FROM ROLLOUT SCORES TO POLICY UPDATES: ESTIMATOR COMPLETION FOR AGENTIC RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic reinforcement learning often scores groups of policy-generated rollouts before selecting which ones enter an update. Selection compresses graded priorities into a mask, leaving open how the discarded score mass should affect the estimator. We formulate this missing step as *Estimator Completion* (EC). Given a scorer and an optimizer's native aggregation on selected groups, conserving the selected conditional, the scorer's tail conditional, and their partition mass uniquely determines a completed group measure. It is a KL projection of the score distribution, differs from hard filtering by exactly the omitted mass, and yields an explicit head-recalibration and tail-restoration correction. Reward-dispersion filtering provides a concrete test: logged batches retain ordered scores and reward-active rejected groups, and we carry EC's score-conditioned coordinates into PPO, GRPO, DAPO, and DrGRPO policy reductions. Across four tasks, all four Qwen2.5-3B EC variants improve the four-task average of peak-validation means over RAGEN-2; EC-PPO does so across all six tested models. This gives graded rollout selection a principled estimator interface beyond its keep/drop rule.

1 INTRODUCTION

Agentic reinforcement learning improves policies with trajectories generated through interaction, where feedback quality varies across prompts and environments (Wang et al., 2025b; Wang & Ammanabrolu, 2025). Policy updates depend on both the experience admitted to training and the reduction applied to its losses. PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and DrGRPO (Liu et al., 2025) prescribe different reductions. Recent rollout filtering targets the admitted experience: RAGEN-2, for example, ranks groups by reward dispersion and retains a high-priority subset (Wang et al., 2026).

These developments leave a consequential choice between selection and optimization. A filter computes graded priorities, but passes a selected set to the policy loss:

$$\text{rollouts} \longrightarrow \text{graded scores } \pi \longrightarrow \text{selected set } S \longrightarrow \text{gradient estimator.} \tag{1}$$

The score-to-set map is many-to-one. Two batches can select identical groups even if rejected groups carry different score mass and ordering; a hard-filtered update treats them alike. Top-p specifies which groups receive priority, but the mask alone does not determine the estimator's treatment of the omitted mass or the scale of its selected term. This ambiguity arises whenever a graded reward, uncertainty, or difficulty signal becomes a keep/drop decision.

We resolve it at the level shared by these optimizers. *Estimator Completion* (EC) takes the native selected aggregation as fixed, retains the scorer's relative priorities among rejected groups, and conserves the scorer's total mass on each side of the mask. The three requirements yield one reference distribution and an exact correction with two interpretable parts: recalibrating the selected mass and restoring tail support. Its KL characterization, information-loss bound, and minimal retained score state show precisely what the filter must pass to the optimizer.

We examine this question with the reward-dispersion filter of RAGEN-2, using the same softmax top-p setting for both the filtered comparator and our variants. Logged score profiles and tail reward activity reveal information omitted by the mask. Figure 1 shows how the completion coordinates enter native policy reductions.

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

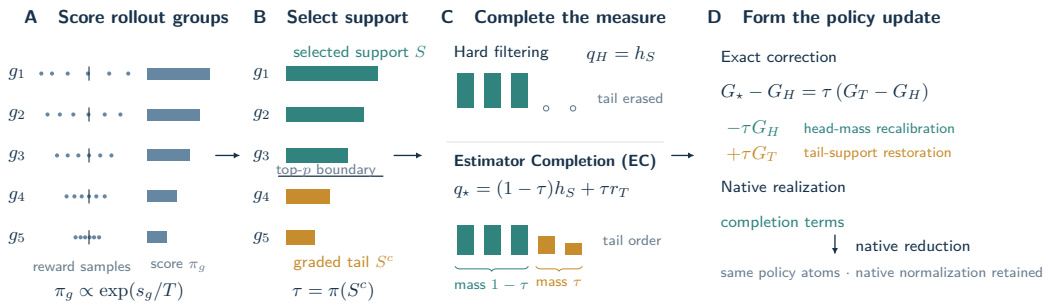

Figure 1: Schematic of the group-level reference construction, illustrated with a reward-dispersion scorer. Top-p turns graded priorities into selected support. The optimizer's selected aggregation, scorer's conditional tail, and head–tail mass determine a reference completed measure. Native policy-loss variants place these coordinates within their own reductions.

Our contributions are:

- We identify *score-to-estimator compression* in post-rollout filtering and quantify the information lost when graded priorities become a selected set.
- We derive a unique mass-consistent reference estimator, its KL projection and minimal retained score state, and the exact correction to hard filtering.
- We connect the reference construction to four native policy reductions. Across four tasks, every Qwen2.5-3B optimizer variant improves the four-task average over RAGEN-2; the PPO variant does so for all six tested models. Full results are in the appendix.

2 Problem Formulation

2.1 The score–support–estimator chain

Consider one policy update containing rollout groups $B_1, \dots, B_G$, and write $[G] = \{1, \dots, G\}$. Before optimization, a scoring rule ϕ attaches priority $s_g = \phi(B_g)$ to each group. We retain the normalized form

$$\pi_g = \frac{\exp(s_g/T)}{\sum_{j=1}^{G} \exp(s_j/T)}. \tag{2}$$

The next stage returns only a support S; write $\tau = \sum_{g \notin S} \pi_g$ for the omitted score mass. For top-p, S is the shortest score-ranked prefix carrying at least mass p, with a fixed tie-breaking rule. Nothing below depends on reward variance or top-p specifically: the required objects are a pre-selection measure π, its retained support S, and the reduction used downstream. Here π records relative priority within the realized batch; it is neither the rollout law nor a desired policy-gradient distribution.

Algorithm-native selected aggregation. Different policy reductions aggregate selected groups differently. Conditioned on a fixed rollout batch and a chosen selected reduction, let h_S denote its normalized group aggregation:

$$h_S(g) \geq 0, \quad \sum_{g \in S} h_S(g) = 1, \quad h_S(g) = 0 \ (g \notin S). \tag{3}$$

This covers uniform prompt aggregation, token-weighted PPO reductions, and algorithm-specific masks. Let $\mu_g \in \mathbb{R}^d$ be the group update after native advantage, clipping, and token reductions. The selected-only reference update under this reduction is

$$G_H = \sum_{g \in S} h_S(g) \mu_g. \tag{4}$$

Separating h_S from μ_g matters: completion acts through aggregation of policy atoms, leaving each optimizer's internal surrogate intact.

The no-filter boundary. PPO, GRPO, DAPO, and DrGRPO specify different policy atoms and selected reductions; the construction does not equate their objectives. Without post-rollout selection, $S = [G]$ and $\tau = 0$, so completion returns the chosen full-batch reduction $h_{[G]}$. In particular, choosing the base optimizer's reduction recovers its original estimator. The Baseline entries in our tables report this no-filter training condition. We now fix a concrete graded selector to see what changes when $\tau > 0$.

