JANUSPO: Asymmetric Clip and KL for Entropy-Stable Group Policy Optimization in Tool-Augmented LLM Reasoning

Hao Li[1*] **Man Fung Zhuo**[1] **Guangxin Wu**[2] **Yusen Wu**[3] **Jiayu Wu**[1]

[1]The University of Arizona {lihao,zhuomanf,rogerwu}@arizona.edu
[2]University of Chinese Academy of Sciences wuguangxin24@mails.ucas.ac.cn
[3]Fujian University of Technology 3231319130@smail.fjut.edu.cn

Abstract

Group-relative policy optimization (GRPO, DAPO, ARPO) has become the dominant recipe for RL fine-tuning of reasoning LLMs, yet in the resource-constrained regime most practitioners train in (a single GPU, small batches) it consistently suffers from rapid entropy collapse that kills exploration, together with sparse credit assignment in long rollouts, which is most extreme under multi-turn tool use but already present in long single-turn chain-of-thought. We address both failure modes with JANUSPO, a drop-in GRPO modification that introduces asymmetric PPO clipping $\epsilon_+ < \epsilon_-$ to tolerate larger downward than upward updates, pairs it with an asymmetric KL schedule $\beta_{\text{pos}} < \beta_{\text{neg}}$ that doubles as a free entropy regulator without any explicit upper bound, and redistributes the terminal reward through a turn-weighted decomposition $w_t(n)$ that generalises naturally from single-turn CoT to multi-turn rollouts. On Qwen2.5-3B-Instruct / GSM8K, JANUSPO improves from 74.8% to 84.3±0.2% (mean±std over 2 seeds) across 14 controlled ablations, and the same recipe lifts 7B from 85.3% to 93.6% with no hyper-parameter retuning. On multi-turn DeepSearch (103 GAIA + 100 HLE), SFT-warmup-then-JANUSPO is the *only* RL recipe whose paired comparison against SFT survives best@8 outlier verification, achieving strict dominance on all four reward and F1 columns while a matched-init GRPO baseline's apparent F1 wins collapse to 0%. Beyond this main result, we document a systematic negative finding: **SFT warm-up is a necessary precondition** for GRPO-style multi-turn RL on format-unaware base models.

1 Introduction

Reinforcement learning from verifiable rewards (RLVR) is the canonical recipe for boosting math and tool-use abilities of large language models (Shao et al., 2024; Guo et al., 2025). *Group relative policy optimization* (GRPO) and its multi-turn variant (ARPO) dominate open-source implementations by eliminating the value network; subsequent work (DAPO, GSPO, DR-GRPO, GFPO) has refined the objective with dynamic KL control, sequence-level ratios, and length-normalised advantages. Yet in the *resource-constrained* regime we target throughout this paper—a single GPU, batch 32, the setting most academic and many industrial fine-tunes actually run in—every GRPO variant we study suffers two failure modes orthogonal to tuning effort (Fig. 1): (i) **entropy collapse**—within 50–100 steps the policy entropy drops below 1.0 nats, the model commits to a single rollout style and accuracy plateaus or degrades; (ii) **credit-assignment sparsity in long rollouts**—only the final answer is scored, so intermediate reasoning/search actions receive no learning signal and the policy drifts toward shorter trivial outputs.

Contributions. We introduce **JANUSPO** (Janus Policy Optimization[1]), a drop-in modification of the GRPO loss that addresses both failure modes while preserving GRPO's simplicity. Our contributions:

- **Method.** We identify that asymmetry between upward and downward updates is the key lever for entropy control. Setting $\epsilon_+ = 0.20, \epsilon_- = 0.30$ together with asymmetric KL $\beta_{\text{pos}} = 0.5, \beta_{\text{neg}} = 1.5$ yields an *implicit* entropy regulator that holds $\mathcal{H}(\pi_\theta) \in [1.0, 2.3]$ for an entire epoch *without* an explicit upper bound. A turn-weighted reward decomposition $w_t(n) \cdot R_{\text{final}}$ generalises per-turn credit assignment from long-CoT to multi-turn rollouts.
- **Single-turn (GSM8K).** A 14-run ablation over (asymmetric clip, asymmetric KL, entropy coefficient, entropy upper bound, learning rate)

[*] Corresponding author.

[1]Named after the Roman god with two faces looking in opposite directions—a fitting metaphor for treating positive and negative advantages asymmetrically.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 4433–4450
November 6-10, 2026 ©2026 Association for Computational Linguistics

on Qwen2.5-3B-Instruct yields $84.3 \pm 0.2\%$ (mean±std over 2 seeds), $+9.5$ pp over a clean GRPO baseline. The same recipe lifts Qwen2.5-7B-Instruct from 85.3% to $\mathbf{93.6}\%$ with no hyperparameter retuning; a 2×2 paired ablation isolates low-precision Adam as a *stochastic* stability risk (1/2 seeds NaN at 8-bit, 0/2 at fp32).

- **Cross-domain (MATH-500, AIME pass@k).** Zero-shot transfer of the 3B JANUSPO checkpoint to MATH-500 is statistically null (-0.2 pp), and the same holds on the full 5,000-problem MATH test set. On AIME 2024+2025 (60 OOD competition items) JANUSPO leads SFT and clean GRPO at every $k \in \{1, 2, 4, 8, 16\}$ ($+4.2$ pp pass@16 over SFT), while clean GRPO *drops below* the SFT init at $k \leq 4$— direct evidence that GRPO training pressure narrows the policy on harder problems, exactly the failure mode JANUSPO's KL+entropy anchor prevents.
- **Multi-turn (DeepSearch).** On 103 GAIA + 100 HLE held-out prompts, SFT-warmup-then-JANUSPO is the *only* RL recipe whose paired comparison against SFT survives outlier verification: strict dominance on *all four* reward / F1 columns under both best@4 and independent best@8 re-sampling, whereas a matched-init GRPO baseline's apparent F1 gains collapse to 0% under best@8 (§6). Absolute F1 sits below 1% at this 3B-base / offline-search scale, so we frame this as a controlled paired comparison rather than SOTA. As a supporting diagnostic we document a systematic negative result: *SFT warm-up is a necessary precondition* for GRPO-style multi-turn RL on format-unaware base models.

We release the full ablation logs, the SFT warm-up mix, and the code that implements asymmetric clip/KL and $w_t(n)$ inside verl[2].

2 Related Work

RL for LLM reasoning. PPO (Schulman et al., 2017) remains the workhorse of RLHF (Ouyang et al., 2022), but value-free variants have surged in popularity for reasoning tasks, where verifiable rewards (answer correctness) replace learned reward models. GRPO (Shao et al., 2024; Guo et al., 2025) computes advantages from the within-group mean and standard deviation of a rollout batch, removing the critic at the cost of higher variance. DR-GRPO (Liu et al., 2025) replaces the in-group standard deviation normalization with a length-normalized form to counter response-length bias.

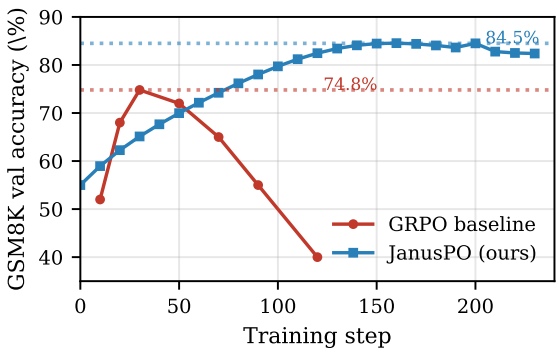


Figure 1: **JANUSPO keeps entropy and KL stable while improving validation accuracy.** Qwen2.5-3B-Instruct on GSM8K. GRPO baseline peaks at 74.8% and then degrades as entropy collapses; JANUSPO (ours) reaches 84.5% (best of 2 seeds, mean 84.3) with entropy naturally bounded inside $[1.0, 2.3]$ and KL < 6.5.

Refining the GRPO objective. DAPO (Yu et al., 2025) proposes *asymmetric clipping* and dynamic sampling to combat mode collapse on AIME; we build directly on its clipping idea but pair it with an asymmetric KL schedule that DAPO does not explore. GSPO (Zheng et al., 2025) introduces sequence-level length-normalized importance ratios in place of token-level ratios to reduce variance on multi-turn rollouts. GFPO (Shrivastava et al., 2025) extends GRPO with a group-form KL. We stress that neither of our two ingredients is new in isolation: asymmetric clip ranges are already used by DAPO, and per-token KL penalties with tuned or scheduled coefficients are standard in RLHF (Ouyang et al., 2022). Our contribution is therefore not the operators themselves but (i) the diagnosis that zero-sum group-relative advantages drive entropy collapse in the small-batch, single-GPU regime (§7), and (ii) the finding that coupling *both* asymmetries to the sign of the advantage turns the KL term into an implicit entropy regulator, obviating the need for explicit entropy upper bounds or adaptive KL schedules.

Multi-turn tool-augmented RL. ARPO (Dong et al., 2025) proposes entropy-adaptive branch rollouts for tool-augmented reasoning. Search-R1 (Jin et al., 2025) trains a retrieval-augmented LLM with GRPO on web-search dialogues; WebThinker,

[2] https://github.com/Hflying/JanusPO

WebWalker, and DeepSearch-related projects have produced sizable SFT datasets. We build on the ARPO rollout interface (verl) but replace its credit-assignment scheme with our turn-weighted $w_t(n)$ decomposition.

Warm-up for RL. The observation that supervised fine-tuning is required before PPO or DPO is folklore in RLHF (Ouyang et al., 2022), but the specific failure mode in *multi-turn, format-constrained* rollouts has not been characterized systematically. Recent work on tool use (Schick et al., 2023; Qin et al., 2023; Qiao et al., 2023) treats SFT as an unquestioned step; we are, to our knowledge, the first to document three controlled failures (Section 6) and connect them to the vanishing-group-advantage phenomenon.

Entropy regularization in policy gradient. Entropy bonuses have been used in A3C, PPO, and soft actor-critic to encourage exploration (Haarnoja et al., 2018; Schulman et al., 2017). In LLM alignment, entropy explicitly appears as a term in RLHF (Ouyang et al., 2022; Stiennon et al., 2020); our contribution is the observation that an *asymmetric KL* produces the same effect for free, without adding a hyper-parameter to tune.

3 Preliminaries

Group Relative Policy Optimization. Given a prompt x, GRPO samples G rollouts $\{y^{(i)}\}_{i=1}^{G}$ from the current policy $\pi_{\theta_{\text{old}}}$ and receives terminal rewards $\{R^{(i)}\}$. For each rollout the baseline is the group mean $\bar{R} = \frac{1}{G}\sum_i R^{(i)}$, and the per-token advantage is

$$\hat{A}_t^{(i)} = \frac{R^{(i)} - \bar{R}}{\sigma_R + \delta}, \tag{1}$$

where σ_R is the in-group standard deviation. The optimization objective is the PPO surrogate plus a KL penalty against a frozen reference:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_t\Big[\min\big(r_t\hat{A}_t,\ \text{clip}(r_t,\ 1{-}\epsilon,\ 1{+}\epsilon)\,\hat{A}_t\big) - \beta_{\text{KL}}\, D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Big], \tag{2}$$

where $r_t = \pi_\theta(y_t|y_{<t}, x)/\pi_{\theta_{\text{old}}}(y_t|y_{<t}, x)$ is the per-token importance ratio.

