ESPO: EARLY-STOPPING PROXIMAL POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

When a reasoning trajectory becomes unrecoverable, standard reinforcement learning algorithms still generate until an end-of-sequence (EOS) token is generated or the rollout horizon is reached. This can waste computation and add estimation noise from post-failure tokens. We propose **ESPO** (**E**arly-**S**topping Proximal **P**olicy **O**ptimization), which identifies likely trajectory failures on-the-fly and terminates rollouts early. At each generation step, ESPO computes a surrogate regret using only the logits already computed during sampling, and terminates when the normalized, smoothed deviation exceeds a critic-dependent threshold. Truncated trajectories are treated as absorbing failure states with a terminal penalty at the detected stopping step, propagated through the retained prefix by GAE, without an additional reward model or human annotation. Across three backbones from two model series, on both mathematical reasoning and code generation, ESPO reaches 46.35 on AIME 2024, 85.86 on AMC 2023 and 87.42 on MATH-500 with DeepSeek-R1-Distill-Qwen-7B — each the best figure among the five methods we compare, as is its average over six benchmarks — while decoding 21.7% fewer rollout tokens than PPO and cutting per-step wall-clock by 23.2%. Our code is available in the supplementary material.

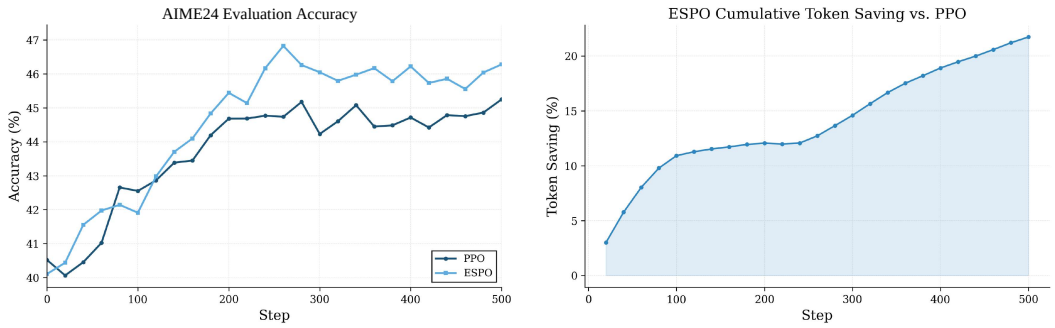
1 INTRODUCTION

Reinforcement learning (RL) improves reasoning in large language models (LLMs), particularly on multi-step reasoning tasks (Shao et al., 2024; DeepSeek-AI, 2025). Process supervision further improves chain-of-thought reasoning through step-level feedback (Lightman et al., 2024). RLHF also supports instruction following and alignment with human preferences (Ouyang et al., 2022; Casper et al., 2023). In these settings, models must generate trajectories of hundreds to thousands of tokens, each conditioned on the entire preceding context. This long-horizon generation process introduces significant challenges in credit assignment and training efficiency (Shao et al., 2024). Some reasoning errors can be corrected later; others leave prefixes from which the model has little or no chance of recovery. Examples include an uncorrected mathematical error or an abandoned proof path. The eventual reward is zero or negative, yet standard policy gradient algorithms force the policy to continue generating until the end or the fixed rollout cap $T_{\max}$. Tokens after an irrecoverable failure cannot change the terminal outcome, yet continuing to generate and train on them consumes computation and can introduce estimation noise. We call this the *rollout continuation problem*.

Existing work addresses adjacent problems but leaves rollout continuation waste largely unsolved. Human-supervised process reward models (Lightman et al., 2024) provide step-level feedback at the cost of annotating intermediate reasoning steps. GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025) use group-normalized advantages and clipped importance weights, but normally generate until EOS or a length cap, without an explicit failure-detection stopping rule. Option-Critic (Bacon et al., 2017) learns a dedicated termination function alongside option policies under a shared expected-return objective, adding termination parameters and their updates. These methods do not implement failure-conditioned stopping using only the actor and critic signals available in PPO (Schulman et al., 2017).

We introduce **ESPO** (**E**arly-**S**topping Proximal **P**olicy **O**ptimization), a rollout termination mechanism that uses the policy's existing logit vector together with the critic's value estimate. Our working hypothesis is that a high-regret, low-value prefix is less likely to recover: large mode-versus-sample

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) Accuracy

(b) Token Saving

Figure 1: **ESPO surpasses PPO on AIME 2024 at lower token cost (DeepSeek-R1-Distill-Qwen-7B). Left:** AIME 2024 avg@32 (%) vs. training steps. ESPO surpasses PPO earlier and maintains the lead through training. **Right:** It records ESPO's cumulative rollout token saving during training compared with PPO. (tokens per step differ because ESPO truncates failing trajectories). Both methods share prompts, the task scorer, and evaluation protocol; ESPO additionally applies its truncation penalty.

log-probability gaps, combined with low predicted remaining value, provide a candidate failure signal. ESPO formalizes this intuition through the following components:

1. **Per-step surrogate regret.** A regret signal computed from the logit gap at each timestep, capturing deviation from the greedy action.
2. **EMA normalization.** Historical batch means and variances standardize the regret signal, while a separate temporal EMA smooths token-level fluctuations within each trajectory.
3. **Value-gated stopping criterion.** A termination rule stops the rollout when the normalized, smoothed deviation exceeds a value-dependent threshold.
4. **Terminal failure penalty.** An absorbing terminal penalty provides explicit failure feedback for stopped trajectories, and GAE propagates the terminal change through the retained prefix.

We evaluate ESPO on three backbones from two model series (DeepSeek-R1-Distill-Qwen at 1.5B and 7B, and Qwen3-8B) against PPO, GRPO, GSPO and DAPO, on both mathematical reasoning and code generation. At the 7B scale ESPO attains an average accuracy of 64.89 across six math benchmarks against 62.56 for PPO and 63.41 for GSPO, the strongest group-normalized baseline, while decoding 21.7% fewer rollout tokens than PPO. Compared with rate-matched random truncation (Variant F in Table 2), ESPO achieves higher accuracy at a similar average rollout length. This supports the contribution of selecting *where* to stop beyond simply shortening rollouts.

2 Related Work

PPO-based LLM alignment and reasoning. PPO (Schulman et al., 2017) underpins RLHF (Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023) with clipped updates. GRPO (Shao et al., 2024) replaces the learned baseline with group-normalized advantages; DAPO (Yu et al., 2025) adds dynamic sampling and decoupled clipping; GSPO (Zheng et al., 2025) uses sequence-level importance ratios and clipping. These methods normally generate to EOS or a length cap. ESPO adds failure-conditioned stopping with a critic-based value gate. Changing termination affects both the sampled trajectory and its terminal reward, not only the weighting of a completed response.

Process rewards and credit assignment. Human-supervised process reward models (PRMs) (Lightman et al., 2024) use intermediate annotations, whereas outcome-based verifiers (Cobbe et al., 2021; Uesato et al., 2022) score completed responses. PRIME (Cui et al., 2025) learns implicit process rewards from rollouts and outcome labels, showing that step annotations are not essential for

process supervision. Offline DPO (Rafailov et al., 2023) instead uses fixed preference data. ESPO reuses actor logits and critic values for termination, without training a separate PRM.

Learned termination. Option-Critic (Bacon et al., 2017) learns option policies and termination functions under a shared return objective. Pardo et al. (2018) distinguish time limits imposed on continuing tasks, which require bootstrapping, from true terminals. ESPO introduces penalized absorbing failure transitions, with zero bootstrap and no dedicated termination network.

Efficient reasoning and rollout stopping. Chen et al. (2025) study redundant reasoning and mitigate overthinking through self-training. ESTAR (Wang et al., 2026) learns stop cues with supervised fine-tuning and stop-aware RL; ESTAR-LITE trains a stopping classifier. TERMINATOR (Nagle et al., 2026) trains an exit module on first-arrival answer positions with a frozen backbone. DRPO (Li et al., 2026) decouples length rewards for correct and incorrect rollouts; Latent-GRPO (Deng et al., 2026) compresses reasoning in latent space. ESPO instead stops likely failing token-level training rollouts using actor–critic signals and a terminal failure penalty, without supervised stop labels or per-step length rewards.

Test-time compute allocation. Token-Budget-Aware Reasoning (Han et al., 2025) specifies reasoning budgets in prompts and adapts them to problem complexity; s1 (Muennighoff et al., 2025) combines supervised reasoning traces with budget forcing to terminate or extend thinking. Such budget controls do not themselves identify a prefix as likely to fail. DeepConf (Fu et al., 2025) filters low-confidence traces and can stop them during generation, making it closely related to confidence-based rollout control. Its online mode uses local token-group confidence and calibrates a stopping threshold from warmup traces for each prompt. This inference-time selection does not update the policy, whereas ESPO uses a critic gate and changes training rewards and prefix GAE. Neither confidence signal alone certifies that a stopped prefix was irrecoverable. Depth-adaptive decoding (Elbayad et al., 2020) and CALM (Schuster et al., 2022) allocate layer depth per token, while speculative decoding (Leviathan et al., 2023) verifies multiple draft tokens in parallel. These methods target token-generation efficiency rather than deciding which training suffixes to omit; their efficiency gains therefore address a different axis from ESPO's training-time trajectory selection.

3 PRELIMINARIES

Token-level RL for LLM generation. We formulate autoregressive decoding as a finite-horizon Markov decision process. Given an input prompt $x \sim \mathcal{D}$, tokens are indexed by $t = 1, \ldots, T$ and the state at step t is $s_t = (x, y_{<t})$, namely the prompt together with the previously generated tokens. The action $a_t \in \mathcal{V}$ is the next token selected from the vocabulary, and the transition is deterministic: $s_{t+1} = (x, y_{<t}, a_t)$. An episode terminates when the model emits an end-of-sequence token or reaches the horizon $T_{\max}$. Following standard outcome-supervised RL for reasoning tasks, we consider sparse rewards,

$$r_t = 0 \quad (t < T), \qquad r_T = R(x, y_{1:T}), \tag{1}$$

where $R(x, y_{1:T})$ denotes the final task reward, e.g., a binary correctness score.

Actor-critic PPO. Let $\pi_\theta(a_t \mid s_t)$ denote the policy and $V_\phi(s_t)$ the critic's value function. PPO updates π_θ by maximizing the clipped surrogate objective

$$\mathcal{L}_{\mathrm{PPO}}(\theta) = \mathbb{E}_t\Big[\min\Big(\rho_t \hat{A}_t,\ \mathrm{clip}(\rho_t, 1-\epsilon_{\mathrm{ppo}}, 1+\epsilon_{\mathrm{ppo}})\hat{A}_t\Big)\Big], \tag{2}$$

where

$$\rho_t = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\mathrm{old}}(a_t \mid s_t)} \tag{3}$$

is the importance ratio. Advantages are estimated with generalized advantage estimation (GAE) (Schulman et al., 2016):

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \tag{4}$$

$$\hat{A}_t = \sum_{l=0}^{T-t} (\gamma\lambda_{\mathrm{GAE}})^l \delta_{t+l}, \qquad V_\phi(s_{T+1}) = 0. \tag{5}$$

This actor-critic formulation is important for our method because the critic provides a state-dependent estimate of remaining return, while GAE propagates the effect of early termination backward to preceding tokens.

