OBJECTIVE-AWARE GAUSSIAN SAMPLING FOR ZERO-VARIANCE PROMPTS IN LLM REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Group Relative Policy Optimization (GRPO) has emerged as an efficient reinforcement learning paradigm for fine-tuning large language models, yet its effectiveness can be hindered by zero-variance prompts—cases where all sampled responses yield identical rewards, producing zero group-relative advantages while still incurring generation costs. We introduce Objective-Aware Gaussian Sampling (OAGS), a prompt-level curriculum framework that dynamically reallocates the rollout budget toward prompts in an evolving target pass-rate region. OAGS estimates prompt-specific pass rates from observed outcome rewards, constructs Gaussian candidate proposals centered on the current curriculum target, and adaptively adjusts the curriculum via a progress-controlled pacing mechanism. By concentrating training on informative prompts while preserving controlled exploration of harder and easier regions, OAGS prioritizes non-zero-variance response groups within a fixed candidate budget. Our approach retains the GRPO advantage estimator and clipped surrogate while adapting prompt sampling and post-rollout group selection. Experimental results show improved macro-average performance across six mathematical reasoning benchmarks compared to standard GRPO and the evaluated data-selection baselines, with fewer training rollouts than Dynamic Sampling and GRESO at matched policy-update counts.

1 INTRODUCTION

Reinforcement learning with verifiable rewards has become an effective approach to improving the reasoning capabilities of LLMs (Shao et al., 2024; Guo et al., 2025). Group Relative Policy Optimization (GRPO) supports this approach by estimating advantages from reward comparisons among responses to the same prompt, without a separately learned value model. Under binary outcome rewards, however, all-correct and all-incorrect response groups have identical rewards and therefore zero group-relative advantages, giving rise to the *zero-variance prompt* (ZVP) problem (Yu et al., 2025; Fang et al., 2026; Wan et al., 2026; Le et al., 2026). Such groups incur generation and reward-evaluation costs without providing an advantage-based task-reward signal. Improving training efficiency thus requires allocating rollout computation to prompts that are informative under the evolving policy (Noukhovitch et al., 2025; Qu et al., 2026; Dong et al., 2026; Yang et al., 2026).

This allocation must satisfy two coupled requirements. *(1) Rollout efficiency.* Before generation, the sampler should favor prompts likely to produce mixed-outcome response groups. Filtering after generation can improve the composition of the update batch, but cannot avoid the computation already spent on discarded groups. *(2) Adaptive capability tracking.* Sampling decisions must remain responsive to policy changes, even though fresh outcomes are observed only for evaluated prompts. A previously all-incorrect prompt may become informative as the policy improves, while an initially useful prompt may become saturated. Concentrating exclusively on historically informative prompts can therefore leave other capability estimates stale. These requirements create an exploitation–exploration trade-off: the sampler must obtain useful training groups while collecting the observations needed to revise future allocation decisions.

Existing work improves data efficiency through offline data curation and online sampling. Offline data curation (Li et al., 2025; 2026) selects training subsets using statistics from preliminary train-

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.

ing, which may be outdated as the policy evolves. Online methods filter uninformative response groups (Yu et al., 2025; Yue et al., 2025) or guide rollout allocation using historical utility and adaptive exploration (Yao et al., 2025; Zheng et al., 2025; Tang et al., 2026). Although exploration and revisiting help refresh prompt-level estimates, these updates do not specify how fresh capability feedback should control the focus, pace, and direction of the sampling curriculum.

We propose *Objective-Aware Gaussian Sampling* (OAGS), a feedback-controlled curriculum over prompt-level empirical pass rates. Following an initial evaluation of the training set, OAGS maintains the latest observation for each prompt and constructs Gaussian candidate proposals in this capability space. The Gaussian core concentrates candidates around a target pass-rate region, while tail exploration revisits less frequently evaluated prompts outside it. Crucially, the target region is not fixed: its movement is controlled by the magnitude and directional consistency of differences between stored pass rates and fresh observations. Broad improvement accelerates progression toward lower-pass-rate prompts, mixed evidence leads to slower progression, and sufficiently consistent deterioration causes the curriculum to retreat. OAGS thus uses outcome feedback to control where rollout computation is concentrated, rather than only deciding whether individual prompts should be included or skipped.

OAGS implements this curriculum through an oversampled candidate batch whose size is fixed before each rollout. Every candidate receives a fresh response group, and all observed outcomes refresh the capability tracker and inform subsequent curriculum movement. A fixed-size subset is then retained for optimization, prioritizing groups with non-zero empirical reward variance. This separates *probing* from *training selection*: a candidate excluded from the update can still revise the evidence used for future sampling. The retained groups use the standard GRPO advantage estimator and clipped surrogate objective. Together, feedback-controlled candidate allocation and outcome-based batch selection coordinate the collection of capability information with its use in policy optimization.

2 Background and Related Work

2.1 Group Relative Policy Optimization

Reinforcement learning with verifiable rewards (RLVR) improves LLM reasoning using automatically evaluated task outcomes (Shao et al., 2024; Guo et al., 2025). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) avoids learning a separate value network by estimating advantages through reward comparisons among responses to the same prompt.

Given a prompt $x \sim \mathcal{D}$, the behavior policy $\pi_{\theta_{\text{old}}}$ independently generates G responses $\{o_i\}_{i=1}^{G}$, each receiving an outcome reward R_i. We write $o_i = (o_{i,1}, \ldots, o_{i,|o_i|})$, where j indexes token positions and $o_{i,<j}$ denotes the preceding prefix. Under outcome supervision, every token in a response receives the same group-normalized advantage:

$$A_{i,j} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})}, \qquad j = 1, \ldots, |o_i|. \tag{1}$$

The numerical stabilizer used in the denominator is omitted for readability. Using token-level likelihood ratios, GRPO maximizes a clipped surrogate regularized toward a reference policy:

$$\mathcal{J}_{\text{GRPO}} = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{j=1}^{|o_i|}\min\Big(r_{i,j}(\theta)\,A_{i,j},\ \text{clip}\big(r_{i,j}(\theta), 1-\epsilon, 1+\epsilon\big)A_{i,j}\Big) - \beta\,\mathbb{D}_{\text{KL}}\big(\pi_\theta \big\| \pi_{\text{ref}}\big)\,, \tag{2}$$

where $r_{i,j}(\theta) = \frac{\pi_\theta(o_{i,j}|x, o_{i,<j})}{\pi_{\theta_{\text{old}}}(o_{i,j}|x, o_{i,<j})}$ is the conditional likelihood ratio, ϵ controls clipping, and β weights reference-policy regularization. For compactness, $\mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ denotes the conditional policy KL averaged over the sampled prefixes using the same response and token normalization as the reward surrogate.

2.2 Zero-Variance Prompts

A prompt is referred to as a *zero-variance prompt* (ZVP) for a particular sampled group when all group rewards are identical (Yu et al., 2025; Zheng et al., 2025; Le et al., 2026). Under binary

outcome rewards, this corresponds to an all-incorrect or all-correct response group. The centered rewards then vanish, yielding $A_{i,j} = 0$ at every token under the numerically stabilized version of Equation 1. For a fixed sampled group, let $\mathcal{J}_{\mathrm{GRPO}}(\theta; x)$ denote the reward surrogate obtained from Equation 2 by removing clipping and KL regularization. Holding the sampled responses and their advantages fixed gives:

$$\nabla_\theta \mathcal{J}_{\mathrm{GRPO}}(\theta; x) = \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{j=1}^{|o_i|} r_{i,j}(\theta) A_{i,j} \cdot \nabla_\theta \log \pi_\theta(o_{i,j} \mid x, o_{i,<j}). \tag{3}$$

A ZVP therefore contributes no gradient through the reward surrogate. Clipping does not alter this conclusion, although KL regularization may still produce a gradient. Such groups nevertheless incur generation and reward-evaluation costs and may occupy positions in the optimization batch.

