# Scaling Reasoning Efficiently via Relaxed On-Policy Distillation

**Jongwoo Ko** **Sara Abdali** **Young Jin Kim** **Tianyi Chen** **Pashmina Cameron**
https://relaxed-opd.github.io

## Abstract

On-policy distillation (OPD) is a promising approach for transferring reasoning capabilities to capacity-constrained models, yet sampled-token OPD often suffers from entropy collapse and negative transfer. We identify uniform credit allocation as a key bottleneck: existing objectives obtain dense teacher feedback on student-generated trajectories, but apply it uniformly across sampled tokens despite highly heterogeneous token-level signals. Most tokens carry redundant near-zero credit, while rare heavy-tailed negative credits can dominate updates and prematurely suppress plausible reasoning trajectories. We introduce REOPOLD (**Re**laxed **O**n-**Pol**icy **D**istillation), a framework that relaxes this uniform assignment by controlling where teacher feedback is applied, how strongly it affects the update, and when the allocation rule changes during training. Across diverse reasoning tasks, REOPOLD improves sampled-token OPD over recent post-training baselines with up to 12× higher sample efficiency, and extends to cross-vocabulary distillation, self-distillation, and test-time scaling.

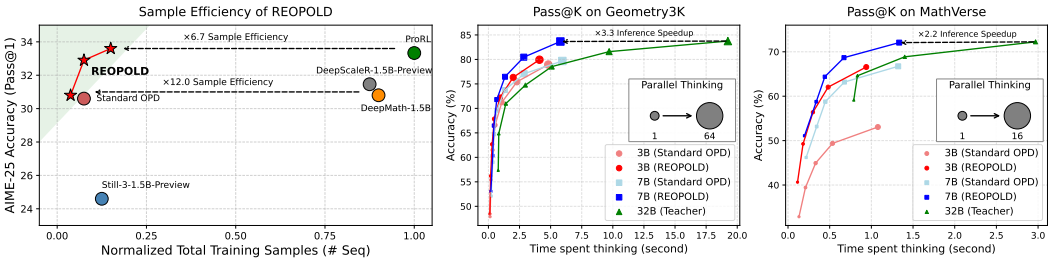
## 1 Introduction

Large language models (LLMs) have achieved remarkable reasoning capabilities through large-scale post-training and test-time scaling, as demonstrated by recent reasoning-oriented systems such as OpenAI's o1/o3 [27] and DeepSeek-R1 [17]. However, replicating this success in small language models (SLMs) proves difficult. Due to limited capacity, SLMs often benefit less efficiently from direct reward optimization, making standard reinforcement learning (RL) less effective for improving reasoning in capacity-constrained models [11, 17, 71]. This disparity necessitates alternative mechanisms specifically tailored for transferring reasoning abilities to capacity-constrained models.

To address this, recent work [46, 73] has adopted on-policy distillation (OPD), where the student learns from its own trajectories under the guidance of a high-capacity teacher with stronger reasoning capabilities. By providing dense teacher-derived feedback on student-sampled trajectories, OPD reduces the burden of reward design while preserving the on-policy nature of training. Empirically, OPD has been shown to outperform RL and supervised fine-tuning (SFT) on reasoning tasks, making it a promising framework for transferring reasoning capabilities. More recently, sampled-token supervision [41, 68] has emerged as a common alternative to full-vocabulary distillation [2, 31], as it avoids scoring the entire vocabulary and therefore scales more naturally to long reasoning trajectories.

However, a central bottleneck in OPD lies not only in acquiring teacher feedback, but also in allocating it effectively. Existing sampled-token methods compute teacher-derived credits (*i.e.,* log-likelihood ratios) on student-generated trajectories, but largely apply these credits uniformly over all sampled tokens. This uniform allocation overlooks the highly imbalanced nature of token-level credits: most sampled tokens provide redundant near-zero signals, whereas rare but heavy-tailed negative credits can dominate optimization, as shown in Figure 2. As a result, stable and efficient OPD requires mechanisms that decide *where* teacher feedback should be applied, *how much* it should influence the update, and *when* it should be trusted.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).




(a) REOPOLD on DeepSeek-R1-Distill-Qwen-1.5B (b) Test-time scaling of REOPOLD

Figure 1: **Performance of REOPOLD.** (a) **Sample Efficiency**: REOPOLD achieves a favorable accuracy–sample-efficiency trade-off on AIME-25 compared with existing post-training baselines, where sample efficiency is measured by the number of generated training sequences. Detailed explanation can be found in Section 4.1. (b) **Test-time Scaling**: On visual reasoning tasks, REOPOLD demonstrates superior test-time scaling capabilities compared to the standard OPD baseline. Notably, it allows smaller models to approach the performance of the 32B teacher. Detailed explanation can be found in Section 4.2.

Empirically, this imbalance manifests as instability in standard OPD [14, 16]. We observe frequent negative transfer, where the student degrades relative to its base initialization (Section 4.1 and Figure 5), along with rapid entropy collapse that leads to premature convergence (Section 2 and Figure 4). This motivates our selective credit allocation view, which is related to prior work on non-uniform sample allocation and robust example reweighting [37, 50, 52, 56]. Rather than treating all sampled-token credits uniformly, OPD should selectively allocate teacher supervision according to the student's current learning dynamics.

**Contributions.** We analyze sampled-token OPD as a problem of selective credit allocation. Instead of treating all teacher-derived token signals as equally useful, we ask where teacher feedback should be applied, how strongly it should influence the update, and when the allocation rule should change during training. Our contributions are summarized as follows:

- **Credit allocation diagnosis:** We show that standard OPD implicitly applies a uniform credit allocation rule over sampled tokens. This rule is poorly matched to the heterogeneous credit structure of OPD, where a large fraction of sampled tokens carry near-zero credit, while rare teacher-student mismatches induce heavy-tailed negative credit and premature entropy collapse.
- **Student-conditioned credit allocation:** To address these, we propose **REOPOLD**, a unified framework that selectively regulates teacher credit along three axes: update magnitude, token uncertainty, and training stage. Concretely, REOPOLD integrates bounded negative credit, entropy-guided token selection, and exploration-to-refinement scheduling to stabilize sampled-token distillation.
- **State-of-the-art efficiency and scalability:** Empirically, REOPOLD achieves superior training sample efficiency and enables stronger test-time scaling for SLMs (Figure 1). Across mathematical, visual [39], and tool-use [47] reasoning tasks, REOPOLD consistently outperforms baselines and remains effective beyond same-family distillation, including cross-vocabulary [5, 46] and privileged self-distillation [79].

## 2 Motivation: Indiscriminate Credit Allocation in On-Policy Distillation

### 2.1 Background: Sampled-token on-policy distillation

Let $\pi_\theta$ denote the student policy and $\pi_T$ the teacher policy. For a query $q$, OPD samples responses from the student's behavior policy $\pi_{\theta_{\text{old}}}$ and updates the student using teacher feedback on the sampled tokens. A common sampled-token reverse-KL objective, written in maximization form, is

$$\mathcal{J}_{\text{RKL}}(\theta) = \mathbb{E}_{q\sim\mathcal{Q},\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{\sum_i|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\rho_{i,t}(\theta)R_{i,t}(\theta)\right], \tag{1}$$

where

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}, \qquad R_{i,t}(\theta) = \log\frac{\pi_T(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})}. \tag{2}$$

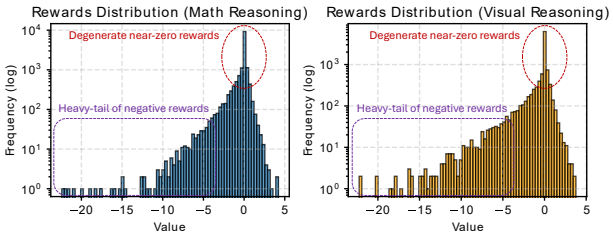



Figure 2: **Log-scale histogram of teacher-derived credit.** The red dashed ellipse indicates degenerate near-zero credits, while the purple dashed region highlights the heavy tail of negative credits.

Table 1: Update-energy (Eq. 3) concentration in negative-credit tails.

| Bottom | ∼1% | ∼5% | ∼10% |
|---|---|---|---|
| Math | 95.9% | 97.1% | 98.4% |
| Visual | 97.8% | 98.6% | 99.5% |

Table 2: Retention (Eq. 4) on base policy samples during OPD training.

| Step ($k$) | 0 | 10 | 300 |
|---|---|---|---|
| Math | -1.583 | -2.554 | -2.841 |
| Visual | -1.427 | -2.315 | -2.741 |

We refer to $R_{i,t}$ as the teacher-derived token credit. Following sampled-token OPD practice, we detach $R_{i,t}$ and treat it as a fixed token-level credit within each update; the corresponding gradient derivation is provided in Appendix B.1.

This sampled-token formulation differs from full-vocabulary distillation [2, 14, 31], which compares teacher and student distributions over the entire vocabulary at each generated position. While full-vocabulary supervision provides richer feedback, it is memory-intensive for modern reasoning distillation with long trajectories and multiple rollouts (Table 12). A lightweight alternative is to evaluate the teacher only on the token sampled by the student, which has become widely used in recent OPD work [41, 68, 74]. We also provide detailed related work in Appendix A and empirical comparisons with full-vocabulary OPD and their corresponding Top-$K$ approximations in Table 12.

### 2.2 Diagnosing Credit Misallocation in Sampled-Token OPD

Sampled-token OPD provides teacher feedback only on tokens sampled by the student, but standard objectives (Eq. 1) aggregate these credits uniformly across samples. This treats all sampled tokens as equally useful supervision targets, despite many credits being uninformative, redundant, or extreme. Because token-level credits are highly skewed in practice, this uniform allocation creates two finite-sample failure modes.

**Most token credits are redundant.** As shown in Figure 2, the sampled-token credit distribution has a large mass near zero. These credits occur when the teacher and student assign similar probability to the sampled token, yielding little corrective signal. While such tokens are not harmful individually, uniform aggregation makes them consume normalization mass and mini-batch budget that could be allocated to more informative tokens. As a result, many updates are diluted by nearly inactive credits, reducing the effective learning signal obtained from each sampled batch.

**A few negative credits dominate the update.** The more severe failure mode arises from the negative tail of the credit distribution. As shown in Figure 2, when the student samples a token that the teacher assigns very low probability, the credit $R_{i,t}(\theta)$ in Eq. 2 can become highly negative. Although this tail accounts for only a small fraction of sampled positions, its large magnitude can dominate optimization. To measure this concentration, we compute a logit-level update-energy proxy,

$$E_{i,t}(\theta) = R_{i,t}(\theta)^2 \left(1 - \|\pi_\theta(\cdot \mid q, o_{i,<t})\|_2^2\right), \tag{3}$$

which serves as an expected logit-level proxy for the squared gradient and allows us to decompose update energy by credit quantile. This proxy assumes stop-gradient through $R_{i,t}$, replacing the token-specific score norm by its expectation $1 - \|\pi_\theta\|_2^2$; see Appendix B.2 for the derivation. We find that the bottom $1\%$ most negative credits account for over $95\%$ of the total update energy in early standard OPD training (Table 1).

This concentration can be harmful because large negative credits strongly suppress the probability of sampled tokens [32, 67]. In autoregressive models, suppressing an intermediate token can also disrupt the subsequent reasoning path conditioned on it. Thus, standard OPD may reduce support for trajectories that are locally disfavored by the teacher but still useful for exploration. We therefore use a base-policy retention diagnostic to measure how much support later checkpoints assign to trajectories sampled from the initial student $\pi_{\theta_0}$:

$$\mathrm{Ret}(k) = \mathbb{E}_{q,o\sim\pi_{\theta_0}}\left[\frac{1}{|o|}\sum_t \log \pi_{\theta_k}(o_t|q, o_{<t})\right]. \tag{4}$$

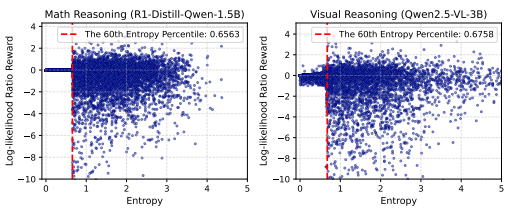



Figure 3: **Correlation between token entropy and credit.** Experimental results on math reasoning and visual reasoning benchmarks demonstrate that credits in the bottom 60th entropy percentile are heavily concentrated around zero. This suggests that while teacher and student policies may diverge overall, they remain highly consistent on low-entropy, deterministic tokens, with significant deviations occurring primarily in high-entropy regimes.