2.2 Running instance: dispersion-ranked rollout groups

We use the reward-dispersion filter introduced by RAGEN-2 (Wang et al., 2026). The experimental implementation ranks groups by within-prompt reward spread and applies softmax top-p selection. With M returns in group g, the implemented score is the sample standard deviation

$$s_g = \sqrt{\frac{1}{M-1}\sum_{m=1}^{M}(R_{g,m} - \bar{R}_g)^2}, \tag{5}$$

followed by (2) with $T = 1$ and the common top-p selector. The RAGEN-2 comparator and our variants use this same scorer, threshold, rewards, and rollout process. EC acts after selection, carrying the graded priorities into the actor-loss reduction. The same estimator question arises with uncertainty or difficulty scores.

2.3 The compression gap

Proposition 2.1 (Selection is insufficient)**.** *Let $0 < p < 1$. For any nontrivial selected set $\varnothing \subsetneq S \subsetneq [G]$ attainable with a strict top-p boundary and strictly ordered positive scores, the map from graded score distributions to selected sets is many-to-one. Distributions inducing the same S can have different tail mass $\tau = \sum_{g \notin S} \pi_g$; if $|S^c| \geq 2$, they can also have different conditional tail distributions while preserving τ.*

A constructive proof is in Appendix A.1. Thus a mask cannot recover τ or tail ordering. Hard filtering is a lossy projection followed by an extra modeling decision: assign zero estimator mass outside S and renormalize the selected region to one.

This loss has an operational consequence. Fix the selected aggregation h_S and all group update atoms $\{\mu_g\}$. For two score states π, π' in the same selection fiber, let $G_\star(\pi)$ and $G_\star(\pi')$ be their completed reference updates. A rule A that sees the mask and atoms but not the scores must return one update for both states. It obeys

$$\max\{\|A - G_\star(\pi)\|, \|A - G_\star(\pi')\|\} \geq \tfrac{1}{2}\|G_\star(\pi) - G_\star(\pi')\|. \tag{6}$$

When the two completed updates differ, no mask-only estimator can reconstruct both. A proof is in Appendix A.2. For example, with $p = 0.8$, both $\pi = (0.60, 0.38, 0.02)$ and $\pi' = (0.43, 0.39, 0.18)$ select the first two of three groups. If their head atoms are zero and their third atom is one, the completed scalar updates are 0.02 and 0.18, although the hard-selected update is zero in both cases. Even a mask-only rule allowed to inspect all three atoms cannot match both completed updates: (6) gives a worst-case error of at least 0.08. The example isolates the information loss without invoking any stochastic-gradient assumption.

The ambiguity can also be measured in the logged batches. If u_S is uniform on the selected groups, then $\mathrm{TV}(\pi, u_S)$ measures the combined support and aggregation change from graded priority to selected-only estimation.

The scorer and optimizer answer different questions: π ranks groups, whereas h_S specifies how the selected policy atoms are reduced. The same mask can conceal distinct scorer states, as (6) shows; using π as the update distribution would instead replace the native selected reduction. Completion must therefore retain h_S on the head and the scorer's conditional structure on the tail. This reasoning applies to any graded top-k, threshold, or acceptance selector that reduces priorities to a mask. Our reward-dispersion case makes the pre-mask measure, selected support, and native update observable together, allowing the information loss to be measured before defining its repair.

3 DIAGNOSING THE LOST INFORMATION

The proposition establishes an information loss. To see whether its missing coordinates occur in practice, we inspect logged batches from the reward-dispersion filter. Figures 2 and 3 show the graded score profiles, omitted mass, projection gap, and tail reward activity, both in aggregate and across training.

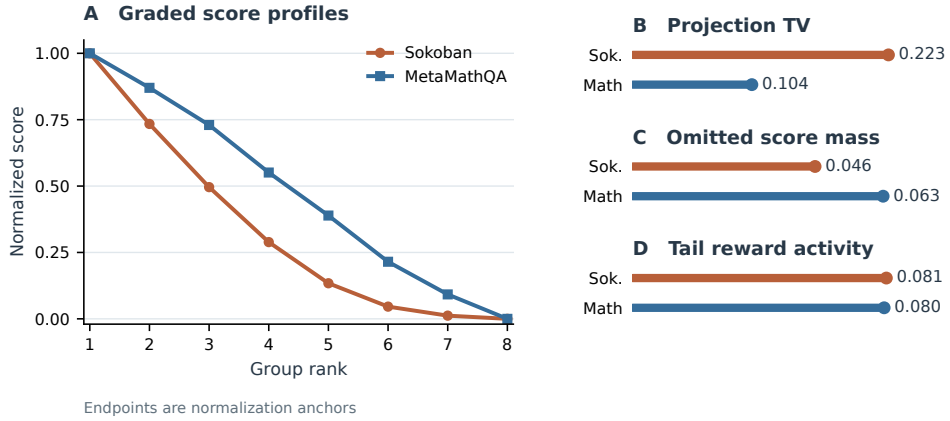

Figure 2: Filter diagnostics from Qwen2.5-1.5B, seed 12002. **A:** ranked normalized reward-standard-deviation profiles; endpoints are normalization anchors. **B–D:** projection TV, omitted score mass, and mass-scaled mean absolute tail reward activity. Panel D averages steps with nonempty tails. Bar lengths are comparable within each panel; panel scales differ.

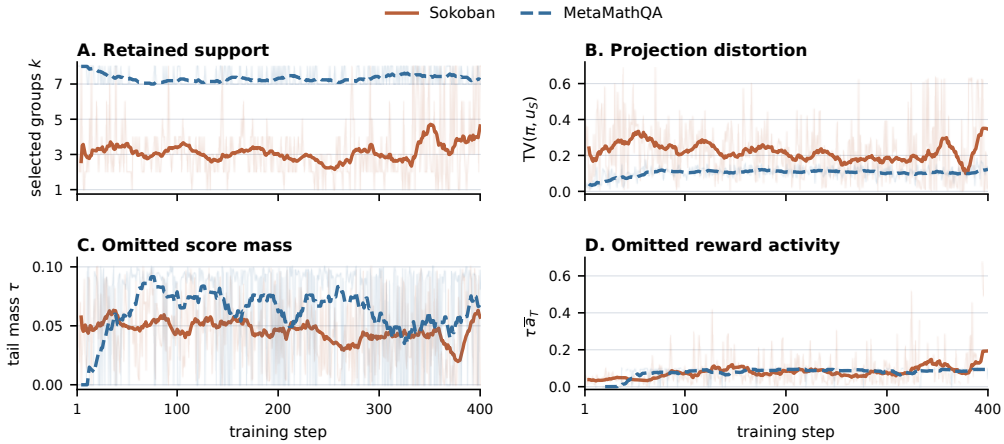

Figure 3: The same diagnostics across training (Qwen2.5-1.5B, seed 12002). Faint lines show logged steps; heavier solid and dashed lines show trailing 21-step means for Sokoban and MetaMathQA. **A:** selected count k; **B:** projection TV; **C:** omitted score mass; **D:** mass-scaled mean absolute tail reward activity, on steps with nonempty tails.

