Extreme Region Policy Distillation

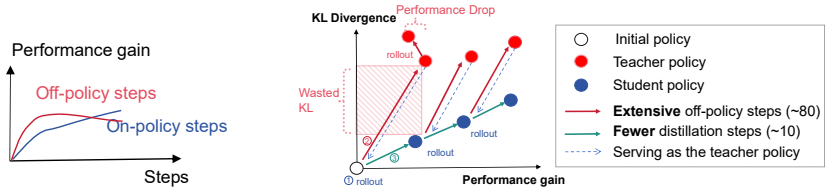
Anonymous authors
Paper under double-blind review

Abstract

Reinforcement learning for large language models faces a fundamental trade-off between sample efficiency and asymptotic performance: on-policy methods require repeatedly generating expensive trajectories, while off-policy reuse introduces distribution mismatch. We find that aggressive optimization on fixed data brings rapid initial gains. Continued updates then shift token probabilities and reduce entropy after performance plateaus. Stronger KL regularization reduces both divergence and attainable performance. This motivates Extreme Region Policy Distillation (ERPD), a two-stage framework that separates aggressive rollout reuse from the final policy update. Stage 1 performs weakly constrained optimization on a fixed rollout batch to construct a strong policy-difference signal. Stage 2 restarts from the policy that generated the batch and optimizes the fixed teacher–reference log-probability difference with a clipped update, retaining useful policy changes while limiting divergence. The resulting policy matches or exceeds its teacher with substantially smaller KL divergence. ERPD also accommodates strong and weak teachers: when the Stage 1 teacher remains below the base model, contrasts between intermediate and recovered checkpoints still provide effective supervision. Across mathematical reasoning benchmarks, iterative ERPD raises Qwen3-4B-Thinking-2507 accuracy on HMMT Nov 25 from 66.6% to 79.0%; at larger scale, it raises Qwen3.5-27B exact-match accuracy on APEX2025 from 2.08% to 14.6%, while improving the other reported transfer benchmarks.

1 Introduction

Reinforcement learning (RL) has substantially improved the reasoning abilities of large language models (LLMs) (Ouyang et al., 2022; Guo et al., 2025). These gains are expensive: long reasoning trajectories must be generated repeatedly, while common on-policy pipelines use each rollout batch for only a small number of optimizer steps. Reusing a fixed batch can improve rollout efficiency, but prolonged optimization creates distribution mismatch and can eventually destabilize training (Schulman et al., 2017; 2015; Xi et al., 2025). Existing clipping and KL penalties manage this tension by limiting each update, potentially leaving useful signal in an expensive batch underexploited.



(a) Previous framework requires a trade-off.

(b) Our two-stage method decouples sample efficiency from KL efficiency, optimizing each separately.

Figure 1: **Fixed-batch trade-off and two-stage ERPD.** "Wasted KL" denotes measured divergence accumulated after validation accuracy plateaus or declines; update counts illustrate the reported runs.

We study the learning signal available beyond the usual few updates. Across fixed-batch experiments, validation accuracy continues to improve for dozens of steps. Further optimization then increases policy divergence and reduces entropy as accuracy plateaus or declines. This separation

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.

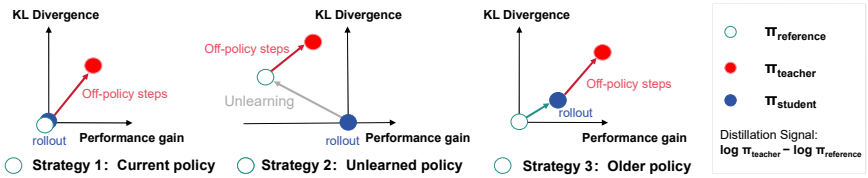


Figure 2: **ERPD score-reference strategies.** The strategies vary the denominator π_{ref} in $\log \pi_e - \log \pi_{\text{ref}}$. By default, the previous-round final policy also generates rollouts, initializes the student, and forms the distillation importance-ratio denominator.

motivates using the late checkpoint to define a direction of policy change and restarting the deployed policy from the batch generator. Under off-policy batch reuse, imbalanced or poorly separated positive- and negative-example gradients may also produce policy movement before it yields measurable improvement.

We therefore propose *Extreme Region Policy Distillation* (ERPD), shown in Fig. 1. Stage 1 repeatedly optimizes a fixed rollout batch under weak constraints. Stage 2 converts the resulting policy change into a fixed token-level score, restarts from the batch-generating policy, and applies a lightweight clipped update. This separation decouples sample efficiency from KL efficiency. The restarted policy can reach a better measured performance–KL point than the teacher.

The same construction extends to a weak-teacher setting in which the selected teacher remains below the base model. Along its training trajectory, performance first decreases and then recovers; we call the earlier checkpoint the *unlearned* policy and the later checkpoint the *recovered* policy. Their log-ratio still provides effective supervision (Sec. 2.3; Fig. 2).

We summarize our contributions as follows:

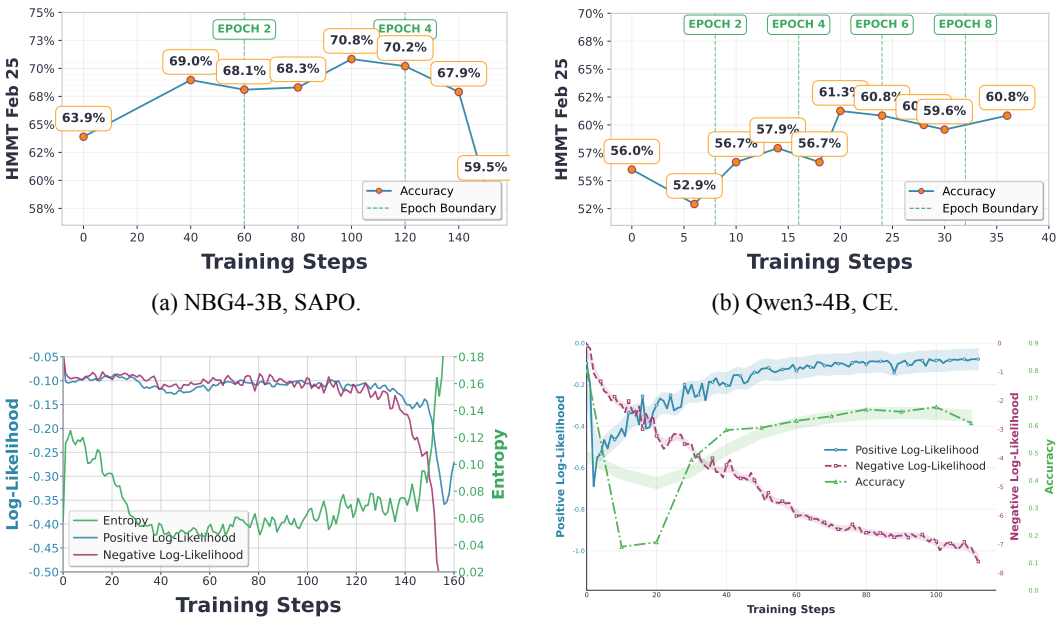
1. We introduce a two-stage training framework that first performs aggressive, weakly constrained fixed-batch policy optimization and then distills the resulting signal through lightweight clipped updates initialized from the rollout policy, improving rollout and KL efficiency.
2. For the strong-teacher strategy, we compare the sample efficiency of several policy losses, show that distillation reduces KL divergence, and analyze when the student can outperform its teacher.
3. For the weak-teacher strategy, we provide a systematic empirical characterization, demonstrate weak-to-strong distillation, and present detailed ablations.
4. On mathematical reasoning tasks, ERPD yields substantial gains for strong base models where standard on-policy optimization gives diminishing returns.

2 Decoupled Two-Stage Optimization

2.1 Stage 1: Off-Policy Updates for Sample Efficiency

Let π_{old} be the previous round's final policy (the base policy initially). It generates the rollout batch $\mathcal{D} = \{(s_t, a_t, R)\}$, where $R \in \{0, 1\}$ is the outcome reward. ERPD first reuses $\mathcal{D}$ for many optimizer steps to obtain an *extreme-region policy* π_e. Teacher optimization may use any policy-optimization objective; we evaluate GRPO (Shao et al., 2024), PPO (Schulman et al., 2017), SAPO (Gao et al., 2025), and cross-entropy (CE) (Yuan et al., 2025a). The term "extreme region" is descriptive: it means that π_e is obtained by substantially more fixed-batch optimization than standard training. Additional notation and training background are provided in Appendix A.

Figure 3 summarizes the motivating fixed-batch dynamics. A fixed batch often requires dozens of steps before validation accuracy peaks. Continuing beyond that point can increase policy change while accuracy plateaus or declines. Consequently, a selected late teacher checkpoint serves as a source of training signal for a restarted policy.



(a) NBG4-3B, SAPO.

(b) Qwen3-4B, CE.

(c) Probability and entropy drift.

(d) MSE unlearn–recover dynamics.

Figure 3: **Aggressive fixed-batch optimization dynamics.** Each run uses trajectories sampled from 1K prompts. Validation accuracy improves for multiple epochs with (a) NBG4-3B under SAPO and (b) Qwen3-4B under CE. Further optimization changes (c) token probabilities and entropy as accuracy plateaus or declines, while (d) MSE exhibits an unlearn–recover trajectory.

