CERO: WHERE AND WHEN TO ALLOCATE ROLLOUTS FOR RL POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive rollout methods for group-relative reinforcement learning typically allocate a fixed per-update budget across prompts. We instead study how to coordinate a finite rollout budget over the entire training horizon. We formulate this problem using a concave surrogate utility of cumulative prompt exposure and introduce CERO, an online primal–dual scheduler for prompt admission and budget pacing. In our experiments, each admitted prompt receives a fixed-size response group. CERO instead adapts which prompts are selected, how often they are revisited across rounds, and how many groups are generated in each round. A compact Fenchel representation linearizes the dependence on cumulative exposure, while projected online gradient descent updates prompt-specific supporting slopes and a shared budget price using reward-variation feedback and budget deviations. We establish pathwise guarantees for the surrogate allocation objective against fixed-rate and same-path time-varying benchmarks, with explicit terms for proxy discrepancy and rate variation. Under matched training-response budgets, CERO attains the highest avg@16 macro-average on each of three backbones across five mathematical reasoning benchmarks. Mechanistic analyses link CERO's prompt choices to within-group reward contrast, while multi-seed ablations show gains from adaptive pacing over both uniform and preset spending schedules.

1 INTRODUCTION

Reinforcement learning with verifiable rewards has become a standard approach for post-training reasoning models. Within this paradigm, recent work has increasingly focused on adaptive rollout allocation to improve training efficiency. Standard GRPO-style training typically assigns the same number of responses to each selected prompt (Shao et al., 2024; Guo et al., 2025), even though the resulting groups can differ substantially in informativeness. In particular, all-correct or all-incorrect groups provide no within-group reward contrast. Recent methods therefore adapt how rollout budgets are distributed across prompts: KnapsackRL (Li et al., 2025) allocates a fixed exploration budget according to estimated task value, VIP (Nguyen et al., 2026) allocates minibatch rollouts to reduce expected gradient variance, and HORA (Wang et al., 2026) redistributes a batch budget according to marginal hit utility. Collectively, these methods primarily adapt *where* rollouts are allocated within a given round-level budget. We focus on a complementary temporal dimension that arises when rollout decisions are coupled by a shared finite training budget:

Given a finite rollout budget, when should we spend more, and when less, across training?

This horizon-level problem remains nontrivial even when each admitted prompt receives a fixed-size response group. The scheduler must still decide which prompts to admit, how often to revisit them across rounds, and how much of the remaining budget to spend at different training stages. Rollouts spent now cannot be reassigned later, and repeated exposure to the same prompt may yield diminishing marginal utility. We capture this effect with a concave utility over cumulative prompt exposure. These couplings motivate a resource allocation formulation over prompts and rounds.

However, this finite-horizon formulation is not directly separable across rounds: each prompt's cumulative allocation appears inside a nonlinear concave utility, and the global budget further couples spending decisions over time. Our key methodological contribution is to adapt Fenchel dualization and projected online gradient descent (OGD) to this rollout optimization problem. The Fenchel representation linearizes dependence on cumulative exposure, reducing the coupled horizon objective

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

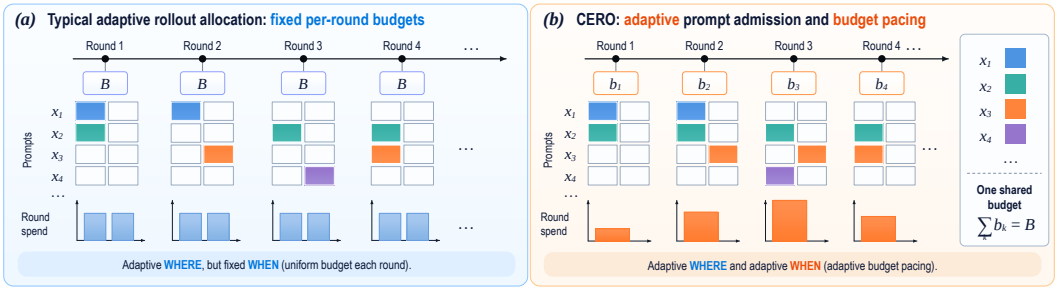

Figure 1: Schematic comparison of rollout optimization. (a) Typical adaptive methods redistribute a fixed per-round budget across prompts. (b) CERO coordinates prompt admission and per-round budget pacing under a shared finite training budget, with admitted prompts receiving fixed-size response groups.

to tractable round-level allocation problems. Projected OGD then adapts prompt-specific supporting slopes from reward feedback and a shared budget price from aggregate allocation deviations.

These dual signals play complementary roles in CERO: prompt-specific slopes guide prompt prioritization, while the shared price regulates budget pacing across rounds. A prompt-level Beta state provides a predictive signal of future within-group reward variation, while the dual variables and the remaining budget are carried forward across rounds, enabling coordination throughout training. This online structure admits pathwise guarantees for the surrogate allocation objective under fixed rates and a same-path time-varying benchmark. Empirically, CERO achieves the highest avg@16 macro-average on all three evaluated backbones under matched training-response budgets. Matched controls and mechanistic analyses further connect these gains to prompt-aware assignment and horizon-level budget adaptation, with CERO outperforming each ablated variant in multi-seed average performance.