Let $k = |S|$ and let $u_S(g) = 1/k$ on S. Omitted mass τ measures lost support; $\mathrm{TV}(\pi, u_S)$ also captures redistribution within the selected head. For rejected groups, define $a_g = M^{-1} \sum_m |R_{g,m}|$ and use its tail mean and τ-weighted value to measure reward activity. To inspect the scores before selection, sort them as $s_{(1)} \geq \cdots \geq s_{(G)}$ and average their normalized profile over logged training batches:

$$\rho_j = \mathbb{E}\left[\frac{s_{(j)} - \min_g s_g}{\max_g s_g - \min_g s_g}\right]. \tag{7}$$

Zero-range batches are excluded.

The endpoints $\rho_1 = 1$ and $\rho_G = 0$ are normalization anchors; the interior profiles decline gradually rather than form a binary plateau. Sokoban retains about 3.19 of eight groups on average, while MetaMathQA retains about 7.32. Their projection TVs are 0.223 and 0.104, and both leave roughly 0.08 of mass-scaled tail reward activity; omitted score masses are 0.046 and 0.063. Figure 3 shows that these features recur across training. The mask therefore removes graded priority from batches where rejected groups still carry score mass and reward activity. Appendix C gives the diagnostic definitions in full.

4 Estimator Completion

The diagnostics show what the mask leaves behind; the formulation shows why that information cannot be recovered from S. We now ask what estimator is determined if we retain it. Assume $0 < \tau < 1$ and define the scorer's conditional tail

$$r_T(g) = \pi_g/\tau \quad (g \notin S), \qquad r_T(g) = 0 \quad (g \in S). \tag{8}$$

4.1 Completion by partition-wise conservation

Every probability measure decomposes over a nontrivial partition as

$$q = q(S)q(\cdot \mid S) + q(S^c)q(\cdot \mid S^c). \tag{9}$$

Thus a full group distribution is determined by a conditional law on each side and the mass between them. The optimizer supplies h_S on the selected side; the scorer supplies r_T and τ on the omitted side. Preserving the two conditional laws while leaving their masses free gives

$$\mathcal{C}(h_S, r_T) = \{ah_S + br_T : a, b \geq 0\}. \tag{10}$$

Its interior ($a, b > 0$) contains precisely the finite measures with conditionals h_S and r_T; the boundary points $a = 0$ or $b = 0$ remove one region. The coordinates separate head and tail scale, while normalization is an additional decision. Requiring a probability measure that also preserves the scorer's partition masses selects one interior point:

$$q(\cdot \mid S) = h_S \qquad \text{(selected-aggregation consistency)}, \tag{11}$$

$$q(\cdot \mid S^c) = r_T \qquad \text{(tail-order consistency)}, \tag{12}$$

$$q(S^c) = \tau \qquad \text{(mass consistency)}. \tag{13}$$

Theorem 4.1 (Unique mass-consistent completion)**.** *The unique distribution satisfying* (11)–(13) *is*

$$q_\star = (1 - \tau)h_S + \tau r_T. \tag{14}$$

The proof is in Appendix A.3. Once the two conditional laws and their relative mass are fixed, no degree of freedom remains. The result also solves a minimum-change problem. Write $\pi_H = \pi(\cdot \mid S)$ and let $\mathcal{Q} = \{q : q(S^c) = \tau,\ q(\cdot \mid S) = h_S\}$. For every $q \in \mathcal{Q}$,

$$D_{\mathrm{KL}}(q\|\pi) = D_{\mathrm{KL}}(q\|q_\star) + D_{\mathrm{KL}}(q_\star\|\pi), \qquad D_{\mathrm{KL}}(q_\star\|\pi) = (1 - \tau)D_{\mathrm{KL}}(h_S\|\pi_H). \tag{15}$$

Hence $q_\star$ is the unique KL projection of the scorer distribution onto the constraints imposed by the native selected aggregation and scorer partition mass. The unavoidable divergence is entirely in the head, where h_S may differ from the scorer's own conditional weights; the tail incurs none. The derivation is in Appendix A.4.

Equivalently, on positive coordinates the completed estimator obeys

$$\frac{q_\star(g)}{q_\star(j)} = \frac{h_S(g)}{h_S(j)}\ (g, j \in S), \quad \frac{q_\star(g)}{q_\star(j)} = \frac{\pi_g}{\pi_j}\ (g, j \notin S), \quad \frac{q_\star(S^c)}{q_\star(S)} = \frac{\tau}{1 - \tau}. \tag{16}$$

The first relation conserves the selected reduction's internal geometry, the second the scorer's unresolved tail geometry, and the third their cross-partition mass. Removing any one leaves infinitely many admissible distributions.

For later use, collect the scorer-side quantities as

$$m_H = 1 - \tau, \qquad \nu_T(g) = \pi_g \mathbb{I}[g \notin S]. \tag{17}$$

Then $q_\star = m_H h_S + \nu_T$. Both quantities come from the scores and mask that produced S; neither is a free scalar, schedule, or searched setting.

This representation identifies the *minimal scorer state* needed after selection. Fix S and h_S, and consider two pre-mask measures π, π' with tail masses τ, τ'. Their completed measures satisfy

$$\mathrm{TV}(q_\star^{\pi}, q_\star^{\pi'}) = \tfrac{1}{2}\left(|\tau - \tau'| + \sum_{g \notin S} |\pi_g - \pi'_g|\right). \tag{18}$$

The two completions coincide exactly when their tail score vectors coincide: the selected scores matter only through their total mass, while their internal ordering is intentionally replaced by h_S. Thus ν_T, together with S and the native head rule, is sufficient for completion; retaining every selected score is unnecessary. A derivation is in Appendix A.5.

4.2 The exact correction to hard filtering

Write $q_H = h_S$ for the hard estimator. Because head and tail supports are disjoint, the distance and update correction at the uniquely determined endpoint are exact.

Corollary 4.2 (Distance and update identities)**.**

$$\mathrm{TV}(q_H, q_\star) = \tau. \tag{19}$$

Let $G_T = \sum_{g \notin S} r_T(g)\mu_g$. *Then*

$$G_\star = (1-\tau)G_H + \tau G_T, \tag{20}$$

$$G_\star - G_H = \tau(G_T - G_H) \tag{21}$$

$$= \underbrace{-\tau G_H}_{\textit{head-mass recalibration}} + \underbrace{\tau G_T}_{\textit{tail-support restoration}}. \tag{22}$$

Thus completion has exactly two effects, irrespective of the downstream optimizer: head-mass recalibration and tail-support restoration. Their shared mass τ is observed rather than tuned; each native reduction determines where these effects can enter without changing its underlying policy surrogate.

The correction scales with omitted mass: if $\|\mu_g\| \le B$, then $\|G_\star - G_H\| \le 2\tau B$. Our $p = 0.9$ selector gives $\tau \le 0.1$ and hence a bound of $0.2B$. Its direction comes from the contrast $G_T - G_H$, so even modest tail mass can matter when selected policy terms cancel. The derivation and directional identity are in Appendix A.10.

The projection separates the sources of score-to-estimator distortion. The plotted $\mathrm{TV}(\pi, u_S)$ uses a uniform selected-head diagnostic; the following identities use the optimizer's actual h_S. Since $\pi = (1-\tau)\pi_H + \tau r_T$ and $q_\star = (1-\tau)h_S + \tau r_T$,

$$\mathrm{TV}(\pi, q_\star) = (1-\tau)\,\mathrm{TV}(\pi_H, h_S), \qquad \mathrm{TV}(q_\star, q_H) = \tau. \tag{23}$$

The first term measures how the native optimizer reshapes the selected head; the second is exactly the tail mass erased by hard filtering. Reporting omitted mass alongside total projection TV separates these two sources of distortion.