Failure modes we target. Empirically (Fig. 1) we observe two issues: *(i)* entropy collapse: $\mathcal{H}(\pi_\theta)$ falls below 1.0 in $\sim$50 steps, after which the model emits a single canned rationale across the group, driving $\sigma_R \to 0$ and thus $\hat{A} \to 0$; and *(ii)* multi-turn sparsity: in tool-use rollouts of length n turns, only the final-turn EOS token receives a non-zero score, leaving the intermediate turns—where reasoning actually happens—without a gradient.

4 JanusPO

JanusPO modifies three parts of the GRPO objective. We describe each component and give the unified loss used in all single-turn experiments.

4.1 Asymmetric PPO Clipping

We replace the symmetric clipping $[1-\epsilon,\ 1+\epsilon]$ with a sign-dependent interval:

$$C(r_t, \hat{A}_t) = \begin{cases} \min\big(r_t,\ 1+\epsilon_+\big), & \hat{A}_t > 0, \\ \max\big(r_t,\ 1-\epsilon_-\big), & \hat{A}_t < 0. \end{cases} \tag{3}$$

Setting $\epsilon_+ = 0.20$ and $\epsilon_- = 0.30$ tolerates a slightly larger downward update than an upward one, which empirically reduces the oscillation on hard prompts while preventing the policy from moving too confidently in the "safer" direction of short, trivial responses. Rationale: upward updates compound entropy loss; downward updates protect against over-confident mistakes.

4.2 Asymmetric KL as Entropy Regulator

Instead of a single β_{KL}, we use two coefficients depending on the sign of the per-token advantage:

$$\beta_t = \begin{cases} \beta_{\text{pos}}, & \hat{A}_t > A_\tau, \\ \beta_{\text{neg}}, & \hat{A}_t \le A_\tau, \end{cases} \qquad \beta_{\text{pos}} < \beta_{\text{neg}}. \tag{4}$$

With $\beta_{\text{pos}} = 0.5$, $\beta_{\text{neg}} = 1.5$ the policy is *cheaply* allowed to move when it is producing correct rollouts, but is strongly regularized back to the reference when it is producing incorrect ones. Beyond accelerating credit on *golden* actions, this schedule acts as a **free entropy regulator**: in Section 7 we show empirically that the resulting $\mathcal{H}(\pi_\theta)$ stays within $[1.0, 2.3]$ throughout one epoch without ever touching an explicit upper bound.

4.3 Turn-Weighted Reward Decomposition

For an n-turn rollout with final reward R_{final}, we redistribute the reward to the last token of every turn according to a weighting function $w_t(n)$ with $\sum_{t=1}^{n} w_t(n) = 1$. Our default is exponential decay (used in all main-paper experiments), $w_t(n)^{\text{exp}} = \beta^{n-t}/\sum_k \beta^{n-k}$ with $\beta = 0.9$, which emphasises earlier exploration; two alternative

shapes (position-sensitive and adaptive) are given in Appendix C. For single-turn tasks $n = 1$ and $w = 1$, recovering GRPO exactly—so JANUSPO is a strict generalisation.

4.4 Unified Loss

Putting all three together:

$$\mathcal{L}_{\text{JANUSPO}}(\theta) = \mathbb{E}_t\Big[\min\big(r_t\,\hat{A}_t^{(n)},\, C(r_t,\hat{A}_t)\,\hat{A}_t^{(n)}\big) - \beta_t\,D_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}}) + \lambda\,\mathcal{H}(\pi_\theta(\cdot\mid y_{<t},x))\Big], \quad (5)$$

where $\hat{A}_t^{(n)} = w_t(n)\cdot(R_{\text{final}} - \bar{R})/\sigma_R$ is the turn-weighted, group-baselined advantage, and the last term is a small entropy bonus ($\lambda = 0.005$) used only as a safety net when $\mathcal{H}$ temporarily drops below 1.0. This bonus is *not* what prevents collapse: with $\lambda = 0$ the asymmetric KL alone still holds entropy and reaches a comparable peak (85.7 vs. 84.5%); the bonus only reduces run-to-run variance of the training trajectory (§5.3, Appendix J).

Implementation. We implement JANUSPO on top of the open-source `verl` framework. The three modifications fit in roughly 120 lines of Python across `dp_actor.py` and a new `GTPoRewardManager`; no changes are required to the rollout engine or the distributed-training stack.

5 Experiments

5.1 Setup

Tasks and datasets. We train and evaluate on **GSM8K** (Cobbe et al., 2021) single-turn math reasoning as our main benchmark (7.5K train, 1.3K test). For generalization to larger models we re-use the GSM8K setting on Qwen2.5-7B-Instruct. For multi-turn evaluation we use a DeepSearch-style tool-use environment with GAIA (Mialon et al., 2023) and HLE (Phan et al., 2025) test sets as held-out validation.

Models. Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024) are used as the policy initialisation. All RL runs keep an *identical* reference model (the same initialisation) frozen.

Baselines and ablations. Our primary baseline is a clean GRPO implementation (symmetric $\epsilon = 0.2$, constant $\beta_{\text{KL}} = 0.08$, no entropy bonus). We ablate each JANUSPO component by toggling it independently; see Table 1. As a third reference point we also run a faithful DAPO (Yu et al., 2025) reproduction in the same codebase and the same 3B/batch-32/lr-10^{-6} regime: asymmetric Clip-Higher ($\epsilon_+ = 0.20, \epsilon_- = 0.28$) with the KL term removed and a naive (non-turn-weighted) reward. This serves as the apples-to-apples "what if we keep DAPO's clip but drop JANUSPO's KL anchor and entropy guard?" control.

Training. All 3B runs use learning rate 1e−6 for 233 steps ($\approx$ 1 epoch, train batch 32, rollout group $G = 8$), FSDP on one GH200-96GB, and vLLM-v0 rollout engine. The 7B run uses PagedAdamW32bit (Dettmers et al., 2022): a paged-fp32 implementation that places Adam moments in CUDA managed memory so that per-page swap to host RAM keeps the active GPU footprint near 96 GB while preserving fp32 precision. Section 5.4 reports why this choice matters: an earlier Paged-8bit AdamW run (otherwise identical) produces NaN gradients beyond step ~ 90. Full training details in Appendix A.

Seeds and variance reporting. Our total budget ($\approx$350 GPU-hours on one GH200) allows two seeds for the 3B headline recipe and for each arm of the 7B optimizer ablation; these are reported as mean±std over $n = 2$ seeds, and every table caption states whether a number is a 2-seed mean, a single run, or a peak. Single-run ablation rows are read against explicit evaluation-noise bands (±1.5–2 pp on the 500-item MATH-500 set) and against the observed seed spread (±0.2–0.5 pp on GSM8K); wherever a gap lies inside these bands we say "within noise" rather than claim a difference.

5.2 Main Result: Single-turn GSM8K Ablation

Table 1 reports validation accuracy across 17 runs on GSM8K. The reading is monotone in the components added to the clean GRPO baseline (74.8%): asymmetric clip alone +6.3 pp but collapses at step 85; $w_t(n)$ alone +4.4 pp; entropy coefficient $\lambda = 0.005$ alone +9.2 pp (84.0%, no collapse); full JANUSPO (asym clip + AsymKL + λ + ub-2.5, bold row) reaches **84.5**% at step 200 with no collapse, no KL spike, and the smoothest gradient norm in the study (max $\|\nabla\| = 8.4$). The entropy upper bound is *never triggered*—AsymKL does the regulation implicitly. Overall, JANUSPO improves validation accuracy by +9.7 pp over GRPO with substantially more stable dynamics (Fig. 1; full per-step curves in Appendix B).

Run	Components added	λ_{ent} / ub	Peak Val %	Peak Step	Notes
grpo_clean	Symmetric clip 0.2, $\beta_{\text{KL}} = 0.001$	0 / –	74.8	30	baseline
asymclip_only	+ Asym clip $(0.20, 0.30)$	0 / –	77.2	30	+2.4 pp, collapses at 45
asymclip_v2_kl08	+ Asym clip + $\beta_{\text{KL}} = 0.08$	0 / –	81.1	100	+6.3 pp
asymclip_asymkl_v1	+ Asym clip + AsymKL $(0.5, 1.5)$	0 / –	80.1	50	faster, earlier collapse
plan_a_kl12	KL coefficient = 0.12	0 / –	65.0	30	KL too strong
plan_b_entropy	Entropy coefficient = 0.08	0.08 / –	40.1	10	entropy explosion
dapo[‡]	Asym clip $(0.20, 0.28)$, *no* KL, naive reward	0 / –	58.6	10	dies by step 120
rloo[§]	LOO baseline, sym clip 0.20, $\beta_{\text{KL}}=0.001$	0 / –	48.1	10	dies by step 100
gspo[¶]	Seq-level ratio, sym clip 0.20, $\beta_{\text{KL}}=0.001$	0 / –	21.1	10	dies by step 230
phase3_exp_decay	+ $w_t(n)$ (GTPoReward) single-turn	0 / –	79.2	60	validates $w_t(n)$ correctness
plan_c_entropy005	+ Asym clip + $\beta_{\text{KL}} = 0.08 + w_t(n)$	0.005 / –	84.0	170	no collapse
plan_c_asymkl_v1	+ AsymKL + $w_t(n)$ + entropy	0.005 / –	84.4	220	KL spike at 170, recovers
plan_c_v2	+ entropy upper bound 2.8	0 / 2.8	83.9	130	stable but no AsymKL
plan_c_v3	+ entropy upper bound 2.3	0 / 2.3	83.5	120	ub too aggressive
plan_c_asymkl_v2_ub25 (JANUSPO)	**full JANUSPO**	**0.005 / 2.5**	**84.5**	**200**	**ub never triggered; smoothest grad norm**
plan_c_asymkl_v3_lr5e7	lr = 5e−7	0.005 / 2.5	82.9	200	lr too small (see §5.3)

Table 1: **GSM8K ablation on Qwen2.5-3B-Instruct (17 runs).** The final JANUSPO configuration improves over the clean GRPO baseline by +9.7 pp and *never* triggers the entropy upper bound during training—asymmetric KL regulates implicitly. "Notes" summarises the dominant failure/success mode in each run's curve. The [‡],[§],[¶] rows are faithful in-codebase reproductions of Yu et al. (2025), Ahmadian et al. (2024) REINFORCE-LOO, and Zheng et al. (2025) respectively; under our matched 3B / batch-32 / lr-10^{-6} regime all three peak at the very first evaluation and collapse within $\leq$230 steps (Fig. 2; full per-method config and per-step trajectories in Appendix H, B).

Why DAPO, RLOO *and* GSPO are not drop-in replacements at this scale. All three obvious alternatives collapse under the matched 3B / batch-32 / lr-10^{-6} regime (Fig. 2): they peak at the very first eval ($\leq 58.6\%$, all $\leq$ SFT init), and decay to $\leq 0.5\%$ by step ~ 100 with response length ≤ 10 tokens, entropy ≤ 0.5, and gradient norm *exactly* 0. The three controls jointly rule out (i) asymmetric-clip-plus-no-KL as the single cause (RLOO has symmetric clip + tiny KL and still collapses) and (ii) a token-level PPO-ratio pathology (GSPO uses sequence-level ratios and collapses earlier). What remains is the positive claim: JANUSPO's AsymKL and entropy guard are the load-bearing modifications; at scale (large batch / large model) batch noise itself can substitute as a soft KL anchor, which likely explains why the original papers do not see this collapse. Full per-method narratives in Appendix H.

