DeSPO: Decoupling Rollout Budget from Set-Valuation Horizon in Policy Optimization

Anonymous authors
Paper under double-blind review

Abstract

Set-level objectives in large language model (LLM) post-training value responses jointly to capture their complementarity. Yet scoring only the realized group couples candidate discovery with set valuation: valuing larger groups requires more generation. We introduce Decoupled Set Policy Optimization (DeSPO), separating the real rollout budget G from an independently chosen set-valuation horizon K. DeSPO optimizes probability allocations over a fixed pool of G responses using task rewards and set value under K draws with replacement. The resulting target guides policy updates without additional text generation. For a determinantal utility, we derive an exact finite-support expansion showing that K reweights interactions of different orders and is not generally equivalent to rescaling a diversity bonus. Experiments across tasks and model scales demonstrate quality–spread gains over baselines at matched rollout budgets and, in selected comparisons, with half as many real training rollouts. Varying K at fixed G produces task-dependent shifts in quality and semantic spread that persist in fresh samples from the learned policy. These findings make valuation horizon an explicit objective-design choice for rollout-efficient post-training, rather than a by-product of the generation budget.

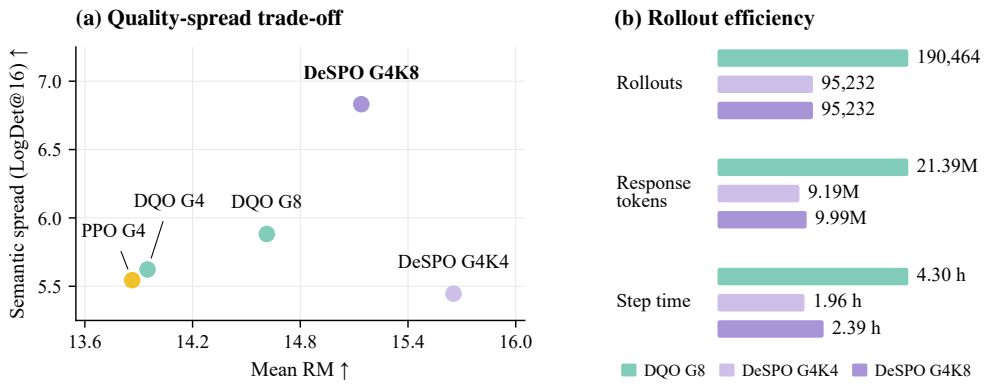

Figure 1: **Quality–spread gains with half the real training rollouts.** On CommonGen with a 3B actor, DeSPO $G4K8$ has higher Mean RM and LogDet@16 than DQO $G8$ in (a). Panel (b) reports rollout counts, response tokens, and accumulated training-log step times. G denotes real rollouts per prompt and K the valuation horizon. Values are three-run means.

1 Introduction

Reinforcement-learning post-training can improve language-model response quality but also narrow the output distribution (Kirk et al., 2024; O'Mahony et al., 2024). This matters when a prompt admits multiple useful answers or when a model is sampled repeatedly: individually strong but redundant responses provide little additional coverage. Recent work introduces diversity-aware objectives that value multiple generations jointly (He et al., 2025; Yao et al., 2026; Li et al., 2025).

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.

Policy optimization learns from finite rollouts. When utility is assigned response by response, the rollout count mainly governs how many samples inform an update. **Set-valued objectives give the same rollout group a second role:** its size also determines how many responses are valued jointly. DQO and SetPO derive their joint signal from the realized group (Chen et al., 2026; Li et al., 2026a), tying candidate discovery to set valuation. Valuing a larger *observed* group requires more generation; rollouts account for roughly 70% of training-step time in the workloads measured by RollPacker (Gao et al., 2026). Yet discovering candidates and valuing repeated selections from them are distinct operations. **Can these roles be decoupled, so a small rollout group supports an independently chosen valuation horizon?**

We introduce **Decoupled Set Policy Optimization (DeSPO)** to separate these roles. DeSPO generates G real responses once, then evaluates probability allocations over this fixed pool by their expected set value under K draws with replacement. With $G = 4$ and $K = 8$, it values hypothetical eight-draw groups formed from the same four responses, including repetitions. Thus, G controls **candidate discovery** and K controls **fixed-support set valuation**; they may coincide or differ, with no additional generation when K increases. For the determinantal utility studied here, an exact finite-support evaluator shows that K reweights interactions involving different numbers of candidates rather than merely scaling a diversity bonus. We combine the logarithm of this expected determinant with task reward and local KL regularization to construct a target allocation, whose probability shifts provide a frozen policy-update signal.

We evaluate DeSPO on open-ended generation and mathematical reasoning across rollout budgets and model scales. At fixed G, varying K produces task-dependent quality–spread shifts that persist under fresh sampling. Comparisons with matched- and larger-rollout baselines show joint gains or higher spread at a quality cost, depending on the task. Controlled interventions probe fitting, allocation movement, geometry, and alternative utilities, while boundary studies highlight constraints from candidate discovery and utility alignment.

We make three contributions:

1) A discovery–valuation perspective on set-valued policy optimization. We identify rollout budget G and valuation horizon K as separate design choices, enabling a fixed pool of generated candidates to support independently chosen horizons of set valuation.

2) Exact set valuation translated into policy learning. We derive an exact determinantal evaluator revealing how K reweights candidate interactions, then convert a target allocation balancing task reward, set value, and local KL regularization into a policy-update signal.

3) Rollout-efficient quality–spread control and its limits. We show quality–spread gains at fixed rollout budgets and, in selected settings, with fewer rollouts, while highlighting candidate discovery and utility alignment as practical constraints on horizon control.

2 Related Work

Diversity-aware and set-level policy optimization. Methods for preserving diversity during language-model post-training include likelihood shaping, diversity-aware rewards, and representation-based exploration (He et al., 2025; Yao et al., 2026; Hu et al., 2026; Tuyls et al., 2026; Li et al., 2025). DQO and SetPO explicitly account for response interactions through leave-one-out credit: how the group value changes when a response is removed. DQO uses a log-determinant objective (Chen et al., 2026), whereas SetPO measures redundancy through kernelized response similarities (Li et al., 2026a). In both, the realized rollout group supplies the candidates and the context for their joint valuation. DeSPO instead optimizes probability allocations over those candidates through the expected set value of K draws with replacement, independently of the number G of real rollouts. Even when $K = G$, this averages over possible repeated-draw groups rather than evaluating only the observed group. For our determinantal objective, changing K reweights interactions involving different numbers of candidates rather than uniformly scaling a diversity bonus.

Multi-sample and inference-aware optimization. Pass@N and best-of-N optimization target the probability of obtaining a successful response or the quality of the best response among multiple samples (Walder & Karkhanis, 2026; Chow et al., 2025). More broadly, inference-aware methods adapt training objectives to the inference procedure (Balashankar et al., 2024; Tang et al., 2025;

Li et al., 2026b). These objectives concern the utility of fresh responses produced by the policy at inference. DeSPO's K instead controls repeated-draw valuation over an already observed candidate pool during training. Increasing K requires no additional generations and does not prescribe the fresh inference sample count N.

Local targets and policy updates. MPO and recent target- or listwise-policy formulations separate local target construction from policy fitting or projection (Abdolmaleki et al., 2018; Kaddour, 2026; Qu et al., 2026). DeSPO builds on this broad structure, with a complementary focus on the set value used to construct the target allocation. Its horizon-dependent objective accounts for interactions among repeated selections from the observed candidates, allowing the valuation horizon to be chosen independently of the discovery budget.

3 PRELIMINARIES

Candidate pool and set utility. For a prompt x, the rollout policy π_{old} generates G responses $Y = (y_1, \dots, y_G)$. Each response y_i receives task reward $r_i = r(x, y_i)$; these rewards form the vector $\mathbf{r} = (r_1, \dots, r_G)^\top$. Identical texts remain separate entries indexed by rollout. The valuation horizon K counts hypothetical draws with replacement from this pool during training; N counts fresh policy samples at inference. Using a frozen encoder f, we define unit embeddings ϕ_i, their similarity (Gram) matrix L_Y, and set utility $D(Y)$:

$$\phi_i = \frac{f(y_i)}{\|f(y_i)\|_2}, \qquad L_Y = (\phi_i^\top \phi_j)_{i,j=1}^G, \qquad D(Y) = \log\det(\mathbf{I}_G + L_Y).$$

The utility captures complementarity in embedding space, with diminishing gains from redundant responses (Kulesza & Taskar, 2012). Adding the identity $\mathbf{I}_G$ keeps the determinant positive even for duplicate or linearly dependent embeddings.

Realized-group credit. DQO augments each task reward by the utility lost when that response is removed (Chen et al., 2026). Writing Y_{-i} for Y without entry i,

$$\Delta_i^{\text{DQO}}(Y) = D(Y) - D(Y_{-i}), \qquad c_i^{\text{DQO}} = r_i + \alpha_D \Delta_i^{\text{DQO}}(Y).$$

Here $\alpha_D \geq 0$ weights the set contribution; c_i^{DQO} is the combined signal before optimizer-specific advantage and regularization processing. This credit is computed on the observed G-response group. DeSPO instead optimizes probability weights over the same candidates under an independently chosen valuation horizon K (Section 4.1).

4 DECOUPLED SET POLICY OPTIMIZATION

DeSPO separates candidate discovery, controlled by the rollout budget G, from set valuation under an independently chosen horizon K. For each prompt, the rollout policy generates G candidate responses $Y = (y_1, \dots, y_G)$. We keep these candidates, their rewards $\mathbf{r}$, and semantic geometry $L := L_Y$ fixed, and optimize a probability allocation $w = (w_1, \dots, w_G)$ over them, with $w_i \geq 0$ and $\sum_{i=1}^G w_i = 1$.

The rollout policy induces a length-normalized local score prior p over the observed candidate indices. We obtain it by normalizing their mean response-token log-probabilities:

$$p_i = \text{softmax}_i\left(\frac{1}{T_i}\sum_{t=1}^{T_i} \log \pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t})\right), \tag{1}$$

where $T_i > 0$ counts response tokens, excluding prompt and padding tokens.

We then solve for a target allocation $\widehat{w}_K$ that balances task reward, set value under horizon K, and proximity to p, without generating additional responses. In the full-target update, the displacement $\widehat{w}_K - p$ serves as a frozen signal guiding the actor's relative scores toward this target. Figure 2 contrasts this fixed-pool horizon valuation with DQO-style realized-group credit and summarizes how the resulting target allocation is transferred to the actor.

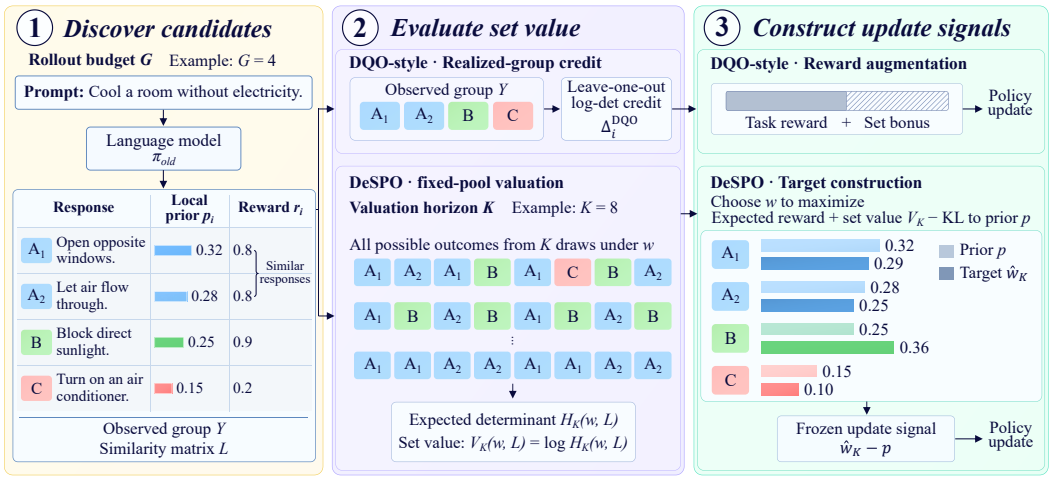

Figure 2: **From realized-group credit to fixed-pool valuation.** In the example, DQO augments task rewards with leave-one-out credit on the observed group. DeSPO instead evaluates allocations under K draws with replacement from the same G candidates. A target balancing reward, set value, and proximity to the prior yields the frozen update signal $\widehat{w}_K - p$, without additional generation.

4.1 Exact evaluation of candidate allocations

To compare allocations, we assess the geometric complementarity of K independent selections with replacement from the fixed pool. The allocation w determines the probability of each outcome. We compute the expected regularized determinant exactly, without Monte Carlo sampling:

$$H_K(w, L) = \mathbb{E}_{I_1,\dots,I_K \overset{\text{iid}}{\sim} \text{Cat}(w)} \left[\det(\mathbf{I}_K + L_{I_{1:K}, I_{1:K}})\right]. \tag{2}$$

These selections require no new text generation. Repeated indices are retained in $L_{I_{1:K}, I_{1:K}}$, so K may exceed G. We use $V_K(w, L) := \log H_K(w, L)$, taking the logarithm after averaging determinants. This form admits the exact finite-support expansion below, enabling deterministic target evaluation conditional on the observed pool.

To optimize w, we need repeated evaluations of H_K and its gradient. Direct enumeration involves G^K ordered outcomes. Expanding each determinant into principal minors and collecting contributions by candidate subset reduces the sum to at most 2^G terms:

Proposition 1 (Exact subset expansion). *For $L \succeq 0$ and integer $K \geq 1$,*

$$H_K(w, L) = \sum_{s=0}^{\min\{K,G\}} (K)_s \sum_{\substack{S \subseteq [G] \\ |S|=s}} \det(L_S) \prod_{i \in S} w_i, \tag{3}$$

where L_S is the submatrix indexed by S, $(K)_s = K!/(K-s)!$, and $(K)_0 = 1$. The empty subset contributes one, and terms above $\text{rank}(L)$ vanish.

Each term combines complementarity $\det(L_S)$ and allocation weight $\prod_{i \in S} w_i$. For $s = |S|$, $(K)_s$ counts the ways to place these s candidates in distinct positions among K draws. Unit-normalized embeddings give $L_{ii} = 1$, making the first-order term $K \sum_i w_i = K$ constant; only pairwise and higher-order interactions distinguish allocations. In our cached subset-enumeration implementation, each value-and-gradient evaluation costs $O(G2^G)$, so we use this exact evaluator for small pools. Appendices A and B give the derivations and implementation details.

How the value handles redundancy. In a simplified version of Figure 2, A_1 and A_2 have identical unit embeddings, while B is orthogonal to them. Writing $b = w_B$ and $w_{A_1} + w_{A_2} = 1 - b$ gives

$$H_K(w, L) = 1 + K + K(K-1)b(1-b). \tag{4}$$

The value depends on A_1 and A_2 only through their combined mass. Uniform candidate weights assign $2/3$ to their shared direction and $1/3$ to B, while geometry alone favors half the mass on each direction for $K > 1$ (Appendix A.4).

How K changes the objective. The example above has only two independent directions; general pools can also contribute higher-order interactions. In Equation 3, increasing K from 4 to 8 changes the pairwise and third-order coefficients from $(12, 24)$ to $(56, 336)$. Thus K reweights interaction orders, whereas the coefficient α below scales the same geometric objective at fixed K. These changes can alter allocation-gradient directions and are not generally reproducible by rescaling α (Appendix A.3).

Even at $K = G$, optimizing an allocation differs from assigning DQO's realized-group leave-one-out credit (Appendix A.5). We next combine V_K with task reward and KL regularization to construct the target allocation.