ZVP status depends on both the policy and the sampled response group, rather than being a permanent property of a prompt. A difficult prompt that initially produces only incorrect responses may yield mixed outcomes as the policy improves, while a previously informative prompt may become saturated. Excluding an observed group from the current update should therefore be distinguished from excluding its prompt from future evaluation: the latter also prevents the collection of observations needed to detect changes in its training value.

2.3 Data Efficiency for RLVR

Data-efficient RLVR methods address training-set selection, rollout allocation, and experience reuse. Offline selection identifies informative subsets using model-dependent statistics, including learning trajectories (Li et al., 2025) and measures of learnability and gradient alignment (Li et al., 2026). Online strategies filter uninformative response groups after generation (Yu et al., 2025; Zhang et al., 2025; Fang et al., 2026), guide pre-rollout selection using historical outcomes with adaptive exploration (Zheng et al., 2025), or predict prompt difficulty through concurrently updated value models (Gao et al., 2026). Other sampling strategies adjust how generation effort is distributed across prompts and repeated responses (Yue et al., 2025; Yao et al., 2025). Joint offline–online approaches combine diversity- and difficulty-aware subset construction with training-time sample prioritization (Tang et al., 2026), while replay-based approaches reuse historical trajectories and reevaluate previously difficult prompts (Wan et al., 2026). Within this broader setting, OAGS uses differences between stored empirical pass rates and fresh observations to control the pace and direction of an online Gaussian curriculum. All candidate outcomes inform subsequent sampling, whereas only selected fresh response groups enter standard GRPO optimization.

3 Method

We introduce *Objective-Aware Gaussian Sampling* (OAGS), a feedback-controlled prompt curriculum for GRPO shown in Figure 1. OAGS uses empirical pass rates to construct Gaussian proposals that balance exploitation around a target capability level with exploration outside the preferred region. Fresh outcomes from all candidates update the prompt-level capability estimates, while differences between stored and current observations regulate the curriculum's pace and direction. From each oversampled candidate batch, OAGS retains a fixed-size subset of response groups according to their observed reward variation. The retained groups are optimized using the standard GRPO advantage estimator and clipped surrogate.

3.1 Objective-Aware Capability Modeling

For each prompt x, let $P_{t-1}(x) \in [0, 1]$ denote the latest empirical pass rate available before training step t. These states are initialized through a preliminary evaluation over the training set. We consider binary outcome rewards, with a positive reward indicating a successful response.

At step t, let $\widetilde{\mathcal{B}}_t$ denote the candidate set of M distinct prompts and $\mathcal{B}_t \subseteq \widetilde{\mathcal{B}}_t$ the subset of $B \leq M$ prompts retained for optimization. The rollout policy generates G responses for each candidate, and the complete response groups of the retained prompts form the GRPO update batch. The observed

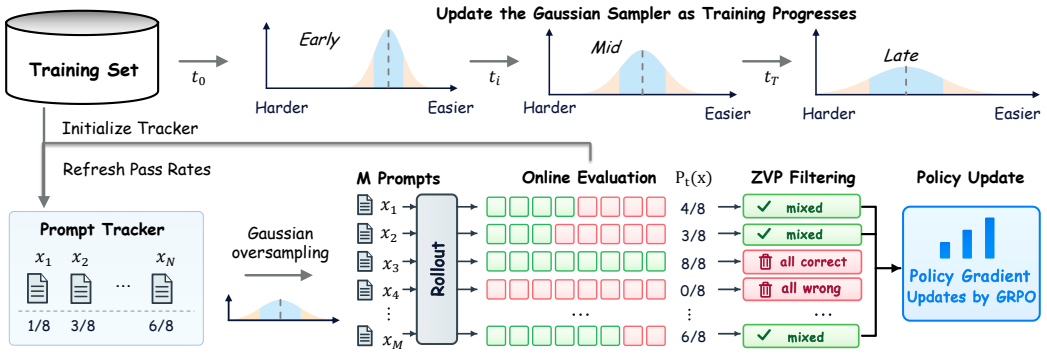


Figure 1: Overview of objective-aware dynamic Gaussian curriculum sampling framework with online evaluation, ZVP filtering, and GRPO policy updates.

outcomes update the capability states as follows:

$$P_t(x) = \begin{cases} \dfrac{1}{G}\displaystyle\sum_{i=1}^{G} \mathbb{I}(R(x, o_i) > 0), & x \in \widetilde{\mathcal{B}}_t, \\ P_{t-1}(x), & x \notin \widetilde{\mathcal{B}}_t. \end{cases} \tag{4}$$

This update incorporates every evaluated candidate, independently of its subsequent inclusion in the policy update.

Under a conditional independent Bernoulli approximation, a response group has zero reward variance when all responses succeed or all fail. Substituting the pre-rollout estimate into these two event probabilities gives:

$$z_t(x) = P_{t-1}(x)^G + (1 - P_{t-1}(x))^G. \tag{5}$$

Its complement estimates the probability of observing both successful and unsuccessful responses, connecting prompt-level capability to the availability of non-degenerate group-relative advantages. Intermediate pass rates yield lower predicted ZVP risk than either capability extreme. Because the underlying estimates are based on finite, potentially stale observations, $z_t(x)$ serves as a sampling criterion rather than a guarantee about the next response group.

3.2 Gaussian Curriculum Proposal

Let $\mathcal{A}_t$ denote the active prompt pool after applying training-usage constraints. During warmup, this pool is additionally restricted to prompts with sufficiently low predicted ZVP risk. The safety restriction is removed after warmup, allowing otherwise quota-eligible prompts to be revisited. The warmup and quota settings are specified in the appendix.

Given the historical capability state $P_{t-1}(x)$, OAGS assigns each active prompt the Gaussian preference:

$$w_t(x) = \frac{1}{\sqrt{2\pi}\sigma_t} \exp\left(-\frac{(P_{t-1}(x) - \mu_t)^2}{2\sigma_t^2}\right), \tag{6}$$

where μ_t is the target pass rate and $\sigma_t > 0$ determines the bandwidth around it. These preferences guide candidate allocation within the core–tail construction below, rather than defining a single categorical distribution over all active prompts.

Exploitation on the Gaussian Core. The Gaussian core contains prompts whose estimated pass rates lie within one bandwidth of the curriculum target:

$$\mathcal{C}_t = \{x \in \mathcal{A}_t : |P_{t-1}(x) - \mu_t| \le \sigma_t\}. \tag{7}$$

Within the core, OAGS groups prompts by their empirical pass rates and allocates candidate positions across these strata according to the Gaussian preferences and available capacities. It then samples uniformly without replacement within each stratum. This allocation operates in capability space, preventing stratum cardinality alone from determining the exploitation budget.

