RISE: RECURSIVE IMPROVEMENT VIA SELF-EXTRAPOLATING POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) provides dense, per-token supervision for language model post-training, but its effectiveness is bottlenecked by teacher quality: external teachers suffer from distribution mismatch, while self-distillation with privileged conditioning is limited by in-context learning capacity. We propose **RISE** (**R**ecursive **I**mprovement via **S**elf-**E**xtrapolating Policy Distillation), which constructs a synthetic teacher directly from the model's own RLVR training trajectory. By extrapolating the displacement between the current checkpoint and a trailing anchor—in parameter space or output logit space—RISE converts a sparse outcome-induced parameter update into a dense token-level target, without any external model or privileged conditioning. RISE combines RLVR and OPD in a complementary loop: outcome rewards ground the extrapolation toward correct reasoning, while the extrapolated teacher refines token-level decisions. Moreover, since the teacher is refreshed every iteration as the student improves, distillation becomes a recursive improvement mechanism rather than a one-shot compression step. Experiments spanning mathematical reasoning, multi-domain STEM, code generation, and multi-turn agentic tasks show that RISE outperforms RLVR-only training and on-policy self-distillation across all settings.

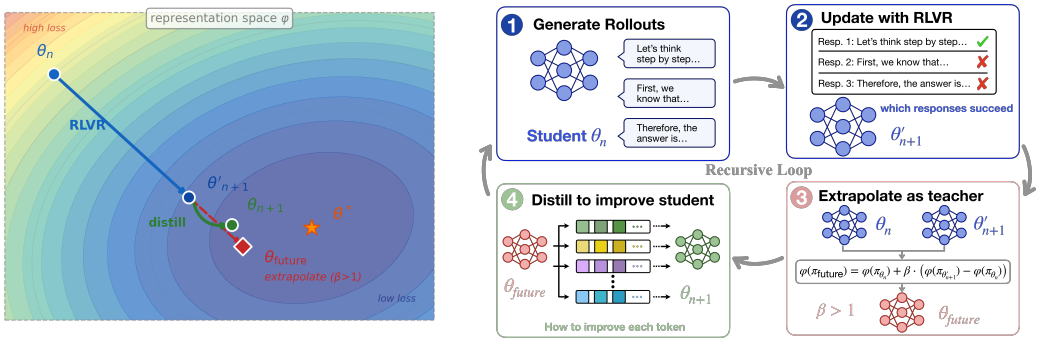
1 INTRODUCTION

A central aspiration of artificial intelligence is to build systems that can *recursively improve themselves*—becoming increasingly capable from their own experience with minimal human supervision (Yang et al., 2026c; Tao et al., 2024). For large language models (LLMs), this means updating parameters from self-generated data via reinforcement learning from human feedback or verifiable rewards (RLHF/RLVR) (Ouyang et al., 2022; Shao et al., 2024; Guo et al., 2025; Yu et al., 2026), self-play (Chen et al., 2024; Liu et al., 2025; Huang et al., 2025; Fang et al., 2026), or rejection sampling fine-tuning (Zelikman et al., 2022; Gulcehre et al., 2023; Xiong et al., 2025). All these approaches share a common limitation: the learning signal is *sequence-level*—an outcome reward, a binary accept/reject label, or a scalar preference—providing no guidance on *which tokens* were responsible for success or failure.

On-policy distillation (OPD) offers a richer signal for weight-based self-improvement. Rather than a single scalar per response, OPD provides dense per-token supervision: a teacher specifies a full next-token distribution at every position, telling the model not just *whether* its response was correct but *how* to improve each token-level decision (Song & Zheng, 2026). However, OPD's promise hinges on a critical question: *where does a reliable teacher come from?* External teachers suffer from distribution mismatch (Li et al., 2026c; Zhu et al., 2026); on-policy self-distillation (OPSD) with privileged conditioning often fails to reflect token-level correctness, due to limited in-context learning (ICL) capability and uninformative privileged information (Zhao et al., 2026b; Hübotter et al., 2026; Li et al., 2026b); and subsequent heuristic remedies—token-level gating (Xu et al., 2026b; Lu et al., 2026), divergence mixing (Jung et al., 2025; Jin et al., 2026), DAgger-style sampling (Li et al., 2026a; Zhao et al., 2026a), trajectory refinement (Yang et al., 2026b)—treat symptoms rather than the root cause. All these approaches accept a flawed teacher as given; none addresses the fundamental question: *what should the teacher be?*

We propose a shift in perspective. A natural candidate teacher for a policy π_θ is its own *converged future self*—the policy π^* that training is converging toward. While π^* is unknown, the training

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Extrapolation geometry. (b) The RISE training loop.

Figure 1: **RISE overview.** (a) RLVR updates $\theta_n \to \theta'_{n+1}$ (blue); extrapolation amplifies this displacement to construct θ_{future} (red); OPD distills θ_{future} into the student, yielding θ_{n+1} (green). (b) The training loop: RLVR grounds the direction, while OPD refines token-level decisions.

trajectory reveals the direction of convergence. Let φ map a policy to a vector space where linear operations are meaningful (e.g., logits or parameters). The displacement $\varphi(\pi_{\theta'}) - \varphi(\pi_\theta)$ between the current checkpoint θ' and an earlier anchor θ captures the direction of recent improvement. We can **extrapolate** this displacement to synthesize a future teacher:

$$\varphi(\pi_{\text{future}}) = \varphi(\pi_\theta) + \beta \cdot \big(\varphi(\pi_{\theta'}) - \varphi(\pi_\theta)\big), \quad \beta > 1. \tag{1}$$

When $\beta = 1$ the teacher equals the current checkpoint (no distillation signal); when $\beta > 1$ it amplifies the model's most recent update, projecting beyond its current state along the training trajectory. Recent analyses reveal that post-training updates are dominated by a low-rank subspace and evolve near-linearly (Cai et al., 2025; Wang et al., 2026a), making extrapolation along the training direction a principled approximation (§2). This construction eliminates the core pathologies of prior OPD: reduced distribution mismatch (the teacher extrapolates the student's own optimization path, so it remains nearby in weight and distribution space), no privileged conditioning, and no external model—only previous checkpoints, naturally available during training.

The construction above is iterative by design: each distillation step refines the current policy, which in turn provides a fresh displacement for the next extrapolation. For this recursion to converge, the extrapolation direction must point toward genuine improvement—yet extrapolation itself is direction-agnostic, amplifying whatever update the model made regardless of quality. We resolve this by using RLVR to produce the update $\theta \to \theta'$: outcome rewards ground the displacement in verified improvement. On-policy distillation then applies the extrapolated teacher to refine θ', providing the per-token supervision that outcome rewards alone cannot. The two signals are complementary: RLVR ensures the direction is meaningful; OPD ensures the refinement is fine-grained (Fig. 1).

Our contributions are threefold. **(1)** We identify teacher quality as the central bottleneck of OPD and propose **RISE**, which constructs the teacher by extrapolating the model's own RLVR trajectory, requiring no external model and no privileged context. Because the teacher is refreshed every iteration from the student's latest update, distillation becomes a recursive improvement loop rather than a one-shot compression step (§3). **(2)** We unify this construction under a representation map φ and instantiate it in two spaces—*logit-space* (geometric mixture of output distributions) and *weight-space* (task arithmetic)—characterizing their computational and statistical trade-offs and ablating both (§3, §4). **(3)** We show that RISE outperforms RLVR and OPSD baselines across mathematical reasoning, multi-domain STEM, code generation, and multi-turn agentic tasks, with improved sample efficiency, at no additional sampling cost and a modest 1.3–1.6× wall-time overhead (§4).

2 Background and Related Work

RLVR and OPD. RLVR optimizes a policy π_θ against an outcome reward $R(x, y) \in [0, 1]$ on tasks with checkable answers, using group-normalized policy gradients such as GRPO (Guo et al., 2025) and DAPO (Yu et al., 2026). Because the reward is assigned at the sequence level, every token of a response receives the same advantage—a credit assignment bottleneck that cannot distinguish helpful reasoning steps from harmful ones. On-policy distillation (OPD) supplies the missing per-token signal by minimizing a divergence between the student's distribution and a teacher's at every

position of a student-generated rollout, typically over a top-K truncation of the vocabulary (Lu & Lab, 2025; Agarwal et al., 2024), but its effectiveness is bounded by the quality of the teacher π_T.

The teacher quality problem. Existing choices of π_T fall into three categories, each with a fundamental limitation. *External teachers* (Xiaomi, 2025; Zeng et al., 2026) suffer from distribution mismatch on the student's own prefixes (Li et al., 2026c; Zhu et al., 2026). *On-policy self-distillation* (OPSD) with privileged conditioning (Zhao et al., 2026b; Hübotter et al., 2026) presumes ICL capability the model may lack (Li et al., 2026b; Kim et al., 2026; Harne et al., 2026). *Remedies*—token gating (Xu et al., 2026b; Lu et al., 2026; Xing et al., 2026; Ko et al., 2026; Liu et al., 2026), KL mixing (Jung et al., 2025; Jin et al., 2026; Jia et al., 2026; Jang et al., 2026), DAgger-style sampling (Ross et al., 2011; Li et al., 2026a; Agrawal et al., 2026; Zhao et al., 2026a), trajectory refinement (Yang et al., 2026b; Jiang et al., 2026)—accommodate a flawed teacher rather than replacing it, as do methods combining RLVR with OPD (Xu et al., 2025; Ramos et al., 2026; Zhang et al., 2026; Yang et al., 2026a; Xu et al., 2026a; Wang et al., 2026b; Oh et al., 2026). Closest to RISE is ExOPD (Yang et al., 2026d), whose optimal policy is the same geometric mixture as our logit-space teacher (Eq. 4). The key distinction is teacher *dynamics*: ExOPD fixes a static external policy, so the teacher gap $J(\pi^*) - J(\pi_T)$ in Theorem 2 is a constant ceiling; RISE's teacher advances in lockstep with the student, turning OPD into a recursive improvement loop. A direct comparison is inapplicable because ExOPD requires an external stronger model—precisely what RISE eliminates.

Linear trajectories. Fine-tuning trajectories are approximately linear and low-dimensional: task vectors compose linearly (Ilharco et al., 2022; Frankle et al., 2020), RLVR updates concentrate in a rank-1 subspace with near-linear dynamics (Cai et al., 2025; Wei et al., 2026; Wang et al., 2026a), and OPD updates lock into a narrow subspace (Shen et al., 2026). Model soups exploit this structure by *interpolating* checkpoints; RISE instead *extrapolates* beyond the current checkpoint. However, RISE does not require strict linearity: confinement to a low-dimensional subspace limits how far extrapolation can deviate from the true trajectory, and our own runs confirm this—three directions capture $\sim$87% of the variance (Appendix B.3). See expanded related work in Appendix A.

3 METHOD

Theorem 2 (Appendix B) formalizes §1's intuition: the optimal teacher for π_{θ_n} is π^*, but since π^* is unknown, RISE approximates it by extrapolating the model's own RLVR-grounded training trajectory, then distills that teacher's per-token distribution into the student.

3.1 SELF-EXTRAPOLATED POLICY DISTILLATION

Consider a policy π_{θ_n} updated by RLVR to $\pi_{\theta'_{n+1}}$. Let φ be a representation map into a vector space where linear operations are meaningful. The self-extrapolated teacher is defined as:

$$\varphi(\pi_{\text{future}}) = \varphi(\pi_{\theta_n}) + \beta \cdot \big(\varphi(\pi_{\theta'_{n+1}}) - \varphi(\pi_{\theta_n})\big), \quad \beta > 1. \tag{2}$$

The extrapolation scale β controls how far beyond the current policy we project: $\beta = 1$ recovers $\pi_{\theta'_{n+1}}$, and $\beta > 1$ continues along the improvement direction. The choice of φ yields a family of instantiations with different computational and statistical properties.

Weight-space extrapolation ($\varphi = \theta$). Setting φ to the identity on parameters gives

$$\theta_{\text{future}} = \theta_n + \beta \cdot (\theta'_{n+1} - \theta_n), \tag{3}$$

which is precisely task arithmetic (Ilharco et al., 2022) with an extrapolation coefficient. The teacher is then the model defined by θ_{future}.

Logit-space extrapolation ($\varphi = \log \pi$). Let $s_t \triangleq (x, y_{<t})$ denote the context at position t. Setting φ to map each policy to its per-token output log-probabilities gives:

$$\log \pi_{\text{future}}(\cdot \mid s_t) = \log \pi_{\theta_n}(\cdot \mid s_t) + \beta \cdot \big(\log \pi_{\theta'_{n+1}}(\cdot \mid s_t) - \log \pi_{\theta_n}(\cdot \mid s_t)\big) + \text{const}, \tag{4}$$

where the constant ensures normalization. Equivalently, $\pi_{\text{future}} \propto \pi_{\theta_n}^{1-\beta} \cdot \pi_{\theta'_{n+1}}^{\beta}$ —a *geometric mixture* that amplifies the probability ratio $\pi_{\theta'_{n+1}}/\pi_{\theta_n}$.

