesn't work out:` `Please reason step by step, and put your final answer within \boxed{}.`

Teacher, SDFT style (SDFT, S^3-OPSD)	`Problem: {question}`
	`This is an example for a response to the question:`
	`{reference solution}`
	`Now answer with a response of your own, including the thinking process:`
	`Please reason step by step, and put your final answer within \boxed{}.`

Teacher, SDPO	`Problem: {question}`
	`Correct solution:`
	`{rollout}` (the student's own rollout if correct, otherwise the first correct rollout of its group; omitted if none is correct)
	`The following is feedback from your unsuccessful earlier attempt:`
	`The final boxed answer was incorrect.` (only if the student's rollout is incorrect)
	`Correctly solve the original question.`
	`Please reason step by step, and put your final answer within \boxed{}.`

maximum of 1,024 new tokens. All methods use a constant learning rate after a 10-step linear warm-up.

Table A3 specifies the objective optimized by each method and, where applicable, the teacher and its privileged context. GRPO uses verified rewards without a distillation teacher. The distillation methods use Jensen–Shannon divergence, forward KL, or reverse KL, with teachers constructed either from the fixed initial policy or an exponential moving average of the student. Their privileged contexts include reference solutions, correct group rollouts, feedback, and teacher-generated prefixes, depending on the method. For S^3-OPSD, we use a forward-KL objective and an EMA teacher with an update rate of 5×10^{-4}, providing the reference solution as an example.

Table A4 reports the optimizer, learning rate, gradient clipping norm, objective cap, and additional method-specific settings. The baselines use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a learning rate of 5×10^{-6}, reduced to 10^{-6} for Gemma4. In contrast, S^3-OPSD uses SGD without momentum at a learning rate of 10^{-2}. We use a gradient clipping norm of 0.1 and a per-entry objective cap of 0.05 for SDPO, SDFT, OPSD, and S^3-OPSD. GRPO, TIP, and TrOPD use a gradient clipping norm of 1.0 without this cap. For S^3-OPSD, the block size for ours is $B = 256$.

Table A2: **Shared training settings.** All methods use LoRA adapters and generate 32 rollouts per step, with method-specific prompt and rollout counts for mathematical reasoning. Sampling configurations, numerical precision, and the learning-rate schedule are shared across methods.

Setting	Value
Adapter	LoRA, $r = 64$
Precision	bf16 weights; fp32 adapter and optimizer state
Prompts per step	32 (SDPO 8, GRPO 4), one rollout per prompt (SDPO 4, GRPO 8)
Rollout sampling	non-thinking mode; temperature 1.1, top-p 0.95, top-k 20
Rollout length	at most 1,024 new tokens
Learning-rate schedule	constant after a linear warm-up of 10 steps

Table A3: **Method-specific objectives and teacher configurations.** We summarize the training objective, teacher construction, and privileged context used by each method. EMA denotes an exponential moving average of the student parameters, with the corresponding update rate reported in the table. Initial policy denotes a teacher fixed at the initial policy checkpoint. Privileged context refers to additional information provided to the teacher when constructing the distillation target.

Method	**Objective**	**Teacher**	**Privileged context**
GRPO	Policy gradient with verified reward	None	None
SDPO	Jensen–Shannon divergence	EMA, rate 0.05	A correct rollout from the group and feedback
SDFT	Forward KL	EMA, rate 0.01	Reference solution as an example
OPSD	Forward KL	Initial policy	Reference solution and re-derive prompt
TIP	Reverse KL	Initial policy	Reference solution and re-derive prompt
TrOPD	Reverse KL	Initial policy	Reference solution, re-derive prompt, and prefix from teacher rollouts
S^3-OPSD	Forward KL	EMA, rate 5×10^{-4}	Reference solution and re-derive prompt

Loss for S^3-OPSD. The teacher is an EMA of the student, taken over the adapter parameters, with update rate 5×10^{-4}; it is conditioned on the reference solution with the SDFT-style teacher prompt. Its next-token distribution is mixed with the initial policy at $\beta = 0.15$. Theis mixed distribution q_t serves as the distillation target and enters the score S_t. The score is computed over the full vocabulary at every rollout position, and one token per block of $B = 256$ tokens is selected. With $\mathcal{S}$ the selected positions of a micro-batch, the trained loss is

$$\mathcal{L} = \frac{1}{|\mathcal{S}|}\sum_{t\in\mathcal{S}}\sum_{a}\min\Big\{q_t(a)\log\frac{q_t(a)}{p_t(a)},\,\tau\Big\},\qquad \tau = 0.05. \tag{A30}$$

That is, the forward KL of Eq. 2 is restricted to the selected positions, with each vocabulary term capped at τ. We adopt this clipping mechanism from OPSD. In the implementation, clipped terms contribute no gradient, so the resulting update retains the policy-gradient interpretation underlying our RL–OPSD connection, with the all-action sum restricted to unclipped actions. This interpretation applies to the clipped update rather than the original, unclipped objective. Following the practical motivation discussed in OPSD, we retain this mechanism to facilitate optimization.

Table A4: **Optimization hyperparameters and additional settings.** We report the optimizer, learning rate (LR), gradient clipping norm, and additional configurations for each method. Cap denotes the per-entry cap applied to the distillation objective. AdamW uses $\beta_1 = 0.9$ and $\beta_2 = 0.999$, whereas SGD uses no momentum. For Gemma4, the AdamW learning rate is 10^{-6}. B denotes the block size used by S^3-OPSD. A dash indicates that the corresponding setting is not used or that no additional setting is listed.

Method	Optimizer	LR	Clip norm	Cap	Additional settings
GRPO	AdamW	5×10^{-6}	1.0	–	8 generations per prompt; 2 gradient updates per batch
SDPO	AdamW	5×10^{-6}	0.1	0.05	4 rollouts per prompt
SDFT	AdamW	5×10^{-6}	0.1	0.05	–
OPSD	AdamW	5×10^{-6}	0.1	0.05	–
TIP	AdamW	5×10^{-6}	1.0	–	Entropy and divergence clipped at the 0.98 quantile
TrOPD	AdamW	5×10^{-6}	1.0	–	Imitation weight 0.001; prefix annealed over 100 steps
S^3-OPSD	SGD	10^{-2}	0.1	0.05	$B = 256$

D Additional Results

This section reports the training cost of the methods (Sec. D.1), additional metrics for mathematical reasoning (Sec. D.2), and per-task results of multi-task learning (Sec. D.3). It then compares the baselines with S^3-OPSD at equal training time (Sec. D.5) and shows the tokens that S_t selects (Sec. D.6).

D.1 Computational and memory cost

Hardware and software. All experiments are conducted on a node with eight NVIDIA B300 SXM6 GPUs, where the the VRAM budget is restricted to 80GB. Rollouts are generated by an in-process vLLM engine on the training GPUs, with model weights synchronized before each step. The Qwen3-1.7B, 4B, and 8B runs use one, two, and four GPUs, respectively, while the Gemma4-2B and 4B runs use one and two GPUs.

Training time and memory. As shown in Tab. A5, S^3-OPSD consistently reduces both step time and peak memory. On Qwen3-1.7B, it requires 7.35s and 47.7GB per step, compared with 18.45–22.58s and 63.1–65.8GB for the baselines. The corresponding results are 14.79s and 43.6GB on Qwen3-4B, and 27.70s and 51.4GB on Qwen3-8B, yielding speedups of 1.8–2.0$\times$ and 2.3–2.7$\times$, respectively. Similar trends hold for Gemma4.

FLOPs. Tab. A6 reports the FLOPs of 500 training steps on Qwen3-1.7B using the observed token lengths. S^3-OPSD requires 4.52×10^{17} FLOPs, which is comparable to the distillation baselines (4.44–4.80×10^{17}). Its additional scoring passes account for 0.51×10^{17} FLOPs, while its training passes require 1.41×10^{17} FLOPs. In contrast, GRPO requires 17.83×10^{17} FLOPs, or 3.94$\times$ the cost of S^3-OPSD.