Exploration on the Gaussian Tail. The complementary tail provides observations outside the preferred capability region:

$$\mathcal{T}_t = \{x \in \mathcal{A}_t : |P_{t-1}(x) - \mu_t| > \sigma_t\}. \tag{8}$$

It includes both lower and higher pass-rate prompts. OAGS forms the tail sampling support starting with eligible prompts having the fewest previous online evaluations, then applies Gaussian-weighted sampling without replacement within that support. Tail exploration therefore prioritizes less frequently observed prompts while retaining a preference for those closer to the curriculum target.

Together, these operations induce a candidate-batch proposal distribution q_t that produces $\widetilde{\mathcal{B}}_t$ containing M distinct prompts. A prescribed majority of the candidate budget is allocated to the core and the remainder to the tail. When one partition cannot meet its target allocation, the other supplies the remaining candidates, provided the active pool has sufficient capacity. This allocation governs the candidate batch, not the composition of the final update batch.

From Candidate Proposals to the Update Batch. After evaluating the candidate groups, OAGS summarizes their observed binary-outcome variation by:

$$V_t(x) \coloneqq \frac{1}{G}\sum_{i=1}^{G} \left(\mathbb{I}(R(x, o_i) > 0) - P_t(x)\right)^2 = P_t(x)\left(1 - P_t(x)\right). \tag{9}$$

Since $V_t(x) > 0$ exactly for mixed-outcome groups, ranking by this empirical variance prioritizes non-ZVP groups and then favors a more balanced success–failure composition, an advantage established by Gao et al. (2026).

Candidates are ranked in descending lexicographic order by $V_t(x)$ and the prior–online discrepancy $|P_t(x) - P_{t-1}(x)|$, with remaining ties broken randomly. OAGS retains the top B prompts together with all G responses in each group, yielding $\mathcal{B}_t \subseteq \widetilde{\mathcal{B}}_t$ and $|\mathcal{B}_t| = B$. If fewer than B groups have positive variance, zero-variance groups fill the remaining positions under the same ranking. Only retained prompts consume training quotas; online evaluations of discarded candidates do not. Selection precedes log-probability computation, advantage estimation, and policy optimization.

3.3 Progress-Adaptive Curriculum Boundary

OAGS controls curriculum movement using changes between stored capability estimates and fresh candidate observations. Averaging these changes over the complete candidate batch gives:

$$\Delta_t = \frac{1}{|\widetilde{\mathcal{B}}_t|}\sum_{x \in \widetilde{\mathcal{B}}_t} \left(P_t(x) - P_{t-1}(x)\right). \tag{10}$$

Here, $P_{t-1}(x)$ is the latest stored observation, which may have been collected several training steps earlier. Thus, Δ_t is an observed capability-change signal, not a direct measurement of improvement caused by the immediately preceding policy update. It uses all candidates, rather than only groups favored by the final selector.

To distinguish broad changes from a few large deviations, the controller also measures directional consistency as $D_t = |\widetilde{\mathcal{B}}_t|^{-1}\sum_{x \in \widetilde{\mathcal{B}}_t} \operatorname{sgn}(P_t(x) - P_{t-1}(x))$. The two statistics update a normalized curriculum phase $c_t \in [0, 1]$:

$$c_{t+1} = \operatorname{clip}\left(c_t + \eta(\Delta_t, D_t), 0, 1\right), \tag{11}$$

where η advances the phase more rapidly under sufficiently strong and consistent positive changes, retreats under sufficiently strong and consistent negative changes, and otherwise advances conservatively. The exact pacing increments and thresholds are given in the appendix.

The progression of the curriculum phase induces a corresponding adaptation of the target capability center and the sampling bandwidth, which are defined as follows:

$$\mu_t = \mu_{\text{start}} + (\mu_{\text{end}} - \mu_{\text{start}}) \cdot \sqrt{c_t}, \tag{12}$$

$$\sigma_t = \sigma_{\text{start}} + (\sigma_{\text{end}} - \sigma_{\text{start}}) \cdot c_t. \tag{13}$$

With $\mu_{\text{start}} > \mu_{\text{end}}$, advancing the phase shifts the proposal toward lower-pass-rate prompts, while retreating returns it toward higher-pass-rate regions. The bandwidth determines the range around this target and remains constant when its two endpoints are equal. Feedback collected at step t therefore controls the Gaussian proposal used at step $t + 1$.

3.4 Optimization with Standard GRPO

The retained groups are optimized using the GRPO clipped surrogate with $\beta = 0$. To express the resulting objective, let $\omega_t(\mathcal{D})$ denote the joint distribution of a prompt and its complete response group obtained by choosing uniformly among the B retained groups. This distribution is induced by candidate proposal, fresh rollout generation, and outcome-dependent selection, conditional on the pre-rollout tracker state and behavior policy. The step-t objective is defined as:

$$\mathcal{J}(\theta; t) = \mathbb{E}_{\left(x,\{o_i\}_{i=1}^{G}\right)\sim\omega_t(\mathcal{D})}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{j=1}^{|o_i|}\min\Big(r_{i,j}(\theta)A_{i,j},\ \mathrm{clip}\big(r_{i,j}(\theta), 1-\epsilon, 1+\epsilon\big)A_{i,j}\Big)\right]. \tag{14}$$

The likelihood ratios follow Equation 2, and the advantages are computed using Equation 1. The sampling and selection distribution is held fixed during the policy update.

All candidate trajectories are freshly generated, OAGS introduces no historical trajectory replay. Nevertheless, retaining groups according to their observed outcomes changes their training distribution, not merely the marginal distribution over prompts. No selection-specific importance weighting is applied to recover the original uniform-prompt objective: OAGS intentionally changes the training-data distribution while retaining the GRPO advantage estimator and clipped surrogate.

4 Experiments

In this section, we first describe the experimental setup, including the evaluated models, training data, benchmarks, baselines, and implementation details (Section 4.1). We then compare OAGS with GRPO and representative zero-variance prompt filtering methods (Section 4.2). Finally, we conduct ablation studies to examine the contribution of the key design choices in OAGS (Section 4.4).

4.1 Experimental Setup

Models and Datasets. We evaluate OAGS on Qwen3-8B-Base and Qwen3-8B (Yang et al., 2025; Liu et al., 2026). All methods draw training data from the same DAPO-Math pool (Yu et al., 2025), which contains approximately 17K mathematical reasoning problems. We evaluate on six benchmarks: Minerva (Lewkowycz et al., 2022), AMC23, MATH500 (Hendrycks et al., 2021), AIME24, AIME25, and OlympiadBench (He et al., 2024). For each problem, we sample eight responses and report mean response accuracy (Acc@8) and the proportion of problems with at least one correct response (Pass@8). Additional generalization results are provided in Appendix A.

Baselines. We compare OAGS with GRPO and representative methods for improving RLVR data efficiency through prompt selection and rollout allocation. GRPO (Shao et al., 2024) serves as the reference without prompt-group filtering. GRPO with Dynamic Sampling (DS) adopts DAPO's post-rollout filtering strategy, resampling until an update batch of non-ZVP groups is obtained (Yu et al., 2025). GRESO (Zheng et al., 2025) uses historical reward dynamics to probabilistically skip uninformative prompts before generation. DEPO (Tang et al., 2026) combines offline training-subset selection with online explorability-based filtering and revisiting of under-explored prompts. These comparisons assess alternative ways to select, allocate, and reuse training data. For DS and GRESO, we report rollout-matched (r) and policy-update-matched (g) comparisons, both referenced to the OAGS checkpoint at 267 policy-update steps. The r variants match OAGS's cumulative generated responses (including filtered ones); the g variants match its policy-update steps. Offline tracker initialization is counted separately from the training rollout budget.