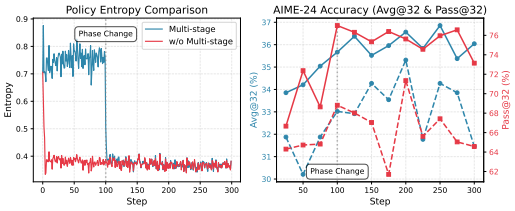



Figure 4: **Analysis of exploration-to-refinement scheduling. (Left)** The proposed multi-stage approach prevents entropy collapse and preserves diverse modes during the initial exploration phase by masking strongly negative credits, whereas the baseline suffers from rapid entropy collapse. **(Right)** The multi-stage strategy (solid lines) achieves superior final performance to the baseline (dashed lines) in both quality (i.e., Avg@32) and diversity (i.e., Pass@32).

Lower values mean that the current policy assigns less probability to trajectories originally generated by the base policy. Standard OPD rapidly reduces this support: within the first 10 steps, retention decreases from -1.58 and -1.43 to -2.55 and -2.32 (Table 2), corresponding to about 40% of the initial token-level likelihood. Together with the update-energy concentration and entropy collapse in Figure 4, this suggests rapid policy concentration under standard OPD.

**Useful teacher signal is student-conditioned.** These observations show that teacher feedback is not uniformly useful across sampled tokens. Near-zero credits provide little corrective signal, while rare large negative credits may capture real teacher-student disagreement but destabilize optimization at full strength. The bottleneck in standard OPD is therefore credit allocation: which sampled-token credits to use, how strongly to apply them, and when to change the allocation rule during training.

## 3 REOPOLD: Relaxed On-Policy Distillation via Selective Credit Allocation

The analysis in Section 2 motivates a selective credit allocation view of OPD. Under this view, an objective should decide *where* teacher feedback should be applied, *how much* credit should be allowed to influence the update, and *when* negative correction should be trusted. REOPOLD instantiates these choices through entropy-guided token selection, bounded negative credit, and an exploration-to-refinement schedule. For each batch $\mathcal{B}$ sampled from the query set $\mathcal{Q}$, we maximize:

$$\mathcal{J}_{\text{REOPOLD}}(\theta) = \mathbb{E}_{\mathcal{B}\sim\mathcal{Q}, q\sim\mathcal{B}, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[ \frac{1}{\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} M_{i,t}^{(k)}} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \rho_{i,t}(\theta)\hat{R}_{i,t}^{\lambda}(\theta) M_{i,t}^{(k)} \right], \tag{5}$$

where

$$\hat{R}_{i,t}^{\lambda}(\theta) = \max\left(R_{i,t}(\theta), \frac{\log\lambda}{1-\lambda}\right), \qquad M_{i,t}^{(k)} = \begin{cases} \mathbb{I}\left[R_{i,t}(\theta) \geq \frac{\log\lambda}{1-\lambda}\right] & \text{if } k < T_{\text{switch}} \\ \mathbb{I}\left[H_t^i \geq \tau_\beta\right] & \text{if } k \geq T_{\text{switch}} \end{cases} \tag{6}$$

Throughout training, we treat the teacher-derived credits and token-selection masks as stop-gradient quantities when computing policy updates. The full training procedure is summarized in Algorithm 1.

### 3.1 Where to Allocate Credit: Entropy-Guided Token Selection

The first axis of credit allocation concerns *where* teacher feedback should be applied. Standard OPD allocates update budget uniformly across all sampled tokens, even though many near-zero credit tokens occur at low-entropy positions. As shown in Figure 3, these tokens are concentrated in contexts where the student and teacher assign similar probabilities to the sampled token, making additional correction largely redundant. The following elementary softmax identity supports using entropy as a lightweight allocation signal.

*Remark* 3.1. *Let $z \in \mathbb{R}^{|\mathcal{V}|}$ be the logit vector over a finite vocabulary $\mathcal{V}$, and let $p = \mathrm{softmax}(z)$ be the categorical distribution, with $p_v$ denoting the probability of token $v \in \mathcal{V}$. If $y \sim p$, then*

$$\mathbb{E}_{y\sim p}\left[\|\nabla_z \log p_y\|_2^2\right] = 1 - \|p\|_2^2 \le H(p), \tag{7}$$

*where $p_y$ is the probability of the sampled token and $H(p) = -\sum_{v\in\mathcal{V}} p_v \log p_v$.*

Applied to $p = \pi_\theta(\cdot|x)$ at context $x = (q, o_{<t})$, $\nabla_z \log p_y$ is the logit-level score of the sampled token, which captures how much a token-level likelihood update can locally redistribute probability mass [30, 65]. Remark 3.1 shows that entropy upper-bounds the expected squared score norm, which captures local update leverage in logit space; the derivation is given in Appendix B.3. The teacher credit provides the direction and magnitude of supervision, while student entropy serves as a proxy for local update leverage, motivating its use as a token-selection signal.

Motivated by this, we use student entropy as a lightweight allocation signal. For each sampled token $o_{i,t}$, let $H_t^i$ denote the entropy of the student's policy $\pi_\theta(\cdot|q, o_{i,<t})$. We define a token-level allocation mask

$$M_{i,t}^{\mathrm{ent}} = \mathbb{I}\left[H_t^i \ge \tau_\beta\right], \tag{8}$$

where $\tau_\beta$ is chosen so that $M_{i,t}^{\mathrm{ent}}$ selects the top-$\beta$ fraction of token entropies within the batch. This mask selects positions where the student remains uncertain, rather than uniformly accepting teacher credit at every sampled token. Integrating this mask into the sampled-token objective gives

$$\mathcal{J}_{\mathrm{Ent}}(\theta) = \mathbb{E}_{\mathcal{B}\sim\mathcal{Q}, q\sim\mathcal{B}, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|q)}\left[\frac{\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\rho_{i,t}(\theta)R_{i,t}(\theta)M_{i,t}^{\mathrm{ent}}}{\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}M_{i,t}^{\mathrm{ent}}}\right]. \tag{9}$$

This creates a dynamic token batch whose effective size is determined by the student's uncertainty rather than sequence length. By filtering out low-entropy, near-zero-credit tokens, entropy-guided allocation reduces gradient dilution and concentrates update budget on positions where teacher feedback is most likely to be useful. Empirically, the sensitivity analysis in Figure 6 confirms that stricter entropy selection accelerates convergence and improves final accuracy, supporting the view that selective allocation of teacher credit is more effective than uniform token-level imitation.

### 3.2 How Much Credit to Use: Bounded Negative Credit

The second allocation axis controls how strongly the student should absorb negative teacher feedback. If the student samples a token with negligible teacher probability, the log-ratio credit can become extremely negative and dominate the update. We bound this tail using a mixture relaxation of the teacher and student distributions. For $\lambda \in [0, 1)$, Jensen's inequality gives

$$R_{i,t}(\theta) = \log\frac{\pi_T(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})} \le \frac{1}{1-\lambda}\log\frac{(1-\lambda)\pi_T(o_{i,t}|q, o_{i,<t}) + \lambda\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_\theta(o_{i,t}|q, o_{i,<t})}. \tag{10}$$

The key insight comes from the asymptotic behavior of the relaxed term; detailed derivation is provided in Appendix B.4. While the original credit $R_{i,t}(\theta)$ diverges to $-\infty$ as $\pi_T(o_{i,t}|q, o_{i,<t}) \to 0$, the relaxed expression converges to $c_\lambda = \frac{\log\lambda}{1-\lambda}$. Motivated by this asymptotic limit, we use $c_\lambda$ as a clipping floor for the original sampled-token credit, rather than directly optimizing the mixture-relaxed objective:

$$\hat{R}_{i,t}^{\lambda}(\theta) = \max\left(R_{i,t}(\theta), c_\lambda\right). \tag{11}$$

In the credit allocation view, this operation decides how much a single negative teacher signal is allowed to influence the student. It preserves the objective for ordinary tokens, while preventing rare teacher-student mismatches from consuming disproportionate update budget. Unlike Skew RKL [31], which changes the full objective and is sensitive to $\lambda$, this rule only targets outlying negative credits, yielding the robustness shown in Figure 11.

### 3.3 When to Change Allocation: Exploration-to-Refinement Scheduling

The third allocation axis controls *when* different teacher credits should be accepted. The usefulness of teacher feedback changes over training: early updates should preserve plausible student trajectories,

Table 3: Performance comparison of REOPOLD and distillation baselines (SFT and standard OPD) on mathematical reasoning benchmarks across different teacher models. Accuracy (%) is reported for all benchmarks. The best result in each column is shown in **bold**, and the second-best is underlined. The results are reported as mean ± standard deviation over 5 runs.

| Model | AIME-24 | AIME-25 | AMC-23 | MATH-500 | Minerva Math | Olympiad Bench | AVG. |
|---|---|---|---|---|---|---|---|
| **SkyWork-OR1-Math-7B [20] → DeepSeek-R1-Distill-Qwen-1.5B [17]** | | | | | | | |
| SkyWork-OR1-Math-7B | 69.8 | 52.3 | 94.1 | 95.8 | 49.3 | 73.5 | 72.5 |
| R1-Distill-Qwen-1.5B | 28.6 | 22.7 | 62.6 | 82.9 | 26.4 | 43.6 | 44.4 |
| + GRPO | $31.8_{\pm 1.0}$ | $23.7_{\pm 0.9}$ | $62.0_{\pm 1.1}$ | $85.4_{\pm 0.6}$ | $33.8_{\pm 0.3}$ | $49.8_{\pm 0.6}$ | $47.8_{\pm 0.3}$ |
| + SFT | $33.5_{\pm 1.5}$ | $24.6_{\pm 1.9}$ | $76.1_{\pm 1.2}$ | $86.6_{\pm 0.8}$ | $\underline{36.4}_{\pm 0.8}$ | $55.6_{\pm 0.8}$ | $51.5_{\pm 0.9}$ |
| + Standard OPD | $\underline{37.1}_{\pm 1.4}$ | $\underline{30.6}_{\pm 0.6}$ | $\underline{80.2}_{\pm 0.9}$ | $\underline{88.0}_{\pm 0.5}$ | $34.6_{\pm 0.4}$ | $\underline{56.0}_{\pm 0.8}$ | $\underline{54.4}_{\pm 0.2}$ |
| + REOPOLD | $\mathbf{41.6}_{\pm 1.1}$ | $\mathbf{32.6}_{\pm 0.4}$ | $\mathbf{83.0}_{\pm 1.2}$ | $\mathbf{89.2}_{\pm 0.4}$ | $\mathbf{38.6}_{\pm 0.7}$ | $\mathbf{57.3}_{\pm 0.5}$ | $\mathbf{57.1}_{\pm 0.3}$ |
| **Phi-4-Mini-Reasoning [69] → DeepSeek-R1-Distill-Qwen-1.5B [17]** | | | | | | | |
| Phi4-Mini-Reasoning | 55.6 | 41.4 | 91.8 | 94.6 | 39.1 | 64.5 | 64.5 |
| + Standard OPD | $13.3_{\pm 0.5}$ | $21.0_{\pm 1.1}$ | $45.5_{\pm 2.1}$ | $78.6_{\pm 2.7}$ | $19.4_{\pm 1.4}$ | $32.8_{\pm 1.0}$ | $35.1_{\pm 1.1}$ |
| + REOPOLD | $\mathbf{32.1}_{\pm 1.7}$ | $24.6_{\pm 0.7}$ | $65.2_{\pm 1.2}$ | $83.7_{\pm 1.3}$ | $30.2_{\pm 0.5}$ | $45.9_{\pm 0.6}$ | $47.0_{\pm 0.4}$ |

while later updates should refine unresolved token decisions. We implement this temporal control through a token-wise allocation mask:

$$\mathcal{J}_{\text{REOPOLD}}(\theta) = \mathbb{E}_{\mathcal{B}\sim\mathcal{Q}, q\sim\mathcal{B}, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[ \frac{\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \rho_{i,t}(\theta)\hat{R}^{\lambda}_{i,t}(\theta) M^{(k)}_{i,t}}{\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} M^{(k)}_{i,t}} \right]. \tag{12}$$

Here, $M^{(k)}_{i,t}$ determines which token credits are accepted at training step $k$. By changing this mask over time, REOPOLD uses different allocation rules for exploration and refinement.