2.2 Stage 2: Distillation for KL Efficiency

For each sampled token, we define the fixed score

$$\hat{A}(s,a) = \log \pi_e(a \mid s) - \log \pi_{\text{old}}(a \mid s), \tag{1}$$

and whiten it over the optimization batch. A positive score means that π_e assigns the sampled token higher probability than π_{old}, whereas a negative score means the opposite. We initialize the distilled student from the same previous-round final policy, $\pi_{\theta,0} = \pi_{\text{old}}$, and optimize

$$\max_{\theta} \; \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\min\Big(r_\theta(s,a)\hat{A}(s,a), \text{clip}(r_\theta(s,a), 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}})\hat{A}(s,a) \Big) \right], \tag{2}$$

where $r_\theta(s,a) = \pi_\theta(a \mid s)/\pi_{\text{old}}(a \mid s)$. Thus, by default, π_{old} generates the rollouts and is the denominator of both the fixed score and the distillation importance ratio. As in PPO, clipping suppresses updates only when moving the ratio farther outside the clipping range would improve the surrogate; gradients that move the ratio back toward the range remain active. We use $(\varepsilon_{\text{low}}, \varepsilon_{\text{high}}) = (0.2, 0.28)$ by default.

Because $\hat{A}$ remains fixed as the student changes, distillation continues to express the policy difference discovered by fixed-batch teacher optimization.

ERPD procedure. For each training round, ERPD executes the following sequence:

1. sample a rollout batch $\mathcal{D}$ from π_{old};
2. optimize $\mathcal{D}$ repeatedly and select π_e using the validation protocol;
3. cache the token scores in Eq. 1;
4. reset the student to π_{old} and optimize Eq. 2 for a small number of steps; and
5. deploy the selected student as the next round's π_{old}.

The fixed score denominator is consequential. Replacing π_{old} in Eq. 1 with the evolving student π_θ makes the score shrink as the student approaches the teacher. Keeping the score fixed can extrapolate beyond π_e; student scores above the teacher therefore reflect additional distillation.

An idealized geometric view is to seek a policy that remains near π_{old} while retaining the improvement of the fixed-batch teacher π_e:

$$\min_{\pi} \ \mathrm{KL}(\pi \| \pi_e) \quad \text{s.t.} \quad \mathrm{KL}(\pi \| \pi_{\text{old}}) \le \delta. \tag{3}$$

Additional geometric motivation and implementation details are provided in Appendix B.

2.3 Distilling from Unlearned Extreme Region Policy

When standard teacher objectives cease to produce a meaningfully stronger teacher, we construct a contrast from an unconstrained trajectory. Starting from π_{old}, we minimize

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{(s,a,R)\sim\mathcal{D}} \left[\left(\pi_\theta(a \mid s) - R \right)^2 \right]. \tag{4}$$

Early in this trajectory, positive-example probabilities can temporarily decrease together with validation accuracy (Fig. 3d); we denote an intermediate checkpoint by π_{un}. A later recovered checkpoint is π_e. As optimization continues, positive probability mass and validation accuracy recover; the final policy reaches near-zero MSE loss and explained variance above 0.9 on the fitted targets.

Using the usual old-policy reference, $\log \pi_e - \log \pi_{\text{old}}$, may yield limited gains in this setting. The intermediate unlearned checkpoint provides a stronger contrast. The resulting weak-teacher signal is

$$\hat{A}_{\text{weak}}(s,a) = \log \pi_e(a \mid s) - \log \pi_{\text{un}}(a \mid s), \tag{5}$$

inside the same distillation objective. More generally, let π_{ref} denote the denominator substituted into the fixed score. Strategy 1 uses $\pi_{\text{ref}} = \pi_{\text{old}}$ as in Eq. 1, Strategy 2 uses $\pi_{\text{ref}} = \pi_{\text{un}}$, and Strategy 3 uses an older successful policy. Every strategy initializes the distilled student from the current π_{old} and uses it in the importance ratio.

3 Experiments

3.1 Setup

Models and data. We study Qwen3-4B-2507-Thinking (Qwen3-4B), Nanbeige4-3B-2511-Thinking (NBG4-3B) (Yang et al., 2025), Qwen3.5-9B, and Qwen3.5-27B. Each offline round samples 16 trajectories for 1,000 prompts from the POLARIS training collection (An et al., 2025). Unless noted otherwise, the learning rate is 10^{-6}. Additional batch-size, sequence-limit, decoding, and loss-hyperparameter ranges are in Appendix C.1; Appendix Table 6 summarizes the default distillation configuration.

Evaluation. We report AVG@K (Guo et al., 2025) on AIME24, AIME25, HMMT Feb/Nov 2025, HMMT Feb 2026 (Balunović et al., 2025), Beyond AIME (Seed et al., 2025), and IMO Answer Bench (Luong et al., 2025). We use $K = 16$ for Beyond AIME, $K = 8$ for APEX2025, $K = 4$ for IMO Answer Bench, and $K = 32$ otherwise. HMMT Feb 2025 is used for checkpoint and hyperparameter selection; its values are therefore validation results, while the remaining columns measure transfer under the selected configuration. All 3B/4B results use exact match. APEX2025, a difficult set selected using several advanced models, also uses exact match; the other Qwen3.5 columns use Seed-2.0-Pro as an LLM judge. Full decoding, judging, and aggregate-check details are in Appendix C.1.

We estimate token-averaged reverse KL on samples from the new policy:

$$\widehat{\mathrm{KL}}(\pi_\theta \| \pi_{\text{old}}) = \frac{1}{N} \sum_{i=1}^{N} \left[\log \pi_\theta(a_i \mid s_i) - \log \pi_{\text{old}}(a_i \mid s_i) \right]. \tag{6}$$

Because the state distribution and generated lengths can change, this measures relative policy change on the sampled trajectories.

3.2 KL Efficiency

Figure 4 compares CE teacher trajectories with their distilled students.

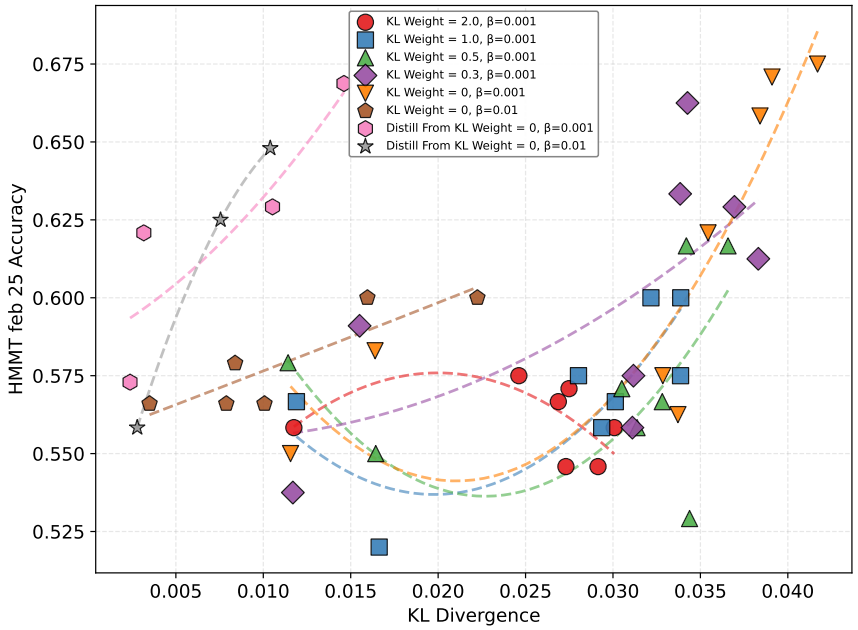


Figure 4: **Performance–KL trajectories on Qwen3-4B.** CE teacher runs use the listed KL weights and β values. At $\beta = 0.01$, the selected distilled student reaches higher HMMT Feb 25 accuracy at lower measured reverse KL than its teacher; at $\beta = 0.001$, the student largely preserves teacher accuracy while reducing KL. Lines connect checkpoints from the same run.

With $\beta = 0.01$, the selected teacher is relatively conservative, and its student reaches higher validation accuracy at lower measured KL. With $\beta = 0.001$, the teacher reaches a stronger operating point; distillation largely preserves its accuracy while reducing KL. Stronger teacher-training KL regularization lowers the teacher's attainable accuracy rather than producing a better frontier. The teacher trajectories can also accumulate KL before validation accuracy improves. One possible explanation is the gradient geometry of fixed-batch off-policy optimization: unequal positive- and negative-example contributions may let one group dominate, while poorly separated directions may prevent the aggregate update from cleanly distinguishing rewarded from unrewarded behavior. In either case, policy movement need not immediately translate into a validation gain. Across these settings, distillation improves the observed performance–KL trade-off relative to its teacher.

Additional NBG4-3B operating points across update horizons and explicit KL regularization are reported in Appendix Table 9.

3.3 Sample Efficiency

Here sample efficiency means reusing one rollout batch for multiple optimizer updates before refreshing trajectories.

Table 1 compares teacher objectives on the same fixed rollout batch. SAPO and CE produce the strongest transfer averages on both base models. PPO and GRPO provide smaller or mixed gains, while the MSE teachers score far below the base models.

Table 2 reports the corresponding distilled students. Distillation improves the SAPO and CE transfer averages on both models, while the GRPO averages decrease slightly. The MSE teachers are weaker than their base models, yet the recovered–unlearned contrast yields students above the base: 73.2 versus 71.6 for Qwen3-4B and 76.5 versus 74.6 for NBG4-3B. This is the weak-to-strong setting studied further below. Appendix Tables 13 and 12 report same-hardware SAPO timing and on-

Table 1: **Extreme-region teachers on matched fixed batches.** Every method within a model uses the same sampled batch. Entries are AVG@K accuracy (%). HMMT Feb 25 is the validation set; "Average" averages the four transfer sets. Bold marks the column best and underlining marks the second best within each model block. Appendix Table 10 gives paired teacher–student results on all benchmarks.