4.2 Main Results

Table 1 compares OAGS with GRPO and data-selection baselines on six mathematical reasoning benchmarks using Qwen3-8B and Qwen3-8B-Base.

Overall Performance. OAGS achieves the highest macro-average Acc@8 and Pass@8 among all evaluated methods for both model initializations. On Qwen3-8B, it reaches 61.1% Acc@8 and 75.2% Pass@8, improving over GRPO by 2.3 and 3.5 percentage points, respectively. On Qwen3-8B-Base, OAGS achieves 48.6% and 64.5%, corresponding to gains of 1.5 and 3.7 points. These

Table 1: **Evaluation results on six mathematical reasoning benchmarks.** We report accuracy (*Acc@8*) and pass rate (*Pass@8*) under an inference budget of eight sampled responses. The final two columns report the macro-average over all six benchmarks.

Method	Minerva		AMC23		MATH500		AIME24		AIME25		OlympiadBench		Average	
	Acc@8	Pass@8	Acc@8	Pass@8	Acc@8	Pass@8	Acc@8	Pass@8	Acc@8	Pass@8	Acc@8	Pass@8	Acc@8	Pass@8
Qwen3-8B trained on **DAPO-Math-17k**														
GRPO	45.8	56.6	84.1	95.0	91.0	95.8	40.0	60.0	33.3	53.3	58.3	69.6	58.8	71.7
DS-r	45.5	58.1	79.1	95.0	89.9	95.6	32.5	46.7	32.1	40.0	57.4	66.7	56.1	67.0
DS-g	46.1	58.1	81.9	95.0	90.7	95.2	41.2	56.7	32.5	46.7	56.9	67.6	58.2	69.9
GRESO-r	36.0	45.6	77.8	95.0	88.8	95.2	32.1	53.3	25.0	43.3	58.4	70.2	53.0	67.1
GRESO-g	36.6	50.0	80.3	92.5	89.1	95.6	35.4	63.3	27.1	36.7	58.0	70.1	54.4	68.0
DEPO	45.8	57.4	84.7	95.0	89.6	95.8	38.8	66.7	31.3	53.3	57.8	69.9	58.0	73.0
OAGS	47.8	58.5	86.9	97.5	90.5	95.4	51.3	76.7	31.7	53.3	58.4	70.1	61.1	75.2
Δ(w.r.t GRPO)	**+2.0**	**+1.9**	**+2.8**	**+2.5**	**-0.5**	**-0.4**	**+11.3**	**+16.7**	**-1.6**	**+0.0**	**+0.1**	**+0.5**	**+2.3**	**+3.5**
Qwen3-8B-Base trained on **DAPO-Math-17k**														
GRPO	36.8	50.4	70.6	87.5	84.5	93.0	22.5	40.0	17.9	30.0	50.2	64.1	47.1	60.8
DS-r	36.8	48.2	63.1	85.0	85.5	94.0	23.8	36.7	18.8	23.3	48.8	62.2	46.1	58.2
DS-g	36.5	51.1	68.8	85.0	85.6	93.8	22.9	46.7	18.3	30.0	48.2	63.0	46.7	61.6
GRESO-r	38.2	49.6	66.2	82.5	84.2	93.2	22.1	36.7	15.0	30.0	46.8	61.6	45.4	58.9
GRESO-g	37.9	46.0	70.3	85.0	83.8	91.2	19.6	36.7	17.9	26.7	47.6	62.7	46.2	58.0
DEPO	38.1	52.9	69.7	92.5	84.0	93.0	20.8	40.0	15.4	36.7	49.4	65.3	46.2	63.4
OAGS	42.2	54.0	68.4	92.5	86.1	94.0	25.0	43.3	18.8	36.7	50.9	66.4	48.6	64.5
Δ(w.r.t GRPO)	**+5.4**	**+3.6**	**-2.2**	**+5.0**	**+1.6**	**+1.0**	**+2.5**	**+3.3**	**+0.9**	**+6.7**	**+0.7**	**+2.3**	**+1.5**	**+3.7**

improvements span both evaluation criteria: OAGS increases average response correctness as well as the proportion of problems solved within eight attempts. The aggregate advantage is therefore not restricted to one metric or one model initialization.

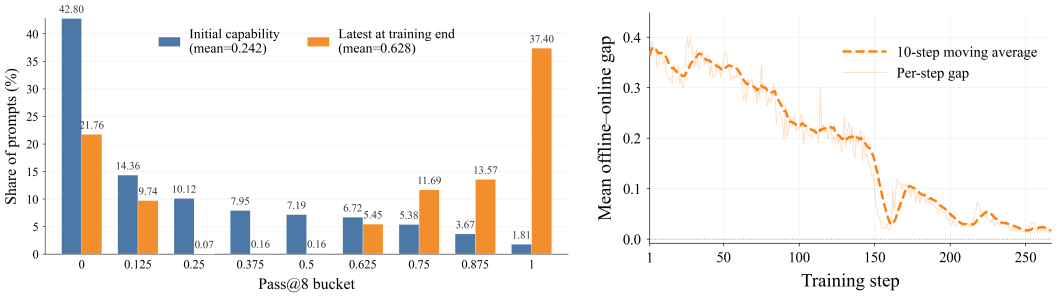
Comparison with Data-Selection Baselines. OAGS exceeds all reported DS and GRESO variants in both macro-average metrics. Relative to DS-g, it improves ACC@8/PASS@8 by 2.9/5.3 points on Qwen3-8B and 1.9/2.9 points on Qwen3-8B-Base. The advantage also extends to DEPO, which achieves the highest baseline macro-average PASS@8 for both models. Compared with DEPO, OAGS improves ACC@8/PASS@8 by 3.1/2.2 points on Qwen3-8B and 2.4/1.1 points on Qwen3-8B-Base. Thus, the gains persist against both prompt-filtering and offline–online data-selection approaches, rather than arising only in comparison with unfiltered GRPO.

Benchmark-Level Analysis. On Qwen3-8B, the largest improvement over GRPO occurs on AIME24: ACC@8 rises from 40.0% to 51.3%, while PASS@8 increases from 60.0% to 76.7%. OAGS also attains the best values for both metrics on Minerva and AMC23. On Qwen3-8B-Base, improvements over GRPO are more broadly distributed: ACC@8 increases on five of the six benchmarks, and PASS@8 increases on all six. Notable gains include 5.4 points in Minerva ACC@8 and 6.7 points in AIME25 PASS@8. The improvements are not uniform across individual metrics. For example, on Qwen3-8B-Base, AMC23 PASS@8 increases from 87.5% to 92.5% despite a decrease in ACC@8 from 70.6% to 68.4%; on Qwen3-8B, ACC@8 is slightly lower than GRPO on MATH500 and AIME25. Overall, the results show improved aggregate response accuracy and problem coverage across both initializations, with benchmark-dependent trade-offs rather than uniform dominance on every task.

4.3 DETAILED ANALYSIS

Tracker dynamics and prior–online discrepancy. Figure 2(a) shows that the mean stored pass rate increases from 0.242 to 0.628, accompanied by a shrinking all-failure bucket and an expanding all-success bucket. This shift is consistent with capability improvement alongside increasing saturation: previously uninformative prompts may become useful, while others lose their within-group reward variation. It therefore motivates revising the sampling preference rather than treating prompt utility as fixed.