Remark 1 (Connection to KL regularization)**.** *Given the geometric mixture, the distillation loss decomposes as*

$$D_{\text{KL}}(\pi_\theta \| \pi_{\textit{future}}) = -(\beta - 1)\, D_{\text{KL}}(\pi_\theta \| \pi_{\theta_n}) + \beta\, D_{\text{KL}}(\pi_\theta \| \pi_{\theta'_{n+1}}) + \log Z, \tag{5}$$

where Z is the normalization constant of π_{future}, and both π_{θ_n} and $\pi_{\theta'_{n+1}}$ are treated as fixed (stop-gradient) when constructing π_{future}. For $\beta > 1$, the first term has a negative coefficient and is repulsive—it pushes the current policy away from the anchor, continuing the RLVR improvement direction; the second term regularizes it toward the post-RLVR checkpoint $\pi_{\theta'_{n+1}}$, preventing overshoot during OPD. Both terms operate at the token level*, providing fine-grained credit assignment that a scalar outcome reward cannot.*

Relationships between instantiations. Let $f(\theta)$ denote the mapping from parameters to output logits. Weight-space extrapolation computes $f(\theta_n + \beta \cdot \Delta\theta)$, while logit-space computes $f(\theta_n) + \beta \cdot \Delta f$—the *first-order Taylor approximation* of weight-space extrapolation around θ_n. The two coincide exactly when f is linear; for neural networks they diverge. Weight-space produces a coherent model whose cross-position predictions are jointly consistent, at the cost of materializing parameters and one teacher forward pass per OPD step; logit-space needs no parameter manipulation and fixes the teacher before OPD begins.

3.2 Distillation Loss Design

Computing a divergence over the full vocabulary is prohibitively expensive. Following prior works (Hübotter et al., 2026; Zhao et al., 2026b), we use a top-K approximation. For logit-space extrapolation, the unbiased procedure would extrapolate over the full vocabulary and then select the top-K of π_{future}; however, π_{future} does not exist as a model—it is defined only through the logit-space formula—so obtaining its full-vocabulary logits would require keeping two extra model copies in memory or caching $O(V)$ logits per position. Instead, letting $S = \text{Top}_K(\pi_{\theta'_{n+1}})$, we project both $\pi_{\theta'_{n+1}}$ and π_{θ_n} onto the same $(K+1)$-simplex (retaining log-probabilities at S and appending a tail bucket for the remaining mass, each renormalized), then apply Eq. 4 to all $K+1$ log-probabilities:

$$\log \pi_{\text{future}}(v \mid \cdot) = \log \pi_{\theta_n}(v \mid \cdot) + \beta \cdot \big(\log \pi_{\theta'_{n+1}}(v \mid \cdot) - \log \pi_{\theta_n}(v \mid \cdot)\big), \quad v \in S \cup \{\text{tail}\}. \tag{6}$$

With $K = 100$, the top-K tokens account for essentially all probability mass under typical LLM distributions, so any bias from using S instead of $\text{Top}_K(\pi_{\text{future}})$ is negligible (Appendix B.4). For weight-space extrapolation, a forward pass through θ_{future} (Eq. 3) yields π_{future} exactly, and we evaluate it at the student's top-K indices with a tail bucket appended and renormalized.

In both cases, π_{future} is held fixed (stop-gradient) throughout all OPD gradient steps. The RISE distillation loss is:

$$\mathcal{L}_{\text{RISE}} = \mathbb{E}_{y \sim \pi_\theta}\left[\sum_{t=1}^{T} D_{\text{KL}}(\pi_\theta(\cdot \mid s_t) \parallel \text{sg}[\pi_{\text{future}}(\cdot \mid s_t)])\right], \tag{7}$$

where the KL divergence is computed over the $(K+1)$-dimensional simplex.

Choice of divergence. Our analysis (Remark 1, Theorem 2) uses reverse KL for an exact trust-region decomposition. In practice we replace KL with the Jensen–Shannon divergence (JSD) in Eq. 7, which is bounded by $\log 2$ thus avoiding the numerical instability of reverse KL. Since $\text{JSD}(P\|Q) \leq \frac{1}{2} D_{\text{KL}}(P\|Q)$, the trust-region guarantees from the KL analysis carry over as conservative bounds. The two losses also share the same unique minimizer ($\pi_\theta = \pi_{\text{future}}$), so the teacher targeted is identical.

3.3 Training Procedure

Each RISE iteration alternates between two phases:

1. **RLVR phase.** Sample rollouts from the current policy π_{θ_n}, compute outcome rewards, and update the policy via a policy gradient method (e.g., GRPO) to obtain θ'_{n+1}.
2. **OPD phase.** Construct π_{future} from $\pi_{\theta'_{n+1}}$ and anchor π_{θ_n} via logit-space (Eq. 4) or weight-space (Eq. 3) extrapolation. Distill π_{future} into $\pi_{\theta'_{n+1}}$ by minimizing $\mathcal{L}_{\text{RISE}}$ (Eq. 7), yielding θ_{n+1}.

The two phases are complementary: RLVR discovers capability improvements via sparse outcome rewards, while OPD compresses the extrapolated teacher's token-level distribution into the current policy (Algorithms 1 and 2). Both phases operate on the *same* set of rollouts—the responses $y \sim \pi_{\theta_n}$ sampled for RLVR are reused for OPD, which therefore adds no additional sampling cost. Reusing pre-RLVR rollouts introduces mild off-policy-ness (contexts come from pre-RLVR), but the

distributional loss is well-defined for any input sequence without importance sampling correction. We confirm empirically that this shift has negligible impact (§4.3).

Extrapolation Decay. A fixed β is unsuitable across the full training trajectory. Under the linear-trajectory model (Proposition 3, Appendix B), the safe β range narrows as the policy approaches the optimum: beyond a β-threshold the extrapolated teacher overshoots π^* and the suboptimality bound worsens. We therefore apply a monotonically decreasing schedule $\beta_n \to 1$ (e.g., linear: $\beta_n = 1 + (\beta_0 - 1)(1 - n/N)$). Large β_n early in training extrapolates aggressively when the policy is far from optimal; as training progresses, decreasing β_n keeps the teacher within the valid range.

Anchor Dynamics. The anchor θ_{anchor}—the reference checkpoint used to compute the displacement vector—is by default the previous checkpoint θ_n. Alternatively, an EMA anchor $\theta_{\text{anchor}} \leftarrow (1-\eta)\theta_{\text{anchor}} + \eta\,\theta_{n+1}$ both smooths and enlarges the displacement: the anchor lags behind the current policy, averaging the direction over multiple iterations while increasing $\|\theta'_{n+1} - \theta_{\text{anchor}}\|$. The previous-checkpoint anchor is the special case $\eta = 1$. We ablate this choice in §4.3.

Role of OPD. Why distill from π_{future} rather than adopt it directly as the next policy? The extrapolated point lies beyond the policy's trust region: adopting it wholesale risks degenerate behavior at large β, and amplifies any noise in the RLVR step. OPD instead acts as a *trust-region projection*—it moves the policy toward π_{future}'s token-level distribution while the divergence term in $\mathcal{L}_{\text{RISE}}$ keeps the update anchored near $\pi_{\theta'_{n+1}}$; we validate this choice and the need for RLVR grounding in §4.2. Importantly, RISE does not introduce new task information beyond what RLVR provides—no external data or model enters the system. Rather, it exploits the model's own inductive structure to convert a sparse outcome-induced parameter update into a dense token-level target, redistributing the same training signal into a form that enables finer-grained credit assignment.

4 Experiments

We evaluate RISE across multiple model scales, architectures, and domains to answer five questions: (1) Does RISE improve over RLVR-only training and existing OPSD methods? (2) Does the improvement hold across model scales and families? (3) Does RISE preserve performance on out-of-distribution tasks while improving in-domain accuracy? (4) What mechanisms drive the improvement? (5) Which design choices matter most?

Models and datasets. We evaluate RISE on four task families. *Mathematical reasoning*: Qwen3-8B, Qwen3-1.7B, and Qwen3-1.7B-Base (Team, 2025) on DAPOMath (Yu et al., 2026), and OLMo3-7B-Instruct-SFT (Olmo et al., 2025) on OpenR1-Math-46K (Yan et al., 2026), with GPQA-Diamond, IFEval, and MMLU-Pro as out-of-distribution checks. *Multi-domain (math + STEM)*: Qwen3-4B-Base on a mixed corpus following Guru (Cheng et al., 2026), using only their STEM data and replacing their math split with DAPOMath (the original problems are low complexity). *Code generation*: Qwen3-8B-Base on Skywork-OR1-Code (He et al., 2025). *Agentic tasks*: Qwen2.5-3B-Instruct (Team, 2024) on ALFWorld (Shridhar et al., 2020) and WebShop (Yao et al., 2022) following GIGPO (Feng et al., 2025). Evaluation benchmarks, sample counts, and full training details are in Appendix D.

Baselines. We report the base/SFT model to establish the starting point and GRPO (Guo et al., 2025) as the RLVR-only baseline. To isolate the contribution of RISE's teacher construction, we compare against three methods that also pair GRPO's sequence-level signal with per-token self-distillation from the *same* privileged teacher—the student conditioned on a sibling correct solution—but differ in how they integrate it: GRPO+SDPO (Hübotter et al., 2026) adds an auxiliary KL loss, SDAR (Lu et al., 2026) gates the GRPO advantage by the teacher–student probability gap, and RLSD (Yang et al., 2026a) reweights advantage estimates for finer-grained credit assignment. We exclude external-teacher OPD, which requires a separate stronger model, whereas RISE and all OPSD baselines use only the model itself (Appendix D.3).

RISE configurations. We report both logit-space and weight-space RISE. Default hyperparameters: $\beta_0 = 1.2$, linear decay to $\beta_N = 1$, $K = 100$ ($K = 20$ for code, where the output distribution is more peaked); for the anchor we use $\eta = 0.1$ (EMA) for Qwen models and $\eta = 1$ (previous checkpoint) for OLMo (ablated in §4.3); the distillation loss uses Jensen–Shannon divergence (cf. §3.2). All other training hyperparameters (learning rate, batch size, rollout length) match the GRPO baseline for fair comparison. All methods are trained for one epoch on the respective training set.

Table 1: **Mathematical reasoning results.** Accuracy (%) across model configurations. We evaluate on in-domain math benchmarks and out-of-distribution (OOD) tasks to assess preservation. Best in **bold**, second best underlined.

Method	In-Domain Math							OOD			
	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.	GPQA	IFEval	MMLU	Avg.
Qwen3-8B (DAPOMath)											
Base	73.2	27.1	23.1	66.1	21.1	48.6	43.2	50.8	82.4	68.4	67.2
GRPO	83.8	54.4	42.9	89.4	31.5	57.9	60.0	57.0	81.5	73.3	70.6
GRPO+SDPO	83.2	42.3	32.7	84.4	30.6	62.5	55.9	55.7	82.8	69.4	69.3
SDAR	84.5	50.2	38.3	89.4	31.2	**64.7**	59.7	55.5	82.4	70.1	69.3
RLSD	83.0	53.1	40.0	90.6	31.3	60.5	59.8	56.2	82.6	72.6	70.5
RISE (logit)	**84.8**	56.9	**46.7**	91.6	32.5	62.4	62.5	58.2	81.3	**74.3**	71.3
RISE (weight)	84.4	**58.1**	45.8	**91.9**	**32.6**	63.3	**62.7**	**59.0**	**83.4**	73.5	**72.0**
Qwen3-1.7B (DAPOMath)											
Base	64.6	11.5	12.1	41.9	17.6	40.3	31.3	34.9	67.7	48.3	50.3
GRPO	75.6	30.0	26.3	65.6	23.9	51.1	45.4	32.3	68.8	53.4	51.5
GRPO+SDPO	76.0	21.5	23.1	65.0	23.1	50.3	43.2	34.9	69.0	52.0	52.0
SDAR	75.0	26.0	26.0	71.3	23.6	50.8	45.5	**35.3**	69.1	53.2	52.5
RLSD	75.3	26.7	25.2	59.4	23.0	49.8	43.2	33.4	67.5	54.1	51.7
RISE (logit)	75.2	**36.3**	**33.3**	**78.8**	24.9	52.9	**50.2**	34.8	68.2	53.7	52.2
RISE (weight)	**77.3**	32.9	31.3	75.0	**25.7**	**53.3**	49.2	34.8	**70.2**	**56.3**	**53.8**
OLMo3-7B-Instruct-SFT (OpenR1)											
Base	57.0	6.0	8.1	43.8	19.6	31.5	27.7	33.0	77.8	35.4	48.7
GRPO	77.5	30.2	28.3	70.6	25.2	53.5	47.6	31.0	76.0	46.6	51.2
GRPO+SDPO	77.7	33.8	28.5	70.6	25.3	53.4	48.2	37.4	77.1	46.1	53.5
SDAR	79.8	35.2	27.1	73.8	25.3	56.3	49.6	39.0	77.8	44.6	53.8
RLSD	77.2	28.3	24.0	68.1	24.6	52.9	45.9	37.6	78.4	44.1	53.4
RISE (logit)	**82.9**	**46.9**	**36.0**	**83.1**	27.6	**61.9**	**56.4**	**40.7**	**79.3**	46.4	**55.5**
RISE (weight)	82.8	42.5	32.5	76.9	**27.8**	59.6	53.7	37.1	77.1	**46.9**	53.7

4.1 Main Results

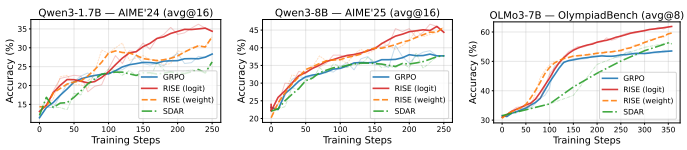
Mathematical Reasoning. Table 1 presents results across three model configurations spanning different scales (1.7B–8B), architectures (Qwen, OLMo), and training corpora (DAPOMath, OpenR1).

RISE consistently outperforms all baselines. Both RISE variants beat every baseline on in-domain Math Avg in all three settings, and at least one ranks first or second on every individual in-domain benchmark. The gains are most pronounced on competition benchmarks: on OLMo3-7B, RISE (logit) improves AIME'24 from 30.2 to 46.9 (+16.7) and Math Avg from 47.6 to 56.4 (+8.8); on Qwen3-1.7B, RISE (logit) lifts Math Avg from 45.4 to 50.2 (+4.8); and on Qwen3-8B, RISE (weight) lifts it from 60.0 to 62.7 (+2.7). Neither extrapolation space consistently dominates, suggesting the gains stem from the extrapolation principle itself. Multi-seed experiments confirm reproducibility: RISE (weight) Math Avg = 62.4 ± 0.2 vs. GRPO 60.1 ± 0.2 on Qwen3-8B, and RISE (logit) 49.6 ± 0.4 vs. GRPO 45.1 ± 0.4 on Qwen3-1.7B, across three seeds (Appendix E.10).