Our contributions are threefold:

- We formulate finite-horizon rollout optimization as a resource-allocation problem with a concave utility over cumulative prompt exposure, and derive an online primal–dual scheduler using a compact Fenchel representation and projected OGD. The resulting scheduler coordinates prompt admission, repeated exposure across rounds, and per-round spending under a shared finite budget.
- We establish pathwise guarantees for the surrogate allocation objective under fixed rates and a same-path time-varying benchmark. The bounds explicitly separate online optimization error, reward-proxy discrepancy, and temporal rate variation.
- We evaluate CERO on three backbones and five mathematical reasoning benchmarks under the same rollout budgets. CERO obtains the highest avg@16 macro-average on each backbone. Matched controls and multi-seed ablations further connect these gains to prompt-aware assignment and horizon-level coordination, with CERO outperforming fixed-pacing, preset-pacing, and direct marginal-utility variants.

Relation to prior work. CERO lies at the intersection of three related directions: difficulty-aware sampling and filtering, adaptive rollout allocation, and online resource allocation. The first focuses on identifying and prioritizing informative prompts, the second studies how limited rollout budgets should be allocated across prompts, and the third provides a broader framework for coordinating sequential decisions under shared resource constraints. Appendix A provides a detailed review of these three lines of research.

2 Finite-Horizon Rollout Optimization

2.1 Problem Formulation

We consider a finite training process of K rounds over a fixed pool $\mathcal{X} = \{x_1, \ldots, x_M\}$ of M prompts. Let $N_{i,k}$ denote the number of units assigned to prompt i in round k, where one allocation

unit corresponds to a fixed quantum of $g \in \mathbb{Z}_{>0}$ generated responses. Define its cumulative exposure by $z_{i,k} := \sum_{t=1}^{k} N_{i,t},\ z_{i,0} = 0$. Let $N_{\max}, C, B \in \mathbb{Z}_{\geq 0}$ be the per-prompt per-round cap, the aggregate per-round capacity, and the total rollout budget. Our goal is to maximize the total utility of cumulative prompt exposure by allocating a finite rollout budget across prompts and rounds:

$$\begin{aligned} \max_{\mathbf{N} \in \{0,\dots,N_{\max}\}^{M \times K}} \quad & \sum_{i=1}^{M} U_i(z_{i,K}) \\ \text{s.t.} \quad & \sum_{i=1}^{M} N_{i,k} \leq C, \qquad k \in [K], \\ & \sum_{k=1}^{K} \sum_{i=1}^{M} N_{i,k} \leq B. \end{aligned} \tag{1}$$

Here U_i is a nondecreasing concave utility of cumulative exposure, specified in the next subsection. We assume

$$0 < B \leq KC, \quad C \leq MN_{\max}. \tag{2}$$

Denote the feasible set in eq. (1) by $\mathcal{N}_{B,C}$. We distinguish allocation units from generated-response counts using the superscript resp: $N_{i,k}^{\text{resp}} = gN_{i,k}$, $N_{\max}^{\text{resp}} = gN_{\max}$, $C^{\text{resp}} = gC$, and $B^{\text{resp}} = gB$. Finally, define

$$Z := KN_{\max}, \qquad \bar{B} := \frac{B}{K}, \qquad R := \frac{B}{M},$$

where Z is the maximum per-prompt exposure over training, $\bar{B}$ is the average per-round budget, and R is the equal-share exposure scale.

2.2 Value-capture utility

We use a concave surrogate utility to model diminishing returns to cumulative prompt exposure. Each prompt has a latent value-capture rate $q_i \in [0, 1/4]$, for which Section 3 provides a reward-variation proxy. We instantiate the utility with the normalized exponential form

$$U(q, z) := 1 - \exp\left(-4\lambda q \frac{z}{R}\right), \quad q \in [0, 1/4], \quad z \in [0, Z],$$

where $\lambda > 0$ controls saturation and z/R measures exposure relative to the equal-share horizon scale. By construction, the utility is nondecreasing and concave in z, encoding diminishing marginal utility with exposure. The marginal utility of an additional allocation unit satisfies

$$0 \leq \partial_z U(q, z) = \frac{4\lambda q}{R} \exp\left(-4\lambda q \frac{z}{R}\right) \leq \frac{\lambda}{R} =: \bar{c}. \tag{3}$$

This bound yields the compact slope domain $[0, \bar{c}]$ used in the Fenchel representation below. For comparison, consider a full-information oracle that knows $\boldsymbol{q} = (q_1, \dots, q_M)$ in advance and chooses the feasible allocation maximizing total value capture:

$$\mathsf{OPT}_K^{\text{stat}}(\boldsymbol{q}) := \max_{\mathbf{N} \in \mathcal{N}_{B,C}} \sum_{i=1}^{M} U(q_i, z_{i,K}). \tag{4}$$

Since the objective depends on the allocation history only through $z_{i,K}$, the oracle problem is determined by the terminal exposure vector. Appendix C.2 gives its optimal allocation.