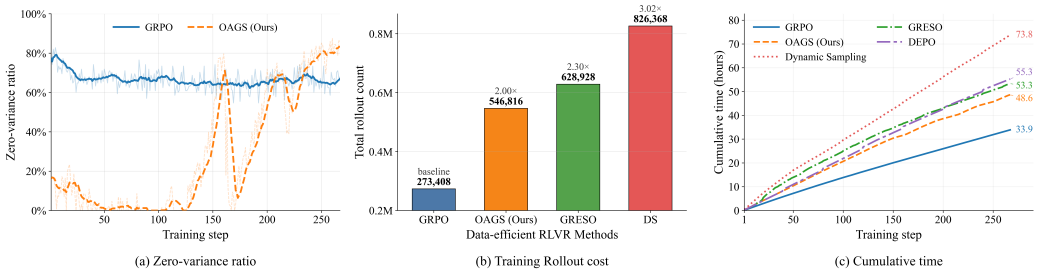
Meanwhile, the mean prior–online discrepancy decreases from approximately 0.37 to 0.02, despite transient fluctuations (Figure 2(b)), indicating closer agreement between stored estimates and fresh outcomes on evaluated candidates. These diagnostics characterize the evolving sampling state,



(a) Tracker success-rate distribution.

(b) Prior–online discrepancy.

Figure 2: **Tracker diagnostics during OAGS training.** (a) Distribution of per-prompt empirical success rates from the first and last available tracker records. (b) Logged mean discrepancy between the sampling prior and online feedback. In (b), the faint curve shows per-step values and the bold curve shows a 10-step trailing moving average.



(a) Zero-variance ratio

(b) Training Rollout cost

(c) Cumulative time

Figure 3: **Sampling behavior, rollout usage, and runtime.** (a) Post-selection zero-variance group ratio in the final update batch. (b) Total training rollout cost through step 267, with eight responses sampled per prompt. (c) Cumulative training time over the same horizon.

rather than a synchronized evaluation of dataset-wide capability. OAGS uses the signed prior–online changes and their directional consistency to translate this feedback into curriculum adjustments.

Post-selection ZVP and sampling efficiency. Figure 3(a) shows that OAGS maintains a near-zero ZVP ratio in the retained batch through much of early and mid training, with an increase toward the end. Dynamic Sampling and GRESO are omitted from this panel because their filtering procedures produce ZVP-free update batches by construction in our experiments. Unlike these repeated-filtering approaches, OAGS evaluates a single candidate pool per iteration and permits zero-variance groups when insufficient non-ZVP candidates are available. Its trade-off must therefore be assessed jointly with the cost of constructing each batch, rather than by post-selection ZVP alone.

This fixed candidate budget limits the generation overhead (Figure 3(b)): OAGS generates twice as many responses as GRPO, but 13.1% fewer than GRESO and 33.8% fewer than Dynamic Sampling. The advantage over the data-selection baselines also appears in wall-clock time (Figure 3(c)). At step 267, OAGS requires 48.6 hours, compared with 33.9 hours for GRPO, but reduces runtime by 8.8%, 12.1%, and 34.1% relative to GRESO, DEPO, and Dynamic Sampling, respectively. Together with the main results, these measurements support a favorable performance–runtime trade-off among the evaluated data-selection methods. In particular, stronger downstream performance does not require strictly ZVP-free updates: OAGS accepts some residual zero-variance groups while avoiding repeated generation to replace every such group.

4.4 Ablation Study

Table 2 examines curriculum scheduling, Gaussian candidate allocation, and tail reservation.

Table 2: Ablation on the components of OAGS. Each cell reports Acc@8 / Pass@8 metrics.

Method	Minerva	AMC23	MATH500	AIME24	AIME25	Olympiad.	Average
OAGS (Full)	42.2 / 54.0	68.4 / 92.5	86.1 / 94.0	25.0 / 43.3	18.8 / 36.7	50.9 / 66.4	48.6 / 64.5
w/ Linear Curriculum	41.4 / 52.2	69.7 / 90.0	85.8 / 93.2	19.6 / 30.0	22.5 / 43.3	51.0 / 66.7	48.3 / 62.6
w/o Curriculum Controller	39.1 / 51.8	68.1 / 87.5	86.7 / 93.8	24.6 / 46.7	19.2 / 33.3	51.6 / 66.2	48.2 / 63.2
w/o Gaussian Proposal	39.9 / 52.6	66.6 / 85.0	85.8 / 92.6	21.7 / 40.0	21.7 / 33.3	51.1 / 64.7	47.8 / 61.4
w/o Tail Reservation	41.2 / 53.7	70.3 / 92.5	86.2 / 94.0	24.2 / 50.0	22.5 / 33.3	50.8 / 67.0	49.2 / 65.1
GRPO	36.8 / 50.4	70.6 / 87.5	84.5 / 93.0	22.5 / 40.0	17.9 / 30.0	50.2 / 64.1	47.1 / 60.8

Curriculum Trajectory. Replacing the feedback-controlled square-root curriculum with a training-step-based linear schedule lowers average ACC@8/PASS@8 from 48.6/64.5 to 48.3/62.6. The square-root trajectory moves more rapidly through the initial higher-pass-rate region and slows toward the lower-pass-rate endpoint, motivating additional attention to harder capability regions later in training. The observed difference favors the full scheduling design, although this variant changes both pacing and trajectory shape and therefore does not isolate the effect of nonlinearity.

Curriculum Controller. Removing the controller replaces feedback from Δ_t and D_t with training-step-based progression while retaining the square-root trajectory. Average ACC@8 and PASS@8 decrease to 48.2 and 63.2, respectively, corresponding to drops of 0.4 and 1.3 percentage points. This comparison supports using differences between fresh rollout outcomes and stored pass rates to guide curriculum movement, rather than advancing according to training progress alone.

Gaussian Proposal. Replacing Gaussian candidate allocation with uniform sampling under the same warmup and quota eligibility rules yields 47.8 ACC@8 and 61.4 PASS@8, the largest aggregate decreases among the evaluated variants. Relative to full OAGS, PASS@8 decreases on all six benchmarks, with a macro-average drop of 3.1 points. Since the online selector is retained, these results support capability-guided candidate allocation in addition to post-rollout filtering, rather than relying on oversampling alone.

Tail Reservation. Removing tail reservation allocates candidates across all eligible capability buckets using Gaussian weights, without a dedicated tail quota. Tail prompts remain eligible, but their share of the candidate batch is no longer prescribed. This variant achieves 49.2 ACC@8 and 65.1 PASS@8, exceeding full OAGS by 0.6 points on both metrics. Thus, fixed tail reservation does not yield the best aggregate performance in the reported results. This finding concerns the explicit allocation constraint, not the necessity of exploration itself, since the variant still samples outside the Gaussian core.

5 CONCLUSION

We presented *Objective-Aware Gaussian Sampling* (OAGS), a feedback-controlled prompt curriculum that addresses zero-variance groups through adaptive rollout allocation. OAGS organizes prompts by empirical pass rates and uses discrepancies between stored estimates and fresh observations to regulate the pace and direction of a Gaussian curriculum. Core exploitation concentrates candidates around the target capability region, while tail exploration revisits less frequently evaluated prompts. By separating candidate evaluation from training selection, OAGS uses all observed outcomes to inform subsequent sampling while retaining a fixed-size subset of response groups for KL-free GRPO optimization.