Micro-batch scaling. Tab. A7 varies the number of rollouts processed together on Qwen3-1.7B without an explicit memory limit. Increasing the micro-batch size reduces baseline step time from 18.04–20.49s with four rollouts to 7.12–8.31s with 16 rollouts, but increases peak memory from 63.1–65.7GB to 185.7–231.4GB. In particular, all baselines exceed 80GB with eight or more rollouts per micro-batch. By contrast, S^3-OPSD processes all 32 rollouts in one micro-batch using 47.7GB and 7.35s per step, demonstrating substantially better memory scaling.

Table A5: **Training cost for mathematical reasoning.** GPUs and rollouts per micro-batch of each run; peak GPU memory; time per step; and GPU hours for 500 steps. Qwen3 values and S^3-OPSD on Gemma4 come from eight-step timing runs (Sec. D.1). GRPO generates the 32 rollouts of a step together and trains on one per micro-batch.

Method	GPUs	Rollouts per micro-batch	Peak memory (GB)	Time per step (s)	500 steps (Wall-clock hours)
Qwen3-1.7B					
SDPO	1	4	63.1	22.58	3.14
SDFT	1	4	65.7	19.48	2.71
OPSD	1	4	65.7	20.39	2.83
TIP	1	4	65.8	20.59	2.86
TrOPD	1	4	63.2	18.45	2.56
S^3-OPSD	1	32	47.7	7.35	1.02
Qwen3-4B					
SDPO	2	2	63.1	29.86	4.21
SDFT	2	2	61.7	28.87	4.01
OPSD	2	2	61.4	28.18	3.91
TIP	2	2	77.7	27.01	3.75
TrOPD	2	2	72.0	27.42	3.76
S^3-OPSD	2	16	43.6	14.79	2.05
Qwen3-8B					
SDPO	4	1	60.0	64.74	9.23
SDFT	4	1	66.0	69.56	9.66
OPSD	4	1	65.4	73.66	10.23
TIP	4	1	64.9	64.80	9.00
TrOPD	4	1	69.2	64.78	9.40
S^3-OPSD	4	8	51.4	27.70	3.85
Gemma4-2B					
SDPO	1	8	60.2	20.25	2.87
SDFT	1	8	73.9	23.41	3.31
OPSD	1	8	75.9	22.88	3.24
TIP	1	8	75.5	22.94	3.22
TrOPD	1	8	73.2	23.57	3.27
S^3-OPSD	1	16	50.32	8.76	1.22
Gemma4-4B					
SDPO	2	4	69.2	25.89	3.61
SDFT	2	4	73.3	31.97	4.55
OPSD	2	4	74.2	30.77	4.27
TIP	2	4	73.7	29.75	4.25
TrOPD	2	4	68.3	30.66	4.38
S^3-OPSD	2	8	52.31	12.67	1.76

Table A6: **FLOPs of 500 training steps on Qwen3-1.7B** ($\times 10^{17}$). Rollout generation, teacher forward passes, the scoring passes of S^3-OPSD over the student context (student and reference model), and training passes. The last column divides each total by that of S^3-OPSD.

Method	Rollout	Teacher	Scoring	Training	Total	Ratio
GRPO	2.13	2.20	–	13.50	17.83	3.94
SDPO	0.61	1.95	–	1.88	4.44	0.98
SDFT	0.58	2.43	–	1.79	4.80	1.06
OPSD	0.56	2.36	–	1.71	4.63	1.02
TIP	0.53	2.33	–	1.62	4.48	0.99
TrOPD	0.57	2.37	–	1.74	4.68	1.04
S^3-OPSD	0.55	2.05	0.51	1.41	4.52	1.00

Table A7: **Memory and time without a memory limit on Qwen3-1.7B.** Peak GPU memory (GB) and time per step (s) on one GPU when the 32 rollouts of a step are generated and trained in micro-batches of 4, 8, 16, or 32 rollouts. Memory above 80 GB is shown in gray.

Method	4 rollouts		8 rollouts		16 rollouts		32 rollouts	
	Memory	Time	Memory	Time	Memory	Time	Memory	Time
SDPO	63.1	18.04	103.1	11.29	185.7	7.12	-	-
SDFT	65.7	19.48	111.2	12.09	222.5	7.99	-	-
OPSD	65.7	20.39	111.6	11.55	223.2	7.44	-	-
TIP	65.7	20.01	111.6	11.79	223.3	7.59	-	-
TrOPD	63.2	20.49	115.7	11.86	231.4	8.31	-	-
S^3-OPSD	–	–	–	–	–	–	47.7	7.35

D.2 Additional quantitative results for mathematical reasoning

Tab. A8 and Tab. A9 report Pass@12 and Maj@12 for mathematical reasoning, computed from the 12 solutions per problem generated with the evaluation protocol of Sec. C.1. Pass@12 is the percentage of problems with at least one correct solution among the 12. Maj@12 is the percentage of problems whose most frequent final answer is correct.

Both metrics follow the trends of Avg@12 in Tab. 1. The methods whose Avg@12 declines under extended training also lose Pass@12 and Maj@12. For example, SDFT solves no problem on the three benchmarks at step 500 for all Qwen3 models, and the average Pass@12 of SDPO on Qwen3-1.7B falls from 67.78 at step 100 to 43.33 at step 500. The average Pass@12 and Maj@12 of S^3-OPSD shows the overall highest performance for almost models and training steps

Table A8: **Pass@12 on mathematical reasoning.** Percentage of problems solved by at least one of 12 samples, for the checkpoints of Tab. 1. Per model and column, the best result is **bold** and the second best underlined; the base model is not ranked.

Method	AIME24			AIME25			HMMT25			Average		
	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500
Qwen3-1.7B												
Base (Instruct)		76.67			56.67			46.67			60.00	
+ GRPO	**80.00**	80.00	76.67	**70.00**	60.00	60.00	46.67	43.33	**53.33**	65.56	61.11	63.33
+ SDPO	76.67	73.33	56.67	63.33	60.00	46.67	**63.33**	46.67	26.67	**67.78**	60.00	43.34
+ SDFT	**80.00**	56.67	0.00	66.67	46.67	0.00	53.33	30.00	0.00	66.67	44.45	0.00
+ OPSD	76.67	73.33	76.67	60.00	**66.67**	50.00	46.67	50.00	40.00	61.11	63.33	55.56
+ TIP	**80.00**	**83.33**	76.67	66.67	53.33	56.67	50.00	50.00	46.67	65.56	62.22	60.00
+ TrOPD	**80.00**	80.00	76.67	60.00	60.00	50.00	40.00	40.00	40.00	60.00	60.00	55.56
+ S^3**-OPSD**	76.67	80.00	**80.00**	66.67	**66.67**	**66.67**	56.67	**56.67**	**53.33**	66.67	**67.78**	**66.67**
Qwen3-4B												
Base (Instruct)		86.67			86.67			66.67			80.00	
+ GRPO	83.33	**83.33**	83.33	**86.67**	83.33	**86.67**	**70.00**	63.33	**66.67**	80.00	76.66	**78.89**
+ SDPO	**90.00**	80.00	63.33	83.33	83.33	53.33	63.33	50.00	43.33	78.89	71.11	53.33
+ SDFT	83.33	53.33	0.00	**86.67**	53.33	0.00	66.67	33.33	0.00	78.89	46.66	0.00
+ OPSD	83.33	**83.33**	80.00	83.33	76.67	70.00	63.33	**66.67**	53.33	76.66	75.56	67.78
+ TIP	83.33	**83.33**	83.33	83.33	76.67	73.33	56.67	60.00	60.00	74.44	73.33	72.22
+ TrOPD	86.67	**83.33**	83.33	80.00	83.33	76.67	63.33	63.33	63.33	76.67	76.66	74.44
+ S^3**-OPSD**	86.67	**83.33**	**90.00**	**86.67**	**86.67**	**86.67**	**70.00**	63.33	60.00	**81.11**	**77.78**	**78.89**
Qwen3-8B												
Base (Instruct)		86.67			83.33			60.00			76.67	
+ GRPO	83.33	86.67	86.67	80.00	**83.33**	83.33	63.33	70.00	66.67	75.55	80.00	78.89
+ SDPO	83.33	86.67	80.00	83.33	**83.33**	83.33	63.33	70.00	**73.33**	76.66	80.00	78.89
+ SDFT	**93.33**	46.67	0.00	83.33	23.33	0.00	**73.33**	36.67	0.00	83.33	35.56	0.00
+ OPSD	86.67	86.67	76.67	**86.67**	**83.33**	66.67	63.33	66.67	53.33	78.89	78.89	65.56
+ TIP	90.00	90.00	86.67	80.00	80.00	83.33	63.33	**73.33**	**73.33**	77.78	81.11	81.11
+ TrOPD	86.67	**93.33**	86.67	**86.67**	**83.33**	**90.00**	60.00	66.67	66.67	77.78	81.11	81.11
+ S^3**-OPSD**	**93.33**	90.00	**90.00**	**86.67**	**83.33**	**90.00**	**73.33**	**73.33**	70.00	**84.44**	**82.22**	**83.33**
Gemma4-2B												
Base (Instruct)		70.00			60.00			43.33			57.78	
+ SDPO	63.33	56.67	56.67	60.00	53.33	43.33	**46.67**	36.67	30.00	56.67	48.89	43.33
+ SDFT	**73.33**	63.33	30.00	56.67	46.67	20.00	36.67	20.00	10.00	55.56	43.33	20.00
+ OPSD	70.00	63.33	63.33	53.33	50.00	60.00	36.67	43.33	40.00	53.33	52.22	54.44
+ TIP	63.33	66.67	**70.00**	53.33	56.67	60.00	36.67	**46.67**	36.67	51.11	56.67	55.56
+ TrOPD	66.67	66.67	63.33	56.67	53.33	**63.33**	40.00	40.00	40.00	54.45	53.33	55.55
+ S^3**-OPSD**	**73.33**	**73.33**	**70.00**	**63.33**	**60.00**	60.00	**46.67**	43.33	**43.33**	**61.11**	**58.89**	**57.78**
Gemma4-4B												
Base (Instruct)		73.33			66.67			50.00			63.33	
+ SDPO	**80.00**	**80.00**	73.33	63.33	**76.67**	70.00	**56.67**	53.33	**56.67**	66.67	70.00	66.67
+ SDFT	**80.00**	66.67	56.67	60.00	63.33	46.67	43.33	50.00	30.00	61.11	60.00	44.45
+ OPSD	70.00	73.33	76.67	**73.33**	63.33	60.00	46.67	50.00	**56.67**	63.33	62.22	64.45
+ TIP	76.67	73.33	70.00	63.33	66.67	63.33	**56.67**	50.00	**56.67**	65.56	63.33	63.33
+ TrOPD	73.33	76.67	76.67	63.33	63.33	**76.67**	**56.67**	53.33	53.33	64.44	64.44	68.89
+ S^3**-OPSD**	**80.00**	**80.00**	**83.33**	70.00	**76.67**	73.33	53.33	**56.67**	**56.67**	**67.78**	**71.11**	**71.11**