**Exploration phase.** The goal of the early phase is to preserve exploration while avoiding destructive teacher corrections. For $k < T_{\text{switch}}$, REOPOLD keeps dense supervision but filters out excessively negative credits:

$$M^{(k)}_{i,t} = \mathbb{I}\left[ R_{i,t}(\theta) \geq \frac{\log\lambda}{1-\lambda} \right]. \tag{13}$$

This mask rejects only credits whose negative magnitude exceeds the bounded-credit threshold in Section 3.2, delaying overly aggressive correction that could prematurely prune candidate reasoning paths. As shown in Figure 4, the first phase improves both Avg@32 and Pass@32, suggesting that bounded negative correction preserves useful exploration rather than merely slowing optimization.

**Refinement phase.** The goal of the later phase is to concentrate update budget on unresolved decisions. For $k \geq T_{\text{switch}}$, REOPOLD switches to the entropy-guided allocation rule from Section 3.1:

$$M^{(k)}_{i,t} = \mathbb{I}\left[ H^i_t \geq \tau_\beta \right]. \tag{14}$$

This mask selects high-entropy positions where the student remains uncertain and teacher feedback is more actionable. Thus, the schedule first preserves exploratory support and then refines uncertain token decisions. Empirically, Figure 4 shows that this transition further improves Avg@32 while maintaining the diversity gains reflected by Pass@32.

# 4 Experimental Results

## 4.1 Main Results: Math Reasoning

**Setup.** We conduct on-policy distillation on DeepSeek-R1-Distill-Qwen-1.5B and 7B [17], employing SkyWork-OR1-7B (or Phi-4-mini-reasoning [69]) and SkyWork-OR1-32B-Preview [20] as teachers, respectively. For training, we utilize the dataset proposed by Yan et al. [71], which contains 45k prompts. While all 1.5B models in Table 3 are trained for 300 steps for fair comparison, we extend the training of REOPOLD to 600 steps for the sample efficiency analysis in Figure 1(a). Detailed training setup is described in Appendix D.

**Evaluation.** We evaluate all models on six competition-level mathematical reasoning benchmarks: AIME-24, AIME-25, AMC-23, MATH-500 [22], Minerva Math [34], and Olympiad Bench [19]. For AIME-24, AIME-25, and AMC-23, we report Avg@32 to ensure robust evaluation considering the relatively small test sets. For the remaining three benchmarks, we report *Pass@1*. In all evaluations, we use a temperature of 0.6 and a top-p value of 0.95.

Table 4: Performance comparison of vision-language models on visual reasoning and perception benchmarks. Accuracy (%) is reported for all benchmarks. The best result in each column is shown in **bold**, and the second-best is underlined. ‡ denotes results obtained with the Top-5 approximation, as the full-vocabulary implementation incurs memory issues. Our reproduced OPD results are reported as mean ± standard deviation over 5 runs.

| Model | Geo3K | MathVerse | MathVision | MathVista | WeMath | Hallusion | AVG. |
|---|---|---|---|---|---|---|---|
| Qwen2.5-VL-32B-Instruct + NoisyRollout | 56.74 | 58.86 | 39.82 | 78.30 | 75.51 | 72.45 | 63.61 |
| **Qwen2.5-VL-32B-Instruct + NoisyRollout [39] → Qwen2.5-VL-3B-Instruct** | | | | | | | |
| Qwen2.5-VL-3B-Instruct | $26.46_{\pm 1.58}$ | $35.58_{\pm 0.44}$ | $22.83_{\pm 0.66}$ | $59.40_{\pm 0.49}$ | $53.41_{\pm 0.41}$ | $61.51_{\pm 0.39}$ | $43.20_{\pm 0.54}$ |
| + PAPO [62] | 32.95 | 40.65 | 24.16 | **65.10** | 63.62 | 61.62 | 48.02 |
| + Standard OPD [41] | $48.09_{\pm 0.39}$ | $33.25_{\pm 0.41}$ | $24.78_{\pm 0.62}$ | $62.50_{\pm 0.56}$ | $64.48_{\pm 0.43}$ | $60.83_{\pm 0.38}$ | $48.99_{\pm 0.34}$ |
| + REOPOLD | **50.58**$_{\pm 0.64}$ | **46.40**$_{\pm 0.40}$ | **26.39**$_{\pm 0.38}$ | $61.50_{\pm 0.45}$ | **64.60**$_{\pm 0.23}$ | **63.62**$_{\pm 0.21}$ | **52.18**$_{\pm 0.21}$ |
| **Qwen2.5-VL-32B-Instruct + NoisyRollout [39] → Qwen2.5-VL-7B-Instruct** | | | | | | | |
| Qwen2.5-VL-7B-Instruct | $39.77_{\pm 1.33}$ | $45.72_{\pm 0.69}$ | $25.05_{\pm 0.68}$ | $67.80_{\pm 0.45}$ | $64.77_{\pm 0.53}$ | $65.62_{\pm 1.39}$ | $51.46_{\pm 0.13}$ |
| + GRPO [53] | 51.4 | 50.8 | 27.3 | 70.5 | 67.4 | 69.8 | 56.20 |
| + NoisyRollout [39] | 50.08 | **53.14** | 26.64 | 72.00 | **70.57** | **70.66** | 57.18 |
| + Standard OPD [41] | $51.75_{\pm 0.40}$ | $47.71_{\pm 0.61}$ | $28.79_{\pm 0.32}$ | $71.27_{\pm 0.68}$ | $70.06_{\pm 0.64}$ | $69.51_{\pm 1.29}$ | $56.51_{\pm 0.19}$ |
| + GKD‡ [2] | $52.07_{\pm 0.84}$ | $48.31_{\pm 0.47}$ | $27.98_{\pm 0.53}$ | $70.89_{\pm 0.36}$ | $69.34_{\pm 0.41}$ | $69.75_{\pm 1.13}$ | $56.39_{\pm 0.24}$ |
| + REOPOLD | **53.58**$_{\pm 0.71}$ | $51.43_{\pm 0.46}$ | **29.21**$_{\pm 0.49}$ | **72.40**$_{\pm 0.68}$ | $69.77_{\pm 0.24}$ | $70.14_{\pm 0.86}$ | **57.76**$_{\pm 0.31}$ |

**Result 1: Better training sample-efficiency.** As shown in Figure 1(a), REOPOLD matches ProRL [38] in 600 steps (vs. 2000); when normalized by the total number of generated sequences, computed as steps × batch size × rollouts, this yields a $> 6.7\times$ efficiency gain. It surpasses DeepScaleR-1.5B-Preview [44] and DeepMath-1.5B [21] even earlier at 300 steps ($> 12\times$ efficiency), and notably outperforms standard OPD (300 steps) in just 150 steps. Finally, superior performance over GRPO in our re-implementation under identical conditions (Table 3) confirms that our gains stem from algorithmic efficacy rather than experimental setup.

**Result 2: Robustness to different model family.** Using the sequence alignment [5, 46], we distill Phi-4-Mini-Reasoning into DeepSeek-R1-Distill-Qwen-1.5B, where the teacher and student come from different model families and use different vocabularies. This setting induces a large teacher-student distribution mismatch, under which standard OPD suffers severe negative transfer, reducing average performance from 44.4 to 35.1. In contrast, REOPOLD improves the student to 47.0, showing that selective credit allocation can stabilize OPD even when teacher feedback is transferred across mismatched vocabularies and model families.

**Result 3: Scaling to large policy models.** We further evaluate REOPOLD on a stronger student, DeepSeek-R1-Distill-Qwen-7B. As shown in Figure 5, standard OPD is unstable: performance drops sharply early in training and fails to recover beyond the base model on AIME-24. In contrast, REOPOLD prevents early degradation and steadily improves across benchmarks, demonstrating that selective credit allocation remains effective when scaling OPD to stronger policy models.

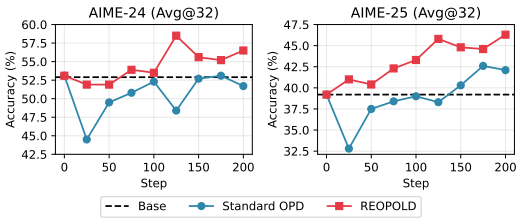



Figure 5: Scaling to a 7B student exposes instability in standard OPD, while REOPOLD maintains stable and consistent improvement.

### 4.2 Main Results: Visual Reasoning

We present the primary evaluation results on visual reasoning tasks below. Please refer to Appendix E for additional ablation studies and detailed discussions.

**Setup.** We adopt Qwen2.5-VL-3/7B-Instruct [3] as the student policy and Qwen2.5-VL-32B-Instruct, trained with NoisyRollout [39], as the teacher. We train the student model on Geometry3K [42], which focuses on geometric problem solving and comprises approximately 2.1K training samples. Following the protocol of Liu et al. [39], we pre-process this dataset by converting all multiple-choice questions into free-form answer formats to mitigate reward hacking and reduce the likelihood of answer guessing. Detailed training setup is described in Appendix D.

**Evaluation.** We assess across six benchmarks: five visual reasoning benchmarks, including the test split of Geometry3K, MathVerse [77], MathVision [59], MathVista [43], and WeMath [48], as well

Table 5: Performance comparison of Qwen2.5-VL-3/7B-Instruct trained with 32B teacher models on visual reasoning benchmarks. Accuracy (%) is reported for all benchmarks.

| | (1) | (2) | (3) | Geo3K | Verse | Vision | AVG. |
|---|---|---|---|---|---|---|---|
| **Qwen2.5-VL-3B-Instruct** | | | | | | | |
| Standard OPD | | | | 48.09 | 33.25 | 24.78 | 35.37 |
| | ✓ | | | 48.59 | 45.53 | 25.07 | 39.73 |
| | ✓ | ✓ | | 50.08 | 45.94 | 26.42 | 40.81 |
| REOPOLD | ✓ | ✓ | ✓ | 50.58 | 46.40 | 26.39 | 41.12 |
| **Qwen2.5-VL-7B-Instruct** | | | | | | | |
| Standard OPD | | | | 51.75 | 47.71 | 28.79 | 42.75 |
| | ✓ | | | 51.75 | 51.45 | 29.05 | 44.08 |
| | ✓ | ✓ | | 52.75 | 51.42 | 29.10 | 44.42 |
| REOPOLD | ✓ | ✓ | ✓ | 53.58 | 51.43 | 29.21 | 44.74 |

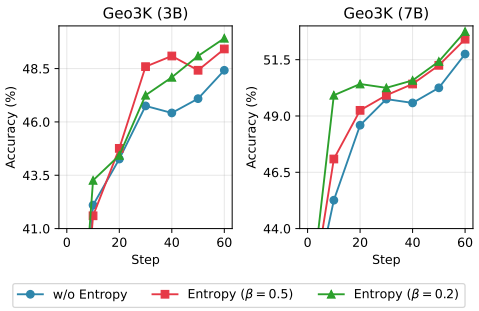



Figure 6: Comparison among REOPOLD using different entropy-selection fractions. Our token selection accelerates convergence and leads to superior final accuracy.

as one visual perception benchmark, HallusionBench [15]. By following the evaluation protocol of Liu et al. [39], we employ greedy sampling and nucleus sampling [24] with a temperature of 0.6 and a top-p of 0.95 for model inference and use Gemini-2.0-Flash-001 [58] as the judge model to parse generated responses.