Experiments on six mathematical reasoning benchmarks show improved macro-average accuracy and eight-attempt problem coverage across both evaluated model initializations. At a matched update count, OAGS also requires less training time than the evaluated adaptive data-selection baselines, although its additional candidate generation remains more expensive than vanilla GRPO. These results show that stronger downstream performance need not require strictly ZVP-free update batches. More broadly, they support treating prompt utility as an evolving state: a rollout that provides no immediate group-relative advantage can still inform future sampling. Coordinating this observational value with the selection of informative training groups offers a practical approach to improving the performance–cost trade-off of reasoning-oriented reinforcement learning.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and language polishing, including refining phrasing, improving clarity and concision, and organizing the presentation. The authors reviewed and revised all AI-assisted text to ensure consistency with the methodology, experimental results, and intended conclusions. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

REFERENCES

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 2026.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 16981–17017, 2026.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Haolin Shi, Chaowen Hu, Lu Pan, Ke Zeng, and Xunliang Cai. How to allocate, how to learn? dynamic rollout allocation and advantage modulation for policy optimization. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14727–14744, 2026.

Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt curriculum learning for efficient LLM post-training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zqOCacBD3P.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Dac Lai, and Eunho Yang. No prompt left behind: Exploiting zero-variance prompts in LLM reinforcement learning via entropy-guided advantage shaping. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kiXFIESZKv.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Shipeng Li, Zhiqin Yang, Shikun Li, Xiaobo Xia, Hengyu Liu, Xinghua Zhang, Gaode Chen, Dong Fang, Ying Tai, and Zhe Peng. Learnalign: Data selection for llm reinforcement learning with improved gradient alignment. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40417–40434, 2026.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling, 2025. URL https://arxiv.org/abs/2502.11886.

Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, Jian Hu, et al. Tricks or traps? a deep dive into RL for LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Faster, more efficient RLHF through off-policy asynchronous learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating rl finetuning of reasoning models? In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1240–1250, 2026.

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

Xinyu Tang, Zhenduo Zhang, Yurou Liu, Xin Zhao, zujie wen, Zhiqiang Zhang, and JUN ZHOU. Towards high data efficiency in reinforcement learning with verifiable reward. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=sruA4AZmZI.

Xu Wan, Yansheng Wang, Wenqi Huang, and Mingyang Sun. Buffer matters: Unleashing the power of off-policy reinforcement learning in large language model reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RduOiisl1S.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ningyuan Yang, Weihua Du, Weiwei Sun, Sean Welleck, and Yiming Yang. Gradalign: Gradient-aligned data selection for LLM reinforcement learning. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=bGUY6CJSbu.

Jiarui Yao, Yifan HAO, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and RL. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tA0l08qS5N.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38, 2025.

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

Ruiqi Zhang, Daman Arora, Song Mei, and Andrea Zanette. SPEED-RL: Faster training of reasoning models via online curriculum learning. In *2nd AI for Math Workshop @ ICML 2025*, 2025. URL https://openreview.net/forum?id=WQLQI6of9T.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *Advances in Neural Information Processing Systems*, 38, 2025.

APPENDIX

A FULL IMPLEMENTATION DETAILS

We present the full implementation and evaluation details of OAGS. OAGS and all baselines are implemented on top of the `verl` framework, ensuring that all experiments are conducted within the same training pipeline. Both Qwen3-B and Qwen3-8B-Base are trained on $8\times$ NVIDIA A100 GPUs. The detailed hyperparameter settings are provided in Table 3.

For both model scales, we use a training batch size of 128, a mini-batch size of 64, and 8 rollout responses per prompt. The maximum response length is 4096 tokens. During rollout generation, we set the sampling temperature to 1.0 and `top_p` to 1.0. We use the `math_rewards` implementation provided by `verl` as the outcome-based reward function. The policy is optimized with a learning rate of 1×10^{-6}, no learning-rate warmup, and a KL coefficient of 0. Both models are trained for 2 epochs, corresponding to 267 training steps. We use 4 rollout responses per prompt for training-time validation.

For final evaluation, we evaluate the final checkpoint for all baselines on Minerva, AMC23, MATH500, AIME24, AIME25, and OlympiadBench using the open-source evaluation framework released by the Qwen2.5-Math project (Yang et al., 2024). For each problem, we sample 8 responses with a maximum response length of 4096 tokens. The evaluation temperature and `top_p` are set to 1.0 and 0.7, respectively. The same evaluation configuration is used for both model scales and across all benchmarks.

Table 3: **Full hyperparameter setting of OAGS.**

Hyperparameters	**Qwen3-8B-Base + DAPO-Math-17k**	**Qwen3-8B + DAPO-Math-17k**
Training hardware	$8\times$ A100	$8\times$ A100
Training batch size	128	128
Mini batch size	64	64
Training rollouts	8	8
Max response length	4096	4096
Number of epochs	2	2
Total training steps	267	267
Training temperature	1.0	1.0
Training top_p	1.0	1.0
Validation rollouts	4	4
Validation top_p	0.7	0.7
Validation temperature	1.0	1.0
Reward function	`math_rewards`	`math_rewards`
Policy objective	vanilla	vanilla
KL coefficient	0	0
Learning rate	1×10^{-6}	1×10^{-6}
LR warmup steps	0	0
Evaluation checkpoint	final	final
Evaluation samples	8	8
Evaluation max length	4096	4096
Evaluation temperature	1.0	1.0
Evaluation top_p	0.7	0.7

We mostly follow the standard hyperparameter settings from recent works (Yu et al., 2025; Cheng et al., 2026; Zheng et al., 2025).

We use the clip-higher scheme with $\epsilon_{\text{high}} = 0.2$ and $\epsilon_{\text{low}} = 0.2$, apply token-level loss, and exclude the KL divergence term from the objective.

In addition to the common settings shared across methods, we also adopt the unique hyperparameter configurations of each baseline. For GRESO (Zheng et al., 2025), we follow their probabilistic filtering setup exactly: $\alpha_{\text{easy}} = \frac{1}{12}$, $\alpha_{\text{hard}} = \frac{1}{6}$, $p_{\text{easy}} = 0.5$, $p_{\text{hard}} = 0.5$, $\Delta p = 0.01$, and $\beta = 1.25$. For Dynamic Sampling, the only deviation from DAPO (Yu et al., 2025) is that we omit the *soft*

overlong punishment to ensure a fair comparison with other methods, which also do not include this form of length-based reward shaping.

Training horizon and checkpoint selection. For both model initializations, the standard training schedule consists of two epochs and 267 policy-update steps. Unless a budget-matched variant is specified, evaluation uses the checkpoint after the final, 267th policy-update step. Here, a policy-update step denotes one training iteration that constructs an update batch and performs the associated policy optimization, rather than an individual mini-batch optimizer step.

The rollout-matched and policy-update-matched comparisons use the OAGS checkpoint at step 267 as their common reference. With $M = 256$ candidate prompts and $G = 8$ responses per prompt, its cumulative training rollout budget is $546{,}816$ generated responses. DS-r and GRESO-r are evaluated at checkpoints matched to this response-generation budget, whereas DS-g and GRESO-g are evaluated at policy-update step 267. Thus, the rollout-matched checkpoints need not occur at the final training step. Budget accounting includes all newly generated training responses, regardless of whether their groups are retained for optimization. The offline initialization described below is excluded from this training-only budget.