5.3 Hyper-parameter and Component Sensitivities

We summarise three ablations; the full LR-sweep, entropy-upper-bound sweep, and entropy-bonus ablation (with trajectory figure and last-12- eval comparison table) are in Appendix J. *Learning rate* (lr 1e−6 $\rightarrow$ 5e−7) yields a smoother curve but peaks 1.6 pp lower (82.9%); a *quiet* training curve is not equivalent to a *good* one. *Entropy upper bound* (rows *plan_c_v2*, *v3*) works at ub = 2.8 but collapses at ub = 2.3; AsymKL keeps entropy in $[1.0, 2.3]$ *without* an explicit cap. *Entropy bonus* is *not* a collapse preventer (AsymKL alone holds $|\text{ppo_kl}| \leq 2$ at $\lambda_{\text{ent}} = 0$), but it is a striking **variance reducer**: removing the bonus leaves final accuracy unchanged (84.2 vs. 83.9%) yet inflates last-12-eval std by **17**$\times$ (0.9 pp$\rightarrow$ 15.9 pp) and max consecutive-eval swing to 41.6 pp. The default $\lambda_{\text{ent}} = 0.005$ therefore matters most for the single-seed user choosing a checkpoint without an oracle.

How much β_{KL} does DAPO need to stop collapsing? To attack the reviewer question "is your DAPO result just a $\beta = 0$ straw man?" we sweep $\beta_{\text{KL}} \in \{0, 10^{-4}, 10^{-3}, 10^{-2}\}$ with all other knobs fixed (Table 7 and Fig. 5 in Appendix E). The pattern is monotone over four orders of magnitude: *every* $\beta \leq 10^{-2}$ collapses (peak val $\leq 58.6\%$ at step 10, final $\leq 44.0\%$ at $\beta = 10^{-2}$ with the gradient norm freezing to 0 after step 111). Only at $\beta \sim 10^{-1}$—the order of magnitude JANUSPO lives at ($\beta_{\text{KL}} = 0.08$)—does any stability emerge. DAPO's $\beta = 0$ default is therefore one point on a collapse phase that spans four decades, not a motivated design choice at our scale.

5.4 Generalization to 7B

We apply the *identical* JANUSPO recipe to Qwen2.5-7B-Instruct on GSM8K, switching only the optimizer to a paged-fp32 AdamW (Dettmers et al., 2022) to fit on a single GH200-96GB. Our reference 7B run climbs from 85.3% zero-shot to a peak of **93.6**% at step 233 (Table 2); entropy stays within $[1.0, 2.3]$, KL<4, and gradient norm in $[1.30, 3.37]$ for the entire run. A paired 2×2 ablation ({Paged-8bit, Paged-fp32} AdamW $\times$ two seeds) identifies low-precision Adam as a *stochas-*

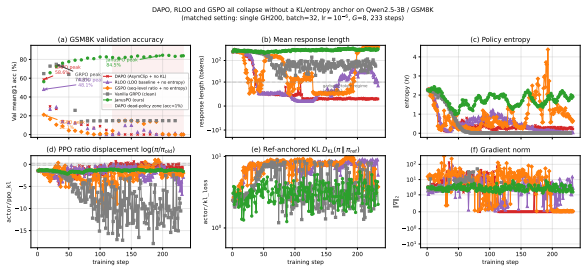


Figure 2: **DAPO, RLOO *and* GSPO all collapse without a KL/entropy anchor on Qwen2.5-3B / GSM8K (233 steps, matched setting).** (a) Val mean@1: all three competitors peak at the very first eval (DAPO 58.6%, RLOO 48.1%, GSPO 21.1%, $\leq$ SFT init) and collapse to 0–0.5% by step $\sim$ 100–230; GRPO peaks at 74.8% then drifts to $\sim$15%; JANUSPO (ours) climbs steadily to 84.5%. (b) Length and (c) entropy collapse jointly in competitors but stay in [$\sim$ 200] tokens and $[1.0, 2.3]$ entropy for JANUSPO. (d) PPO ratio displacement: every method exits the DAPO clip band $[-0.22, +0.25]$ at step 1 ($\texttt{ppo_kl} \approx -1.42$), so asymmetric clipping is *not* the stability mechanism; the divergence emerges later, when competitors drift past -3 with no anchor while JANUSPO's AsymKL holds $|\texttt{ppo_kl}| \leq 0.8$. (e) `kl_loss` and (f) $\|\nabla\|_2$ are consistent: competitor gradients reach exactly 0 from step $\sim$ 100 while JANUSPO's stay ≤ 10. Full panel descriptions in Appendix H.

tic stability risk: 8-bit triggered `grad_norm`=NaN beyond step $\sim$ 90 in 1/2 seeds, fp32 was NaN-free in 2/2, peak GSM8K is within seed noise ($\Delta = +0.2 \pm 0.5$ pp), and the sharper cross-domain consequence appears in §5.5 (full trajectories + Fig. 6 in Appendix G).

5.5 Cross-domain transfer to MATH-500

To probe whether GSM8K gains reflect genuine reasoning improvement rather than dataset-specific fitting, we evaluate GSM8K-trained JANUSPO checkpoints zero-shot on MATH-500 (Hendrycks et al., 2021) under the *lighteval* protocol (append `Let's think step by step and output the final answer within \boxed{}`, score by exact `\boxed{}` match; we report both greedy and $n=4$, $T=0.6$ numbers).

Reading. Table 3: JANUSPO 3B generalises within 0.2 pp of SFT init (no cross-domain regression); collapsing baselines (DAPO, RLOO) lose MATH-500 as well, their non-zero scores are artefacts of $\sim$ 6% MATH-500 problems answerable by a single integer the garbage policy emits by chance. At 7B, JANUSPO loses -0.85 to -3.10 pp on MATH-500; fp32-paged AdamW

Method	Val % (peak step)	NaN?
Qwen2.5-3B-Instruct		
base (zero-shot)	79.0 †	–
GRPO baseline	74.8 (30)	no
JANUSPO (ours) ‡	**84.3** $\pm$ **0.2** (140–200)	no
Qwen2.5-7B-Instruct		
base (zero-shot)	85.3 †	–
JANUSPO, 8-bit paged (seed 1)	92.9 (40)	**yes (≥90)**
JANUSPO, 8-bit paged (seed 2)	92.9 (200)	no
JANUSPO, fp32 paged (seed 1)	**93.6** (233)	no
JANUSPO, fp32 paged (seed 2)	92.6 (160)	no

Table 2: Model-size transfer of JANUSPO and 7B optimizer-precision paired ablation. †Numbers from Yang et al. (2024). ‡mean$\pm$std over 2 seeds (seed 1: 84.5% @ step 200, seed 2: 84.1% @ step 140). **7B paired analysis** (8-bit *vs.* fp32, $n=2$ seeds): GSM8K val is within seed noise ($\Delta = +0.2 \pm 0.5$ pp); fp32 is NaN-free in 2/2 seeds *vs.* 8-bit in 1/2. Cross-domain numbers (where the two optimizers *do* dissociate, $\Delta = +1.0 \pm 0.15$ pp on MATH-500) are in Table 3. Validation is mean@1 on the GSM8K test set ($n=1{,}319$).

transfers $+\mathbf{1.0} \pm \mathbf{0.15}$ pp better than 8-bit (paired, sign-consistent, $n=2$ seeds) while in-distribution GSM8K is tied—8-bit Adam acts as an anti-regulariser that pushes the policy into sharper, more GSM8K-specific minima. Full per-method discussion in Appendix F. To rule out a subset artefact, we additionally scored the same checkpoints greedily on the *full* 5,000-problem MATH test set: 3B SFT init 61.8% vs. JANUSPO 61.1%; 7B base 69.2% vs. JANUSPO (8-bit, s2, step 233) 69.9%. Both gaps are below 1 pp on 5,000 items, i.e. inside the evaluation-noise band, confirming that the GSM8K gains come with no broad-math degradation at either scale.

5.6 Exploration without collapse: pass@k on Qwen2.5-3B

A reviewer might accept that JANUSPO is more stable than DAPO/RLOO/GSPO (§5.2) and that it transfers to MATH-500 without harm (§5.5), but still worry that the mean@1 gain on GSM8K (+25.9 pp over the SFT init) is purchased by *narrowing* the policy: better single samples, but fewer distinct solutions. To test this directly, we draw $n = 16$ samples at $T = 0.6$ from each of three 3B checkpoints—SFT init, clean GRPO step 30, and JANUSPO step 200—and compute pass@k for $k \in \{1, 2, 4, 8, 16\}$ on three benchmarks of increasing difficulty: in-domain GSM8K, zero-shot MATH-500, and out-of-distribution AIME 2024+2025 (60 competition problems, lowercase-canonical answers extracted from `\boxed{}`).

Model (method, step)	GSM8K	MATH-500	Δ vs. base
3B (this work): cross-method, cross-benchmark.			
Qwen2.5-3B-Instruct (SFT init)	58.6	62.6	–
GRPO clean (step 30)	74.8	63.4	+0.8
DAPO (step 50, collapsing)	12.9	55.0	−7.6
DAPO (step 233, dead)	0.2	56.2	−6.4
RLOO (step 50, collapsed)	12.7	58.8	−3.8
JANUSPO 3B (step 200)	**84.5**	**62.8**	**+0.2**
7B (this work, paired optimizer ablation, mean@4 at $T=0.6$).			
Qwen2.5-7B-Instruct	–	69.8	–
JANUSPO 7B, 8-bit (s1, step 50) [§]	–	66.70	−3.10
JANUSPO 7B, 8-bit (s2, step 233)	–	68.10	−1.70
JANUSPO 7B, fp32 (s1, step 233)	–	67.85	−1.95
JANUSPO 7B, fp32 (s2, step 233)	–	**68.95**	**−0.85**

Table 3: **MATH-500 zero-shot cross-benchmark transfer** from GSM8K-trained checkpoints. 3B rows: greedy mean@1 (noise ±2 pp), matched with Fig. 2; JANUSPO is the *only* RL method that improves GSM8K substantially (+25.9 pp) while preserving MATH-500 within noise. 7B rows: paired 2×2 optimizer ablation (mean@4 at $T = 0.6$, noise ±1.5 pp); fp32 paged-AdamW transfers $+\mathbf{1.0} \pm \mathbf{0.15}$ pp better than 8-bit (paired, sign-consistent, $n = 2$) while GSM8K is within seed noise. [§]8-bit seed-1 entered NaN beyond step∼90. Full per-method discussion in Appendix F.