4 ESPO: Early-Stopping Proximal Policy Optimization

Overview. ESPO modifies rollout collection rather than the algebraic PPO surrogate. It smooths a token-level deviation signal and compares it with a critic-dependent threshold. A triggered stop creates an absorbing failure transition; PPO and GAE then use the shortened trajectory. For each training batch, the stopping policy, gate critic, and statistics are frozen, giving a locally fixed augmented MDP for rollout collection and PPO updates (Appendix A.2).

4.1 Stepwise deviation signal

At state s_t, let $a_t \sim \pi_k(\cdot \mid s_t)$ be the sampled token under the batch's frozen behavior policy. The stepwise deviation, or surrogate regret, is

$$g_t = \max_{a \in \mathcal{V}} \log \pi_k(a \mid s_t) - \log \pi_k(a_t \mid s_t) \geq 0. \tag{6}$$

It measures deviation from the policy mode, not task regret or a guaranteed reasoning error. A large gap identifies a sampled token that is unlikely relative to the mode at the same prefix, rather than uncertainty over the whole distribution. It is available from the decoding logits; evaluating the value gate also requires the critic. The total cost is assessed in Appendix I.

4.2 Normalized and smoothed stopping statistic

Let k index training batches, distinct from token positions t. We normalize using statistics available before batch k:

$$\tilde{g}_t = \mathrm{clip}\left(\frac{g_t - \mu_{g,k}}{\sqrt{\sigma^2_{g,k} + \delta}}, -c, c \right), \qquad \delta > 0, \quad c > 0. \tag{7}$$

After collecting the batch, use its emitted-token regrets $g_{\mathcal{B}_k}$ to prepare the next batch's normalization:

$$\mu_{g,k+1} = \alpha_{\mathrm{ema}} \mu_{g,k} + (1 - \alpha_{\mathrm{ema}}) \overline{g}_{\mathcal{B}_k}, \tag{8}$$

$$\sigma^2_{g,k+1} = \alpha_{\mathrm{ema}} \sigma^2_{g,k} + (1 - \alpha_{\mathrm{ema}}) \operatorname{Var}(g_{\mathcal{B}_k}). \tag{9}$$

Here $\overline{g}_{\mathcal{B}_k}$ is the sample mean of the emitted-token regrets $g_{\mathcal{B}_k}$ in batch k, so $\mu_{g,k}$ tracks the running first moment of the regret distribution, while the variance statistic $\sigma^2_{g,k}$ is an exponential moving average of the within-batch variances $\operatorname{Var}(g_{\mathcal{B}_k})$; the decay $\alpha_{\mathrm{ema}} \in [0, 1)$ sets how quickly both statistics track the drift of that distribution across training. Statistics are initialized from an unstopped warmup batch; until they are available, only raw regrets are collected and stopping is disabled. No current-batch suffix enters its own normalization. Within each trajectory,

$$z_t = \alpha_{\mathrm{s}} z_{t-1} + (1 - \alpha_{\mathrm{s}}) \tilde{g}_t, \qquad z_0 = 0. \tag{10}$$

Here α_{s} is a temporal smoothing coefficient, not a normalization parameter; z_t is an exponentially weighted statistic rather than an unweighted cumulative sum. The batch EMA tracks the regret scale across training, while temporal smoothing emphasizes persistent deviations over isolated spikes.

4.3 Value-gated early termination

After emitting a nonterminal token, ESPO tests

$$z_t > \beta_k \max\big(V_k(s_t), \varepsilon\big), \qquad \varepsilon > 0. \tag{11}$$

$V_k(s_t)$ is the frozen *pre-action* critic value. High predicted return grants more tolerance; the floor makes the threshold nonnegative, and strictly positive when $\beta_k > 0$. The signal is heuristic: a correct off-mode token can still trigger a stop. At an active batch boundary, the controller updates

$$\beta_{k+1} = \mathrm{clip}(\beta_k + \eta_\beta(\hat{\rho}_{\mathrm{stop},k} - \tau), \beta_{\min}, \beta_{\max}), \tag{12}$$

Algorithm 1 ESPO Rollout Collection for One Prompt in Batch k

Require: Prompt x, task scorer R, horizon $T_{\max}$, EOS token, failure reward r_{fail}
Require: Frozen $\pi_k, V_k, \beta_k, \mu_{g,k}, \sigma^2_{g,k}$; $\alpha_{\text{s}}, c, \delta, \varepsilon$; activation flag b_k
$s_1 \leftarrow x$; $z_0 \leftarrow 0$; trajectory $\mathcal{T} \leftarrow \varnothing$; strict-stop indicator $\chi_{\text{stop}} \leftarrow 0$
for $t = 1, \ldots, T_{\max}$ **do**
  Sample $a_t \sim \pi_k(\cdot \mid s_t)$; cache its log probability and pre-action value $V_k(s_t)$
  $y_{1:t} \leftarrow (a_1, \ldots, a_t)$; $s_{t+1} \leftarrow (x, y_{1:t})$
  Compute and store g_t by Equation (6); initialize $z_t \leftarrow 0$
  If statistics are initialized, compute $\tilde{g}_t, z_t$ by Equations (7) and (10)
  $r_t \leftarrow 0$; terminal indicator $d_t \leftarrow 0$
  if $a_t = \text{EOS}$ **or** $t = T_{\max}$ **then**
    $r_t \leftarrow R(x, y_{1:t})$; $d_t \leftarrow 1$
  else if $b_k = 1$ **and** $z_t > \beta_k \max(V_k(s_t), \varepsilon)$ **then**
    $r_t \leftarrow r_{\text{fail}}$; $\chi_{\text{stop}} \leftarrow 1$; $d_t \leftarrow 1$
  end if
  Append (s_t, a_t, r_t, d_t) to $\mathcal{T}$
  if $d_t = 1$ **then**
    $T \leftarrow t$; set the terminal bootstrap to zero; **break**
  end if
end for
return $\mathcal{T}, T, \chi_{\text{stop}}$, cached log probabilities and values, emitted-token regrets

where $\hat{\rho}_{\text{stop},k}$ is the fraction of strictly truncated rollouts and τ is the target. Excess stopping raises the next batch's threshold; insufficient stopping lowers it. This adjusts stopping frequency while the joint signal selects which trajectories to stop. Stopping is activated after critic warmup; the warmup procedure and activation criteria are specified in Appendix D.

4.4 Failure transition

If a strict truncation occurs at T_{stop}, the reward is

$$r_t = 0 \ (t < T_{\text{stop}}), \qquad r_{T_{\text{stop}}} = r_{\text{fail}}, \qquad V_k(s_{T_{\text{stop}}+1}) = 0. \tag{13}$$

The trigger token is retained. EOS or the task horizon instead receives the ordinary task reward, including when a trigger would hold on that same token. Both are terminal with zero bootstrap under our finite-horizon task definition. At a strict stop, the penalty supplies explicit failure feedback through the final TD residual $r_{\text{fail}} - V_k(s_{T_{\text{stop}}})$, which is negative when $V_k(s_{T_{\text{stop}}}) > r_{\text{fail}}$. GAE propagates this terminal feedback through the retained prefix; Corollary 5 characterizes the resulting change in the gradient estimate. For a shared sampled prefix and fixed critic, the raw GAE change relative to full continuation at position t equals $(\gamma\lambda_{\text{GAE}})^{T_{\text{stop}}-t}$ times the change at the stop. Thus, stopping removes suffix updates and changes prefix advantages before normalization and clipping.

In Algorithm 1, $b_k = 1$ only after warmup has ended and normalization statistics are initialized. The indicator d_t marks any terminal transition, whereas χ_{stop} marks only a strict ESPO truncation. After collecting the batch, compute terminal-masked GAE, update actor and critic with PPO, and prepare the next batch's statistics, warmup status, and controller value. The reference gate remains unchanged during the PPO epochs. Appendix A motivates selective truncation: under conditional-unbiasedness and orthogonality assumptions, oracle masking of suffixes with zero true advantages preserves the expected gradient while reducing estimation variance. ESPO combines suffix removal with a terminal failure penalty that propagates through retained-prefix GAE, coupling reduced rollout computation with explicit failure feedback for policy learning.

5 Experiments

5.1 Setup

Models and Benchmarks We evaluate three backbones from two model series at three parameter scales: DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI,

2025), and Qwen3-8B (Yang et al., 2025). For mathematical reasoning, models are trained on DAPO-Math-17k (Yu et al., 2025) and evaluated on six held-out benchmarks: AIME24 (Jia, 2024), AMC23 (math-ai, 2025), MATH500 (Lightman et al., 2024), GSM8K (Cobbe et al., 2021), Minerva-Math (Lewkowycz et al., 2022) and OlympiadBench (He et al., 2024). To test whether the stopping criterion transfers beyond mathematics, we further run code-domain RL on the Eurus-2 coding split (Cui et al., 2025) with DeepSeek-R1-Distill-Qwen-7B, evaluating on LiveCodeBench (Jain et al., 2024) and HumanEval+ (Liu et al., 2023).

Baselines and Implementation We compare ESPO against (1) the Base Model; (2) PPO (Schulman et al., 2017), a token-level actor-critic with full-horizon rollouts; (3) GRPO (Shao et al., 2024), which replaces the critic with group-normalized advantages; (4) GSPO (Zheng et al., 2025), which weights importance at sequence rather than token granularity; and (5) DAPO (Yu et al., 2025), which adds dynamic sampling and decoupled clipping to GRPO. In the main experiments, all methods use 8 rollouts and a global batch size of 64, and are trained for 500 steps under shared data, task-reward, sampling, and evaluation settings, apart from method-specific penalties and sampling procedures. All algorithms are implemented via verl (Sheng et al., 2025) using outcome-based rewards, and all experiments were conducted on 8 $\times$ H20 GPUs. Training and evaluation protocols, including the per-benchmark sampling counts and inference hyperparameters, are detailed in Appendix B. Evaluation disables early stopping for all ESPO policies. Reported avg@n scores average correctness across n samples per problem, estimating single-sample accuracy rather than success in any of n attempts.

Computational cost We report **Tokens$_{\text{gen}}$**, the tokens actually decoded during rollouts, to compare generation workload (Tables 1 to 3). ESPO stops decoding at the detected failure step, whereas DAPO also decodes groups discarded by dynamic sampling; Appendix C details this accounting. Since token counts alone do not capture critic computation, optimization, or system overhead, we also compare total wall-clock time per completed training update, averaged over the entire run (Tables 3 and 9). Together, these metrics characterize both generation workload and end-to-end training cost.

Table 1: **Main results on mathematical reasoning.** Five methods, 500 **steps** each; full setup in Section 5. Accuracies are avg@32 on AMC23, AIME24 and MATH500 and avg@8 on the other three. **Tokens$_{\text{gen}}$** counts tokens decoded during rollout, including for DAPO the groups dynamic sampling discards, and **Avg Tokens** is the mean rollout length per trajectory. Best per column **bolded** (minimum for the two token columns); our row shaded.