4.2 From target allocation to policy update

Using the geometric value V_K from Section 4.1, we seek a target among the candidate probability allocations defined above:

$$w_K^\star \in \arg\max_w \left\{ w^\top \mathbf{r} + \alpha V_K(w, L) - \tau \,\mathrm{KL}(w \| p) \right\}. \tag{5}$$

Here $\alpha \geq 0$ weights the geometric value and $\tau > 0$ penalizes departure from the local prior p. The reward term $w^\top \mathbf{r}$ is the expected reward per draw, equivalently the expected average reward over K independent draws, and is therefore independent of K. Intuitively, if p places excessive mass on redundant candidates, V_K makes repeated draws from the same semantic direction less valuable. The target therefore balances reward, geometric complementarity, and proximity to the local prior, rather than optimizing geometric spread alone.

We obtain a numerical target $\widehat{w}_K$ with a constrained solver initialized at p, using the exact finite-support objective and gradient. Implementation details are given in Appendix B.

The target can supply the full local update or augment an existing quality loss. The full-target update encourages higher relative scores for candidates whose target weights exceed their prior weights, and lower scores for those whose weights decrease. Let $z_{\theta,i}$ denote the mean response-token log-probability of y_i under π_θ, using the same token selection and normalization as Equation 1. We use $d = d_{\mathrm{full}}$ in the score-space loss

$$d_{\mathrm{full}} = \mathrm{stopgrad}(\widehat{w}_K - p), \qquad \ell_x(\theta; d) = -\lambda \sum_{i=1}^{G} d_i z_{\theta,i}, \tag{6}$$

where $\lambda > 0$ sets the update scale. The operator $\mathrm{stopgrad}$ preserves its argument's value but treats it as constant during backpropagation; gradients pass only through the actor scores. The displacement is recomputed at rollout refresh and held fixed for all actor updates on that rollout batch. If $\mathrm{softmax}(z_\theta) = p$, this loss has the same score-space gradient as λ-scaled cross-entropy fitting to $\widehat{w}_K$ (Appendix A.6).

Combining with an existing quality loss. To combine the target with an existing quality loss, we compare it with a reward-only allocation. Removing the geometric term from Equation 5 gives the reward-only target $\widehat{w}_Q$, with $\widehat{w}_{Q,i} \propto p_i \exp(r_i/\tau)$, normalized to sum to one. The full displacement decomposes as

$$\underbrace{\widehat{w}_K - p}_{d_{\mathrm{full}}} = \underbrace{\widehat{w}_Q - p}_{d_Q} + \underbrace{\widehat{w}_K - \widehat{w}_Q}_{d_{\mathrm{set}}}. \tag{7}$$

Here d_Q is the reward-only reallocation and d_{set} is the residual relative to that allocation. We apply $\mathrm{stopgrad}$ to d_{set} and add $\ell_x(\theta; d_{\mathrm{set}})$ to the existing quality loss. The native optimizer retains its own quality loss; $\widehat{w}_Q$ serves only as the allocation baseline. Appendix B.4 details the numerical constraints and implementation.

Training loop. At each rollout refresh, we generate G responses per prompt and recompute the targets and displacements. Changing K reuses those responses and embeddings. Inference samples normally from the full learned policy; Appendix B provides the training pseudocode.

5 EXPERIMENTS

Our evaluation addresses four questions.
Q1—Effectiveness. Does DeSPO improve quality–spread outcomes at matched rollout budgets?
Q2—Horizon control. Does K change the learned policy at fixed G and fitting interface?
Q3—Attribution. Can fitting or allocation movement alone explain the effect?
Q4—Persistence and task utility. Does the effect survive fresh sampling, and how does it interact with downstream task utility?

5.1 EXPERIMENTAL SETUP

Tasks and reference setup. We evaluate three open-ended tasks—constrained commonsense generation (CommonGen), instruction following (Dolly), and summarization (CNN/DailyMail)—and verifier-based mathematical reasoning (GSM8K) (Lin et al., 2020; Conover et al., 2023; See et al., 2017; Cobbe et al., 2021). We retain DQO's task-specific base models and quality-only references (Chen et al., 2026): Llama-3.2-1B-Instruct with PPO as the quality-only reference for open-ended tasks, and Qwen2.5-Math-1.5B with GRPO for GSM8K. We add a targeted Llama-3.2-3B-Instruct scale-up. DQO is the closest realized-group set-level comparator; SetPO provides an additional set-level baseline. Formal DeSPO uses the full-target score-space update in Equation 6 for both open-ended generation and GSM8K.

Evaluation and replication. Open-ended quality and semantic spread use Skywork Mean RM and MiniLM LogDet@16; GSM8K uses verifier-based Pass@N. G, K, and N denote real training rollouts per prompt, the training-side valuation horizon, and fresh inference samples. Each formal benchmark configuration uses three training seeds; we report means ± sample SD unless stated otherwise. Appendix C gives full protocols and implementation details.

5.2 MAIN RESULTS ON OPEN-ENDED GENERATION

Table 1 compares complete methods at two rollout budgets across three open-ended tasks.

Table 1: **Open-ended generation with 1B actors at $N = 16$.** Bold marks the highest observed mean within each rollout-budget block.

			CommonGen		Dolly		CNN/DailyMail	
Method	G	K	Mean RM ↑	LogDet@16 ↑	Mean RM ↑	LogDet@16 ↑	Mean RM ↑	LogDet@16 ↑
PPO	4	–	13.52±.22	5.075±.047	11.27±.10	4.270±.102	14.91±.08	3.915±.010
DQO	4	–	13.45±.37	5.681±.417	11.32±.17	4.314±.026	14.91±.07	3.930±.009
SetPO	4	–	13.32±.27	5.782±.333	7.13±.37	**4.572**±.086	14.94±.10	3.894±.027
DeSPO	4	4	**13.96**±.47	6.163±.203	**11.75**±.24	4.394±.018	**15.67**±.17	3.899±.022
DeSPO	4	8	13.89±.14	**6.890**±.220	11.48±.10	4.431±.091	15.41±.15	**3.932**±.048
DQO	8	–	14.39±.22	5.938±.050	11.75±.12	4.279±.026	15.47±.17	3.908±.017
SetPO	8	–	14.01±.16	5.448±.226	10.47±.93	4.383±.126	15.61±.17	3.885±.018
DeSPO	8	8	**16.22**±.50	**7.373**±.328	**12.33**±.46	**4.549**±.124	**16.39**±.12	**3.938**±.010

Matched rollout budgets. Both DeSPO settings have higher observed Mean RM than the $G = 4$ references across all tasks. Relative to DQO, CommonGen shows the clearest joint gains, Dolly a milder shift, and CNN/DailyMail primarily a quality gain. SetPO's higher spread on Dolly accompanies lower Mean RM, illustrating why the two axes should be read jointly.

Larger discovery. DeSPO $G8K8$ has higher observed Mean RM and LogDet@16 than DQO $G8$ on all three tasks. On CommonGen, $G4K8$ instead trades some quality for spread with half the rollouts. Lexical and geometric metrics corroborate that spread shift (Appendix D); the full $G \times K$ grid and cost accounting separate discovery from valuation (Appendix E). These are complete-method comparisons; the following controls isolate design choices.

5.3 Separating candidate discovery from set valuation

Horizon control. At fixed discovery budget and fitting interface, varying K traces a nonmonotonic quality–spread trajectory (Figure 3a). Moving from $K = 4$ to $K = 8$ expands spread with little quality change at $G = 4$ and improves both axes at $G = 8$; extending to $K = 16$ favors spread at a quality cost. The horizon thus controls the learned operating point rather than acting as a larger-is-better hyperparameter. This accords with the finite-support expansion: K reweights interaction orders instead of uniformly rescaling a fixed set term. The positive $G = 8$ contrasts have hierarchical 95% intervals excluding zero (Appendix E).

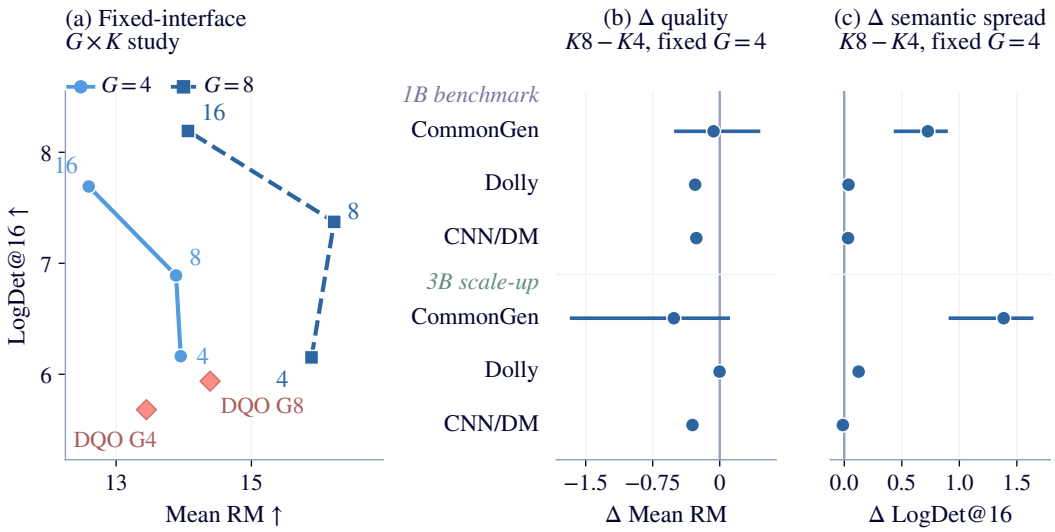

Figure 3: **Discovery–valuation decoupling.** (a) CommonGen 1B operating points from varying K at fixed G and fitting interface. (b–c) Direct $K8 - K4$ changes at $G = 4$ across tasks and scales. Points show means; bars denote available hierarchical 95% intervals; unbarred points have no interval estimates. Full $G \times K$ results and exact contrasts appear in Appendix E.

Discovery remains complementary. At fixed $K = 8$, reducing discovery to $G = 2$ preserves much of the spread but sharply lowers quality (Appendix E). Together with the fixed-G horizon sweep, this separates the roles empirically: G determines what support becomes available, whereas K determines how it is valued. Reallocation cannot recover candidates absent from that support.

Scale-up and task dependence. At 3B, CommonGen $G4K8$ improves quality and spread over DQO $G8$ with half the real training rollouts (Figure 1). Dolly adds a nearly quality-neutral $K4 \to K8$ spread gain. Its $G4K8$ also exceeds DQO $G8$ in Mean RM, Best@16, Distinct-2, and MiniLM/BGE spread; all five hierarchical 95% intervals exclude zero (Appendix D). CNN/DailyMail instead favors $K4$. BGE-M3 preserves all three horizon-induced spread directions (Appendix G). Control also persists at $G = 8$: $K8 \to K16$ raises spread with nearly unchanged reward on CNN/DailyMail, but lower observed reward on Dolly (Appendix E).

5.4 What drives horizon-aware reallocation?

Four controls test four competing explanations for the horizon effect: fitting alone, larger allocation movement, off-diagonal semantic geometry, and the determinant's nonseparable form (Figure 4a). **Reward-only** retains fitting but removes the set term; **KL-matched K8** retains the set term. Both match $K4$'s median target KL on a train-only calibration pool; **No-geometry** removes off-diagonal similarity while retaining K-aware cross-index interactions; and **Occupancy** uses separable expected occupancy in place of the determinant.

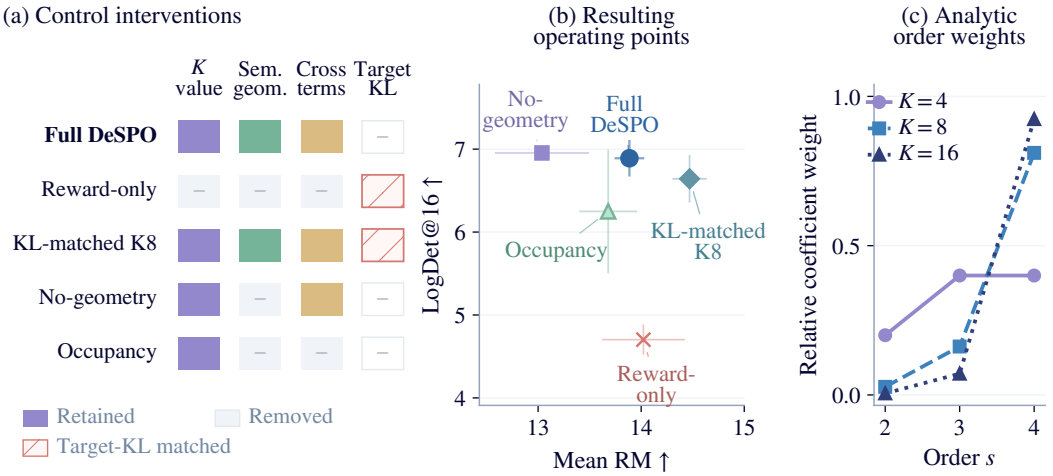

Figure 4: **Interventions, outcomes, and horizon structure.** (a) Component colors and filled/pale cells mark retention/removal; hatching marks median target-KL matching on training prompts. Reward and fitting are shared. (b) CommonGen 1B, $G = 4$, $K = 8$ where applicable: means with sample-SD bars. (c) Falling-factorial weights normalized over orders 2–4, excluding geometry and allocation factors; these are not measured gradient contributions.

Train-only target-KL matching sharpens the attribution. Removing the set term reduces semantic spread, while $K8$ matched to $K4$'s median calibration-pool KL retains a positive quality–spread shift; BGE-M3 preserves the spread direction (Appendix F). This weakens fitting-only and target-KL-magnitude explanations. No-geometry and Occupancy retain substantial spread, while Full DeSPO has higher observed Mean RM than No-geometry. Together with fixed-interface K sweeps, the controls support horizon-aware fixed-support valuation under the tested fitting rule, while leaving the utility form open. Figure 4c shows analytic order weights. Coefficient/decoding sweeps trace different trajectories (Appendix E), and simpler utilities have nearly collinear local directions under calibrated strong KL (Appendix F), without implying end-to-end equivalence.

5.5 Inference budgets and reasoning results

From local valuation to learned-policy behavior. A training-side control must affect outputs beyond its local candidate pool. Extending the fresh-output evaluations above across N reveals similar Best@N trajectories for $K4/K8$, while $K8$ maintains greater spread (Figure 5). The horizon-induced shift thus extends from local reallocation to fresh outputs of the learned policy across inference budgets. LogDet comparisons use matched N because of set-size dependence; Appendix E includes $K16$.

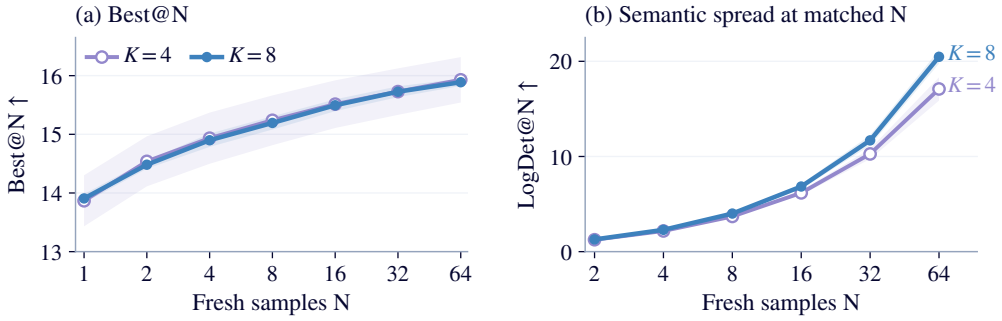

Figure 5: **Training-horizon effects under fresh sampling.** CommonGen 1B curves for DeSPO $G4K4/K8$ from a separate budget-curve evaluation. Lines show means and bands show sample SD. LogDet is compared only within matched N.