2.3 Compact Fenchel representation

To simplify notation, let $U_i(z) := U(q_i, z)$. Using the uniform slope bound in eq. (3), define the restricted concave conjugate

$$U_i^*(\theta) := \inf_{0 \leq s \leq Z} \{\theta s - U_i(s)\}, \quad \theta \in [0, \bar{c}].$$

The interval $[0, \bar{c}]$ contains a supporting slope at every exposure level, so the representation remains exact on this compact domain. By concavity,

$$U_i(z) = \min_{0 \leq \theta \leq \bar{c}} \{\theta z - U_i^*(\theta)\}, \qquad 0 \leq z \leq Z.$$

Thus, the nonlinear dependence on cumulative exposure can be represented exactly through affine functions of z. To capture the scarcity of the shared budget across rounds, let $\mu \in [0, \bar{c}]$ denote the corresponding dual price. For a single round, let $\boldsymbol{n} = (n_1, \ldots, n_M)$ denote the allocation vector and define the continuous feasible region

$$\mathcal{P}_C := \left\{ \boldsymbol{n} \in \mathbb{R}^M : 0 \le n_i \le N_{\max},\ \sum_i n_i \le C \right\}.$$

Since C and $N_{\max}$ are integers, $\mathcal{P}_C$ is integral, so the continuous relaxation is exact for the round-level linear problem. Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M) \in [0, \bar{c}]^M$ collect the prompt-specific supporting slopes. Combining the Fenchel representation with the shared budget price μ makes the horizon-level upper bound separable across rounds, yielding the per-round dual objective

$$L(\boldsymbol{\theta}, \mu) = \bar{B}\mu + \max_{\boldsymbol{n} \in \mathcal{P}_C} \sum_{i=1}^{M} (\theta_i - \mu) n_i - \frac{1}{K} \sum_{i=1}^{M} U_i^*(\theta_i),$$

By weak duality, $\mathsf{OPT}_K^{\mathrm{stat}}(\boldsymbol{q}) \le KL(\boldsymbol{\theta}, \mu)$ for all $(\boldsymbol{\theta}, \mu)$ in the stated domains. Let $d_i := \theta_i - \mu$, which compares prompt x_i's supporting slope against the shared budget price.

3 Reward-Variation Proxy

We use predicted within-group reward variance as a proxy for the latent value-capture rate in Section 2.2. This quantity characterizes expected reward contrast in a fresh response group, rather than directly estimating the downstream improvement from a policy update. Let $r(x, y) \in \{0, 1\}$ be the verifiable reward. At the beginning of round k, we maintain an auxiliary Beta state for prompt x_i:

$$P_{i,k} \mid \mathcal{H}_i^k \sim \mathrm{Beta}(a_{i,k}, b_{i,k}), \quad a_{i,k}, b_{i,k} > 0,$$

where $\mathcal{H}_i^k$ denotes the reward history of prompt i. This state provides a compact predictive summary of the prompt's observed reward history. Let $G_{i,k} := N_{i,k}^{\mathrm{resp}} = gN_{i,k}$ denote the number of responses generated for prompt x_i in round k. Let $S_{i,k}$ and $F_{i,k}$ denote the numbers of successful and failed responses, and set $S_{i,k} = F_{i,k} = 0$ if there is no allocation to this prompt. After round k, we update the Beta state with forgetting factor $\rho \in (0, 1]$:

$$a_{i,k+1} = a_0 + \rho(a_{i,k} - a_0) + S_{i,k},\ b_{i,k+1} = b_0 + \rho(b_{i,k} - b_0) + F_{i,k}, \tag{5}$$

where $a_0, b_0 > 0$ are the prior parameters. When $\rho = 1$, the update reduces to standard cumulative Beta–Bernoulli updating. When $\rho < 1$, older evidence is gradually discounted toward the prior, allowing the state to track changes in the policy. We use predicted reward variation as a proxy for the latent value-capture rate:

$$\widehat{q}_{i,k} := \mathbb{E}[P_{i,k}(1 - P_{i,k}) \mid \mathcal{H}_i^k] = \frac{a_{i,k} b_{i,k}}{(a_{i,k} + b_{i,k})(a_{i,k} + b_{i,k} + 1)} \in [0, 1/4]. \tag{6}$$

Proposition 1 (Predicted within-group reward variance)**.** *For any group size $G \ge 2$, suppose that conditional on $(P_{i,k}, \mathcal{H}_i^k)$ the next G binary rewards are i.i.d. Bernoulli$(P_{i,k})$. Let $\bar{r}_G := G^{-1} \sum_{j=1}^{G} r^{(j)}$ and $\widehat{\mathrm{Var}}_G(r) := \frac{1}{G-1} \sum_{j=1}^{G} \left(r^{(j)} - \bar{r}_G\right)^2$. Then it holds that*

$$\mathbb{E}[\widehat{\mathrm{Var}}_G(r) \mid \mathcal{H}_i^k] = \widehat{q}_{i,k}.$$

Thus, $\widehat{q}_{i,k}$ is the predictive expected within-group reward variance. Let $m_{i,k} := \mathbb{E}[P_{i,k} \mid \mathcal{H}_i^k]$ denote the predictive mean and we can obtain

$$\widehat{q}_{i,k} = m_{i,k}(1 - m_{i,k}) - \mathrm{Var}(P_{i,k} \mid \mathcal{H}_i^k).$$

Therefore, $\widehat{q}_{i,k}$ is largest when the predicted success rate is near $1/2$, but is reduced when the Beta state is more uncertain.