Table A9: **Maj@12 on mathematical reasoning.** Percentage of problems whose most frequent answer among 12 samples is correct, for the checkpoints of Tab. 1. Equivalent answers are merged, and a tie among m answers counts $1/m$. Marks as in Tab. A8.

Method	AIME24			AIME25			HMMT25			Average		
	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500
Qwen3-1.7B												
Base (Instruct)		67.78			41.67			26.67			45.37	
+ GRPO	63.33	66.67	65.00	51.67	50.00	**50.00**	28.33	26.94	29.44	47.78	47.87	48.15
+ SDPO	66.67	59.44	38.06	**55.00**	43.06	15.28	27.14	20.00	16.67	49.60	40.83	23.34
+ SDFT	66.67	43.33	0.00	48.06	26.67	0.00	**35.00**	21.67	0.00	49.91	30.56	0.00
+ OPSD	66.67	66.67	51.67	48.61	42.22	38.61	31.67	26.67	28.33	48.98	45.19	39.54
+ TIP	**68.33**	64.44	58.33	43.33	45.00	38.06	28.33	25.00	26.67	46.66	44.81	41.02
+ TrOPD	63.33	56.67	60.28	50.28	36.94	40.00	21.11	20.00	20.00	44.91	37.87	40.09
+ S^3**-OPSD**	**68.33**	**70.00**	**71.67**	**55.00**	**53.33**	48.61	33.33	**31.67**	**31.11**	**52.22**	**51.67**	**50.46**
Qwen3-4B												
Base (Instruct)		80.00			75.48			55.28			70.25	
+ GRPO	78.33	80.00	**80.00**	78.06	73.33	**76.67**	53.33	48.33	50.00	69.91	67.22	68.89
+ SDPO	**80.00**	75.00	57.78	73.33	70.00	40.67	**56.67**	43.70	32.22	70.00	62.90	43.56
+ SDFT	**80.00**	42.78	0.00	78.33	31.78	0.00	50.83	25.00	0.00	69.72	33.19	0.00
+ OPSD	**80.00**	78.33	67.78	**80.00**	68.33	56.67	50.00	40.00	40.00	70.00	62.22	54.82
+ TIP	**80.00**	**83.33**	**80.00**	73.33	70.00	62.78	45.00	44.44	45.00	66.11	65.92	62.59
+ TrOPD	**80.00**	80.00	75.00	71.67	69.44	67.78	46.67	**55.00**	43.33	66.11	68.15	62.04
+ S^3**-OPSD**	**80.00**	80.00	**80.00**	76.97	**76.97**	**76.67**	**56.67**	**55.00**	**51.67**	**71.21**	**70.66**	**69.45**
Qwen3-8B												
Base (Instruct)		81.67			76.67			53.33			70.56	
+ GRPO	80.00	80.00	80.00	76.67	76.67	**77.08**	52.50	58.61	52.78	69.72	71.76	69.95
+ SDPO	80.00	80.00	80.00	77.00	75.00	70.00	51.11	50.00	35.00	69.37	68.33	61.67
+ SDFT	**81.67**	26.92	0.00	76.67	12.50	0.00	60.00	22.78	0.00	72.78	20.73	0.00
+ OPSD	**81.67**	**86.67**	65.11	**80.00**	72.78	45.72	56.67	48.33	47.22	72.78	69.26	52.68
+ TIP	80.00	**86.67**	81.67	75.00	73.33	70.00	57.14	56.67	45.00	70.71	72.22	65.56
+ TrOPD	80.00	84.44	80.00	75.83	74.44	71.39	53.33	51.67	51.11	69.72	70.18	67.50
+ S^3**-OPSD**	**81.67**	81.67	**83.33**	**80.00**	**80.48**	76.67	**61.67**	**60.00**	**63.33**	**74.45**	**74.05**	**74.44**
Gemma4-2B												
Base (Instruct)		56.67			45.00			27.78			43.15	
+ SDPO	56.11	48.33	45.33	45.00	38.33	37.78	28.33	23.33	17.08	43.15	36.66	33.40
+ SDFT	55.00	41.67	18.33	**46.67**	33.33	15.00	21.11	13.89	3.61	40.93	29.63	12.31
+ OPSD	**63.33**	56.67	61.67	43.33	43.33	47.08	18.33	26.94	23.33	41.66	42.31	44.03
+ TIP	53.33	58.33	60.00	37.08	35.02	40.42	25.00	27.70	20.00	38.47	40.35	40.14
+ TrOPD	55.56	55.00	61.67	46.11	41.67	45.37	20.00	16.67	20.33	40.56	37.78	42.46
+ S^3**-OPSD**	**63.33**	**61.67**	**63.33**	**46.67**	**46.11**	**48.15**	**30.00**	**29.94**	**26.67**	**46.67**	**45.91**	**46.05**
Gemma4-4B												
Base (Instruct)		68.33			50.00			31.11			49.81	
+ SDPO	**66.67**	63.61	60.00	54.17	**58.33**	53.33	36.67	39.17	29.44	52.50	53.70	47.59
+ SDFT	63.61	58.33	43.33	50.00	51.67	43.33	26.67	28.33	13.33	46.76	46.11	33.33
+ OPSD	63.33	56.67	58.64	53.33	53.33	**56.67**	35.00	31.67	31.11	50.55	47.22	48.81
+ TIP	**66.67**	**66.67**	65.00	53.33	46.67	46.94	41.67	31.67	31.94	53.89	48.34	47.96
+ TrOPD	**66.67**	63.33	63.33	50.00	50.83	47.22	35.28	30.00	35.00	50.65	48.05	48.52
+ S^3**-OPSD**	63.30	**66.67**	**68.33**	**58.33**	55.00	**56.67**	**43.33**	**41.67**	**37.08**	**54.99**	**54.45**	**54.03**