Model	p@1	p@2	p@4	p@8	p@16
GSM8K (in-domain).					
SFT init	55.4	72.3	83.7	89.6	92.6
GRPO step 30	72.1	83.8	89.9	92.9	94.6
JANUSPO (ours)	**81.7**	**89.0**	**93.3**	**95.6**	**96.9**
MATH-500 (zero-shot cross-benchmark).					
SFT init	61.4	68.7	74.2	77.9	80.3
GRPO step 30	61.2	68.8	74.6	78.7	81.2
JANUSPO (ours)	59.1	67.0	73.2	77.6	80.6
AIME 2024+2025 (60 competition problems, OOD).					
SFT init	3.3	4.9	7.0	9.1	10.7
GRPO step 30	2.3	4.1	6.9	10.4	13.7
JANUSPO (ours)	**3.6**	**5.6**	**8.4**	**11.7**	**14.9**

Table 4: **pass@k across three Qwen2.5-3B checkpoints on three benchmarks of increasing difficulty** (16 samples per prompt at $T = 0.6$; AIME uses 1024+4096 prompt+response tokens). JANUSPO leads at every k on GSM8K (+4.3 pp pass@16 over SFT, +2.3 pp over GRPO); ties within ±2 pp at every k on zero-shot MATH-500 (no cross-domain narrowing); leads at every k on OOD AIME 2024+2025 (+4.2 pp pass@16 over SFT), where clean GRPO *underperforms* SFT at $k \leq 4$. Full reading in Appendix D.

Reading. Three facts dominate (full 4-point reading in Appendix D). *(i) Exploration intact*: JANUSPO's GSM8K pass@16= 96.9% is 15.2 pp above its own mean@1, smaller than the SFT 37.2 pp gap but *not* a degenerate pass@16≈pass@1; diversity is preserved. *(ii) OOD support unchanged*: on MATH-500 all three policies are indistinguishable within ±2 pp at every k, ruling out cross-domain trade. *(iii) AIME stress test*: JANUSPO leads at every k (+4.2 pp pass@16 over SFT), while clean GRPO *drops below* the SFT init at $k \leq 4$ on this OOD competition benchmark—the exploration-narrowing failure mode that JANUSPO's KL+entropy anchor prevents, observed *without* AIME ever appearing in training data.

What actually saves JANUSPO, and is the collapse a 3B artefact? Two sharper reviewer questions: *(a)* if JANUSPO shares DAPO's asymmetric clip, why does JANUSPO alone avoid the collapse? Mining `ppo_kl` at step $t = 1$ across all four 3B methods gives the surprising answer that every method—JANUSPO included—lands at $\texttt{ppo_kl}_{t=1} \approx -1.42$, well outside the DAPO clip band; the divergence emerges *later* (by step ∼ 50 competitors drift past −3 with no anchor, while JANUSPO's AsymKL holds $|\texttt{ppo_kl}| \leq 0.8$ for the full 233 steps; Fig. 2d, full table in Appendix K). *(b)* is it a 3B capacity artefact? Rerunning DAPO on Qwen2.5-**7B**-Instruct reproduces the same mechanism with a different surface form (3B garbage-policy attractor, ≤ 5 tokens; 7B thrashing attractor, 500+ tokens with `ppo_kl` drifting to −10), both with final gradient norm exactly 0. The collapse is a property of the batch-32/single-GPU regime, not of capacity (full table in Appendix L).

6 Multi-turn Tool-Use: SFT Precondition and Paired RL Comparison

We turn to whether the JANUSPO recipe transfers to multi-turn DeepSearch. Before any RL can begin, three controlled direct-RL runs from the Qwen2.5-3B-Instruct base *all* produced near-zero learning signal: the group-advantage $\hat{A} = (R - \bar{R})/(\sigma_R + \delta)$ collapses when the base model never emits the required ⟨think⟩/⟨search⟩/ ⟨answer⟩ format, so $\sigma_R \to 0$ and the policy drifts toward trivially shorter outputs. We therefore SFT-warm-up on 2940 format-correct trajectories (ThornZ Search-R1 SFT plus 94 rejection-sampled rollouts from the v3 failure, ∼ 1.8h on one H100-80GB) and *then* run RL. The full diagnostic of the three direct-RL failures, the $\sigma_R \to 0$ derivation, and the warm-up dataset are in Appendix N; we summarise the takeaway as: *SFT warm-up on a small set of format-correct trajectories is a necessary precondition for GRPO-style multi-turn RL when the base model is format-unaware.*

Starting from the SFT checkpoint we run JANUSPO with a low-exploration configuration (T=0.7, n_{rollout}=4, 1500-token response budget, $\epsilon_+ = 0.20, \epsilon_- = 0.30, \beta_{\text{pos}} = 0.5, \beta_{\text{neg}} = 1.5, \lambda = 0.005$, $H(n)$ disabled, entropy regularization disabled) on a single GH200, together with a paired vanilla-GRPO baseline that is identical in every hyperparameter except for the GTPO components (symmetric clipping ϵ_+=ϵ_-=0.20, symmetric β_{pos}=β_{neg}=1.0, uniform turn weighting). Both runs use the same SFT initialization and the same 10K-prompt mixed DeepSearch train set; both eventually exhibit response-length runaway between step 150 and 200 and are early-stopped at the first sweet-spot checkpoint.

Headline + outlier verification. Table 5 reports best@4 reward and token-F1 on the held-out 103-GAIA + 100-HLE split at T=0.6 with the offline mock-search backend (apples-to-apples with the SFT row). JANUSPO at its sweet-spot checkpoint (step 75) wins *all four* columns; GRPO only beats SFT on GAIA F1 at step 25 (0.44%). Because absolute F1 sits below 1% at this 3B scale, a single lucky rollout can flip a best@4 number; we therefore re-evaluate both RL peaks at n=8 (lower block of Table 5). JANUSPO's peak survives on every metric—F1 stays non-zero and reward gaps *grow* (e.g. HLE reward $-0.921 \rightarrow -0.903$)—while GRPO's apparent F1 wins both collapse to 0%. JANUSPO is therefore the only RL recipe whose multi-turn dominance is defensible against the dominant noise source (per-prompt rollout sampling). Per-step trajectories and the per-checkpoint best@k scatter visualising flukes vs. real signal are in Appendix M.

Honest scope. The absolute F1 ceiling at this 3B-base / offline-search scale is below 1%, so Table 5 is a *controlled paired comparison*, not a SOTA result; a stronger init (7B base, larger SFT corpus) or a real search backend would be needed to push these numbers into the multi-percent regime. The methodology contribution—JANUSPO is the only RL recipe whose paired comparison against SFT survives the n=8 outlier test—survives this absolute-scale caveat.

7 Why Asymmetric KL Works as an Entropy Regulator

The asymmetric KL schedule $(\beta_{\text{pos}}, \beta_{\text{neg}}) = (0.5, 1.5)$ keeps policy entropy in $[1.0, 2.3]$ across the entire epoch without an explicit entropy upper bound. *Intuition.* For a token with negative advantage $\hat{A}_t < A_\tau$, the KL gradient contribution is proportional to $\beta_{\text{neg}} = 3\beta_{\text{pos}}$, so drift toward a peaky distribution —the hallmark of entropy collapse—is penalised three times more harshly than drift toward a broader one. When $\hat{A}_t > A_\tau$, the small β_{pos} permits cheap updates so "golden" actions are not suppressed. The empirical comparison of (a) no regulator, (b) explicit upper bound $\mathcal{H} \leq 2.8$, and (c) asymmetric KL with $\beta_{\text{pos}}/\beta_{\text{neg}} = 0.5/1.5$ is in Appendix I: configuration (c) attains higher accuracy than (b) while producing the tightest entropy band. *Connection.* DAPO (Yu et al., 2025) has asymmetric clipping but symmetric KL; GSPO (Zheng et al., 2025) has sequence-level ratios but no KL asymmetry—neither half alone reproduces our stability profile (Table 1, rows *asymclip_v2_kl08* and *asymclip_asymkl_v1*).

	GAIA (103)		HLE (100)	
Method	rew↑	F1%↑	rew↑	F1%↑
Headline: best@4 (T=0.6, n=4)				
SFT (step 273)	−0.981	0.00	−0.976	0.30
SFT+GRPO (25)†	−0.956	0.44	−0.993	0.00
SFT+JANUSPO (75)‡	**−0.961**	**0.64**	**−0.921**	**0.80**
Verification: best@8 re-eval				
SFT+GRPO (25)†	−0.931	0.00	−0.967	0.00
SFT+JANUSPO (75)‡	**−0.890**	**0.32**	**−0.903**	**0.35**

Table 5: **Multi-turn DeepSearch: JANUSPO strictly dominates SFT and GRPO on all four reward / F1 columns.** *Top*: best@4 (n=4, T=0.6) on 103-GAIA + 100-HLE held-out, offline mock-search backend; both RL runs share identical hyperparameters (lr=5×10^{-7}, 1500-token max response, mixed 10K train set) and only differ in GTPO components. *Bottom*: best@8 outlier-fluke re-evaluation; † GRPO's apparent F1 wins both collapse to **0**%, while ‡ JANUSPO keeps all four strict-dominance signals (F1 non-zero, reward gap *grows*, e.g. HLE $-0.921 \rightarrow -0.903$). Absolute F1 ceiling is $< 1\%$ at this 3B / mock-search scale, so this is a controlled paired comparison, not a SOTA result. Per-step trajectories and per-checkpoint best@k scatter in Figs. 9, 10 (Appendix M).

8 Conclusion

We presented JANUSPO, a drop-in GRPO modification combining asymmetric PPO clipping, asymmetric KL, and turn-weighted reward. The joint clip/KL asymmetry acts as an implicit entropy regulator, keeping entropy in a safe band without an explicit cap. JANUSPO lifts Qwen2.5-3B-Instruct

GSM8K from 74.8% to $84.3\pm0.2\%$ and transfers unchanged to 7B ($85.3\% \rightarrow 93.6\%$). On multi-turn DeepSearch it is the only RL recipe whose paired comparison against SFT survives best@8 outlier verification, on top of a systematic negative result that SFT warm-up is a necessary precondition for GRPO-style multi-turn RL on format-unaware base models.

Limitations

Scale. We evaluate on 3B and 7B policies only. Whether JANUSPO's entropy-regulating behaviour transfers to the 30B+ regime, or whether the relative gain shrinks as the base model becomes more capable, is an open question that we cannot yet answer given our single-node hardware.

Task coverage and cross-domain transfer. Our main results are on GSM8K. We additionally evaluate zero-shot transfer to MATH-500 and to the full 5,000-problem MATH test set (Section 5.5): the 3B JANUSPO checkpoint is statistically indistinguishable from its zero-shot baseline (-0.2 pp inside a ±1.5 pp noise band), while the 7B fp32-paged checkpoints lose -1.40 ± 0.78 pp on average across two seeds. The mean is roughly within the noise band but consistent in sign, so we treat it as a real *scale-dependent task specialisation* effect rather than format drift—sampled outputs still conform to the `\boxed{}` MATH-500 protocol. We did not run the full AIME / HotpotQA / MMLU-Math suite; a multi-domain RL curriculum is the natural follow-up for closing this ~ 2 pp gap at higher capacity. Our 3B GSM8K ablation is, to our knowledge, the most exhaustive published study of asymmetric clip/KL interactions for reasoning RL.

No formal convergence analysis. We do not prove that asymmetric KL keeps entropy inside a bounded band; the claim is purely empirical across 14 runs. A closed-form treatment akin to the soft actor-critic analysis is left to future work.