Method	GSM8K	Minerva	OlympiadBench	AMC23	AIME24	MATH500	Avg	Tokens$_{\text{gen}}$(M)	Avg Tokens
DeepSeek-R1-Distill-Qwen-1.5B									
Base Model	79.24	18.38	30.57	58.28	20.31	74.81	46.93	–	–
PPO	80.68	20.68	36.04	68.36	23.02	79.65	51.40	1069.66	4178
GRPO	81.24	20.27	36.94	69.14	23.85	80.06	51.92	1290.00	5039
GSPO	80.93	21.05	37.61	70.16	22.81	80.58	52.19	1283.00	5012
DAPO	81.12	20.86	**38.47**	70.23	**24.27**	80.28	52.54	2943.23	4781
ESPO (Ours)	**81.47**	**21.32**	38.15	**71.88**	23.96	**81.53**	**53.05**	**927.96**	**3625**
DeepSeek-R1-Distill-Qwen-7B									
Base Model	84.80	29.32	40.12	78.59	40.10	83.36	59.38	–	–
PPO	86.61	30.79	44.45	82.89	45.21	85.43	62.56	1072.40	4189
GRPO	87.11	30.19	46.60	83.36	45.63	86.48	63.23	1059.00	4137
GSPO	87.74	31.53	45.01	83.83	45.83	86.55	63.41	1088.00	4250
DAPO	**87.95**	32.44	45.07	83.75	44.48	85.95	63.27	3392.11	4043
ESPO (Ours)	87.48	**34.28**	**47.93**	**85.86**	**46.35**	**87.42**	**64.89**	**839.24**	**3278**
Qwen3-8B									
Base Model	93.82	27.57	34.83	59.06	17.19	77.58	51.68	–	–
PPO	94.88	33.00	49.42	89.77	39.69	89.28	66.01	1224.00	4781
GRPO	**95.08**	34.83	50.02	89.53	40.00	88.71	66.36	1207.00	4715
GSPO	94.80	34.74	50.67	91.72	40.52	87.84	66.72	1132.00	4422
DAPO	94.69	34.65	50.49	91.41	41.77	89.72	67.12	5388.00	5036
ESPO (Ours)	94.56	**35.71**	**52.11**	**92.66**	**43.02**	**90.85**	**68.15**	**945.35**	**3693**

5.2 Main Results

Comparison with PPO. Table 1 shows that ESPO improves the six-benchmark average over PPO while reducing rollout generation on all three backbones. On DeepSeek-R1-Distill-Qwen-1.5B,

DeepSeek-R1-Distill-Qwen-7B, and Qwen3-8B, the respective ESPO/PPO averages are 53.05/51.40, 64.89/62.56, and 68.15/66.01, with 13.2%, 21.7%, and 22.8% fewer decoded tokens. Mean rollout lengths also decrease: from 4189 to 3278 tokens on 7B and from 4781 to 3693 on Qwen3-8B. The reduction extends to end-to-end training cost: on the 7B math runs, ESPO takes 249.1 s per completed update versus PPO's 324.3 s, a 23.2% reduction averaged over all 500 training steps, including warmup (Table 9). Appendix E further breaks down accuracy and evaluation-time token usage by problem difficulty.

Comparison with group-normalized methods. GRPO, GSPO and DAPO reshape advantage estimation and policy updates, whereas ESPO changes rollout termination and the terminal reward. On the 7B math runs, ESPO achieves a six-benchmark average of 64.89 versus 63.23 (GRPO), 63.41 (GSPO), and 63.27 (DAPO), gaining $+1.47$ points over GSPO, the highest-scoring baseline by average. On 1.5B and Qwen3-8B, it exceeds DAPO, the strongest baseline by average on each backbone, by $+0.51$ and $+1.03$ points, respectively. ESPO also decodes fewer tokens than all three methods on each backbone. Against the lowest-token group-normalized baseline, it saves 20.8% on 7B (839.24 M versus GRPO's 1059 M) and 16.5% on Qwen3-8B (945.35 M versus GSPO's 1132 M). On 7B, its mean completed-update time of 249.1 s is below GRPO's 288.1 s, GSPO's 274.5 s, and DAPO's 376.7 s (Table 9), even with value-network computation included. These times cover all 500 steps, including every dynamic-sampling batch required by DAPO. Overall, ESPO maintains a modest advantage in average accuracy while delivering substantial savings in rollout generation and training costs.

6 Ablation Studies

Table 2 isolates the contribution of each ESPO component by removing or replacing it while keeping all other settings fixed, on DeepSeek-R1-Distill-Qwen-7B at the same protocol as Table 1.

Table 2: **Component ablation on DeepSeek-R1-Distill-Qwen-7B**, at the same 500-step protocol as Table 1. (A) is full ESPO — the same run as the 7B ESPO row of Table 1, whose AIME24 score is 46.35 there — and every other row removes or replaces exactly one component. AIME24 is avg@32; the token columns follow Table 1. ↑ higher is better, ↓ lower is better; the best value in each column is **bolded**.

Variant	AIME24 ↑	$\text{Tokens}_{\text{gen}}$(M) ↓	Avg Tokens ↓
(A) Full ESPO (ours)	**46.4**	**839.24**	**3278**
(B) w/o warmup	44.2	858.37	3353
(C) w/o terminal failure penalty	43.6	901.65	3522
(D) Value-only stop ($V_\phi < \nu$, no regret)	44.0	1090.05	4258
(E) Regret-only stop ($z_t > \zeta$, no value gate)	44.8	1086.51	4244
(F) Random stop (rate-matched to A)	42.4	855.59	3342

Critic warmup and terminal penalty (A vs. B, C). Removing the adaptive warmup of Section 4 costs 2.2 points (46.4 to 44.2). Warmup gives the critic an initial fitting period before its predictions guide stopping. Removing the terminal failure penalty costs 2.7 points and lengthens the average rollout by 244 tokens (3278 to 3522). The penalty gives a negative final TD residual when $V_k(s_T) > r_{\text{fail}}$, and GAE propagates it through the retained prefix. Even without the penalty, truncation changes the return and bootstrap; it is not merely a token mask. The terminal penalty provides explicit failure feedback for stopped trajectories. The lower accuracy and longer rollouts observed without it demonstrate its contribution to effective training.

Stopping rule (A vs. D, E, F). Variant D stops when the critic value falls below a value-only threshold ν ($V_\phi(s_t) < \nu$). Variant E stops when the smoothed deviation exceeds a regret-only threshold ζ ($z_t > \zeta$), without value gating. Variant F uses random stopping with its stop rate matched to full ESPO. Value-only stopping depends on the critic's absolute scale, while regret-only stopping lacks the recovery allowance provided by the value gate. Variants D and E score 44.0 and 44.8, below full ESPO's 46.4, supporting the use of both signals. The random baseline scores 42.4 at a similar average rollout length (3342 versus 3278 tokens). Its 4.0-point accuracy deficit at comparable generation cost supports the contribution of selecting *where* to stop beyond shortening the training sequences alone. Appendix G places these variants on the accuracy–token plane.

Table 3: **Code-domain RL on the Eurus-2 coding split (DeepSeek-R1-Distill-Qwen-7B).** All methods are trained for **500 steps**, matching Table 1. LiveCodeBench and HumanEval+ are reported as avg@8 (mean per-sample accuracy). Token columns follow Table 1: **Tokens$_{\text{gen}}$** counts the tokens actually decoded, including for DAPO the groups that dynamic sampling discards, and **Avg Tokens** is the mean training rollout length. **Wall-clock/Step** is the mean total time to complete one training update, averaged over all 500 steps, including warmup where applicable, as in Table 9. For DAPO, this includes all dynamic-sampling batches needed for that update. The best result in each column is **bolded** (the minimum for the two token columns and for wall-clock, where lower is better).

Method	LiveCodeBench	HumanEval+	Tokens$_{\text{gen}}$(M)	Avg Tokens	Wall-clock/ Step(s)
Base Model	21.11	82.77	–	–	–
PPO	27.84	85.37	1415	5528	442.55
GRPO	26.57	85.59	1339	5232	387.68
GSPO	26.87	86.05	1363	5323	368.14
DAPO	27.99	86.51	3807	5374	581.04
ESPO (Ours)	**28.59**	**87.88**	**1145**	**4473**	**349.67**

7 Code-Domain Generalization

The stopping criterion in Section 4 is domain-agnostic by construction: it reads only the actor's logit gap and the critic's value estimate. To evaluate whether ESPO remains effective in reasoning domains beyond mathematics, we conduct experiments on code generation. Table 3 reports the code-domain results. ESPO obtains 28.59 on LiveCodeBench and 87.88 on HumanEval+, the best figure in both columns, against 27.84 / 85.37 for PPO and 27.99 / 86.51 for DAPO, while decoding the fewest tokens among the five methods (1145 M, 14.5% below GRPO's 1339 M and 19.1% below PPO's 1415 M) at the shortest mean rollout (4473 against PPO's 5528 tokens) and the lowest mean total time per completed training update (349.67 s against PPO's 442.55 s, a 21.0% reduction), averaged over the entire 500-step run. ESPO's correct-yet-truncated rate of 2.98% in the code domain lies within the 2.70%–3.08% range of the three math runs (Appendix F). Following the protocol in Figure 2c, we identify false stops by continuing generation beyond the stopping point and verifying the completed output. The correct-yet-truncated rate measures the fraction of all rollouts that trigger early stopping despite yielding correct solutions when allowed to complete.

Effectiveness in the code domain. Table 3 shows that ESPO combines higher solution accuracy with lower training cost. Relative to DAPO, the strongest baseline on both benchmarks, ESPO gains 0.60 points on LiveCodeBench and 1.37 points on HumanEval+. Its efficiency advantage also extends beyond PPO: mean update time is 5.0% lower than GSPO, the fastest baseline, and decoded-token usage is below GRPO, the most token-efficient baseline. Thus, the accuracy improvements come with reductions in both generation work and end-to-end update time. Together with the low correct-yet-truncated rate, these results demonstrate that ESPO is effective in the code domain as well: the same regret–value stopping criterion improves the accuracy–cost trade-off without introducing code-specific stopping features.

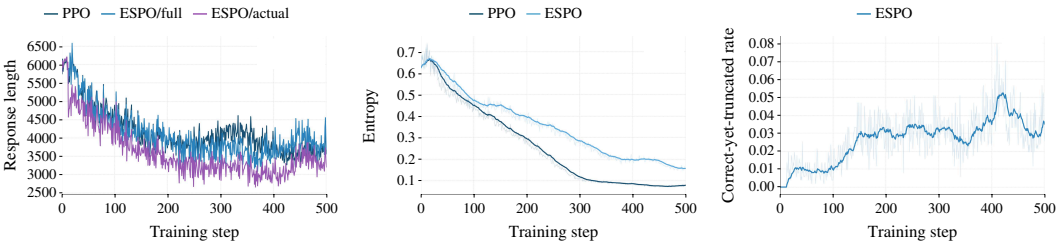
8 Training Dynamics

Response length. Figure 2a compares full counterfactual lengths (*ESPO/full*) with realized lengths (*ESPO/actual*). For this diagnostic only, we record the first strict stop trigger but continue sampling from the same policy to EOS or the length cap. *ESPO/full* counts all generated tokens; *ESPO/actual* counts the prefix through the trigger token, or the full response if no trigger occurs. Both are averaged over all rollouts; post-stop tokens are excluded from training (Appendix J). The full lengths track PPO's mean, while the realized lengths are substantially lower. This suggests that token savings primarily come from the stopping mechanism rather than from training the model to produce intrinsically shorter responses.