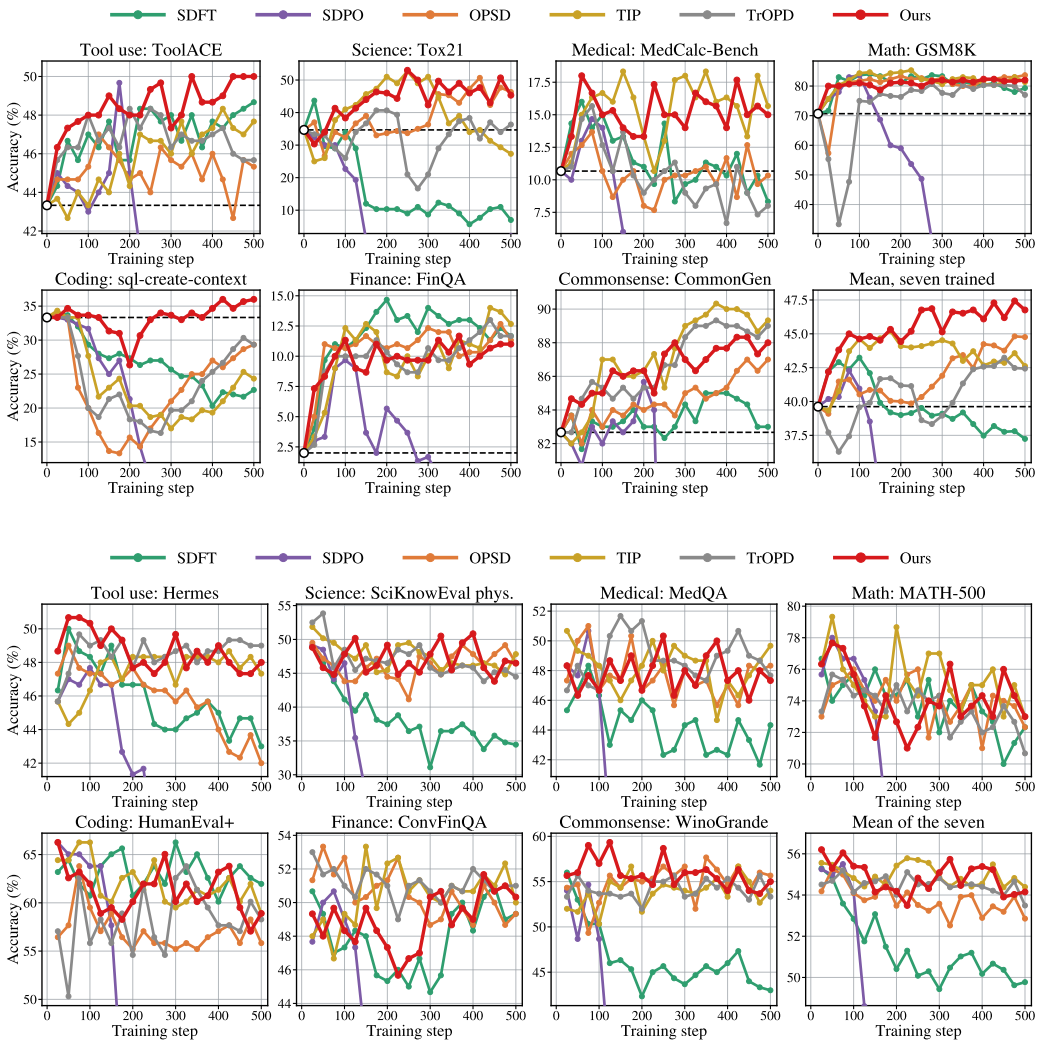


Figure A1: **Accuracy on seen tasks and unseen benchmarks during multi-task learning.** Qwen3-1.7B is evaluated every 25 steps. **Top:** Evaluation splits of the seven training datasets; the dashed line and open circle mark the base model. **Bottom:** Seven unseen benchmarks, one per domain, evaluated from step 25; base-model values are reported in Tab. A10. The last panel in each row shows the mean over the seven tasks or benchmarks.

D.3 Per-task results of multi-task learning

Fig. 4 summarizes the mean accuracy on the seven seen tasks and seven unseen benchmarks, while Fig. A1 shows the per-task learning curves. Tab. A10 reports the corresponding results at step 500. All methods fully fine-tune Qwen3-1.7B for 500 steps under the same multi-task setting. For the seen tasks, S^3-OPSD achieves the highest mean accuracy and improves over the base model on every seen task. Although several baselines perform well on individual domains, they often degrade substantially on others. These results indicate that S^3-OPSD provides more balanced adaptation across heterogeneous tasks. For unseen benchmarks, S^3-OPSD largely preserves the base model's average accuracy and remains among the best-performing methods on unseen benchmarks. In contrast, several baselines exhibit considerably greater degradation, suggesting stronger catastrophic forgetting. Overall, S^3-OPSD improves the training tasks while retaining broad generalization to unseen domains.

Table A10: **Per-task accuracy at step 500 of multi-task learning.** Top: evaluation splits of the seven training datasets. Bottom: one unseen benchmark per domain. Best result in each column is **bold** and second best is underlined; the base model is not ranked.

Seen	ToolACE	Tox21	MedCalc-Bench	GSM8K	SQL	FinQA	CommonGen	Mean
Base (Instruct)	43.33	34.67	10.67	70.67	33.33	2.00	82.67	39.62
+ SDPO	0.00	2.00	3.00	7.67	0.00	0.00	0.00	1.81
+ SDFT	48.67	7.00	8.33	79.33	22.67	11.67	83.00	37.24
+ OPSD	45.33	**46.33**	10.33	**83.67**	29.33	11.33	87.00	44.76
+ TIP	47.67	27.33	**15.67**	81.33	24.33	**12.67**	**89.33**	42.62
+ TrOPD	45.67	36.33	8.00	77.00	29.33	11.67	89.00	42.43
+ S^3**-OPSD**	**50.00**	45.33	15.00	82.00	36.00	11.00	88.00	**46.76**
Unseen	**Hermes FC**	**SciKnowEval phys.**	**MedQA**	**MATH-500**	**HumanEval+**	**ConvFinQA**	**WinoGrande**	**Mean**
Base (Instruct)	44.33	49.50	48.00	72.33	66.26	50.67	55.00	55.16
+ SDPO	0.00	0.00	0.00	0.00	0.00	21.33	0.00	3.05
+ SDFT	43.00	34.45	44.33	72.33	**61.96**	49.33	43.00	49.77
+ OPSD	42.00	46.49	48.33	72.33	55.83	49.33	**55.67**	52.85
+ TIP	47.33	**47.83**	**49.67**	**73.00**	58.90	50.00	54.00	**54.39**
+ TrOPD	**49.00**	44.48	47.67	70.67	58.28	**51.00**	53.33	53.49
+ S^3**-OPSD**	48.00	46.49	47.33	**73.00**	58.90	50.33	55.00	54.15

D.4 Matched-Time Performance

To account for differences in computational cost, we additionally compare methods at matched wall-clock training time. For each model, we use the training times of S^3-OPSD at steps 100, 300, and 500 as reference points (t_{100}, t_{300}, and t_{500}). Then, we select the checkpoint reached at the closest corresponding time using the per-step costs in Tab. A5, while keeping S^3-OPSD at its original checkpoints. As shown in Tab. A11, S^3-OPSD achieves the strongest or comparable performance across most matched-time settings. In particular, its advantage becomes clearer at longer training budgets: on Qwen3-4B, it reaches average accuracies of 61.39, 61.67, and 62.64 at t_{100}, t_{300}, and t_{500}, compared with the strongest baseline results of 59.73, 57.50, and 56.81, respectively, while on Qwen3-8B it achieves 63.20, 63.06, and 63.06 and remains ahead of the best matched-time baseline at all three reference points. Similar trends are observed on Gemma4-2B and Gemma4-4B. These results show that the gains of S^3-OPSD are not limited to equal-step comparisons, but also persist under a fixed wall-clock budget due to its lower per-step cost and stable optimization.