Persistence and task utility. Persistence alone does not guarantee task value. We evaluate the same full-target DeSPO update on verifier-based mathematical reasoning. On GSM8K, DeSPO $G4K4$ approaches the larger-rollout DQO $G8$ curve as N grows, while $K8$ yields no additional observed correctness gain across the evaluated budgets (Figure 6).

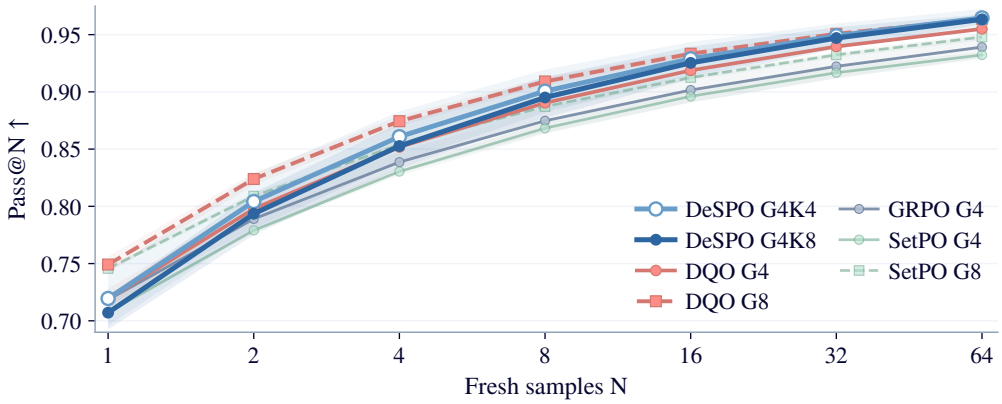

Figure 6: **GSM8K performance across fresh inference budgets.** Qwen2.5-Math-1.5B; formal DeSPO curves use the full-target update in Equation 6. Lines show mean Pass@N with sample-SD ribbons. $G8$ references use twice the real training rollout budget.

A cached-pool diagnostic finds more multi-correct groups at larger discovery budgets. Separately, a residual-interface gradient diagnostic finds the $K8 - K4$ structural increment negatively aligned with the native quality gradient across three checkpoints, while the $K4/K8$ structural directions remain highly aligned. This motivates examining utility alignment without causally explaining the formal $K4/K8$ difference. Neither diagnostic measures $G8K16$ policy performance; that arm was not trained (Appendix H).

Together, the evaluations separate persistence from task utility: a horizon-induced policy shift can survive fresh sampling, while its downstream value depends on alignment between the set utility and the task objective.

6 Discussion

On CommonGen, $K = 8$ yields higher mean semantic spread than $K = 4$ at every tested decoding temperature, while increasing DeSPO's α trades quality for spread (Appendix E.3). BGE-M3 preserves the cross-task directions of horizon-induced spread changes (Appendix G), supporting consistency under independent semantic evaluation.

Limitations. Local reallocation is restricted to discovered candidates, and its downstream value depends on alignment between the set utility and task objective. Our current exact subset evaluator scales exponentially with G. These constraints motivate richer discovery, task-aligned utilities, and scalable valuation for larger candidate pools.

7 Conclusion

We introduced DeSPO, decoupling the rollout budget G from the set-valuation horizon K in LLM post-training. By valuing repeated selections from a fixed candidate pool, DeSPO constructs horizon-aware policy targets without additional generation. Our exact determinantal expansion shows that changing K reweights interaction orders and is not generally equivalent to rescaling a diversity bonus. Across tasks and model scales, DeSPO achieves quality–spread gains at matched rollout budgets and, in selected settings, outperforms baselines using more training rollouts. Horizon-induced shifts persist under fresh sampling, extending the effects of local reallocation to the learned policy. Together, these findings establish a complementary route to rollout-efficient post-training: improving how generated candidates are valued, not only how many are discovered.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript restructuring and editing, mathematical verification, reference checking, and the development of software, plotting, and experiment-support code. The proposed method, experimental design, and interpretation of the reported results were developed by the authors. All AI-assisted mathematical and software outputs were reviewed and verified by the authors before use, and all reported measurements come from completed evaluations. The authors take responsibility for the final text, claims, code, and results.

REPRODUCIBILITY STATEMENT

The paper and appendices document the mathematical formulation, algorithm, implementation details, training and evaluation protocols, and statistical procedures used in this work. Appendix A provides mathematical derivations and the numerical scope of finite-support target computation; Appendix B describes the algorithm and implementation. Reproducing training also requires the training code, configurations, checkpoints, raw generations, and evaluator revisions. Appendix C documents the experimental protocols and remaining provenance boundaries; Appendix H distinguishes formal reasoning experiments from separate mechanism diagnostics.

ETHICS STATEMENT

DeSPO optimizes the specified task reward and semantic geometry. Downstream applications should separately evaluate factuality, safety, fairness, and user preferences using domain-appropriate criteria and safeguards. Pretrained actors, reward models, and semantic encoders may inherit biases from their training data.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation, 2018.

Ananth Balashankar, Ziteng Sun, Jonathan Berant, Jacob Eisenstein, Michael Collins, Adrian Hutter, Jong Lee, Chirag Nagpal, Flavien Prost, Aradhana Sinha, et al. Infalign: Inference-aware language model alignment. 2024.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the association for computational linguistics: ACL 2024*, pp. 2318–2335, 2024.

Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Ioannis Paschalidis, and Aldo Pacchiano. Post-training large language models for diverse high-quality responses. In *International Conference on Learning Representations*, volume 2026, pp. 127946–127975, 2026.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Aviral Kumar, Rishabh Agarwal, Sridhar Thiagarajan, Craig Boutilier, and Aleksandra Faust. Inference-aware fine-tuning for best-of-n sampling in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 78936–78959, 2025.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned LLM, 2023.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

Wei Gao, Yuheng Zhao, Dakai An, Tianyuan Wu, Lunxi Cao, Shaopan Xiong, Ju Huang, Weixun Wang, Siran Yang, Wenbo Su, et al. RollPacker: Taming Long-Tail rollouts for RL Post-Training with tail batching. In *23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI 26)*, pp. 849–866, 2026.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25559–25571, 2025.

Zican Hu, Shilin Zhang, Yafu Li, Jianhao Elliott Yan, Xuyang Hu, Leyang Cui, Xiaoye Qu, Chunlin Chen, Yu Cheng, and Zhi Wang. Diversity-incentivized exploration for versatile reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 71175–71205, 2026.

Jean Kaddour. Target policy optimization, 2026.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. In *International Conference on Learning Representations*, volume 2024, pp. 20620–20653, 2024.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. Setpo: Set-level policy optimization for diversity-preserving llm reasoning. *arXiv preprint arXiv:2602.01062*, 2026a.

Muheng Li, Jian Qian, and Wenlong Mou. What should post-training optimize? a test-time scaling law perspective. *arXiv preprint arXiv:2605.10716*, 2026b.

Tianjian Li, Yiming Zhang, Ping Yu, Swarnadeep Saha, Daniel Khashabi, Jason Weston, Jack Lanchantin, and Tianlu Wang. Jointly reinforcing diversity and quality in language model generations. 2025.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. Commongen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1823–1840, 2020.

Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, et al. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. In *International Conference on Learning Representations*, volume 2026, pp. 133805–133838, 2026.

Meta. Llama-3.2-1B-Instruct model card. Hugging Face model card, 2024. URL `https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct`. Accessed 21 September 2026.

Laura O'Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, volume 2, pp. 2, 2024.

Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Yingyue Li, Wutong Xu, Lizhou Cai, Weijie Liu, Clive Bai, et al. Listwise policy optimization: Group-based rlvr as target-projection on the llm response simplex. *arXiv preprint arXiv:2605.06139*, 2026.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

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

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Rémi Munos. Optimizing language models for inference time objectives using reinforcement learning. 2025.

Jens Tuyls, Dylan Foster, Akshay Krishnamurthy, and Jordan Ash. Representation-based exploration for language models: From test-time to post-training. In *International Conference on Learning Representations*, volume 2026, pp. 120621–120645, 2026.

Christian Walder and Deep Tejas Karkhanis. Pass@ k policy optimization: Solving harder reinforcement learning problems. volume 38, pp. 152416–152445, 2026.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning. volume 38, pp. 94801–94826, 2026.

A Exact computation and mathematical scope

Throughout this appendix, the observed candidates, reward vector $\mathbf{r}$, Gram matrix $L \succeq 0$, and prior p are fixed. Allocations satisfy $w_i \geq 0$ and $\sum_i w_i = 1$, and the valuation horizon $K \geq 1$ is an integer.

A.1 Proof of the finite-support expansion

For a fixed index sequence $I_{1:K}$, the principal-minor identity gives

$$\det(\mathbf{I}_K + L_{I_{1:K},I_{1:K}}) = \sum_{A \subseteq [K]} \det(L_{I_A,I_A}). \tag{8}$$

A term whose index sequence I_A repeats a candidate has identical rows and determinant zero. Fix a subset $S \subseteq [G]$ of s distinct candidates, with $s \leq K$. There are $\binom{K}{s}$ choices of positions and $s!$ bijections from S to those positions. After marginalizing the unused draws, each bijection has probability $\prod_{i \in S} w_i$ and determinant $\det(L_S)$. Taking expectations and grouping contributions by S yields

$$H_K(w, L) = \sum_{s=0}^{\min\{K,G\}} (K)_s \sum_{\substack{S \subseteq [G] \\ |S|=s}} \det(L_S) \prod_{i \in S} w_i, \tag{9}$$

where $(K)_s = K!/(K-s)!$ and $(K)_0 = 1$. We set $(K)_s = 0$ for $s > K$. The empty subset contributes one, and principal minors above $\operatorname{rank}(L)$ vanish. This proves Equation 3. The iid assumption concerns the index draws; the candidate embeddings are held fixed.

Analytic gradient. Differentiating this polynomial before imposing the sum constraint gives

$$\frac{\partial H_K}{\partial w_i} = \sum_{\substack{S \subseteq [G],\, i \in S \\ |S| \leq K}} (K)_{|S|} \det(L_S) \prod_{j \in S \setminus \{i\}} w_j. \tag{10}$$

All principal minors of L are nonnegative, so $H_K \geq 1$ on the simplex. Writing the objective in Equation 5 as $J_K(w) = w^\top \mathbf{r} + \alpha \log H_K(w, L) - \tau \operatorname{KL}(w \| p)$, its gradient at positive weights and a positive prior is

$$\nabla_w J_K = \mathbf{r} + \alpha \frac{\nabla_w H_K}{H_K} - \tau\big(\log w - \log p + \mathbf{1}\big), \tag{11}$$

where logarithms act coordinatewise. In simplex coordinates $u_i = w_i$ for $i < G$ and $w_G = 1 - \sum_{i<G} u_i$, the derivative is $\partial_{u_i} J_K = \partial_{w_i} J_K - \partial_{w_G} J_K$.

A.2 Repeated candidates and the two-candidate calculation

Let $\phi_i \in \mathbb{R}^d$ be the embedding of candidate i, and let Φ_I have rows $\phi_{I_t}^\top$. Sylvester's determinant identity gives

$$\begin{aligned}\det(\mathbf{I}_K + \Phi_I \Phi_I^\top) &= \det(\mathbf{I}_d + \Phi_I^\top \Phi_I) \\ &= \det\left(\mathbf{I}_d + \sum_i n_i \phi_i \phi_i^\top\right),\end{aligned} \tag{12}$$

where n_i counts occurrences of candidate i among the K draws. Repeated candidates therefore enter the value through their occurrence counts.

For two unit embeddings with inner product ρ, the determinant is $1+n_1+n_2+(1-\rho^2)n_1n_2$. Under allocation $(u, 1-u)$, we have $n_1 + n_2 = K$ and $\mathbb{E}[n_1 n_2] = K(K-1)u(1-u)$, giving

$$H_K((u, 1-u), L) = 1 + K + K(K-1)(1-\rho^2)u(1-u). \tag{13}$$

The allocation-dependent term combines geometric complementarity $1-\rho^2$ with mass assigned to both directions.

If all unit embeddings are identical, then $L = \mathbf{1}\mathbf{1}^\top$ and $H_K = 1 + K$. If $K = 1$ and $L_{ii} = 1$, then $H_1 = 2$ for every allocation. In both cases, the geometric value is constant on the simplex.

Replacing an embedding by identical copies whose weights sum to its original weight leaves the distribution of $\sum_{t=1}^K \phi_{I_t}\phi_{I_t}^\top$ unchanged and therefore preserves H_K. This invariance concerns the geometric value under the specified weight split; the local prior and actor coefficients are constructed over the indexed candidates.

A.3 Why the horizon is not generally a scalar coefficient

For $G = 3$ and $L = \mathbf{I}_3$, define $e_2 = w_1w_2 + w_1w_3 + w_2w_3$ and $e_3 = w_1w_2w_3$. On the simplex,

$$H_K = 1 + K + (K)_2 e_2 + (K)_3 e_3. \tag{14}$$

At $w = (1/2, 1/3, 1/6)$, the values are $H_3 = 6$ and $H_8 = 319/9$. Using coordinates (u, v) with $w = (u, v, 1-u-v)$, the log-value gradients at this allocation are

$$\begin{aligned}\nabla_{u,v} \log H_3 &= (-4/9, -1/4), \\ \nabla_{u,v} \log H_8 &= (-504/319, -336/319).\end{aligned} \tag{15}$$

These vectors are not collinear. Any relation $V_8(w, L) = aV_3(w, L) + c$ with constants a, c would make their simplex gradients proportional at every allocation. Thus changing K can change the geometric update direction and cannot in general be absorbed into a scalar adjustment of α. Figure 4(c) illustrates the underlying reweighting through the displayed-order normalized coefficients $(K)_s / \sum_{j=2}^4 (K)_j$ for $s = 2, 3, 4$ at $G = 4$.

A.4 Scalar reduction of the worked example

For the idealized three-candidate example in Section 4.1, take rewards $r_i = 0.9$, prior $p_i = 1/3$, and

$$L = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \alpha = 1.$$

Write $b = w_B$. For fixed mass $1-b$ on the shared direction, reward and geometry are independent of how that mass is split between A_1 and A_2. The KL penalty is minimized at $w_{A_1} = w_{A_2} = (1-b)/2$. The target problem therefore reduces to

$$H_K(b) = 1 + K + K(K-1)b(1-b), \tag{16}$$

$$J_K(b) = 0.9 + \alpha \log H_K(b) - \tau\left[(1-b)\log\frac{3(1-b)}{2} + b\log(3b)\right]. \tag{17}$$

For $\tau > 0$, this objective is strictly concave on $0 < b < 1$: H_K is positive and concave, its logarithm is concave, and the negative KL term is strictly concave. For $K > 1$, the derivative is positive at $b = 1/3$ and negative at $b = 1/2$. Hence the unique maximum satisfies

$$\alpha \frac{K(K-1)(1-2b)}{H_K(b)} - \tau \log \frac{2b}{1-b} = 0, \qquad b \in (1/3, 1/2). \tag{18}$$

The target therefore lies between uniform candidate allocation ($b = 1/3$) and equal mass on the two geometric directions ($b = 1/2$). For fixed $K > 1$, the solution approaches $1/3$ as $\tau \to \infty$ and $1/2$ as $\tau \to 0^+$. At $K = 1$, geometry is constant and the optimum is $b = 1/3$. This rank-two example explains redistribution between redundant and complementary candidates; Appendix A.3 establishes the general distinction between changing K and rescaling α.