**Result 1: Efficacy on compact models.** As shown in Table 4, REOPOLD achieves superior overall performance compared to standard OPD, GKD [2], and RL baselines across visual reasoning and perception benchmarks for both 3B and 7B models. REOPOLD also surpasses specialized perception algorithms like NoisyRollout (7B) and PAPO ([62]; 3B). The Top-$K$ study in Appendix E further confirms that REOPOLD extends beyond sampled-token supervision by outperforming Top-5 approximated full-vocabulary OPD methods. This highlights its robustness for compact models: by selectively applying teacher credits, REOPOLD avoids unnecessary imitation that can be harmful in low-capacity regimes.

**Result 2: Superior test-time scaling.** We evaluate test-time scaling by measuring *Pass@*$K$ as a function of parallel thinking latency which is the average time to generate $K$ responses per question. As shown in Figure 1(b), REOPOLD achieves stronger scaling curves than standard OPD and the Qwen2.5-VL-32B teacher, reaching up to $3.32\times$ higher inference efficiency in terms of Pass@$K$. This gain comes from combining higher generation quality than standard OPD with the lower latency of a compact student model. Extended Maj@$K$ results in Figure 12 show the same trend across test-time scaling metrics.

### 4.3 Analysis: Visual Reasoning

We conduct additional analyses using the visual reasoning setup in Section 4.2, unless otherwise specified, to further evaluate the effectiveness and generalizability of REOPOLD.

**Result 1: Impact of module design.** Table 5 validates the three axes of selective credit allocation: (1) bounded negative credit (2) entropy-guided token selection, and (3) exploration-to-refinement scheduling. Each component improves over standard OPD by addressing a distinct allocation failure: overly large negative credits, redundant low-entropy tokens, and the changing usefulness of teacher feedback over training. Combining all three achieves the best average performance for both 3B and 7B students. Consistently, Figure 6 shows that selecting a smaller fraction of high-entropy tokens (*e.g.*, $\beta = 0.2$ or $\beta = 0.5$) accelerates convergence and improves final accuracy, supporting the view that filtering redundant low-entropy tokens mitigates gradient dilution. We refer the reader to Figure 11 and Figure 4 for extended analyses on bounded negative credit and exploration-to-refinement scheduling.

**Result 2: Scalability with longer training.** We further evaluate scalability by extending training to 1000 steps for both 3B and 7B students. As shown in Figure 7, standard OPD does not consistently benefit from additional optimization: it saturates around 300 steps for the 3B model and gradually degrades for the 7B model. In contrast, REOPOLD continues improving beyond the exploration-to-refinement transition at step 200 and maintains higher performance across both model sizes. This suggests that REOPOLD better stabilizes long-horizon OPD, enabling continued policy refinement with longer training. Detailed 300-step results are additionally provided in Table 10 and Table 11.

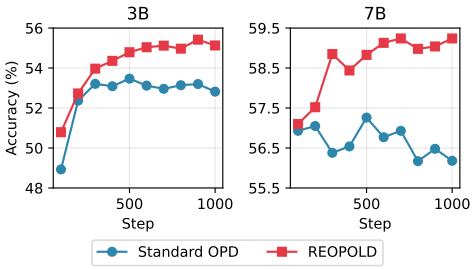



Figure 7: Convergence curves of OPD and REOPOLD over 1000 training steps.

Table 6: Performance comparison of Qwen2.5-VL-3/7B-Instruct trained by OPSD [79] on visual reasoning benchmarks. Accuracy (%) is reported for all benchmarks.

| | Geo3K | Verse | Vision | Vista | AVG. |
|---|---|---|---|---|---|
| **Qwen2.5-VL-3B-Instruct** | | | | | |
| OPSD | 44.27 | 41.94 | 24.16 | 61.75 | 43.03 |
| Relaxed OPSD | **49.82** | **45.14** | **25.71** | **61.82** | **45.62** |
| **Qwen2.5-VL-7B-Instruct** | | | | | |
| OPSD | 50.72 | 49.41 | 28.22 | 70.33 | 49.67 |
| Relaxed OPSD | **51.82** | **50.43** | **29.07** | **70.84** | **50.54** |

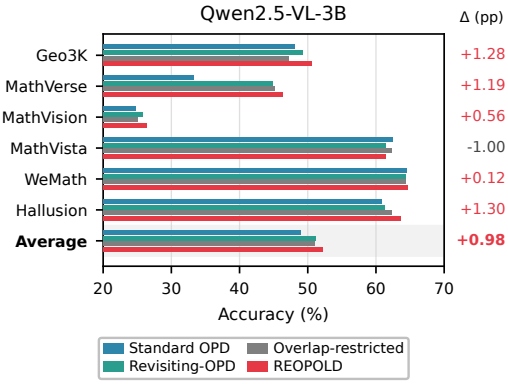



Figure 8: OPD comparisons on Qwen2.5-VL-3B. The axis starts at 20; Δ is REOPOLD minus the best other method in each row (pp).

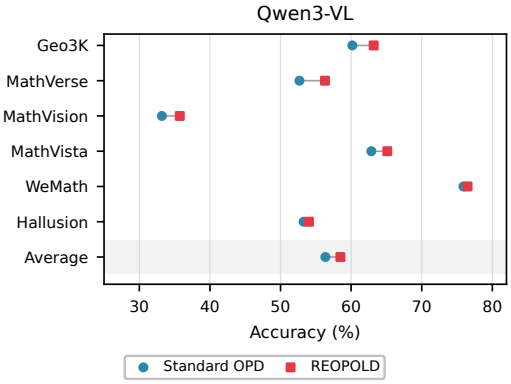



Figure 9: Extension to Qwen3-VL. Paired markers compare standard OPD and REOPOLD across six benchmarks. Shading marks the average.

**Result 3: Extension to self-distillation.** We further test whether selective credit allocation extends to on-policy self-distillation (OPSD). In OPSD, the same model serves as both teacher and student: the teacher policy is conditioned on privileged information, such as ground-truth solutions, while the student policy observes only the original prompt [54, 79]. Applying our selective credit allocation strategy to OPSD yields Relaxed OPSD. As shown in Table 6, Relaxed OPSD consistently improves average accuracy over OPSD for both Qwen2.5-VL-3B-Instruct (43.03 to 45.62) and Qwen2.5-VL-7B-Instruct (49.67 to 50.54). This shows that selective credit allocation is not limited to external-teacher distillation, but also stabilizes privileged self-distillation.

**Result 4: Comparison with other OPD objectives.** We further compare REOPOLD with recent OPD objectives. Revisiting-OPD matches the teacher's top-$K$ local support [12], while the overlap-restricted variant uses the shared-support objective from Li et al. [36]. As shown in Figure 8, REOPOLD achieves the highest average accuracy of 52.18, outperforming Revisiting-OPD (51.20) and the overlap-restricted variant (51.08) by 0.98 and 1.10 points, respectively. These results suggest that selective credit allocation provides benefits beyond modifying the supervision support alone.

**Result 5: Extension to Qwen3-VL.** We additionally evaluate REOPOLD on Qwen3-VL [73] to examine its effectiveness across different model architectures. As shown in Figure 9, REOPOLD consistently outperforms standard OPD across all six benchmarks, improving average accuracy from 56.35 to 58.48 (+2.13 points). This demonstrates that the benefits of selective credit allocation extend beyond Qwen2.5-VL.

### 4.4 Extension: Agentic Reasoning with Visual Tool-Use

Traditional visual reasoning approaches typically treat images as static inputs, limiting the model's ability to actively explore visual information. To address this limitation, we implement image operation tools that enable agents to zoom into specific regions, select key frames, and perform other visual manipulations. This approach, following Pixel-Reasoner [57], enhances reasoning capabilities over dense visual data.

Table 7: Performance comparison of vision-language models on agentic visual tool-use tasks. Accuracy (%) is reported for all benchmarks. The best and the second-best results in each column are shown in **bold** and underlined, respectively.

| Model | Pixel | V-Star | InfoVQA | TallyQA | AVG. |
|---|---|---|---|---|---|
| Pixel-Reasoner-7B | 64.00 | 84.29 | 74.37 | 73.69 | 74.05 |
| **Pixel-Reasoner-7B [57] → Qwen2.5-VL-3B-Instruct + SFT [28]** | | | | | |
| Qwen2.5-VL-3B-Instruct + SFT | 46.00 | 71.20 | 34.88 | 56.91 | 52.25 |
| + GRPO [28] | **60.00** | 76.96 | 59.47 | 60.56 | 64.25 |
| + Standard OPD | 52.00 | 76.55 | 61.09 | 64.34 | 63.27 |
| + REOPOLD | 57.00 | **77.43** | **63.12** | **65.43** | **65.75** |

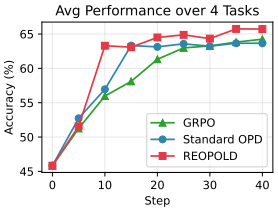



Figure 10: **Average score by training step.** REOPOLD outperforms all baselines at 50% training.

**Setup.** We implement our proposed method based on the VerlTool [28] framework. We adopt the SFTed Qwen2.5-VL-3B-Instruct [28] as the student policy and Pixel-Reasoner-7B [57] as the teacher. We use the official training dataset from Pixel-Reasoner, comprising 15K queries from InfographicVQA, supplemented by additional public datasets. Detailed training setup is described in Appendix D.

**Evaluation.** Following Su et al. [57], we evaluate our model and baselines on four representative multi-modal benchmarks using nucleus sampling with temperature of 1.0 and a top-p of 1.0: test-split of Pixel-Reasoner, V-Star [66], InfographicVQA [45], and TallyQA [1]. This selection offers a wide spectrum of visual understanding tasks, ranging from fine-grained object recognition to high-level reasoning in both static and dynamic scenarios.

**Results.** As reported in Table 7, REOPOLD outperforms both standard OPD and GRPO, notably surpassing the latter even when it utilizes the complex reward designs proposed by Su et al. [57]. Although GRPO achieves slightly higher accuracy on the Pixel test split, REOPOLD demonstrates superior performance across other benchmarks, indicating stronger generalization capabilities. Furthermore, we show in Figure 10 that REOPOLD exhibits better sample efficiency compared to standard OPD and GRPO. Unlike traditional RL approaches that necessitate intricate reward engineering for sophisticated agentic tasks, REOPOLD can be applied directly.

## 5 Conclusion

We studied sampled-token OPD through the lens of selective credit allocation. Standard OPD applies teacher credits uniformly despite their imbalanced structure, causing inefficient updates and premature suppression of useful reasoning paths. To address this, we introduced REOPOLD, a unified framework that regulates teacher feedback along three axes: entropy-guided token selection focuses supervision on uncertain decisions, bounded negative credit controls update magnitude, and an exploration-to-refinement schedule changes the allocation rule over training. Empirically, REOPOLD improves OPD across mathematical, visual, and agentic tool-use reasoning, and remains effective under cross-vocabulary distillation, self-distillation, longer training, and test-time scaling. These results suggest selective credit allocation is a practical principle for robust reasoning distillation in compact models.

## References


[1] Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8076–8084, 2019.

[2] 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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Mateo Espinosa Zarlenga, Lucie Charlotte Magister, Alberto Tonda, Pietro Lio, Frederic Precioso, Mateja Jamnik, and Giuseppe Marra. Interpretable neural-symbolic concept reasoning. In *ICML 2023 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, 2023. URL https://openreview.net/forum?id=oRj82I2apn.