Privileged-conditioning baselines provide limited or negative gains. GRPO+SDPO, which jointly optimizes the GRPO loss and a KL divergence toward the privileged teacher, underperforms GRPO on both Qwen3-8B (55.9 vs. 60.0) and Qwen3-1.7B (43.2 vs. 45.4). SDAR and RLSD fare better but remain within about two points of GRPO in most settings. This pattern is consistent with our analysis in §2: privileged-conditioning teachers are limited by ICL capability, capping the benefit of per-token supervision regardless of integration strategy.

Out-of-distribution preservation. RISE maintains or slightly improves OOD performance across all models—Qwen3-8B OOD Avg rises from 70.6 (GRPO) to 72.0 (RISE weight), OLMo3-7B from 51.2 to 55.5 (RISE logit)—so token-level refinement does not degrade general capabilities.

Sample efficiency. RISE also learns faster: Figure 2 plots evaluation accuracy at successive checkpoints across three configurations, and RISE reaches higher accuracy earlier throughout, with the gap widest in early training where the extrapolated teacher supplies dense signal before RL advantage estimates stabilize (further curves in Appendix E.2).



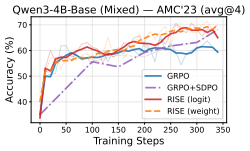
(a) 1.7B AIME'24 (b) 8B AIME'25 (c) OLMo OlyBench

Figure 2: RISE reaches higher accuracy in fewer steps across three model scales, demonstrating improved sample efficiency.

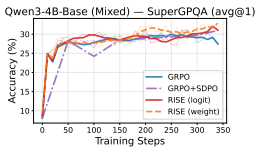
Table 2: **Multi-domain results (Qwen3-4B-Base, mixed math + STEM training).** Accuracy (%) on math reasoning and STEM benchmarks. Best in **bold**, second best underlined.

Method	Math							STEM				
	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.	GPQA	SuperGPQA	MMLU	ThrmQA	Avg.
Base	48.2	9.8	10.4	35.0	12.1	27.5	23.8	14.1	8.0	6.6	27.0	13.9
GRPO	69.1	23.8	20.6	59.4	26.6	42.0	40.2	42.4	32.5	59.5	47.8	45.5
GRPO+SDPO	70.5	24.8	20.4	68.1	25.9	43.1	42.1	44.7	31.0	61.7	50.5	47.0
SDAR	69.1	22.9	20.0	65.6	23.7	41.2	40.4	40.6	30.3	58.7	49.1	44.7
RLSD	68.9	23.1	20.0	60.0	24.6	39.9	39.4	33.7	28.8	55.1	46.5	41.0
RISE (logit)	**71.9**	25.6	22.1	70.6	25.9	43.8	43.3	44.1	32.1	**61.9**	51.3	47.4
RISE (weight)	71.3	**28.8**	**25.4**	**71.3**	**26.9**	**44.9**	**44.8**	**44.7**	**32.9**	60.9	**51.5**	**47.5**

Multi-Domain Results. Table 2 tests whether RISE holds when the displacement aggregates improvements across heterogeneous domains. On Qwen3-4B-Base trained with mixed math and STEM data, RISE (weight) achieves the highest Math Avg (44.8 vs. GRPO's 40.2) and STEM Avg (47.5 vs. 45.5), improving competition math (AIME'24: 28.8, +5.0) while simultaneously lifting STEM (TheoremQA: 51.5, +3.7)—extrapolation along a multi-domain trajectory does not dilute gains in any single domain. GRPO+SDPO is more competitive here (STEM Avg 47.0) than in the math-only setting, consistent with STEM providing more informative privileged context (Hübotter et al., 2026), yet RISE still outperforms it without privileged information.

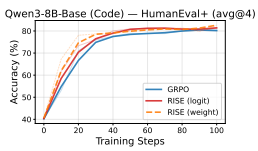


(a) AMC'23

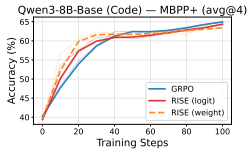


(b) SuperGPQA

Figure 3: RISE reaches higher accuracy on both math and STEM benchmarks.



(a) HumanEval+



(b) MBPP+

Figure 4: RISE converges faster than GRPO on code generation.

The sample-efficiency advantage extends to this setting. On AMC'23 (Fig. 3a) both RISE variants stay above GRPO throughout training, with RISE (weight) reaching 70.0% while GRPO rises mid-training but declines to 59.4%; on SuperGPQA (Fig. 3b) RISE (weight) holds a consistent edge, so the signal benefits both math and STEM.

Code Generation. We evaluate Qwen3-8B-Base on code generation (Fig. 4). Both RISE variants converge faster than GRPO early and reach comparable final accuracy—RISE (logit) matches GRPO's final HumanEval+ accuracy at step 50 versus step 90—so the extrapolation principle transfers beyond mathematical reasoning. Gains are more modest than in math, likely because all methods saturate quickly on these benchmarks. Full curves (avg@4 and pass@4) are in Appendix E.3.

Agentic Tasks. Table 3 reports results on two multi-turn agentic benchmarks using Qwen2.5-3B-Instruct as the base model, following the GIGPO training setup (Feng et al., 2025). RISE (weight) substantially outperforms GRPO on both ALFWorld (+9.4) and WebShop (+10.9 Acc), demonstrating that the extrapolation principle extends to sequential decision-making tasks where the reward signal is sparse and delayed.

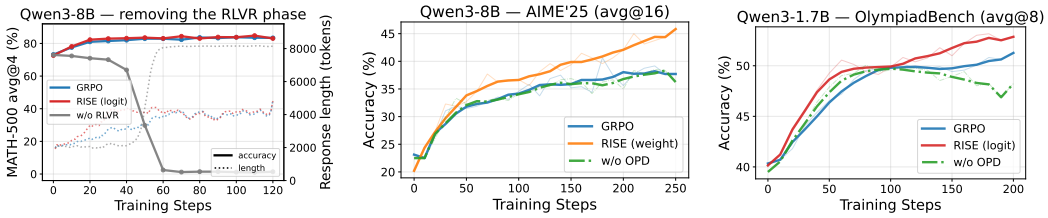
Table 3: Success rate on ALFWorld, Score/Acc on WebShop.

Method	ALF	WS-S	WS-A
Base	21.9	6.7	0.8
GRPO	75.0	79.8	63.3
RISE (logit)	78.1	78.8	68.0
RISE (weight)	**84.4**	**86.3**	**74.2**

4.2 Analysis: How Does Extrapolation Help?

Beyond aggregate accuracy, we ask *why* extrapolation helps. RISE couples two ingredients—an RLVR phase that supplies the displacement direction, and an OPD phase that projects the extrapolated teacher back onto the policy. We first isolate each phase by removing it (Fig. 5), then test the extrapolation premise directly on saved checkpoints (Fig. 6).

Extrapolation is only safe when the direction is grounded in verifiable reward. We remove the RLVR phase entirely and extrapolate along the displacement induced by self-distillation alone. Training collapses within 60 steps (Fig. 5a): MATH-500 accuracy drops from the base level to 2.4%, response length explodes from 2K to the 8K context cap, and training reward drops to zero (Fig. E.3). GRPO and RISE show no such behaviour. Without an outcome reward to anchor it, the displacement



(a) Removing RLVR (8B) (b) Removing OPD (8B) (c) Removing OPD (1.7B)

Figure 5: **Removing either phase hurts.** (a) Without RLVR, accuracy collapses (solid) as generation length explodes (dotted). (b, c) Without OPD, training is stable but gains vanish.

is purely self-referential—each iteration extrapolates along the model's prior move—and repeated amplification drives the policy into degenerate non-terminating generation.

The gain comes from distillation, not from taking a longer step. We next remove the OPD phase, adopting the extrapolated θ_{future} directly as the next policy. This variant is stable but does not improve the *average*: on Qwen3-8B Math Avg moves from 60.0 (GRPO) to 60.3 (w/o OPD), and on Qwen3-1.7B from 45.4 to 45.6—negligible gains compared to RISE's $+2.7$ and $+4.8$ (Table E.3). Individual benchmarks shift in opposite directions at different scales (Figs. 5b, 5c), but these redistributions do not translate into consistent improvement. Taking a longer step in weight space *shifts* the policy without reliably improving it; it is the OPD phase—projecting the extrapolated teacher's per-token distribution back onto the policy—that converts the extrapolation direction into consistent gains across benchmarks and scales.

The safe extrapolation range narrows over training. We test the extrapolation premise directly by materializing $\theta_{\text{base}} + \beta(\theta_n - \theta_{\text{base}})$ from GRPO checkpoints of Qwen3-1.7B and evaluating on AIME'24 (Fig. 6). Two patterns emerge. First, within RISE's operating range ($\beta \leq 1.2$, shaded), extrapolation is safe throughout: gains are near-neutral early and reach $+2$–4 points at steps 100–200, while never degrading accuracy. Second, the *tolerable range* of β contracts sharply as training proceeds: at step 50, even $\beta = 2.0$ adds $+7.3$ points, but by step 100 the same β is catastrophic (-15 points), and by step 200 $\beta = 1.5$ already costs 7.7 points. This is the empirical counterpart of Proposition 3 and the direct justification for the decaying schedule: a fixed aggressive β would eventually destroy the teacher, whereas RISE's conservative, decaying β stays inside the safe region throughout.

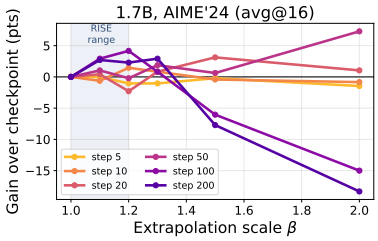


Figure 6: Gain on AIME'24 (avg@16) from extrapolating Qwen3-1.7B GRPO checkpoints at varying β, relative to $\beta = 1$. Shaded: RISE's operating range ($\beta_0 = 1.2$, decaying to 1).

RISE broadens solution coverage, not just average quality. A natural concern with distillation is that it sharpens the policy around solutions it already finds, improving mean accuracy at the expense of coverage. We test this by comparing avg@16 with pass@16 (best-of-16) on the AIME benchmarks (Appendix E.4). RISE improves pass@16 over GRPO in nearly every setting, with the effect strongest at 1.7B: RISE (logit) lifts AIME'24 pass@16 by $+9.6$ points versus $+6.3$ on avg@16, indicating that the extrapolated teacher expands the set of solvable problems rather than merely sharpening existing solutions. At 8B the coverage gains are more modest, consistent with less headroom when the base policy is already stronger.

4.3 Ablation Studies

We conduct ablations primarily on Qwen3-8B and Qwen3-1.7B(-Base) to isolate the contribution of each design choice.

Extrapolation scale β and decay schedule. Table 4 reports sensitivity to β_0 and the decay schedule on Qwen3-8B (logit-space extrapolation). Performance is stable across $\beta_0 \in [1.2, 1.5]$, with $\beta_0 = 1.2$ slightly ahead; $\beta_0 = 2.0$ diverged in preliminary experiments, consistent with the narrowing safe range shown

Table 4: β_0 **and schedule ablation** (Qwen3-8B, Math Avg %).

	β_0 (linear decay)			Schedule ($\beta_0 = 1.2$)			
	1.2	1.3	1.5	cosine	linear	exp	fixed
Math Avg	62.5	62.3	61.9	**63.0**	62.5	62.0	61.8

in Fig. 6: although $\beta = 2.0$ is benign early in training, it becomes catastrophic once the policy nears optimality, and linear decay from $\beta_0 = 2.0$ does not reduce β fast enough to avoid this regime. All decay schedules perform comparably, though any form of decay edges out a fixed β—consistent with the prediction that β should shrink as the policy approaches optimality. On Qwen3-1.7B-Base the pattern is similar: $\beta_0 = 1.3$ (linear) gives the best average (29.4), while $\beta_0 = 1.5$ degrades to 28.5 (per-benchmark breakdowns in Appendix E.6).

Anchor dynamics (η). The EMA anchor ($\eta < 1$) smooths the displacement direction over multiple iterations, which helps when per-step RLVR updates are noisy. Fig. 7 ablates η on Qwen3-1.7B. In logit space, reducing η monotonically improves Math Avg: from 46.2 ($\eta = 1.0$) to 48.4 ($\eta = 0.3$) to 50.2 ($\eta = 0.1$); in weight space the trend is consistent ($+2.8$ from $\eta = 1.0$ to $\eta = 0.1$). However, on OLMo3-7B $\eta = 1.0$ outperforms $\eta = 0.1$ (56.4 vs. 50.1), likely because GRPO on OLMo already produces large, stable per-step directions, making EMA smoothing unnecessary and its lag counterproductive. We therefore default to $\eta = 0.1$ for Qwen and $\eta = 1$ for OLMo (Appendix E.8).

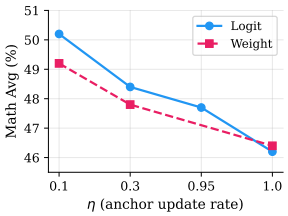


Figure 7: Anchor ablation on Qwen3-1.7B. Lower η improves Math Avg.

Resample after RLVR vs. rollout reuse. Generating fresh rollouts from $\pi_{\theta'_{n+1}}$ for the OPD phase eliminates the mild off-policy-ness discussed in §3.3, but does not improve results: on Qwen3-8B the resample variant matches rollout reuse exactly (62.5 Math Avg), and on Qwen3-1.7B-Base it slightly underperforms (28.4 vs. 29.2). Rollout reuse is therefore the practical default, halving the per-iteration sampling cost (per-benchmark details in Appendix E.7).