Actor entropy. Figure 2b shows slower policy-entropy decay under ESPO than PPO, with no entropy collapse over the plotted training period. The update-level explanation involves two effects: truncation removes suffix-gradient terms, while the terminal penalty changes retained-prefix advantages through GAE. After an irrecoverable failure, subsequent token choices cannot restore success, yet their estimated advantages still update the policy. For entropy-decreasing suffix updates,

removal eliminates pressure that suppresses low-probability alternatives or reinforces dominant continuations. The resulting entropy trend reflects the combined effect of suffix removal and prefix updates. Appendix J provides the detailed analysis.

Correct-yet-truncated trajectories. Figure 2c shows a low incidence of false stops: averaged over batches, only 2.7% of all rollouts on this 1.5B run are selected for truncation yet would complete correctly. Relative to the overall stop rate, this indicates that ESPO predominantly stops trajectories whose natural continuations would fail, supporting the precision of its early-stopping decisions while limiting the loss of successful continuations. Appendix F provides detailed stop rates and correct-yet-truncated results across runs; Appendix G reports the accuracy–token trade-off as stopping becomes more aggressive, which provides a more comprehensive analysis.



(a) Response length. (b) Actor entropy. (c) Correct-yet-truncated rate.

Figure 2: Training dynamics on DeepSeek-R1-Distill-Qwen-1.5B. (a) *ESPO/full* and *ESPO/actual* are the full counterfactual and realized lengths, averaged over all ESPO rollouts; *PPO* is the mean PPO length. (b) Mean next-token policy entropy computed over full diagnostic rollouts before truncation masking, including tokens beyond ESPO's stopping point. ESPO shows slower entropy decay than PPO (Appendix J). (c) Fraction of all rollouts per training step that trigger early stopping yet yield correct answers when continued to completion. The mean of 2.7% measures the incidence of successful continuations lost to early stopping (Appendix F).

9 Limitations and Future Work

Confidently incorrect reasoning can evade detection because a small logit gap keeps surrogate regret low. Undetected failures continue as in PPO, whereas false stops discard successful continuations and assign failure feedback to recoverable prefixes. Reducing false stops therefore takes priority in preserving useful learning signals. Such stops can occur when a correct but off-mode token raises the smoothed deviation above the threshold set by a low critic value. Although the measured correct-yet-truncated rates are low (Appendix F), future work can refine the joint stopping rule to further reduce these losses and explore automatic hyperparameter adaptation to reduce sensitivity across models and tasks (Appendix H). Extending the criterion to tool-use and multi-turn agentic settings, where errors span environment steps rather than individual tokens, is another natural direction.

10 Conclusion

We introduced ESPO, a regret-aware rollout termination method for LLM reasoning RL: a surrogate regret signal from the actor's own logits, gated by a value threshold, truncates likely failing trajectories on-the-fly and assigns a terminal penalty at the stopping step, with no reward model or additional annotation. Across three backbones and two domains, ESPO reaches 64.89 on DeepSeek-R1-Distill-Qwen-7B against 62.56 for PPO and 63.41 for GSPO, at 21.7% fewer rollout tokens. Rate-matched random truncation performs worse than the regret–value rule at a similar average rollout length, supporting the contribution of selecting where to stop beyond shortening rollouts. The ablations further support retaining both critic warmup and the terminal failure penalty. Together, these results highlight selective rollout termination as a practical way to improve the accuracy–cost trade-off in reasoning RL.

AI USE STATEMENT

We used AI assistance for polishing language, checking literature, and checking mathematical arguments. It was not used to run experiments or generate the reported results. The authors take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The rollout procedure and update order are given in Section 4 and Algorithm 1; training and evaluation settings and warmup criteria are in Appendices B and D. Token accounting and the instrumented diagnostics are specified in Appendices C, F and J. Appendix captions identify experiments with budgets different from the main tables. Appendix A.7 summarizes the assumptions and limitations of the theory. Code is provided in the supplementary material.

REFERENCES

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022. URL https://arxiv.org/abs/2212.08073.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? On the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 9487–9499. PMLR, 2025. URL https://proceedings.mlr.press/v267/chen25bx.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Jingcheng Deng, Zihao Wei, Liang Pang, Junhong Wu, Shicheng Xu, Zenghao Duan, and Huawei Shen. Latent-GRPO: Group relative policy optimization for latent reasoning. *arXiv preprint arXiv:2604.27998*, 2026.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. Depth-adaptive transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SJg7KhVKPH.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv preprint arXiv:2508.15260*, 2025. URL https://arxiv.org/abs/2508.15260.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1274. URL https://aclanthology.org/2025.findings-acl.1274/.

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

Naman Jain, King Han, 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. *CoRR*, abs/2403.07974, 2024. URL https://arxiv.org/abs/2403.07974.

Maxwell Jia. AIME 2024 dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024, 2024.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286. PMLR, 2023. URL https://proceedings.mlr.press/v202/leviathan23a.html.

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.

Gang Li, Yan Chen, Ming Lin, and Tianbao Yang. DRPO: Efficient reasoning via decoupled reward policy optimization. In *14th International Conference on Learning Representations, ICLR 2026*, 2026. URL https://arxiv.org/abs/2510.04474.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024.

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 *Advances in Neural Information Processing Systems*, volume 36, pp. 21558–21572. Curran Associates, Inc., 2023. doi: 10.52202/075280-0943. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/43e9d647ccd3e4b7b5baab53f0368686-Abstract-Conference.html.

math-ai. AMC23 dataset. https://huggingface.co/datasets/math-ai/amc23, 2025. Test split, 40 problems.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Alliot Nagle, Jakhongir Saydaliev, Dhia Garbaya, Michael Gastpar, Ashok Vardhan Makkuva, and Hyeji Kim. TERMINATOR: Learning optimal exit points for early stopping in chain-of-thought reasoning. *arXiv preprint arXiv:2603.12529*, 2026.

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.

Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. Time limits in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4045–4054. PMLR, 2018. URL http://proceedings.mlr.press/v80/pardo18a.html.

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. In *Advances in Neural Information Processing Systems 36: NeurIPS 2023, New Orleans, LA, USA*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing Methods in Statistics*, pp. 233–257. Elsevier, 1971.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *4th International Conference on Learning Representations, ICLR 2016*, 2016. URL http://arxiv.org/abs/1506.02438.

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

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems 35: NeurIPS 2022, New Orleans, LA, USA*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6fac9e316a4ae75ea244ddcef1982c71-Abstract-Conference.html.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025*, pp. 1279–1297. ACM, 2025. URL https://doi.org/10.1145/3689031.3696075.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, et al. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022. URL https://arxiv.org/abs/2211.14275.

Junda Wang, Zhichao Yang, Dongxu Zhang, Sanjit Singh Batra, and Robert E. Tillman. ESTAR: Early-stopping token-aware reasoning for efficient inference. *arXiv preprint arXiv:2602.10004*, 2026.

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

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. *arXiv preprint arXiv:2503.14476*, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. URL https://arxiv.org/abs/2507.18071.

A THEORETICAL ANALYSIS

We analyze three distinct objects: PPO's surrogate on a locally frozen, stopped process; an oracle that masks terms of a full-rollout gradient estimate; and practical ESPO, which also changes rewards and recomputes advantages. The analysis characterizes local surrogate consistency, conditional oracle variance reduction, and the gradient changes induced by practical truncation. Appendix A.7 summarizes the assumptions and scope.

Indexing and information. Tokens are indexed by $t = 1, \dots, T$, with $s_t = (x, y_{<t})$ and terminal state s_{T+1}. Thus T is both the final action index and the number of emitted tokens. The training-batch index is k, not t. Conditional on the configuration ω_k available before batch k, define

$$\mathcal{F}_t = \sigma(\omega_k, s_1, a_1, \dots, s_t), \qquad \mathcal{G}_t = \sigma(\mathcal{F}_t, a_t) = \mathcal{F}_{t+1}. \tag{14}$$

The equality uses deterministic prefix transitions, with terminal states retaining the emitted history. The score $u_t = \nabla_\theta \log \pi_\theta(a_t \mid s_t)$ and regret g_t are $\mathcal{G}_t$-measurable; the pre-action critic value is $\mathcal{F}_t$-measurable. We assume square-integrable gradient estimates and the usual support and differentiability conditions for likelihood-ratio differentiation. For gradient comparisons in Appendix A.3, we use the undiscounted task objective ($\gamma = 1$). For $\gamma < 1$, its discounted counterpart requires the corresponding γ^{t-1} weights in the gradient sums.

A.1 THE LOCALLY FROZEN ABSORBING PROCESS

Let $T_{\mathrm{nat}} \le T_{\max}$ be the final action index under the base task, including EOS or the task's finite horizon. Let $b_k \in \{0, 1\}$ indicate whether warmup has ended before batch k. Freeze the reference actor π_k, gate critic V_k, β_k, normalization statistics, and b_k throughout collection and all PPO epochs on that batch. Define

$$T_{\mathrm{stop}} = \min\Big(\{t < T_{\mathrm{nat}} : b_k = 1,\ z_t > \beta_k \max(V_k(s_t), \varepsilon)\} \cup \{T_{\mathrm{nat}}\}\Big). \tag{15}$$

Natural termination takes precedence over a trigger on the same token. A strict truncation replaces the transition after $a_{T_{\mathrm{stop}}}$ by an absorbing state with reward r_{fail}; natural completions keep their task reward. The gate's actor and critic are frozen copies: they are not differentiated when updating the actor or critic on the collected batch. A state augmented with time and the smoothing accumulator defines a finite-horizon MDP $\widetilde{\mathcal{M}}_k$ for this fixed configuration.

Lemma 1 (Non-anticipating termination)**.** *If normalization statistics and the gate configuration are fixed before collection, and $z_0 = 0$ is reset for each trajectory, then $\{T_{\mathrm{stop}} \le t\} \in \mathcal{G}_t$ for every t.*

Proof. Each g_i and its clipped normalization are $\mathcal{G}_i$-measurable. Unrolling the smoother gives

$$z_t = (1 - \alpha_{\mathrm{s}}) \sum_{i=1}^{t} \alpha_{\mathrm{s}}^{t-i} \tilde{g}_i. \tag{16}$$

Thus z_t and the trigger at t are $\mathcal{G}_t$-measurable. The union of triggers up to t, together with $\{T_{\mathrm{nat}} \le t\}$, is precisely $\{T_{\mathrm{stop}} \le t\}$. No token beyond t is needed. □

For $0 \le \alpha_{\mathrm{s}} < 1$, the weights in Equation (16) sum to $1 - \alpha_{\mathrm{s}}^t$, not one. Including the initial zero makes a convex combination, and $|z_t| \le c(1 - \alpha_{\mathrm{s}}^t)$. Historical batch statistics transfer distributional information for normalization, while the accumulator contains only the current trajectory's regrets.