[5] Nicolas Boizard, Kevin El Haddad, CELINE HUDELOT, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for LLMs. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=bwRxXiGO9A.

[6] Diego Calanzone, Stefano Teso, and Antonio Vergari. Logically consistent language models via neuro-symbolic integration. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=7PGluppo4k.

[7] Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

[8] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.

[9] Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*, 2022.

[10] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

[11] Quy-Anh Dang and Chris Ngo. Reinforcement learning for reasoning in small llms: What works and what doesn't. *arXiv preprint arXiv:2503.16219*, 2025.

[12] Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=uZbZb058bA.

[13] Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. Puzzle solving using reasoning of large language models: A survey. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11574–11591, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.646. URL https://aclanthology.org/2024.emnlp-main.646/.

[14] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=5h0qf7IBZZ.

[15] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385, 2024.

[16] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*, 2023.

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

[18] Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.507. URL https://aclanthology.org/2023.emnlp-main.507/.

[19] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

[20] Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

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

[22] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[23] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.830. URL https://aclanthology.org/2023.acl-long.830/.

[24] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

[25] 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 Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

[26] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.67. URL https://aclanthology.org/2023.findings-acl.67/.

[27] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[28] Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025.

[29] Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=J5i09faOOf.

[30] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[31] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=lsHZNNoC7r.

[32] Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rc65N9xIrY.

[33] Jongwoo Ko, Sungnyun Kim, Sungwoo Cho, and Se-Young Yun. Flex-judge: Text-only reasoning unleashes zero-shot multimodal evaluators. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=v6kyF3S7dM.

[34] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=IFXTZERXdM7.

[35] Qing Li, Yixin Zhu, Yitao Liang, Ying Nian Wu, Song-Chun Zhu, and Siyuan Huang. Neural-symbolic recursive machine for systematic generalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FWJAmwE0xH.

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

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[38] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

[39] Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Qizhe Shieh. Noisyrollout: Reinforcing visual reasoning with data augmentation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=9zD2i7YRot.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[41] Kevin Lu et al. On-policy distillation. https://thinkingmachines.ai/blog/on-policy-distillation/, 2025. Thinking Machines Blog, accessed on 2025-10-27.

[42] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021.

[43] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

[44] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 2025.

[45] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[46] Carlos Miguel Patiño, Kashif Rasul, Quentin Gallouédec, Ben Burtenshaw, Sergio Paniego, Vaibhav Srivastav, Thibaud Frere, Ed Beeching, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Unlocking on-policy distillation for any model family. https://huggingface.co/spaces/HuggingFaceH4/on-policy-distillation/, 2025. HuggingFace Blog, accessed on 2025-10-29.

[47] Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

[48] Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20023–20070, 2025.

[49] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain yourself! leveraging language models for commonsense reasoning. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1487. URL https://aclanthology.org/P19-1487/.

[50] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

[51] Manuel Alejandro Borroto Santana, Katie Gallagher, Antonio Ielo, Irfan Kareem, Francesco Ricca, and Alessandra Russo. Question answering with llms and learning from answer sets. *Theory and Practice of Logic Programming*, pages 1–25, 2025.

[52] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[53] 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.

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

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

[56] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

[57] Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*, 2025.

[58] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[59] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.

[60] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

[61] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

[62] Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, et al. Perception-aware policy optimization for multimodal reasoning. *arXiv preprint arXiv:2507.06448*, 2025.

[63] 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. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[64] Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Kang Liu, and Jun Zhao. Mastering symbolic operations: Augmenting language models with compiled neural networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9nsNyN0vox.

[65] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[66] Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094, 2024.

[67] Taiqiang Wu, Chaofan Tao, Jiahao Wang, Runming Yang, Zhe Zhao, and Ngai Wong. Rethinking Kullback-Leibler divergence in knowledge distillation for large language models. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5737–5755, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.383/.

[68] Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

[69] Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, et al. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math. *arXiv preprint arXiv:2504.21233*, 2025.

[70] Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. Kdrl: Post-training reasoning llms via unified knowledge distillation and reinforcement learning. *arXiv preprint arXiv:2506.02208*, 2025.

[71] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

[72] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[73] 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.

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

[75] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[76] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[77] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024.

[78] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=2cczgOfMP4.

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

[80] Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

[81] Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Wjp1AYB8lH.

[82] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

# A Related Work

**Reasoning models.** Reasoning models represent a distinct class of machine learning systems designed to execute structured, logical, and multi-step inference over input queries [8, 9, 26]. In contrast to standard predictive models that rely on direct input-to-output mapping, reasoning models emulate human cognitive processes by integrating learned knowledge with stepwise deduction [9], chain-of-thought processing [63], or symbolic manipulation [64]. These capabilities are essential for tasks involving complex problem-solving [13, 61], question answering [51, 78], planning [18, 75], and commonsense inference [49, 81], where a single forward pass is often insufficient. Recent advancements have increasingly integrated reasoning into LLMs and neuro-symbolic architectures [6, 35]. This integration allows models to decompose complex problems into intermediate steps, verify logical consistency [6], and generate interpretable solutions [4, 76]. Fundamentally, reasoning models ensure that outputs are both accurate and justifiable, emphasizing the rationale behind a decision as much as the decision itself.

**Policy optimization for reasoning models.** Recent advancements in policy optimization focus on enhancing the sample efficiency, stability, and reasoning depth of LLMs. While initial approaches relied on standard outcome-based RL, recent works demonstrate that scaling RL on smaller architectures, as seen in DeepScaleR [44] and Skywork OpenReasoner [20], can achieve performance rivaling proprietary frontiers like OpenAI-o1. To improve algorithmic stability beyond basic group-relative updates, GSPO [82] introduces step-level granularity for precise credit assignment, whereas GMPO [80] adopts a group-wise minimax formulation to bolster robustness against distribution shifts. Extending this to hybrid training objectives, KDRL [70] proposes a unified framework that synergizes knowledge distillation with RL, effectively balancing teacher supervision and self-exploration. Addressing the critical balance between exploration and exploitation, the Entropy Mechanism [10] dynamically regulates policy entropy to prevent premature convergence, while LUFFY [71] improve optimization efficiency by effectively leveraging diverse, off-policy trajectories. Furthermore, emphasizing the generation of extended reasoning chains, ProRL [38] explicitly incentivizes prolonged thought process to expand the models' reasoning boundaries, a capability that underpins state-of-the-art large-scale systems such as MiniMax-M1 [7].

**On-policy distillation for reasoning models.** Traditional knowledge distillation typically relies on offline datasets generated by a teacher model, which creates a distribution mismatch as the student's policy drifts from the static training data [23, 25, 33]. To bridge this gap, on-policy distillation aligns the student with the teacher's distribution by training on trajectories sampled directly from the student's current policy [2, 14, 31]. This paradigm is particularly critical for reasoning tasks, where models must learn to recover from their own logical errors rather than merely mimicking perfect teacher paths [41, 46]. Recently, there has been a growing movement to adapt on-policy distillation for reasoning tasks. Frontier models like Qwen3 [73] utilize iterative on-policy feedback to refine long-chain reasoning capabilities, while MiMo-V2-Flash [68] demonstrates that such methods achieve superior compute-efficiency by targeting "hard" examples where the student's confidence diverges from the teacher. Furthermore, self-distillation [54, 79], where the same model serves as both teacher and student and the teacher policy is conditioned on privileged information such as the ground-truth solution, has recently gained popularity.

## B Mathematical Derivations

### B.1 A Fixed-credit View

The teacher-derived token credit $R_{i,t}(\theta)$ depends on $\theta$, since it contains the student probability in the denominator. Following the stop-gradient formulation, we treat this credit as a fixed teacher-derived signal during each update. Let $\mathcal{J}^{\text{sg}}_{\text{RKL}}$ denote the corresponding objective. Then, the expected gradient satisfies

$$\nabla_\theta \mathcal{J}_{\text{RKL}}(\theta) = \nabla_\theta \mathcal{J}^{\text{sg}}_{\text{RKL}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_i |o_i|}\sum_{i,t} \rho_{i,t}(\theta) R_{i,t}(\theta) \nabla_\theta \log \pi_\theta(o_{i,t}|q, o_{i,<t})\right]. \tag{15}$$

We derive the gradient of RKL. From Eq. 1, the RKL objective and its gradient are:

$$\mathcal{J}_{\text{RKL}}(\theta) = \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \rho_t(\theta) R_t(\theta)\right], \tag{16}$$

where $\rho_t(\theta) = \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}$ and $R_t(\theta) = \log \frac{\pi_T(o_t|q,o_{<t})}{\pi_\theta(o_t|q,o_{<t})}$. Then following holds,

$$\begin{aligned}
\nabla_\theta \mathcal{J}_{\text{RKL}}(\theta) &= \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \rho_t(\theta)\nabla_\theta R_t(\theta) + R_t(\theta)\nabla_\theta \rho_t(\theta)\right], \\
&= \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \rho_t(\theta) R_t(\theta)\nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] - \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \frac{\nabla_\theta \pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}\right], \\
&= \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \rho_t(\theta) R_t(\theta)\nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] - \nabla_\theta \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}\right], \\
&= \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \rho_t(\theta) R_t(\theta)\nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] \\
&\quad - \nabla_\theta \mathbb{E}_{q\sim Q, o\sim\pi_\theta(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}\cdot\frac{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}{\pi_\theta(o_t|q,o_{<t})}\right], \\
&= \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \rho_t(\theta) R_t(\theta)\nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right] - \nabla_\theta \mathbb{E}_{q\sim Q, o\sim\pi_\theta(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} 1\right], \\
&= \mathbb{E}_{q\sim Q, o\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} \rho_t(\theta) R_t(\theta)\nabla_\theta \log \pi_\theta(o_t|q,o_{<t})\right]
\end{aligned}$$

This justifies treating $R_t$ as a fixed token-level credit within each update. The fixed-credit view separates how teacher credit is computed from how the resulting sampled-token credits are allocated during optimization. Our work focuses on the latter: standard RKL allocates update budget uniformly across sampled tokens, regardless of whether their credits are informative, redundant, or extreme.

### B.2 Logit-Level Update-Energy Proxy

Let $p_\theta(\cdot \mid x) = \pi_\theta(\cdot \mid q, o_{<t})$ denote the student distribution at context $x = (q, o_{<t})$, and let $y = o_t$ be the sampled token. For a sampled-token objective with scalar credit $R_{i,t}(\theta)$, the token-level likelihood-ratio update with respect to the logits $z$ is proportional to

$$g_{i,t} = R_{i,t}(\theta)\nabla_z \log p_\theta(y \mid x).$$

For a softmax distribution, we have

$$\nabla_z \log p_\theta(y \mid x) = e_y - p_\theta(\cdot \mid x),$$

where $e_y$ is the one-hot vector of the sampled token. Therefore,

$$\|g_{i,t}\|_2^2 = R_{i,t}(\theta)^2 \|e_y - p_\theta(\cdot \mid x)\|_2^2.$$

Taking expectation over $y \sim p_\theta(\cdot \mid x)$ gives

$$\mathbb{E}_{y \sim p_\theta}\left[\|e_y - p_\theta(\cdot \mid x)\|_2^2\right] = 1 - \|p_\theta(\cdot \mid x)\|_2^2.$$

Thus, the expected squared logit-level update induced by token credit $R_{i,t}(\theta)$ is

$$\mathbb{E}_{y \sim p_\theta}\left[\|g_{i,t}\|_2^2\right] = R_{i,t}(\theta)^2 \left(1 - \|p_\theta(\cdot \mid x)\|_2^2\right).$$

This motivates the update-energy proxy used in Section 2:

$$E_{i,t}(\theta) = R_{i,t}(\theta)^2 \left(1 - \|\pi_\theta(\cdot \mid q, o_{i,<t})\|_2^2\right).$$