A.5 Effective set reward and KL-regularized allocation

Define the utility $F_K(w) = w^\top \mathbf{r} + \alpha \log H_K(w, L)$, so that $J_K(w) = F_K(w) - \tau \,\mathrm{KL}(w\|p)$. Its marginal value for candidate i is

$$R_i^{\mathrm{set}}(w) = r_i + \alpha\, \partial_{w_i} \log H_K(w, L). \tag{19}$$

For any feasible infinitesimal transfer of mass from candidate j to i,

$$\frac{d}{d\epsilon} F_K\big(w + \epsilon(\mathbf{e}_i - \mathbf{e}_j)\big)\Big|_{\epsilon=0} = R_i^{\mathrm{set}}(w) - R_j^{\mathrm{set}}(w), \tag{20}$$

where $\mathbf{e}_i$ denotes the ith coordinate unit vector. Relative effective rewards determine local utility gains. Adding the same constant to every effective reward leaves these directional derivatives unchanged. At interior weights, the corresponding derivative of J_K additionally contains $-\tau\{\log(w_i/p_i) - \log(w_j/p_j)\}$.

Stationary allocation. Assume $p_i > 0$ and $\tau > 0$. At an interior target $w_K^\star$, the simplex first-order conditions are

$$R_i^{\mathrm{set}}(w_K^\star) - \tau\big(\log(w_{K,i}^\star/p_i) + 1\big) - \nu = 0, \tag{21}$$

where ν enforces the sum constraint. Exponentiating and normalizing gives

$$w_{K,i}^\star = \frac{p_i \exp(R_i^{\mathrm{set}}(w_K^\star)/\tau)}{\sum_j p_j \exp(R_j^{\mathrm{set}}(w_K^\star)/\tau)}. \tag{22}$$

These conditions also hold at any interior stationary allocation. The effective rewards depend on the allocation itself, so this is a coupled stationary relation. Evaluating the marginals only once at p generally gives a different target. For a numerical target with inactive probability bounds, stationarity can be assessed through the coordinate range of $R_i^{\mathrm{set}}(\widehat{w}_K) - \tau \log(\widehat{w}_{K,i}/p_i)$, which vanishes at exact stationarity.

Small-step interpretation. For fixed interior p and smooth F_K, consider a stationary branch approaching p as $\tau \to \infty$, with $K, \alpha, \mathbf{r}, L, p$ held fixed. Expanding Equation 22 in $1/\tau$ gives

$$w_{K,i}^\star - p_i = \frac{p_i}{\tau}\left[R_i^{\mathrm{set}}(p) - \sum_j p_j R_j^{\mathrm{set}}(p)\right] + O(\tau^{-2}). \tag{23}$$

To first order, probability moves toward candidates whose effective rewards exceed the mean under p.

For any numerical target on the simplex, define $A_i^{\mathrm{alloc}} = \widehat{w}_{K,i}/p_i - 1$. Holding these coefficients fixed, the full-target loss can be written exactly as

$$\ell_x(\theta; d_{\mathrm{full}}) = -\lambda \sum_i p_i A_i^{\mathrm{alloc}} z_{\theta,i}, \qquad \sum_i p_i A_i^{\mathrm{alloc}} = 0. \tag{24}$$

This expresses the signed displacement in Equation 6 as a centered allocation signal weighted by the local prior. The native-quality interface uses the residual $\widehat{w}_K - \widehat{w}_Q$ defined in Appendix B.4.

Relation to leave-one-out determinant credit. For the realized indexed group Y, DQO uses a contribution of the form $r_i + \alpha_D[D(Y) - D(Y_{-i})]$, with $D(Y) = \log\det(\mathbf{I} + L_Y)$ in the regularized log-determinant implementation. The subtraction measures the effect of removing candidate i from that realized group. DeSPO's effective reward differentiates the conditional valuation with respect to local allocation weights while holding the candidate pool fixed. The geometric objectives also differ: in general,

$$\log \mathbb{E}[\det(\mathbf{I}_K + L_{I_{1:K},I_{1:K}})] \neq \mathbb{E}[\log\det(\mathbf{I}_K + L_{I_{1:K},I_{1:K}})]\,.$$

Setting $K = G$ still allows repeated candidate indices and retains the allocation-based marginal rewards.

A.6 Frozen actor update and target-fitting interpretation

Let $q_{\theta,i} = \exp(z_{\theta,i})/\sum_j \exp(z_{\theta,j})$ and hold both p and $\widehat{w}_K$ fixed. Write $\mathcal{L}_{\text{frozen}} = \ell_x(\theta; d_{\text{full}})$. Categorical cross-entropy $\mathcal{L}_{\text{CE}} = -\lambda \sum_i \widehat{w}_{K,i} \log q_{\theta,i}$ has gradient

$$\nabla_\theta \mathcal{L}_{\text{CE}} = \lambda \sum_i (q_{\theta,i} - \widehat{w}_{K,i}) \nabla_\theta z_{\theta,i}. \tag{25}$$

The frozen loss has gradient

$$\nabla_\theta \mathcal{L}_{\text{frozen}} = \lambda \sum_i (p_i - \widehat{w}_{K,i}) \nabla_\theta z_{\theta,i}. \tag{26}$$

Their score-space gradients are therefore

$$\begin{aligned} \nabla_z \mathcal{L}_{\text{frozen}} &= \lambda(p - \widehat{w}_K), \\ \nabla_z \mathcal{L}_{\text{CE}} &= \lambda(q_\theta - \widehat{w}_K), \end{aligned} \tag{27}$$

and their difference is

$$\nabla_z \mathcal{L}_{\text{frozen}} - \nabla_z \mathcal{L}_{\text{CE}} = \lambda(p - q_\theta). \tag{28}$$

Thus the score-space and parameter gradients agree whenever $q_\theta = p$.

In the implementation, p uses token log-probabilities recorded by the rollout backend, while z_θ is recomputed by the training model. Both use the same candidate tokens, response mask, and mean-token normalization. The gradient identity is conditional on the resulting local distributions matching.

Cross-entropy keeps the target fixed while its score gradient tracks q_θ. The frozen loss keeps $p - \widehat{w}_K$ fixed throughout the actor updates on a rollout batch, while its parameter gradient can change through the score Jacobian. This target-fitting interpretation concerns the local distribution over the observed candidate indices.

B Algorithm and implementation details

This appendix describes the full-target interface used by formal DeSPO experiments in both open-ended generation and GSM8K, together with a separate native-quality residual interface used only in GSM8K diagnostics. Both use the target construction in Section 4.2, with the actor losses and settings specified below.

B.1 Target construction and exact evaluation

Scores, rewards, and geometry. Equation 1 uses token log-probabilities recorded by the rollout backend. The response mask $m_{i,t}$ is taken from the response span of the attention mask; the averaging denominator is $\max(1, T_i)$, where $T_i = \sum_t m_{i,t}$. There is no additional temperature in this candidate-local softmax. The reported runs use rollout temperature 1, top-p equal to 1, and unrestricted top-k sampling. The actor recomputes its token log-probabilities with the training model, using the same response mask and mean-token normalization (Appendix A.6).

The target receives the task rewards directly, with both reward-scaling factors set to one. The semantic Gram matrix uses unit-normalized MiniLM embeddings.

Exact valuation and caching. For $G \leq 16$, our subset-enumeration implementation evaluates the complete expansion in Equation 3 using CPU float64 computations in NumPy/SciPy. Each prompt-level solve constructs its principal-minor terms once and reuses them throughout the objective and gradient evaluations. Direct computation of these minors in our implementation costs at most $O(G^3 2^G)$ per solve.

Numerical target. The solver uses SLSQP, initialized at p, with the analytic objective gradient in Equation 11. It imposes coordinate bounds $[10^{-12}, 1]$ and the equality $\sum_i w_i = 1$, with function tolerance 10^{-12} and an iteration limit of 500. The returned coordinates are clipped below at 10^{-15} and normalized to sum to one. An unsuccessful SLSQP status raises an error in reward computation. Exactness refers to evaluating the finite-support objective and gradient; a successful SLSQP solve is not a certificate of global target optimality or of improvement in the full language-model policy.

B.2 Full-target actor update

Using the frozen displacement in Equation 6, the loss for an actor minibatch of B_p prompt groups is

$$\mathcal{L}_{\text{full}} = -\frac{\lambda}{B_p} \sum_{u=1}^{B_p} \sum_{i=1}^{G} d_{\text{full},u,i} z_{\theta,u,i}. \tag{29}$$

Here u indexes prompts and i their candidates. Response scores average over valid tokens, candidate contributions are summed, and prompt contributions are averaged. Microbatch accumulation preserves this minibatch reduction.

Formal interfaces. Both open-ended and GSM8K DeSPO optimize Equation 29 with $\lambda = 1$. The PPO/GRPO policy surrogate, entropy term, and actor reference-KL term have zero contribution to this actor loss. Task quality enters through $\mathbf{r}$ in the allocation target. Open-ended runs also update an independent critic whose predictions and loss do not enter the actor gradient; formal GSM8K disables the critic. Its GRPO policy loss is computed and logged with zero actor-loss weight: the formal update is listwise-only. Appendix C.1 gives the GSM8K settings.

Open-ended rollout and actor schedule. Each rollout batch contains 128 prompts and $128G$ responses. The prior, target, and displacement are computed once for each prompt and remain fixed during the actor updates on that batch. One actor pass splits the batch into four minibatches of 32 complete prompt groups, each containing $32G$ responses. Microbatches accumulate gradients within an actor minibatch, followed by one optimizer step. Thus each rollout batch produces four actor optimizer steps, and each prompt group is used once in that pass. After these updates, the actor weights are synchronized to the rollout backend for the next batch.

The CommonGen configuration uses three passes through the prompt dataset, totaling 186 rollout/global steps. These dataset epochs are separate from the single actor pass per rollout batch, configured as `ppo_epochs=1`. Table B1 summarizes the full-target settings. The CommonGen actor uses AdamW with betas $(0.9, 0.999)$, weight decay 0.01, a constant learning-rate schedule, and no warmup.

Table B1: Full-target training settings for the open-ended tasks.

Setting	Value
Target coefficients	$\alpha = 1, \tau = 600$
Actor loss scale	$\lambda = 1$
Rollout batch	128 prompts, $128G$ responses
Actor minibatch	32 prompt groups, $32G$ responses
Actor passes per rollout	1
Actor steps per rollout	4
Response microbatch per GPU	CommonGen/Dolly: 4; CNN/DailyMail: 1
Actor / critic learning rate	10^{-6} / 10^{-5}
Actor gradient clipping	Global norm threshold 1.0

Algorithm 1 summarizes the open-ended full-target training loop. The target solver includes the bounds and postprocessing described above.

Algorithm 1 Open-ended full-target training with exact fixed-pool valuation

Inputs: Actor π_θ, rollout count G, horizon K, and coefficients α, τ, λ.
For each rollout batch $\mathcal{B}$ of 128 prompts:
  Generate G responses per prompt using the rollout backend.
  For each prompt $u \in \mathcal{B}$:
    Compute $\mathbf{r}_u$, normalized embeddings, L_u, and the rollout-score prior p_u.
    Cache the principal-minor terms for this target solve.
    Obtain $\widehat{w}_{K,u}$ and solver status by SLSQP, initialized at p_u.
    If the solver is unsuccessful, **raise an error**.
    Store $d_{\mathrm{full},u} = \mathrm{stopgrad}(\widehat{w}_{K,u} - p_u)$.
  End for.
  Compute the value-training quantities and update the independent critic.
  For each minibatch of 32 prompt groups, in one pass:
    Clear actor gradients.
    Accumulate gradients of Equation 29 across response microbatches, using the minibatch normalizer B_p.
    Clip the accumulated gradient norm at 1.0 and take one actor optimizer step.
  End for.
  Synchronize actor weights to the rollout backend.
End for.

B.3 Reward-only target computation

For $p_i > 0$ and $\tau > 0$, the reward-only objective in Section 4.2 is strictly concave on the simplex. Its first-order condition is $\log(w^\star_{Q,i}/p_i) = r_i/\tau + c$. Normalizing gives the unique optimum

$$w^\star_{Q,i} = \frac{p_i \exp(r_i/\tau)}{\sum_j p_j \exp(r_j/\tau)}. \tag{30}$$

We compute $\widehat{w}_Q$ by applying a stable softmax to $\log p + \mathbf{r}/\tau$.

The decomposition in Equation 7 applies algebraically to the returned targets. Their unit sums imply $\sum_i d_{\mathrm{set},i} = 0$, while the reward change $\mathbf{r}^\top d_{\mathrm{set}}$ can have either sign.

The two targets use the same candidates, rewards, prior, and τ. SLSQP imposes a 10^{-12} coordinate lower bound for the full target, while the reward-only target is evaluated from the analytic simplex solution. If the lower bound is active, it also contributes to the difference between the numerical targets. At $\alpha = 0$, the ideal targets on the full simplex coincide.

B.4 Native-quality score-space residual

This interface is used only for the GSM8K diagnostics in Appendices H.3–H.4; formal Figure 6 checkpoints use the full-target update.

Using the frozen residual from Equation 7, the minibatch structural loss and total actor loss are

$$\begin{aligned} \mathcal{L}_S &= -\frac{1}{B_p} \sum_{u=1}^{B_p} \sum_{i=1}^{G} d_{\mathrm{set},u,i} z_{\theta,u,i}, \\ \mathcal{L}_{\mathrm{actor}} &= \mathcal{L}_{\mathrm{native}} + \lambda_t \mathcal{L}_S. \end{aligned} \tag{31}$$

The score-space structural term contains no token probability ratios or ratio clipping.

In the GSM8K diagnostic, $\mathcal{L}_{\mathrm{native}}$ is the PPO clipped surrogate with GRPO advantages and clipping parameter 0.2. Group normalization is applied to the native advantages, while target construction uses the task rewards described above. The old log-probabilities are recomputed by the training model once per rollout batch and held fixed for the actor updates. The critic is disabled, and the entropy and actor reference-KL terms are disabled. The coefficient λ_t is chosen for each actor minibatch by the gradient budget below.

B.5 GRADIENT-BUDGET DIAGNOSTIC

The GSM8K diagnostic uses Qwen2.5-Math-1.5B, $G = 4$, one shared training seed, and 48 training steps per run. It compares native GRPO, score-space residuals at $K = 4$ and $K = 8$, and a clipped structural control at $K = 8$. The target coefficients are $\alpha = 1$ and $\tau = 600$. The diagnostic runs use one GPU, response microbatches of size 2, actor learning rate 10^{-6}, and global gradient clipping at 1.0.

For the same actor parameters and minibatch, let $g_Q = \nabla_\theta \mathcal{L}_{\text{native}}$ and $g_S = \nabla_\theta \mathcal{L}_S$, with $\mathcal{L}_S$ evaluated at unit coefficient. The minibatch scale and combined gradient are

$$\lambda_t = \min\left(\lambda_{\max}, \frac{\rho\|g_Q\|_2}{\|g_S\|_2 + \epsilon}\right), \qquad g_{\text{total}} = g_Q + \lambda_t g_S. \tag{32}$$

The diagnostic sets $\rho = 0.25$ and $\lambda_{\max} = 783.876075$. In the current implementation, $\epsilon = 10^{-12}$. The coefficient is a detached scalar.