Base model	Method	Val.	Transfer				
		HMMT Feb 25	AIME 2024	AIME 2025	Beyond AIME	HMMT Nov 25	Average
Qwen3-4B	Base	56.0	84.9	81.1	53.8	66.6	71.6
	PPO	56.5	84.8	79.8	53.5	66.0	71.0
	GRPO	57.8	86.8	81.2	53.5	65.7	71.8
	SAPO	60.3	85.6	81.6	55.3	66.8	72.3
	CE	**67.6**	**87.3**	**83.6**	**55.7**	**70.4**	**74.3**
	MSE	27.7	57.2	36.6	19.5	40.8	38.5
NBG4-3B	Base	63.9	90.9	84.8	55.5	67.3	74.6
	PPO	65.4	91.5	86.7	55.5	70.5	76.1
	GRPO	67.3	90.4	86.4	55.3	68.9	75.3
	SAPO	70.2	**91.8**	**89.1**	59.0	71.1	77.8
	CE	**70.6**	91.0	88.3	**61.0**	**72.8**	**78.3**
	MSE	31.8	63.9	48.1	15.1	26.4	38.4

Table 2: **Effect of distillation on matched batches.** AVG@K accuracy (%) is shown as teacher $\rightarrow$ distilled student. "Transfer" averages AIME24, AIME25, Beyond AIME, and HMMT Nov 25. For MSE†, distillation uses the recovered–unlearned contrast rather than directly matching the MSE teacher. Bold marks student improvements over the teacher. Appendix Tables 10 and 18 give all benchmarks and score-sign controls.

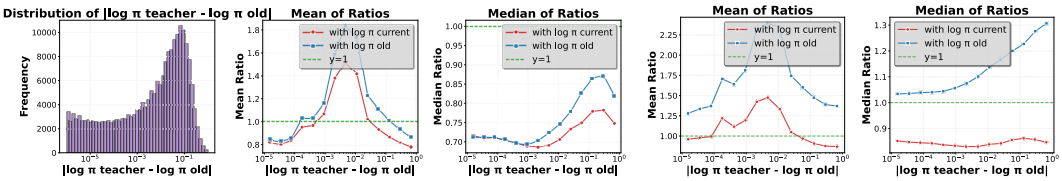
Base model	Teacher loss	AIME25	HMMT Nov 25	Transfer
Qwen3-4B	GRPO	81.2→80.0	65.7→64.2	71.8→71.3
	SAPO	81.6→**82.0**	66.8→**68.0**	72.3→**73.0**
	CE	83.6→**85.1**	70.4→69.9	74.3→**74.8**
	MSE†	36.6→**81.9**	40.8→**68.0**	38.5→**73.2**
NBG4-3B	GRPO	86.4→**87.1**	68.9→66.5	75.3→75.1
	SAPO	89.1→**89.6**	71.1→**72.0**	77.8→**78.5**
	CE	88.3→**88.6**	72.8→**73.9**	78.3→**79.2**
	MSE†	48.1→**87.4**	26.4→**68.1**	38.4→**76.5**

policy GRPO controls. The latter show no sustained gain: a 120-update run peaks at 56.9 and ends at 53.8, compared with 56.0 for the base, while the shorter trajectories are flat or decline.

Surpassing the teacher. Figure 5 compares the fixed teacher–old-policy score with an evolving teacher–student score; both use π_{old} as the importance-ratio denominator. The evolving contrast shrinks as the student approaches the teacher and reaches accuracy comparable to the teacher. The fixed contrast persists throughout optimization and reaches higher selected accuracy. This effect is not universal: it appears in several SAPO and CE configurations, whereas GRPO gives little or negative transfer gain. For the stronger $\beta = 0.001$ CE teacher in Fig. 4, the primary benefit is lower KL rather than higher accuracy. The teacher objective and its regularization therefore determine whether the student can surpass its teacher.

3.4 Iterative Training and Scaling

In the fixed-protocol first three rounds of Table 3, the deployed-policy average rises from 67.9 to 76.4; AIME25 and HMMT Nov 25 rise from 81.1 to 84.5 and 66.6 to 73.5. The no-distillation controls continue CE directly, without a restarted distillation update, from two checkpoints on the same first-round CE trajectory: path (a) at 0.041 KL and path (b) at 0.022 KL. At Batch 2, ERPD



(a) 10 distillation steps. (b) 16 distillation steps.

Figure 5: **Fixed versus evolving score references on Qwen3-4B after (a) 10 and (b) 16 distillation steps.** This comparison uses the $\beta = 0.01$ CE teacher, which scores 61.0 on HMMT Feb 25. The fixed score $\log \pi_e - \log \pi_{\text{old}}$ reaches 65.1 and 63.8 in (a) and (b), respectively, whereas the evolving score $\log \pi_e - \log \pi_\theta$ reaches 61.2 and 61.1. The plotted ratio is $|\log \pi_\theta - \log \pi_{\text{old}}| / |\log \pi_e - \log \pi_{\text{old}}|$; a value of one means that student and teacher moved equally far from the rollout policy on that token.

Table 3: **Iterative ERPD on Qwen3-4B.** Rows report deployed policies. HMMT Feb 25 is the selection set; AIME25 and HMMT Nov 25 are additional test sets. "Avg." averages all three. KL is measured from the base; corresponding teacher rows are in Appendix Table 11.

Round	AIME25	HMMT Feb 25 (val.)	HMMT Nov 25	Avg.	KL
Base	81.1	56.0	66.6	67.9	0
Batch 1	82.5	62.9	68.1	71.1	0.010
Batch 2	84.5	67.9	68.5	73.6	0.026
Batch 3	84.5	71.2	73.5	76.4	0.037
Batch 4 (Strategy 3)	87.9	71.7	78.3	79.3	0.062
Batch 5 (Strategy 2)	88.0	72.8	79.0	79.9	0.065
Batch 6 (Strategy 2)	**89.2**	**73.6**	**79.0**	**80.6**	**0.073**
No distill (a), Batch 1	83.6	67.6	70.4	73.8	0.041
No distill (a), Batch 2	84.1	67.9	70.0	74.0	0.054
No distill (b), Batch 1	83.3	61.2	67.5	70.6	0.022
No distill (b), Batch 2	79.2	61.2	65.0	68.4	0.030

reaches 73.6 at 0.026 KL, versus 74.0 at 0.054 KL and 68.4 at 0.030 KL for the two no-distillation paths, exposing an accuracy–KL trade-off.

After the standard CE contrast plateaus on HMMT Feb 25, Batches 4–6 change the score construction: Batch 4 uses Strategy 3 and Batches 5–6 use Strategy 2 (Sec. 2.3). Their deployed-policy averages are 79.3, 79.9, and 80.6; the final policy reaches 89.2 on AIME25, 73.6 on the selection set, and 79.0 on HMMT Nov 25. Appendix Table 11 gives the teacher–student comparisons.

Table 4 reports final-round transfer averages of 60.2 for Qwen3.5-9B and 71.1 for Qwen3.5-27B, gains of 7.8 and 7.1 over their respective bases. Intermediate averages are unavailable because APEX2025 was evaluated only for Base and Batch 3; the available columns show that individual benchmark peaks can occur earlier. On the exact-match APEX2025 set, scores rise from 0.52 to 9.4 at 9B and from 2.08 to 14.6 at 27B; the final 27B model also improves on all three other transfer benchmarks. Results on coding and three additional model families are reported in Appendix C; improvements are smaller for the shorter-generation models tested there.

Appendix Table 8 shows that the +H path combines its transfer gains with stable or higher entropy.

Online–offline switching. Figure 6 places fixed-batch stages in an iterative trajectory: they produce sharper selected validation gains and recover progress after online phases.

3.5 Unlearned Extreme Region Policy

Table 5 summarizes the complementary weak-teacher signals. Used alone, the teacher–old-policy score exceeds the recovered–unlearned score by 0.4, 3.2, 4.4, and 1.0 points on AIME24, AIME25,

Table 4: **Scaling of iterative weak-teacher ERPD.** Batch rows are the deployed student after each round; all benchmark entries and averages are AVG@K accuracy (%). HMMT Feb 25 is used for selection and excluded from the transfer average. Transfer averages include HMMT Nov 25, HMMT Feb 26, and exact-match APEX2025; the 27B average also includes IMO. APEX2025 is available only for Base and Batch 3, so intermediate averages are omitted.