### B.3 Softmax Score Magnitude

We prove Remark 3.1. Let $p = \mathrm{softmax}(z)$ over vocabulary $\mathcal{V}$, and denote by $e_y$ the one-hot vector for token $y$. For each coordinate $v \in \mathcal{V}$,

$$\frac{\partial}{\partial z_v} \log p_y = \mathbb{I}[v = y] - p_v,$$

and therefore $\nabla_z \log p_y = e_y - p$. Taking expectation over $y \sim p$ gives

$$\begin{aligned}
\mathbb{E}_{y \sim p}\left[\|\nabla_z \log p_y\|_2^2\right] &= \sum_{y \in \mathcal{V}} p_y \|e_y - p\|_2^2 \\
&= \sum_{y \in \mathcal{V}} p_y \left(1 - 2p_y + \|p\|_2^2\right) \\
&= 1 - \|p\|_2^2.
\end{aligned}$$

Finally, since $-\log u \geq 1 - u$ for $u \in (0, 1]$, we have $-p_v \log p_v \geq p_v(1 - p_v)$ for every $v$ with $p_v > 0$. Summing over $v$ and using continuity at $p_v = 0$ yields

$$H(p) = -\sum_{v \in \mathcal{V}} p_v \log p_v \geq \sum_{v \in \mathcal{V}} p_v(1 - p_v) = 1 - \|p\|_2^2.$$

### B.4 Derivation of Bounded Threshold

Here, we derive the relationship between the standard log-likelihood ratio and the convex mixture ratio used to motivate our clipping threshold. Since the logarithm is a concave function, for $\forall \lambda \in [0, 1)$, Jensen's inequality implies:

$$(1-\lambda) \cdot \log \pi_T(o_t|q, o_{<t}) + \lambda \cdot \log \pi_\theta(o_t|q, o_{<t}) \leq \log\left[(1 - \lambda) \cdot \pi_T(o_t|q, o_{<t}) + \lambda \cdot \pi_\theta(o_t|q, o_{<t})\right]. \tag{17}$$

To isolate the log-likelihood ratio $R_{i,t}(\theta) = \log \frac{\pi_T(o_t|q,o_{<t})}{\pi_\theta(o_t|q,o_{<t})}$, we subtract $\log \pi_\theta(o_t|q, o_{<t})$ from both sides and divide by $(1 - \lambda)$:

$$R_{i,t}(\theta) = \log \frac{\pi_T(o_t|q, o_{<t})}{\pi_\theta(o_t|q, o_{<t})} \leq \frac{1}{1 - \lambda} \log \frac{(1 - \lambda)\pi_T(o_t|q, o_{<t}) + \lambda \pi_\theta(o_t|q, o_{<t})}{\pi_\theta(o_t|q, o_{<t})}. \tag{18}$$

This inequality upper-bounds the log-ratio between the teacher and student policies by the log-ratio induced by a convex mixture of the two.

**Comparison to Skew RKL [31].** Although the RHS of Eq. (17) is identical to skew RKL [31], our approach differs in application by using the bound $\frac{\log \lambda}{1-\lambda}$ strictly as a clipping threshold rather than modifying the global objective. Figure 11 demonstrates that while Skew RKL is highly sensitive to $\lambda$ (e.g., dropping significantly at $\lambda = 0.7$), our method remains robust. Remarkably, our lowest accuracy surpasses even the peak performance of Skew RKL. This confirms that selectively clipping heavy-tailed outliers stabilizes training more effectively than altering the global divergence.

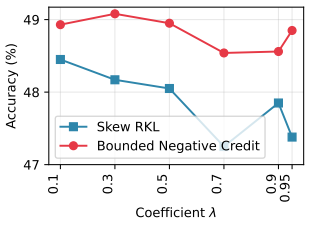



Figure 11: **Sensitivity analysis of $\lambda$.**

## C Full Algorithm

Here, we provide the algorithm for REOPOLD in Algorithm 1.

**Algorithm 1** REOPOLD

**Input:** student policy $\pi_\theta$, teacher policy $\pi_T$, query set $\mathcal{Q}$
**Hyperparameters:** total steps $K$, switch step $T_{\text{switch}}$, bounding coefficient $\lambda$, top-entropy fraction $\beta$, learning rate $\eta$
**Output:** trained student model $\pi_\theta$
Initialize $\theta_{\text{old}} \leftarrow \theta$
**for** $k = 1$ to $K$ **do**
  Sample a batch of queries $\mathcal{B} \sim \mathcal{Q}$
  Generate $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid q)$ for each $q \in \mathcal{B}$
  Compute $R_{i,t}(\theta) \leftarrow \log \frac{\pi_T(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})}$
  Clip $\hat{R}^{\lambda}_{i,t}(\theta) \leftarrow \max\left(\text{sg}(R_{i,t}(\theta)), \frac{\log\lambda}{1-\lambda}\right)$ ▷ Eq. (11)
  **if** $k < T_{\text{switch}}$ **then**
    *Phase I: Exploration (Credit-based Filtering)*
    Set mask $M^{(k)}_{i,t} \leftarrow \mathbb{I}\left[R_{i,t}(\theta) \geq \frac{\log\lambda}{1-\lambda}\right]$ ▷ Eq. (13)
  **else**
    *Phase II: Refinement (Entropy-Guided Credit Allocation)*
    Compute entropy $H^i_t$ for each token $o_{i,t}$ in batch
    Compute $\tau_\beta$ as the $(1-\beta)$-quantile of $\{H^i_t\}$
    Set mask $M^{(k)}_{i,t} \leftarrow \mathbb{I}\left[H^i_t \geq \tau_\beta\right]$ ▷ Eq. (14)
  **end if**
  Compute gradients $\nabla_\theta \mathcal{J}_{\text{REOPOLD}}$ using Eq. (5)
  Update parameters $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{J}_{\text{REOPOLD}}$
  Update old policy parameters $\theta_{\text{old}} \leftarrow \theta$
**end for**

## D Detailed Experimental Setup

Section 4 details the experimental setup. We list the hyperparameter values for each setting in Table 8. Sensitivity analyses for the additive hyperparameters $\lambda$ and $\beta$ are presented in Figure 11 and Figure 6, respectively. Regarding the switch step $T_{\text{switch}}$ introduced in Section 3.3, we did not perform specific hyperparameter tuning; instead, we set the switch step to approximately one-quarter to one-third of the total training horizon, without extensive task-specific tuning. For $K$ and $\eta$, we adopted the same values as our baselines, where $\eta$ was determined via hyperparameter tuning based on vanilla RKL results.

Table 8: Hyperparameter values used in REOPOLD experiments in Section 4.

| | **Math** (Section 4.1) | | **Visual** (Section 4.2) | | **Agentic** (Section 4.4) |
|---|---|---|---|---|---|
| Model Size | 1.5B | 7B | 3B | 7B | 3B |
| Total step $K$ | 300 | 200 | 60 | 60 | 40 |
| Learning rate $\eta$ | $1 \times 10^{-5}$ | $3 \times 10^{-6}$ | $5 \times 10^{-6}$ | $5 \times 10^{-6}$ | $1 \times 10^{-6}$ |
| Clipping coefficient $\lambda$ | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| Top-entropy fraction $\beta$ | 0.2 | 0.2 | 0.2 | 0.2 | 0.4 |
| Switch step $T_{\text{switch}}$ | 100 | 50 | 20 | 20 | 10 |

**Math reasoning.** We employ Verl [55] for OPD. During rollout, we sample $n = 8$ responses per prompt with a maximum response length of 8192 and a sampling temperature of 1.0. The global batch size is set to 128 with a mini-batch size of 32, resulting in 4 gradient updates per rollout step. The student policy is trained for 300 iterations. We utilize the AdamW optimizer [40] with a constant learning rate of $1 \times 10^{-5}$. All training runs are conducted on a single node equipped with 8× NVIDIA H100 80GB GPUs, requiring approximately 200 and 312 GPU hours for the 1.5B and 7B models, respectively. For evaluation, we use maximum response length of 32768 and a sampling temperature of 0.6 and top-p of 0.95. Our evaluation protocol follows the setup established by Qwen2.5-Math [72] by using Verl [55] for implementation. We employ training[1] and evaluation[2] data from LUFFY [71].

**Visual reasoning.** For visual tasks, we conduct OPD using Verl [55]. We generate $n = 12$ responses for each prompt, enforcing a maximum response length of 2048 and a sampling temperature of 1.0. Following the protocol in Liu et al. [39], the models are trained for 60 iterations with a batch size of 128 and a mini-batch size of 64 (equating to 2 gradient updates per step). Optimization is performed via AdamW [40] with a learning rate of $5 \times 10^{-6}$. Using the same hardware setup (a single 8× NVIDIA H100 node), the training takes roughly 20 and 24 GPU hours for the 3B and 7B models. For evaluation, we use maximum response length of 8192 and a sampling temperature of 0.6 and top-p of 0.95 for nucleus sampling. We utilize the training[3] and evaluation[4] data from NoisyRollout [39].

**Agentic reasoning with visual tool-use.** We implement OPD and RL based on the VerlTool framework [28]. For rollout, the policy samples $n = 8$ trajectories per prompt with a maximum response length of 8192 and a temperature of 1.0. We set the maximum number of tool-use rounds to 2. We maintain a batch size of 128 and a mini-batch size of 64, corresponding to 2 gradient updates per rollout step. The student policy undergoes training for 40 iterations using the AdamW optimizer [40] with a constant learning rate of $1 \times 10^{-6}$. The entire process consumes approximately 120 GPU hours on a single node with 8× NVIDIA H100 80GB GPUs. Following Su et al. [57], we set the maximum number of tool-use rounds to 5 during evaluation, with a maximum response length of 8192, a sampling temperature of 1.0, and a top-p of 1.0. For training, we utilize the PixelReasoner-RL dataset[5]. For evaluation, we employ the InfoVQA[6], TallyQA[7], and VStar[8] datasets provided by the PixelReasoner [57].

[1] https://huggingface.co/datasets/Elliott/Openr1-Math-46k-8192
[2] https://github.com/ElliottYan/LUFFY/blob/main/data/valid.parquet
[3] https://huggingface.co/datasets/xyliu6/geometry3k
[4] https://huggingface.co/datasets/xyliu6/noisyrollout_evaluation_data
[5] https://huggingface.co/datasets/TIGER-Lab/PixelReasoner-RL-Data
[6] https://huggingface.co/datasets/JasperHaozhe/InfoVQA-EvalData-PixelReasoner
[7] https://huggingface.co/datasets/JasperHaozhe/TallyQA-EvalData-PixelReasoner
[8] https://huggingface.co/datasets/JasperHaozhe/VStar-EvalData-PixelReasoner

Table 9: Performance comparison of Qwen2.5-VL-3B-Instruct trained with various teacher models on visual reasoning reasoning and perception benchmarks. Accuracy (%) is reported for all benchmarks.

| | RKL | REOPOLD | Δ% |
|---|---|---|---|
| **Qwen2.5-VL-7B-Instruct + NoisyRollout** | | | |
| **Geo3K** | 49.75 | 51.41 | +1.66 |
| **MathVerse** | 41.66 | 44.27 | +2.61 |
| **MathVision** | 23.33 | 24.01 | +0.68 |
| **MathVista** | 62.20 | 63.10 | +0.90 |
| **WeMath** | 64.83 | 65.66 | +0.83 |
| **Hallusion** | 61.30 | 62.67 | +1.37 |
| **AVG.** | 50.51 | 51.85 | +1.34 |
| **Qwen2.5-VL-32B-Instruct + NoisyRollout‡** | | | |
| **Geo3K** | 43.93 | 45.76 | +1.83 |
| **MathVerse** | 41.99 | 43.22 | +1.23 |
| **MathVision** | 25.13 | 25.79 | +0.66 |
| **MathVista** | 63.60 | 64.20 | +0.60 |
| **WeMath** | 64.02 | 64.66 | +0.64 |
| **Hallusion** | 63.72 | 64.35 | +0.63 |
| **AVG.** | 50.40 | 51.33 | +0.93 |