Norms are measured after all microbatches in the actor minibatch have accumulated, before global clipping and optimizer preconditioning. The implementation first probes the quality gradient and clears it, then accumulates the unscaled structural gradient. After computing λ_t, it scales the stored structural gradient and backpropagates the quality loss again to form g_{total}. The combined gradient is then clipped and passed to the optimizer. This construction gives

$$\|\lambda_t g_S\|_2 \le \rho\|g_Q\|_2. \tag{33}$$

The bound concerns the accumulated actor-parameter gradients before clipping and preconditioning. We record both branch norms, their cosine, λ_t, cap activation, the scaled structural-to-quality norm ratio, and the total gradient norm before and after clipping.

Clipped structural control. The matched $K = 8$ control broadcasts the response-level residual to response tokens and uses a PPO ratio-clipped structural surrogate, with `seq-mean-token-mean` aggregation. It retains the same native quality loss and gradient-budget rule, with g_S computed from this clipped structural loss. The score-space and clipped controls therefore share target construction and the norm-budget parameters while using different actor surrogates. Additional residual-interface experiments are described in Appendix H.

B.6 NUMERICAL VERIFICATION

We verified the finite-support evaluator against direct iid enumeration at $G = 4$ and $K = 1, \ldots, 5$, and checked analytic gradients using finite differences, including simplex-direction derivatives. Geometric clone invariance is tested at $K = 2, 8, 32$. Further checks cover the reward-only exponential tilt, the residual definition and its zero-sum property, horizon sensitivity, and feasible finite solutions at $K = 128$.

Comparisons with a reference optimizer on eight random $G = 4$ instances give coordinate differences below 2×10^{-6} and objective differences below 10^{-8}. The gradient-budget tests check the coefficient formula and its implementation structure.

C EXPERIMENTAL PROTOCOL AND STATISTICS

This appendix specifies the conditions behind the main comparisons. We distinguish complete-method comparisons, within-DeSPO horizon interventions, and diagnostic analyses: they share evaluation infrastructure where applicable, but answer different attribution questions.

C.1 TASKS, MODELS, AND GENERATION BUDGETS

Following DQO's task suite (Chen et al., 2026), we evaluate constrained commonsense generation on CommonGen, instruction following on Dolly, summarization on CNN/DailyMail, and mathematical reasoning on GSM8K. The main open-ended benchmark uses Llama-3.2-1B-Instruct (Meta, 2024), with a targeted Llama-3.2-3B-Instruct scale-up; GSM8K uses Qwen2.5-Math-1.5B (Yang et al., 2024). Throughout, G counts real training responses per prompt, K is DeSPO's fixed-support valuation horizon, and N counts fresh evaluation samples. DQO and SetPO do not acquire a DeSPO horizon merely because their configurations contain a shared-schema horizon field.

Table C1: Open-ended generation settings. Lengths are maximum prompt/response token counts; steps count rollout refreshes, not repeated actor passes. Dolly evaluation excludes 29 overlong prompts.

Task	Length limits	Steps/seed	Eval. prompts	Responses, $G4/G8$
CommonGen	64 / 128	186	1,024	95,232 / 190,464
Dolly	1,024 / 512	279	2,974	142,848 / 285,696
CNN/DailyMail	1,536 / 256	168	1,024	86,016 / 172,032

The training schedule uses three dataset passes, batches of 128 prompts, and actor minibatches of 32 prompts. CommonGen uses 8,000 training prompts; dropping incomplete batches gives 186 rollout refreshes. At $G = 4$, this produces $186 \times 128 \times 4 = 95{,}232$ training responses. Increasing K reuses the discovered responses and embeddings; increasing G changes the generation count. Equal response counts do not imply equal token counts, since learned output lengths can differ.

Table C2: Shared open-ended settings and DeSPO-specific target settings. The target coefficients apply to the DeSPO branch, not automatically to every method using the same configuration schema.

Component	Setting
Reward model	Skywork-Reward-V2-Llama-3.2-1B (Liu et al., 2026)
Structural encoder	all-MiniLM-L6-v2; unit-normalized embeddings (Reimers & Gurevych, 2019)
Independent encoder	BGE-M3 (Chen et al., 2024)
Training sampling	Temperature 1; top-$p = 1$; unrestricted top-k
Frozen main evaluation	$N = 16$; temperature 1; top-$p = 1$; top-$k = -1$; repetition penalty 1
Actor / critic learning rates	10^{-6} / 10^{-5}
Actor optimizer	AdamW; $\beta = (0.9, 0.999)$; weight decay 0.01; constant rate; no warmup
Actor reuse and clipping	One actor pass per rollout batch; global gradient-norm limit 1
DeSPO target	Default $\alpha = 1$, $\tau = 600$; default $G4K8$
Full-target actor scale	$\lambda = 1$; frozen displacement and mean-token scores

All formal rows use final checkpoints. Table C2 lists the open-ended settings specified in Appendix B.

Formal GSM8K DeSPO. The Qwen2.5-Math-1.5B runs use $G = 4$, $K \in \{4, 8\}$, $\alpha = 1$, $\tau = 600$, and $\lambda = 1$, with actor minibatches of $B_p = 32$ prompt groups and final checkpoints at step 174. The critic is disabled. Both horizons use the full-target, listwise-only update; the residual diagnostics have separate protocols in Appendices H.3–H.4.

C.2 Baselines and actor-update interfaces

Reference optimizers. We retain DQO's task-specific quality-only references: PPO for the three open-ended tasks and GRPO for GSM8K (Schulman et al., 2017; Shao et al., 2024). DQO and SetPO provide set-level comparators. Formal DeSPO uses full-target score fitting in both task families. Credit assignment and actor updates therefore remain method-specific even when task, model, generation, reward, schedule, and evaluation settings are shared. Main comparisons evaluate complete methods; fixed-interface K sweeps isolate horizon variation within DeSPO.

Table C3: Signal and interface provenance. Common infrastructure does not imply identical optimization procedures. Formal DeSPO uses full-target fitting; the residual interface is diagnostic only.

Result family	Signal / target	Actor update and provenance
PPO / GRPO	Quality reward or native advantage	Native quality update in the shared infrastructure.
DQO	Realized-group leave-one-out credit	DQO credit integrated into the shared RL stack; no independent valuation horizon.
SetPO	Official released credit computation	Credit core ported into the shared stack; not a full native-recipe reproduction.
DeSPO formal open-ended	Numerical horizon target $\widehat{w}_K$	Full-target score fitting with $\widehat{w}_K - p$; Equation 6.
DeSPO formal GSM8K	Numerical horizon target $\widehat{w}_K$	Full-target score fitting with $\widehat{w}_K - p$; logged GRPO policy loss has zero actor-loss weight; critic disabled.
DeSPO residual diagnostic	Structural residual $\widehat{w}_K - \widehat{w}_Q$	Native GRPO plus score-space residual and gradient budget; not the formal Figure 6 interface.

SetPO implementation. SetPO uses the official released leave-one-out credit computation within our shared implementation, with its recorded $\tau = 1$ and weight 0.1. These values belong to SetPO's credit rule and are distinct from DeSPO's target regularization. The comparison preserves the released credit core but does not claim execution of the complete official training recipe.

Solver and score fitting. The full-target computation uses the exact finite-support expansion, cached principal minors, and CPU float64 SLSQP initialized at p. The numerical settings and input checks are given in Appendix B. Targets are recomputed at rollout refresh and held fixed during actor fitting. The full-target actor branch has no contributing PPO/GRPO policy surrogate, entropy term, or reference-policy KL. In open-ended runs, a separately optimized critic does not enter the actor loss; formal GSM8K disables the critic. The residual diagnostic instead combines native GRPO with the structural residual in Appendix B.4.

C.3 Evaluation estimands and uncertainty

Quality and spread. Mean RM averages reward-model scores across generated responses and prompts. Best@N measures the best reward among N responses, rather than mean response quality. Distinct-2 measures lexical variety, Self-BLEU measures within-set lexical similarity, and LogDet and Vendi (Friedman & Dieng, 2022) summarize embedding geometry. For the normalized Gram matrix L_N, the reported geometric quantities use regularized log-determinant spread and an entropy-based effective rank. Their scales depend on encoder and set size. Thus LogDet is compared at matched N; agreement between MiniLM and BGE is assessed by the direction of a within-encoder contrast, not by equating their absolute scores. All metrics except Self-BLEU are oriented upward.

Inference-budget evaluation. For a pool of M responses with sorted rewards $r_{(1)} \le \cdots \le r_{(M)}$ and c verifier-correct responses, the all-subsets estimands are

$$\widehat{\mathrm{Best@}N} = \sum_{j=N}^{M} \frac{\binom{j-1}{N-1}}{\binom{M}{N}} r_{(j)}, \qquad \widehat{\mathrm{Pass@}N} = 1 - \frac{\binom{M-c}{N}}{\binom{M}{N}}. \tag{34}$$

The latter numerator is zero when $M - c < N$. These evaluation quantities do not set training $K = N$. The CommonGen inference-budget curves come from a separate evaluation and are reported under their own protocol rather than pooled with the frozen $N = 16$ benchmark. Singleton LogDet is omitted from the curves because unit-diagonal geometry makes it constant.

Replication and uncertainty. Formal benchmark results report mean $\pm$ sample SD over three training runs. Sample SD reflects run-to-run variation and is not a confidence interval.

Contrasts and bootstrap intervals. Where aligned per-prompt records are available, we additionally compute intervals from 10,000 paired-prompt and hierarchical seed-to-prompt bootstrap draws. Prompt pairing preserves the evaluation questions shared by the two arms; hierarchical resampling additionally accounts for training-seed variation. We report hierarchical 95% intervals in the appendix tables. With three seeds, even these intervals reflect limited run replication. No multiple-comparison correction is claimed. A positive interval is evidence for the specified contrast, not a universal ranking across tasks or metrics.

Differences are calculated before rounding when the source provides full-precision contrasts. For report-only aggregates, subtraction can differ by one unit in the last displayed decimal. Marginal SDs do not determine paired-difference uncertainty: unprovided interval endpoints are left unreported, never reconstructed from independent-error assumptions.

C.4 Evidence provenance and reproducibility

Benchmark aggregates, inference-budget evaluations, bootstrap contrasts, and diagnostic analyses retain their respective protocols and replication units rather than being pooled across incompatible evaluations. Numerical intervals are reported where the corresponding bootstrap records are available.

Control calibration. The Reward-only and KL-matched K8 controls in Appendix F are calibrated on a 512-prompt train-derived pool with zero task-ID overlap with the formal CommonGen test set and trained from the base actor for three runs. Calibration uses only target-distribution statistics, with no downstream evaluation metrics. The benchmark test set was also used during broader project development, so these evaluations are not a never-seen confirmation set, nor do we claim that every earlier design decision was test-independent.

Formal reasoning, short training diagnostics, gradient batches, and cached-pool partitions have different replication units. Appendix H keeps them separate. Exact reproduction also requires the original model revisions, parser revision, run configurations, and evaluator implementation.

D COMPLETE OPEN-ENDED BENCHMARK RESULTS

The following tables retain quality, lexical, and geometric metrics together. Reporting them together exposes both joint gains and trade-offs beyond the primary metrics. D2 abbreviates Distinct-2, S-BLEU denotes Self-BLEU, and LogDet uses MiniLM unless stated otherwise. Values are mean ± sample SD.

D.1 FULL 1B BENCHMARK

CommonGen shows the clearest increase in semantic spread across complementary metrics. Dolly and CNN/DailyMail exhibit smaller or different shifts, reinforcing the task dependence of the quality–spread operating point. Horizontal separators distinguish rollout budgets. Appendix E.2 reports the additional horizon settings.

Table D1: CommonGen 1B, $N = 16$. Additional $G \times K$ settings and DQO $G16$ appear in Appendix E.2.

Setting	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
PPO $G4$	13.523±.217	14.977±.225	.636±.013	.452±.015	5.075±.047	2.475±.050
DQO $G4$	13.448±.373	14.934±.339	.687±.033	.395±.044	5.681±.417	3.087±.448
SetPO $G4$	13.319±.265	14.805±.231	.686±.043	.378±.051	5.782±.333	3.199±.344
DeSPO $G4K4$	13.955±.468	15.509±.395	.678±.074	.389±.093	6.163±.203	3.605±.254
DeSPO $G4K8$	13.886±.144	15.486±.109	.755±.017	.308±.008	6.890±.220	4.554±.339
DQO $G8$	14.389±.220	15.802±.191	.690±.027	.393±.039	5.938±.050	3.316±.065
SetPO $G8$	14.007±.161	15.439±.163	.671±.006	.404±.013	5.448±.226	2.831±.240
DeSPO $G8K8$	16.223±.498	17.502±.385	.565±.077	.524±.076	7.373±.328	5.309±.559

Table D2: Dolly 1B, $N = 16$. PPO provides a quality-only reference.

Setting	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
PPO $G4$	11.271±.097	13.995±.042	.473±.024	.612±.026	4.270±.102	1.860±.073
DQO $G4$	11.316±.170	14.084±.210	.501±.018	.588±.018	4.314±.026	1.891±.015
SetPO $G4$	7.130±.365	9.878±.449	.399±.021	.416±.040	4.572±.086	2.215±.109
DeSPO $G4K4$	11.751±.239	14.659±.259	.538±.010	.548±.015	4.394±.018	1.950±.018
DeSPO $G4K8$	11.476±.100	14.447±.176	.550±.026	.536±.030	4.431±.091	1.984±.067
DQO $G8$	11.746±.123	14.404±.128	.489±.010	.601±.010	4.279±.026	1.865±.019
SetPO $G8$	10.466±.928	13.228±.799	.470±.008	.543±.073	4.383±.126	1.985±.123
DeSPO $G8K8$	12.326±.459	15.235±.136	.580±.034	.512±.034	4.549±.124	2.066±.098

Table D3: CNN/DailyMail 1B, $N = 16$.

Setting	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
PPO $G4$	14.913±.082	16.950±.070	.464±.005	.632±.007	3.915±.010	1.588±.007
DQO $G4$	14.912±.073	16.930±.136	.470±.003	.628±.002	3.930±.009	1.597±.005
SetPO $G4$	14.941±.095	16.947±.106	.460±.013	.634±.011	3.894±.027	1.575±.016
DeSPO $G4K4$	15.670±.166	17.731±.223	.478±.008	.620±.010	3.899±.022	1.579±.014
DeSPO $G4K8$	15.408±.149	17.403±.087	.493±.004	.605±.005	3.932±.048	1.599±.030
DQO $G8$	15.465±.169	17.398±.153	.465±.003	.635±.004	3.908±.017	1.584±.011
SetPO $G8$	15.607±.168	17.524±.174	.461±.006	.638±.007	3.885±.018	1.569±.011
DeSPO $G8K8$	16.390±.116	18.309±.060	.496±.004	.602±.010	3.938±.010	1.603±.006

D.2 TARGETED 3B SCALE-UP

The 3B study compares DeSPO with DQO across all three open-ended tasks; CommonGen also includes PPO. It tests whether the discovery–valuation behavior survives scale, rather than claiming a second full-baseline benchmark. CommonGen $G4K8$ improves both reward quality and spread over DQO $G8$ with half the real rollouts. Dolly also improves both axes over DQO $G8$, while CNN/DailyMail favors $K4$ over $K8$ within DeSPO.

Table D4: Complete 3B absolute results, $N = 16$. Independent-encoder results appear in Appendix G.