A.2 PPO ON THE LOCALLY FROZEN PROCESS

Proposition 2 (Local surrogate consistency)**.** *For a fixed configuration ω_k, applying PPO and terminal-masked GAE to rollouts collected as above is PPO on $\widetilde{\mathcal{M}}_k$. The clipped surrogate is a well-defined optimization objective on those data. At the behavior parameters, the unclipped on-policy score estimator with true advantages is a policy-gradient estimator for this fixed MDP, with the objective's appropriate time weights.*

Proof. The frozen gate defines a non-anticipating transition and reward law independent of the actor parameters being optimized. The likelihood of a trajectory factors into the actor's action probabilities and these fixed transitions. Differentiating its expected return therefore yields the usual score-function gradient. An absorbing state has value zero, giving exactly the terminal-masked TD and GAE recursions in Section 3. Replacing true advantages with GAE and optimizing the clipped surrogate gives the standard PPO procedure on this process. □

The exact score identity uses true advantages, whereas practical PPO uses estimated advantages and a clipped surrogate. The gate is held fixed during these updates; changing ω_k between batches changes $\widetilde{\mathcal{M}}_k$. Thus the result concerns PPO's local formulation on the stopped process. Its original-objective and trust-region scope is summarized in Appendix A.7.

A.3 Oracle masking and gradient variance

Fix a policy in the base task and let $\mathcal{D}_0$ be an absorbing set of prefix states from which every reachable continuation has terminal reward zero. Define the first entry into this set by

$$d := \min\big(\{t \le T : s_t \in \mathcal{D}_0\} \cup \{T+1\}\big), \qquad m_t := \mathbf{1}\{d \le t \le T\}. \tag{17}$$

Unlike a definition that inspects future values along a realized path, d is a state-filtration stopping time. The action at $d-1$ that enters the doomed state is retained. Put $A_t = A^\pi(s_t, a_t)$, $e_t = \hat{A}_t - A_t$, and extend all sums to $T_{\max}$ by setting terms after termination to zero.

Assumption 1 (Zero post-failure signal)**.** On $\{m_t = 1\}$, $Q^\pi(s_t, a_t) = V^\pi(s_t) = 0$, hence $A_t = 0$. This holds for the absorbing zero-reward set just defined, not for every trajectory that happens to receive reward zero.

Assumption 2 (Conditional estimator accuracy)**.** $\mathbb{E}[e_t \mid \mathcal{G}_t] = 0$ at each valid action.

Remark 1 (Scope of the estimation assumption). Even for a fixed critic and deterministic prefix transitions, multi-step GAE usually depends on the random suffix; its error is not generally determined by (s_t, a_t). Assumption 2 holds with an exact value function under standard on-policy GAE conditions, but does not follow merely from marginally unbiased critic predictions. A learned shared critic can violate it. For one-step TD and deterministic transitions with a fixed critic, the error is already $\mathcal{G}_t$-measurable, so the assumption would require that error to vanish.

Define the full and oracle-masked estimators by

$$\hat{g}_{\text{full}} = \sum_{t=1}^{T} \hat{A}_t u_t, \qquad \hat{g}_{\text{orc}} = \sum_{t=1}^{T} (1 - m_t)\hat{A}_t u_t, \qquad N = \sum_{t=1}^{T} m_t e_t u_t. \tag{18}$$

The oracle keeps the original advantages; it neither changes the terminal reward nor recomputes GAE. The diagnostic label *ESPO/full* in Section 8 denotes a counterfactual length, not this gradient estimator.

Assumption 3 (Orthogonality of retained and removed estimation terms)**.** $\mathrm{Cov}(\hat{g}_{\text{orc}}, N) = 0$. When a diagonal covariance formula is used, assume additionally $\mathbb{E}[m_s m_t e_s e_t u_s u_t^\top] = 0$ for $s \neq t$. These are assumptions about the masked terms, including the random selection.

Proposition 3 (Conditional oracle variance reduction)**.** *Under Assumptions 1 and 2,*

$$\mathbb{E}[\hat{g}_{\text{orc}}] = \mathbb{E}[\hat{g}_{\text{full}}] = g^\star := \mathbb{E}\sum_{t=1}^{T} A_t u_t. \tag{19}$$

If also Assumption 3 holds, then

$$\mathrm{Cov}(\hat{g}_{\text{full}}) - \mathrm{Cov}(\hat{g}_{\text{orc}}) = \mathrm{Cov}(N) = \sum_{t=1}^{T_{\max}} \mathbb{E}[m_t e_t^2 u_t u_t^\top] \succeq 0. \tag{20}$$

The last equality uses the additional diagonal assumption. The trace reduction is strict if $\operatorname{tr}\mathrm{Cov}(N) > 0$.

Proof. Since m_t and u_t are $\mathcal{G}_t$-measurable, the tower rule gives $\mathbb{E}[m_t e_t u_t] = 0$ and $\mathbb{E}[(1 - m_t)e_t u_t] = 0$. By Assumption 1, $\hat{g}_{\text{full}} = \hat{g}_{\text{orc}} + N$ and $\sum_t (1 - m_t)A_t u_t = \sum_t A_t u_t$. This proves the means. Expanding the covariance of the sum and using the first orthogonality condition leaves $\text{Cov}(N)$. Expanding $NN^\top$ and using the second condition removes the off-diagonal terms. Each remaining matrix is positive semidefinite. □

Remark 2 (Correlated errors and GAE). Without retained–removed orthogonality the exact identity is

$$\text{Cov}(\hat{g}_{\text{full}}) - \text{Cov}(\hat{g}_{\text{orc}}) = \text{Cov}(N) + \text{Cov}(\hat{g}_{\text{orc}}, N) + \text{Cov}(\hat{g}_{\text{orc}}, N)^\top. \tag{21}$$

Writing $\sigma_Z^2 = \operatorname{tr} \text{Cov}(Z)$, its trace is at least $\sigma_N^2 - 2\sigma_{\text{orc}}\sigma_N$, which need not be positive. Even at $\lambda_{\text{GAE}} = 0$, adjacent TD errors share a critic value: for independent state errors ξ_t, the critic-noise terms $\gamma\xi_{t+1} - \xi_t$ and $\gamma\xi_{t+2} - \xi_{t+1}$ have covariance $-\gamma \text{Var}(\xi_{t+1})$. GAE introduces further overlap. No correction depending only on $\gamma\lambda_{\text{GAE}}$ bounds all these correlations without extra moment and dependence assumptions. Masking is not automatically a Rao–Blackwellization, and correlated errors can reverse the variance comparison.

Corollary 4 (Approximate irrecoverability with second moments)**.** *For any adapted binary removal mask m_t satisfying $|A_t| \le \eta$ on removed terms, put $D = \sum_t m_t A_t u_t$, $N = \sum_t m_t e_t u_t$, and*

$$S = \sum_t m_t \|u_t\|, \qquad U_1 = \mathbb{E}S, \qquad U_2 = (\mathbb{E}S^2)^{1/2} < \infty.$$

Under Assumption 2 and $\text{Cov}(\hat{g}_{\text{orc}}, N) = 0$,

$$\begin{aligned} \|\mathbb{E}\hat{g}_{\text{orc}} - g^\star\| &\le \eta U_1, \\ \operatorname{tr} \text{Cov}(\hat{g}_{\text{full}}) - \operatorname{tr} \text{Cov}(\hat{g}_{\text{orc}}) &\ge \sigma_N^2 - 2\eta U_2(\sigma_N + \sigma_{\text{orc}}). \end{aligned} \tag{22}$$

Oracle masking strictly reduces mean-squared error relative to $g^\star$ if $\sigma_N^2 > 2\eta U_2(\sigma_N + \sigma_{\text{orc}}) + \eta^2 U_1^2$.

Proof. Conditional unbiasedness gives $\mathbb{E}N = 0$ and the full estimator's mean $g^\star$. Thus the bias has norm at most $\mathbb{E}\|D\| \le \eta U_1$. Also $\sigma_D \le (\mathbb{E}\|D\|^2)^{1/2} \le \eta U_2$. Expanding the variance of $\hat{g}_{\text{orc}} + D + N$, discarding $\sigma_D^2 \ge 0$, and bounding the two remaining cross-covariance traces by Cauchy–Schwarz yields the displayed lower bound. The retained–noise cross term is zero by assumption. Adding squared bias gives the MSE criterion. The second moment U_2 cannot be replaced by U_1: Jensen's inequality has $U_1^2 \le U_2^2$, the opposite direction to that replacement. □

Corollary 5 (Practical truncation changes the entire retained GAE prefix)**.** *Couple a full trajectory with a stopped one using the same sampled prefix and fixed critic. Let $I = \mathbf{1}\{T_{\text{stop}} < T_{\text{nat}}\}$, $K = T_{\text{stop}}$, $q = \gamma\lambda_{\text{GAE}}$, and, on $I = 1$, define*

$$d_K = r_{\text{fail}} - V_\phi(s_K) - \hat{A}_K^{\text{full}}, \qquad H_q(K) = \sum_{t=1}^{K} q^{K-t}\|u_t\|.$$

Using raw GAE advantages before any batch normalization or PPO clipping,

$$\hat{g}_{\text{ESPO}} - \hat{g}_{\text{full}} = I\left(d_K \sum_{t=1}^{K} q^{K-t} u_t - \sum_{t=K+1}^{T_{\text{nat}}} \hat{A}_t^{\text{full}} u_t\right). \tag{23}$$

Under Assumption 2, and $|A_t| \le A_{\max}$, this implies

$$\|\mathbb{E}\hat{g}_{\text{ESPO}} - g^\star\| \le \mathbb{E}[I|d_K|H_q(K)] + \rho_{\text{rec}} G_{\text{rec}}, \tag{24}$$

where $\rho_{\text{rec}} = \Pr(I = 1, s_{K+1} \notin \mathcal{D}_0)$ and $G_{\text{rec}} = A_{\max}\mathbb{E}[\sum_{t>K} \|u_t\| \mid I = 1, s_{K+1} \notin \mathcal{D}_0]$; set the product to zero if this event has probability zero.

Proof. At K, the stopped advantage is $r_{\text{fail}} - V_\phi(s_K)$; hence its difference from the full advantage is d_K. For $t < K$, both trajectories have the same TD residual, so backward induction in the GAE recursion gives $\hat{A}_t^{\text{ESPO}} - \hat{A}_t^{\text{full}} = q^{K-t} d_K$. Subtracting their gradient sums proves Equation (23). For each removed term, $I\mathbf{1}\{K < t\}$ is $\mathcal{F}_t$-measurable, so Assumption 2 makes its estimation-error contribution zero in expectation. If $s_{K+1} \in \mathcal{D}_0$, all remaining true advantages vanish. Otherwise bound their total norm by $A_{\max}\sum_{t>K}\|u_t\|$. Triangle inequalities give Equation (24), without conditioning an unconditional unbiasedness statement on a future success event. □

The first term is a reward-and-GAE perturbation, not just a terminal critic calibration error. If $q = 1$, it reaches all retained tokens with equal GAE weight; for $q < 1$ it decays geometrically. An oracle stop with $r_{\text{fail}} = -1$ still changes an originally zero-reward objective. The measured correct-yet-truncated rate is instead

$$\rho_{\text{cyt}} = \mathbb{E}[IY] = \mathbb{E}[Ip_K^+], \qquad p_K^+ = \Pr(Y = 1 \mid \mathcal{G}_K), \tag{25}$$

where Y is correctness under natural continuation. In general $\rho_{\text{cyt}} \le \rho_{\text{rec}}$, so it cannot replace ρ_{rec} in the bias upper bound. A recoverable prefix can still produce an incorrect sampled suffix. Without Assumption 2, add the full estimator's bias and the expected removed estimation-error term to the bound. These comparisons do not establish variance reduction for practical ESPO.