Table 10: Performance comparison of Qwen2.5-VL-3B-Instruct trained for 60 and 300 training steps on visual reasoning reasoning and perception benchmarks. Accuracy (%) is reported for all benchmarks.

| | RKL | REOPOLD | Δ% |
|---|---|---|---|
| **60 Training Steps (3B)** | | | |
| **Geo3K** | 48.09 | 50.58 | +2.49 |
| **MathVerse** | 33.25 | 46.40 | +13.15 |
| **MathVision** | 24.78 | 26.39 | +1.61 |
| **MathVista** | 62.50 | 61.50 | -1.00 |
| **WeMath** | 64.48 | 64.60 | +0.12 |
| **Hallusion** | 60.83 | 63.62 | +2.79 |
| **AVG.** | 48.99 | 52.18 | +3.19 |
| **300 Training Steps (3B)** | | | |
| **Geo3K** | 49.08 | 51.08 | +2.00 |
| **MathVerse** | 46.60 | 47.79 | +1.19 |
| **MathVision** | 26.16 | 27.44 | +1.28 |
| **MathVista** | 67.20 | 66.30 | -0.90 |
| **WeMath** | 66.43 | 67.18 | +0.75 |
| **Hallusion** | 65.19 | 66.35 | +1.16 |
| **AVG.** | 53.44 | 54.36 | +0.92 |

Table 11: Performance comparison of Qwen2.5-VL-7B-Instruct trained for 60 and 300 training steps on visual reasoning reasoning and perception benchmarks. Accuracy (%) is reported for all benchmarks.

| | RKL | REOPOLD | Δ% |
|---|---|---|---|
| **60 Training Steps (7B)** | | | |
| **Geo3K** | 51.75 | 53.58 | +1.83 |
| **MathVerse** | 47.71 | 51.43 | +3.72 |
| **MathVision** | 28.79 | 29.21 | +0.42 |
| **MathVista** | 71.27 | 72.40 | +1.13 |
| **WeMath** | 70.06 | 69.77 | -0.29 |
| **Hallusion** | 69.51 | 70.14 | +0.63 |
| **AVG.** | 56.51 | 57.76 | +1.25 |
| **300 Training Steps (7B)** | | | |
| **Geo3K** | 49.42 | 53.58 | +4.16 |
| **MathVerse** | 50.72 | 51.97 | +1.25 |
| **MathVision** | 29.74 | 31.12 | +1.38 |
| **MathVista** | 71.20 | 73.60 | +2.40 |
| **WeMath** | 69.43 | 71.84 | +2.41 |
| **Hallusion** | 69.72 | 70.87 | +1.15 |
| **AVG.** | 56.71 | 58.83 | +2.12 |

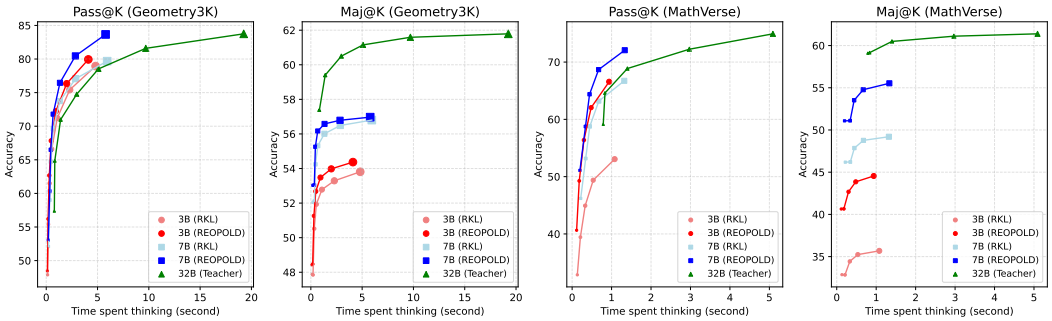



Figure 12: **Extended results of Figure 1(b).** We visualize the accuracy (Pass@$K$ and Maj@$K$) against inference latency as the sample budget $K$ increases (up to 64 for Geometry3K; 16 for MathVerse). REOPOLD (solid lines) consistently yields a better trade-off than the RKL baseline (faded lines). Notably, the 7B student matches or beats the 32B teacher's accuracy with significantly lower latency, confirming the efficiency of our distillation.

## E Additional Analyses and Discussions

In this section, we provide comprehensive analyses to offer a deeper understanding of the inner workings and robustness of REOPOLD. Unless otherwise specified, all experiments follow the visual reasoning evaluation protocols introduced in Section 4.2.

**Training with different teacher.** We further evaluate our approach by distilling from different teacher models to assess generalization. As detailed in Table 9, we utilize Qwen2.5-VL-7B-Instruct and 32B-Instruct models—fine-tuned on Geometry3K and MMK12 respectively via NoisyRollout [39]—as teachers. The results demonstrate the consistent effectiveness of REOPOLD. Compared to the RKL baseline, our method yields uniform improvements across all six benchmarks for both teacher settings. Specifically, we observe an average accuracy gain of 1.34% with the 7B teacher and 0.93% with the 32B teacher. This confirms that REOPOLD is robust to variations in teacher architecture and domain-specific expertise, reliably enhancing the student's visual reasoning and perception capabilities.

**Scalability with longer training.** We further evaluate REOPOLD under an extended training setting with 300 optimization steps and a combined training dataset consisting of Geometry3K and MMK12, using Qwen2.5-VL-3B-Instruct (Table 10) and Qwen2.5-VL-7B-Instruct (Table 11). As shown in

Table 10 and Table 11, extending the training horizon generally improves REOPOLD's performance, while standard OPD exhibits less consistent gains, including degradation on some benchmarks. These results highlight the superior scalability of REOPOLD under longer training. It outperforms the RKL baseline in both model sizes, achieving the highest average accuracy of 54.36% with the 3B model and 58.83% with the 7B model. These results indicate that our method is capable of continuously refining its policy given more compute and data, leading to robust improvements in both visual reasoning and perception tasks.

**Extended test-time scaling results.** We provide a comprehensive evaluation of the test-time scaling capabilities of REOPOLD on Geometry3K [42] and MathVerse [77] benchmarks. Figure 12 demonstrates that the superior scaling of REOPOLD is not limited to coverage metrics (Pass@$K$). Both the 3B and 7B models maintain a consistent lead in Maj@$K$, a metric that measures consensus robustness. This confirms that our method fundamentally increases the probability of correct reasoning chains, rather than merely generating over diverse "lucky guess" to boost Pass@$K$.

**Training time breakdown.** We analyze training wall-clock time to quantify computational overhead as shown in Figure 13. Contrary to concerns about the teacher's cost in on-policy distillation, our breakdown reveals it accounts for only a moderate fraction (8%–22%) of the total step. Crucially, this relative cost is inversely proportional to generation length. In long-context math tasks (8192 tokens), student generation dominates the runtime (77.3%), rendering the teacher's impact marginal (8.2%). Even in shorter visual reasoning tasks (2048 tokens), where the teacher's share rises to 21.8%, student generation remains the largest component of runtime rather than teacher supervision.

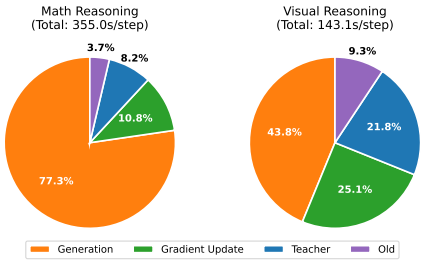



Figure 13: Breakdown of training wall-clock time per step.

**Extension to Top-K OPD.** We further compare REOPOLD with GKD [2] and DistiLLM [31] which are full-vocabulary OPD. As shown in Table 12, both GKD and DistiLLM with full vocabulary incur out-of-memory (OOM) errors since they need to store the value for 150K vocab size for both student and the teacher models. To alleviate this cost, we apply the commonly used Top-K ($K = 5$) approximation [36]; however, this approximation is less effective and yields lower efficiency than sampled-token approaches. Nevertheless, under the same Top-5 setting, REOPOLD still outperforms Top-5 approximated GKD and DistiLLM, showing that its gains are not limited to sampled-token training. Importantly, applying either full-vocabulary scoring or the Top-5 approximation to REOPOLD does not bring meaningful improvement over sampled-token REOPOLD, suggesting that the proposed credit allocation strategy is already sufficient without costly vocabulary-level approximation.

Table 12: Comparison with full-vocabulary OPD and Top-K ($K = 5$) approximations. Full-vocabulary GKD and DistiLLM (DLM) runs incur OOM under our visual reasoning setup; therefore, we additionally report Top-5 approximations.

| | GKD/DLM | GKD† [2] | DLM† [31] | REOPOLD | REOPOLD † |
|---|---|---|---|---|---|
| **Geo3K** | OOM | 51.85 | 52.68 | **53.58** | 53.41 |
| **MathVerse** | OOM | 48.14 | 48.65 | **51.97** | 50.96 |
| **MathVision** | OOM | 29.90 | 30.12 | **31.12** | 30.94 |
| **MathVista** | OOM | 71.25 | 72.12 | 73.60 | **73.64** |
| **WeMath** | OOM | 70.87 | 70.92 | **71.84** | 71.80 |
| **Hallusion** | OOM | 70.21 | 70.37 | 70.87 | **71.52** |
| **AVG.** | OOM | 57.04 | 57.48 | **58.83** | 58.71 |

**Effect of the selection signal.** We compare different token-selection signals within OPD, along with GRPO and Beyond 80/20 [60]. As shown in Figure 14, student-entropy selection improves Geo3K accuracy from 45.2 to 50.8 at step 10 and from 51.8 to 53.8 at step 60, outperforming teacher-entropy selection and other baselines. Unlike EOPD [29], which introduces an additional forward-KL term, the teacher-entropy variant modifies only the selection signal. These results highlight the effectiveness of student entropy for selective credit allocation.

**Choice of the selection fraction.** We further examine the choice of $\beta$ by comparing fixed fractions with a calibration heuristic based on two-means clustering of token entropies from 16 sampled sequences. As shown in Figure 15, the calibrated fractions achieve comparable or slightly better performance than the best fixed choices across math, visual, and agentic reasoning. These results suggest that entropy-based calibration provides a practical heuristic for selecting $\beta$ without extensive tuning.

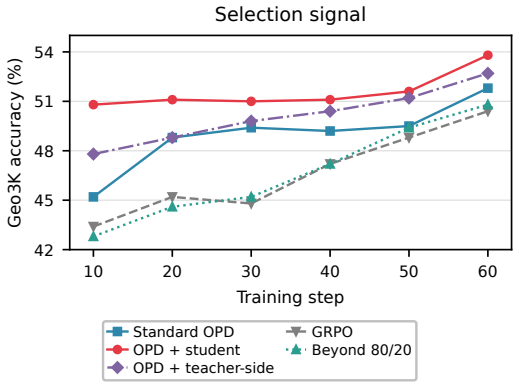



Figure 14: **Effect of token-selection signals.** Comparison of student-entropy and teacher-entropy selection within OPD. Student-entropy selection achieves the highest accuracy throughout training.

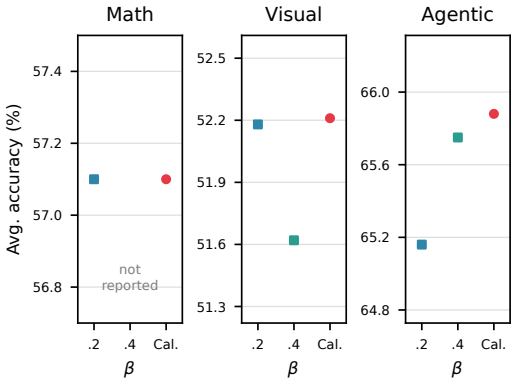



Figure 15: **Sensitivity to the selection fraction.** Comparison of fixed and calibrated selection fractions across math, visual, and agentic reasoning. Entropy-based calibration achieves competitive performance without extensive tuning.