Setting	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
CommonGen — Llama-3.2-3B-Instruct						
PPO $G4$	13.864±.123	15.212±.135	.653±.033	.440±.048	5.545±.255	2.933±.249
DQO $G4$	13.949±.212	15.283±.223	.628±.034	.467±.040	5.623±.221	3.010±.218
DQO $G8$	14.613±.160	15.936±.132	.634±.024	.462±.032	5.882±.094	3.281±.113
DeSPO $G4K4$	15.653±1.120	16.995±.956	.524±.141	.569±.154	5.447±.304	2.821±.313
DeSPO $G4K8$	15.140±.129	16.528±.091	.706±.023	.358±.028	6.832±.170	4.441±.247
Dolly — Llama-3.2-3B-Instruct						
DQO $G4$	12.886±.063	15.093±.028	.421±.012	.679±.015	4.035±.029	1.696±.016
DQO $G8$	13.499±.017	15.634±.052	.418±.015	.679±.018	4.016±.030	1.681±.021
DeSPO $G4K4$	13.644±.120	15.863±.171	.452±.015	.649±.022	4.063±.062	1.714±.043
DeSPO $G4K8$	13.642±.047	16.014±.059	.498±.019	.601±.021	4.188±.049	1.798±.034
CNN/DM — Llama-3.2-3B-Instruct						
DQO $G4$	16.813±.078	18.433±.066	.381±.003	.745±.003	3.705±.005	1.463±.003
DQO $G8$	17.154±.138	18.710±.153	.379±.008	.745±.006	3.701±.012	1.461±.007
DeSPO $G4K4$	17.268±.259	18.882±.224	.424±.009	.693±.013	3.748±.022	1.489±.013
DeSPO $G4K8$	16.963±.042	18.548±.095	.410±.013	.713±.018	3.736±.021	1.481±.013

Dolly's fixed-G $K4 \to K8$ change leaves Mean RM nearly unchanged while increasing Best@16, lexical variety, and both encoder-based spread measures. Relative to DQO $G8$, $G4K8$ improves all five metrics with positive hierarchical intervals in Table D5. These are end-to-end method contrasts, distinct from the within-DeSPO horizon effect.

Table D5: Dolly 3B: DeSPO $G4K8$ minus DQO $G8$. Contrasts and 10,000-draw hierarchical 95% intervals, calculated before rounding.

Metric	Δ (DeSPO $G4K8$−DQO $G8$)	Hierarchical 95% interval
Mean RM	+.143	[.091, .198]
Best@16	+.381	[.262, .468]
Distinct-2	+.079	[.051, .107]
MiniLM LogDet	+.172	[.131, .234]
BGE LogDet	+.115	[.093, .137]

D.3 TRAINING-SEED VARIATION

CommonGen 3B horizon variability. The large reward SD for $K4$ is retained. The three runs use identical frozen protocols, the same 1,024 prompt IDs, and 16 responses per prompt, with complete metric records. Their reward means yield 15.653 ± 1.120, so the precise $K8 - K4$ reward difference is seed-sensitive even though the spread direction is clear. Table D6 reports the run-level values.

Table D6: CommonGen 3B DeSPO $G4K4$: run-level metrics under the same frozen evaluation.

Run	Mean RM	Best@16	D2	S-BLEU	LogDet	Vendi
1	16.946	18.082	.362	.747	5.172	2.532
2	14.982	16.282	.621	.473	5.395	2.779
3	15.032	16.622	.588	.488	5.773	3.153

E DISCOVERY, HORIZON, SAMPLING, AND COST

This appendix separates direct horizon effects from external method comparisons and from additional discovery. The numerical contrasts complement Figure 3; they do not attribute all DeSPO–DQO differences to K.

E.1 DIRECT HORIZON CONTRASTS AND THEIR UNCERTAINTY

Table E1 summarizes $K8 - K4$ at fixed $G = 4$ and fixed DeSPO fitting interface. The quality effect is not uniformly positive. CommonGen has the largest spread shift, Dolly 3B adds spread at nearly unchanged Mean RM, and CNN/DailyMail 3B provides a negative boundary. This is evidence for a controllable operating point rather than a larger-is-better horizon rule.

Table E1: Direct $K8 - K4$ differences at fixed $G = 4$. Intervals are source-matched hierarchical 95% bootstrap intervals. A dash denotes an unavailable interval estimate; all displayed means are from completed runs.

Environment	Δ RM	95% interval	Δ LogDet	95% interval
CommonGen 1B	$-.069$	$[-.510, .454]$	$+.727$	$[.430, .900]$
Dolly 1B	$-.275$	–	$+.037$	–
CNN/DM 1B	$-.262$	–	$+.033$	–
CommonGen 3B	$-.514$	$[-1.678, .114]$	$+1.385$	$[.907, 1.644]$
Dolly 3B	$-.002$	–	$+.125$	–
CNN/DM 3B	$-.305$	–	$-.012$	–

The contrast tables below report mean differences and hierarchical intervals. Variation across trained policies can exceed paired-prompt variation within a checkpoint, so prompt-only uncertainty can understate run-to-run variability.

Table E2: Raw-output horizon and discovery contrasts. Intervals use 10,000 hierarchical bootstrap draws.

Metric	Δ	Hierarchical 95% interval
CommonGen 1B K8 minus CommonGen 1B K4		
RM	$-.069$	$[-.510, .454]$
Best@16	$-.023$	$[-.279, .422]$
D2	$+.076$	$[.022, .135]$
S-BLEU	$-.080$	$[-.170, -3.86 \times 10^{-4}]$
LogDet	$+.727$	$[.430, .900]$
Vendi	$+.949$	$[.549, 1.170]$
CommonGen 1B G8K8 minus CommonGen 1B G8K4		
RM	$+.336$	$[.028, .606]$
Best@16	$+.431$	$[.144, .594]$
D2	$+.077$	$[.001, .154]$
S-BLEU	$-.102$	$[-.192, -.018]$
LogDet	$+1.221$	$[.922, 1.415]$
Vendi	$+1.776$	$[1.208, 2.105]$
CommonGen 1B G8K16 minus CommonGen 1B G8K8		
RM	-2.161	$[-2.678, -1.726]$
Best@16	-1.528	$[-1.941, -1.264]$
D2	$+.336$	$[.284, .423]$
S-BLEU	$-.422$	$[-.504, -.368]$
LogDet	$+.819$	$[.460, 1.283]$
Vendi	$+1.571$	$[.902, 2.413]$

Table E2 continued

Metric	Δ	Hierarchical 95% interval
CommonGen 1B G8K8 minus CommonGen 1B K8		
RM	+2.337	[2.029, 2.901]
Best@16	+2.016	[1.759, 2.444]
D2	−.190	[−.294, −.127]
S-BLEU	+.216	[.165, .313]
LogDet	+.484	[−.133, .822]
Vendi	+.755	[−.259, 1.315]
CommonGen 3B K8 minus CommonGen 3B K4		
RM	−.514	[−1.678, .114]
Best@16	−.467	[−1.448, .159]
D2	+.181	[.080, .323]
S-BLEU	−.211	[−.366, −.107]
LogDet	+1.385	[.907, 1.644]
Vendi	+1.620	[1.080, 1.928]
BGE LogDet	+.816	[.630, .924]
BGE Vendi	+.708	[.499, .828]
Dolly-G8K16 minus Dolly-G8K8		
RM	−.461	[−1.127, .349]
Best@16	−.324	[−.725, −.092]
D2	+.021	[−.018, .056]
S-BLEU	−.025	[−.062, .012]
LogDet	+.197	[.059, .329]
Vendi	+.157	[.043, .257]
BGE LogDet	+.143	[.033, .243]
BGE Vendi	+.099	[.021, .165]
CNN-G8K16 minus CNN-G8K8		
RM	+.013	[−.103, .181]
Best@16	−.001	[−.108, .094]
D2	+.013	[.006, .021]
S-BLEU	−.015	[−.027, −.009]
LogDet	+.052	[.008, .079]
Vendi	+.032	[.004, .048]
BGE LogDet	+.035	[−.001, .057]
BGE Vendi	+.020	[−.001, .033]

E.2 Complete discovery–valuation grid

At a fixed discovery budget, increasing K changes the valuation applied to the support. At a fixed $K = 8$, reducing discovery to $G = 2$ retains substantial spread but lowers Mean RM. These interventions expose different roles: a richer discovered support can contain better responses, whereas valuation controls allocation over the support already found.

Table E3: CommonGen 1B: full $G4/G8 \times K4/K8/K16$ grid, lower-discovery control, and DQO $G16$ reference. Mean ± sample SD at $N = 16$.

Setting	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
DeSPO $G4K4$	13.955±.468	15.509±.395	.678±.074	.389±.093	6.163±.203	3.605±.254
DeSPO $G4K8$	13.886±.144	15.486±.109	.755±.017	.308±.008	6.890±.220	4.554±.339
DeSPO $G4K16$	12.595±.471	14.685±.421	.929±.018	.061±.031	7.692±.184	5.882±.341
DeSPO $G8K4$	15.888±.752	17.071±.634	.488±.107	.626±.111	6.152±.097	3.533±.111
DeSPO $G8K8$	16.223±.498	17.502±.385	.565±.077	.524±.076	7.373±.328	5.309±.559
DeSPO $G8K16$	14.062±.063	15.974±.016	.901±.010	.102±.016	8.192±.125	6.881±.272
DeSPO $G2K8$	11.949±.162	13.910±.109	.846±.027	.148±.037	6.826±.609	4.540±.867
DQO $G16$	14.871±.056	16.119±.067	.609±.066	.494±.083	5.709±.190	3.075±.200

The larger-discovery cross-task runs extend this control beyond CommonGen. Dolly's $K8 \to K16$ increment increases spread with a reward cost. On CNN/DailyMail, it raises several spread measures at nearly unchanged Mean RM. These outcomes depend on both task and available support; they do not prescribe a universal K/G ratio.

Table E4: Dolly and CNN/DailyMail 1B at $G = 8$: $K8/K16$ comparisons, $N = 16$. BGE results appear in Appendix G.

Setting	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
Dolly — 1B, $G = 8$						
DeSPO $G8K8$	12.326±.459	15.235±.136	.580±.034	.512±.034	4.549±.124	2.066±.098
DeSPO $G8K16$	11.866±.342	14.911±.219	.601±.006	.487±.009	4.746±.026	2.223±.024
CNN/DM — 1B, $G = 8$						
DeSPO $G8K8$	16.390±.116	18.309±.060	.496±.004	.602±.010	3.938±.010	1.603±.006
DeSPO $G8K16$	16.403±.061	18.308±.071	.509±.005	.586±.005	3.990±.032	1.634±.019

External comparisons. The following source-aligned contrasts compare complete methods. In CommonGen 1B, $G4K8$ trades some quality for spread relative to DQO $G8$. At 3B, the same comparison improves both axes. These observations show quality–spread trade-offs under different generation budgets; they compare complete methods rather than isolating the target-fitting interface.

Table E5: Raw-output external-method contrasts. Intervals use 10,000 hierarchical bootstrap draws.

Metric	Δ	Hierarchical 95% interval
CommonGen 1B K8 minus CommonGen 1B DQO4		
RM	+.439	[−.084, .775]
Best@16	+.551	[.077, .852]
D2	+.068	[.047, .083]
S-BLEU	−.086	[−.118, −.047]
LogDet	+1.208	[1.031, 1.485]
Vendi	+1.467	[1.299, 1.661]
CommonGen 1B K8 minus CommonGen 1B DQO8		
RM	−.503	[−.575, −.404]
Best@16	−.316	[−.387, −.218]
D2	+.065	[.032, .096]
S-BLEU	−.084	[−.124, −.052]
LogDet	+.952	[.813, 1.187]
Vendi	+1.238	[1.016, 1.611]
CommonGen 3B K8 minus CommonGen 3B DQO8		
RM	+.526	[.359, .726]
Best@16	+.592	[.432, .748]
D2	+.071	[.033, .123]
S-BLEU	−.104	[−.170, −.057]
LogDet	+.949	[.880, 1.050]
Vendi	+1.160	[1.046, 1.346]
BGE LogDet	+.280	[−.096, .649]
BGE Vendi	+.253	[−.106, .598]

E.3 Coefficient and decoding sensitivity

The coefficient sweep retrains DQO $G4/G8$ and DeSPO $G4K8$ on the shared $\alpha \in \{0.5, 1, 2\}$ grid. The decoding sweep evaluates trained policies at $T \in \{0.3, 0.5, 0.7, 1.0, 1.2, 1.3\}$. Both move the quality–spread operating point, but along different observed trajectories. The tested DeSPO settings do not uniformly dominate DQO $G8$ at every quality level; temperature also does not simply reproduce every DeSPO operating point.

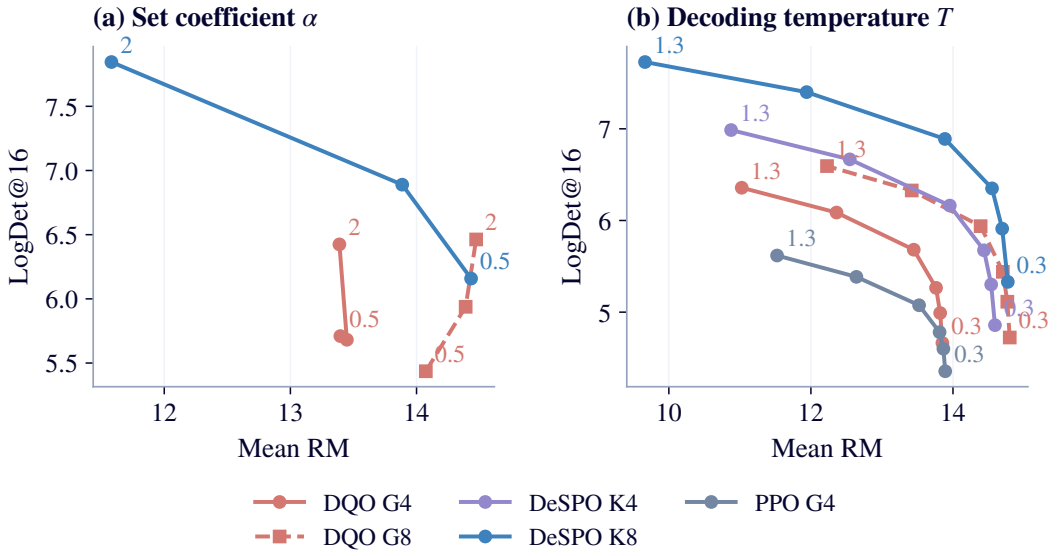

Figure E1: CommonGen sensitivity trajectories. Lines connect tested means; endpoint labels give α or T. They do not imply evaluation of intervening settings. Exact means and sample SD are given in Tables E6 and E7.

These finite sweeps complement the analytic non-equivalence in Appendix A.3; they do not prove that no coefficient or decoding choice could match a particular horizon effect. In particular, the coefficient experiment is not a fully crossed $K \times \alpha$ design.

Table E6: CommonGen coefficient sweep: mean ± sample SD at $N = 16$.

α	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
DQO G4						
.5	13.396±.243	14.905±.228	.696±.006	.370±.007	5.710±.207	3.094±.224
1.0	13.448±.373	14.934±.339	.687±.033	.395±.044	5.681±.417	3.087±.448
2.0	13.389±.263	14.950±.199	.707±.024	.359±.033	6.424±.337	3.936±.449
DQO G8						
.5	14.074±.477	15.506±.430	.667±.025	.412±.032	5.436±.217	2.818±.224
1.0	14.389±.220	15.802±.191	.690±.027	.393±.039	5.938±.050	3.316±.065
2.0	14.473±.150	15.873±.135	.647±.028	.439±.030	6.463±.068	3.958±.097
DeSPO G4K8						
.5	14.432±.463	15.861±.439	.631±.012	.449±.016	6.158±.225	3.592±.276
1.0	13.886±.144	15.486±.109	.755±.017	.308±.008	6.890±.220	4.554±.339
2.0	11.581±.351	13.759±.285	.950±.006	.030±.004	7.845±.057	6.162±.110

Table E7: CommonGen decoding-temperature sweep: mean ± sample SD at $N = 16$. The training setting is fixed within each block.