Tracker initialization. Before online training, we initialize the tracker with a single evaluation pass over the full training pool $\mathcal{D}$. For each prompt x, an initialization policy π_{init} generates $G_{\text{init}} = 4$ responses, and the initial state is:

$$P_0(x) = \frac{1}{G_{\text{init}}} \sum_{i=1}^{G_{\text{init}}} \mathbb{I}\left\{R\big(x, o_i^{\text{init}}\big) > 0\right\}, \qquad o_i^{\text{init}} \sim \pi_{\text{init}}(\cdot \mid x). \tag{15}$$

This initialization evaluates every training prompt once as a group of four responses. It is distinct from online candidate evaluation, which generates $G = 8$ responses per prompt.

The initial tracker provides coarse prompt-level difficulty information for constructing early candidate proposals, rather than a precisely calibrated estimate of each target policy's success probability. The initialization policy need not coincide with the target policy: the same prompt-indexed prior can initialize runs with different target models on the same training pool. Such reuse treats the stored values as a starting difficulty prior, not as model-independent success probabilities.

During training, fresh outcomes from the current target policy replace the stored values of all evaluated candidates according to the capability-update rule in the main text. Prompts not evaluated in the current iteration retain their previous states. Consequently, a reused prior is progressively adapted through target-model observations rather than remaining a fixed offline assignment. When the initialization and target policies differ, the first prior–online changes can also reflect cross-model capability differences, in addition to sampling noise and subsequent learning progress.

Initialization and training cost accounting. The one-time tracker initialization generates $G_{\text{init}}|\mathcal{D}| = 4|\mathcal{D}|$ responses. The training rollout cost reported in Figure 3(b) counts only responses generated during the subsequent online training loop, including discarded candidate groups. It does not include this offline initialization pass.

For an OAGS run charged with the full initialization cost, the combined initialization and training response count is therefore:

$$C_{\text{OAGS}}^{\text{init+train}} = 4|\mathcal{D}| + 546{,}816. \tag{16}$$

If the same initialization is reused across multiple target-model runs, its construction is a shared preprocessing cost rather than a separate initialization pass for every run. We distinguish this one-time cost from the training-only budget used in rollout-matched comparisons.

B Curriculum Controller and Diagnostics

This appendix specifies the pacing function in Section 3.3, the Gaussian proposal configuration, and the interpretation of the curriculum diagnostics. We retain the main-text notation: $\widetilde{\mathcal{B}}_t$ contains M distinct candidate prompts, $\mathcal{B}_t$ contains the B prompts retained for optimization, and $P_{t-1}(x)$ is the latest empirical pass rate available before iteration t.

B.1 Exact Feedback Controller

Virtual clock and normalized phase. The controller maintains a virtual curriculum step s_t, distinct from the training-iteration index t. Its normalized phase is

$$c_t = \frac{s_t}{T_{\text{cur}}}, \qquad 0 \le s_t \le T_{\text{cur}}, \tag{17}$$

where $T_{\text{cur}} \ge 1$ is the curriculum normalization horizon, with implementation default 300. It is configured separately from the training horizon of 267 iterations: the virtual clock may advance faster or slower than the iteration counter and may also retreat.

Magnitude and directional consistency. For each evaluated candidate, write $\delta_t(x) = P_t(x) - P_{t-1}(x)$. The mean change Δ_t is defined in Equation 10. The directional statistic in the main text can equivalently be written as

$$D_t = \frac{1}{M} \sum_{x \in \widetilde{\mathcal{B}}_t} \left(\mathbb{I}\{\delta_t(x) > 0\} - \mathbb{I}\{\delta_t(x) < 0\} \right). \tag{18}$$

Thus, Δ_t measures the mean signed change, whereas D_t measures the imbalance between positive and negative changes across candidates. Unchanged observations contribute zero to D_t but remain in its denominator. Directional consistency is measured within the current candidate batch, not across successive iterations. Both statistics are computed before training selection using all M candidate groups, rather than only the B retained groups.

Piecewise pacing rule. The phase increment in Equation 11 is

$$\eta(\Delta, D) = \frac{1}{T_{\text{cur}}} \begin{cases} 1.5, & \Delta > \tau_\Delta \text{ and } D > \tau_{\text{fwd}}, \\ -1.0, & \Delta < -\tau_\Delta \text{ and } D < -\tau_{\text{bwd}}, \\ 0.4, & \text{otherwise}, \end{cases} \tag{19}$$

with magnitude threshold $\tau_\Delta = 0.05$, forward directional threshold $\tau_{\text{fwd}} = 0.10$, and backward directional threshold $\tau_{\text{bwd}} = 0.05$. Equivalently, the implementation updates

$$s_{t+1} = \text{clip}\left(s_t + T_{\text{cur}} \cdot \eta(\Delta_t, D_t), 0, T_{\text{cur}}\right). \tag{20}$$

Dividing this update by T_{cur} gives exactly the phase update in the main text. In particular, 1.5, -1.0, and 0.4 are increments of the virtual clock, not of the normalized phase.

Acceleration and retreat each require their magnitude and directional conditions to hold jointly. The comparisons are strict; all remaining cases use conservative advancement, including weak negative feedback or disagreement between the two statistics. Near-zero feedback therefore does not stop the curriculum. The controller uses raw per-iteration statistics; the moving average in Figure 4(b) is only a visualization.

For a fresh run, the implementation initializes $s_0 = 0$ and uses $\Delta_0 = D_0 = 0$ as the initial feedback state. Applying the conservative increment before the first proposal gives $s_1 = 0.4$ and $c_1 = 0.4/T_{\text{cur}}$. Thereafter, the proposal at iteration $t+1$ uses the feedback collected at iteration t.

B.2 Proposal Configuration and Update Order

Candidate budget and core–tail allocation. OAGS evaluates a single candidate pool of $M = 256$ distinct prompts per policy-update step and retains $B = 128$ prompt groups for optimization, giving an oversampling factor of $M/B = 2$. Each candidate receives $G = 8$ freshly generated responses. All M candidate groups update the tracker, while only the retained B groups enter the policy update.

The Gaussian core is the one-standard-deviation region $|P_{t-1}(x) - \mu_t| \le \sigma_t$, as defined in the main text. We use a nominal core allocation fraction $\rho_{\text{core}} = 0.6827$, motivated by the mass of an untruncated Gaussian within one standard deviation:

$$\Pr_{Z \sim \mathcal{N}(0,1)}(|Z| \le 1) = \Phi(1) - \Phi(-1) \approx 0.6827, \tag{21}$$

where Φ is the standard normal cumulative distribution function. This gives an approximately 7:3 core–tail allocation, with nominal candidate counts:

$$M_{\text{core}} = \text{round}(\rho_{\text{core}} M) = 175, \qquad M_{\text{tail}} = M - M_{\text{core}} = 81. \tag{22}$$

The allocation is imposed on candidate positions; it is not an assumption about the fraction of training prompts whose stored pass rates lie inside the core. If either partition has insufficient eligible prompts, the other supplies the remaining candidates, provided the active pool contains at least M prompts. The realized core–tail ratio can therefore differ from its nominal target. This allocation applies before rollout and does not constrain the composition of the final update batch.