Model	Round	HMMT Feb 25 (val.)	HMMT Nov 25	HMMT Feb 26	IMO	APEX2025	Transfer avg.
Qwen3.5-9B	Base	83.5	84.0	72.7	–	0.52	52.4
	Batch 1	85.6	84.1	78.8	–	–	–
	Batch 2	88.7	88.3	82.2	–	–	–
	Batch 3	**88.9**	**87.3**	**83.8**	–	**9.4**	**60.2**
Qwen3.5-27B	Base	92.0	89.8	84.3	79.9	2.08	64.0
	Batch 1	94.1	93.3	83.5	84.0	–	–
	Batch 2	93.0	92.2	88.0	85.2	–	–
	Batch 3	**94.1**	**94.1**	**89.2**	**86.3**	**14.6**	**71.1**

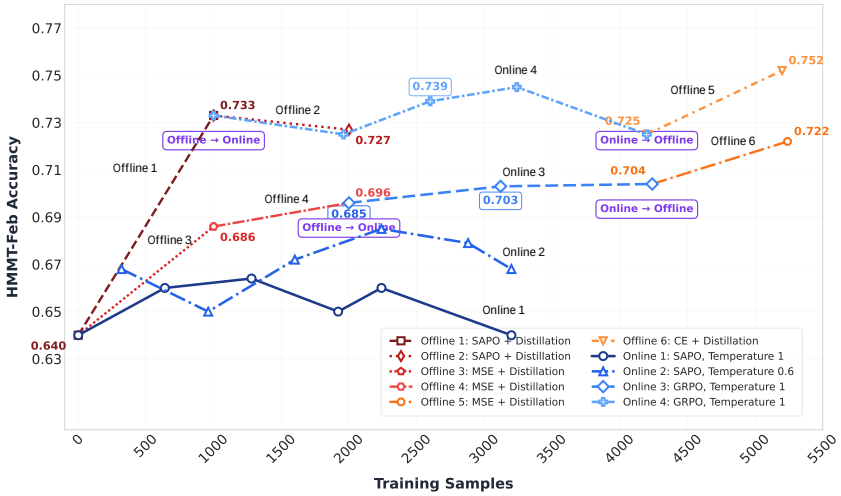


Figure 6: **Online–offline switching on NBG4-3B.** Offline stages yield sharper selected validation-accuracy gains and recover progress after online phases.

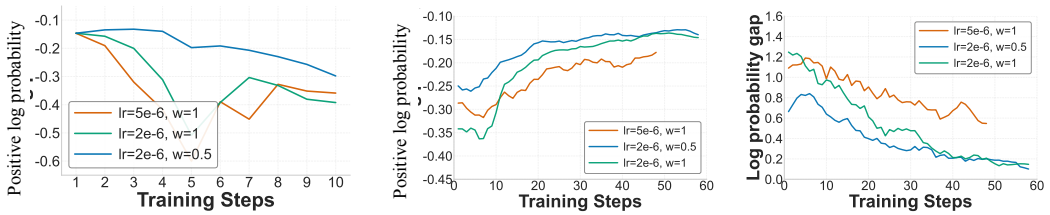
HMMT Feb 25, and Beyond AIME, respectively. Applying the teacher–old-policy and recovered–unlearned scores as a two-score ensemble gives the best result in every column: it gains $+2.2$, $+0.3$, $+1.9$, and $+0.4$ points over the strong score and raises the four-column mean from 74.1 to 75.3, versus 71.9 for the recovered–unlearned score alone.

The gains cover all three transfer benchmarks and the HMMT Feb 25 selection set, although those on AIME25 and Beyond AIME are small. For these selected configurations, this pattern supports using the recovered–unlearned score as a complementary second update rather than as a standalone signal. The selected Qwen3-4B and NBG4-3B MSE+D configurations apply $\min(0, \hat{A}_{\text{weak}})$ before full-batch whitening. Approximately 10% of the raw response-token scores are negative (Appendix Fig. 8). This transformation yields higher HMMT Feb 25 accuracy than the all-scores variant for both models; its transfer effects are mixed (Appendix Table 18).

The selected two-score ensemble with CE recovery reaches 69.0, versus 66.7 with MSE recovery, suggesting that the unlearned–recovered contrast matters more than MSE itself (Appendix Table 15). Appendix Table 14 compares checkpoint choices. The selected 50-step recovered checkpoint and 10-step unlearned checkpoint yield 64.3 on HMMT Feb 25, compared with 53.1–61.3 for the other substitutions. CE is also stronger than SFT in two recovery settings (Appendix Table 7); Fig. 7 relates the strongest setting to a wider positive–negative probability gap. During recovery, the weaker settings' gaps contract toward zero, whereas the strongest setting retains a visible separation. Together, these controls show that recovery quality depends on both the unlearning strength and recovery objective.

Table 5: **Complementary fixed-score constructions.** Post-distillation Qwen3-4B AVG@K accuracy (%) for the recovered–unlearned, teacher–old-policy, and two-score ensemble. Bold marks the column best. Appendix Table 15 varies recovery loss and weak/strong update strengths.

Score construction	AIME 2024	AIME 2025	HMMT Feb 25	Beyond AIME
Recovered–unlearned	87.5	81.9	62.7	55.3
Teacher–old-policy	87.9	85.1	67.1	56.3
Two-score ensemble	**90.1**	**85.4**	**69.0**	**56.7**



(a) MSE unlearning. (b) SFT from the 10-step unlearned policy.

Figure 7: **Recovery after MSE unlearning.** Panel (a) compares the listed unlearning settings; panel (b) shows SFT recovery from the 10-step checkpoint. The strongest unlearning preserves the widest positive–negative probability gap.

Cross-model and coding results are in Appendix Tables 16 and 17; the two shorter-generation models show smaller gains.

4 Related Work

Token-level policy signals. Sparse outcome rewards can induce high-variance token-level optimization signals. A common alternative uses a policy–reference log-ratio as token-level reward information. Under a maximum-entropy RL view, a DPO-optimized policy implicitly encodes Q-value information in this ratio (Rafailov et al., 2023; 2024). Subsequent methods use such signals in preference objectives and PPO/RLOO updates (Zhu et al., 2025; Yuan et al., 2025a; Cui et al., 2025; Zhong et al., 2025). In particular, Yuan et al. (2025a) parameterize a reward model with a policy–reference log-ratio and fit it using cross-entropy, corresponding to our CE baseline. ERPD instead studies how such signals interact with policy learning on the same sampled batch, including whether the restarted student can surpass its teacher. Its recovered–unlearned contrast further connects critic-style return fitting (Yue et al., 2025; Yuan et al., 2025b) with policy log-ratios.

Conservative and aggressive policy updates. PPO and GRPO use clipped importance-ratio objectives (Schulman et al., 2017; Shao et al., 2024), which suppress updates that move beyond the clipping margin. SAPO and VESPO replace hard clipping with soft weighting (Gao et al., 2025; Shen et al., 2026), while CISPO and DISPO permit more aggressive unclipped updates (Chen et al., 2025; Karaman et al., 2026). In our fixed-batch comparisons, SAPO is more rollout-efficient than GRPO. ERPD separates aggressive signal construction from a shorter conservative deployment update, decoupling the strength of teacher optimization from final-policy stability. Restarting at the rollout policy may also discard drift accumulated during teacher optimization.

5 Conclusion

ERPD separates aggressive fixed-batch signal construction from lightweight clipped updates initialized at the rollout policy. It improves measured performance–KL trade-offs at 3B/4B, supports iterative gains at 9B/27B, and extracts useful supervision from weak teachers. Same-hardware controls further show that batch reuse replaces repeated rollout generation with cached-score optimizer updates.

References

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmark*, 2025.

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

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *CoRR*, 2025.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pp. 1407–1416. PMLR, 2018.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

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

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, Ning Ding, and Zhiyuan Liu. Justrl: Scaling a 1.5b llm with a simple rl recipe, 2025a. URL https://arxiv.org/abs/2512.16649.

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, 2025b. URL https://arxiv.org/abs/2505.22312.

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.

Batuhan K Karaman, Aditya Rawal, Mohammad Ghavamzadeh, Suhaila Shakiah, Arijit Biswas, and Ruida Zhou. Dispo: Enhancing training efficiency and stability in reinforcement learning for large language model mathematical reasoning. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026.

Nicolas Le Roux, Marc G Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alex Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Toth, and Sam Work. Tapered off-policy reinforce: Stable and efficient reinforcement learning for llms. *arXiv preprint arXiv:2503.14286*, 2025.

Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Clara Huiyi Hu, Henryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu H. Trinh, Quoc V. Le, and Junehyuk Jung. Towards robust mathematical reasoning, 2025. URL https://arxiv.org/abs/2511.01846.

Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

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

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

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to $q*$: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

ByteDance Seed, Jiaze Chen, Tiantian Fan, Xin Liu, Lingjun Liu, Zhiqi Lin, Mingxuan Wang, Chengyi Wang, Xiangpeng Wei, Wenyuan Xu, et al. Seed1. 5-thinking: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*, 2025.

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.

Guobin Shen, Chenxiao Zhao, Xiang Cheng, Lei Huang, and Xing Yu. Vespo: Variational sequence-level soft policy optimization for stable off-policy llm training, 2026. URL https://arxiv.org/abs/2602.10693.

ModelScope Team. EvalScope: Evaluation framework for large models, 2024. URL https://github.com/modelscope/evalscope.

Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, et al. Bapo: Stabilizing off-policy reinforcement learning for llms via balanced policy optimization with adaptive clipping. *arXiv preprint arXiv:2510.18927*, 2025.

Chen Yang, Guangyue Peng, Jiaying Zhu, Ran Le, Ruixiang Feng, Tao Zhang, Wei Ruan, Xiaoqi Liu, Xiaoxue Cheng, Xiyun Xu, et al. Nanbeige4-3b technical report: Exploring the frontier of small language models. *arXiv preprint arXiv:2512.06266*, 2025.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *Proceedings of Machine Learning Research*, 267:73511–73525, 2025a.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025b.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

Han Zhong, Zikang Shan, Guhao Feng, Wei Xiong, Xinle Cheng, Li Zhao, Di He, Jiang Bian, and Liwei Wang. Dpo meets ppo: Reinforced token optimization for rlhf. In *International Conference on Machine Learning*, pp. 78498–78521. PMLR, 2025.