Table A11: **Accuracy at matched training time.** Baseline checkpoints are matched to the estimated training times of S^3-OPSD at steps 100, 300, and 500 using the per-step times in Tab. A5, rounded to the nearest multiple of 50. Steps lists the matched checkpoints; – denotes unavailable results or timing information. Best and second-best available post-training results are bold and underlined, respectively, including ties.

Method	Steps	AIME24			HMMT25			Average		
		t_{100}	t_{300}	t_{500}	t_{100}	t_{300}	t_{500}	t_{100}	t_{300}	t_{500}
Qwen3-1.7B										
Base (Instruct)	0		46.94			23.33			35.14	
+ SDPO	50/100/150	52.50	53.89	50.28	24.44	24.72	23.06	38.47	39.31	36.67
+ SDFT	50/100/200	53.33	54.44	44.44	26.11	**26.94**	21.11	39.72	40.69	32.78
+ OPSD	50/100/200	**55.28**	54.44	46.39	25.56	25.28	24.17	40.42	39.86	35.28
+ TIP	50/100/200	50.83	51.39	47.22	24.72	24.17	22.50	37.78	37.78	34.86
+ TrOPD	50/100/200	43.61	44.17	43.06	22.22	21.94	20.28	32.92	33.06	31.67
+ **S^3-OPSD**	100/300/500	54.72	**55.00**	**54.17**	**26.67**	26.39	**25.56**	**40.70**	**40.70**	**39.87**
Qwen3-4B										
Base (Instruct)	0		71.67			44.17			57.92	
+ SDPO	50/150/250	73.06	72.22	66.11	43.33	42.78	38.89	58.20	57.50	52.50
+ SDFT	50/150/250	74.17	70.28	35.56	45.28	41.94	22.78	59.73	56.11	29.17
+ OPSD	50/150/250	71.94	70.56	65.83	40.83	41.11	39.44	56.39	55.84	52.64
+ TIP	50/150/250	72.22	73.06	71.67	39.44	40.00	39.44	55.83	56.53	55.56
+ TrOPD	50/150/250	72.78	73.33	72.22	40.56	41.67	41.39	56.67	57.50	56.81
+ **S^3-OPSD**	100/300/500	**76.94**	**77.22**	**77.22**	**45.83**	**46.11**	**48.06**	**61.39**	**61.67**	**62.64**
Qwen3-8B										
Base (Instruct)	0		77.50			42.22			59.86	
+ SDPO	50/150/200	74.17	74.44	74.72	42.22	41.94	41.67	58.20	58.19	58.20
+ SDFT	50/100/200	77.22	77.50	55.56	47.22	**48.33**	30.56	62.22	62.92	43.06
+ OPSD	50/100/200	78.06	**78.61**	74.17	43.89	44.44	41.67	60.98	61.53	57.92
+ TIP	50/150/200	75.56	76.11	75.56	43.61	44.17	44.44	59.59	60.14	60.00
+ TrOPD	50/150/200	77.22	77.22	76.67	41.39	42.22	42.50	59.31	59.72	59.59
+ **S^3-OPSD**	100/300/500	**78.33**	77.78	**78.33**	**48.06**	**48.33**	**47.78**	**63.20**	**63.06**	**63.06**
Gemma4-2B										
Base (Instruct)	0		41.94			18.89			30.42	
+ SDPO	50/150/200	43.61	43.06	40.56	19.44	19.17	17.78	31.53	31.12	29.17
+ SDFT	50/100/200	**44.44**	**45.28**	40.28	19.17	18.61	15.28	31.81	31.95	27.78
+ OPSD	50/100/200	43.33	43.89	42.78	17.22	16.67	18.61	30.28	30.28	30.70
+ TIP	50/100/200	42.22	42.50	43.06	19.17	19.44	18.33	30.70	30.97	30.70
+ TrOPD	50/100/200	42.50	43.06	41.67	18.33	17.78	15.56	30.42	30.42	28.62
+ **S^3-OPSD**	100/300/500	44.17	43.89	**43.33**	**20.00**	**20.56**	**19.44**	**32.09**	**32.23**	**31.39**
Gemma4-4B										
Base (Instruct)	0		52.22			24.17			38.20	
+ SDPO	50/150/250	53.06	53.89	53.06	25.83	26.67	26.11	39.45	40.28	39.59
+ SDFT	50/100/200	52.78	53.06	52.22	24.72	24.72	23.06	38.75	38.89	37.64
+ OPSD	50/100/200	52.22	52.50	52.22	25.56	26.11	25.56	38.89	39.31	38.89
+ TIP	50/150/200	54.44	**55.56**	54.72	27.78	27.78	27.22	41.11	41.67	40.97
+ TrOPD	50/100/200	54.17	54.72	53.61	28.89	**29.72**	27.22	41.53	**42.22**	40.42
+ **S^3-OPSD**	100/300/500	**55.00**	54.72	**55.28**	**29.44**	28.33	**28.61**	**42.22**	41.53	**41.95**

Table A12: **Quantitative comparison on Qwen3-1.7B under matched batch size.** All methods are trained on OpenThoughts-30K for 500 steps, with Avg@12 reported at the 100-, 300-, and 500-step checkpoints. Best performance is marked with **bold** and second-best performance is marked with underline. Ties are marked equally.

Method	AIME24			AIME25			HMMT25			Average		
	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500	step 100	step 300	step 500
Qwen3-1.7B												
Base (Instruct)		46.94			34.17			23.33			34.81	
+ GRPO	45.28	43.06	46.11	33.89	34.72	33.06	22.78	18.33	19.17	33.98	32.04	32.78
+ SDPO	53.33	40.28	24.17	35.83	34.17	12.22	23.61	19.44	11.39	**37.59**	31.30	15.93
+ SDFT	52.22	29.44	0.00	**36.11**	17.22	0.00	**24.17**	11.67	0.00	37.50	19.44	0.00
+ OPSD	**55.00**	40.56	38.61	33.89	30.83	23.89	23.89	21.11	18.33	**37.59**	30.83	26.94
+ TIP	49.17	41.39	36.94	33.06	31.67	29.72	20.28	18.89	16.39	34.17	30.65	27.68
+ TrOPD	44.44	39.72	35.00	31.39	27.22	25.83	20.00	15.28	15.28	31.94	27.41	25.37
+ S^3**-OPSD**	53.89	**53.61**	**52.22**	35.00	**37.78**	**35.83**	23.89	**23.33**	**23.06**	**37.59**	**38.24**	**37.04**

D.5 Matched Micro-Batch Size Comparison

To further control for differences in training configuration, we compare all methods on Qwen3-1.7B under the same batch size of 4 to match micro-batch so that all method can exhibit similar wall-clock time for same steps. All methods are trained on OpenThoughts-30K for 500 steps and evaluated at steps 100, 300, and 500. As shown in Tab. A12, several baselines achieve competitive performance early in training but degrade substantially as optimization proceeds, which is a consistent trench from previous experiments. Meanwhile, S^3-OPSD maintains consistently strong performance throughout training, achieving average scores of 37.59, 38.24, and 37.04 at steps 100, 300, and 500, respectively, and obtaining the best performance at the latter two checkpoints.

D.6 Additional qualitative analysis

We provide visualizations of the S_t values assigned to each token and indicate which tokens were selected during rollouts generated by the actual student model. Since most tokens have small S_t values, we display only tokens whose values exceed a specified threshold. The visualizations are obtained from the first iteration of mathematical reasoning with Qwen3-1.7B and are presented in Fig. A2, Fig. A3, Fig. A4, Fig. A5, Fig. A6, Fig. A7, Fig. A8, Fig. A9, Fig. A10, Fig. A11, Fig. A12, and Fig. A13.

E Limitations

Interpreting the teacher as a reward-guided Doob-h transform of the student provides useful intuition and motivates metrics with desirable properties. However, directly verifying this interpretation is infeasible. Although an implicit reward function consistent with the transform exists under the bounded-reward assumption, this existence result alone does not establish alignment with the intended reward. As discussed in the main manuscript, the teacher's higher probability of producing correct answers when given privileged context provides indirect support for this alignment, but does not establish the proposed interpretation. Developing tractable empirical tests and stronger theoretical characterizations of when this interpretation holds is an important direction for future work.