Center and bandwidth schedules. The center endpoints are $\mu_{\text{start}} = 0.625$ and $\mu_{\text{end}} = 0.375$; the bandwidth endpoints are $\sigma_{\text{start}} = 0.1875$ and $\sigma_{\text{end}} = 0.25$. Substitution into Equations 12 and 13 yields

$$\begin{aligned} \mu_t &= 0.625 - 0.25\sqrt{c_t}, \\ \sigma_t &= 0.1875 + 0.0625 c_t. \end{aligned} \tag{23}$$

Accordingly, curriculum advancement both lowers the target stored pass rate and broadens the bandwidth; retreat reverses both changes. Throughout the curriculum, $\mu_t \in [0.375, 0.625]$ and $\sigma_t \in [0.1875, 0.25]$. The bandwidth is therefore not constant in this configuration.

Before applying prompt-eligibility constraints, the nominal core band $[\mu_t - \sigma_t, \mu_t + \sigma_t]$ changes from $[0.4375, 0.8125]$ at $c_t = 0$ to $[0.125, 0.625]$ at $c_t = 1$. These are phase endpoints, not necessarily the states reached in the first and last training iterations. The schedule specifies a wider capability band around a lower target, rather than a fixed-width band translated toward harder prompts. It changes both the Gaussian preferences and the core–tail partition, while the candidate budget and prescribed core–tail allocation remain separate controls.

Proposal-safety warm-up. During the first 10 iterations, the quota-eligible proposal pool is additionally restricted to prompts satisfying $z_t(x) \leq 0.12$, using the predicted ZVP risk defined in the main text. This risk restriction is removed after warm-up, while training-usage constraints remain in force. The warm-up changes candidate eligibility; it neither suspends the feedback controller nor guarantees non-zero reward variance in fresh groups. It is also separate from learning-rate warm-up.

Feedback and training selection. The proposal at iteration t uses the stored rates P_{t-1} and Gaussian parameters determined by c_t. After generation and reward evaluation, all candidate outcomes update the tracker and determine (Δ_t, D_t). The trainer then applies the main-text selection rule: descending lexicographic ordering by $(V_t(x), |\delta_t(x)|)$, with random tie-breaking. Under binary outcome rewards, $V_t(x) > 0$ exactly for non-ZVP groups, so an explicit non-ZVP priority flag gives the same ordering.

All evaluated candidates increment their probe counts, whereas only retained prompts consume training-selection quotas. The retained response groups alone proceed to old-policy log-probability recomputation, advantage estimation, and policy optimization. The new feedback is used for the next proposal, not for the proposal that generated it. The implementation sets the data-loader worker count to zero to prevent prefetched candidate batches from bypassing this step-level feedback.

B.3 Online Feedback and Curriculum Diagnostics

Logged states and their relationship. Figure 4 reports complementary quantities from the same feedback loop. Let $N = |\mathcal{D}|$ be the size of the full training pool. Panel (a) shows the pre-rollout tracker mean

$$\bar{P}_t = \frac{1}{N} \sum_{x \in \mathcal{D}} P_{t-1}(x). \tag{24}$$

Panels (c) and (d) likewise show the pre-rollout states s_t and μ_t. Panel (b), in contrast, shows Δ_t from the resulting candidate outcomes and therefore governs the subsequent proposal.

Because the candidate prompts are distinct and only their stored values are replaced, the capability update implies

$$\bar{P}_{t+1} - \bar{P}_t = \frac{1}{N} \sum_{x \in \widetilde{\mathcal{B}}_t} \delta_t(x) = \frac{M}{N} \Delta_t. \tag{25}$$

For the raw logged quantities, panel (a) thus accumulates the signed feedback in panel (b), scaled by the fraction of the pool evaluated per iteration. Their agreement follows from the update rule and is not an independent calibration test.

Observed progression. The trajectory exhibits three broad stages. Initially, the signed feedback is negative, approximately -0.2, and the stored pool mean decreases slightly from approximately 0.24 to 0.22. The virtual clock remains near its lower bound, with the proposal center near its upper endpoint. This pattern is consistent with lower-bound clipping during early negative feedback rather than a warm-up rule that explicitly freezes the clock.

The feedback subsequently becomes predominantly positive, reaching approximately 0.25–0.28. Over much of the interval before step 150, the pool mean increases toward 0.62, the virtual clock advances rapidly, and the proposal center moves toward 0.4. This trajectory is consistent with frequent accelerated updates. The increasing pool mean and decreasing proposal center are not contradictory: the former summarizes stored capability observations, whereas the latter directs proposals toward lower-pass-rate regions of that evolving state space.

Later, the signed feedback fluctuates closer to zero and the pool mean stabilizes around 0.63. The virtual clock continues to advance at a lower rate, while the proposal center approaches approximately 0.38. This behavior is consistent with conservative progression rather than cessation of the curriculum. Under Equation 23, the slower phase progression also implies slower bandwidth expansion; this is a consequence of the specified schedule, not a separately plotted observation. The center trajectory additionally reflects the square-root dependence on c_t, so its slope alone is not a direct measure of the pacing increment.

Signed feedback versus absolute discrepancy. The signed feedback used by the controller is distinct from the mean absolute prior–online discrepancy discussed in the main-text tracker analysis. Define the latter as

$$E_t^{\mathrm{abs}} = \frac{1}{M} \sum_{x \in \widetilde{\mathcal{B}}_t} |P_t(x) - P_{t-1}(x)|. \tag{26}$$

Although $|\Delta_t| \le E_t^{\mathrm{abs}}$, a small signed mean need not imply a small absolute discrepancy, since positive and negative changes can cancel. Moreover, both quantities are measured on the proposed candidates, and the stored observations may have been collected under earlier policies. Both rates are estimated from finite response groups, so their differences also contain sampling noise. Neither quantity is a full-pool evaluation of the current policy.

Together, the panels illustrate accelerated progression followed by a feedback-dependent slowdown. They do not show a sustained late-stage retreat, and the unplotted directional statistic D_t is also required to identify the active controller branch. The central observation is that candidate feedback regulates subsequent proposal movement while refreshing the sampling state, including through groups excluded from the current policy update.

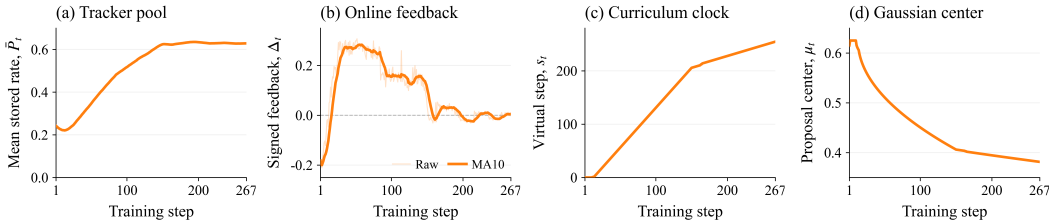


Figure 4: **Online feedback and curriculum dynamics in OAGS.** (a) Mean stored empirical pass rate $\bar{P}_t$ over the full training pool. (b) Signed online-minus-prior feedback Δ_t, averaged over all M candidate prompts before training selection. (c) Virtual curriculum step s_t. (d) Gaussian proposal center μ_t. Panels (a), (c), and (d) show pre-rollout states at iteration t; panel (b) shows feedback obtained from that rollout and used for the proposal at iteration $t + 1$. In (b), the faint curve shows raw feedback and the bold curve a 10-step trailing mean used only for visualization. The other panels show unsmoothed logged states. All panels cover training iterations 1–267.