Mingkang Zhu, Xi Chen, Zhongdao Wang, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Tgdpo: Harnessing token-level reward guidance for enhancing direct preference optimization. In *Forty-second International Conference on Machine Learning*, 2025.

AI use statement

In this work, we used generative AI tools to assist with translation and the presentation and interpretation of experimental results. We additionally used generative AI tools for English language checking and readability editing and writing plotting code used to create scientific figures.

We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for or assist with mathematical proofs, implement the proposed methods, develop training or evaluation code, clean or reformat datasets, or conduct qualitative or thematic data analysis. Generative AI tools did not execute experiments or determine training configurations. Survey or interview design and research-material transcription are not applicable to this work.

The authors reviewed all AI-assisted text against the intended meaning and verified scientific claims against the experimental records and cited sources. AI-assisted plotting code was inspected, and the resulting figures were checked against the underlying experimental data. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

A Background

A.1 Notations

Method notation follows Sec. 2: π_{old} is the previous-round final policy, current rollout behavior policy, and initial student for distillation; it is also the denominator of the default fixed token score. The selected teacher is π_e, π_{ref} denotes a substituted score denominator in Strategies 2 and 3, π_{un} is an intermediate unlearned checkpoint, and π_θ is the evolving distilled student. For the generic trust-region background below, θ_{old} and θ parameterize the old and candidate policies. Define the probability ratio $r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$. Let $A^{\pi_{\theta_{\text{old}}}}(s,a)$ be the advantage estimated under the old policy and $\rho_{\pi_{\theta_{\text{old}}}}$ its state-visitation distribution.

A.2 Token-Level Training Signals

Dense token-level rewards provide finer supervision than sparse sequence-level outcomes and can improve rollout efficiency (Ouyang et al., 2022; Guo et al., 2025; Zhong et al., 2025). Such rewards can also be constructed from model probabilities (Rafailov et al., 2024; Yuan et al., 2025a). Direct Preference Optimization (DPO), for example, transforms pairwise preference data into log-ratio rewards relative to a reference policy (Rafailov et al., 2023). Under the maximum-entropy RL interpretation, a DPO policy induces the token-level reward (Rafailov et al., 2024)

$$R(s,a) = \beta \log \frac{\pi_\theta(a \mid s)}{\pi_{\text{ref}}(a \mid s)}, \tag{7}$$

where β controls reference-policy regularization.

Yuan et al. (2025a) directly fit this reward with cross-entropy. Given a sequence $y = (a_1, \ldots, a_T)$ and binary label $l \in \{0,1\}$, their loss is

$$L_{\text{CE}} = -l \cdot \log \sigma\big(R(y)\big) - (1-l) \cdot \log \big[1 - \sigma\big(R(y)\big)\big], \tag{8}$$

where the sequence-level reward decomposes into the sum of token-level rewards:

$$R(y) = \sum_{t=1}^{T} R(s_t, a_t) = \beta \sum_{t=1}^{T} \log \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\text{ref}}(a_t \mid s_t)}. \tag{9}$$

This formulation raises an underexplored question: if the teacher and student share their initialization and training data, can the fixed token-level reward provide a useful alternative optimization path?

A.3 Trust Region Methods

On-policy methods periodically refresh trajectories from the current behavior policy. Additional update epochs improve rollout utilization as distribution mismatch grows. Large policy changes can destabilize importance weights and increase gradient variance (Espeholt et al., 2018; Le Roux et al., 2025). Trust-region methods control this change, typically through a KL constraint. TRPO formalizes this approach as

$$\begin{aligned} \theta_{\text{new}} &= \arg\max_\theta \; \mathbb{E}_{s\sim\rho_{\pi_{\theta_{\text{old}}}}, a\sim\pi_{\theta_{\text{old}}}} \Big[r(\theta)\, A^{\pi_{\theta_{\text{old}}}}(s,a) \Big] \\ \text{s.t.} \quad & \mathbb{E}_{s\sim\rho_{\pi_{\theta_{\text{old}}}}} \Big[D_{\text{KL}}\big(\pi_{\theta_{\text{old}}}(\cdot|s) \,\big\|\, \pi_\theta(\cdot|s)\big) \Big] \le \delta, \end{aligned} \tag{10}$$

where $\delta > 0$ denotes the trust-region radius, which limits the magnitude of policy updates.

PPO replaces the constrained optimization with a clipped probability-ratio surrogate (Schulman et al., 2017); GRPO uses the same clipping principle for grouped outcomes (Shao et al., 2024).

The PPO clipped surrogate objective is

$$\theta_{\text{new}} = \arg\max_\theta \; \mathbb{E}_{\substack{s\sim\rho_{\pi_{\theta_{\text{old}}}} \\ a\sim\pi_{\theta_{\text{old}}}}} \Big[\min\big(r(\theta) A^{\pi_{\theta_{\text{old}}}}(s,a),\; \text{clip}\big(r(\theta), 1-\varepsilon, 1+\varepsilon\big) A^{\pi_{\theta_{\text{old}}}}(s,a) \big) \Big], \tag{11}$$

where ε controls the clipping range. Clipping suppresses the gradient only when moving the ratio farther outside this range would improve the surrogate, with the relevant side determined by the sign of the advantage.

A.4 Fixed-Batch Motivation

Current LLM reasoning pipelines typically refresh rollouts after each update round or perform only a small number of minibatch updates on trajectories from the current behavior policy (He et al., 2025b;a; Gao et al., 2025). ERPD instead asks how much optimization signal remains in one rollout batch after this usual update budget.

For the preliminary experiments in Fig. 3, we sample one fixed batch from π_{old} and continue optimizing it for many gradient steps. The selected validation score does not peak within the first four updates in the reported SAPO and CE runs. It instead improves over dozens of updates, showing that the batch still contains usable signal beyond a typical short on-policy update phase.

Longer optimization also changes the policy after the selected accuracy has plateaued. Token log-probabilities, entropy, and divergence from π_{old} move together; at later checkpoints, positive- and negative-example gradients become imbalanced and token probabilities can change sharply. These observations motivate separating the checkpoint used to construct a signal from the policy ultimately deployed. Fixed-batch teacher optimization extracts the policy difference from the data, while distillation restarts from the batch generator and controls how strongly that difference is applied.

B Additional Method Details

This appendix supplements Sec. 2 with the geometric motivation and implementation details behind the two-stage update.

B.1 Geometric and Trust-Region Motivation

The initial geometric motivation separates task-improving policy changes from changes unrelated to improvement. If a policy π_- containing only the latter changes were available, one could consider

$$\min_{\pi} \ \mathrm{KL}(\pi \| \pi_e) \quad \text{s.t.} \quad \mathrm{KL}(\pi \| \pi_-) \geq \epsilon. \tag{12}$$

The policy π_- is not observed, so this formulation only motivates retaining the teacher's useful change while avoiding its unrelated drift.

The operational idealization in Eq. 3 replaces this unavailable policy with the rollout behavior policy π_{old}. ERPD does not directly solve that constrained problem: the fixed score omits the evolving $-\log \pi_\theta$ term of reverse-KL minimization, and PPO clipping does not enforce an explicit KL radius. Restarting from π_{old}, limiting the distillation horizon, and optionally adding KL regularization provide practical controls on the final policy change.

The TRPO analysis provides related intuition by bounding policy-optimization error with policy divergence (Schulman et al., 2015):

$$|\mathrm{Err}(\pi)| = |L_{\pi_{\text{old}}}(\pi) - \eta(\pi)| \leq C\, D_{\mathrm{KL}}(\pi_{\text{old}} \| \pi), \tag{13}$$

where $L_{\pi_{\text{old}}}(\pi)$ is the old-policy surrogate objective.

B.2 Implementation Details

Distillation treats the score in Eq. 1 as a stop-gradient target. An evolving $\log \pi_e - \log \pi_\theta$ score shrinks as the student approaches the teacher, whereas the fixed score continues to express the teacher–reference policy difference. The two choices therefore induce different optimization paths. The evolving score increasingly resembles direct teacher matching as $\pi_\theta \to \pi_e$. The fixed score preserves the original teacher–reference contrast throughout training and can continue updating beyond the teacher checkpoint, as tested in Fig. 5.

We apply any score mask before whitening over the full rollout batch, then use the clipped objective in Eq. 2. The token-score reference defines the supervision, while π_{old} remains the importance-ratio denominator.

B.3 Weak-Teacher Signal Construction

The MSE trajectory in Eq. 4 directly fits the sampled terminal outcome through the policy head. It is analogous to fitting a Monte Carlo return with a critic, but parameterizing the fit through action probabilities induces saturation toward zero or one. In the reported trajectories, the MSE loss approaches zero and explained variance exceeds 0.9 on the fitted targets.

Early updates are dominated by negative-example gradients, while positive-example probabilities and validation accuracy can temporarily fall. We call this the unlearning phase and select π_{un} from an intermediate checkpoint, typically between steps 10 and 30. Continued optimization increases positive probability mass and validation accuracy, yielding the recovered checkpoint π_e. The resulting teacher can remain below the original base model even though the contrast $\log \pi_e - \log \pi_{\text{un}}$ provides useful distillation supervision.

Using $\log \pi_e - \log \pi_{\text{old}}$ does not improve the selected weak-teacher runs. Replacing the score reference with π_{un} isolates the change along the recovery segment of the same trajectory. Appendix Table 14 reports the sensitivity to both numerator and reference checkpoints, and Table 7 compares CE and positive-example SFT recovery from matched unlearned starts.

C Additional Experiments and Ablations

C.1 Settings

Compared Methods. We write "X+Distillation" for the two-stage pipeline, where "X" is the teacher-training loss and "+Distillation" denotes the restarted distillation update.