T	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
DQO G4						
.3	13.851±.379	15.045±.375	.488±.031	.648±.033	4.664±.195	2.116±.163
.5	13.819±.372	15.080±.346	.566±.032	.554±.037	4.990±.255	2.395±.238
.7	13.762±.364	15.063±.341	.618±.034	.487±.041	5.265±.319	2.654±.323
1.0	13.448±.373	14.934±.339	.687±.033	.395±.044	5.681±.417	3.087±.448
1.2	12.361±.406	14.617±.329	.737±.030	.327±.043	6.087±.504	3.558±.602
1.3	11.026±.506	14.273±.354	.765±.029	.289±.043	6.357±.537	3.903±.681
DQO G8						
.3	14.799±.208	15.954±.182	.494±.026	.641±.030	4.723±.033	2.151±.032
.5	14.765±.208	15.981±.173	.571±.027	.548±.033	5.114±.064	2.482±.064
.7	14.702±.211	15.961±.177	.622±.027	.482±.036	5.440±.071	2.786±.079
1.0	14.389±.220	15.802±.191	.690±.027	.393±.039	5.938±.050	3.316±.065
1.2	13.419±.310	15.483±.211	.739±.026	.326±.041	6.327±.084	3.785±.118
1.3	12.227±.389	15.134±.238	.766±.025	.291±.041	6.594±.088	4.138±.133
DeSPO K4						
.3	14.591±.391	15.772±.375	.479±.062	.644±.062	4.858±.111	2.276±.091
.5	14.538±.400	15.801±.372	.550±.069	.556±.074	5.301±.081	2.673±.086
.7	14.436±.405	15.757±.389	.604±.073	.486±.084	5.675±.091	3.046±.113
1.0	13.955±.468	15.509±.395	.678±.074	.389±.093	6.163±.203	3.605±.254
1.2	12.547±.678	15.046±.466	.735±.074	.316±.097	6.667±.240	4.261±.346
1.3	10.876±.984	14.477±.564	.767±.072	.276±.098	6.987±.284	4.737±.439
DeSPO K8						
.3	14.772±.128	15.965±.124	.548±.014	.575±.014	5.330±.327	2.707±.312
.5	14.693±.121	15.928±.106	.620±.014	.486±.009	5.911±.342	3.303±.378
.7	14.550±.121	15.854±.090	.676±.014	.412±.007	6.350±.321	3.823±.397
1.0	13.886±.144	15.486±.109	.755±.017	.308±.008	6.890±.220	4.554±.339
1.2	11.940±.270	14.755±.157	.815±.010	.230±.004	7.400±.214	5.376±.375
1.3	9.665±.450	13.883±.211	.848±.007	.189±.005	7.729±.192	5.974±.366
PPO G4						
.3	13.891±.211	15.040±.206	.436±.009	.698±.010	4.355±.037	1.876±.023
.5	13.865±.209	15.090±.198	.514±.011	.607±.013	4.600±.046	2.065±.028
.7	13.812±.195	15.090±.205	.565±.012	.543±.014	4.781±.043	2.214±.031
1.0	13.523±.217	14.977±.225	.636±.013	.452±.015	5.075±.047	2.475±.050
1.2	12.640±.302	14.712±.230	.682±.016	.392±.013	5.385±.105	2.772±.139
1.3	11.524±.404	14.430±.252	.709±.017	.358±.014	5.618±.132	3.014±.184

E.4 Training horizon across fresh inference budgets

The complete curve evaluation includes $K = 4, 8, 16$, extending the main-text $K4/K8$ comparison. At matched N, longer horizons favor geometric spread, while $K16$ has a clearer quality cost. Thus persistence under fresh sampling does not imply that the largest training horizon is preferable.

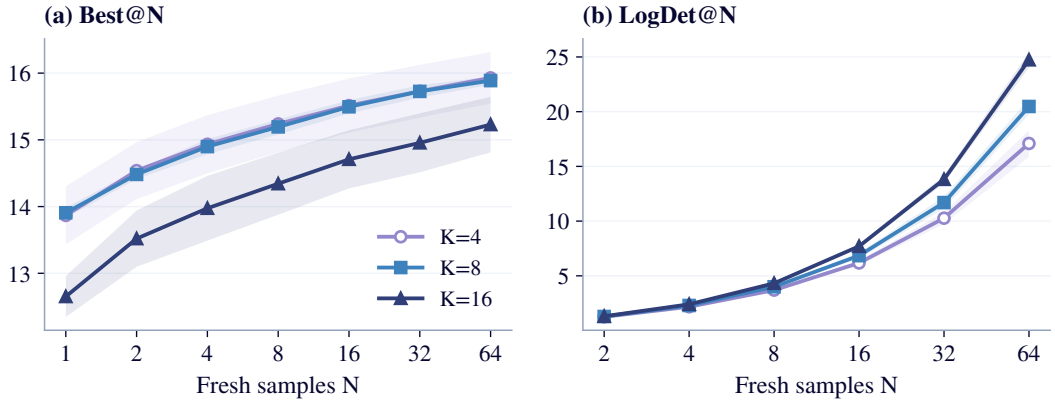

Figure E2: CommonGen 1B DeSPO $G4$: all three training horizons under fresh sampling. Lines show means and ribbons show sample SD. LogDet is compared only within the same N; the evaluation is separate from the main frozen $N = 16$ benchmark.

These curves sample the learned policy afresh. Their separation therefore extends beyond the fixed training pools used to construct individual targets. It remains a comparison of policies trained with different horizons, not an evaluation-time reuse of the training candidates.

E.5 Recorded training telemetry

We report recorded generation and training-step quantities for the CommonGen studies.

1B study. Table E8 summarizes the 1B training records.

Table E8: CommonGen 1B recorded training telemetry. Values are three-run means; recorded step time is the accumulated per-step training-log time, with sample SD. Memory distinguishes allocated and reserved GiB.

Arm	Real responses	Recorded response tokens	Recorded step time (s)	Alloc. GiB	Res. GiB
DQO G4	95,232	8,503,068	3457.7±64.7	36.87	46.20
DQO G8	190,464	16,871,152	6057.8±227.6	38.56	47.23
DeSPO G4K8	95,232	8,591,126	3581.9±62.0	36.86	46.20
DeSPO G2K8	47,616	4,038,374	2183.5±39.3	36.85	45.90
Occupancy G4K8	95,232	8,375,556	3520.3±96.5	36.88	46.20

Relative to DQO $G8$, DeSPO $G4K8$ uses 50% fewer real training responses, with lower recorded response-token totals and step time in this setting.

3B study. Table E9 reports the corresponding telemetry for the CommonGen experiments with a 3B actor.

Table E9: CommonGen 3B recorded training telemetry. Real-response counts follow the completed rollout schedule; response-token totals and step time are aggregated from the corresponding per-step training logs. Values are means ± sample SD over three training runs.

Arm	Real responses	Recorded response tokens (M)	Recorded step time (h)
PPO G4	95,232	10.410 ± 1.484	$2.382 \pm .158$
DQO G4	95,232	$11.348 \pm .826$	$2.476 \pm .113$
DQO G8	190,464	21.388 ± 4.709	4.300 ± 1.392
DeSPO G4K4	95,232	$9.187 \pm .368$	$1.959 \pm .040$
DeSPO G4K8	95,232	9.992 ± 1.600	$2.391 \pm .367$

The 3B study shows the same generation-budget distinction emphasized in the main text. DQO $G8$ uses 190,464 real training responses per run, whereas DeSPO $G4K4$ and $G4K8$ each use 95,232. The two $G4$ DeSPO settings also have lower recorded response-token totals and step time. Figure 1 reports the rounded three-run means for DQO $G8$ and these two variants. We treat the logged token and timing quantities as recorded training telemetry rather than as a separate end-to-end wall-clock benchmark.

F Allocation controls and alternative utilities

The controls test whether fitting alone, allocation-movement magnitude, or a particular utility form is sufficient to explain the observed behavior. They complement the fixed-interface K sweeps; they are not an exhaustive comparison of target-policy optimizers.

F.1 Control definitions and calibration

All primary controls use CommonGen 1B with $G = 4$, the same rewards and fitting machinery, and $K = 8$ when a set horizon applies. Their names below match the main text.

Table F1: Interventions used for attribution. Calibrating allocation KL does not match the complete neural parameter update.

Variant	Change relative to Full DeSPO
Reward-only	Set $\alpha = 0$. Retain the reward, prior, and full-target score-fitting rule; remove set valuation. Match median target KL to the same $K4$ reference.
KL-matched K8	Match median target $\mathrm{KL}(\widehat{w}_K \| p)$ to the $K4$ reference on the train-derived calibration pool. Retain the set utility and fitting rule.
No-geometry	Use $L = I$ in $\log H_K$. Remove off-diagonal semantic similarity while retaining K-dependent interactions across candidate indices.
Occupancy	Replace the determinant value with $\sum_i [1 - (1 - w_i)^K]$, the expected number of occupied indices. This utility is separable and does not use a semantic kernel.
Lower discovery	Use $G = 2, K = 8$ with the full utility; isolate the effect of reducing the available support.

For $L = I$, the finite-support expansion remains a polynomial in products of distinct allocation coordinates. No-geometry therefore removes semantic similarity, not all cross-index interactions. Occupancy instead sums coordinate-wise terms, with candidate identities retained even when their texts or embeddings are similar. These interventions distinguish geometric form from horizon-aware repeated-draw valuation.

Target-KL calibration. We calibrate τ on 512 CommonGen training prompts with zero task-ID overlap with the 1,024-prompt test set. DeSPO $G4K4$ with $\tau = 600$ provides the reference median $\mathrm{KL}(\widehat{w} \| p) = 8.2111 \times 10^{-6}$. Matching this median gives $\tau = 628.07$ for Reward-only

and $\tau = 553.40$ for KL-matched K8. Relative matching errors are below 1.1×10^{-6}, with no solver failures in the calibration trace. Both controls are trained from the base actor for three runs, each completing 186 updates, and evaluated at $N = 16$ with MiniLM and BGE-M3. This matches target-distribution KL on the calibration pool, not per-step actor KL or parameter-gradient norms.

F.2 Complete end-to-end outcomes

Table F2: CommonGen 1B controls: mean ± sample SD, $N = 16$. Reward-only has no active set horizon. The lower-discovery row is separate. Reward-only and KL-matched K8 use the train-derived calibration protocol in Section F.1.

Variant	Mean RM ↑	Best@16 ↑	D2 ↑	S-BLEU ↓	LogDet ↑	Vendi ↑
Full DeSPO	13.886±.144	15.486±.109	.755±.017	.308±.008	6.890±.220	4.554±.339
Reward-only	14.022±.402	15.395±.279	.582±.082	.515±.098	4.703±.182	2.142±.138
KL-matched K8	14.469±.167	16.024±.146	.738±.069	.325±.084	6.642±.286	4.204±.408
No-geometry	13.037±.456	14.981±.349	.870±.020	.158±.027	6.954±.063	4.625±.100
Occupancy	13.680±.281	15.423±.191	.857±.049	.187±.073	6.249±.746	3.740±.963
DeSPO $G2K8$	11.949±.162	13.910±.109	.846±.027	.148±.037	6.826±.609	4.540±.867

KL-matched K8 exceeds Reward-only in MiniLM LogDet and Best@16, with hierarchical intervals above zero; its Mean RM interval crosses zero (Table F3). Relative to $K4$, it has positive Mean RM, Best@16, and MiniLM LogDet contrasts. The spread conclusions persist under BGE-M3: the LogDet differences are $+.894$ over Reward-only (95% interval [.627, 1.193]) and $+.251$ over $K4$ ([.016, .667]). All four quality/spread contrasts against $K4$ have positive differences in each of the three runs. These descriptive intervals use 10,000 hierarchical draws over paired runs and prompts, without multiple-comparison adjustment.

Calibration moves the operating point: relative to fixed-τ $K8$, KL-matched K8 increases Mean RM by .583 and reduces MiniLM LogDet by .247. No-geometry and Occupancy retain substantial spread at different reward levels. Full DeSPO exceeds No-geometry in Mean RM by .849 (95% interval [.537, 1.352]); its LogDet difference is $-.064$ ([$-.293$, .156]). Thus geometry can improve the reward–spread balance without being required for high spread; the latter interval does not establish spread equivalence.

Table F3: Control contrasts. The first two blocks use train-derived target-KL calibration. Intervals use 10,000 hierarchical bootstrap draws.

Metric	Δ	Hierarchical 95% interval
KL-matched K8 minus Reward-only		
RM	$+.448$	$[-.141, .901]$
Best@16	$+.629$	$[.246, .850]$
D2	$+.157$	$[-.006, .252]$
S-BLEU	$-.190$	$[-.311, -4.94 \times 10^{-4}]$
LogDet	$+1.940$	$[1.487, 2.356]$
Vendi	$+2.062$	$[1.506, 2.523]$
KL-matched K8 minus DeSPO $G4K4$		
RM	$+.514$	$[.322, .858]$
Best@16	$+.515$	$[.160, .847]$
D2	$+.060$	$[-.022, .193]$
S-BLEU	$-.064$	$[-.229, .034]$
LogDet	$+.479$	$[.041, .821]$
Vendi	$+.599$	$[.022, 1.036]$

Table F3 continued

Metric	Δ	Hierarchical 95% interval
CommonGen 1B K8 minus CommonGen 1B No-geometry		
RM	+.849	[.537, 1.352]
Best@16	+.505	[.258, .905]
D2	−.115	[−.140, −.076]
S-BLEU	+.150	[.111, .177]
LogDet	−.064	[−.293, .156]
Vendi	−.071	[−.431, .265]
CommonGen 1B K8 minus CommonGen 1B Occupancy		
RM	+.206	[−.080, .633]
Best@16	+.062	[−.113, .370]
D2	−.102	[−.140, −.030]
S-BLEU	+.122	[.028, .171]
LogDet	+.640	[−.346, 1.546]
Vendi	+.814	[−.525, 2.013]
CommonGen 1B K8 minus CommonGen 1B G2K8		
RM	+1.938	[1.881, 1.986]
Best@16	+1.576	[1.529, 1.618]
D2	−.091	[−.138, −.066]
S-BLEU	+.161	[.136, .207]
LogDet	+.064	[−.582, .503]
Vendi	+.015	[−.940, .580]

F.3 Alternative objectives and local alignment

A second-order approximation (P2) truncates the determinant expansion above order two. Expected log-determinant (EL) places the logarithm inside the expectation, replacing $\log \mathbb{E}[\det]$ with $\mathbb{E}[\log \det]$. Occupancy uses the separable expected-index count defined above. These local target diagnostics are distinct from end-to-end retraining of every alternative.

Table F4: Reported local target-direction cosines with the full target under strong KL. These values characterize local alignment, not performance equivalence or independent training-seed replication.

Alternative utility	Direction cosine
Second-order truncation (P2)	0.999377
Expected log-determinant (EL)	0.999953
Expected occupancy (SEP)	0.999419

These local measurements use 512 calibration groups with reference median allocation KL about 6.9875×10^{-6}. Recorded utility multipliers are 2.300032 for P2, .992110 for expected log-determinant, and .541139 for Occupancy. They describe the calibrated local diagnostic, not a replacement for the end-to-end training results in Table F2. In these calibrated strong-KL settings, the measured reallocations are nearly collinear; KL strength is not isolated from coefficient calibration or the shared reward component. This helps explain the similar local behavior of simpler alternatives without contradicting the general non-equivalence of K and scalar rescaling. The interaction-order coefficients in main-text Figure 4(c) visualize analytic weights only; determinant minors, allocation products, and the KL-constrained optimization also determine actual target and gradient contributions.