Compute-matched comparison. Since RISE applies two gradient phases per iteration (RLVR + OPD), we compare against GRPO that performs a second inner-loop gradient pass on the same rollouts within each iteration (GRPO-2×), matching the total gradient budget. The extra pass helps only marginally—$+1.9$ Math Avg on Qwen3-1.7B (47.3 vs. 45.4) and $+0.5$ on Qwen3-8B (60.5 vs. 60.0)—while RISE gains $+4.8$ and $+2.7$ over GRPO at the same budget, outperforming GRPO-2× by $+2.9$ and $+2.2$ respectively (per-benchmark details in Appendix E.9). The gains therefore stem from the quality of extrapolated supervision rather than additional optimization steps.

Training cost. RISE reuses the RLVR rollouts for OPD, so the overhead comes from the OPD gradient phase and the anchor forward pass. On Qwen3-8B, RISE runs at $\sim$1.6× GRPO wall time per iteration; on Qwen3-1.7B, $\sim$1.3×. The smaller relative overhead at 1.7B reflects that rollout generation—shared between GRPO and RISE—dominates at smaller scale (73% of GRPO wall time vs. 63% at 8B).

5 Conclusion

We introduced RISE, which constructs a per-token teacher by extrapolating the model's own RLVR training trajectory—in logit space or weight space—then distills the resulting future policy back into the current student. The two phases are complementary: outcome rewards ground the extrapolation in verified improvement, while distillation converts the teacher's token-level distribution into durable policy gains. Across four task families (math, STEM, code, and agentic tasks) spanning 1.7B–8B parameters, RISE consistently outperforms RLVR and privileged-conditioning baselines, with the largest margins on challenging competition benchmarks. Analysis confirms that both phases are necessary, that the safe extrapolation range narrows over training, and that RISE broadens solution coverage alongside mean accuracy—all at 1.3–1.6× GRPO wall time with no additional sampling cost. More broadly, RISE demonstrates that a model's own training trajectory contains sufficient structure to serve as a self-improving teacher—converting sparse outcome signals into dense token-level supervision without any external knowledge. We believe this principle extends naturally to other post-training paradigms such as preference optimization and multi-agent settings.

Limitations. RISE relies on the training trajectory being sufficiently low-dimensional for linear extrapolation to remain meaningful at modest β. The β-decay schedule and EMA anchor provide empirical robustness, yet principled detection of when extrapolation becomes unreliable remains open. Additionally, RISE inherits any biases in the RLVR reward signal: if the reward is hackable, extrapolation amplifies the spurious direction. Integrating reward model uncertainty or multi-objective rewards to qualify the extrapolation is a promising avenue for future work.

AI USE STATEMENT

In this work, we used generative AI tools for coding assistance, manuscript proofreading, and revising text for clarity and fluency. We have not used generative AI tools for generating research ideas, designing experiments, or producing experimental results. All AI-generated code was verified and tested for correctness by the authors; all AI-suggested text edits were reviewed and accepted or rejected by the authors. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All training hyperparameters are reported in Table D.1 (Appendix D), including learning rates, batch sizes, rollout lengths, and RISE-specific parameters (β_0, η, K). The evaluation protocol—benchmarks, sample counts, and the bootstrap pass@N estimator—is detailed in Appendix D.2. Complete pseudocode for both RISE variants is provided in Algorithms 1 and 2. All theoretical results include full proofs in Appendix B. Multi-seed variance is reported in Appendix E.10. We will release the training code upon acceptance.

REFERENCES

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

Rishabh Agrawal, Jacob Fein-Ashley, and Paria Rashidinejad. Reinforcement learning from rich feedback with distributional dagger. *arXiv preprint arXiv:2606.05152*, 2026.

Yuchen Cai, Ding Cao, Xin Xu, Zijun Yao, Yuqing Huang, Zhenyu Tan, Benyi Zhang, Guangzhong Sun, Guiquan Liu, and Junfeng Fang. On predictability of reinforcement learning dynamics for large language models. *arXiv preprint arXiv:2510.00553*, 2025.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, 2023.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

Jorge Zhoujun Cheng, Shibo Hao, Tianyang Liu, Fan Zhou, Yutao Xie, Feng Yao, Yuexin Bian, Nilabjo Dey, Yonghao Zhuang, Yuheng Zha, et al. Revisiting reinforcement learning for llm reasoning from a cross-domain perspective. *Advances in Neural Information Processing Systems*, 38, 2026.

Xeron Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *Advances in Neural Information Processing Systems*, 38, 2026.

Wenkai Fang, Shunyu Liu, Yang Zhou, Kongcheng Zhang, Tongya Zheng, Kaixuan Chen, Mingli Song, and Dacheng Tao. Serl: Self-play reinforcement learning for large language models with limited data. *Advances in Neural Information Processing Systems*, 38:103706–103738, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International conference on machine learning*, pp. 3259–3269. PMLR, 2020.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

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

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. *arXiv preprint arXiv:2508.05004*, 2025.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence*, pp. 876–885, 2018.

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. *arXiv preprint arXiv:2601.07155*, 2026.

Maxwell Jia. Aime problem set 2024, 2024. URL `https://huggingface.co/datasets/Maxwell-Jia/AIME_2024`.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. *arXiv preprint arXiv:2605.06387*, 2026.

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

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

Seongryong Jung, Suwan Yoon, DongGeon Kim, and Hwanhee Lee. Todi: Token-wise distillation via fine-grained divergence control. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8089–8102, 2025.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pp. 267–274, 2002.

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

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

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

Changhao Li, Rushi Qiang, Jiawei Huang, Chenxiao Gao, Chao Zhang, Niao He, and Bo Dai. Revisiting dagger in the era of llm-agents. *arXiv preprint arXiv:2605.12913*, 2026a.

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

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

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

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, et al. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1qvx610Cu7.

Kaiyuan Liu, Ziyuan Zhuang, Yang Bai, Bing Wang, Rongxiang Weng, and Jieping Ye. Prefix teach, suffix fade: Local teachability collapse in strong-to-weak on-policy distillation. *arXiv preprint arXiv:2605.13643*, 2026.

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

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

MAA. AMC 10/12 2023, 2023. URL https://www.maa.org/math-competitions/amc-1012.

math ai. Aime problem set 2025, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

Minjae Oh, Sangjun Song, Gyubin Choi, Yunho Choi, and Yohan Jo. Kl for a kl: On-policy distillation with control variate baseline. *arXiv preprint arXiv:2605.07865*, 2026.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe

Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

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

Miguel Moura Ramos, Duarte M Alves, and André FT Martins. Combining on-policy optimization and distillation for long-context reasoning in large language models. *arXiv preprint arXiv:2605.12227*, 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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.

Zhennan Shen, Yanshu Li, Qingyu Yin, Chak Tou Leong, Zhilin Wang, Yanxu Chen, Rongduo Han, Sunbowen Lee, and Yi R Fung. On the geometry of on-policy distillation. *arXiv preprint arXiv:2606.07082*, 2026.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

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

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Tianle Wang, Zhongyuan Wu, Shenghao Jin, Hao Xu, Wei Chen, and Ning Miao. Not all steps are informative: On the linearity of llms' rlvr training. *arXiv preprint arXiv:2601.04537*, 2026a.

Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. Openclaw-rl: Train any agent simply by talking. *arXiv preprint arXiv:2603.10165*, 2026b.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Zhepei Wei, Xinyu Zhu, Wei-Lin Chen, Chengsong Huang, Jiaxin Huang, and Yu Meng. You only need minimal rlvr training: Extrapolating llms via rank-1 trajectories. *arXiv preprint arXiv:2605.21468*, 2026.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carlin, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

LLM-Core Xiaomi. Mimo-v2-flash technical report, 2025. URL https://github.com/XiaomiMiMo/MiMo-V2-Flash/paper.pdf.

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

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

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.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Beyond grpo and on-policy distillation: An empirical sparse-to-dense reward principle for language-model post-training. *arXiv preprint arXiv:2605.12483*, 2026a.

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

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *Advances in Neural Information Processing Systems*, 38:117157–117186, 2026.

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

Han Yang, Mingyan Wu, Bailan He, Zeyu Cao, Sikuan Yan, Kevin Qinghong Lin, and Zifeng Ding. Reasoning compression with mixed-policy distillation. *arXiv preprint arXiv:2605.08776*, 2026b.

Haoyan Yang, Mario Xerri, Solha Park, Huajian Zhang, Yiyang Feng, Sai Akhil Kogilathota, and Jiawei Zhou. Self-improvement of large language models: A technical overview and future outlook. *arXiv preprint arXiv:2603.25681*, 2026c.

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*, 2026d.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

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

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for llm reasoning. *arXiv preprint arXiv:2602.22495*, 2026.

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

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

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026.

A Extended Background and Related Work

This appendix expands §2 with the full formulation of each component and a complete discussion of related work.

A.1 Reinforcement Learning with Verifiable Rewards

RLVR optimizes a language model policy π_θ to maximize expected reward on tasks with objectively checkable outputs. Given a prompt x, the model generates a response $y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x)$ and receives an outcome reward $R(x, y) \in [0, 1]$. Policy gradient algorithms such as GRPO (Guo et al., 2025) and DAPO (Yu et al., 2026) estimate per-token advantages from group-normalized outcome rewards and update the policy via clipped surrogate objectives:

$$\mathcal{L}_{\text{PG}} = -\mathbb{E}\left[\sum_{t=1}^{T} \hat{A}_t \cdot \min\Big(\rho_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)\Big)\right], \quad \rho_t = \frac{\pi_\theta(y_t|x, y_{<t})}{\pi_{\text{old}}(y_t|x, y_{<t})}, \tag{A.1}$$

where $\hat{A}_t$ is the advantage estimate (constant across tokens within a response in GRPO/DAPO). While effective, the outcome-level reward assigns identical advantage to every token in a response, creating a credit assignment bottleneck: the gradient signal cannot distinguish helpful reasoning steps from irrelevant or harmful ones.

A.2 On-policy Distillation

On-policy distillation (OPD) augments policy gradient training with per-token distributional supervision from a teacher model π_T. At each position t, the student minimizes a divergence between its distribution and the teacher's:

$$\mathcal{L}_{\text{OPD}} = \mathbb{E}_{y\sim\pi_\theta}\left[\sum_{t=1}^{T} D_{\text{KL}}(\pi_\theta(\cdot \mid x, y_{<t}) \,\|\, \pi_T(\cdot \mid x, y_{<t}))\right]. \tag{A.2}$$

Unlike the outcome reward, this loss provides a rich gradient at every token: the teacher's full distribution over the vocabulary specifies not only which token is best, but the relative desirability of all alternatives. In practice, computing the divergence over the full vocabulary is expensive, so most implementations resort to either a top-K truncation or a REINFORCE-style sample-based estimator (Lu & Lab, 2025; Agarwal et al., 2024). The critical challenge is the choice of teacher π_T. Existing approaches fall into two categories, each with fundamental limitations.

External teacher OPD. The most direct approach uses a separate, stronger model as π_T—for example, distilling from a larger model in the same family (Xiaomi, 2025; Zeng et al., 2026). The teacher evaluates $\pi_T(\cdot \mid x, y_{<t})$ conditioned on the student's generated prefix $y_{<t}$. As training progresses, the student explores increasingly diverse reasoning paths, many unseen by the teacher. The teacher's distribution conditioned on such out-of-distribution prefixes becomes unreliable: it reflects how the teacher would continue from an alien context, not whether the student's reasoning is sound. This *prefix distribution mismatch* degrades teacher quality precisely when the student needs guidance most—at novel or challenging reasoning steps (Li et al., 2026c; Zhu et al., 2026).

On-policy self-distillation (OPSD). An alternative formulation, on-policy self-distillation (OPSD), uses the *same* base model as both teacher and student, conditioning the teacher on privileged information such as correct solutions or environmental feedback (Zhao et al., 2026b; Hübotter et al., 2026). The teacher distribution becomes $\pi_T(\cdot \mid x, C, y_{<t})$, where C is the privileged context. The hope is that ICL enables the model to produce a more informed distribution when conditioned on C. However, this approach has its own limitations. First, the model's ICL capability may be insufficient to meaningfully incorporate the privileged context (Li et al., 2026b). Second, the teacher and student see different inputs, creating a different form of distribution mismatch: the teacher's distribution may reflect the privileged context in ways that are not transferable or even harmful to the student's unconditioned setting (Kim et al., 2026; Harne et al., 2026).

Heuristic Remedies. Recognizing these limitations, recent work has proposed several heuristic strategies. **Token-level gating** selectively weights the per-token loss using the teacher–student probability gap (Xu et al., 2026b; Lu et al., 2026; Xing et al., 2026), token entropy (Xu et al., 2026b; Ko et al., 2026), or teacher confidence (Liu et al., 2026). **Divergence mixing** interpolates between forward and reverse KL or selects adaptively per token, trading mode coverage against mode seeking (Hübotter et al., 2026; Jung et al., 2025; Jin et al., 2026; Jia et al., 2026; Jang et al., 2026).

DAgger-style rollout mixing (Ross et al., 2011) interleaves teacher and student tokens during generation to keep prefixes on-distribution (Li et al., 2026a; Agrawal et al., 2026; Zhao et al., 2026a). **Trajectory refinement** refines rollouts before distillation so the teacher conditions on higher-quality prefixes (Yang et al., 2026b; Jiang et al., 2026). All these remedies accept a flawed teacher and engineer ways to tolerate its noise, rather than questioning whether a better teacher exists. In §3, we show that the model's own training trajectory can be used to construct such a teacher.