Teacher-objective comparisons include GRPO (Mroueh, 2025), PPO (Schulman et al., 2017), SAPO (Gao et al., 2025), CE (Yuan et al., 2025a; Cui et al., 2025), and MSE. CE fits the policy log-ratio with a cross-entropy objective; CE and SAPO both use smooth sigmoid weighting. Section C.4 reports the detailed MSE experiments.

Section C.3 compares fully on-policy baselines with iterative ERPD.

Models and benchmarks. The models, mathematical benchmarks, AVG@K values, and reverse-KL estimator are defined in Sec. 3. Unless otherwise specified, maximum sequence lengths are 81,920 for Qwen3-4B, 65,536 for NBG4-3B, and 192,000 for Qwen3.5. Coding experiments use LiveCodeBench (Jain et al., 2025).

Training Settings. The default sampling temperature is 0.6, or 1.0 for Qwen3.5, and the default response limit is 32,768 tokens unless extended for a model's generation capacity. Each offline round samples 16 trajectories for each of 1,000 POLARIS prompts and then holds this dataset fixed. This is the same collection scale as one batch of the corresponding online pipeline; the experimental difference is that fixed-batch teacher optimization performs many more updates before refreshing the rollouts. Unless otherwise specified, the learning rate is 10^{-6}.

Default distillation configuration. By default, the selected final policy from the previous round serves as the current behavior policy, token-score reference, and initial distilled student; the first round uses the base policy. The teacher is obtained by optimizing this checkpoint on the fixed rollout batch. In iterative training, the selected student from one round therefore becomes $\pi_{\text{old}} = \pi_{\text{ref}}$ in the next round, rather than restarting from the original pretrained checkpoint.

The default token score is $\hat{A} = \log \pi_e - \log \pi_{\text{old}}$ and is treated as a stop-gradient target. Strategies 2 and 3 replace only this score denominator. Log probabilities may be cached or recomputed before distillation; this choice does not change the objective. We first exclude prompt, padding, EOS, and other response-masked tokens. In the default configuration, all remaining token scores are whitened over the full rollout batch without an additional score transformation. The headline Qwen3-4B and NBG4-3B MSE+D rows are selected exceptions: positive scores are set to zero via $\min(0, \hat{A})$ before whitening, as reported in the outlier diagnostic below. Other MSE experiments, including Qwen3.5, use all response-token scores unless an additional transformation is explicitly stated.

The importance ratio always uses the rollout behavior policy in its denominator, even when the token-score reference is an unlearned or older checkpoint. We use asymmetric PPO clipping with

lower and upper margins 0.2 and 0.28, respectively. The default objective has no additional KL penalty or entropy bonus, uses token-level reduction, and applies no sequence-length normalization. Optimization uses Adam with learning rate 10^{-6}, no warmup, and gradient-norm clipping at 1.0; Adam moments, numerical epsilon, weight decay, microbatching, accumulation, precision, packing, and shuffling follow the DouVERL framework defaults. The global optimization batch is 32 or 64 depending on the experiment. Iterative experiments use 10 distillation optimizer steps. We evaluate every 10 steps on HMMT Feb 25 and select the first checkpoint in a tie.

Baseline SAPO and GRPO experiments use minibatches of 32, 64, or 256 and continue for multiple epochs. Distillation runs typically use at most one epoch; minibatches of 32 and 64 correspond to 32 and 16 optimizer steps in the reported configurations. HMMT Feb 25 is the validation set for hyperparameter and checkpoint selection and is the default accuracy metric in this appendix.

Online runs use fully on-policy training with sampling temperature 0.6 or 1.0; other hyperparameters follow the corresponding offline setting.

GRPO uses clipping margins 0.2 and 0.28. SAPO uses negative-sample temperature 1.05. CE tests $\beta \in [0.001, 0.05]$ and sum versus mean aggregation of token log-ratios. Following prior PPO work (Yuan et al., 2025b; Yue et al., 2025), value-network training uses $\lambda = 1$ and on-policy training uses $\lambda = (0.05 \times \text{length})^{-1}$.

During evaluation, the temperature is set to 0.6 (or 1.0 for Qwen3.5 models) and TopP to 0.95, with all other sampling parameters following the default settings of each model. We use the tools provided by POLARIS (An et al., 2025) across all benchmarks except HMMT Feb 26 and IMO Answer Bench, for which we use the MathArena tools (Balunović et al., 2025). Scoring protocols and the Seed-2.0-Pro aggregate checks are reported in Sec. 3.

Unless a column is explicitly labeled "transfer average," appendix averages include HMMT Feb 25 and are therefore validation-inclusive summaries.

C.2 Matched-Batch Results

Table 10 expands Tables 1 and 2 with all five mathematical evaluation sets, reporting extreme-region teachers and restarted distilled students trained from the same sampled batches.

Choice of teacher objective. SAPO and CE produce the strongest teacher transfer averages on both primary models. Their smooth weighting allows more aggressive fixed-batch updates than hard clipping in the tested configurations. GRPO is mixed: its teachers improve some validation or transfer columns, but the restarted students do not improve the transfer average. These results motivate SAPO or CE as the default strong-teacher objective in the reported runs rather than claiming that every teacher loss benefits from distillation.

Students beyond their teachers. The restarted student exceeds the teacher in several SAPO and CE configurations. This is possible because the fixed-score objective is not behavioral cloning: the score remains constant as the student changes and can continue to optimize beyond π_e. Figure 5 isolates this choice. Replacing the fixed reference with the evolving student causes the score to shrink and produces results close to the teacher, whereas the fixed reference reaches the higher selected score.

Weak teachers as score sources. The MSE teachers remain far below their corresponding base models, but the recovered–unlearned checkpoint differences yield students above those bases in the selected Qwen3-4B and NBG4-3B configurations. The useful object is therefore the checkpoint contrast, rather than the teacher's standalone behavior. The selected MSE+D rows in Table 10 apply $\min(0, \hat{A})$ before full-batch whitening; Table 18 reports the all-score and score-sign controls. Table 14 shows that this result depends more strongly on the unlearned reference checkpoint than on the later recovered checkpoint.

C.3 Iterative Training and Scaling

We next compare the offline results with online and iterative training.

Table 6: **Default distillation configuration.** Experiments that depart from these defaults are identified in their corresponding tables.

Component	Default
Initialization	Reset student to the current round's behavior policy
Token score	Stop-gradient $\log \pi_e - \log \pi_{\text{old}}$
Token processing	Response tokens; MSE+D optionally uses $\min(0, \hat{A})$
Normalization	Apply mask, then whiten over the full rollout batch
Importance ratio	$\pi_\theta / \pi_{\text{old}}$
PPO clipping	Lower margin 0.20; upper margin 0.28
Regularization	No extra KL or entropy bonus; token-level reduction
Optimizer	Adam, LR 10^{-6}, no warmup, grad-norm clip 1.0
Minibatch	32 or 64 examples (32 or 16 optimizer steps per epoch)
Iterative budget	10 distillation updates
Selection	Evaluate every 10 updates on HMMT Feb 25; first wins ties

Table 7: **Recovery from differently unlearned Qwen3-4B checkpoints.** HMMT Feb 25 is the selection set; HMMT Nov 25 is transfer. The unlearning learning rate and negative-example weight w jointly control the observed degradation.

Unlearn setting	Recovery	Feb 25 (val.)	Nov 25
2×10^{-6}, $w = 0.5$	SFT	57.6	66.4
	CE	**62.5**	**71.0**
2×10^{-6}, $w = 1$	SFT	53.5	64.9
	CE	**64.1**	**70.4**
5×10^{-6}, $w = 1$	SFT	**63.5**	67.9
	CE	62.6	67.9

Iterative distillation with a shorter update budget. The iterative path uses 10 distillation steps, compared with 16 or 32 in Table 10, allowing efficient iterative improvement.

Table 11 expands Table 3. Its "Average" includes HMMT Feb 25 and is therefore validation-inclusive. With 10 distillation steps, this average rises from 67.9 for the base to 76.4 for the Batch 3 student. The corresponding student KL remains below the CE teacher KL in each of the first three rounds. For the no-distillation controls, paths (a) and (b) select checkpoints with KL 0.041 and 0.022, respectively, from the same first-round CE trajectory and continue CE training directly from those checkpoints. These continuations reach 74.0 in path (a) and decline to 68.4 in path (b).

Adaptive switch to Strategies 2 and 3. At Batch 4, the CE teacher becomes the score numerator and the Batch 1 student becomes the score reference: $\log \pi_{\text{Batch 4 CE}} - \log \pi_{\text{Batch 1 student}}$. The current Batch 3 student remains the distillation initialization and importance-ratio denominator. This Strategy 3 update coincides with higher KL and longer generations in this run.

For Batches 5 and 6, Strategy 2 uses the MSE signal $\log \frac{\pi_{\theta_e}(a|s)}{\pi_{\theta_{\text{un}}}(a|s)}$. The adaptive schedule improves after the CE teacher plateaus, showing that the weak-teacher strategy can sustain iterative progress.

Online–offline switching. We additionally record one adaptive trajectory that alternates online and offline stages: In Fig. 6, Online-1 and Online-2 increase slowly relative to Offline-1 on NBG4-3B. The Qwen3-4B GRPO trajectories in Table 12 show no sustained gain, including a 120-update run with batch size 128 and minibatch size 32. Recollecting 1K samples for Offline-2 also gives no additional gain, while Offline-3 and Offline-4 increase the selected score by less than Offline-1.