A.4 Structural assumptions for the regret–value ordering

Lemma 6 (Mode-versus-sample log-likelihood ratio)**.** *For $a_t^\star \in \arg\max_a \pi(a \mid s_t)$,*

$$g_t = \log \frac{\pi(a_t^\star \mid s_t)}{\pi(a_t \mid s_t)} \ge 0, \qquad \sum_{i=1}^{t} g_i = \log \frac{M_t}{\pi(y_{1:t} \mid x)}, \quad M_t = \prod_{i=1}^{t} \pi(a_i^\star \mid s_i). \tag{26}$$

Proof. The first identity is the definition of g_t; multiplying the stepwise ratios and using autoregressive factorization proves the second. □

M_t is a modal envelope along the realized path, not the likelihood of a greedy rollout, which would generally visit different prefixes. The identity refers to the distribution whose log probabilities define g_t. It says nothing by itself about correctness: a low-probability token may be necessary for a correct answer.

Assumption 4 (Post-action calibration and sufficient statistics)**.** At a nonterminal decision point after token t, the natural-continuation success probability satisfies $\Pr(Y = 1 \mid \mathcal{G}_t) = p_t(z_t, v_t)$, where $v_t = V_k(s_t)$. On the attainable state space, p_t is non-increasing in z_t at fixed v_t and non-decreasing in v_t at fixed z_t.

This is a structural assumption on joint sufficiency and monotonicity for the stopping analysis. With binary rewards and $\gamma = 1$, $V^\pi(s_t) = \Pr(Y = 1 \mid \mathcal{F}_t)$ is a *pre-action* quantity, whereas the relevant post-action probability is $Q^\pi(s_t, a_t)$. Assumption 4 is therefore specified separately from calibration of the pre-action critic. Its following direct consequence defines the product order used in the finite-horizon stopping analysis.

Proposition 7 (Product-order consequence of calibration)**.** *Under Assumption 4, p_t is non-increasing in the order*

$$(z_1, v_1) \preceq (z_2, v_2) \quad \Longleftrightarrow \quad z_1 \le z_2 \text{ and } v_1 \ge v_2, \tag{27}$$

provided the intermediate points used for the two coordinate comparisons are in the assumed domain (in particular, on a rectangular domain).

Proof. Increasing z cannot increase p_t, and then decreasing v cannot increase it either, by the two directions explicitly assumed in Assumption 4. □

This consequence supplies the ordering for Appendix A.5; it is a structural implication of Assumption 4. The performance of the learned regret–value rule is assessed empirically in Section 6 and Appendix G.

A.5 Finite-horizon optimal stopping

Consider the fixed-policy stopping problem where a discarded rollout has accuracy zero, independently of ESPO's negative *training* reward. For a stopping rule $\sigma \le T_{\text{nat}}$, let

$$A(\sigma) = \mathbb{E}[Y\mathbf{1}\{\sigma = T_{\text{nat}}\}], \qquad C(\sigma) = \mathbb{E}[\sigma], \qquad J_\kappa(\sigma) = A(\sigma) - \kappa C(\sigma),$$

where $\kappa \ge 0$ is a token price, distinct from the GAE parameter. Solving J_κ provides an optimal rule at its own induced cost. We do not claim that every prescribed budget is attained by a deterministic rule or that this Lagrangian is equivalent to every constrained problem without randomization and additional duality conditions.

Assumption 5 (Markov and monotone continuation kernel)**.** At each nonterminal post-token decision point, $X_t = (z_t, v_t)$ together with t is a sufficient Markov state for the next-token transition, completion event, and reward. For every bounded non-increasing function f in the order Equation (27),

$$x \longmapsto \mathbb{E}[\mathbf{1}\{\text{completes next}\}Y + \mathbf{1}\{\text{continues}\}f(X_{t+1}) \mid X_t = x]$$

is non-increasing. The state domains are rectangular and the continuation values below are continuous in z at fixed v. At the horizon the next transition completes the task and receives its prescribed task reward.

Markov sufficiency and kernel monotonicity are additional idealizations; they do not follow from success-probability calibration alone.

Theorem 8 (Time-dependent monotone boundary)**.** *Under Assumption 5, an optimal rule for J_κ stops at nonterminal decision points in sets*

$$\mathcal{S}_t^\star = \{(z, v) : z \geq h_t^\star(v)\}, \qquad \sigma^\star = \min\big(\{t < T_{\text{nat}} : X_t \in \mathcal{S}_t^\star\} \cup \{T_{\text{nat}}\}\big), \tag{28}$$

where each $h_t^\star$ is non-decreasing. Extended thresholds represent empty or full sections. Without boundary continuity the sections are still upper intervals, but may be open at the threshold.

Proof. Ignore already incurred token costs. Set $W_{T_{\max}} = 0$ after terminal payment, and at nonterminal decision points define

$$\begin{aligned} Q_t^{\text{cont}}(x) &= -\kappa + \mathbb{E}[\mathbf{1}\{\text{completes next}\}Y + \mathbf{1}\{\text{continues}\}W_{t+1}(X_{t+1}) \mid X_t = x], \\ W_t(x) &= \max\{0, Q_t^{\text{cont}}(x)\}. \end{aligned} \tag{29}$$

The terminal payoff is on the completing transition, so no horizon reward is lost. Backward induction and kernel monotonicity imply both Q_t^{cont} and W_t are non-increasing in the order. Choosing stop on ties gives $\mathcal{S}_t^\star = \{x : Q_t^{\text{cont}}(x) \leq 0\}$, an upper set. Its section at fixed v is an upper interval; continuity includes a finite boundary point. For $v_1 < v_2$, its section at v_2 is contained in that at v_1, so their infima satisfy $h_t^\star(v_1) \leq h_t^\star(v_2)$. The argument gives a separate boundary at each t, not a stationary limiting boundary for a finite horizon. □

ESPO's boundary $h_\beta(v) = \beta \max(v, \varepsilon)$, for $\beta \geq 0$, has the same monotone direction. It is a time-independent, one-parameter subfamily; the theorem does not show that this subfamily contains an optimal boundary. Regret-only and value-only rules are other restricted upper sets.

Proposition 9 (Decision regret, not geometric boundary error)**.** *For any candidate stopping policy μ in the fixed stopping problem, its Lagrangian deficit is the expected sum, along its own visited decision states, of the Bellman gaps $W_t(X_t) - Q_t(X_t, a_t^\mu)$, where $Q_t(x, \text{stop}) = 0$ and $Q_t(x, \text{continue}) = Q_t^{\text{cont}}(x)$. The deficit is nonnegative. At equal expected cost, it equals the accuracy deficit relative to an optimal rule.*

Proof. Subtract the chosen action's reward plus optimal next-state value from W_t. Summing over the candidate's visited states telescopes the intermediate values; terminal values are zero. Taking expectations leaves optimal value minus the candidate's return. Each Bellman gap is nonnegative by maximization. Equal costs cancel the κC terms. □

Consequently, disagreement with an optimal boundary incurs strict loss only if it visits states with a positive Bellman gap with positive probability. Ties and unreachable regions can yield different boundaries with equal return. A nonconstant optimum need not make every constant rule strictly worse, and closeness in unweighted geometric distance does not rank the families. The relative quality of ESPO and the single-signal rules remains empirical.

Proposition 10 (Random versus informed selection on fixed candidates)**.** *Fix a common candidate-prefix distribution at a single decision stage, and let $p = \Pr(Y = 1 \mid \text{observed prefix})$. Among measurable truncation probabilities $q(X) \in [0, 1]$ with $\mathbb{E}q = m$, the accuracy loss $\mathbb{E}[pq]$ is minimized by selecting the lowest p values (randomizing at a quantile tie) and maximized by selecting the highest. Independent selection $q = m$ has loss $m\mathbb{E}p$, between these two extrema. For $0 < m < 1$ and nonconstant p, the minimum is strictly smaller than the independent-selection loss.*

Proof. Select c so that $q^\star = 1$ for $p < c$, $q^\star = 0$ for $p > c$, and randomize at $p = c$ to obtain mass m. Pointwise, $(p - c)(q - q^\star) \geq 0$, so $\mathbb{E}[p(q - q^\star)] \geq c\mathbb{E}[q - q^\star] = 0$. Reversing the selection gives the maximum. Independent selection is feasible; when p is nonconstant and $0 < m < 1$, moving positive selection mass from higher to lower p strictly improves its objective. □

This is a fixed-population, equal-truncation-mass statement. Equal mass need not mean equal saved tokens when suffix lengths differ, and sequential rules change the population they reach. Thus it is not a general Pareto-dominance result at matched compute and does not by itself compare a learned ESPO rule with arbitrary random stopping.

A.6 Stability of the stop-rate controller

Assumption 6 (Stationary response and conditionally centered noise)**.** For a fixed actor, critic, and normalization, the mean response $\rho(\beta)$ on $[\beta_{\min}, \beta_{\max}]$ is differentiable with $-L \leq \rho'(\beta) \leq -\ell < 0$. The target satisfies $\rho(\beta_{\max}) < \tau < \rho(\beta_{\min})$. For stochastic observations, $\hat{\rho}_k = \rho(\beta_k) + \xi_k$, where with respect to the training history $\mathcal{H}_k$, $\mathbb{E}[\xi_k \mid \mathcal{H}_k] = 0$ and $\mathbb{E}[\xi_k^2 \mid \mathcal{H}_k] \leq s^2$.

Proposition 11 (Projected controller bounds)**.** *Under Assumption 6, there is a unique interior root $\rho(\beta^\star) = \tau$. For a constant step size $0 < \eta < 2/L$, the deterministic projected update is a contraction with $q_c = \max\{|1 - \eta\ell|, |1 - \eta L|\} < 1$:*

$$|\beta_k - \beta^\star| \leq q_c^k |\beta_0 - \beta^\star|, \qquad |\rho(\beta_k) - \tau| \leq L q_c^k |\beta_0 - \beta^\star|. \tag{30}$$

With the stated noise,

$$\mathbb{E}|\beta_k - \beta^\star|^2 \leq q_c^{2k}\mathbb{E}|\beta_0 - \beta^\star|^2 + \frac{\eta^2 s^2 (1 - q_c^{2k})}{1 - q_c^2}. \tag{31}$$

For $\eta \leq 1/L$, the limiting upper bound is at most $\eta s^2/\ell$, and the mean-square rate error is at most L^2 times that quantity. For deterministic positive step sizes with $\sum_k \eta_k = \infty$ and $\sum_k \eta_k^2 < \infty$, $\beta_k \to \beta^\star$ almost surely.