Proposition 2 (Fixed-sample stationary calibration)**.** *Let $Y_1, \ldots, Y_n$ be i.i.d. Bernoulli(p) observations with fixed sample size n. Define $a := a_0 + \sum_{j=1}^{n} Y_j,\ b := b_0 + n - \sum_{j=1}^{n} Y_j,\ s_0 := a_0 + b_0$, and let $\widehat{q}_n := ab/[(a + b)(a + b + 1)]$ and $q := p(1 - p)$. Then it holds that*

$$\mathbb{E}|\widehat{q}_n - q| \le \frac{\sqrt{n}}{2(n + s_0)} + \frac{s_0}{n + s_0} + \frac{1}{4(n + s_0 + 1)}. \tag{7}$$

Consequently, $\widehat{q}_n$ is consistent at rate $O(n^{-1/2})$ under a stationary Bernoulli model with a fixed, nonadaptive sample size

Proposition 2 concerns fixed-sample stationary calibration. Our allocation guarantees instead hold pathwise for arbitrary realized proxy sequences, with the bounds explicitly accounting for proxy error relative to the target rates.

4 CERO ALGORITHM

To solve the finite-horizon rollout-allocation problem, we develop CERO, an online primal–dual scheduler summarized in Algorithm 1. At each round, CERO maintains normalized prompt-specific supporting slopes and a normalized shared budget price:

$$\theta_i^k = \bar{c}\,\vartheta_i^k, \quad \mu^k = \bar{c}\,\nu^k, \quad \vartheta_i^k, \nu^k \in [0,1].$$

After generating the selected responses and observing their rewards, CERO updates the predictive state and dual variables for the next round.

Round-level allocation Since $\theta_i^k - \mu^k = \bar{c}(\vartheta_i^k - \nu^k)$, define the normalized margin $d_{i,k} := \vartheta_i^k - \nu^k$. A positive margin indicates that prompt x_i's supporting slope exceeds the shared budget price. CERO first computes the capacity-constrained virtual allocation

$$\widetilde{\boldsymbol{N}}_k \in \arg\max_{\boldsymbol{n} \in \mathcal{P}_C \cap \mathbb{Z}^M} \sum_{i=1}^{M} d_{i,k} n_i. \tag{8}$$

This problem has a simple exact solution: prompts with positive margins are ranked in decreasing order, with up to $N_{\max}$ units assigned to each while capacity remains. CERO then allocates rollout units in this order up to the remaining global budget, which is updated as $B_{k+1} = B_k - \sum_i N_{i,k}$.

Since the round-level objective is linear in the allocation units, the ranked solution assigns selected prompts the per-prompt cap, except possibly for a residual allocation. In Section 6, we use $N_{\max} = 2$ and choose the round capacity and total budget as multiples of $N_{\max}$, so every selected prompt receives one eight-response group. CERO therefore adapts which prompts receive groups, how often they are revisited across rounds, and how many groups are generated per round.

Reward feedback and supporting exposure After observing the rewards from round k, CERO updates the Beta state using eq. (5) and recomputes the reward-variation proxy $\widehat{q}_{i,k+1}$ using eq. (6). Let $\widehat{U}_{i,k+1}(z) := U(\widehat{q}_{i,k+1}, z)$, $c_{i,k+1} := \frac{4\lambda \widehat{q}_{i,k+1}}{R}$. For a supporting slope θ, the corresponding exposure is

$$\hat{z}^*_{i,k+1}(\theta) \in \arg\min_{0 \le z \le Z} \left\{ \theta z - \widehat{U}_{i,k+1}(z) \right\}.$$

For our exponential utility, this minimizer has the closed form $\hat{z}^*_{i,k+1}(\theta) = \hat{z}^*(\theta; c_{i,k+1})$, where

$$\hat{z}^*(\theta; c) = \begin{cases} 0, & c = 0 \text{ or } \theta \ge c, \\ Z, & c > 0 \text{ and } 0 \le \theta \le c e^{-cZ}, \\ \dfrac{1}{c} \log \dfrac{c}{\theta}, & c > 0 \text{ and } c e^{-cZ} < \theta < c. \end{cases} \tag{9}$$