After an intermediate online phase, the subsequent Offline-5 and Offline-6 stages again increase the selected score. Overall, these results suggest that switching between offline and online training can support sustained performance improvement.

The long-horizon run used eight NVIDIA H20 GPUs for more than seven days, with most wall-clock time spent on rollout generation. It peaks at 56.9 and ends at 53.8, providing no sustained

Table 8: **Entropy control on iterative Qwen3-4B.** Both paths use CE teachers; "+H" adds a distillation entropy bonus of 0.5. AIME25 and HMMT Nov 25 are transfer evaluations.

Distillation	Round	AIME25	HMMT Nov 25	Entropy	KL
–	Base	81.1	66.6	0.20	0
Default	1	82.5	68.1	0.19	0.010
	2	84.5	68.5	0.16	0.026
	3	84.5	73.5	0.14	0.037
+H	1	83.7	71.0	0.19	0.014
	2	**88.3**	74.5	0.21	0.023
	3	85.6	**76.0**	0.25	0.032

Table 9: **Distillation performance–KL operating points on NBG4-3B.** Accuracy is HMMT Feb 25. Distillation steps and explicit KL penalties provide different operating points. Shorter distillation runs can match the teacher at lower KL, while longer runs exchange more divergence for higher accuracy; explicit KL regularization provides intermediate operating points.

Method	Steps	Accuracy	KL
Base	–	63.9	0
SAPO teacher	60	68.1	0.016
SAPO teacher	120	70.2	0.023
SAPO+D	16	68.3	**0.009**
SAPO+D	20	70.2	0.017
SAPO+D	32	**73.3**	0.026
SAPO+D, KL weight 0.2	32	70.2	0.014
SAPO+D, KL weight 0.05	32	69.6	0.020
SAPO+D, KL weight 0.001	32	71.7	0.018

improvement over the 56.0 base. By comparison, the CE teacher runs in Fig. 4 use batch size 32 and reach stronger selected operating points after 40 updates ($\beta = 0.01$) and 100 updates ($\beta = 0.001$).

Optimization-time accounting. All runs in Table 13 are trained on eight NVIDIA A100 GPUs. ERPD reuses the rollout batch and cached token scores, so its additional optimizer steps require neither new sampling nor teacher inference. The offline path spends most of its time in teacher optimization because the same batch is updated for many epochs. Distillation is short, and its teacher log-probabilities can be precomputed, so teacher and student need not be loaded together during the update.

On NBG4-3B, ERPD reaches 73.3 HMMT Feb 25 accuracy in approximately 21 hours, whereas the selected on-policy SAPO checkpoints reach at most 68.5 over the same elapsed time and end at 66.8 after approximately 31 hours. The 21-hour ERPD total consists of approximately 17 hours for rollout collection and teacher training, followed by 4 hours for the selected distillation run. In the measured setup, offline training also avoids keeping a generative rollout worker active during optimizer-only steps. These implementation properties reduce rollout and memory overhead even though the offline path performs more optimizer updates.

C.4 Unlearned Extreme Region Policy

The selected MSE checkpoint contrast yields an unexpectedly effective weak-teacher result. We next report signal-construction, ensemble, loss, and cross-model ablations.

Checkpoint comparison for signal construction. Table 14 varies selected numerator and denominator checkpoints to characterize their individual contributions and sensitivity.

Table 14 shows greater sensitivity to the unlearned checkpoint than to the recovered checkpoint. Later recovered checkpoints generally work well, while poor unlearned-checkpoint choices degrade the student. The selected 50-step recovered checkpoint and 10-step unlearned checkpoint give the

Table 10: Extreme-region teachers and restarted distilled students on mathematical evaluation sets. Rows labeled "teacher" are fixed-batch teachers; "+D student" rows are their restarted students. AVG@K metrics are reported. Here "Avg." includes all five benchmark columns, including the HMMT Feb 25 validation set; it is therefore distinct from the transfer-only average in Table 1.

Base Model	Method	AIME 2024	AIME 2025	HMMT Feb 25	Beyond AIME	HMMT Nov 25	Avg.
Qwen3-4B-Thinking-2507	Base	84.9	81.1	56.0	53.8	66.6	68.5
	PPO teacher	84.8	79.8	56.5	53.5	66.0	68.1
	GRPO teacher	86.8	81.2	57.8	53.5	65.7	69.0
	GRPO+D student	86.8	80.0	61.0	54.0	64.2	69.2
	SAPO teacher	85.6	81.6	60.3	55.3	66.8	69.9
	SAPO+D student	86.3	82.0	61.1	55.5	68.0	70.6
	CE teacher	87.3	83.6	67.6	55.7	70.4	72.9
	CE+D student	87.9	85.1	67.1	56.3	69.9	**73.3**
	MSE teacher	57.2	36.6	27.7	19.5	40.8	36.4
	MSE+D student	87.5	81.9	62.7	55.3	68.0	71.1
Nanbeige4-3B-2511-Thinking	Base	90.9	84.8	63.9	55.5	67.3	72.5
	PPO teacher	91.5	86.7	65.4	55.5	70.5	73.9
	GRPO teacher	90.4	86.4	67.3	55.3	68.9	73.6
	GRPO+D student	91.4	87.1	67.1	55.5	66.5	73.5
	SAPO teacher	91.8	89.1	70.2	59.0	71.1	76.2
	SAPO+D student	91.2	89.6	73.3	61.3	72.0	77.5
	CE teacher	91.0	88.3	70.6	61.0	72.8	76.7
	CE+D student	90.9	88.6	71.8	63.2	73.9	**77.7**
	MSE teacher	63.9	48.1	31.8	15.1	26.4	37.1
	MSE+D student	91.3	87.4	68.6	59.1	68.1	74.9

highest HMMT Feb 25 accuracy in this comparison, 64.3. During development, later recovered checkpoints were used as the numerator unless validation indicated otherwise; the larger variation comes from the reference selected along the early unlearning trajectory. Numerator and reference substitutions in the table separate these two roles.

Replacing the teacher with the current starting model ("Curr." in the numerator) or the unlearned policy with the current model ("Curr." in the denominator) yields 59.1 and 61.3, respectively. Omitting either term also degrades performance, with the unlearn-only variant reaching 56.3. Both terms therefore matter among the tested settings, and the result is sensitive to checkpoint choice.

Complementary score constructions. Table 15 combines the strong teacher–old-policy score from Strategy 1 with the recovered–unlearned score from Strategy 2. We apply several distillation steps with the strong score, followed by the weak score, and select 16 or 32 total steps using validation performance. In the selected configurations, the two-score ensemble produces higher student accuracy than either score alone, indicating complementary information in the two checkpoint contrasts.

In the reward signal constructed from the MSE trajectory, the unlearned policy π_{un} serves as the token-score reference. The current rollout policy π_{old} remains the denominator of the distillation importance ratio. CE recovery from the same unlearned start raises the selected ensemble result from 66.7 to 69.0 on HMMT Feb 25 in the compared configurations. The selected row uses $\beta_u = 10^{-3}$ for the recovered–unlearned path and $\beta_o = 5{\times}10^{-2}$ for the teacher–old-policy path, indicating that the two score constructions benefit from different update strengths.

SFT recovery control. Table 7 asks whether positive-example SFT from the same unlearned checkpoint is sufficient. CE recovery is stronger in the first two settings, while SFT becomes competitive under the strongest unlearning configuration. Figure 7 traces the corresponding probability dynamics: stronger unlearning preserves a wider positive–negative probability gap during recovery. Together, these controls show that recovery quality depends on both the unlearning strength and the recovery objective.

Cross-model MSE results. Table 16 reports generation lengths for the two main models and three additional model comparisons. The gain is larger for Llama-3.1-Nemotron-Nano-4B than for Ace-

Table 11: Iterative Qwen3-4B results with and without distillation. In Batches 4–6, distillation uses score-reference Strategies 2 and 3 from Fig. 2. The comparison teacher rows remain CE-trained teachers. Strategy labels apply to the distillation score reference. KL divergence is measured from the original base.

Batch	Role	Init. Model	AIME 2025	HMMT Feb 25	HMMT Nov 25	Average	KL Div.
Base	-	-	81.1	56.0	66.6	67.9	0
With Distillation							
Batch 1	Teacher	Base	83.6	67.6	70.4	73.8	0.041
	Student	Base	82.5	62.9	68.1	71.1	0.010
Batch 2	Teacher	Batch 1 Student	85.8	68.7	72.5	75.6	0.044
	Student	Batch 1 Student	84.5	67.9	68.5	73.6	0.026
Batch 3	Teacher	Batch 2 Student	86.3	70.3	75.1	77.2	0.042
	Student	Batch 2 Student	84.5	71.2	73.5	76.4	0.037
Using Strategy 3, Selecting Batch 1 Student							
Batch 4	Teacher	Batch 3 Student	86.1	71.4	75.2	77.5	0.043
	Student	Batch 3 Student	87.9	71.7	78.3	79.3	0.062
Using Strategy 2: MSE							
Batch 5	Teacher	Batch 4 Student	89.3	72.0	77.0	79.4	0.066
	Student	Batch 4 Student	88.0	72.8	79.0	79.9	0.065
Using Strategy 2: MSE							
Batch 6	Teacher	Batch 5 Student	90.4	72.2	77.9	80.2	0.077
	Student	Batch 5 Student	89.2	73.6	79.0	80.6	0.073
Without Distillation							
Batch 1 (a)		Base	83.6	67.6	70.4	73.8	0.041
Batch 2 (a)		Batch 1 (a)	84.1	67.9	70.0	74.0	0.054
Batch 1 (b)		Base	83.3	61.2	67.5	70.6	0.022
Batch 2 (b)		Batch 1 (b)	79.2	61.2	65.0	68.4	0.030

Table 12: **On-policy GRPO trajectories on Qwen3-4B.** The top panels start from the base model and the Batch 4 student with batch size 32 and are fully on-policy. The bottom panel is a long-horizon base-model run with batch size 128 and minibatch size 32.