The identities isolate responsibilities: the scorer supplies τ and r_T, while the optimizer supplies h_S and the policy atoms μ_g. Equation (20) is exact for a probability-weighted group mean. We next carry this head/tail interface into the native reductions of token-, sample-, and group-normalized losses.

5 A Prototype for Native Policy Estimators

The group-level construction combines the optimizer's selected aggregation with the scorer's tail measure. Native losses add token masks, clipping, and their own normalization. EC carries the same head/tail information through each loss's existing reduction.

Estimator interface. For optimizer a, let H_a denote its native selected policy term and T_a the same optimizer's policy atoms aggregated over the tail according to r_T. At the group-linear reference level, the prototype is

$$J_{a,\star} = (1-\tau)H_a + \tau T_a, \qquad J_{a,\star} - H_a = \tau(T_a - H_a). \tag{24}$$

For a group-linear reducer this is an exact finite-batch identity. The score sets the size of the correction through τ; the policy terms decide its direction through $T_a - H_a$. This separates score-derived scale from policy-derived direction.

Native lifting. Native token, sample, and group reductions transform the reference masses into effective batch coefficients. Their common implementation interface is the two-coordinate form

$$J_a^{\text{native}} = \alpha_a H_a + \beta_a T_a, \qquad \alpha_a, \beta_a \geq 0, \tag{25}$$

where each realization specifies how its native reducer uses the score-conditioned coordinates. The mass-conserving reference is $(\alpha_a, \beta_a) = (1-\tau, \tau)$ when the reducer is group-linear. It is the probability-calibrated point in (10); that cone also admits head-only calibration and tail-only restoration when a native reduction holds the other component fixed. The scorer, top-p set, and native policy-surrogate form are retained, while detached weighting or scaling determines the realized coefficients. This shared coordinate system makes the variants comparable without assigning them identical numerical coefficients. Exact placements and boundary cases are in Appendix B.

We call these realizations EC-PPO, EC-GRPO, EC-DAPO, and EC-DrGRPO. EC names the shared score-to-estimator interface; the suffix identifies the native policy reduction. PPO makes the effect of native normalization particularly clear.

For PPO, this difference can be written exactly. Let n_g be the number of valid response tokens in group g. With raw group weight 1 on S and π_g on S^c, the effective tail fraction is

$$\lambda_B = \frac{\sum_{g \notin S} \pi_g n_g}{\sum_{g \in S} n_g + \sum_{g \notin S} \pi_g n_g}. \tag{26}$$

The PPO policy term is $(1-\lambda_B)H_{\text{PPO}} + \lambda_B T_{\text{PPO}}$, with H_{PPO} and T_{PPO} the selected and score-weighted tail token means defined in Appendix B. Token counts and the native denominator map scorer tail weights to the effective mass λ_B, generally distinct from the group-level τ. The other optimizers use the same score-derived coordinates to calibrate the selected term, restore tail policy terms, or both.

6 Experiments

The logged diagnostics establish that selection discards graded information, and the prototype shows how native losses can use it. The remaining question is whether doing so improves learning. We compare Baseline, RAGEN-2 filtering, and EC variants across optimizers, then vary PPO model scale and initialization. Each filtered comparison keeps the scorer and selector fixed and changes the specified actor-loss reduction.

6.1 Evaluation matrix and reporting

We draw four environments from the controlled agent-RL suite of RAGEN (Wang et al., 2025b). They create different aggregation regimes rather than merely adding task count: Sokoban has shaped feedback but irreversible decisions (Schrader, 2018); FrozenLake combines sparse success with transition noise (Brockman et al., 2016); MetaMathQA permits rewarded revisions (Yu et al., 2023); and Countdown demands an exact one-shot construction (Katz et al., 2025). Their relevant contrasts are summarized in Table 3.

An update normally contains eight prompt groups with 16 rollouts each. Optimization is implemented in veRL/HybridFlow (Sheng et al., 2024). The estimator axis comprises PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and DrGRPO (Liu et al., 2025); the model axis covers Qwen2.5 scales and initializations (Qwen Team, 2024), plus Llama-3.2-3B-Instruct (Meta Llama, 2024).

For each of three seeds, we take the peak held-out validation success over the common training horizon, following the RAGEN-2 reporting protocol. Main tables report the mean of those three peaks; Appendix D gives the standard deviations. Each comparison uses Baseline (without the RAGEN-2 filter), RAGEN-2, and the corresponding estimator-completion variant.

6.2 CORE COMPARISON

Table 1: Qwen2.5-3B success (%). Avg. is the unweighted mean across four tasks. EC denotes the native variant of each optimizer; standard deviations are in Appendix D.

Optimizer	Setting	Sokoban	FrozenLake	MetaMathQA	Countdown	Avg.
PPO	Baseline	13.4	64.0	91.1	96.3	66.2
	RAGEN-2	25.4	74.1	91.5	96.8	72.0
	EC	30.4	70.2	92.6	96.7	72.5
GRPO	Baseline	10.3	67.4	90.1	96.0	66.0
	RAGEN-2	19.5	64.4	91.0	96.3	67.8
	EC	20.3	65.1	91.9	98.0	68.8
DAPO	Baseline	13.9	62.5	89.7	96.0	65.5
	RAGEN-2	19.7	63.5	90.3	95.9	67.4
	EC	18.9	66.5	91.8	96.9	68.5
DrGRPO	Baseline	12.5	20.8	90.6	94.5	54.6
	RAGEN-2	10.3	20.5	91.3	94.5	54.2
	EC	13.5	25.8	91.1	95.6	56.5

For Qwen2.5-3B, each optimizer's four-task average exceeds its RAGEN-2 counterpart: by 0.53 points with PPO, 1.03 with GRPO, 1.18 with DAPO, and 2.35 with DrGRPO, before rounding the averages in the table. Twelve of the 16 task means rise. GRPO improves on all four tasks, including a 1.7-point gain on Countdown; PPO adds 5.0 points on Sokoban, and DrGRPO adds 5.3 on FrozenLake. The average gains across distinct policy reductions show the reach of the common estimator interface.

6.3 TRANSFER ACROSS MODELS, OPTIMIZERS, AND TASKS

The core comparison varies the optimizer at fixed Qwen2.5-3B. We next hold PPO fixed and vary model scale and initialization to test whether the pattern is specific to that model.

Table 2: PPO success (%) across additional models. Each cell lists Baseline / RAGEN-2 / EC-PPO means; bold marks EC-PPO. Standard deviations are in the appendix. Inst. denotes Instruct.