Problem Let \(x, y, z \) be the roots of the equation \(t^3 - 2t^2 - 9t - 1 = 0 \). Find \(\frac{yz}{x} + \frac{xz}{y} + \frac{xy}{z} \).

Answer reference **77** rollout **77** ✔ correct

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **722 tokens**

```
We are given the cubic equation:
$$
t^3 - 2t^2 - 9t - 1 = 0
$$
Let $ x, y, z $ be the roots of this
  equation. Then by **Vieta's
  formulas**, we know:
- $ x + y + z = 2 $
- $ xy + yz + zx = -9 $
- $ xyz = 1 $
We are asked to compute:
$$
\frac{yz}{x} + \frac{xz}{y} +
  \frac{xy}{z}
$$
---
### Step 1: Write the expression to
  evaluate
We want to compute:
$$
\frac{yz}{x} + \frac{xz}{y} +
  \frac{xy}{z}
$$
Let's write this as:
$$
\frac{yz}{x} + \frac{xz}{y} +
  \frac{xy}{z} = \frac{y z}{x} +
  \frac{x z}{y} + \frac{x y}{z}
$$
---
### Step 2: Find a common denominator
The common denominator is $ xyz $, so
  we rewrite each term:
$$
\frac{yz}{x} = \frac{y z \cdot y z}{x
  y z} = \frac{y^2 z^2}{xyz}
$$
$$
\frac{xz}{y} = \frac{x z \cdot x z}{x
  y z} = \frac{x^2 z^2}{xyz}
$$
$$
\frac{xy}{z} = \frac{xy \cdot x y}{x y
  z} = \frac{x^2 y^2}{xyz}
$$
So the expression becomes:
$$
\frac{y^2 z^2 + x^2 z^2 + x^2
  y^2}{xyz}
$$
---
### Step 3: Compute the numerator
We know:
- $ x + y + z = 2 $
- $ xy + yz + zx = -9 $
- $ xyz = 1 $
We need to compute:
$$
x^2 y^2 + y^2 z^2 + x^2 z^2
$$
Note that:
$$
x^2 y^2 + y^2 z^2 + x^2 z^2 = (xy + yz
  + zx)^2 - 2xyz(x + y + z)
$$
This is a standard identity:
$$
x^2 y^2 + y^2 z^2 + x^2 z^2 = (xy + yz
  + zx)^2 - 2xyz(x + y + z)
$$
Substitute the known values:
- $ xy + yz + zx = -9 $
- $ xyz = 1 $
- $ x + y + z = 2 $
So:
$$
x^2 y^2 + y^2 z^2 + x^2 z^2 = (-9)^2 -
  2(1)(2) = 81 - 4 = 77
$$
---
### Step 4: Final expression
$$
\frac{y^2 z^2 + x^2 z^2 + x^2
  y^2}{xyz} = \frac{77}{1} = 77
$$
---
### ✅ Final Answer:
$$
\boxed{77}
$$<eos>
```

Figure A2: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem In a circle, a square \(O M K N \) is inscribed such that vertex \(O \) coincides with the center of the circle and vertex \(K \) lies on the circle. Chord \(A B \) of the circle passes through vertex \(M \), and chord \(C D \) passes through vertex \(N \). Prove that \(A M \cdot M B = C N \cdot N D \).

Answer reference **A M \cdot M B = C N \cdot N ...** rollout **AM \cdot MB = CN \cdot ND** ✓ correct

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **501 tokens**

```
We are given a circle with a **square
  $ O M K N $** inscribed such that:
- $ O $ is the **center** of the
  circle,
- $ K $ is a **vertex** on the circle,
- The square is **inscribed** in the
  circle.

We are also told:
- Chord $ AB $ passes through $ M $,
- Chord $ CD $ passes through $ N $.

We are to **prove** that:
$$
AM \cdot MB = CN \cdot ND
$$

---

### Step 1: Use the Power of a Point
  Theorem

The **Power of a Point Theorem**
  states that if a chord passes
  through a point $ P $ inside a
  circle, and intersects the circle at
  two points $ X $ and $ Y $, then:
$$
PX \cdot PY = \text{(power of } P)
$$
If the point lies **on the chord**,
  then the product of the distances
  from the point to the two endpoints
  of the chord is equal for any point
  on the chord.

That is, for a chord $ AB $ passing
  through point $ M $:
$$
AM \cdot MB = \text{(power of } M
  \text{ with respect to the circle)}
$$
```

```
Similarly, for a chord $ CD $ passing
  through $ N $:
$$
CN \cdot ND = \text{(power of } N
  \text{ with respect to the circle)}
$$
Since $ M $ and $ N $ are **points on
  the circle**, their **powers** are
  zero:
$$
AM \cdot MB = CN \cdot ND
$$

---

### Step 2: Interpret the Geometric
  Configuration

- The square $ O M K N $ is inscribed
  in the circle, with $ O $ at the
  center and $ K, M, N $ on the
  circle.
- So, the square is **symmetrical**
  about the center of the circle.
- Therefore, the chord $ AB $, passing
  through $ M $, is **symmetrical** to
  chord $ CD $, passing through $ N $.
- Because the square is symmetric and
  the chords go through symmetric
  points on the circle, the product $
  AM \cdot MB = CN \cdot ND $ must be
  equal.

---

### Final Result

$$
\boxed{AM \cdot MB = CN \cdot ND}
$$
This is the **required identity** that
  the problem asks to prove.<eos>
```

Figure A3: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem From a general population, a sample of size $n=50$ was taken: | variant | x_{i} | 2 | 5 | 7 | 10 | | :--- | :--- | ---: | ---: | ---: | ---: | | frequency | n_{i} | 16 | 12 | 8 | 14 | Find the unbiased estimator of the population mean.

Answer reference **5.76** rollout **5.76** ✔ correct

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **383 tokens**

```
To find the **unbiased estimator** of
  the population mean, we use the
  **sample mean**.

---

### Step 1: Understand the data

We are given a frequency table of a
  sample of size $ n = 50 $. The table
  shows the values of the variant $
  x_i $, and the corresponding
  frequencies $ n_i $, which represent
  how many times each $ x_i $ appears
  in the sample.

Let's tabulate the data:

| variant | $ x_i $ | $ n_i $ |
|---------|-----------|-----------|
| 2       | 16        | 16        |
| 5       | 12        | 12        |
| 7       | 8         | 8         |
| 10      | 14        | 14        |

---

### Step 2: Compute the sample mean

The **unbiased estimator** of the
  population mean is the **sample
  mean**, which is calculated as:

$$
\bar{x} = \frac{\sum (x_i \cdot
  n_i)}{n}
$$

Now compute the numerator:

$$
\sum (x_i \cdot n_i) = (2 \cdot 16) +
  (5 \cdot 12) + (7 \cdot 8) + (10
  \cdot 14)
$$

$$
= 32 + 60 + 56 + 140 = 288
$$

Then divide by the sample size $ n =
  50 $:

$$
\bar{x} = \frac{288}{50} = 5.76
$$

---

### ☑ Final Answer:

$$
\boxed{5.76}
$$

This is the **unbiased estimator** of
  the population mean.<eos>
```

Figure A4: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem Given that \(\tan x + \cot x = 8\), find the value of \(\sqrt{\sec ^{2} x + \csc ^{2} x - \frac{1}{2} \sec x \csc x}\).