Dual updates Let $G_B := \max\{\bar{B}, C - \bar{B}\}$ denote the normalization scale for the budget-price update. CERO applies the normalized projected updates

$$\vartheta_i^{k+1} = \Pi_{[0,1]}\left[\vartheta_i^k - \gamma_\vartheta \frac{\widetilde{N}_{i,k} - \hat{z}^*_{i,k+1}(\bar{c}\vartheta_i^k)/K}{N_{\max}} \right], \; \nu^{k+1} = \Pi_{[0,1]}\left[\nu^k - \gamma_\nu \frac{\bar{B} - \sum_i \widetilde{N}_{i,k}}{G_B} \right] \tag{10}$$

The prompt-specific update compares the virtual allocation $\widetilde{N}_{i,k}$ with the per-round supporting exposure $\hat{z}^*_{i,k+1}(\bar{c}\vartheta_i^k)/K$: allocating above this level decreases the supporting slope, while allocating below it increases the slope. The shared price increases when total virtual allocation exceeds $\bar{B}$ and decreases when it falls below $\bar{B}$. Both updates use the virtual allocations $\widetilde{\boldsymbol{N}}_k$ rather than the budget-clipped allocations. Overall, the prompt-specific dual variables adapt relative prompt priority, while the shared price regulates budget pacing across rounds.

Algorithm 1 CERO: Cross-Epoch Adaptive Rollout Optimization

Require: Prompt pool $\{x_i\}_{i=1}^{M}$; horizon K; global budget B; round capacity C; per-prompt cap $N_{\max}$.
Set $R = B/M$, $Z = KN_{\max}$, $\bar{c} = \lambda/R$, $\bar{B} = B/K$, and $G_B = \max\{\bar{B}, C - \bar{B}\}$
Initialize $B_1 = B$, $(a_{i,1}, b_{i,1}) = (a_0, b_0)$, $\widehat{q}_{i,1}$ by equation 6, $\vartheta_i^1 = 4\widehat{q}_{i,1}$, and $\nu^1 \in [0, 1]$
for $k = 1, \ldots, K$ **do**
  Compute margins $d_{i,k} = \vartheta_i^k - \nu^k$ and virtual proposal $\widetilde{\boldsymbol{N}}_k$ by the top-score solution of equation 8
  Execute proposals in decreasing-margin order subject to remaining budget B_k, producing $\boldsymbol{N}_k$
  for $i = 1, \ldots, M$ **do**
    If $N_{i,k} > 0$, generate $N_{i,k}^{\mathrm{resp}} = gN_{i,k}$ responses from $\pi_{\phi^k}(\cdot \mid x_i)$ and observe $S_{i,k}$ successes and $F_{i,k}$ failures; otherwise, set $S_{i,k} = F_{i,k} = 0$.
    Update $(a_{i,k+1}, b_{i,k+1})$ by equation 5 and compute $\widehat{q}_{i,k+1}$
    Compute $\widehat{z}^*_{i,k+1}(\bar{c}\vartheta_i^k)$ by equation 9
    Update ϑ_i^{k+1} by equation 10
  end for
  Update ν^{k+1} by equation 10 and set $B_{k+1} = B_k - \sum_i N_{i,k}$
  If $\sum_i N_{i,k} > 0$, update the policy using all executed round-k groups; otherwise, set $\phi^{k+1} = \phi^k$.
  if $B_{k+1} = 0$ **then**
    Terminate.
  end if
end for

5 Performance Guarantees

We now analyze CERO with respect to the surrogate allocation objective in Section 2, while the downstream policy performance is evaluated empirically in Section 6.

Fixed-rate benchmark. Let $\mathbf{N}^{CERO}$ be the executed allocation and define

$$\mathsf{ALG}_K^{\mathrm{stat}}(\boldsymbol{q}) := \sum_{i=1}^{M} U(q_i, z_{i,K}^{CERO}), \qquad \mathrm{Reg}_K^{\mathrm{stat}} := \mathsf{OPT}_K^{\mathrm{stat}}(\boldsymbol{q}) - \mathsf{ALG}_K^{\mathrm{stat}}(\boldsymbol{q}).$$

Let $\bar{N} = \frac{B}{KM} = \frac{R}{K}$ denote the average per-prompt per-round budget. Let $\widehat{q}_{i,k+1}$ be the post-feedback proxy used in round-k prompt-specific dual update and define the average absolute proxy discrepancy

$$\bar{e}_K^{\mathrm{stat}} := \frac{1}{MK} \sum_{k=1}^{K} \sum_{i=1}^{M} |\widehat{q}_{i,k+1} - q_i|.$$

Theorem 1 (Fixed-rate pathwise guarantee)**.** *Assume eq.* (2)*. For any fixed target vector* $\boldsymbol{q} \in [0, 1/4]^M$ *and any realized proxy sequence* $\{\hat{q}_{i,k+1}\}$*, Algorithm 1 with* $\gamma_\vartheta = \gamma_\nu = \frac{1}{\sqrt{K}}$ *satisfies*

$$\frac{\mathrm{Reg}_K^{\mathrm{stat}}}{M} \le \min\left\{1,\ \lambda \frac{N_{\max} + G_B/M}{\bar{N}} \frac{1}{\sqrt{K}} + 8\lambda \frac{N_{\max}}{\bar{N}} \bar{e}_K^{\mathrm{stat}}\right\}. \tag{11}$$

This pathwise bound separates online optimization error from reward-proxy discrepancy. In particular, if $N_{\max}/\bar{N}$, $G_B/(M\bar{N})$, and λ remain bounded and $\bar{e}_K^{\mathrm{stat}} = O(K^{-1/2})$, then the regret per prompt vanishes at rate $O(K^{-1/2})$. The proof is given in Appendix D.