Model	Sokoban	FrozenLake	MetaMathQA	Countdown
Qwen2.5-0.5B	4.3 / 24.5 / **23.1**	18.1 / 19.0 / **23.3**	10.3 / 8.9 / **16.1**	20.2 / 20.6 / **22.7**
Qwen2.5-1.5B	16.1 / 20.0 / **19.8**	36.5 / 38.6 / **42.9**	79.5 / 86.0 / **85.1**	57.0 / 57.4 / **62.0**
Qwen2.5-7B	43.2 / 44.8 / **47.7**	83.0 / 82.4 / **80.8**	82.8 / 94.6 / **95.7**	97.1 / 96.5 / **98.1**
Qwen2.5-3B-Inst.	21.3 / 37.5 / **41.2**	82.4 / 83.7 / **84.0**	90.7 / 90.1 / **90.7**	95.9 / 95.0 / **95.6**
Llama3.2-3B-Inst.	22.2 / 40.3 / **40.3**	81.2 / 83.4 / **83.7**	84.1 / 87.6 / **88.0**	98.4 / 97.5 / **98.6**

The same PPO variant raises the four-task average for each additional model: by 3.05, 1.95, 1.00, 1.30, and 0.45 points in table order. Including Qwen2.5-3B, all six model-level averages rise, with gains in 15 of the 20 additional task comparisons. Qwen2.5-0.5B rises from 8.9 to 16.1 on MetaMathQA, while Qwen2.5-3B-Instruct rises from 37.5 to 41.2 on Sokoban. This breadth across scale and instruction tuning complements the optimizer comparison. The per-seed dispersions appear in Appendix D.

Reading the task-level results. The completion identity gives a useful lens on task-level variation. Across batches B, let $D(B) = G_T(B) - G_H(B)$ when the tail is nonempty, and $D(B) = 0$ otherwise. At the group-linear reference level,

$$\mathbb{E}_B[G_\star - G_H] = \mathbb{E}_B[\tau D] = \mathbb{E}_B[\tau]\,\mathbb{E}_B[D] + \mathrm{Cov}_B(\tau, D). \tag{27}$$

The scorer controls when and how much mass is omitted, while the policy atoms determine the head–tail contrast in the same batch. Both the mean contrast and its coupling with omitted mass enter the mean reference correction. The score therefore supplies an adaptive gate, rather than a fixed tail coefficient. Native reductions then turn that gate into optimizer-specific effective masses, as the PPO token denominator in (26) makes explicit. This batch-dependent view connects the measured score structure to the breadth and variation of the gains in the tables.

An interface for rollout filters. The minimal-state result in (18) also has a practical consequence. After choosing S, a filter can pass the mask and the omitted score vector ν_T to the optimizer; ν_T already determines the omitted mass τ. The optimizer then supplies its own selected aggregation h_S and policy atoms. Thus the same scorer output supports head-mass calibration, tail restoration, or both, while keeping the policy surrogate of each optimizer. This separation is what allows one EC construction to have PPO, GRPO, DAPO, and DrGRPO realizations without redesigning the filter for each optimizer.

7 Related Work

Selection and truncation. The intervention above acts after rollout sampling. Nucleus sampling truncates tokens during generation (Holtzman et al., 2020), and Best-of-N selects inference candidates (Sun et al., 2024); neither determines how the graded score of an already sampled training batch enters its optimizer.

Policy-gradient weighting and normalization. PPO, GRPO, DAPO, and DrGRPO differ in clipping, advantages, and reduction (Schulman et al., 2017; Shao et al., 2024; Yu et al., 2025; Liu et al., 2025). Other work changes variance estimates (Romoff et al., 2018) or within-group weighting, entropy, and diversity (Feng et al., 2025; Wang et al., 2025a; Xu et al., 2025; Yao et al., 2025). Our reference measure describes the outer aggregation after selection, while the reported variants keep their native surrogates.

Scored experience and information loss. RAGEN studies interaction-generated training data (Wang et al., 2025b; Wang & Ammanabrolu, 2025). RAGEN-2 links within-prompt reward dispersion to the choice of high-signal groups and studies reasoning collapse through input-dependence diagnostics (Wang et al., 2026). We take its scorer and selected set as inputs. The score-to-set map is a deterministic coarse-graining (Cover & Thomas, 2006); our analysis begins with the estimator ambiguity this creates and ends with ways to carry the remaining score information into native policy updates. This makes the question meaningful for other scorers as well.

8 Conclusion

The mask produced by a rollout filter does not settle how its scores should enter the update. We preserve the optimizer's aggregation on selected groups and the scorer's relative ordering in the tail. Together with the scorer's partition mass, these determine a unique reference measure and the exact correction $\tau(G_T - G_H)$. Only the tail score vector must be retained after selection; native policy losses can use it through head and tail terms.

The resulting interventions raise the four-task average of peak-validation means over RAGEN-2 for each of four Qwen2.5-3B optimizers and, under PPO, for every tested model scale and initialization. Together, the group-level identities, native policy implementations, and cross-model results establish estimator completion as a reusable design principle for graded rollout selection. A rollout filter should specify the update it induces as carefully as the groups it selects.

AI USE STATEMENT

Generative AI tools were used to improve the wording of the manuscript and to assist in checking the correctness of the proofs. The authors reviewed all AI-assisted suggestions, independently verified the final proofs, and take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

The score-to-set formulation and estimator completion rule are specified in Sections 2 and 4; assumptions and formal guarantees appear in Section 4, with complete proofs in Appendix A. The native policy-loss realizations are described in Section 5 and Appendix B. Experimental protocols and principal implementation settings are reported in Section 6 and Appendix E, with full results in Appendix D.

REFERENCES

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Michael Katz, Harsha Kokel, and Sarath Sreedharan. Benchmarking llms on the game of countdown, 2025. URL https://arxiv.org/abs/2508.02900.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

Meta Llama. Llama 3.2 3b model card, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B. Accessed 2026-01-28.

Qwen Team. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Joshua Romoff, Peter Henderson, Alexandre Piché, Vincent Francois-Lavet, and Joelle Pineau. Reward estimation for variance reduction in deep reinforcement learning, 2018. URL https://arxiv.org/abs/1805.03359.

Max-Philipp B. Schrader. gym-sokoban, 2018. URL https://github.com/mpSchrader/gym-sokoban. Accessed 2026-01-29.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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, 2024. URL https://arxiv.org/abs/2402.03300.

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, 2024. URL https://arxiv.org/abs/2409.19256.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. Fast best-of-n decoding via speculative rejection, 2024. URL https://arxiv.org/abs/2410.20290.

Jiawei Wang, Jiacai Liu, Yuqian Fu, Yingru Li, Xintao Wang, Yuan Lin, Yu Yue, Lin Zhang, Yang Wang, and Ke Wang. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon llm agents, 2025a. URL https://arxiv.org/abs/2509.09265.

Ruiyi Wang and Prithviraj Ammanabrolu. A practitioner's guide to multi-turn agentic reinforcement learning, 2025. URL https://arxiv.org/abs/2510.01132.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning, 2025b. URL https://arxiv.org/abs/2504.20073.

Zihan Wang, Chi Gui, Xing Jin, Qineng Wang, Licheng Liu, Kangrui Wang, Shiqi Chen, Linjie Li, Zhengyuan Yang, Pingyue Zhang, et al. Ragen-2: Reasoning collapse in agentic rl. *arXiv preprint arXiv:2604.06268*, 2026.

Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N. Metaxas. Epo: Entropy-regularized policy optimization for llm agents reinforcement learning, 2025. URL https://arxiv.org/abs/2509.22576.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning, 2025. URL https://arxiv.org/abs/2505.23433.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models, 2023. URL https://arxiv.org/abs/2309.12284.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

A PROOFS

A.1 MANY-TO-ONE SCORE-TO-SET MAP

Proof of Proposition 2.1. Let $S = \{1, \dots, k\}$ after sorting, with $1 \le k < G$. Top-p selects S whenever

$$\sum_{i=1}^{k-1} \pi_i < p \le \sum_{i=1}^{k} \pi_i \tag{28}$$

and the ordering does not change. By the proposition's attainability assumption, choose a distribution satisfying both inequalities strictly and with strictly ordered positive coordinates. These conditions define an open neighborhood in the simplex. A sufficiently small perturbation that transfers mass between a head and tail coordinate preserves the ordering and (28), hence preserves S, while changing τ. If $G - k \ge 2$, a sufficiently small mass transfer between two tail coordinates instead preserves S and τ while changing r_T. For the one-element-tail case, varying τ alone establishes non-injectivity. Thus S determines neither tail mass in general nor tail structure when the tail has at least two elements. □

A.2 MASK-ONLY UPDATE BOUND

For fixed S, h_S, and $\{\mu_g\}$, a rule that does not observe π returns the same A for any two score states in the selection fiber. The triangle inequality gives

$$\|G_\star(\pi) - G_\star(\pi')\| \le \|G_\star(\pi) - A\| + \|A - G_\star(\pi')\| \le 2\max\{\|A - G_\star(\pi)\|, \|A - G_\star(\pi')\|\},$$

which proves (6).

A.3 UNIQUE MASS-CONSISTENT COMPLETION

Mass consistency gives $q(S) = 1 - \tau$. For $g \in S$, conditional consistency requires $q(g) = (1-\tau)h_S(g)$; for $g \notin S$, it requires $q(g) = \tau r_T(g)$. These values cover the partition and sum to one, proving uniqueness and existence in Theorem 4.1.

A.4 INFORMATION PROJECTION IDENTITY

For $q \in \mathcal{Q}$, write $t = q(\cdot \mid S^c)$. The scorer and feasible measures have the common partition masses

$$\pi = (1-\tau)\pi_H + \tau r_T, \qquad q = (1-\tau)h_S + \tau t, \qquad q_\star = (1-\tau)h_S + \tau r_T.$$

Their head and tail supports are disjoint. Applying the KL chain rule on this partition gives

$$\begin{aligned} D_{\mathrm{KL}}(q\|\pi) &= (1-\tau)D_{\mathrm{KL}}(h_S\|\pi_H) + \tau D_{\mathrm{KL}}(t\|r_T), \\ D_{\mathrm{KL}}(q\|q_\star) &= \tau D_{\mathrm{KL}}(t\|r_T). \end{aligned}$$

The first term of the first line is $D_{\mathrm{KL}}(q_\star\|\pi)$, proving (15). Strict positivity of the softmax score distribution makes r_T positive on the tail, so equality in the minimization holds only for $t = r_T$.

A.5 SENSITIVITY TO THE RETAINED SCORE STATE

Fix S and h_S. For two scorer measures, the head difference is $(\tau' - \tau)h_S$ and the tail difference is $\nu_T^\pi - \nu_T^{\pi'}$. The supports are disjoint, so their ℓ_1 norms add:

$$2\,\mathrm{TV}(q_\star^\pi, q_\star^{\pi'}) = |\tau - \tau'| + \sum_{g \notin S} |\pi_g - \pi'_g|.$$

This proves (18). Equality of the two completed measures requires equality of their restrictions to S^c, exactly $\nu_T^\pi = \nu_T^{\pi'}$; that equality also fixes τ and hence the head mass.

A.6 TOTAL-VARIATION AND UPDATE IDENTITIES

Recall that $q_H = h_S$ and

$$q_\star = (1-\tau)h_S + \tau r_T.$$

On S, $q_\star - q_H = -\tau h_S$; on S^c, $q_\star - q_H = \tau r_T$. Since both conditional distributions sum to one,

$$\|q_\star - q_H\|_1 = \tau \sum_{g\in S} h_S(g) + \tau \sum_{g\notin S} r_T(g) = 2\tau,$$

which yields $\mathrm{TV}(q_H, q_\star) = \tau$.

For any fixed group contributions $\{\mu_g\}_{g=1}^G$,

$$\begin{aligned} G_\star &= \sum_g q_\star(g)\mu_g \\ &= (1-\tau)G_H + \tau G_T. \end{aligned}$$

Subtracting G_H proves (21). These are conditional finite-batch identities: they require no assumption about unbiasedness, smoothness, or alignment with an unknown population gradient.

A.7 BOUNDARY CASES

If $\tau = 0$, the scorer already places all mass on S and completion reduces to $q_\star = h_S = q_H$; r_T need not be defined. If $\tau = 1$, the selected set has zero scorer mass and requirement (11) is vacuous under ordinary conditional-probability conventions. Top-p selection with strictly positive softmax scores and $p > 0$ excludes this case. The main text therefore treats $0 < \tau < 1$.

A.8 SUPPORT-CHANGING GEOMETRY

Completion changes support, which explains the use of total variation in the main text. Whenever $\tau > 0$, $q_\star$ assigns positive mass where q_H assigns zero, and hence

$$D_{\mathrm{KL}}(q_\star \| q_H) = \infty.$$

The reverse direction remains finite because the conditional law on S is unchanged:

$$D_{\mathrm{KL}}(q_H \| q_\star) = -\log(1-\tau).$$

A symmetric KL argument would therefore require smoothing, whereas total variation remains finite and equals the restored mass exactly.

A.9 AVERAGING OVER ON-POLICY BATCHES

The main identities condition on the realized batch B, where scores, selection, advantages, and policy ratios are jointly determined. Set $D(B) = 0$ when the tail is empty. Averaging the correction identity gives

$$\mathbb{E}_B[G_\star(B)] - \mathbb{E}_B[G_H(B)] = \mathbb{E}_B[\tau(B)D(B)], \quad D(B) = G_T(B) - G_H(B).$$

Thus completion is a batch-adaptive, scorer-determined gate rather than a tail loss with a global coefficient. Both terms use the same current-policy rollouts; completion changes their post-reward aggregation and does not introduce replay or off-policy samples.

A.10 PERTURBATION CONTROL AND DIRECTIONAL IDENTITY

If $\|\mu_g\| \le B$ for every group, convexity gives $\|G_H\|, \|G_T\| \le B$. Equation (21) and the triangle inequality therefore imply

$$\|G_\star - G_H\| \le 2\tau B.$$

For any diagnostic direction d, linearity also gives the exact identity

$$\langle d, G_\star - G_H\rangle = \tau\big(\langle d, G_T\rangle - \langle d, G_H\rangle\big).$$

Thus the measured restored mass controls magnitude, while the optimizer- and environment-specific head–tail contrast determines direction.