A.3 Task Arithmetic and Linear Trajectories

RISE builds on convergent empirical evidence that post-training trajectories are low-dimensional and approximately linear. In the model merging literature, **task vectors** $\boldsymbol{\tau} = \theta_{\text{ft}} - \theta_{\text{pre}}$ encode task-specific knowledge in a composable, approximately linear fashion (Ilharco et al., 2022), and fine-tuning trajectories exhibit approximate **linear mode connectivity**—checkpoints along the training path can be linearly interpolated without loss barriers (Frankle et al., 2020). The related **model soups** and **weight averaging** literature (Wortsman et al., 2022; Izmailov et al., 2018) demonstrates that uniform or stochastic averaging of checkpoints along (or near) the training trajectory consistently improves generalization, reinforcing that these trajectories lie in well-behaved, low-curvature regions of parameter space amenable to linear operations.

More recently, analyses of RLVR training reveal that parameter updates are dominated by a **rank-1 subspace** whose projection coefficient evolves near-linearly throughout training (Cai et al., 2025; Wei et al., 2026), and that these low-rank weight dynamics propagate to linear evolution in output log-probabilities (Wang et al., 2026a). A parallel finding holds for OPD, whose cumulative updates rapidly lock into a narrow low-dimensional channel (Shen et al., 2026). Together, these properties justify extrapolation along the training direction: if $\theta_0 \to \theta_n$ is approximately linear and confined to a low-dimensional subspace, then $\theta_0 + \beta \cdot (\theta_n - \theta_0)$ with $\beta > 1$ is a principled estimate of a more capable future checkpoint. Whereas model soups exploit linearity through interpolation ($\beta \in [0, 1]$) for robustness, RISE exploits the same structure through extrapolation ($\beta > 1$) for capability improvement.

A.4 Joint RLVR and OPD Training

RLVR and OPD provide complementary signals, and recent work seeks to combine them. The OPD gradient is dense but bounded by teacher quality, while the RLVR gradient is sparse but can drive the student beyond the teacher's capability ceiling (Song & Zheng, 2026). The most direct approach augments the policy gradient objective with an auxiliary KL divergence term toward a teacher distribution (Xu et al., 2025; Ramos et al., 2026; Zhang et al., 2026). ExOPD (Yang et al., 2026d) and others (Wang et al., 2026b; Oh et al., 2026) formalize an equivalence between standard OPD and dense KL-constrained RL, where the log-probability ratio between teacher and student serves as a per-token reward. RLSD (Yang et al., 2026a) uses OPD to reweight the advantage from GRPO for finer-grained credit assignment. An alternative sequential strategy allocates RL to the teacher for capability discovery and OPD to the student for compression (Xu et al., 2026a). Whether combining with an external teacher or a privileged self-teacher, all these approaches take the teacher as given and focus on how to best integrate its signal with RL. RISE departs from this paradigm by eliminating the need for an external or privileged-conditioned teacher altogether: the teacher is constructed by extrapolating the model's own training trajectory.

Among these, ExOPD's reward extrapolation resembles RISE structurally: its optimal policy is the same geometric mixture as RISE's logit-space formulation (Eq. 4). The key distinction is not merely teacher source but teacher dynamics. ExOPD fixes a static external policy as teacher before distillation begins; no matter how well the student trains, the teacher gap $J(\pi^*) - J(\pi_T)$ in Theorem 2 remains constant—a hard ceiling. RISE's teacher is non-stationary by construction: as the student improves via RLVR, the displacement vector updates, and the extrapolated teacher advances in lockstep. This eliminates the static teacher ceiling and converts OPD from a one-shot compression step into a recursive improvement loop.

B Theoretical Analysis

B.1 Suboptimality Decomposition and Optimal Teacher

We first state the performance difference lemma adapted to the autoregressive setting (Kakade & Langford, 2002).

Lemma 1 (Performance difference)**.** *For two policies π_1, π_2 generating responses of length T with terminal reward $R(x, y) \in [0, 1]$:*

$$J(\pi_1) - J(\pi_2) = \sum_{t=1}^{T} \mathbb{E}_{s_t \sim \rho^t_{\pi_1}} \left[\sum_v \left(\pi_1(v \mid s_t) - \pi_2(v \mid s_t)\right) Q^{\pi_2}(s_t, v) \right], \tag{B.1}$$

where $s_t = (x, y_{<t})$ is the context at position t, $\rho^t_{\pi_1}$ is the distribution over position-t contexts induced by π_1, and $Q^{\pi_2}(s_t, v) = \mathbb{E}_{y_{t+1:T} \sim \pi_2}[R(x, y) \mid s_t, y_t = v] \in [0, 1]$ is the action-value function under π_2.

Proof. The performance difference lemma (Kakade & Langford, 2002), originally stated for the discounted infinite-horizon setting, specializes to the finite-horizon undiscounted case as

$$J(\pi_1) - J(\pi_2) = \sum_{t=1}^{T} \mathbb{E}_{s_t \sim P_t^{\pi_1}} [\mathbb{E}_{a_t \sim \pi_1}[A^{\pi_2}(s_t, a_t)]], \tag{B.2}$$

where $A^{\pi_2}(s, a) = Q^{\pi_2}(s, a) - V^{\pi_2}(s)$ is the advantage under π_2. In the autoregressive setting, states are $s_t = (x, y_{<t})$ and actions are tokens v. Expanding $A^{\pi_2} = Q^{\pi_2} - V^{\pi_2}$ and noting that $V^{\pi_2}(s_t)$ is constant w.r.t. the sum over v:

$$\begin{aligned}
\sum_v \pi_1(v|s_t)\, A^{\pi_2}(s_t, v) &= \sum_v \pi_1(v|s_t)\, Q^{\pi_2}(s_t, v) - V^{\pi_2}(s_t) \underbrace{\sum_v \pi_1(v|s_t)}_{=1} \\
&= \sum_v \pi_1(v|s_t)\, Q^{\pi_2}(s_t, v) - \underbrace{\sum_v \pi_2(v|s_t)\, Q^{\pi_2}(s_t, v)}_{=V^{\pi_2}(s_t)} \\
&= \sum_v \left(\pi_1(v|s_t) - \pi_2(v|s_t)\right) Q^{\pi_2}(s_t, v).
\end{aligned} \tag{B.3}$$

Finally, $Q^{\pi_2}(s_t, v) \in [0, 1]$ because it is a conditional expectation of $R \in [0, 1]$. □

Theorem 2 (Suboptimality decomposition for OPD)**.** *Let π_θ be a student and π_T a teacher, with $\pi^* = \arg\max_\pi J(\pi)$, $J(\pi) = \mathbb{E}_x[\mathbb{E}_{y \sim \pi}[R(x, y)]]$, $R \in [0, 1]$, and response length T. The student's suboptimality decomposes as:*

$$J(\pi^*) - J(\pi_\theta) \;\le\; \underbrace{J(\pi^*) - J(\pi_T)}_{\textit{teacher gap}} + T \cdot \underbrace{\sqrt{\bar{\mathcal{L}}_{\mathrm{OPD}}/2}}_{\textit{distillation error}}, \tag{B.4}$$

where $\bar{\mathcal{L}}_{\mathrm{OPD}} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{s_t \sim \rho^t_{\pi_\theta}}[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s_t) \,\|\, \pi_T(\cdot \mid s_t))]$ is the average per-token reverse KL under the student's own context distribution. The bound identifies two independent error sources: a distillation error *that vanishes as OPD converges ($\pi_\theta \to \pi_T$), and a* teacher gap *that persists regardless of optimization quality. Because the T-dependent prefactor makes the bound a qualitative mechanism decomposition rather than a tight numerical certificate, its practical value is directional: the decomposition shows that, all else equal, a stronger teacher (smaller gap) admits a tighter bound on student suboptimality.*

Proof. Decompose the gap as $J(\pi^*) - J(\pi_\theta) = [J(\pi^*) - J(\pi_T)] + [J(\pi_T) - J(\pi_\theta)]$. For the second term, apply Lemma 1 with $\pi_1 = \pi_\theta$ and $\pi_2 = \pi_T$, which gives $J(\pi_\theta) - J(\pi_T) = \sum_{t=1}^{T} \mathbb{E}_{s_t \sim \rho^t_{\pi_\theta}}[\sum_v (\pi_\theta(v \mid s_t) - \pi_T(v \mid s_t))\, Q^{\pi_T}(s_t, v)]$. Multiplying by -1 flips both the performance difference and the inner difference:

$$J(\pi_T) - J(\pi_\theta) = \sum_{t=1}^{T} \mathbb{E}_{s_t \sim \rho^t_{\pi_\theta}} \left[\sum_v \left(\pi_T(v \mid s_t) - \pi_\theta(v \mid s_t)\right) Q^{\pi_T}(s_t, v) \right]. \tag{B.5}$$

Note that this orientation places the expectation under the *student's* own context distribution. Since $Q^{\pi_T}(s_t, v) \in [0,1]$ (it is a conditional expectation of $R \in [0,1]$), the range $\max_v Q - \min_v Q \leq 1$. Because $\sum_v(\pi_T(v|s_t) - \pi_\theta(v|s_t)) = 0$, we may subtract any constant from Q inside the sum. Choosing $c = (\max Q + \min Q)/2$ and applying Hölder's inequality: $|\sum_v(\pi_T - \pi_\theta)Q| = |\sum_v(\pi_T - \pi_\theta)(Q - c)| \leq \|Q - c\|_\infty \cdot \|\pi_T - \pi_\theta\|_1 = \frac{\max Q - \min Q}{2} \cdot 2\,\mathrm{TV} \leq \mathrm{TV}$, where $\mathrm{TV}(p,q) = \frac{1}{2}\sum_v |p(v) - q(v)|$. Therefore:

$$J(\pi_T) - J(\pi_\theta) \leq \sum_{t=1}^{T} \mathbb{E}_{s_t \sim \rho^t_{\pi_\theta}} [\mathrm{TV}(\pi_T(\cdot|s_t), \pi_\theta(\cdot|s_t))]\,. \tag{B.6}$$

Applying Pinsker's inequality ($\mathrm{TV}(p,q) \leq \sqrt{D_{\mathrm{KL}}(p\|q)/2}$) to each term—valid in either argument order, since TV is symmetric—then Jensen's inequality ($\mathbb{E}[\sqrt{X}] \leq \sqrt{\mathbb{E}[X]}$ since $\sqrt{\cdot}$ is concave):

$$\sum_{t=1}^{T} \mathbb{E}_{s_t \sim \rho^t_{\pi_\theta}} [\mathrm{TV}_t] \leq \sum_{t=1}^{T} \sqrt{\mathbb{E}_{s_t \sim \rho^t_{\pi_\theta}} [D_t]/2} = T \cdot \frac{1}{T} \sum_{t=1}^{T} \sqrt{\mathbb{E}[D_t]/2}, \tag{B.7}$$

where $D_t = D_{\mathrm{KL}}(\pi_\theta(\cdot|s_t)\|\pi_T(\cdot|s_t))$. By Cauchy–Schwarz, $\frac{1}{T}\sum_t \sqrt{a_t} \leq \sqrt{\frac{1}{T}\sum_t a_t}$ for $a_t = \mathbb{E}[D_t]/2 \geq 0$. Therefore $\sum_t \sqrt{\mathbb{E}[D_t]/2} \leq T \cdot \sqrt{\bar{\mathcal{L}}_{\mathrm{OPD}}/2}$, giving the stated bound. The teacher gap $J(\pi^*) - J(\pi_T) \geq 0$ by definition of π^*, with equality iff π_T is itself optimal, i.e., $J(\pi_T) = J(\pi^*)$. □

B.2 Extrapolation Gap and Convergence

Proposition 3 (Safe extrapolation range under a linear trajectory)**.** *Let $\varphi : \Pi \to \mathcal{Z}$ be a representation map such that the training trajectory is linear: $\varphi(\pi_{\theta_n}) = \varphi(\pi_{\theta_0}) + \alpha_n \cdot \mathbf{d}$ for a unit direction $\mathbf{d}$ and monotonically increasing coefficients $0 = \alpha_0 < \alpha_n < \alpha^*$, where $\varphi(\pi^*) = \varphi(\pi_{\theta_0}) + \alpha^* \cdot \mathbf{d}$. The extrapolated teacher $\varphi(\pi_{\mathrm{future}}) = \varphi(\pi_{\theta_0}) + \beta \cdot \alpha_n \cdot \mathbf{d}$ is closer to π^* than the student is, i.e.*

$$\|\varphi(\pi_{\mathrm{future}}) - \varphi(\pi^*)\| < \|\varphi(\pi_{\theta_n}) - \varphi(\pi^*)\|, \tag{B.8}$$

exactly when $1 < \beta < 2\alpha^*/\alpha_n - 1$. This is an immediate consequence of the linear-trajectory assumption—it defines the safe β range rather than derives it from weaker premises—and its value lies in making two properties explicit: (i) any $\beta \in (1, \alpha^*/\alpha_n)$ places the teacher strictly between the student and π^* without overshooting, and (ii) the upper limit $2\alpha^*/\alpha_n - 1$ shrinks as $\alpha_n \to \alpha^*$, so the safe range narrows as training converges.