Time-varying rates and a same-path benchmark. For a realized rate path $\boldsymbol{q}_{1:K}$, define the value of round-k exposure using the current rate:

$$\Delta U_{i,k}(\mathbf{N}; q_{i,k}) := U(q_{i,k}, z_{i,k}) - U(q_{i,k}, z_{i,k-1}).$$

Using the same $q_{i,k}$ in both terms ensures that $\Delta U_{i,k}$ reflects only the utility gained from the new exposure $N_{i,k}$, rather than changes due to variation in the rate itself. Writing $\boldsymbol{q}_{1:K} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_K)$, the same-path offline benchmark evaluates every feasible allocation plan against the same realized rate path:

$$\mathsf{OPT}_K^{\mathrm{dyn}}(\boldsymbol{q}_{1:K}) := \max_{\mathbf{N} \in \mathcal{N}_{B,C}} \sum_{k=1}^{K} \sum_{i=1}^{M} \Delta U_{i,k}(\mathbf{N}; q_{i,k}).$$

Define $\mathsf{ALG}_K^{\mathrm{dyn}}$ by evaluating the executed CERO allocation in the same way, and let $\mathrm{Reg}_K^{\mathrm{dyn}} = \mathsf{OPT}_K^{\mathrm{dyn}} - \mathsf{ALG}_K^{\mathrm{dyn}}$. By holding the realized rate path fixed across allocation plans, the benchmark focuses specifically on the quality of allocation decisions along a common realized rate path. Set

$$\bar{e}_K^{\mathrm{dyn}} := \frac{1}{MK} \sum_{k=1}^{K} \sum_{i=1}^{M} |\widehat{q}_{i,k+1} - q_{i,k}|, \; V_K := \sum_{k=1}^{K-1} \|\boldsymbol{q}_{k+1} - \boldsymbol{q}_k\|_\infty.$$

Theorem 2 (Same-path drifting-rate guarantee)**.** *Under the same algorithmic setup and step sizes as in Theorem 1, for every realized rate path $\boldsymbol{q}_{1:K}$ and proxy sequence $\{\widehat{q}_{i,k+1}\}$, Algorithm 1 satisfies*

$$\frac{\mathrm{Reg}_K^{\mathrm{dyn}}}{M} \le \min\left\{\lambda,\; \lambda \frac{N_{\max} + G_B/M}{\bar{N}} \frac{1}{\sqrt{K}} + 8\lambda \frac{N_{\max}}{\bar{N}} \bar{e}_K^{\mathrm{dyn}} + 4\lambda \left(1 + \frac{N_{\max}}{\bar{N}}\right) V_K \right\}. \quad (12)$$

Comparing with Theorem 1, the additional V_K term explicitly isolates the effect of temporal drift from reward-proxy discrepancy. The proof is given in Appendix E.

6 Experiments

6.1 Main Results

Experimental setup. We evaluate DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025), Qwen3-4B-Base (Yang et al., 2025), and Qwen2.5-7B-Instruct (Qwen et al., 2025), training all methods on the same uniformly sampled 50% subset of DAPO-Math-17K. Each run spans $K = 500$ scheduler rounds with a training-response budget of $B^{\mathrm{resp}} = 256{,}000$, and CERO uses eight responses per admitted prompt, with adaptation occurring through prompt admission and per-round group counts. We evaluate on AIME, AMC, MATH, MINERVA, and OlympiadBench using 16 responses per problem. We report avg@16 as the mean binary reward across responses and problems within each benchmark, and macro-average the five benchmark scores. Implementation details and compute measurements are provided in Appendices G and B.4, respectively.

Baselines. We evaluate four methods: (i) **GRPO** (Guo et al., 2025), a widely used reinforcement learning method for training reasoning models; (ii) **KnapsackRL** (Li et al., 2025), a representative prior method for adaptive rollout allocation, which dynamically allocates different numbers of rollouts across prompts; (iii) **CERO**, our proposed method; and (iv) **Random-matched CERO**, an ablation that preserves the rollout allocation produced by CERO while assigning it to a random permutation of prompts. The latter controls for the allocation pattern and isolates the effect of our prompt selection mechanism.