Answer reference **2\sqrt{15}** rollout **2\sqrt{15}** ✓ correct

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **766 tokens**

```
We are given:
$$
\tan x + \cot x = 8
$$
We are to find the value of:
$$
\sqrt{\sec^2 x + \csc^2 x -
  \frac{1}{2} \sec x \csc x}
$$
---
### Step 1: Use identity to express
  everything in terms of sine and
  cosine
We know:
- $\tan x = \frac{\sin x}{\cos x}$
- $\cot x = \frac{\cos x}{\sin x}$
- $\sec x = \frac{1}{\cos x}$
- $\csc x = \frac{1}{\sin x}$
So:
$$
\tan x + \cot x = \frac{\sin x}{\cos
  x} + \frac{\cos x}{\sin x}
$$
Let's compute that:
$$
\tan x + \cot x = \frac{\sin^2 x +
  \cos^2 x}{\sin x \cos x} =
  \frac{1}{\sin x \cos x}
$$
So:
$$
\frac{1}{\sin x \cos x} = 8
\Rightarrow \sin x \cos x =
  \frac{1}{8}
$$
---
### Step 2: Use identity to compute
  $\sec^2 x + \csc^2 x$
We know:
- $\sec^2 x = \frac{1}{\cos^2 x}$
- $\csc^2 x = \frac{1}{\sin^2 x}$
So:
$$
\sec^2 x + \csc^2 x = \frac{1}{\cos^2
  x} + \frac{1}{\sin^2 x}
$$
We can use the identity:
$$
\frac{1}{\cos^2 x} + \frac{1}{\sin^2
  x} = \frac{\sin^2 x + \cos^2
  x}{\sin^2 x \cos^2 x} =
  \frac{1}{\sin^2 x \cos^2 x}
$$
We already know $\sin x \cos x =
  \frac{1}{8}$, so:
$$
\sin^2 x \cos^2 x =
  \left(\frac{1}{8}\right)^2 =
  \frac{1}{64}
$$
Hence:
$$
\sec^2 x + \csc^2 x =
  \frac{1}{\frac{1}{64}} = 64
$$
---
### Step 3: Compute the expression
  under the square root
We are given:
$$
\sqrt{\sec^2 x + \csc^2 x -
  \frac{1}{2} \sec x \csc x}
$$
We already found:
- $\sec^2 x + \csc^2 x = 64$
- $\sec x \csc x = \frac{1}{\sin x
  \cos x} = 8$
So:
$$
\sec^2 x + \csc^2 x - \frac{1}{2} \sec
  x \csc x = 64 - \frac{1}{2}(8) = 64
  - 4 = 60
$$
---
### Step 4: Take the square root
$$
\sqrt{60} = \sqrt{4 \cdot 15} =
  2\sqrt{15}
$$
---
### ✅ Final Answer:
$$
\boxed{2\sqrt{15}}
$$<eos>
```

Figure A5: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem A soccer ball is stitched using black and white pieces of leather. The black pieces are regular pentagons, and the white pieces are regular hexagons, with the side lengths of both shapes being equal. Each black pentagon is stitched to five white hexagons, and each white hexagon has three sides stitched to black pentagons and the other three sides stitched to other white hexagons. If there are 12 black …

Answer reference **20** rollout **20** ✔ correct

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **381 tokens**

We are given the following information about a soccer ball:

- Each **black pentagon** is stitched to **five white hexagons**.
- Each **white hexagon** has **three sides** stitched to **black pentagons** and **three sides** stitched to **white hexagons**.
- There are **12 black pentagons** on the surface.
- All **side lengths** are equal.
- The soccer ball is made of **black pentagons** and **white hexagons**.

We are asked to find the number of **white hexagons**.

Step 1: Use the given relationship between black and white pieces

Each black pentagon is stitched to **5 white hexagons**, and each white hexagon is stitched to **3 black pentagons** and **3 white hexagons**.

Let:
- $ B $ = number of black pentagons = 12
- $ W $ = number of white hexagons = ?

Each black pentagon is connected to **5 white hexagons**, so the total number of pentagon-hexagon connections is:

$$
12 \times 5 = 60
$$

Each white hexagon is connected to **3 black pentagons**, so the total number of pentagon-hexagon connections from white hexagons is:

$$
W \times 3
$$

Since each connection is shared between one pentagon and one hexagon, the total number of connections is the same from both perspectives:

$$
60 = 3W \Rightarrow W = \frac{60}{3} = 20
$$

☑ Final Answer:

$$
\boxed{20}
$$

There are **20 white hexagons** on the surface of the soccer ball.<eos>

Figure A6: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem Find the radius of a circle that is tangent to two concentric circles with radii 3 and 5.

Answer reference **1** rollout **8** ✗ **incorrect**

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **569 tokens**

We are given two **concentric circles** with radii **3** and **5**, and we are to find the **radius of a circle** that is **tangent to both**.

Step 1: Understand the problem

- Two **concentric circles** share the same center.
- One has radius 3, the other 5.
- We need to find the radius of a circle that is **tangent to both**.

There are two possible circles that can be tangent to both concentric circles:

1. A **circle inside** the larger circle (radius 5) and **tangent** to the smaller circle (radius 3).
2. A **circle outside** the smaller circle (radius 3) and tangent to the larger circle (radius 5).

So we need to consider both cases.

Step 2: Case 1 – Circle inside the larger circle and tangent to the smaller one

Let the radius of the inner circle be $ r $.

- It is **inside** the larger circle of radius 5.
- It is **tangent to** the smaller circle of radius 3.
- The distance between the centers is 0 (since they are concentric).
- The distance between the centers is also the difference in radii if the circles are tangent internally.

So, the distance between the centers is:

$$
5 - r = 3
$$

Solving for $ r $:

$$
r = 5 - 3 = 2
$$

Step 3: Case 2 – Circle outside the smaller circle and tangent to the larger one

Let the radius of the outer circle be $ R $.

- It is **outside** the smaller circle of radius 3.
- It is **tangent to** the larger circle of radius 5.
- The distance between the centers is 0 (they are concentric).
- The distance between the centers is also the difference in radii if the circles are tangent externally.

So, the distance between the centers is:

$$
R - 3 = 5
$$

Solving for $ R $:

$$
R = 5 + 3 = 8
$$

Step 4: Final Answer

The possible radii of the circles tangent to both concentric circles are:

- $ \boxed{2} $ for the circle inside the larger circle.
- $ \boxed{8} $ for the circle outside the smaller circle.

So the **radius of a circle tangent to both** is either:

$$
\boxed{2} \quad \text{or} \quad \boxed{8}
$$

If only one answer is required, then **both are valid**, depending on the direction of tangency.<eos>

Figure A7: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem What is the value of $((1-1)-1)-((1-(1-1))$? A. -2 B. -1 C. 0 D. 1 E. 2

Answer reference **-2** rollout **-2** ✔ correct

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **183 tokens**

We are given the expression:

$$
((1 - 1) - 1) - ((1 - (1 - 1)))
$$

We will evaluate it step by step from the innermost parentheses outward.

Step 1: Evaluate the innermost parentheses

- $1 - 1 = 0$
- $1 - 1 = 0$

Now substitute back into the original expression:

$$
((0) - 1) - ((1 - 0))
$$

Step 2: Simplify the first part

- $0 - 1 = -1$

Step 3: Simplify the second part

- $1 - 0 = 1$

Step 4: Combine the results

$$
-1 - 1 = -2
$$

Final Answer:

$$
\boxed{-2}
$$<eos>

Figure A8: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem On the board, a natural number is written. If the number x is written on the board, it is possible to write the number $2x + 1$ or $\frac{x}{x+2}$ on the board. At some point, it was found that the number 2008 was present on the board. Prove that it was there from the very beginning.

Answer reference **2008** rollout **2008 \text{ was on the board from ...** ✓ correct

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **738 tokens**

We are given a process on the board:

- A natural number $ x $ is written on the board.
- It is possible to write $ 2x + 1 $ or $ \frac{x}{x+2} $ on the board.
- At some point, the number $ 2008 $ is present on the board.
- We are to **prove** that $ 2008 $ was there **from the very beginning**, i.e., $ 2008 $ was already on the board at the start, not introduced by some operation.

Step 1: Understand the operations

We are told:

- Starting with some number $ x $, it is possible to write $ 2x + 1 $ or $ \frac{x}{x+2} $.
- We are told that 2008 is present on the board at some point.
- We are to prove that 2008 was **already** on the board, not introduced by any operation.

So, the goal is to prove **that 2008 could not have been introduced by any operation**, i.e., that it could not have come from $ 2x + 1 $ or $ \frac{x}{x+2} $ from a previous number.

Step 2: Assume the opposite

Suppose 2008 is on the board **introduced by an operation**. That would mean there was a number $ y $ such that:

1. $ 2008 = 2y + 1 $, or
2. $ 2008 = \frac{y}{y + 2} $

We will check both cases and see if **any** such $ y $ exists such that 2008 is generated from it.