Proof. The gap between the extrapolated teacher and π^* is $\|\varphi(\pi_{\mathrm{future}}) - \varphi(\pi^*)\| = |\beta\alpha_n - \alpha^*|$. The gap between the current student and π^* is $\alpha^* - \alpha_n$. The extrapolated teacher is closer whenever $|\beta\alpha_n - \alpha^*| < \alpha^* - \alpha_n$, which holds iff $1 < \beta < 2\alpha^*/\alpha_n - 1$. For $\beta \in (1, \alpha^*/\alpha_n)$, we have $\beta\alpha_n < \alpha^*$, so the teacher lies strictly between the student and π^*. For $\beta \in [\alpha^*/\alpha_n, 2\alpha^*/\alpha_n - 1)$, the teacher reaches or overshoots π^* (with equality at $\beta = \alpha^*/\alpha_n$) but remains closer to it than the student is. □

Corollary 4 (RISE contracts the suboptimality bound)**.** *Consider one RISE iteration: RLVR updates $\theta_n \to \theta'_{n+1}$ (advancing the trajectory coefficient $\alpha_n \to \alpha'_{n+1} < \alpha^*$), then OPD distills the extrapolated teacher π_{future} into θ'_{n+1} to yield θ_{n+1}. Under the assumptions of Proposition 3 (applied with $\pi_{\theta'_{n+1}}$ in the role of the current policy) and L-Lipschitz continuity of J in φ-space, the Lipschitz suboptimality bound contracts:*

$$\underbrace{J(\pi^*) - J(\pi_{\theta_{n+1}})}_{\textit{post-OPD}} \leq \underbrace{\gamma(\beta)}_{<1} \cdot \underbrace{L(\alpha^* - \alpha'_{n+1})}_{\geq J(\pi^*) - J(\pi_{\theta'_{n+1}})\ \textit{(pre-OPD)}} + T\sqrt{\bar{\mathcal{L}}_{\mathrm{OPD}}/2}, \tag{B.9}$$

where the contraction factor is governed by the teacher's distance to π^*,

$$\gamma(\beta) = \frac{|\alpha^* - \beta\alpha'_{n+1}|}{\alpha^* - \alpha'_{n+1}}, \tag{B.10}$$

and $\gamma(\beta) < 1$ exactly when β lies in the range $(1,\, 2\alpha^*/\alpha'_{n+1} - 1)$ established by Proposition 3. In the idealized limit $\bar{\mathcal{L}}_{\mathrm{OPD}} \to 0$ the residual term vanishes and the bound contracts by $\gamma(\beta) < 1$. In practice, OPD runs for only a few gradient steps and deliberately does not converge to π_{future}—it acts as a trust-region projection (§3) that moves the policy toward *the extrapolated teacher while remaining anchored near $\pi_{\theta'_{n+1}}$. The corollary's role is therefore directional: it identifies the extrapolated teacher as a descent direction for the bound rather than prescribing full convergence, which §4.2 confirms is unnecessary and even harmful (the w/o-OPD variant that adopts π_{future} directly shows negligible gains).*

Proof. Before OPD, the post-RLVR student $\pi_{\theta'_{n+1}}$ lies at φ-distance $\alpha^* - \alpha'_{n+1}$ from π^*, so L-Lipschitz continuity of J in φ-space gives

$$J(\pi^*) - J(\pi_{\theta'_{n+1}}) \le L(\alpha^* - \alpha'_{n+1}). \tag{B.11}$$

The extrapolated teacher sits at φ-distance $|\alpha^* - \beta\alpha'_{n+1}|$ from π^*. Applying L-Lipschitz continuity again and factoring out the pre-OPD bound:

$$J(\pi^*) - J(\pi_{\mathrm{future}}) \le L\big|\alpha^* - \beta\alpha'_{n+1}\big| = \frac{\big|\alpha^* - \beta\alpha'_{n+1}\big|}{\alpha^* - \alpha'_{n+1}} \cdot L(\alpha^* - \alpha'_{n+1}) = \gamma(\beta) \cdot L(\alpha^* - \alpha'_{n+1}). \tag{B.12}$$

Lipschitz continuity therefore constrains $J(\pi_{\mathrm{future}})$ through the teacher's distance to π^* alone, irrespective of which side of π^* the teacher falls on; this is what makes Eq. B.10 valid for overshooting as well as non-overshooting β.

We then apply Theorem 2 with teacher $\pi_T = \pi_{\mathrm{future}}$ and student $\pi_\theta = \pi_{\theta_{n+1}}$, the checkpoint produced by OPD. Substituting the teacher gap bounded above gives

$$J(\pi^*) - J(\pi_{\theta_{n+1}}) \;\le\; \underbrace{J(\pi^*) - J(\pi_{\mathrm{future}})}_{\le \gamma(\beta) L(\alpha^* - \alpha'_{n+1})} + T\sqrt{\bar{\mathcal{L}}_{\mathrm{OPD}}/2}, \tag{B.13}$$

which is Eq. B.9. Finally, $\gamma(\beta) < 1$ holds iff $|\alpha^* - \beta\alpha'_{n+1}| < \alpha^* - \alpha'_{n+1}$, which by Proposition 3 is exactly the condition $1 < \beta < 2\alpha^*/\alpha'_{n+1} - 1$. Therefore, once the distillation error vanishes, the post-OPD bound is $\gamma(\beta)$ times the pre-OPD bound, hence strictly smaller. □

Remark 2 (Limitations of the linear assumption)**.** *Proposition 3 and Corollary 4 assume a perfectly linear trajectory in φ-space. In practice, linearity holds only approximately and locally: empirical studies of task vectors and linear mode connectivity (Ilharco et al., 2022; Frankle et al., 2020) show that interpolation and modest extrapolation preserve performance, but large extrapolation factors β can exit the linear regime and degrade the teacher quality. The bounds above should therefore be understood as characterizing the* mechanism *by which extrapolation helps—a teacher closer to π^* yields a tighter suboptimality bound—rather than as providing exact convergence rates. In our experiments, we find that moderate values of β consistently improve over self-distillation, consistent with this local-linearity picture.*

B.3 How Low-Dimensional Are Our Trajectories?

We quantify the gap between the linear assumption and the actual training dynamics, refining the low-rank picture reported in prior work (Cai et al., 2025; Wang et al., 2026a) with a direct measurement on our own runs. For GRPO checkpoints of Qwen3-1.7B and Qwen3-8B on DAPOMath, we form the displacement $\delta_t = \theta_t - \theta_0$ at each saved step and compute the Gram matrix $\langle \delta_i, \delta_j \rangle$ over all parameter tensors; its eigenspectrum gives the principal components of the trajectory (Fig. B.1). An exactly linear trajectory would be rank one, placing all variance in a single component. The leading component accounts for 68.2% at *both* scales, and the top three for 86.6% and 88.5% respectively. That the two agree to within 0.1% on the leading component despite a $5\times$ difference in parameter count suggests this is a property of the training dynamics rather than of a particular model.

A single direction therefore does not capture the trajectory exactly: the residual 31.8% lies orthogonal to the leading component. What the measurement does show is that the trajectory is confined to a genuinely low-dimensional subspace: three components capture $\sim 87\%$ of the variance at both

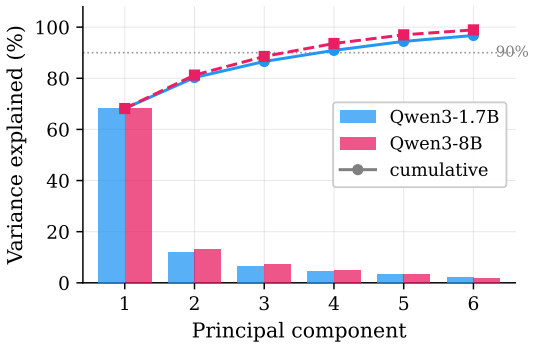


Figure B.1: **The RLVR weight trajectory is low-dimensional.** Principal spectrum of the displacements $\delta_t = \theta_t - \theta_0$ from GRPO checkpoints on DAPOMath. Bars show variance explained per component; lines show the cumulative total. A perfectly linear trajectory would place 100% on the first component; the observed 68.2% reflects curvature along the path, while three components capture $\sim 87\%$ at both scales, confining the trajectory to a low-dimensional subspace.

scales, out of a parameter space with billions of degrees of freedom. This confinement bounds the degree of non-linearity—within a rank-3 subspace the leading direction accounts for $\sim 78\%$ of the subspace variance, limiting the magnitude of orthogonal curvature. For RISE's operating range ($\beta \leq 1.2$, i.e. extending 20% beyond the current displacement), the error introduced by the linear approximation is therefore controlled by the small residual variance in the orthogonal components.

Fig. 6 provides empirical confirmation: extrapolation gains persist out to $\beta \approx 1.3$ and the safe range contracts as the policy converges, consistent with Proposition 3's prediction even though its linear premise holds only approximately.

B.4 Top-K Approximation Bias

As described in §3.2, we approximate the logit-space extrapolated teacher by selecting $S = \text{Top}_K(\pi_{\theta'_{n+1}})$ and projecting the post-RLVR checkpoint $\pi_{\theta'_{n+1}}$ and the anchor π_{θ_n} onto the same $(K{+}1)$-simplex before applying the extrapolation (Eq. 6). This differs from the unbiased procedure, which would extrapolate over the full vocabulary *first* and then select $S^* = \text{Top}_K(\pi_{\text{future}})$. The approximation error is captured by the set difference $S \triangle S^*$: tokens outside $\pi_{\theta'_{n+1}}$'s top-K but with a large log-ratio $\log \pi_{\theta'_{n+1}} / \pi_{\theta_n}$ can be promoted into S^* by extrapolation, yet are absent from S.

In practice this discrepancy is negligible. With $K = 100$, the top-K tokens of $\pi_{\theta'_{n+1}}$ account for $>99\%$ of the probability mass under typical autoregressive LLM distributions, leaving a residual tail mass $\varepsilon = 1 - \sum_{v \in S} \pi_{\theta'_{n+1}}(v) < 0.01$. This does not formally bound $|S \triangle S^*|$, since extrapolation with $\beta > 1$ can in principle amplify tail tokens with large log-ratios. However, for the moderate β values used in our experiments, almost all of π_{future}'s mass concentrates on tokens already in S, so the induced bias is negligible in practice. For weight-space extrapolation, no such approximation arises: the forward pass through θ_{future} directly produces π_{future}, and top-K is applied to the exact extrapolated distribution.

C Algorithm

We present the full pseudocode for both RISE variants. Algorithm 1 describes logit-space extrapolation, where the teacher distribution is constructed by extrapolating cached top-K logits from the post-RLVR checkpoint and the anchor without materializing a separate model. Algorithm 2 describes weight-space extrapolation, where the extrapolated parameters θ_{future} are materialized explicitly and a forward pass through them produces the teacher distribution. Both variants share the same two-phase structure—RLVR followed by OPD on the same rollouts—and differ only in how π_{future} is computed.

Algorithm 1 RISE (logit-space). The teacher is built *once* per iteration, before the OPD loop.

Require: Initial policy π_{θ_0}, extrapolation scale β_0, top-K, anchor rate η, iterations N

Initialize anchor $\theta_{\text{anchor}} \leftarrow \theta_0$
for $n = 0, 1, \ldots, N-1$ **do**
  $\beta_n \leftarrow 1 + (\beta_0 - 1) \cdot (1 - n/N)$ ▷ Extrapolation decay
  // RLVR phase
  Sample rollouts $\mathcal{D} = \{(x_i, y_i)\} \sim \pi_{\theta_n}$, compute rewards $R(x_i, y_i)$
  $\theta'_{n+1} \leftarrow \text{PolicyGradient}\big(\theta_n, \{(x_i, y_i, R_i)\}\big)$
  // Teacher construction ▷ two forward passes over $\mathcal{D}$
  At every position of $\mathcal{D}$: set $S \leftarrow \text{Top}_K(\pi_{\theta'_{n+1}})$ and project $\pi_{\theta'_{n+1}}, \pi_{\theta_{\text{anchor}}}$ onto $S \cup \{\text{tail}\}$
  Cache $\log \pi_{\text{future}} \leftarrow \log \pi_{\theta_{\text{anchor}}} + \beta_n \cdot (\log \pi_{\theta'_{n+1}} - \log \pi_{\theta_{\text{anchor}}})$ ▷ Eq. 6
  // OPD phase
  $\theta \leftarrow \theta'_{n+1}$
  for each minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
    Retrieve cached π_{future} at the positions of $\mathcal{B}$ ▷ no teacher forward pass
    $\theta \leftarrow \text{Optimizer}(\theta,\ \nabla_\theta \mathcal{L}_{\text{RISE}}(\mathcal{B}))$ ▷ Eq. 7
  end for
  $\theta_{n+1} \leftarrow \theta$
  Update anchor: $\theta_{\text{anchor}} \leftarrow (1-\eta)\,\theta_{\text{anchor}} + \eta\,\theta_{n+1}$ ▷ $\eta=1$: previous ckpt; $\eta<1$: EMA
end for
return π_{θ_N}

Algorithm 2 RISE (weight-space). The teacher is evaluated *inside* the OPD loop.

Require: Initial policy π_{θ_0}, extrapolation scale β_0, top-K, anchor rate η, iterations N

Initialize anchor $\theta_{\text{anchor}} \leftarrow \theta_0$
for $n = 0, 1, \ldots, N-1$ **do**
  $\beta_n \leftarrow 1 + (\beta_0 - 1) \cdot (1 - n/N)$ ▷ Extrapolation decay
  // RLVR phase
  Sample rollouts $\mathcal{D} = \{(x_i, y_i)\} \sim \pi_{\theta_n}$, compute rewards $R(x_i, y_i)$
  $\theta'_{n+1} \leftarrow \text{PolicyGradient}\big(\theta_n, \{(x_i, y_i, R_i)\}\big)$
  // Teacher construction
  Materialize $\theta_{\text{future}} \leftarrow \theta_{\text{anchor}} + \beta_n \cdot (\theta'_{n+1} - \theta_{\text{anchor}})$ ▷ Eq. 3
  // OPD phase
  $\theta \leftarrow \theta'_{n+1}$
  for each minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
    At every position of $\mathcal{B}$: set $S \leftarrow \text{Top}_K(\pi_\theta)$ from the student forward pass
    Forward pass through θ_{future} to get π_{future} on $S \cup \{\text{tail}\}$ ▷ one teacher forward per step
    $\theta \leftarrow \text{Optimizer}(\theta,\ \nabla_\theta \mathcal{L}_{\text{RISE}}(\mathcal{B}))$ ▷ Eq. 7
  end for
  $\theta_{n+1} \leftarrow \theta$
  Update anchor: $\theta_{\text{anchor}} \leftarrow (1-\eta)\,\theta_{\text{anchor}} + \eta\,\theta_{n+1}$
end for
return π_{θ_N}

D Experiment Setup

This appendix supplements the setup summary in §4 with full training hyperparameters (§D.1), evaluation protocol details (§D.2), and a description of the OPSD baselines and their shared privileged-teacher construction (§D.3).