Table 1 shows that CERO attains avg@16 macro-averages of 32.81, 45.36, and 41.81, improving over GRPO by 2.62, 2.10, and 1.59 points, respectively. It also achieves a higher macro-average than the adaptive KnapsackRL baseline on all three backbones, with the clearest improvement on DSR1-Q1.5B and smaller margins on the larger models. The Random-matched control further isolates the contribution of prompt-aware assignment. Its macro-average falls below CERO by 3.63, 1.50, and 1.63 points across the three backbones. Since Random-matched CERO preserves allocation sizes and timing while randomizing prompt assignment, this comparison supports the contribution of prompt-aware assignment under the same rollout schedule.

6.2 Ablation and Mechanism Analysis

Is the reward-variation proxy informative? We examine whether the proxy anticipates the reward contrast available in future rollout groups. On the DSR1-Q1.5B backbone, we pair each pre-decision prediction $\hat{q}_{i,k}$ with the unbiased sample variance of the eight binary rewards generated afterward, covering $32,000$ selected prompt–round groups. Across ten equal-budget intervals, higher predicted means tend to coincide with greater observed reward variation (Pearson $r = 0.769$; Figure 2), showing that the proxy tracks changes in reward contrast during training. Across seven prediction bins, higher-prediction bins generally exhibit larger observed mean variance, with closely preserved relative ordering (Pearson $r = 0.971$, Spearman $\rho = 0.964$; Figure 6). Together, these analyses provide evidence that the proxy offers an informative pre-decision signal for subsequent reward contrast among selected prompt–round groups. Appendix B.2 further examines absolute calibration.

Table 1: Evaluation performance (avg@16) comparison across different models and benchmarks.

Base Model	Method	AIME	AMC	MATH	MINERVA	Olympiad	Avg.
DSR1-Q1.5B	GRPO	5.62	32.23	66.12	21.90	25.07	30.19
	KnapsackRL	4.58	32.38	65.14	20.93	24.59	29.53
	CERO (ours)	**7.92**	**38.03**	**68.05**	**22.66**	**27.41**	**32.81**
	Random-matched CERO	4.17	31.02	64.54	21.48	24.69	29.18
Qwen3-4B	GRPO	13.54	50.98	80.10	30.72	40.94	43.26
	KnapsackRL	**17.50**	53.24	80.08	31.36	41.26	44.69
	CERO (ours)	17.29	**53.69**	**80.83**	31.94	**43.06**	**45.36**
	Random-matched CERO	**17.50**	50.00	79.13	**32.84**	39.82	43.86
Q2.5-7B-Inst	GRPO	12.92	48.42	74.36	28.06	37.34	40.22
	KnapsackRL	13.33	**50.90**	77.15	28.72	**38.64**	41.75
	CERO (ours)	**15.83**	49.77	**77.44**	27.55	38.45	**41.81**
	Random-matched CERO	11.88	46.23	76.35	**29.30**	37.17	40.18



Figure 2: Temporal tracking of CERO's reward-variation proxy on DSR1-Q1.5B.

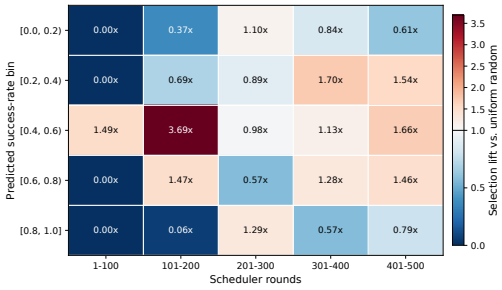

Figure 3: Prompt-selection lift by pre-decision predicted success rate.

Does CERO prioritize informative prompts? We characterize CERO's selection pattern by partitioning the 500 scheduler rounds into five 100-round stages and binning candidate prompt–round pairs by their pre-decision predicted success rate. For bin b in stage t, we define the selection lift as

$$L_{b,t} = \frac{n^{\text{sel}}_{b,t}/n^{\text{cand}}_{b,t}}{n^{\text{sel}}_{t}/n^{\text{cand}}_{t}},$$

where n^{sel} and n^{cand} denote the numbers of selected and candidate prompt–round pairs. Consequently, $L_{b,t} = 1$ corresponds to uniform random selection, while $L_{b,t} > 1$ and $L_{b,t} < 1$ indicate over-selection and under-selection respectively. Figure 3 shows that CERO preferentially selects prompts with intermediate predicted success rates. Over the full run, the $[0.4, 0.6)$ bin has a $1.72\times$ selection lift, while both extreme bins have approximately $0.65\times$. With the full prompt pool available in every round, the lift directly measures CERO's selection preference across predicted-success ranges relative to uniform selection. This pattern aligns with CERO's reward-variation-based design: prompts for which both success and failure remain plausible offer opportunities for the within-group reward contrast needed by group-relative learning. The strength and breadth of the preference vary across training stages, with all three interior bins over-selected in the final stage.