Proof. Strict monotonicity and the strict endpoint inequalities give the root. By the mean value theorem, the unprojected deterministic error equals $(1 + \eta\rho'(\zeta_k))(\beta_k - \beta^\star)$, whose factor has absolute value at most q_c. Projection onto the interval is nonexpansive and fixes $\beta^\star$. This proves the first bound; Lipschitzness gives the rate bound. Before projection, conditional centering eliminates the noise cross term:

$$\mathbb{E}[|\beta_{k+1} - \beta^\star|^2 \mid \mathcal{H}_k] \leq q_c^2 |\beta_k - \beta^\star|^2 + \eta^2 s^2.$$

Iterating gives Equation (31). If $\eta \leq 1/L$, then $q_c = 1 - \eta\ell$ and $1 - q_c^2 \geq \eta\ell$. For the stated diminishing step sizes, let $e_k = \beta_k - \beta^\star$. The same argument gives

$$\mathbb{E}[e_{k+1}^2 \mid \mathcal{H}_k] \leq (1 + L^2\eta_k^2) e_k^2 - 2\ell\eta_k e_k^2 + s^2\eta_k^2.$$

The almost-supermartingale convergence lemma (Robbins & Siegmund, 1971) implies that e_k^2 converges and $\sum_k \eta_k e_k^2 < \infty$ almost surely. Since $\sum_k \eta_k = \infty$, the limit must be zero. □

Remark 3 (Limits of the controller model). For frozen policies, a higher threshold cannot cause an earlier first crossing on a coupled full path, so the stop rate is non-increasing. Strict smoothness and a slope bounded away from zero are stronger assumptions, often violated by finite-sample or discrete responses. A weak slope permits slow contraction. An unreachable target saturates the projected controller at an endpoint; an endpoint target need not have an interior root. During training, the actor, critic, and normalization drift, so the theorem is not a convergence guarantee for the coupled RL system. Without estimating L, the default step size $\eta_\beta = 0.1$ cannot be certified stable by this theorem. A time-averaged stop rate near the target does not prove that the final rate has converged.

A.7 Summary of assumptions and scope

- Lemma 1 and Proposition 2 require a causal frozen gate and terminal handling as specified by the algorithm. They establish a local surrogate interpretation, not an original-objective or trust-region guarantee.

- Oracle unbiasedness needs an adapted mask, zero removed true advantages, and conditional estimator accuracy. Covariance reduction further needs retained–removed orthogonality; its diagonal form needs pairwise orthogonality. Neither follows just from using GAE or from critic warmup.
- Approximate irrecoverability requires second-moment score control. Practical ESPO changes all retained GAE advantages and has separate reward and selection bias; measured correct-yet-truncated frequency is not the probability of cutting a recoverable prefix.
- Monotone finite-horizon stopping needs a sufficient Markov state and a monotone transition kernel. Its boundary can depend on time, and it need not belong to ESPO's parametric family. Fixed-mass selection does not imply matched-token Pareto dominance.
- Controller convergence requires a stationary response, a reachable target, a suitable step size, and conditionally centered noise. The learning system need not satisfy these conditions globally.

The theory therefore supplies conditional mechanisms and explicit limitations; the empirical comparisons, rather than an unconditional theorem, establish the observed performance of the practical method.

B Training Details

For the ESPO training hyperparameters, the learning rate is set to 1×10^{-6}. The maximum rollout length (T_{max}) is configured to 8192 tokens, with 64 prompts per global batch and 8 rollouts per prompt (512 trajectories). Regarding the algorithmic specifics, the failure reward r_{fail} is set to -1.0, the EMA α_{ema} to 0.99, and the temporal smoothing coefficient α_{s} to 0.9. The initial value of β is set to 7.0. A β adjustment rate of 0.1 is applied to maintain a target stop rate of 25%. The baseline methods maintain the same settings, such as the global batch size. Additionally, for DAPO, the clip_ratio_low is set to 0.2 and the clip_ratio_high is set to 0.28 as stated in (Yu et al., 2025).

For evaluation, let N be the number of problems and n the number of samples per problem. We report mean per-sample accuracy:

$$\text{avg@}n = \frac{100}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} c_{ij},$$

where $c_{ij} = 1$ if sample j for problem i is correct, and $c_{ij} = 0$ otherwise. This estimates pass@1, not pass@n (success in at least one of n attempts).

We use $n = 32$ on AMC23, AIME24, and MATH500, and $n = 8$ on GSM8K, Minerva-Math, OlympiadBench, LiveCodeBench, and HumanEval+. The stopping rule is disabled at evaluation. Inference uses temperature 1.0, top-$p = 0.7$. For Tables 1 and 2, accuracy means and differences are computed before display rounding. Thus, averaging or subtracting the displayed cells can give a different final digit.

Table 4: ESPO hyperparameters.

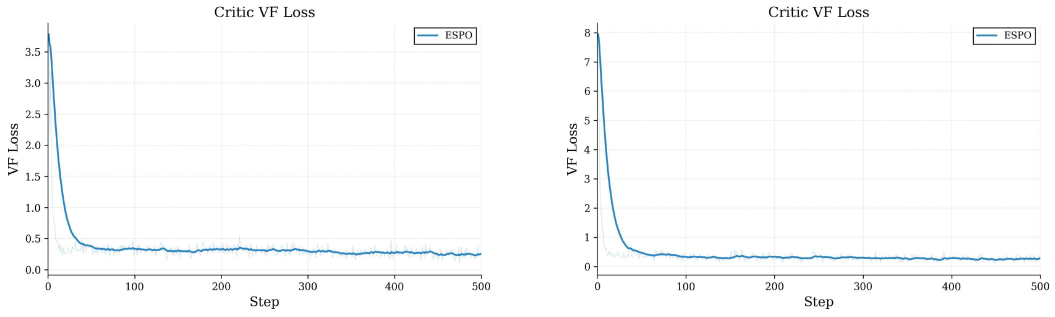
Hyperparameter	**Value**
Failure reward r_{fail}	-1.0
Batch EMA α_{ema}	0.99
Temporal smoothing α_{s}	0.9
Initial β	7.0
Minimum β_{min}	0.0
β adjustment rate	0.1
Target stop rate τ	0.25
Max rollout length T_{max}	8192
Learning rate	1×10^{-6}
Value floor ε	0.2

C TOKEN ACCOUNTING

Methods in this comparison differ in *which* generated tokens ultimately contribute to a gradient: ESPO ends generation at the detected failure step, so the post-stop suffix is neither decoded nor backpropagated, while DAPO's dynamic sampling decodes groups that are discarded when their rewards are degenerate. We therefore report **Tokens$_{\text{gen}}$**, the tokens actually decoded, as a measure of generation workload that is directly comparable across all five methods. It does not capture differences in critic computation, optimization, or system overhead; Appendix I therefore complements it with the total wall-clock time per training update. For ESPO decoded and trained-on tokens coincide, since a token it does not decode is also a token it cannot train on; the distinction matters only for DAPO, whose discarded groups are decoded but never trained on. All **Avg Tokens** entries are rounded to the nearest integer; token totals are rounded independently.

D ADAPTIVE CRITIC WARMUP DETAILS

During warmup, the stopping criterion is disabled, β is held at its initial value without controller updates, the actor uses the base PPO surrogate, and the critic is fitted with its value loss. Let L_k be the critic loss after training step k. Warmup ends if $L_k < 0.5$, or if $|L_j - L_{j-1}| < 0.03$ for each of three consecutive comparisons $j = k-2, k-1, k$. The second test is used only when all four losses are available. Otherwise, warmup ends after $\lceil 0.1 K_{\text{train}} \rceil$ training steps, where K_{train} is the total budget. Activation takes effect in the next rollout batch, not partway through a trajectory. These thresholds are practical stability criteria, not guarantees of critic convergence or calibration.



(a) Critic loss on DeepSeek-R1-Distill-Qwen-1.5B (b) Critic loss on DeepSeek-R1-Distill-Qwen-7B

Figure 3: The left figure illustrates the critic loss for DeepSeek-R1-Distill-Qwen-1.5B, while the right figure depicts the critic loss for DeepSeek-R1-Distill-Qwen-7B.

E DIFFICULTY-STRATIFIED RESULTS

Table 5 examines how ESPO's accuracy gains and natural generation lengths vary with problem difficulty on the 7B math run. Evaluation uses the trained policies with stopping disabled, separating their response-length behaviour from the direct token savings of early termination during training.

ESPO improves the measured accuracy in all four difficulty buckets. The gains are larger in the Hard and Very-hard buckets ($+1.31$ and $+1.56$ points) than in the Easy and Medium buckets ($+0.72$ and $+0.43$ points), showing that the observed benefit extends to harder problems. Meanwhile, evaluation-time token usage remains close to PPO: ESPO uses 1.0% and 0.9% more tokens in the Medium and Hard buckets, and 0.8% and 0.2% fewer in the Easy and Very-hard buckets. These small, mixed-sign length differences are consistent with the training-dynamics analysis in Section 8: rollout savings primarily arise from the stopping mechanism rather than a systematic shift toward intrinsically shorter responses. The table thus complements the training-cost results with positive accuracy differences across difficulty levels and limited changes in natural generation length.

Table 5: Difficulty-stratified accuracy and evaluation-time token usage, DeepSeek-R1-Distill-Qwen-7B, 250 **steps** — a shorter budget than the 500-step protocol of Table 1. Token savings are measured during evaluation with early stopping disabled; positive values indicate fewer generated tokens than PPO, while negative values indicate more tokens in that bucket.

Difficulty bucket	**PPO Acc**	**ESPO Acc**	Δ	**Eval. Token Saving vs. PPO**
Easy (MATH500 L1–2)	89.12	89.84	+0.72	+0.8%
Medium (MATH500 L3–4)	88.99	89.42	+0.43	−1.0%
Hard (MATH500 L5 / AMC23)	77.10	78.41	+1.31	−0.9%
Very hard (AIME24)	41.98	43.54	+1.56	+0.2%

F Mechanism Diagnostics

Table 6 characterizes the frequency and selectivity of ESPO's stopping decisions across the four runs. The stop rate measures how often the adaptive β controller triggers strict truncation; the correct-yet-truncated rate measures the fraction of all rollouts whose natural continuations would have succeeded despite being selected for stopping. Both are measured with the instrumented configuration of Section 8, which continues decoding beyond the first trigger and scores the completed response. This diagnostic incurs additional suffix generation that the deployed stopping rule avoids.

Table 6: Realized stopping behaviour per run, all at the 500-**step** budget of Table 1. **Stop rate** is the mean percentage of trajectories truncated per step, against a target of 25%; **correct-yet-truncated** is the proportion of all trajectories per step that were truncated yet would have been correct. Its denominator is all rollouts, not only correct natural completions.

Run	**Stop rate**	**Correct-yet-truncated**
DeepSeek-R1-Distill-Qwen-1.5B, math	24.95%	2.70%
DeepSeek-R1-Distill-Qwen-7B, math	25.65%	2.91%
DeepSeek-R1-Distill-Qwen-7B, code	24.65%	2.98%
Qwen3-8B, math	23.85%	3.08%

The time-averaged stop rates range from 23.85% to 25.65%, within 5% relative deviation from the target 25%, showing that stopping is active at a comparable average frequency across runs. Correct-yet-truncated trajectories account for only 2.70%–3.08% of all rollouts. Together, the two columns indicate that most selected trajectories have unsuccessful natural continuations, supporting the selectivity of the regret–value criterion. The code result, 2.98%, lies within the math range, extending this pattern beyond mathematical reasoning.

These are full-run averages; the stationary controller analysis and its conditions are given in Appendix A.6. The correct-yet-truncated denominator is all rollouts, not only correct natural completions or stopped trajectories. It measures lost successful sampled continuations, whereas the recoverable-prefix probability in Equation (24) concerns whether success remains possible. The two quantities are distinct.

G Accuracy–Token Operating Points

Table 7 complements the component ablation in Section 6 by comparing accuracy and generation cost jointly. It places the stopping rules on the accuracy–token plane and varies ESPO's target stop rate τ to characterize the trade-off between solution accuracy and rollout length.