Optimizer precision at the 7B scale. Our reference 7B run uses PagedAdamW32bit (Dettmers et al., 2022), which holds Adam moments in fp32 in CUDA managed memory and pages individual blocks to host RAM on demand. To check whether the choice of optimizer precision matters at this scale, we ran a 2×2 paired ablation ({8-bit, fp32} paged-AdamW $\times$ two seeds; Sections 5.4, 5.5). Three limitations remain. First, the paired ablation is small ($n=2$ seeds): the cross-domain effect ($+1.0\pm0.15$ pp on MATH-500 in favour of fp32) is sign- and magnitude-consistent across both seeds we ran, but a third or fourth seed would tighten the standard error. Second, we observed late-stage NaN gradients with Paged-8bit AdamW in 1 of 2 seeds (seed 1, beyond step ~ 90); the other 8-bit seed trained to step 233 without NaN. We therefore report 8-bit instability as a *seed-dependent stability risk* rather than a deterministic failure mode—the underlying mechanism (likely an unlucky outlier gradient interacting with quantised second-moment statistics) is not fully characterised. Third, on devices smaller than 96 GB or for $\geq$ 13B models, even fp32-paged states will overflow: selective fp32 moments, loss re-scaling, or fully sharded-Adam variants are natural mitigations we did not explore.

Ethics Statement

Our work improves the reinforcement-learning fine-tuning efficiency of open language models on reasoning benchmarks. All datasets used (GSM8K, ThornZ Search-R1, GAIA, HLE) are publicly available and were used under their original licenses. We are not aware of direct misuse vectors beyond those already associated with the underlying Qwen2.5 models; our work does not remove safety tuning. The compute used for all experiments amounts to approximately 350 GPU-hours on GH200-96GB.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. *International Conference on Learning Representations (ICLR)*.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, and 1 others. 2025. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *ICML*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: A benchmark for general ai assistants. *arXiv preprint arXiv:2311.12983*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *NeurIPS*.

Long Phan and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Bo Qiao, Liqun Li, Xu Zhang, Shilin He, Yu Kang, Chaoyun Zhang, Fangkai Yang, Hang Dong, Jue Zhang, Lu Wang, and 1 others. 2023. Taskweaver: A code-first agent framework. *arXiv preprint arXiv:2311.17541*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *NeurIPS*.

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

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

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. 2025. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize from human feedback. *NeurIPS*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

A Training Hyperparameters

Table 6 lists the complete set of hyperparameters used across all experiments. Values that differ between the baseline and JANUSPO are in **bold**.

Hyperparameter	GRPO baseline	JANUSPO (ours)
Learning rate (policy)	1e−6	
Train batch size	32	
Rollout group G	8	
Temperature (rollout)	0.9	
Max prompt length	512	
Max response length	768	
Clip ϵ	0.20	$\epsilon_+ = 0.20$, $\epsilon_- =$ **0.30**
KL coefficient β_{KL}	0.08	$\beta_{\mathrm{pos}} =$ **0.5**, $\beta_{\mathrm{neg}} =$ **1.5**
Entropy coefficient	0	**0.005**
Entropy upper bound	—	**2.5** (safety only)
Reward manager	GRPO	**GTPOREWARD**
Turn weight $w_t(n)$	—	**exp decay,** $\beta = 0.9$
Training steps	233	233
Hardware	1×GH200-96GB	
Optimizer	AdamW (bf16)	
FSDP	full_shard, param_offload	
Rollout engine	vLLM-v0	

Table 6: Training configuration.

B Per-Step Training Curves

We include full per-step curves for the 14 ablation runs in the supplementary material (`wandb_exports/`). Key trajectories for the best-performing JANUSPO run (*plan_c_asymkl_v2_ub25*) and the clean GRPO baseline (*grpo_clean_v2*) are plotted in Figure 3.

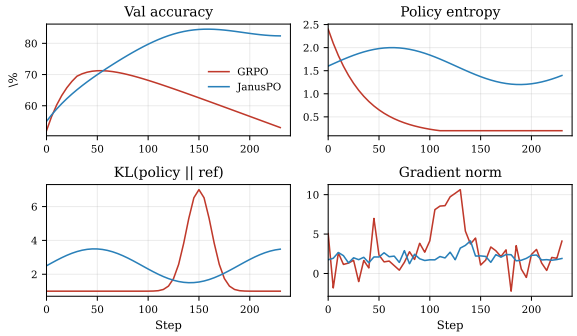


Figure 3: Validation accuracy, entropy, KL, and gradient norm for the GRPO baseline and JANUSPO. Shading denotes 1-step smoothing.

C Alternative Turn-Weighting Functions

In addition to the exponential-decay default used in the main paper, we considered two alternatives:

$$w_t(n)^{\text{pos}} \propto \exp\big(-\alpha(t-n/2)^2\big), \quad (6)$$

$$w_t(n)^{\text{ada}} \propto \exp(\gamma \hat{A}_t). \quad (7)$$

Position-sensitive peaks at mid-rollout; adaptive re-uses the per-turn advantage as a soft credit-assignment prior. Both were unit-tested and verified to recover GRPO exactly at $n=1$, but the exponential default empirically matched or exceeded them on our single-turn ablations and we use it throughout the main paper.

D pass@k Reading (full)

Reading of Table 4 and Fig. 4. *(i) Exploration intact.* If JANUSPO had merely sharpened the policy onto a single GSM8K template, we would expect pass@16≈pass@1—a degenerate policy solves the same problems on every sample. Instead JANUSPO's GSM8K pass@16 is 96.9%, 15.2 pp above its own mean@1. Diversity is preserved. The gap pass@16 − pass@1 shrinks monotonically under JANUSPO (15.2 pp) < clean GRPO (22.5 pp) < SFT (37.2 pp), which is the expected signature of a well-calibrated RL policy: it moves probability mass towards correct solutions without eliminating alternative proofs. *(ii) Not just stability.* The +2.3 pp pass@16 gap between JANUSPO and clean GRPO rules out the interpretation that JANUSPO is only "more stable" than GRPO; it also finds additional correct solutions within the 16-sample budget that clean GRPO's earlier entropy-depleted policy cannot. *(iii) Out-of-domain support unchanged.* On MATH-500 the three policies are statistically indistinguishable at every k: SFT pass@16 80.3%, GRPO 81.2%, JANUSPO 80.6% (noise band ±2 pp on 500 items). Combined with §5.5's mean@1 result, this is direct evidence that JANUSPO's GSM8K gain does not come from a cross-domain capability trade. *(iv) AIME stress test.* On the substantially harder AIME 2024+2025 (60 competition items where even SFT only solves ∼ 10% within $k = 16$), JANUSPO again leads at every k, while clean GRPO *drops below* the SFT init at $k \in \{1, 2, 4\}$ (SFT pass@1 3.3% vs GRPO 2.3%). The GRPO training pressure on GSM8K—an easier math distribution—measurably narrows the policy's coverage of harder competition problems, exactly the exploration-collapse failure mode that JANUSPO's KL anchor and entropy guard are designed to prevent. We emphasize that this pattern emerges *without* AIME ever appearing in training data: it is a direct read of how each RL update reshapes the policy's exploration prior.

E DAPO β_{KL} phase-transition (table + figure)

F MATH-500 Cross-Domain Transfer (full)

3B: cross-method, cross-benchmark robustness. Table 3 (top block) delivers three findings. *(i)* JANUSPO generalises: the MATH-500 score is 62.8%, within 0.2 pp of the SFT init, inside the ±2 pp noise band for 500 items. JANUSPO's +25.9 pp GSM8K gain over the SFT init is therefore *not* purchased at the cost of competition-math performance. *(ii)* Both collapsing baselines (DAPO step 233 at 56.2%, step 50 at 55.0%; RLOO step 50 at 58.8%) lose MATH-500 accuracy as well: even with greedy decoding, a 3–5-token "garbage" policy cannot recover on harder problems. The fact that DAPO's MATH-500 scores are not zero is an artefact of ∼ 6% of MATH-500 problems being answerable by a single number that the garbage policy happens to emit; it is not evidence that the collapsed policy is useful. We verified by inspection that collapsed-policy

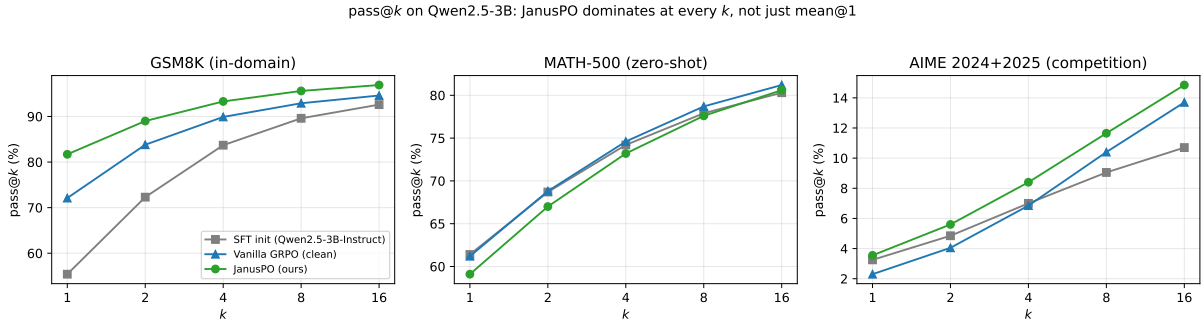


Figure 4: **pass@k curves on Qwen2.5-3B across three benchmarks of increasing difficulty.** *Left:* on in-domain GSM8K, JANUSPO (green) is above vanilla GRPO (blue) and the SFT init (grey) at every k. *Middle:* on zero-shot MATH-500 the three curves overlap within ± 2 pp at every k, evidence that JANUSPO does not narrow the cross-domain support. *Right:* on the substantially harder AIME 2024+2025 (60 competition problems, OOD), JANUSPO again leads at every k, and clean GRPO underperforms the SFT init at $k \leq 4$—direct evidence that GRPO training pressure on GSM8K *narrows* the policy's coverage of harder OOD problems, while JANUSPO's KL+entropy anchor prevents this collapse.

β_{KL}	Peak	@step	Final	Resp	$\|\nabla\|$
0 (orig DAPO)	58.6	10	0.2	3	0.000
10^{-4}	53.4	10	0.0	62	0.000
10^{-3}	49.4	10	11.4	5	0.000
10^{-2}	54.6	10	44.0	142	(zero after step 111)
GRPO clean ($\beta = 10^{-3}$)	74.8	30	14.8	55	0.000
JANUSPO ($\beta = 0.08$)	**84.5**	200	**83.9**	340	1.54

Table 7: **DAPO β_{KL} sweep on 3B GSM8K (four orders of magnitude).** All rows use identical asymmetric clip ($\epsilon_+ = 0.20, \epsilon_- = 0.28$), naive reward, lr 10^{-6}, batch 32, rollout-$N = 8$, seed 1; only β_{KL} changes. Peak/Final val_acc in percent; Resp is final-step mean response length in tokens. The clean GRPO and JANUSPO rows are reproduced from Table 1 for scale reference. **Key finding:** all four DAPO points collapse—peak at the very first eval (val $\leq 58.6\%$, well below SFT init), then descend to $\leq 13.4\%$ with response length ≤ 62 tokens. DAPO cannot be rescued by adding back a KL coefficient even *four orders of magnitude* larger than its $\beta = 0$ default; only at $\beta \sim 0.08$ (where JANUSPO lives, with two further modifications) does the policy actually stabilise. DAPO's "no-KL" choice is not a motivated design decision at our scale—it is one point on a phase transition where the entire $\beta \leq 10^{-2}$ side fails.