B ALGORITHM AND IMPLEMENTATION DETAILS

B.1 ONE CONSTRUCTION ACROSS NATIVE REDUCTIONS

Every realization follows the same sequence:

1. Compute group reward-dispersion scores s_g and normalized scores π_g.
2. Apply the unchanged top-p rule to obtain S; retain both the mask and the graded scores.
3. Form the same completion coordinates $m_H = 1 - \sum_{g \notin S} \pi_g$ and $\nu_T = \pi \mathbb{I}[S^c]$.
4. Keep the base policy atoms and introduce the two effects at the outer actor-loss reduction. A realization that fixes its selected term carries tail restoration; one that fixes selected support carries head-mass recalibration; a full-support realization may carry both.
5. Express the result in the optimizer's existing token, sample, or group normalization. No coefficient is chosen to trade the head against the tail.

The first three steps are identical across all experiments. The policy-loss placement is specified once for each optimizer realization and is held fixed across tasks and seeds.

B.2 NATIVE REALIZATIONS

The reported variants share the two-coordinate policy form (25). The reference completion in (14) fixes the common head/tail coordinates, while each native reducer determines their effective coefficients without a swept completion-strength hyperparameter. Here H_a is the selected-only policy term under that realization's reduction. For tail-admitting variants, T_a uses the scorer's conditional tail weights r_T and the optimizer's own policy atoms. These identities describe the policy component. EC-PPO and EC-DrGRPO also apply their actor weights to entropy regularization; EC-GRPO and EC-DAPO modify the policy component alone.

EC-PPO. The same EC-PPO reduction is used for every PPO model row. The selected rollouts keep unit raw actor weight, and critic treatment is unchanged. Unselected groups obtain raw actor weight π_g. Let $J_g^{\mathrm{clip}}(\theta)$ be the mean clipped PPO policy atom over the n_g valid response tokens of group g. For any detached $c_g \geq 0$, clipping is positively homogeneous in the advantage:

$$J_g^{\mathrm{clip}}(\theta; c_g \widehat{A}) = c_g J_g^{\mathrm{clip}}(\theta; \widehat{A}). \tag{29}$$

This follows because both branches of the clipped minimum scale by c_g; it justifies placing a fixed group weight on the policy atom or its advantage before the same reducer.

Let $D_H = \sum_{g \in S} n_g$ and $D_T = \sum_{g \notin S} \pi_g n_g$. Define the selected and conditional tail token means

$$H_{\mathrm{PPO}} = \frac{\sum_{g \in S} n_g J_g^{\mathrm{clip}}}{D_H}, \qquad T_{\mathrm{PPO}} = \frac{\sum_{g \notin S} \pi_g n_g J_g^{\mathrm{clip}}}{D_T}.$$

For nonempty regions, the implemented sum-weight policy reduction is exactly

$$J_{\mathrm{PPO}}^{\mathrm{EC}} = (1 - \lambda_B) H_{\mathrm{PPO}} + \lambda_B T_{\mathrm{PPO}}, \qquad \lambda_B = \frac{D_T}{D_H + D_T}. \tag{30}$$

Because λ_B is detached, $\nabla J_{\mathrm{PPO}}^{\mathrm{EC}} - \nabla H_{\mathrm{PPO}} = \lambda_B (\nabla T_{\mathrm{PPO}} - \nabla H_{\mathrm{PPO}})$ holds for the policy component at a fixed batch. This is the exact code-level bridge to (24): score mass determines a positive gate, while variable token lengths and native normalization determine its realized value. With equal n_g and uniform selected aggregation, $\lambda_B = \tau/(|S| + \tau)$ and the induced group measure is $(|S| h_S + \nu_T)/(|S| + \tau)$, a point in (10). It prevents low-weight tail tokens from diluting the head through an unweighted denominator; if $D_T = 0$, the hard selected term is recovered.

EC-GRPO. For each selected row, the policy advantage is multiplied by

$$m_S = \sum_{g \in S} \pi_g = 1 - \tau.$$

Unselected rows remain outside the policy update and non-policy components are not modified. Positive homogeneity of the clipped policy atom gives $(\alpha, \beta) = (1 - \tau, 0)$ exactly. This isolates head-mass calibration: the batch-dependent score mass sets the policy scale relative to the unchanged entropy term and enters gradient clipping and AdamW moment accumulation.

EC-DAPO. Let j_{gm} be the token-mean clipped DAPO policy atom for valid sample m in group g, and let M_g be that group's valid sample count. For $N_H = \sum_{g \in S} M_g > 0$, the selected sample mean and tail auxiliary are

$$J^{\mathrm{EC}}_{\mathrm{DAPO}} = \underbrace{\frac{1}{N_H} \sum_{g \in S} \sum_{m=1}^{M_g} j_{gm}}_{H_{\mathrm{DAPO}}} + \underbrace{\sum_{g \notin S:\, M_g > 0} \pi_g \left(\frac{1}{M_g} \sum_{m=1}^{M_g} j_{gm} \right)}_{\tau T_{\mathrm{DAPO}}}. \tag{31}$$

Thus $(\alpha, \beta) = (1, \tau)$ under this sample-level reduction, taking $T_{\mathrm{DAPO}} = 0$ when no tail sample is valid. Both terms use the same DAPO policy atoms.

EC-DrGRPO. Selected rows retain unit weight. Let A be the token-level advantages and let w denote the detached scorer-derived weights assigned to tail tokens. Let $\mathrm{RMS}_S(A)$ and $\mathrm{RMS}_T(wA)$ be selected and raw weighted-tail RMS values. On batches with nonzero head and tail mass and nonzero RMS values, the tail scale is

$$\frac{\tau}{1-\tau} \frac{\mathrm{RMS}_S(A)}{\mathrm{RMS}_T(wA)},$$

The implementation returns the selected-only term when either region is empty or the required RMS vanishes. The probability ratio specifies a relative tail budget and RMS matching prevents raw advantage scale from overwhelming it. Writing the scorer-weighted raw tail policy term as τT_{DrGRPO}, this gives $(\alpha, \beta) = (1, c\tau)$ with c equal to the displayed scale. The RMS factor is the unique positive rescaling that makes weighted-tail advantage RMS equal to $\tau/(1-\tau)$ times selected advantage RMS, when both are nonzero. This matches a second-moment budget, not the mass-conserving $q_\star$ update.

C DIAGNOSTIC DETAILS

Table 3: Evaluation tasks expose different sources of tail information.

Task	interaction	feedback	primary challenge
Sokoban	multi-turn	dense	irreversible planning
FrozenLake	multi-turn	sparse	stochastic navigation
MetaMathQA	retry-based	graded	mathematical reasoning
Countdown	single-turn	sparse	exact composition

For Qwen2.5-1.5B seed 12002, the exact tuples (average selected count, projection TV, tail mass, mean absolute tail reward activity, mass-scaled mean activity) are $(3.1925, 0.2227, 0.0458, 1.6610, 0.0807)$ on Sokoban and $(7.3175, 0.1037, 0.0629, 0.8665, 0.0800)$ on MetaMathQA. The corresponding mean per-batch min–max-normalized ranked score profiles are:

$$\text{Sokoban: } [1.000, 0.734, 0.496, 0.289, 0.134, 0.046, 0.012, 0.000],$$
$$\text{MetaMathQA: } [1.000, 0.870, 0.730, 0.551, 0.389, 0.215, 0.092, 0.000].$$

The endpoints are normalization anchors. The informative observation is that the interior coordinates decay progressively instead of becoming a binary plateau matching the hard mask. Projection total variation is computed against u_S, a deliberately simple diagnostic. The theorem itself uses the optimizer-native h_S and does not assume that every optimizer aggregates selected groups uniformly.