Step 3: Case 1 – $ 2008 = 2y + 1 $

Solve for $ y $:

$$
2008 = 2y + 1 \Rightarrow 2y = 2007 \Rightarrow y = \frac{2007}{2}
$$

But $ \frac{2007}{2} $ is **not an integer**, and **so** **no natural number** $ y $ satisfies this.

Step 4: Case 2 – $ 2008 = \frac{y}{y + 2} $

Solve for $ y $:

$$
2008 = \frac{y}{y + 2}
$$

Multiply both sides by $ y + 2 $:

$$
2008(y + 2) = y \Rightarrow 2008y + 4016 = y \Rightarrow 2008y - y = -4016 \Rightarrow 2007y = -4016 \Rightarrow y = \frac{-4016}{2007}
$$

This is **not a natural number** (it is negative), so no such $ y $ exists.

Step 5: Conclusion

From both cases, we find that there is **no natural number** $ y $ such that 2008 is generated from it through either operation.

Therefore, **2008 must have been on the board from the very beginning**, not introduced by any operation.

Final Answer:

$$
\boxed{2008 \text{ was on the board from the very beginning}}
$$

$$<eos>

Figure A9: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem How many solutions does the equation \(x + y + z = 2016 \) have, where \(x, y \), and \(z \) are integers with \(x > 1000 \), \(y > 600 \), and \(z > 400 \) ?

Answer reference **105** rollout **105** ✔ correct

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **498 tokens**

```
We are given the equation:

$$
x + y + z = 2016
$$

with the constraints:

- $ x > 1000 $
- $ y > 600 $
- $ z > 400 $

We are to find how many integer
  solutions exist satisfying these
  constraints.

---

### Step 1: Change variables to
  simplify the inequalities

Let's define new variables to shift
  the bounds:

- Let $ x' = x - 1001 $, so $ x' > 0 $
  (since $ x > 1000 $)
- Let $ y' = y - 601 $, so $ y' > 0 $
  (since $ y > 600 $)
- Let $ z' = z - 401 $, so $ z' > 0 $
  (since $ z > 400 $)

Substituting into the original
  equation:

$$
x + y + z = 2016
\Rightarrow (x' + 1001) + (y' + 601) +
  (z' + 401) = 2016
$$

$$
x' + y' + z' + 1001 + 601 + 401 = 2016
\Rightarrow x' + y' + z' + 2003 = 2016
\Rightarrow x' + y' + z' = 13
$$

---

### Step 2: Count the number of
  non-negative integer solutions to $
  x' + y' + z' = 13 $

This is a standard stars and bars
  problem:

- The number of non-negative integer
  solutions to $ x' + y' + z' = 13 $
  is:

$$
\binom{13 + 3 - 1}{3 - 1} =
  \binom{15}{2}
$$

$$
\binom{15}{2} = \frac{15 \times 14}{2}
  = 105
$$

---

### Final Answer:

$$
\boxed{105}
$$

So, the number of solutions to the
  original equation under the given
  constraints is:

$$
\boxed{105}
$$<eos>
```

Figure A10: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem Let \(f(x) = x^3 + ax + b \), with \(a \neq b \), and suppose the tangent lines to the graph of \(f \) at \(x = a \) and \(x = b \) are parallel. Find \(f(1) \).

Answer reference **1** rollout **1** ✔ **correct**

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 | 0.2–0.3 | 0.3–0.4 | ≥ 0.4 **506 tokens**

We are given the function:

$$
f(x) = x^3 + ax + b
$$

with $ a \neq b $, and it's given that the **tangent lines** to the graph of $ f $ at $ x = a $ and $ x = b $ are **parallel**. We are to find $ f(1) $.

Step 1: Find the derivative of $ f(x) $

The derivative of the function is:

$$
f'(x) = \frac{d}{dx}(x^3 + ax + b) = 3x^2 + a
$$

Step 2: Find the slopes of the tangent lines at $ x = a $ and $ x = b $

At $ x = a $, the slope is:

$$
f'(a) = 3a^2 + a
$$

At $ x = b $, the slope is:

$$
f'(b) = 3b^2 + a
$$

Step 3: Use the condition that the tangents are parallel

Since the tangents are parallel, their slopes must be **equal**:

$$
f'(a) = f'(b)
\Rightarrow 3a^2 + a = 3b^2 + a
$$

$$
$$

Subtract $ a $ from both sides:

$$
3a^2 = 3b^2
\Rightarrow a^2 = b^2
$$

So, $ a^2 = b^2 \Rightarrow a = b $ or $ a = -b $

But it is given that $ a \neq b $, so we must have:

$$
a = -b
$$

Step 4: Use the condition $ a = -b $ to find $ f(1) $

Let $ a = -b $

Now substitute into the original function:

$$
f(x) = x^3 + ax + b = x^3 + (-b)x + b = x^3 - bx + b
$$

So we have:

$$
f(x) = x^3 - bx + b
$$

Now compute $ f(1) $:

$$
f(1) = 1^3 - b(1) + b = 1 - b + b = 1
$$

☑ Final Answer:

$$
\boxed{1}
$$<eos>

Figure A11: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem How many 100-digit even numbers exist, where each digit in the decimal representation is one of the digits 0, 1, or 3?

Answer reference **2 \cdot 3^{98}** rollout **3^{99}** ✘ incorrect

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **348 tokens**

```
We are asked to find how many
  **100-digit even numbers** can be
  formed using only the digits **0, 1,
  or 3**, with the condition that
  **each digit is 0, 1, or 3**.

---

### Step 1: Understand the problem
  constraints

- A **100-digit number** must have
  **100 digits**.
- Each digit is either **0**, **1**,
  or **3**.
- The number must be **even**.

---

### Step 2: Determine the condition
  for a number to be even

A number is even if its **last digit
  is even**. In the given set of
  digits {0, 1, 3}, **only 0 is
  even**.

So, the **last digit of the number
  must be 0**.

---

### Step 3: Count the valid numbers

We need to construct a 100-digit
  number with digits from {0, 1, 3},
  where:
- The **last digit is 0**.
- The **first 99 digits can be any
  combination of 0, 1, or 3**.

So, the number of such numbers is:

- **99 digits** (first 99 digits):
  Each can be 0, 1, or 3 → $3^{99}$
  possibilities.
- **1 digit** (last digit): Must be 0
  → 1 possibility.

Total number of such numbers:

$$
3^{99} \times 1 = 3^{99}
$$

---

### ✅ Final Answer:

$$
\boxed{3^{99}}
$$<eos>
```

Figure A12: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.

Problem Find the pattern to fill in the next number: $1, 4, 3, 8, 5, 12, 7, _$

Answer reference **16** rollout **21** ✘ **incorrect**

selected: largest S_t in its block of 256 tokens | block boundary

S_t: 0.1–0.2 0.2–0.3 0.3–0.4 ≥ 0.4 **422 tokens**

We are given the sequence:

$$
1, 4, 3, 8, 5, 12, 7, _
$$

We need to find the **next number** in the pattern.

Step 1: Analyze the sequence

Group the sequence into pairs of two numbers:

$$
(1, 4), (3, 8), (5, 12), (7, _)
$$

Let's look at each pair:

- **First pair:** $1, 4$
  - $4 = 1 \times 4$
- **Second pair:** $3, 8$
  - $8 = 3 \times 2 + 2$
  - Or, $8 = 3 + 5$
- **Third pair:** $5, 12$
  - $12 = 5 \times 2 + 2$
  - Or, $12 = 5 + 7$
- **Fourth pair:** $7, _$
  - We expect a similar pattern: $7 + 5 = 12$, so the next number might be $7 + 5 = 12$
  - But 12 is already in the sequence (from the third pair), so that doesn't make sense.

Let's look at the **difference** between the numbers in the pairs:

- $4 - 1 = 3$
- $8 - 3 = 5$
- $12 - 5 = 7$
- So the **differences** are: 3, 5, 7, ...

This is an **arithmetic sequence** with a common difference of **2**.

So the differences are:
$3, 5, 7, 9, 11, \dots$

So the **next number** in the sequence is:

$$
12 + 9 = 21
$$

Final Answer:

$$
\boxed{21}
$$
<eos>

Figure A13: **Distribution of S_t values and selected tokens during actual rollouts.** For qualitative examples from real rollouts, we visualize the S_t value assigned to each token and highlight the tokens that were ultimately selected.