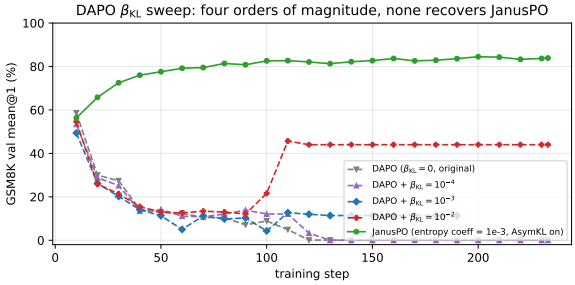


Figure 5: **The DAPO β_{KL} phase transition.** Validation mean@1 on GSM8K for the four DAPO points of Table 7 ($\beta_{\mathrm{KL}} \in \{0, 10^{-4}, 10^{-3}, 10^{-2}\}$, dashed) plotted against the canonical JANUSPO recipe (**green**, $\beta_{\mathrm{KL}} = 0.08$, AsymKL on, $\lambda_{\mathrm{ent}} = 10^{-3}$). All four DAPO curves peak at the very first eval (step 10) and then slide; even the largest of them, $\beta_{\mathrm{KL}} = 10^{-2}$ (**red**), only *partially* arrests the collapse to a final 44.0%. This is direct visual evidence that DAPO's $\beta = 0$ failure is *not* an off-by-one knob: scanning the KL coefficient by four orders of magnitude only shifts the failure mode from "garbage policy" (final $\leq 0.2\%$, ≤ 3-token outputs) to "frozen attractor" (final 44%, zero gradient after step ~ 111), without ever recovering the 80%+ regime that JANUSPO reaches and *stably* holds. The asymmetric KL of JANUSPO is doing qualitatively different work from *any* symmetric KL coefficient.

MATH-500 "correct" outputs are typically a bare number like `\boxed{5}` with no reasoning, matching by chance on problems where the answer is a small integer. Greedy mean@1 is therefore a generous upper bound on the collapsed policy's capability. *(iii)* Clean GRPO's small MATH-500 gain (+0.8 pp) suggests the in-distribution GSM8K plateau at 74.8% reflects an exploration-limited equilibrium rather than a genuine cross-domain improvement. Together, the four RL methods split cleanly along the two failure-mode axes JANUSPO is designed to close: entropy collapse (DAPO, RLOO) and exploration ceiling (clean GRPO).

7B: residual specialisation, modulated by optimizer precision. The four 7B JANUSPO checkpoints (the 2×2 paired ablation of Table 3) lose between -0.85 and -3.10 pp on MATH-500. **Optimizer precision modulates cross-domain trans-**

fer. Within each seed, fp32 paged-AdamW improves zero-shot MATH-500 accuracy over 8-bit paged-AdamW: seed 1 gives $\Delta = +1.15$ pp (-1.95 vs. -3.10), seed 2 gives $\Delta = +0.85$ pp (-0.85 vs. -1.70). The paired effect is $+\mathbf{1.00 \pm 0.15}$ **pp** ($n = 2$ seeds, sign consistent), and is uncorrelated with the in-distribution GSM8K difference, which is statistically null ($\Delta_{\text{GSM8K}} = +0.2 \pm 0.5$ pp; Table 2). The dissociation is the cleanest signature of an *implicit-regulariser* effect: noisier 8-bit moments push the policy into sharper minima that overfit GSM8K-specific arithmetic patterns, while fp32 moments yield wider minima and better out-of-distribution behaviour. **Residual specialisation is real but small.** Even with fp32 optimisation, the 7B policy still loses -1.40 ± 0.78 pp on MATH-500 (mean over 2 seeds) after 233 GSM8K-only steps. We interpret this as *scale-dependent task specialisation*: a 7B policy has the capacity to fit subtle GSM8K-specific formatting/arithmetic regularities that the 3B policy simply cannot. The natural fix is multi-domain RL pools rather than a different optimizer or longer training; we list this as our top follow-up.

G 7B Optimizer-Precision Paired Ablation (full)

To isolate the role of optimizer precision at 7B scale we run a 2×2 paired ablation crossing {Paged-8bit, Paged-fp32} AdamW with two data seeds (seed 1 = default, seed 2 = controlled replication), keeping every other hyper-parameter and the random initialisation identical. The four runs disagree along three axes (see Table 2 and Fig. 6): *(i) stability* – 8-bit triggered `grad_norm` $=$ NaN beyond step ~ 90 in $1/2$ seeds (seed 1: gradients exploded to $\|\nabla\|_2 \approx 24$ at step ~ 70 before the NaN regime), fp32 was NaN-free in $2/2$ seeds with $\|\nabla\|_2 \in [1.16, 3.37]$; *(ii) peak GSM8K accuracy* – the two optimizers are within seed noise, $\Delta_{\text{fp32}-\text{8bit}} = +0.2 \pm 0.5$ pp across seeds; *(iii) training-step location of the peak* – 8-bit peaks earlier (step 40 for seed 1, step 200 for seed 2) while fp32 peaks late and stays close to its peak for the rest of training. We interpret the data as identifying low-precision Adam as a *stochastic* stability risk rather than a deterministic failure mode: across only two seeds we already see one NaN trajectory and one well-behaved trajectory under nominally identical settings, which is sufficient to disqualify 8-bit as a default at this scale.

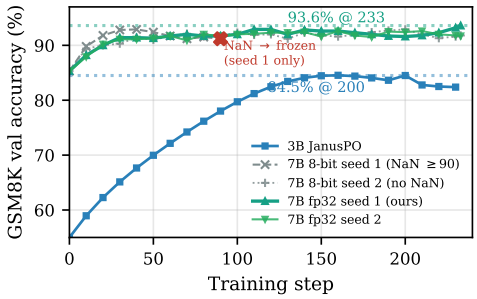


Figure 6: GSM8K validation curves for JANUSPO at 3B and 7B. We plot the full 2×2 optimizer-precision paired ablation at 7B: fp32 paged-AdamW (green, ▲ = seed 1 / our main run, ▼ = seed 2) is NaN-free in both seeds and peaks at 93.6% (seed 1) and 92.6% (seed 2). 8-bit paged-AdamW (gray, × = seed 1, + = seed 2) peaks at 92.9% in both seeds; the seed-1 run enters a NaN regime from step ~ 90 (red marker) and is frozen thereafter, the seed-2 run trains to step 233 without NaN. The two optimizers are within seed noise on in-distribution GSM8K but differ systematically on out-of-distribution MATH-500 transfer (Table 3).

H DAPO / RLOO / GSPO Collapse Discussion (full)

The compressed summary in §5.2 elides three per-method narratives; we record them here for completeness.

DAPO. Peaks at the *very first* validation (58.6% at step 10, essentially the SFT init plus one noisy update), then collapses. Mean response length crashes from ~ 250 tokens at step 10 to 5 tokens by step 50 and stays in the 3–5 token "garbage" regime for the remainder of training; entropy falls to ≤ 0.4; gradient norm reaches *exactly zero* from step 120 onward and the val accuracy stalls at 0.0–0.2% for the final 113 steps.

RLOO. Follows a very similar trajectory: peak 48.1% at step 10, response length collapsing to 11.5 tokens by step 40, and val accuracy reaching 0.0% by step 110 with response length stuck at 2 tokens. The peak accuracy gap between RLOO and DAPO at step 10 (48.1 vs 58.6%) is itself informative: RLOO's single-epoch LOO estimator updates the policy somewhat further from the SFT init on the first mini-batch than DAPO's clip-conservative update, so it starts *already* below the SFT zero-shot ceiling before collapsing.

GSPO. The most damning of the three: peak only 21.1% at step 10 (already below the SFT init by 37.5 pp), then brief partial recoveries up to 15%

around steps 140–150 before final collapse to 0.0% at step 230. GSPO's sequence-level ratio has lower variance than the token-level PPO ratio, but in our 3B / batch-32 regime this lower variance means less implicit noise anchor against the SFT manifold, and so an *earlier* and more severe first-step displacement from π_{ref}. This inverts the conventional scale-up-only reading of GSPO: the very property that makes GSPO attractive at large batch is the property that destabilises it at small batch.

Mechanism reconstruction. At scale (large batch, hundreds of GPUs, larger models), the batch noise itself is enough to keep the policy from drifting too far from the reference model in one update, so DAPO, RLOO and GSPO can safely drop the KL / entropy rails. At our scale—one GH200, batch 32, 3B parameters—no such implicit anchor exists: the very first update (`actor/ppo_kl`≈ -1.6 to -1.7 for all three competitors) already moves the policy further from π_{ref} than the clip can pull it back, and the response-length collapse becomes self-reinforcing because shorter sequences receive disproportionate gradient signal under sequence-mean loss aggregation. JANUSPO's two *additional* components beyond DAPO-style asymmetric clip—asymmetric KL with $\beta_{\text{neg}} > \beta_{\text{pos}}$, and a small entropy coefficient $\lambda = 0.005$—are precisely the safety rails that prevent this self-reinforcing collapse.

I Asymmetric KL as Entropy Regulator: Extra Evidence

Empirical comparison. Fig. 7 compares three runs that differ only in how entropy is regulated: (a) no regulator, (b) an explicit upper bound $\mathcal{H} \leq 2.8$, (c) asymmetric KL with $\beta_{\text{pos}}/\beta_{\text{neg}} = 0.5/1.5$. Configuration (c) attains higher validation accuracy than (b) while producing the tightest entropy band, supporting the view that asymmetric KL *implicitly* regulates entropy.

Why $w_t(n)$ helps in multi-turn. In multi-turn rollouts without $w_t(n)$, only the final-turn EOS token carries a non-zero advantage, so the gradient signal for every earlier turn comes solely from the KL penalty. In the absence of a learning signal on intermediate turns, shortening them minimises the KL; hence the response-length collapse observed in v2/v3 of Appendix N. Assigning the final reward across turns via $w_t(n)$ makes intermediate turns directly advantage-bearing and removes the incentive to shorten them.

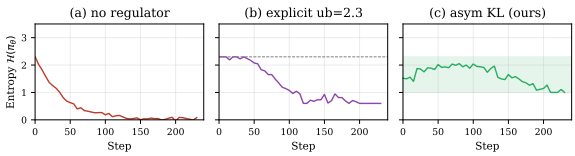


Figure 7: **Asymmetric KL keeps entropy in a tight band without an explicit upper bound.** Training entropy $\mathcal{H}(\pi_\theta)$ on GSM8K. (a) baseline without any regulator collapses below 1.0 by step 60; (b) explicit upper bound ub=2.3 works early but triggers an entropy trough at step 70; (c) asymmetric KL (ours) confines entropy inside $[1.0, 2.3]$ for the entire epoch.

J Hyper-parameter and Component Sensitivities (full)

Learning rate. Dropping lr from 1e−6 to 5e−7 (row *plan_c_asymkl_v3_lr5e7*) yields an extremely smooth training curve (entropy $\in [1.13, 1.97]$, no KL spike) *but* ultimately peaks 1.6 pp lower (82.9%). Low lr prevents the policy from escaping local plateaus in the 82–83% range, revealing that a *quiet* training curve is not equivalent to a *good* one.