D.1 Training Setup

All runs use the VeRL framework on a single node with 8 GPUs, training for one epoch over the respective training set. We use AdamW with a constant learning rate and no warmup, GRPO advantage estimation without standard-deviation normalization, and token-level importance-sampling

correction (clip threshold 2.0). Prompts are truncated to 2,048 tokens and responses to 8,192 tokens. All methods—GRPO, the three OPSD baselines, and both RISE variants—share the hyperparameters in Table D.1 within each configuration, so performance differences are attributable to the teacher construction rather than training dynamics.

Table D.1: Training hyperparameters. n denotes the number of rollouts sampled per prompt. *Train batch size* is the number of prompts sampled per iteration for rollout generation; *mini-batch size* is the number of prompts per gradient update within each iteration (the train batch is split into train batch size/mini-batch size gradient steps). The OPD phase uses the same mini-batch size as the RLVR phase. Qwen2.5-3B agentic runs follow GIGPO's setup (Feng et al., 2025); see their paper for full details.

	Qwen3-1.7B (DAPOMath)	Qwen3-1.7B-Base (DAPOMath)	Qwen3-8B (DAPOMath)	Qwen3-4B-Base (multi-domain)	Qwen3-8B-Base (code)	OLMo3-7B-Instruct-SFT (OpenR1-Math)
Learning rate	1e-6	1e-6	1e-6	1e-6	1e-6	1e-6
Train batch size	64	64	64	64	64	128
Mini-batch size	32	32	32	32	32	64
Rollouts n	8	8	8	8	8	8
Epochs	1	1	1	1	1	1
Rollout temperature	1.0	1.0	1.0	1.0	1.0	1.0
KL coefficient	0	0	0	0	0	0
RISE β_0	1.2	1.2	1.2	1.2	1.2	1.2
RISE β_N	1.0	1.0	1.0	1.0	1.0	1.0
RISE η	0.1	0.1	0.1	0.1	0.1	1.0
RISE K	100	100	100	100	20	100

D.2 Evaluation Setup

Benchmarks. Mathematical reasoning is evaluated on MATH-500 (Lightman et al., 2024), AIME 2024/2025 (Jia, 2024; math ai, 2025), AMC 2023 (MAA, 2023), Minerva (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024), with GPQA-Diamond (Rein et al., 2023), IFEval (Zhou et al., 2023), and MMLU-Pro (Wang et al., 2024) as out-of-distribution checks. The multi-domain suite adds SuperGPQA (Du et al., 2026) and TheoremQA (Chen et al., 2023). Code generation is evaluated on HumanEval+, MBPP+ (Liu et al., 2023), and LiveCodeBench v6 (Jain et al., 2025). Agentic tasks use ALFWorld (Shridhar et al., 2020) (success rate: fraction of episodes completed) and WebShop (Yao et al., 2022), which reports two metrics: *Score* ($100\times$ average reward, where reward is a product of attribute recall, option selection, price satisfaction, and category match) and *Acc* (success rate: fraction of episodes achieving a perfect reward of 1.0).

Sampling protocol. Evaluation samples are drawn stochastically with temperature 1.0, top-p = 1.0, and top-k = -1 (i.e., unrestricted), matching the rollout distribution used during training. The number of samples per prompt is set per benchmark according to its size and variance: 16 for AIME 2024/2025, GPQA-Diamond, and Minerva, 8 for OlympiadBench, and 4 for MATH-500 and AMC 2023. Deterministic benchmarks that admit a single correct completion—IFEval, MMLU-Pro, SuperGPQA, and TheoremQA—use a single sample. All Qwen3 models are evaluated in non-thinking mode (thinking is disabled). For code generation we use 4 samples on HumanEval+, MBPP+, and LiveCodeBench.

We report two complementary metrics. *avg@*N is the mean accuracy across N sampled completions, measuring expected single-sample quality. *pass@*N estimates the probability that at least one of N independent draws is correct: for each problem we draw 1,000 bootstrap subsets of size N (with replacement) from the N sampled completions, record whether each subset contains a correct solution, and average; this per-problem estimate is then averaged across prompts. Reporting both separates per-sample reliability from exploration breadth: a method can improve avg@N while leaving pass@N unchanged (sharpening the distribution) or improve both (genuinely expanding coverage).

D.3 BASELINES

All three OPSD baselines—GRPO+SDPO (Hübotter et al., 2026), SDAR (Lu et al., 2026), and RLSD (Yang et al., 2026a)—share the same *privileged teacher* and differ only in how its signal enters the update.

Privileged teacher construction. For each prompt, the model generates n rollouts. If the group contains at least one correct solution, the teacher is the model re-prompted with that correct solution as privileged context, and its token distribution serves as the distillation target for all responses to that prompt. If no rollout in the group succeeds, the prompt contributes only the RLVR loss—no distillation signal is available. The teacher is maintained as an exponential moving average of the student (rate 0.05), following the recommended hyperparameters in Hübotter et al. (2026). Averaged over training, the teacher is active on $\sim 85\%$ of prompts for Qwen3-8B and $\sim 70\%$ for Qwen3-1.7B, so the baselines' limited gains are not attributable to low coverage.

Integration strategies. GRPO+SDPO adds a KL divergence toward the teacher's token distribution as an auxiliary loss alongside the policy gradient. SDAR uses the teacher only as a *gate*: the teacher–student probability gap scales the GRPO advantage per token, upweighting positions where the teacher is more confident than the student. RLSD similarly uses the teacher's per-token signal to reweight advantage estimates, providing finer-grained credit assignment than a single sequence-level reward.

Contrast with RISE. All three baselines require privileged information—a correct solution the student did not produce—so their teacher is available only on problems where such a solution exists in the rollout group. RISE derives its teacher from the training trajectory itself and therefore applies uniformly to every prompt, regardless of whether any rollout succeeded.

E EXPERIMENT RESULTS

This appendix provides supplementary results for the experiments in §4: Qwen3-1.7B-Base results (§E.1), per-metric convergence curves (§E.2), code generation convergence (§E.3), grounding and OPD ablation details (§E.5), extrapolation strength and schedule ablations (§E.6–§E.8), a compute-matched comparison (§E.9), and a multi-seed reproducibility analysis (§E.10).

E.1 QWEN3-1.7B-BASE RESULTS

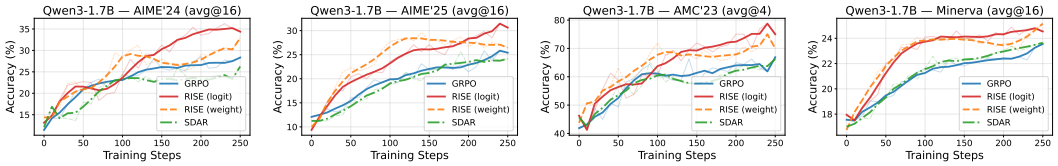
Table E.1 reports results for Qwen3-1.7B-Base on DAPOMath. RISE (logit) achieves the best in-domain average (29.2) and competitive OOD performance, while RISE (weight) leads on MATH-500 (62.4) and Minerva (20.7). The margin over GRPO is the narrowest of any configuration ($+1.2$ Math Avg), consistent with the capability ceiling of a base model that has not been instruction-tuned.

Table E.1: **Qwen3-1.7B-Base (DAPOMath).** Accuracy (%) on math and OOD benchmarks. Best in **bold**, second best underlined.

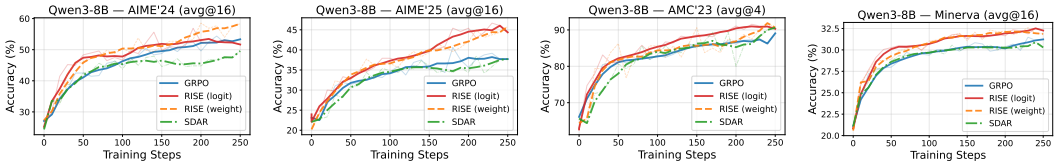
Method	In-Domain Math							OOD			
	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.	GPQA	IFEval	MMLU	Avg.
Base	40.0	4.4	2.9	27.5	11.6	19.8	17.7	12.7	14.4	8.0	11.7
GRPO	60.6	9.4	6.9	42.5	19.3	29.2	28.0	26.5	31.1	39.6	32.4
GRPO+SDPO	61.3	8.3	6.7	41.9	20.1	**32.3**	28.4	**28.7**	30.5	40.5	**33.2**
SDAR	55.6	9.6	**8.3**	43.1	17.7	27.1	26.9	23.2	28.5	42.5	31.4
RLSD	57.2	11.9	7.5	43.1	17.8	28.2	27.6	23.0	29.0	42.4	31.5
RISE (logit)	59.6	**12.1**	8.1	**45.6**	20.2	29.4	**29.2**	25.4	**31.1**	**42.9**	33.1
RISE (weight)	**62.4**	10.4	5.8	38.8	**20.7**	30.8	28.1	25.7	29.9	41.0	32.2

E.2 PER-METRIC CONVERGENCE CURVES

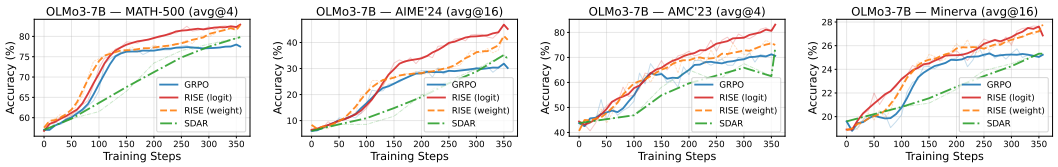
Figure E.1 shows per-metric training curves for all model configurations discussed in §4.1.



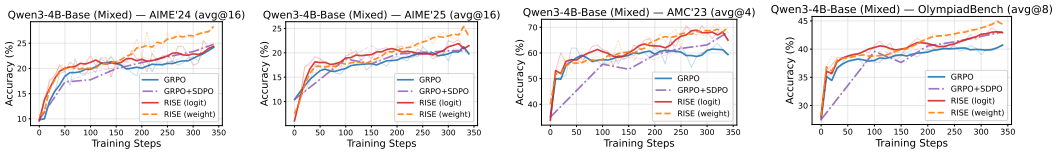
(a) Qwen3-1.7B (DAPOMath)



(b) Qwen3-8B (DAPOMath)



(c) OLMo3-7B-Instruct-SFT (OpenR1Math)



(d) Qwen3-4B-Base (mixed math + STEM)

Figure E.1: **Per-metric convergence curves** for all model configurations. Each row shows four representative benchmarks. RISE (both variants) consistently reaches higher accuracy in fewer training steps.

E.3 Code Generation Convergence Curves

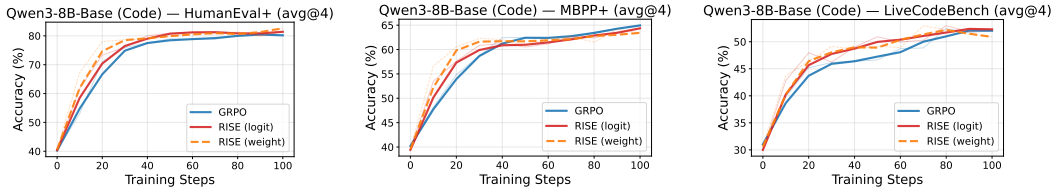
Figure E.2 shows avg@4 and pass@4 convergence curves for Qwen3-8B-Base trained on Skywork-OR1-Code, evaluated on HumanEval+, MBPP+, and LiveCodeBench. Both RISE variants converge faster than GRPO in early training across all three benchmarks.

E.4 Average vs. Best-of-N Accuracy

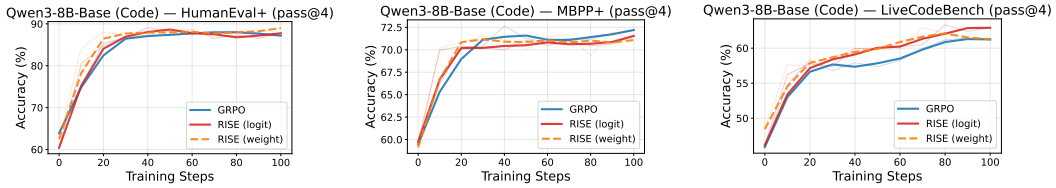
Table E.2 reports avg@16 and pass@16 on the AIME benchmarks, isolating whether RISE's gains come from sharpening the policy around solutions it already finds or from expanding the set of solvable problems. If distillation merely concentrated probability mass on existing solutions, avg@16 would improve while pass@16 stagnated or declined. Instead, RISE improves both, and at 1.7B the pass@16 gains exceed the avg@16 gains on AIME'24 ($+9.6$ vs. $+6.3$ for RISE (logit)), indicating genuinely broadened coverage. At 8B the coverage margins are smaller, consistent with a stronger base policy leaving less headroom; the one regression is RISE (logit) on AIME'25 pass@16 (-0.6), where the weight-space variant instead gains $+3.9$.