The tail statistic uses

$$a_g = \frac{1}{M} \sum_{m=1}^{M} |R_{g,m}|, \qquad \bar{a}_T = \frac{1}{|S^c|} \sum_{g \notin S} a_g, \qquad \text{mass-scaled activity} = \tau \bar{a}_T.$$

It is an observational reward statistic. It is not advantage energy, gradient energy, or an estimate of gradient alignment.

D FULL REPORTED RESULTS

All entries below report the mean and standard deviation (%) of peak held-out validation success over three seeds per cell, following the RAGEN-2 peak-validation reporting protocol. EC-PPO, EC-GRPO, EC-DAPO, and EC-DrGRPO are the native realizations defined in Appendix B.

Table 4: Qwen2.5-3B success (%): means ± standard deviations for every optimizer and setting.

Setting	Optimizer	Sokoban	FrozenLake	MetaMathQA	Countdown
Baseline	PPO	13.4 ± 2.8	64.0 ± 3.2	91.1 ± 1.0	96.3 ± 0.9
Baseline	GRPO	10.3 ± 2.2	67.4 ± 4.5	90.1 ± 0.8	96.0 ± 1.1
Baseline	DAPO	13.9 ± 4.3	62.5 ± 5.3	89.7 ± 0.8	96.0 ± 0.3
Baseline	DrGRPO	12.5 ± 2.3	20.8 ± 3.5	90.6 ± 0.9	94.5 ± 1.2
RAGEN-2	PPO	25.4 ± 4.6	74.1 ± 4.2	91.5 ± 2.1	96.8 ± 1.1
RAGEN-2	GRPO	19.5 ± 3.3	64.4 ± 5.3	91.0 ± 0.7	96.3 ± 1.1
RAGEN-2	DAPO	19.7 ± 2.9	63.5 ± 5.0	90.3 ± 1.6	95.9 ± 0.9
RAGEN-2	DrGRPO	10.3 ± 1.8	20.5 ± 2.9	91.3 ± 1.0	94.5 ± 1.5
EC	PPO	30.4 ± 3.8	70.2 ± 2.3	92.6 ± 0.7	96.7 ± 1.6
EC	GRPO	20.3 ± 2.7	65.1 ± 5.9	91.9 ± 0.6	98.0 ± 1.3
EC	DAPO	18.9 ± 3.6	66.5 ± 5.1	91.8 ± 0.5	96.9 ± 0.4
EC	DrGRPO	13.5 ± 1.8	25.8 ± 4.8	91.1 ± 0.8	95.6 ± 2.1

The model comparison below uses PPO for all three settings. Its EC-PPO row uses the same realization as the Qwen2.5-3B PPO row in the optimizer table above.

Table 5: PPO success (%): means ± standard deviations across additional model sizes and initializations.

Setting	Model	Sokoban	FrozenLake	MetaMathQA	Countdown
Baseline	Qwen2.5-0.5B	4.3 ± 2.2	18.1 ± 2.9	10.3 ± 1.4	20.2 ± 1.1
Baseline	Qwen2.5-1.5B	16.1 ± 2.9	36.5 ± 1.6	79.5 ± 0.9	57.0 ± 0.5
Baseline	Qwen2.5-7B	43.2 ± 1.9	83.0 ± 1.6	82.8 ± 0.6	97.1 ± 1.2
Baseline	Qwen2.5-3B-Instruct	21.3 ± 2.3	82.4 ± 3.1	90.7 ± 1.2	95.9 ± 2.0
Baseline	Llama3.2-3B-Instruct	22.2 ± 3.8	81.2 ± 1.4	84.1 ± 1.3	98.4 ± 0.7
RAGEN-2	Qwen2.5-0.5B	24.5 ± 1.9	19.0 ± 3.5	8.9 ± 2.2	20.6 ± 0.6
RAGEN-2	Qwen2.5-1.5B	20.0 ± 2.2	38.6 ± 1.6	86.0 ± 1.3	57.4 ± 1.1
RAGEN-2	Qwen2.5-7B	44.8 ± 2.6	82.4 ± 1.8	94.6 ± 1.2	96.5 ± 0.7
RAGEN-2	Qwen2.5-3B-Instruct	37.5 ± 2.5	83.7 ± 1.2	90.1 ± 0.8	95.0 ± 1.2
RAGEN-2	Llama3.2-3B-Instruct	40.3 ± 3.9	83.4 ± 2.2	87.6 ± 0.7	97.5 ± 1.7
EC-PPO	Qwen2.5-0.5B	23.1 ± 3.1	23.3 ± 3.0	16.1 ± 4.2	22.7 ± 0.9
EC-PPO	Qwen2.5-1.5B	19.8 ± 2.9	42.9 ± 5.5	85.1 ± 2.7	62.0 ± 4.4
EC-PPO	Qwen2.5-7B	47.7 ± 4.1	80.8 ± 4.0	95.7 ± 4.8	98.1 ± 2.1
EC-PPO	Qwen2.5-3B-Instruct	41.2 ± 4.2	84.0 ± 1.8	90.7 ± 1.9	95.6 ± 2.9
EC-PPO	Llama3.2-3B-Instruct	40.3 ± 5.0	83.7 ± 2.7	88.0 ± 1.2	98.6 ± 0.7

E TRAINING DETAILS

Each core update contains eight prompt groups and 16 trajectories per group (128 trajectories), with policy minibatches of 32 trajectories. Runs last at most 400 rollout–update iterations. PPO uses

GAE with $(\gamma, \lambda) = (1, 1)$ and a critic; GRPO, DAPO, and DrGRPO use their native group-relative advantages. The protocol uses AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, an actor learning rate of 10^{-6}, a PPO critic learning rate of 10^{-5}, entropy coefficient 0.001, and asymmetric clipping thresholds $(\epsilon_{\text{low}}, \epsilon_{\text{high}}) = (0.2, 0.28)$. Per-GPU actor microbatches are set by model memory requirements without changing the logical minibatch. Invalid structured responses receive a -0.1 format penalty.

The experimental configuration uses top-p mass $p = 0.9$ with softmax top-p selection and the sample standard deviation in (5) as its reward-spread score. The scorer temperature in (2) is $T = 1$. Training rollouts use decoding temperature 1, whereas validation uses stochastic decoding at temperature 0.5. Success is evaluated on the same fixed set of 512 held-out prompts per environment. Interactive tasks permit up to five turns and two actions per turn; MetaMathQA and Countdown are single-turn and use automatically verifiable outcomes. Group identity, selection mask, normalized score, head mass, and tail mass are derived from the current on-policy batch and detached before loss reduction. The RAGEN-2 comparator and EC settings share this filter configuration; the intervention is at policy-loss aggregation.