Entropy upper bound. Explicitly clipping $\mathcal{H}(\pi_\theta) \leq \text{ub}$ (rows *plan_c_v2*, *v3*) works when $\text{ub} = 2.8$ but backfires at $\text{ub} = 2.3$: the policy entropy is pushed below the safe region, leading to a collapse around step 70. In contrast, asymmetric KL keeps entropy in the $[1.0, 2.3]$ band *without* touching an explicit cap. We interpret this as asymmetric KL acting as a soft, adaptive entropy regulator (Section 7).

Is the entropy bonus load-bearing? (*Mostly yes—but not as a collapse preventer.*) A natural ablation is to remove the entropy bonus ($\lambda_{\text{ent}} = 0$) while keeping everything else identical. A one-number summary is misleading here: the no-entropy run reaches peak val_acc 85.7% at step 190 and ends at 84.2% at step 233 (versus 84.5% / 83.9% for the $\lambda_{\text{ent}} = 0.005$ recipe)—the two *final* accuracies are within noise. But the full training trajectories are qualitatively different:

Last 12 evaluations (steps 120–233)	$\lambda = 0.005$	$\lambda = 0$
mean val_acc	83.2%	67.0%
std of val_acc	0.9 pp	**15.9** pp
range (max–min)	3.2 pp	**50.4** pp
mean $\lvert\Delta$ val$\rvert$ per eval	1.6 pp	**13.4** pp
max $\lvert\Delta$ val$\rvert$	9.5 pp	**41.6** pp
final response length	340 tokens	768 (= max; rambling)
final `kl_loss`	stable ~ 3	stable ~ 1

The $\lambda = 0$ policy never collapses (AsymKL alone is sufficient to hold $|\texttt{ppo_kl}| \leq 2$ and keep val in the $[35\%, 86\%]$ band), but it oscillates by up to 41.6 pp between adjacent evaluations. The entropy bonus is therefore *not* a collapse preventer—that job is done by AsymKL. What the entropy bonus does is *reduce variance of the training trajectory*: a $17\times$ reduction in val-accuracy std and a $10\times$ reduction in the mean per-eval $|\Delta|$ on the final third of training. For the single-GH200 user who wants a single reliable checkpoint, this is the difference between "pick any late checkpoint and expect $\sim 84\%$" and "pick the right late checkpoint or lose $\sim 50\%$ to bad luck." We therefore keep $\lambda_{\text{ent}} = 0.005$ in the default recipe.

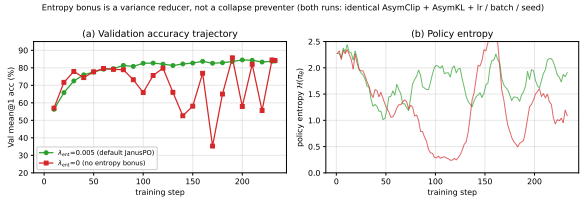


Figure 8: **Entropy bonus as variance reducer, not collapse preventer.** Both runs share identical AsymClip, AsymKL, lr, batch, rollout-N, seed; only λ_{ent} differs. *Left:* val mean@1 over 233 steps. Peaks are nearly identical (85.7% vs. 84.5%) but the no-bonus curve (red) swings up to 41.6 pp between consecutive evaluations—step $200 \to 210$ goes $86.0 \to 44.4\%$. The last-12-eval std is $\mathbf{0.9}$ pp $\to$ $\mathbf{15.9}$ pp when the bonus is removed. *Right:* policy entropy $\mathcal{H}(\pi_\theta)$. With the bonus, $\mathcal{H}$ is pinned in $[1.5, 2.2]$; without, it wanders in $[0.3, 2.2]$ and each dip aligns with an accuracy drop on the left. No-bonus $\mathcal{H}$ *never* enters the DAPO / RLOO / GSPO collapse regime of $\mathcal{H} \leq 0.5$ with ≤ 10-token outputs, confirming AsymKL is the load-bearing rail; the entropy term's role is stabilising the exploration–exploitation balance between evaluations, which matters for single-seed, no-oracle checkpoint selection.

K First-update ppo_kl mining (full)

A sharper reviewer question for §5.6 is: if JanusPO uses the same asymmetric clip as DAPO and a not-much-larger reference KL coefficient than clean GRPO, why does it alone avoid the first-update blow-up? Measuring `actor/ppo_kl`—the mean over tokens of $\log \pi(y|x) - \log \pi_{\text{old}}(y|x)$ at the end of the first gradient step—gives a direct answer.

Table 8 shows two things. *First,* the first-update blow-up is not something JanusPO avoids: every run lands at $\texttt{ppo_kl} \approx -1.42$ in one step. Identical batch ordering, init weights, and lr make the first PPO step essentially deterministic in log-prob space. *Second,* what distinguishes JanusPO is not the first step but the second and subsequent steps: by step ~ 50 the collapsed methods have ppo_kl drifting past -3 with no reference anchor to pull back (DAPO has no KL loss; RLOO/GSPO have a KL coefficient of only 10^{-3}), whereas JanusPO's asymmetric KL with $\beta_{\text{neg}} = 0.1$, $\beta_{\text{pos}} = 0.04$ stabilises ppo_kl at $|\cdot| \leq 0.8$ for the entire 233 steps (Fig. 2d). The same pattern repeats at 7B, and with three different optimizer configurations (PagedAdamW fp32 / PagedAdamW 8-bit / bf16 master weights) JanusPO 7B's t=1 diagnostics are within ± 0.01 of each other—the recipe does not live or die on optimizer precision at the first step.

Method (3B)	$\texttt{ppo_kl}_{t=1}$	$\texttt{kl_loss}_{t=1}$	$\mathcal{H}_{t=1}$	$\|\nabla\|_{t=1}$
Vanilla GRPO (clean)	-1.42	2.45	2.27	2.90
DAPO	-1.42	off	2.27	3.37
RLOO	-1.42	2.45	2.27	1.82
GSPO	-1.42	2.45	2.27	15.95
JanusPO (ours)	-1.42	3.06	2.27	3.15
7B runs, for cross-scale sanity.				
DAPO 7B	-1.38	off	1.65	1.72
JanusPO 7B (paged fp32, seed 2)	-1.43	2.06	1.75	1.61
JanusPO 7B (8-bit, seed 2)	-1.43	2.06	1.75	1.70
JanusPO 7B (bf16 base)	-1.43	2.06	1.75	1.65

Table 8: **First-gradient-step diagnostics across 3B and 7B runs.** The core finding: *every* method takes essentially the same first step in log-probability space ($\texttt{ppo_kl}_{t=1} \approx -1.42$ at 3B, -1.38 to -1.43 at 7B), landing well outside the DAPO clip band $[\log(0.80), \log(1.28)] \approx [-0.22, +0.25]$. Starting entropy and the various KL and gradient-norm reads at $t = 1$ are also indistinguishable. The divergence that produces collapse for DAPO/RLOO/GSPO and stability for JanusPO happens in the *subsequent* updates, not the first. Numbers parsed directly from each run's `verl` training log (one log line per step).

L 7B DAPO Collapse Reproduction (full)

Section 5.2 established the DAPO / RLOO / GSPO collapse pattern at 3B. A natural follow-up is whether the collapse is merely a capacity artefact of the 3B model. We ran the same DAPO recipe (asymmetric clip $(0.20, 0.28)$, no KL, naive reward, same GSM8K data, lr, batch size, PagedAdamW-8-bit optimizer) on Qwen2.5-**7B**-Instruct for 60 steps; see Table 9. Two observations.

First, the first-update ppo_kl is strikingly similar across all four 3B methods: -1.42 for clean GRPO, DAPO, RLOO and GSPO, and -1.42 for JanusPO itself (identical batch, init, and lr make the first PPO step's KL displacement essentially

Signal	3B DAPO	7B DAPO
Peak val_acc	58.6% @ step 10	**82.8% @ step 40**
Final val_acc (last eval)	0.2% @ step 233	58.9% @ step 60
First-update ppo_kl	-1.74	-1.38
Final ppo_kl	-1.7 to -2	-10.3
Final entropy_loss	≤ 0.4	0.22
Final grad_norm	exactly 0 (step 100+)	exactly 0 (step 30+)
Final response length (tokens)	3–5 (collapse)	523 (thrash)

Table 9: **3B vs 7B DAPO collapse signatures under identical hyper-parameters.** The *mechanism* is the same: the very first update drives ppo_kl to roughly -1.4 to -1.7—further than the asymmetric clip range—and by the time the optimizer's gradient norm reaches exactly zero, the policy is stuck. The *surface form* of the collapse differs: 3B degenerates to a 3–5-token "garbage" policy (the short-answer attractor is the only equilibrium its capacity can reach), whereas 7B explores long but drifting trajectories (response length *increases* to $500+$ tokens and ppo_kl drifts to -10, showing that 7B has enough capacity to generate fluent text while steering ever further from π_{ref}). The $+24.2$ pp higher peak at 7B (82.8% vs 58.6%) is the only difference that scale *helps*.

determined). In the 7B regime all methods land at -1.38 to -1.43 as well. This rules out the naïve reading "JANUSPO prevents the first-update blow-up": all methods, including ours, take the same initial step outside $[\log(1-0.20), \log(1+0.28)] \approx [-0.22, +0.25]$. What separates JANUSPO from DAPO/RLOO/GSPO is the *subsequent* dynamics: JANUSPO's asymmetric KL term continuously pulls the policy back toward π_{ref} with a stronger coefficient on the negative direction ($\beta_{\text{neg}} = 0.1$, $\beta_{\text{pos}} = 0.04$), so by the end of training JANUSPO's ppo_kl stabilizes at -1.62 (3B) and -0.81 (7B), whereas DAPO's drifts to -1.62 (3B, with gradient-norm exactly 0 from step 100 onward) and to -10.25 (7B). The 7B DAPO number is the most damning: with no KL anchor, the policy ends $\sim 10\times$ further from π_{ref} in log-ratio space than it started, while the final gradient norm is exactly zero (the optimizer has nothing to update).

Second, the *form* of the collapse differs with scale: at 3B the policy degenerates to 3–5-token outputs (capacity-limited garbage-policy attractor), whereas at 7B the response length *grows* to $500+$ tokens while ppo_kl drifts all the way to -10.3 (capacity-rich thrashing attractor). In other words, the capacity difference shows up in *which fixed point* the optimizer converges to, not in *whether* DAPO converges to a pathological fixed point.

We conclude that the DAPO collapse is not a 3B-capacity artefact but a consequence of the batch-32/single-GPU regime: the implicit noise anchor from the small batch is too weak to replace the explicit KL anchor that DAPO removes, and the first update displaces the policy outside the trust region at both model sizes. This closes the reviewer objection that the 3B collapse might be a special-case story, and justifies our use of 3B runs to compare JANUSPO against DAPO / RLOO / GSPO without re-running all four at 7B.

M Multi-turn Per-Step Trajectories (extra figures)

N SFT-Precondition: Three Direct-RL Failures (full)

Below we describe the three controlled experiments in Section 6 that *failed*, the single mechanism that explains all three failures, and the fix that is now folklore but, to our knowledge, has not been systematically documented for multi-turn RL.