E.5 Grounding and OPD Ablation

Figure E.3 plots training reward and policy entropy for the grounding ablation (§4.2). Without RLVR, mean training reward collapses from 0.56 to 0.00 by step 60, tracking the accuracy cliff in Fig. 5a. Policy entropy spikes to 0.78 at step 49—just before the final crash—then drops to near



(a) HumanEval+ (avg@4) (b) MBPP+ (avg@4) (c) LiveCodeBench (avg@4)



(d) HumanEval+ (pass@4) (e) MBPP+ (pass@4) (f) LiveCodeBench (pass@4)

Figure E.2: **Code generation convergence curves (Qwen3-8B-Base).** Both RISE variants consistently converge faster than GRPO across all three benchmarks under both avg@4 and pass@4 metrics.

Table E.2: **Average vs. best-of-16 accuracy (%) on AIME benchmarks.** Δ denotes improvement over GRPO. RISE improves pass@16 (coverage) alongside avg@16 (mean quality), with the largest coverage gains at 1.7B.

Model	Method	AIME'24 avg@16	AIME'24 pass@16	AIME'25 avg@16	AIME'25 pass@16
Qwen3-8B	GRPO	54.4	78.7	42.9	63.3
	RISE (logit)	56.9 +2.5	79.0 +0.3	46.7 +3.8	62.7 -0.6
	RISE (weight)	58.1 +3.7	81.5 +2.8	45.8 +2.9	67.2 +3.9
Qwen3-1.7B	GRPO	30.0	54.4	26.3	45.8
	RISE (logit)	36.3 +6.3	64.0 +9.6	33.3 +7.0	53.9 +8.1
	RISE (weight)	32.9 +2.9	57.0 +2.6	31.3 +5.0	49.2 +3.4

zero, consistent with the policy degenerating into repetitive non-terminating generation. GRPO and RISE show stable entropy throughout.

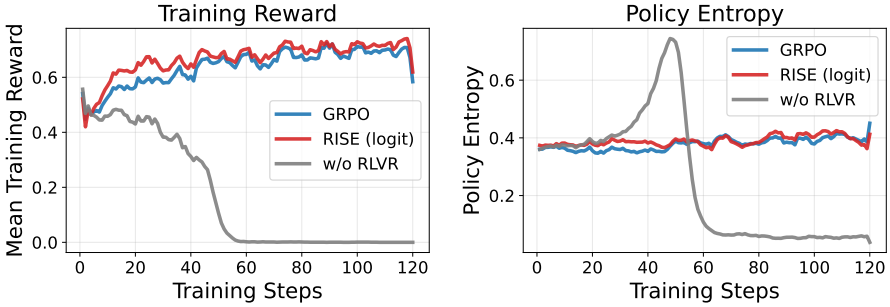


Figure E.3: **Grounding ablation: training reward and policy entropy.** Same three Qwen3-8B runs as Fig. 5a. Without RLVR (grey), training reward collapses to zero (left) while policy entropy spikes then crashes (right), confirming the phase-transition nature of the collapse.

Table E.3 reports the full without-OPD ablation results discussed in §4.2. Across all three model scales, removing the OPD phase (i.e., directly adopting θ_{future} without distillation) leaves Math Avg essentially unchanged relative to GRPO (60.0→60.3 at 8B, 45.4→45.6 at 1.7B, 28.0→27.7 at 1.7B-Base), forfeiting the gains that RISE achieves. Per-benchmark accuracy shifts—e.g., the no-OPD

arm gains on OlympiadBench at 8B but loses on the AIME pair, while the pattern reverses at 1.7B—but these redistributions do not translate into consistent improvement without the distillation phase.

Table E.3: **Without-OPD ablation.** Accuracy (%) on math benchmarks. The w/o OPD variant adopts θ_{future} directly without distillation. All runs are configuration-matched to their RISE counterpart.

Model	Method	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.
Qwen3-8B	GRPO	83.8	54.4	42.9	89.4	31.5	57.9	60.0
	RISE (logit)	84.8	56.9	46.7	91.6	32.5	62.4	62.5
	RISE (weight)	84.4	58.1	45.8	91.9	32.6	63.3	62.7
	w/o OPD	84.6	51.7	39.8	90.6	31.2	64.1	60.3
Qwen3-1.7B	GRPO	75.6	30.0	26.3	65.6	23.9	51.1	45.4
	RISE (logit)	75.2	36.3	33.3	78.8	24.9	52.9	50.2
	RISE (weight)	77.3	32.9	31.3	75.0	25.7	53.3	49.2
	w/o OPD	72.0	32.3	27.3	70.0	24.6	47.4	45.6
Qwen3-1.7B -Base	GRPO	60.6	9.4	6.9	42.5	19.3	29.2	28.0
	RISE (logit)	59.6	12.1	8.1	45.6	20.2	29.4	29.2
	RISE (weight)	62.4	10.4	5.8	38.8	20.7	30.8	28.1
	w/o OPD	58.8	10.6	5.8	43.1	19.4	28.6	27.7

E.6 Extrapolation Strength and Decay Schedule

Table E.4 reports per-benchmark results for the β_0 and decay schedule ablations summarised in Table 4. All runs use logit-space extrapolation.

Table E.4: **Detailed β_0 and decay schedule ablation results.** Accuracy (%) on individual math benchmarks. The top block varies β_0 with linear decay; the middle block varies the schedule with $\beta_0 = 1.2$. Both blocks use Qwen3-8B. The bottom block varies β_0 on Qwen3-1.7B-Base (schedule noted in parentheses).

Configuration	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.
Qwen3-8B — β_0 sensitivity (linear decay)							
$\beta_0 = 1.2$	84.8	56.9	46.7	91.6	32.5	62.4	**62.5**
$\beta_0 = 1.3$	83.3	56.9	47.3	92.5	32.1	61.9	62.3
$\beta_0 = 1.5$	83.5	55.2	47.9	91.2	31.9	61.6	61.9
Qwen3-8B — decay schedule ($\beta_0 = 1.2$)							
Linear	84.8	56.9	46.7	91.6	32.5	62.4	62.5
Cosine	84.3	58.5	47.7	91.8	32.0	63.8	**63.0**
Exponential	82.4	57.5	47.1	91.2	31.7	61.8	62.0
Fixed ($\beta = \beta_0$)	83.8	56.2	44.4	90.6	31.6	63.9	61.8
Qwen3-1.7B-Base							
$\beta_0 = 1.2$ (linear)	59.6	12.1	8.1	45.6	20.2	29.4	29.2
$\beta_0 = 1.2$ (cosine)	58.6	10.8	8.3	42.5	19.8	29.7	28.3
$\beta_0 = 1.3$ (linear)	61.0	11.7	8.3	46.9	19.1	29.4	**29.4**
$\beta_0 = 1.5$ (linear)	59.2	11.0	6.9	45.6	19.5	28.6	28.5

E.7 Resample vs. Rollout Reuse

Table E.5 reports per-benchmark results for the resample ablation discussed in §4.3. All runs use logit-space extrapolation with $\beta_0 = 1.2$ and linear decay.

E.8 Anchor Dynamics (η) Ablation

Table E.6 reports per-benchmark results for the anchor ablation discussed in §4.3. All runs use $\beta_0 = 1.2$. On OLMo3-7B-Instruct-SFT, the drop from EMA smoothing is substantial (-6.3 logit,

Table E.5: **Resample vs. rollout reuse.** Accuracy (%) on individual math benchmarks. "Resample" generates fresh rollouts from $\pi_{\theta'_{n+1}}$ for OPD; "Rollout reuse" reuses RLVR rollouts.

Model	Variant	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.
Qwen3-8B	Rollout reuse	84.8	56.9	46.7	91.6	32.5	62.4	62.5
	Resample	83.7	58.5	47.9	90.6	31.3	63.1	62.5
Qwen3-1.7B -Base	Rollout reuse	59.6	12.1	8.1	45.6	20.2	29.4	29.2
	Resample	59.6	11.9	7.5	43.8	18.8	29.0	28.4

-2.7 weight), likely because GRPO on OLMo produces large, stable per-step displacements that do not benefit from smoothing.

Table E.6: **Anchor ablation.** Accuracy (%) on individual math benchmarks. $\eta = 1$ uses the previous checkpoint as anchor; $\eta < 1$ uses an EMA anchor. EMA helps on Qwen3-1.7B but degrades on OLMo3-7B-Instruct-SFT.

Configuration	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.
Qwen3-1.7B — Logit-space							
$\eta = 1.0$	77.4	29.4	26.0	71.3	24.8	48.6	46.2
$\eta = 0.95$	75.4	32.7	29.8	73.1	25.0	50.2	47.7
$\eta = 0.3$	74.6	33.3	29.4	76.9	25.5	50.6	48.4
$\eta = 0.1$	75.2	36.3	33.3	78.8	24.9	52.9	**50.2**
Qwen3-1.7B — Weight-space							
$\eta = 1.0$	75.8	31.3	29.2	67.5	24.5	50.0	46.4
$\eta = 0.3$	76.3	31.3	29.0	73.1	24.8	52.5	47.8
$\eta = 0.1$	77.3	32.9	31.3	75.0	25.7	53.3	**49.2**
OLMo3-7B-Instruct-SFT — Logit-space							
$\eta = 1.0$	82.9	46.9	36.0	83.1	27.6	61.9	**56.4**
$\eta = 0.1$	78.4	31.3	32.3	76.9	25.4	56.2	50.1
OLMo3-7B-Instruct-SFT — Weight-space							
$\eta = 1.0$	82.8	42.5	32.5	76.9	27.8	59.6	**53.7**
$\eta = 0.1$	80.6	35.0	31.0	75.0	26.6	57.9	51.0

E.9 Compute-Matched Comparison

Table E.7 reports per-benchmark results for the compute-matched comparison discussed in §4.3. GRPO-2× performs a second inner-loop gradient pass on the same rollouts within each iteration, matching the total gradient budget of RISE's RLVR + OPD phases.

Table E.7: **Compute-matched comparison: RISE vs. GRPO-2×.** Accuracy (%) on math benchmarks. GRPO-2× performs a second inner-loop gradient pass on the same rollouts within each iteration, matching RISE's total gradient budget at equal sampling cost.

Model	Method	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.
Qwen3-8B	GRPO	83.8	54.4	42.9	89.4	31.5	57.9	60.0
	GRPO-2×	84.5	53.5	42.3	92.5	31.4	58.9	60.5
	RISE (logit)	84.8	56.9	46.7	91.6	32.5	62.4	62.5
	RISE (weight)	84.4	58.1	45.8	91.9	32.6	63.3	**62.7**
Qwen3-1.7B	GRPO	75.6	30.0	26.3	65.6	23.9	51.1	45.4
	GRPO-2×	74.5	32.9	30.6	70.6	25.6	49.7	47.3
	RISE (logit)	75.2	36.3	33.3	78.8	24.9	52.9	**50.2**
	RISE (weight)	77.3	32.9	31.3	75.0	25.7	53.3	49.2

E.10 Reproducibility Across Random Seeds

Table E.8 reports mean ± standard deviation across three random seeds for Qwen3-1.7B and Qwen3-8B on DAPOMath. Table 1 reports a single representative seed; the multi-seed statistics confirm that RISE's gains are well outside seed-to-seed variance at both scales. At 1.7B, RISE (logit) improves Math Avg by $+4.5$ over GRPO (49.6 vs. 45.1), roughly $10\times$ the per-method standard deviation (≤ 0.6). At 8B, RISE (weight) improves Math Avg by $+2.2$ over GRPO (62.4 vs. 60.1) and by $+2.8$ over SDAR (59.6), again far exceeding the per-method standard deviation (≤ 0.6). Individual competition benchmarks (AIME, AMC) exhibit larger variance due to small problem sets, but the aggregate Math Avg is stable across all methods.

Table E.8: **Multi-seed reproducibility (DAPOMath): mean ± std across 3 random seeds.** Standard deviations are shown in gray. RISE's Math Avg gains over GRPO far exceed seed-to-seed variance at both scales.

Method	In-Domain Math							OOD			
	MATH500	AIME24	AIME25	AMC23	Minerva	OlyBench	Avg.	GPQA	IFEval	MMLU	Avg.
Qwen3-8B (DAPOMath)											
GRPO	84.0 ±0.5	53.1 ±1.2	42.5 ±0.3	88.5 ±2.1	31.4 ±0.4	61.3 ±2.7	60.1 ±0.2	56.3 ±0.5	82.2 ±0.5	71.6 ±1.2	70.0 ±0.4
SDAR	84.7 ±0.4	48.9 ±1.0	39.1 ±1.2	90.0 ±1.3	30.6 ±0.6	64.2 ±0.7	59.6 ±0.6	55.3 ±0.3	83.2 ±0.6	71.4 ±1.5	69.9 ±0.6
RISE (weight)	84.9 ±0.5	55.8 ±1.7	44.1 ±1.2	92.8 ±0.6	32.3 ±0.2	64.4 ±0.9	**62.4** ±0.2	58.1 ±0.7	84.1 ±0.5	72.3 ±1.0	71.5 ±0.4
Qwen3-1.7B (DAPOMath)											
GRPO	75.4 ±0.1	28.2 ±1.3	26.2 ±0.3	66.5 ±1.6	24.1 ±0.5	50.4 ±0.6	45.1 ±0.4	31.2 ±0.8	68.7 ±0.2	54.4 ±0.7	51.4 ±0.1
RISE (logit)	75.3 ±0.2	34.5 ±1.5	31.7 ±1.5	78.8 ±2.0	25.0 ±0.1	52.4 ±0.7	**49.6** ±0.4	34.3 ±0.5	69.1 ±1.0	55.5 ±1.5	53.0 ±0.7
RISE (weight)	76.5 ±0.6	32.3 ±0.8	30.2 ±0.8	74.8 ±0.5	25.1 ±0.5	51.7 ±1.3	48.4 ±0.6	35.9 ±1.1	68.7 ±1.4	55.3 ±1.5	53.3 ±1.0