Init. model	Step 0	Step 25	Step 50
Base model	56.0	57.2	53.2

Init. model	Step 0	Step 40	Step 60
Batch 4 student	71.6	70.4	71.4

Init. model	Step 20	Step 40	Step 60	Step 80	Step 100	Step 120
Base model	55.2	55.8	55.0	55.0	56.9	53.8

7B and JustRL-1.5B, suggesting that longer generations may provide more token-level signals for distillation. The comparison also changes architecture, model family, and baseline strength, so generation length is an observed correlate rather than the only varying factor.

For code, we sample from AReaL-boba-2 and evaluate on LiveCodeBench with EvalScope Team (2024). CE reaches 50.7, below the 56.1 base. Strategy 2 reaches 40.2, 63.7, and 62.4 at unlearned-policy learning rates 10^{-6}, 5×10^{-6}, and 10^{-5}, respectively. The unlearning magnitude is therefore a consequential part of score construction for this task.

Hyperparameter-tuning heuristic. To tune the unlearned-policy learning rate without training a full unlearn–recover trajectory for every candidate, we first train one recovered policy π_e at a fixed learning rate of 10^{-6}. We then vary the unlearning rate and train each candidate π_{un} for approximately ten updates, reusing the same π_e when constructing $\log \pi_e - \log \pi_{\text{un}}$. This reduces the cost of iterating over unlearning rates.

Table 13: **Same-hardware wall-clock comparison on NBG4-3B.** All runs use eight NVIDIA A100 GPUs. On-policy SAPO and ours denote Online-2 and Offline-1 in Fig. 6, respectively.

Stage	Step	Score	Cumulative time
On-policy SAPO 800 s rollout + 300 s update/step	10	66.8	∼3 h
	50	67.2	∼15 h
	70	68.5	∼21 h
	100	66.8	∼31 h
Ours, teacher optimization 300 s/step + 7 h offline rollout	40	68.9	∼10 h
	80	68.3	∼14 h
	100	**70.8**	∼15 h
	120	70.2	∼17 h
Ours, distillation 400 s/step + 17 h teacher optimization	10	67.2	∼18 h
	16	68.3	∼19 h
	32	**73.3**	∼21 h

Table 14: Checkpoint comparison for signal construction on Qwen3-4B. The fixed score is $\log \pi_e(a \mid s) - \log \pi_{\text{un}}(a \mid s)$. We vary the recovered numerator and unlearned reference checkpoints. "Curr." substitutes the current π_{old} at that position; "None" omits the corresponding term; "–" denotes the base model.

	Checkpoint training steps										
Recovered	–	70	70	70	60	50	40	Curr.	10	10	None
Unlearned	–	10	20	30	10	10	10	50	Curr.	None	50
Student HMMT Feb 25	56.0	60.1	59.1	53.1	61.0	**64.3**	59.5	59.1	61.3	56.3	59.3

Probability tails. For positive examples, the language-model policy head produces a low-probability tail (Fig. 8), whereas negative examples remain concentrated near zero. A conventional linear critic has smoother positionwise outputs. Softmax competition is one possible explanation for these low-probability outliers. We next test whether transforming them changes downstream performance.

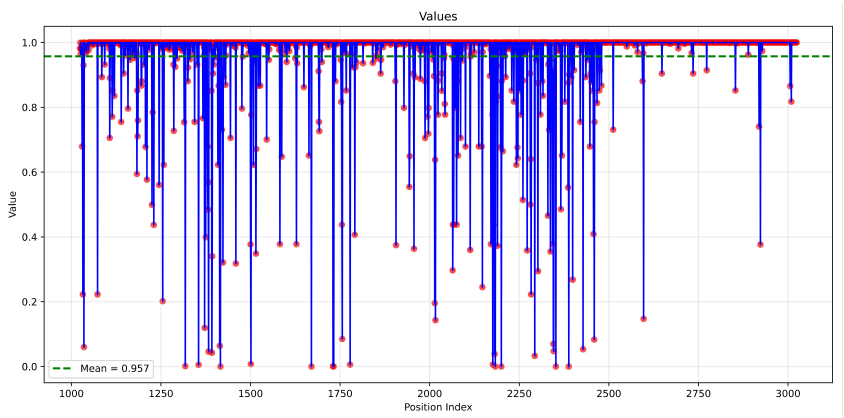


Figure 8: Token-probability distributions by response label and token position along the MSE trajectory. Positive examples contain a low-probability tail, whereas negative examples remain concentrated near zero.

Score-sign ablation. To examine sensitivity to score sign, we apply either $\max(0, \cdot)$ or $\min(\cdot, 0)$ to the raw log-ratio before full-batch whitening. In Table 18, approximately 10% of the raw log-ratio values are negative. Clipping them to zero lowers the selected accuracies, whereas clipping positive raw scores to zero yields the highest HMMT Feb 25 accuracy for both models. Whitening then recenters every response-token position, so these variants are pre-whitening score transformations.

Table 15: Post-distillation student accuracy for complementary score constructions on Qwen3-4B. "Weak score" denotes the recovered–unlearned construction; "strong score" denotes the teacher–old-policy construction. **Recovery Loss** trains the recovered numerator, while **Old Policy Loss** trains the strong teacher. β_u and β_o are their CE regularization strengths.

Weak score	Strong score	Recovery Loss	β_u	Old Policy Loss	β_o	AIME 2024	AIME 2025	HMMT Feb 25	Beyond AIME
–	–	–	–	–	–	84.9	81.1	56.0	53.8
✓	–	MSE	–	–	–	87.5	81.9	62.7	55.3
–	✓	–	–	CE	5×10^{-2}	89.2	84.1	63.8	54.4
–	✓	–	–	CE	10^{-3}	87.9	85.1	67.1	56.3
✓	✓	MSE	–	CE	5×10^{-2}	<u>89.3</u>	<u>85.2</u>	66.7	<u>56.4</u>
✓	✓	CE	10^{-3}	CE	10^{-3}	87.5	83.5	66.5	55.9
✓	✓	CE	10^{-2}	CE	5×10^{-2}	89.1	84.3	<u>66.8</u>	55.1
✓	✓	CE	10^{-3}	CE	5×10^{-2}	**90.1**	**85.4**	**69.0**	**56.7**

Table 16: Cross-model results using the MSE training strategy. Accuracies are AVG@K (%); the final column reports average generated response length on AIME 2024. The NBG4-3B MSE+D row reproduces the original all-score configuration; Table 10 reports the selected $\min(0, \hat{A})$ result.

Base Model	Method	Accuracy			Generation
		AIME 2024	AIME 2025	HMMT Feb 25	AIME 2024 average length
Qwen3-4B	Base	84.9	81.1	56.0	19.8K
	MSE+D ($\min(0, \hat{A})$)	87.5	81.9	62.7	20.8K
NBG4-3B	Base	90.9	84.8	63.9	24.8K
	MSE+D (all scores)	91.2	86.6	67.6	23.3K
Llama-3.1-Nemotron-Nano-4B-v1.1	Base	71.0	54.3	40.8	19.8K
	MSE+D	71.8	58.6	44.3	18.0K
AceReason-Nemotron-1.1-7B	Base	73.3	65.0	45.0	14.8K
	MSE+D	73.9	66.2	44.8	14.9K
JustRL-Nemotron-1.5B	Base	69.2	60.8	39.3	12.8K
	MSE+D	69.7	61.8	40.1	12.9K

The results suggest that negative-valued token scores can encode informative corrective signals. This transformation is not the default for Qwen3.5, where the diagnostic comparison shows no material score change.

Table 17: Qwen3.5-9B mathematical-reasoning and LiveCodeBenchV6 controls. The Base and CE rows restore the original mathematical comparison. MSE+D coding rows use all response-token scores; parentheses give the unlearned-policy learning rate.

Method	Config	HMMT Nov 25	HMMT Feb 25	HMMT Feb 26	LiveCodeBenchV6
Base	–	84.0	83.5	72.7	56.1
CE	–	83.3	84.1	75.0	50.7
MSE+D	10^{-6}	–	–	–	40.2
MSE+D	5×10^{-6}	–	–	–	**63.7**
MSE+D	10^{-5}	–	–	–	62.4

Table 18: Post-distillation score-sign ablation for the MSE checkpoint contrast.

Model	Method	AIME 2024	AIME 2025	HMMT Feb 25
Qwen3-4B	Base	84.9	81.1	56.0
	+Distillation (all response scores)	<u>86.7</u>	**82.8**	<u>60.1</u>
	+Distillation ($\max(0,\cdot)$)	86.6	81.4	59.5
	+Distillation ($\min(0,\cdot)$)	**87.5**	<u>81.9</u>	**62.7**
NBG4-3B	Base	90.9	84.8	63.9
	+Distillation (all response scores)	<u>91.2</u>	<u>86.6</u>	<u>67.6</u>
	+Distillation ($\max(0,\cdot)$)	90.5	85.5	65.7
	+Distillation ($\min(0,\cdot)$)	**91.3**	**87.4**	**68.6**