Among the reported configurations, ESPO at $\tau = 0.35$ achieves both higher accuracy and lower token usage than all three ablated rules: 49.75 at 4247 tokens, compared with 47.85 at 4262 for value-only, 47.52 at 4454 for random stopping, and 47.92 at 4685 for regret-only. The default $\tau = 0.25$ setting attains the highest accuracy, 50.65, exceeding the best ablated rule by 2.73 points while using fewer tokens than that rule (4488 versus 4685). Thus the measured benefit of combining regret and value is visible in both solution accuracy and generation cost.

Table 7: Accuracy–token operating points on the six-benchmark math average (DeepSeek-R1-Distill-Qwen-1.5B), 250 **steps**. **Avg Acc** is the unweighted mean over the same six benchmarks as the **Avg** column of Table 1; the three ESPO rows are the τ block of Table 8, the same runs at the same budget. Rows are ordered by **Avg Tokens**, the mean length over all training rollouts.

Rule	Avg Acc	Avg Tokens
ESPO joint ($\tau = 0.35$)	49.75	**4247**
Value-only	47.85	4262
Random stop	47.52	4454
ESPO joint ($\tau = 0.25$, default)	**50.65**	4488
ESPO joint ($\tau = 0.15$)	50.05	4573
Regret-only	47.92	4685

The joint rule also outperforms the three ablated rules in Table 2, which uses the 7B backbone, 500 steps, and AIME24 accuracy. The two settings provide complementary evidence for selective stopping across model scales and evaluation protocols.

Within ESPO, raising τ from the default 0.25 to 0.35 reduces average rollout length by 241 tokens (5.4%), with a 0.90-point decrease in accuracy. Lowering τ to 0.15 increases length to 4573 tokens while scoring 50.05, below the default. These measured operating points show how stopping aggressiveness changes the accuracy–cost trade-off: the default gives the highest observed accuracy, while $\tau = 0.35$ gives the lowest generation cost in this comparison. The ESPO runs also form the target-rate sweep in Appendix H.

H Hyperparameter Sensitivity

We sweep four of the additional ESPO hyperparameters: the target stop rate τ, the β adjustment rate η_β, the normalization EMA coefficient α_{ema}, and the temporal smoothing coefficient α_{s}. Table 8 sweeps each in turn on the 1.5B backbone, holding the others at their defaults. The defaults were selected through hyperparameter sweeps; this analysis characterizes accuracy and cost around that selected configuration.

Table 8: Hyperparameter sweeps on the six-benchmark math average (DeepSeek-R1-Distill-Qwen-1.5B), 250 **steps**. **Avg Acc** is the unweighted mean over the same six benchmarks as the **Avg** column of Table 1, evaluated under identical settings; **Avg Tokens** is the mean length over all training rollouts, not only truncated ones. The τ block below is the same set of runs as the three ESPO rows of Table 7. Defaults marked *. Each block varies one parameter with the others held fixed. Empirical stop rates (τ block) and their standard deviations (η_β block) are quoted in the text. **Tokens$_{\text{gen}}$(M)** counts decoded tokens as in Table 1.

Parameter	Value	Avg Acc	Tokens$_{\text{gen}}$(M)	Avg Tokens
Target rate τ	0.15	50.05	585.3	4573
	0.25*	**50.65**	574.5	4488
	0.35	49.75	543.6	4247
Adjust rate η_β	0.05	49.98	602.4	4706
	0.1*	**50.65**	574.5	4488
	0.2	49.45	547.3	4276
Batch EMA α_{ema}	0.95	49.55	592.6	4630
	0.99*	**50.65**	574.5	4488
	0.999	50.18	560.8	4381
Temporal smoothing α_{s}	0.8	48.88	448.3	3502
	0.9*	**50.65**	574.5	4488
	0.95	49.13	625.2	4884

The token columns show systematic changes across the sweep. Raising τ from 0.15 to 0.35 increases the empirical stop rate (11.8%, 17.0%, 22.1%) and shortens rollouts (4573, 4488, 4247). Increasing η_β from 0.05 to 0.2 also lowers average rollout length, from 4706 to 4276 tokens; the observed stop-rate standard deviation falls from 6.1 through 4.9 to 3.2 percentage points. These values describe the finite training runs; the stationary controller bounds are analyzed separately in Appendix A.6.

Among the four parameters in the tested ranges, temporal smoothing has the largest effect on generation cost. Increasing α_{s} from 0.8 to 0.95 changes average rollout length from 3502 to 4884 tokens, a 39% increase. By comparison, the batch-EMA sweep spans 4381–4630 tokens. Temporal smoothing therefore provides the strongest cost adjustment among these components of the stopping rule.

The default configuration attains the highest measured accuracy in all four sweeps, 50.65, compared with 48.88–50.18 for the alternatives. The full accuracy range is 1.77 points, and all tested settings exceed the base model's 46.93 (Table 1). Across these settings, generation cost varies substantially while accuracy remains within this narrower observed range. This characterizes the accuracy–cost sensitivity around the selected defaults, with temporal smoothing producing the largest cost variation.

I Wall-Clock Cost

Table 9 complements the decoded-token counts in Table 1 with end-to-end training cost on the 7B math runs. It reports the total time per completed update, capturing generation, value-network computation, optimization, and system overhead.

Table 9: Mean total wall-clock time per completed training update on DeepSeek-R1-Distill-Qwen-7B math runs, averaged over all 500 training steps, including warmup where applicable. Timing covers the full per-update training workflow, including all generation batches drawn by DAPO's dynamic sampling.

Method	**Wall-clock/ Step(s)**
GSPO	274.5
GRPO	288.1
PPO	324.3
ESPO (Ours)	**249.1**
DAPO	376.7

Averaged over the entire training run, ESPO completes one training update in 249.1 s against PPO's 324.3 s, a 23.2% reduction, and has the lowest mean update time of the five methods, including DAPO. It is also faster than critic-free GRPO and GSPO (288.1 s and 274.5 s) despite requiring value-network computation.

The mean rollout length is 3278 decoded tokens for ESPO against PPO's 4189 (Table 1). The corresponding reductions are 21.7% in decoded tokens and 23.2% in total update time, showing that lower generation workload is accompanied by end-to-end training savings. The larger wall-clock reduction is consistent with the sequence-length dependence of training cost. Early stopping removes late decoding steps with longer attention contexts and shortens the sequences used for critic evaluation and PPO updates. Attention work depends on both token count and context length, allowing time savings to exceed the proportional reduction in decoded tokens.

Timing includes critic warmup and every subsequent update over the full 500-step run, including all dynamic-sampling batches required by DAPO. ESPO uses the deployed rule, which ends decoding at the stopping point. The additional suffix decoding used for counterfactual diagnostics in Section 8 is separate from this timing protocol.

J Entropy Dynamics under Early Stopping

Measuring the counterfactual length. Measuring what ESPO *would* have generated requires a trajectory it did not generate, so the runs behind Figure 2 use an instrumented variant of the rule: we

compute the smoothed stopping statistic as usual but continue decoding after the first strict trigger. Only the post-stop suffix is masked; the trigger token is retained. The same failure reward, zero terminal bootstrap, and retained-prefix GAE as in the deployed rule are used. For equivalence, diagnostic suffix tokens must also be excluded from the stopping-statistic updates for later batches. Under these conditions the retained training data match the deployed rule; the extra decoding buys the counterfactual length and the identity of the truncated-but-correct sequences, and is what makes Figure 2a,c and Table 6 measurable at all.

Two effects on policy updates. ESPO affects entropy through the policy updates induced by stopping. Relative to full-rollout training, truncation removes suffix updates and changes the advantages of the retained prefix through the terminal penalty. These two effects appear separately in Equation (23) before advantage normalization and PPO clipping. The entropy dynamics in Figure 2b depend on the balance of these two effects.

How far the penalty propagates. At a strict stop $K = T_{\text{stop}}$, the final TD residual is $\delta_\dagger = r_{\text{fail}} - V_\phi(s_K)$. It is negative if $V_\phi(s_K) > r_{\text{fail}}$; a learned critic trained with negative failure rewards need not be nonnegative. The residual contributes to each retained advantage:

$$\hat{A}_t = \sum_{j=t}^{K-1} (\gamma\lambda_{\text{GAE}})^{j-t}\delta_j + (\gamma\lambda_{\text{GAE}})^{K-t}\delta_\dagger, \quad t \le K. \tag{32}$$

For $\gamma\lambda_{\text{GAE}} < 1$, this contribution decays geometrically. For $\gamma\lambda_{\text{GAE}} = 1$, it reaches the entire retained prefix without decay. The same GAE weights apply to full-horizon PPO, so no unconditional short-window localization follows from using an absorbing penalty. Equation (23) describes the exact change relative to full GAE.

A local entropy calculation. Consider a single state with independently parameterized softmax logits ℓ and probabilities p_b. For an isolated negative score step $\Delta\ell = -\eta\nabla_\ell \log p_a$, $\eta > 0$, let $H = -\sum_b p_b \log p_b$. Since $\partial H/\partial \ell_b = -p_b(\log p_b + H)$,

$$\Delta H = \eta[\psi(a) - \bar{\psi}] + O(\eta^2), \quad \psi(b) = p_b(\log p_b + H), \quad \bar{\psi} = \sum_b p_b\psi(b). \tag{33}$$

The weighted baseline satisfies $\bar{\psi} = \text{Cov}_{b\sim p}(p_b, \log p_b) \ge 0$, because both arguments are increasing functions of p_b. For example, if $p_a < e^{-H}$ then $\psi(a) < 0$ and the first-order change is negative. Being non-modal alone does not imply this inequality. In a shared neural parameterization, the logit update contains the parameter Jacobian, while PPO clipping, the optimizer, and other states contribute further terms. Thus this isolated calculation cannot establish the sign of the actual training-wide entropy change or of deleting an entire suffix.

Removing post-failure suffix updates. Once a trajectory becomes irrecoverable, later token choices cannot restore task success, yet nonzero estimated advantages still assign these tokens policy-gradient weights. In the local logit model above, negative updates suppress sampled alternatives, including tokens that did not cause the failure, while positive updates reinforce sampled continuations. For alternatives satisfying $p_a < e^{-H}$, Equation (33) gives an entropy-decreasing negative update to first order. By excluding the suffix after a detected stop, ESPO removes its update terms. When these terms are entropy-reducing on balance, their removal reduces distribution sharpening and preserves probability mass on alternative continuations. This mechanism links suffix removal to slower entropy decay without requiring all suffix advantages to be negative.

Combined effect on entropy. Suffix removal acts alongside the changed prefix updates: the terminal penalty still propagates through GAE. Negative prefix updates satisfying the local entropy condition above exert mode-seeking pressure. The observed entropy trend reflects the balance of these effects, shaped by normalization, PPO clipping, and shared-parameter interactions. Reduced entropy-decreasing pressure from suffix updates is consistent with the slower decay under ESPO in Figure 2b, alongside the entropy-decreasing prefix updates described above. Entropy is computed on all tokens of the full instrumented rollouts before applying the truncation mask, including tokens beyond the detected stopping position. Thus, the plotted difference is not caused by excluding suffix entropies from the average. Over the plotted training period, ESPO maintains higher mean policy entropy than PPO without entropy collapse.