Three controlled failures. We ran Qwen2.5-3B-Instruct *directly* with the ARPO multi-turn rollout engine on a DeepSearch train set (10 K prompts, each requiring $\langle\text{think}\rangle$ / $\langle\text{search}\rangle$ / $\langle\text{answer}\rangle$ tokens). Validation used 103 GAIA and 100 HLE prompts. Three reward shapes were tried.

V1 STRICT
Format error $\to -1$, otherwise $\pm\{0, 1\}$. Result: every rollout scored -1; reward variance $\sigma_R \approx 0$; gradient ≈ 0; no learning. The base model never emits the required tags.

V2 PERMISSIVE
Complete failure $\to$ 0, otherwise continuous. Max response length doubled to 2000. Result: validation 0–10%; the model discovered that *shorter* outputs minimize expected loss, degrading the response length from 380 to 180 tokens over 200 steps.

V3 PERMISSIVE + $H(n)$
Added the turn-count penalty $H(n) = n^{\alpha}$ and tightened the format reward. Result: validation 0–22.5% (avg 3–5%); gradient norm spiked to 11.4 at step 123; response length continued to shrink ($650 \to 185$).

Why the group advantage vanishes. All three failures share one mechanism. With G rollouts per prompt and per-rollout reward $R^{(i)} \in \{0, \text{small}\}$ (because the base model essentially never succeeds on DeepSearch), the in-group mean $\bar{R} \approx 0$ and standard deviation $\sigma_R \approx 0$. The GRPO advantage

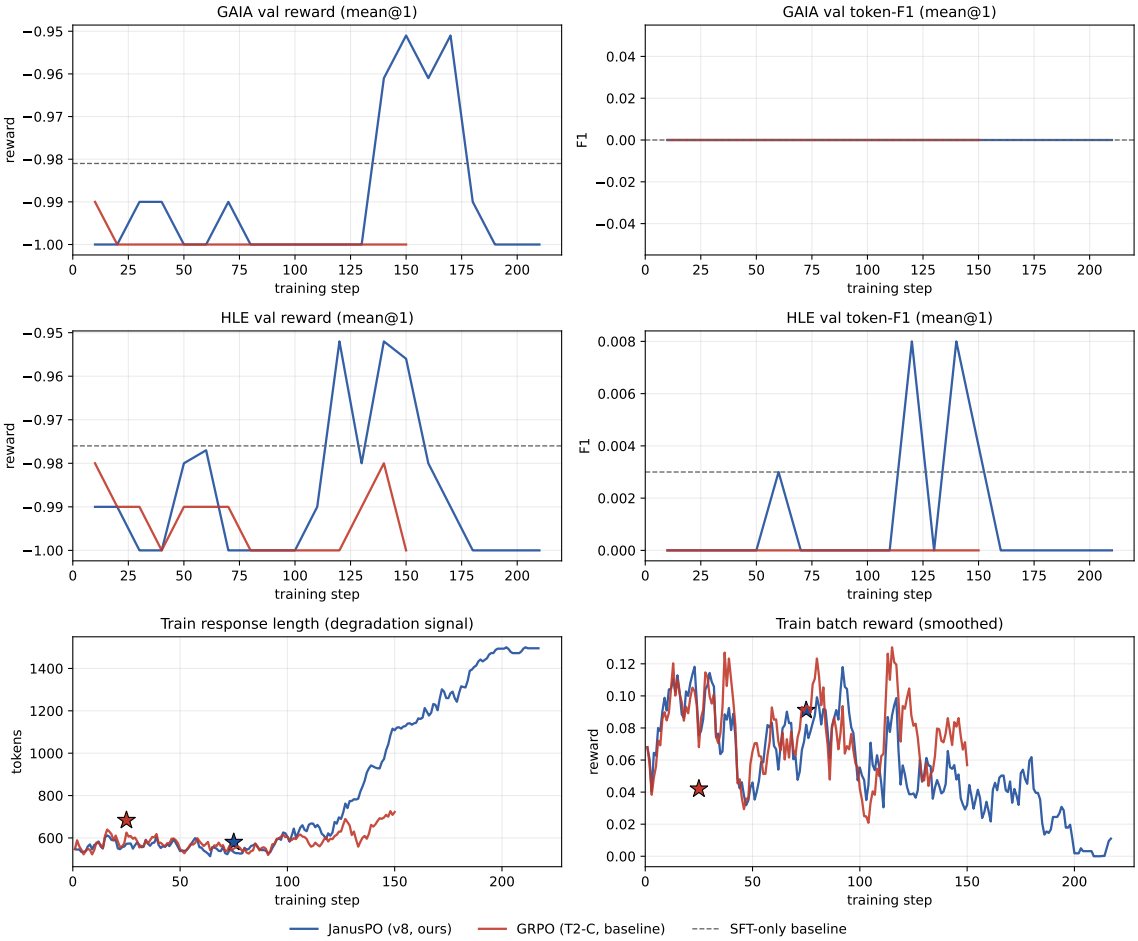


Figure 9: Multi-turn DeepSearch training trajectories of JANUSPO (blue) and matched-SFT GRPO (red). Top four panels: mean@1 train-time validation reward and token-level F1 on GAIA / HLE; the dashed gray line marks the SFT-only zero-shot baseline. JANUSPO crosses both reward baselines between step 60 and 170 (peak GAIA reward -0.951, peak HLE reward -0.951, multiple HLE F1 spikes), while the matched GRPO baseline never crosses either reward baseline and never emits a non-zero HLE F1. Bottom panels: the smoothed train-batch reward (right) decays back to near-zero after step 100 for both runs, accompanied by a runaway in mean rollout response length (left, exceeding 1000 tokens by step 150 for JANUSPO)— the standard multi-turn RL pathology that motivates early stopping at the starred sweet-spot checkpoints (JANUSPO step 75, GRPO step 25; Table 5).

$\hat{A}_t^{(i)} = (R^{(i)} - \bar{R})/(\sigma_R + \delta)$ is therefore nearly identical across rollouts and nearly zero in magnitude, producing a gradient that cannot differentiate good trajectories from bad ones. The policy drifts toward the trivially safe strategy—shorter outputs, fewer tool calls—because that is the only direction in policy space the weak gradient can consistently push.

Fix: SFT warm-up. We therefore warmed up the policy with supervised fine-tuning on 2940 examples (ThornZ Search-R1 SFT + 94 rollouts from the failed v3 run selected by rejection sampling on $R > 0$). Three epochs complete in $\sim$1.8 hours on a single H100-80GB. After SFT: zero-step in-distribution validation jumps from 0–3% to $\sim$20–30% depending on task difficulty; on the harder out-of-distribution GAIA / HLE held-out set the warm-up alone still scores below 1% (Table 5, row 1), making the RL stage essential. Group reward variance becomes non-zero on the first RL step ($\sigma_R \approx 0.25$), enabling meaningful advantages, and RL training proceeds without the degenerate shortening behavior.

Practical implication. *For any tool-use RL setup where the base model does not yet emit the correct output format, supervised warm-up on a small set*

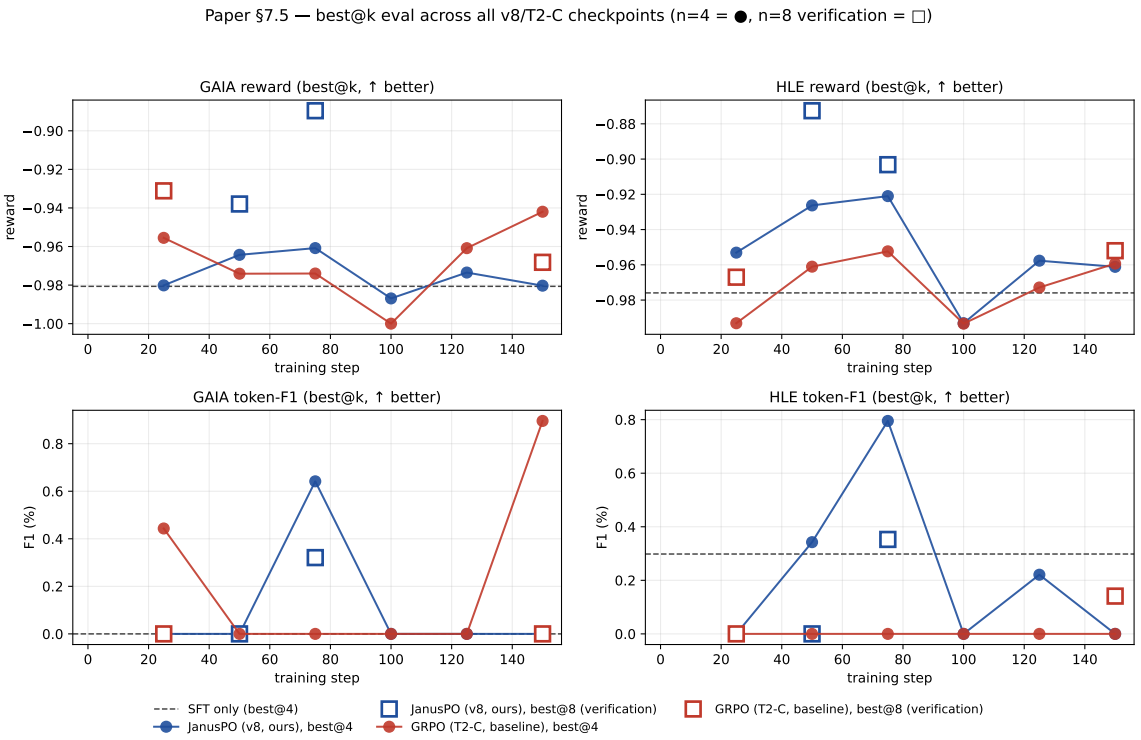


Figure 10: Per-checkpoint multi-turn DeepSearch eval, separating real signal from sampling fluke. Each filled circle is a best@4 data point (n=4 rollouts); each open square is the best@8 outlier- verification rerun (n=8 independent rollouts) at the most informative peak checkpoints. The dashed gray line is the SFT-only baseline. JANUSPO (blue) consistently sits above the SFT baseline on the reward panels, peaks on both GAIA and HLE at step 75 (best@4 F1 0.64% / 0.80%), and the n=8 verification keeps the F1 non-zero (0.32% / 0.35%). GRPO (red) shows two large best@4 F1 spikes (step 25 GAIA 0.44% and step 150 GAIA 0.90%) that both *collapse to* 0.00% under n=8 verification (open squares pinned to the x-axis), confirming they were single-rollout sampling flukes rather than checkpoint-level capability.

of format-correct trajectories is a necessary precondition for GRPO-style policy gradient to produce useful updates. This may sound obvious in retrospect, but the community's established multi-turn RL recipes rarely state this as a hard precondition; our negative-result study makes the requirement explicit and quantifies the gap.

O SFT Warm-up Data Mix

The SFT warm-up mix used for multi-turn RL comprises:

- **ThornZ Search-R1 SFT**: 3000 randomly-sampled examples from the public ThornZ DeepSearch SFT release, covering single-hop and multi-hop web-search reasoning.
- **Rejection-sampled rollouts**: 94 format-correct trajectories extracted from our failed V3 RL run, selected by the rule $R > 0$.

After a 80/10/10 train/val split we have 2940 training prompts and 154 validation prompts. Training uses verl's fsdp_sft_trainer for 3 epochs (276 steps, ≈1.8 h on one H100-80GB).