**Effect of switch timing.** We analyze the sensitivity to switch timing by varying the switch fraction over $1/6, 1/3, 2/3$ for 60- and 300-step training. As shown in Figure 16, switching at one-third achieves the best performance at 60 steps, whereas two-thirds performs best among fixed schedules at 300 steps. Notably, the plateau-triggered schedule further improves Pass@$K$ at 300 steps, while the two-thirds switch achieves higher Avg@$K$. These results suggest that the preferred switch timing depends on the training horizon and evaluation metric.

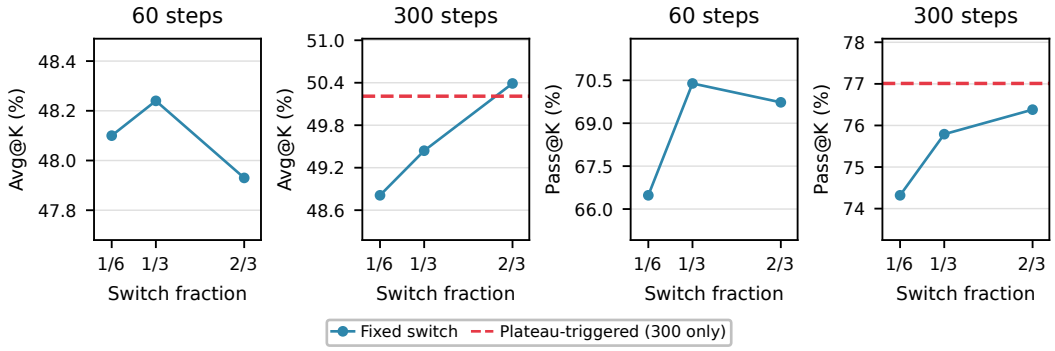



Figure 16: **Sensitivity to switch timing.** We compare fixed switch fractions at 60 and 300 training steps on Geo3K and MathVerse. Switching at one-third performs best at 60 steps, while two-thirds performs best among fixed schedules at 300 steps. Red dashed lines indicate the plateau-triggered results at 300 steps.

**Effect on response coverage.** We further examine the effectiveness of exploration-to-refinement scheduling by comparing REOPOLD with a variant that removes the initial exploration phase while retaining bounded credit and entropy-guided token selection. As shown in Figure 17, scheduling improves MathVerse Pass@16 by 10.54 and 9.08 points at 60 and 300 steps, respectively, while Avg@16 improves by only 0.46 and 1.16 points. These results suggest that the exploration phase primarily improves solution coverage, which is not fully captured by average accuracy.

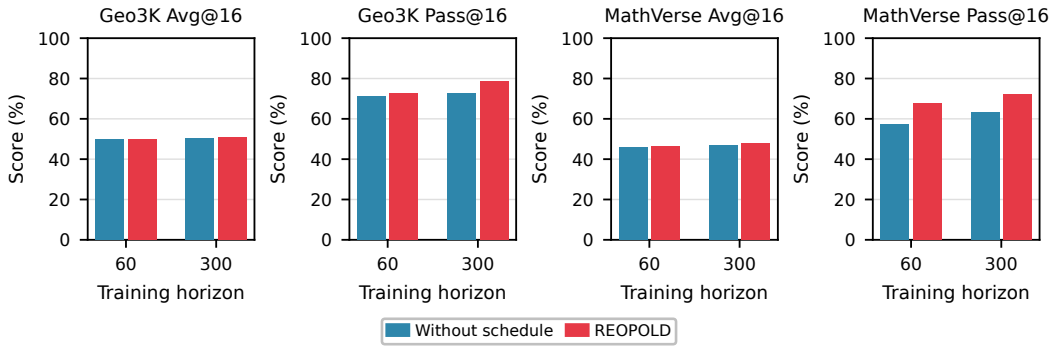
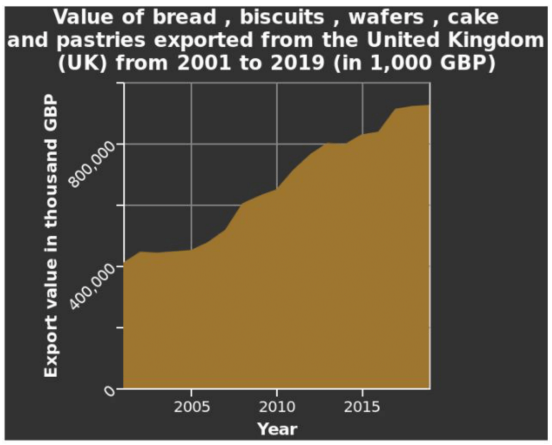



Figure 17: **Effect of exploration-to-refinement scheduling.** We compare REOPOLD with a variant that omits the initial exploration phase at 60 and 300 training steps. Scheduling consistently improves Pass@16 while yielding smaller gains in Avg@16, highlighting its contribution to solution coverage.

## F Qualitative Evaluation

Question: From 2001 to 2010, Is 2010 exported the most value of bread, biscuits, wafers, cakes and pastries from the UK? Your final answer can only be $\boxed{yes}$ or $\boxed{no}$.




---

[Qwen2.5-VL-3B-Instruct (REOPOLD)]

<think>
To determine if 2010 exported the most value... we need to analyze the graph step by step.
1. Identify the Time Period: The graph shows data from 2001 to 2019 ...
2. Analyze the Graph: The graph is a line graph ...
6. Conclusion: Based on the graph, the export value in 2010 is the highest ...
</think>

Final Answer: $\boxed{yes}$

---

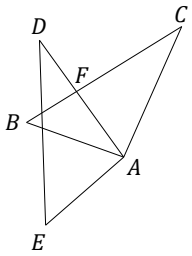
[Qwen2.5-VL-3B-Instruct (Standard OPD)]

### Step-by-Step Reasoning:
1. Identify the Time Period: The question asks about ...
4. Determine the Maximum Value: ... values are relatively close to 400,000 GBP for each year from 2001 to 2010. There is no clear indication that 2010 had a significantly higher ...
5. Conclusion: Therefore, it is not possible to definitively say ...

### Final Answer: $\boxed{No}$

Figure 18: **Qualitative comparison on Hallusion Bench.** While the baseline trained with Standard OPD suffers from visual perception degradation (hallucinating a flat trend despite the visual evidence), REOPOLD maintains robust visual grounding, accurately identifying the peak in the chart.

Question: As shown in the figure, angle C = 36.0, angle BAE = 60.0, then the degree of angle AFC is ()
Choices: A:84° B:80° C:60° D:90°




---

[Qwen2.5-VL-7B-Instruct (REOPOLD)]

<think>
To determine the measure of angle AFC, let's analyze the given information...
4. Analyze the angle AFC:
... Substituting the given values:

$$\text{Angle AFC} = 60^\circ + 36^\circ = 96^\circ$$

However, we need to re-evaluate the problem because the angle AFC is actually the angle formed by the intersection...
5. Re-evaluate using the properties of the angles:
... The correct measure of angle AFC is:

$$\text{Angle AFC} = 180^\circ - 60^\circ - 36^\circ = 84^\circ$$

</think>

$\boxed{A}$

---

[Qwen2.5-VL-7B-Instruct (Standard OPD)]

<think>
To determine the measure of angle AFC...
5. Consider the angle BAE:
... Therefore:

$$60^\circ = x + (144^\circ - x)$$

This equation is not directly useful for finding $x$, but it confirms the relationship...
...
8. Final calculation:
The correct measure of angle AFC is:

$$\text{Angle AFC} = 156^\circ - 60^\circ$$

$$\text{Angle AFC} = 90^\circ$$

</think>

$\boxed{90}$

Figure 19: **Qualitative comparison on MathVerse.** The baseline (Standard OPD) gets trapped in circular logic (Step 5) and hallucinates the final calculation (Step 8). Our model initially derives an incorrect value ($96^\circ$) but explicitly triggers a self-correction process ("However, we need to re-evaluate...") to reach the correct solution ($84^\circ$).

## G Limitations & Impact Statement

**Limitations.** A limitation of REOPOLD is that its gains may be smaller when the teacher and student are already closely matched, since most token-level credits become near zero and standard OPD is less unstable in this regime. In addition, REOPOLD improves the allocation and stability of teacher feedback but does not directly reduce the cost of teacher inference; when teacher evaluation dominates wall-clock training time, the practical speedup may be limited. Future work could adapt the allocation rule to the teacher-student gap and combine selective credit allocation with cheaper teacher scoring or caching strategies.

**Broader Impact.** This work improves the efficiency and stability of reasoning distillation for compact language models. By making sampled-token OPD more effective across mathematical, visual, and agentic tool-use tasks, REOPOLD may reduce the data and optimization cost required to train capable smaller models, lowering deployment cost and latency in resource-constrained settings. However, stronger compact models may also lower the barrier to misuse, particularly in automated or tool-use applications. These risks are not unique to REOPOLD and should be mitigated through safety evaluation, access control, and deployment monitoring.

## H Reproducibility Statement

For our experiments, we have clearly described the datasets used, the models fine-tuned, and the hyperparameters used (Section 4.1, Section 4.2, Section 4.4, and Appendix D). While all datasets and base models are publicly available, we provide a summary in Appendix D, along with relevant links. To further aid reproducibility, we provide a standalone implementation of our core algorithm in the supplementary material.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the central bottleneck, the proposed REOPOLD framework, and the empirical scope of the claims. The claims are supported by the method sections, theoretical derivations, and experiments in Sections 2–4 and Appendix E.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations & Impact Statement that discusses smaller gains when teacher and student are closely matched and the fact that REOPOLD does not directly reduce teacher-inference cost.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical statements are stated with their assumptions in the main text and Appendix B.1. Appendix B.1 provides the fixed-credit derivation, update-energy proxy, softmax score identity, and clipping-threshold derivation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 4.1, 4.2, and 4.4 describe the experimental protocols, models, datasets, and evaluation metrics. Appendix D reports hyperparameters, training lengths, sampling settings, hardware, GPU hours, and links to the datasets.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All datasets and base models used in the experiments are publicly available and linked in Appendix D, and we include a standalone implementation of the core algorithm in the supplementary material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main experimental sections describe the models, datasets, benchmarks, evaluation metrics, and decoding settings. Appendix D and Table 8 provide batch sizes, rollout counts, learning rates, total steps, optimizer details, and method hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main math and visual reasoning results in Tables 3 and 4 report mean $\pm$ standard deviation over 5 runs. These error bars capture variability across repeated experimental runs under the reported training and evaluation protocols.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D reports that experiments were run on a single node with 8 NVIDIA H100 80GB GPUs and gives approximate GPU hours for the math, visual, and agentic settings. Appendix E further analyzes wall-clock training-time components.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses public datasets, models, and benchmarks, does not involve human-subject experiments, and discusses possible misuse risks in the Limitations & Impact Statement. We are not aware of any aspect of the work that deviates from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Limitations & Impact Statement discusses positive impacts such as improving the efficiency and accessibility of compact reasoning models. It also discusses negative impacts, including reduced barriers to misuse in automated or tool-use applications, and mentions mitigation through safety evaluation, access control, and monitoring.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new high-risk pretrained models, image generators, or scraped datasets as part of the submission. The broader-impact discussion notes potential misuse risks of stronger compact models and points to deployment-level mitigations.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use publicly available models, datasets, benchmarks, and training frameworks. The original papers or sources are cited throughout the paper, and dataset links are provided in Appendix D. We respect the corresponding terms of use and will include any additional license information in the supplementary material when applicable.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper introduces a new algorithmic method but does not release new datasets, benchmark suites, or trained model artifacts as new assets. The core algorithm and experimental details are documented in the method section, Algorithm 1, and Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not conduct crowdsourcing experiments or research with human subjects. All experiments use existing public datasets, models, and benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject research, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The paper does not involve LLMs for the core method development in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.