F.4 What the controls identify

The train-only calibrated controls strengthen the attribution to horizon-aware fixed-support valuation. Reward-only shows that fitting without the set term does not recover the spread shift. After matching median target KL to $K4$ on a disjoint calibration pool, $K8$ retains positive spread contrasts under both MiniLM and BGE-M3. This weakens an explanation based solely on target-reallocation magnitude. No-geometry and Occupancy show that high spread does not require the exact determinant geometry, although geometry can improve the quality–spread balance. These conclusions concern the tested

interventions; they neither attribute every DeSPO–DQO difference to K nor exhaust possible target-policy optimizers.

In particular, $K = G$ still averages repeated selections from a fixed indexed support. It is not a no-expectation ablation. The available evidence does not include an otherwise identical fitting procedure in which H_K is replaced by a realized-group utility. Thus the supported contribution is useful horizon-indexed fixed-support valuation; the exact expected determinant is an analyzable realization, not an empirically proven unique choice.

G INDEPENDENT SEMANTIC-ENCODER EVALUATION

An independent encoder tests whether the observed semantic-spread changes depend on the encoder used during training. We compare MiniLM and BGE-M3 using within-encoder contrasts.

Across the three 3B tasks, BGE-M3 preserves the sign of the $K8 - K4$ LogDet change: positive for CommonGen and Dolly, negative for CNN/DailyMail. The agreement includes the boundary case, not only the favorable comparisons. Absolute LogDet values from different encoders are not calibrated to a common quality scale.

Table G1: 3B encoder results: mean ± sample SD. A dash denotes an unreported BGE Vendi value, not a missing training run.

Setting	MiniLM LogDet	BGE LogDet	BGE Vendi
CommonGen 3B			
PPO $G4$	$5.545 \pm .255$	$5.063 \pm .237$	$2.435 \pm .198$
DQO $G4$	$5.623 \pm .221$	$5.131 \pm .248$	$2.495 \pm .218$
DQO $G8$	$5.882 \pm .094$	$5.223 \pm .170$	$2.573 \pm .154$
DeSPO $G4K4$	$5.447 \pm .304$	$4.687 \pm .192$	$2.118 \pm .152$
DeSPO $G4K8$	$6.832 \pm .170$	$5.503 \pm .356$	$2.826 \pm .335$
Dolly 3B			
DeSPO $G4K4$	$4.063 \pm .062$	$3.712 \pm .029$	–
DeSPO $G4K8$	$4.188 \pm .049$	$3.799 \pm .020$	–
DQO $G4$	$4.035 \pm .029$	$3.724 \pm .015$	–
DQO $G8$	$4.016 \pm .030$	$3.685 \pm .022$	–
CNN/DM 3B			
DQO $G4$	$3.705 \pm .005$	$3.443 \pm .004$	–
DQO $G8$	$3.701 \pm .012$	$3.441 \pm .010$	–
DeSPO $G4K4$	$3.748 \pm .022$	$3.498 \pm .006$	–
DeSPO $G4K8$	$3.736 \pm .021$	$3.483 \pm .001$	–

Table G2: Independent-encoder checks for the 1B $G8$ horizon extension. Mean ± sample SD at $N = 16$.

1B setting	MiniLM LogDet	BGE LogDet	BGE Vendi
Dolly $G8K8$	$4.549 \pm .124$	$4.082 \pm .101$	$1.724 \pm .068$
Dolly $G8K16$	$4.746 \pm .026$	$4.225 \pm .006$	$1.823 \pm .006$
CNN/DM $G8K8$	$3.938 \pm .010$	$3.681 \pm .005$	$1.453 \pm .003$
CNN/DM $G8K16$	$3.990 \pm .032$	$3.716 \pm .036$	$1.473 \pm .020$

The BGE increase for Dolly 3B $G4K8$ relative to DQO $G8$ is also supported by the positive interval in Table D5. For the $G8$ extension, both Dolly and CNN/DailyMail gain BGE spread from $K8$ to $K16$, consistent with their MiniLM directions.

H REASONING RESULTS AND UTILITY-ALIGNMENT DIAGNOSTICS

Formal DeSPO GSM8K results use the full-target interface. Separate residual-interface studies probe local interactions with task quality, while cached-pool regrouping measures correct-candidate availability. These diagnostics have distinct replication units and do not specify the training interface of the formal checkpoints.

H.1 FORMAL CHECKPOINT EVALUATION AND EXACT PASS@N

The formal GSM8K evaluation uses completed Qwen2.5-Math-1.5B checkpoints and the canonical answer parser. Tables H1 and H2 reproduce all seven budgets underlying main-text Figure 6, in probability units.

Table H1: Formal GSM8K Pass@N, $N \leq 8$, under the canonical parser. DeSPO uses full-target checkpoints; values are mean $\pm$ sample SD.

Setting	Pass@1	Pass@2	Pass@4	Pass@8
GRPO G4	.7184±.0051	.7890±.0078	.8386±.0094	.8747±.0104
DQO G4	.7190±.0091	.7979±.0053	.8518±.0026	.8904±.0014
DQO G8	.7492±.0070	.8239±.0053	.8744±.0044	.9091±.0036
DeSPO G4K4	.7195±.0237	.8042±.0236	.8608±.0221	.9006±.0188
DeSPO G4K8	.7071±.0147	.7935±.0184	.8527±.0196	.8951±.0178
SetPO G4	.7075±.0022	.7790±.0015	.8304±.0037	.8683±.0050
SetPO G8	.7456±.0019	.8089±.0028	.8539±.0031	.8872±.0028

Table H2: Formal GSM8K Pass@N, $N \geq 16$, continuing Table H1 under the same protocol and full-target DeSPO checkpoints.

Setting	Pass@16	Pass@32	Pass@64
GRPO G4	.9016±.0109	.9223±.0107	.9391±.0102
DQO G4	.9188±.0008	.9396±.0007	.9550±.0019
DQO G8	.9333±.0025	.9507±.0018	.9639±.0018
DeSPO G4K4	.9289±.0147	.9491±.0106	.9646±.0078
DeSPO G4K8	.9253±.0140	.9470±.0095	.9634±.0043
SetPO G4	.8961±.0052	.9167±.0045	.9323±.0027
SetPO G8	.9125±.0024	.9324±.0024	.9479±.0031

DeSPO $G4K4$ approaches the larger-rollout DQO $G8$ reference as N grows, but $K8$ does not yield an additional observed correctness gain. The reported seed dispersion is substantial, so proximity of the curves does not establish equivalence. SetPO uses the shared-stack credit implementation in Appendix C.2.

The formal DeSPO $K4/K8$ checkpoints use Equation 6: task quality enters $\widehat{w}_K$, while the separately logged GRPO policy branch has zero actor-loss weight. Across all six runs, the logged total actor loss matches the full-target listwise loss within 7×10^{-13}. The residual and gradient-budget studies in Appendices H.3–H.4 are separate diagnostics, not the training interface underlying these Pass@N curves.

H.2 Reward and semantic metrics versus correctness

Table H3 reports reward and semantic metrics for the formal checkpoints.

Table H3: Formal GSM8K reward and semantic metrics at $N = 16$, mean $\pm$ sample SD. DeSPO uses full-target checkpoints; no comparable native-GRPO semantic row is available.

Setting	Mean RM	Best@16	LogDet	Vendi
DQO G4	$12.310 \pm .101$	$14.323 \pm .018$	$3.665 \pm .016$	$1.433 \pm .009$
DQO G8	$12.869 \pm .085$	$14.665 \pm .033$	$3.708 \pm .028$	$1.457 \pm .015$
DeSPO G4K4	$11.835 \pm .066$	$13.784 \pm .103$	$3.682 \pm .094$	$1.444 \pm .053$
DeSPO G4K8	$11.824 \pm .064$	$13.750 \pm .011$	$3.658 \pm .093$	$1.430 \pm .052$
SetPO G4	$12.167 \pm .080$	$13.994 \pm .140$	$3.460 \pm .064$	$1.318 \pm .034$
SetPO G8	$12.809 \pm .096$	$14.427 \pm .134$	$3.450 \pm .014$	$1.313 \pm .009$

Reward-model scores and geometric spread do not substitute for verifier correctness. In particular, a setting's RM or LogDet position does not alone predict its ordering at large Pass@N. This motivates the utility-alignment diagnostics below rather than interpreting semantic novelty as reasoning progress.

H.3 Single-seed, 48-step residual-interface diagnostic

This study uses a different actor interface from the formal full-target checkpoints. Four diagnostic arms share the same Qwen2.5-Math-1.5B initialization, data order, single training seed, $G = 4$, and 48 rollout updates. Evaluation uses 256 prompts, $N = 10$, temperature 1, and a 256-token response limit. The native-quality score-space arms use a gradient budget with $\rho = .25$ and $\lambda_{\max} = 783.876075$; the clipped-residual arm tests a different structural surrogate. Appendix B.4 specifies the current native-quality implementation. These runs are not extra seeds for the formal benchmark.

Table H4: Single-seed 48-step diagnostic. "Correct" is mean response correctness; Pass@10 is prompt-level coverage. LogDet uses MiniLM and BGE is BGE LogDet.

48-step arm	Pass@10	Correct	Mean RM	Best@10	LogDet	BGE
Native GRPO G4	.5664	.0918	7.114	10.914	3.151	2.856
Score-space G4K4	.5742	.1109	7.191	11.091	3.228	2.926
Score-space G4K8	.3750	.0473	6.925	10.735	3.225	2.920
Clipped residual G4K8	.4531	.0672	6.997	10.912	3.182	2.879

The score-space $K4/K8$ arms have nearly equal MiniLM LogDet but markedly different correctness. Controlling branch scale in this short study therefore does not recover a useful $K8$ gain. These residual-interface outcomes remain separate from the formal full-target benchmark.

H.4 Gradient alignment and candidate-space displacement

Parameter-space gradients. We probe a separate native-quality residual interface at initialization and native-training steps 32 and 48. Let $g_Q = \nabla_\theta \mathcal{L}_{\text{native}}$ be the native GRPO quality-loss gradient, and $g_K = \nabla_\theta \mathcal{L}_{S,K}$ the unclipped listwise structural-loss gradient from $d_K = \widehat{w}_K - \widehat{w}_Q$, with $\Delta g = g_8 - g_4$. This structural residual differs from the formal displacement $\widehat{w}_K - p$. Each checkpoint has four paired step-level probes, evaluated using direct FP64 inner products; all 12 records pass the three-vector Gram consistency check.

Table H5: Residual-interface gradient alignment. Mean ± sample SD over four paired step-level records at each checkpoint; dispersion is not training-seed uncertainty. These are not gradients from the formal full-target GSM8K checkpoints.

Alignment	Initialization	Native step 32	Native step 48
$\cos(g_Q, g_4)$	$-.3032 \pm .0523$	$-.3902 \pm .0788$	$-.2102 \pm .1003$
$\cos(g_Q, g_8)$	$-.3190 \pm .0468$	$-.3974 \pm .0772$	$-.2085 \pm .1022$
$\cos(g_4, g_8)$	$.9967 \pm .0011$	$.9977 \pm .0004$	$.9979 \pm .0008$
$\cos(g_8 - g_4, g_Q)$	$-.3305 \pm .0430$	$-.4021 \pm .0756$	$-.2062 \pm .1034$

The $K4/K8$ structural directions are highly aligned ($\cos = 0.9967$–0.9979), and the mean $K8$ norm is 2.1–2.5$\times$ the mean $K4$ norm. Both have moderately negative quality alignment, as does Δg. Here the larger horizon mainly strengthens an existing structural direction rather than rotating it substantially. This local tension motivates examining utility alignment, not a causal decomposition of the formal $K4/K8$ result.

Under a diagnostic strength match $\beta = 0.14$, the combined gradients $g_Q + \beta(\|g_Q\|/\|g_K\|)g_K$ retain cosine 0.9904–0.9911 with g_Q at both horizons. The complete matched update therefore remains quality-dominated; negative structural alignment does not imply its reversal. This probe is distinct from the $\rho = .25$ training budget in Appendix B.5.

The recorded raw-reward and GRPO-relative local target directions have cosine above 0.9999996, weakening that coordinate mismatch as a sufficient explanation.

Candidate-space displacement. For a fixed indexed pool, with identical rewards, prior, kernel, and regularization, define

$$d_K = \widehat{w}_K - \widehat{w}_Q, \qquad \Delta d = d_{2K} - d_K = \widehat{w}_{2K} - \widehat{w}_K. \tag{35}$$

The cross-task diagnostic uses 256 prompts per task, with 256 groups for the open-ended tasks and 512 for GSM8K. Table H6 reports group means of the effective quality support $S_{\text{eff}} = 1/\sum_i \widehat{w}_{Q,i}^2$ and the displacement magnitude $\|\Delta d\|_1 = \sum_i |\Delta d_i|$. These quantities describe allocation concentration and movement; neither measures alignment with correctness or novelty.

Table H6: Fixed-pool allocation concentration and horizon displacement (group means). The 10^5 factor rescales L_1; these are descriptive diagnostics, not independent training-seed estimates.

Task	Groups	S_{eff}	$\|\Delta d\|_1 \times 10^5$
CNN/DM	256	3.978	5.198
CommonGen	256	3.943	8.901
Dolly	256	3.950	8.309
GSM8K	512	3.960	9.385

Effective quality support is close to four under strong KL, indicating a diffuse reward-only allocation. It does not count correct answers or establish their scarcity. The regrouping study below measures verifier-correct candidate availability directly.

H.5 CACHED-POOL REGROUPING AND THE TRAINING DECISION

For 256 prompts with 64 cached responses each, deterministic grouping and five shuffled partitions form $G4/G8$ groups from the same pools. These are six grouping checks, not six training seeds. For a group b, let $c_{bi} \in \{0, 1\}$ denote candidate correctness under the verifier. The multi-correct fraction is

$$M_G = \frac{1}{|\mathcal{B}_G|} \sum_{b \in \mathcal{B}_G} \mathbf{1}\left\{ \sum_{i=1}^{G} c_{bi} \geq 2 \right\}, \tag{36}$$

where $\mathcal{B}_G$ is the collection of groups at size G in a partition. This statistic measures the availability of multiple correct candidates, not whether valuation favors them.

Increasing group size raises the deterministic multi-correct fraction from .783447 to .847168, with the same direction in all five shuffled partitions (Table H7). Thus larger groups contain more alternatives already verified as correct. This descriptive availability result does not establish a correctness benefit from a larger valuation horizon.

Table H7: Multi-correct group fractions under cached-pool regrouping. All partitions reuse the same 256 prompt pools; shuffles test grouping sensitivity, not independent training replications.

Partition	$G = 4$: M_4	$G = 8$: M_8
Deterministic	.7834	.8472
Shuffle 0	.7803	.8447
Shuffle 1	.7747	.8447
Shuffle 2	.7812	.8418
Shuffle 3	.7781	.8428
Shuffle 4	.7786	.8467

No DeSPO $G8K16$ policy was trained after the recorded stop decision. Table H7 measures candidate availability, not a stopping rule or predicted performance. The separate residual-interface diagnostic (Appendix H.4) finds negative alignment between the horizon increment and quality gradient. Together with the formal $K4/K8$ results, this motivates examining utility alignment; the counts alone do not establish its causal role.