Does prompt-aware admission increase the fraction of effective groups? Let $S_{i,k}$ of $G_{i,k}$ responses in group (i,k) be correct. A group is effective if $0 < S_{i,k} < G_{i,k}$. Define the following effective-group rate

$$\text{EGR} := \frac{\sum_{i,k} \mathbf{1}\{0 < S_{i,k} < G_{i,k}\}}{\sum_{i,k} \mathbf{1}\{G_{i,k} > 0\}},$$

the fraction of executed groups supplying nonzero reward contrast. Figure 4 shows that under the matched budget of $256,000$ training responses, CERO produces $18,347$ effective groups out of $32,000$ executed groups, yielding an EGR of 57.3%, compared with 52.7% for Random-matched CERO and 48.7% for KnapsackRL. Because Random-matched CERO preserves CERO's group

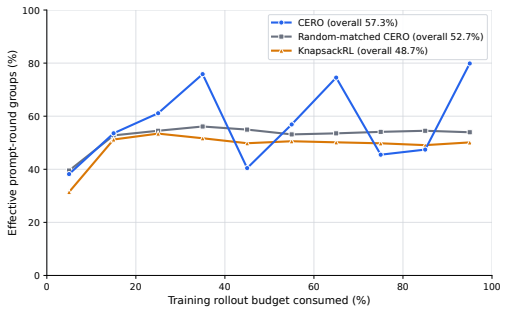

Figure 4: Effective-group rate over consumed rollout budget.

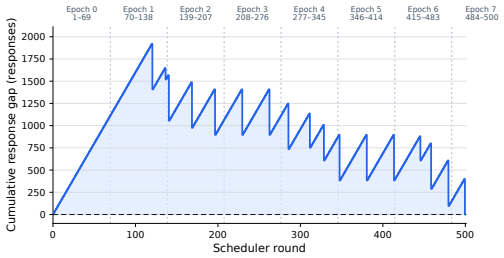

Figure 5: Cross-epoch budget pacing in CERO. Epoch boundaries are marked above trajectory.

sizes and spending schedule while randomizing prompt assignment, the 4.6-point gap provides controlled evidence that prompt-aware admission increases the fraction of mixed-outcome groups under the same rollout schedule.

What do dual-based selection and adaptive pacing contribute? We first characterize CERO's budget pacing through the cumulative response gap from a uniform spending schedule:

$$G_k^{\text{exec}} := \sum_{\tau=1}^{k} \sum_{i} N_{i,\tau}^{\text{resp}} - k\bar{B}^{\text{resp}}, \qquad \bar{B}^{\text{resp}} := \frac{B^{\text{resp}}}{K} = 512.$$

Positive and negative gaps indicate spending ahead of and behind the uniform schedule, respectively. Figure 5 shows that spending deviations persist across data-epoch boundaries. The cumulative surplus peaks at 1,920 responses and returns to zero at the end of training, exhausting the 256,000-response budget exactly. This trajectory characterizes CERO's nonuniform budget consumption.

Table 2 evaluates the contributions of selection and pacing. CERO-fixed retains CERO's reward-proxy and prompt-dual updates but fixes $\nu = 0$ and allocates 512 responses per round. CERO-preset follows a predetermined front-loaded schedule: 528 responses per round for the first 250 rounds and 496 for the remaining 250 rounds. CERO-greedy retains CERO-fixed's proxy estimator, utility function, and per-round quota, but selects prompts by the marginal utility of an additional complete group rather than by prompt-dual scores.

Across three training seeds, CERO achieves the highest macro-average of 32.26 ± 0.94, outperforming CERO-fixed, CERO-preset, and CERO-greedy by 1.13, 1.55, and 1.29 points on average, respectively. The gains over CERO-fixed and CERO-preset support the value of adapting budget consumption over the training horizon rather than using a fixed or predetermined spending schedule. Under the fixed 512-response quota, dual-guided and direct marginal-utility ranking yield similar mean scores of 31.13 and 30.97, whereas full CERO improves on both by more than one point. Together, these results support CERO's central design: jointly coordinating prompt prioritization and budget pacing over the training horizon.

Table 2: Prompt-selection and pacing ablations on DSR1-Q1.5B. Avg. denotes the five-benchmark avg@16 macro-average (mean $\pm$ std over 3 seeds).

Method	Avg.
CERO	**32.26 $\pm$ 0.94**
CERO-fixed	31.13 $\pm$ 0.72
CERO-greedy	30.97 $\pm$ 0.38
CERO-preset	30.71 $\pm$ 0.54

7 Conclusion

We introduced CERO, a finite-horizon scheduler that coordinates prompt admission, repeated selection, and budget pacing under a shared rollout budget. The Fenchel-based primal–dual formulation yields tractable round-level decisions and pathwise guarantees for the surrogate allocation objective under fixed-rate and same-path time-varying benchmarks. Under matched training-response budgets, CERO achieves the highest avg@16 macro-average on all three backbones. Mechanistic analyses link its prompt choices to stronger within-group supervision, while multi-seed ablations show gains from adaptive pacing over fixed and preset schedules. Together, these results demonstrate the practical value of horizon-level rollout coordination for efficient RL post-training.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, LaTeX restructuring, and code-level consistency checks. All AI-assisted material, including mathematical notation, citations, and experimental descriptions, was reviewed by the authors. The authors take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

The proofs of the theoretical results are provided in Appendices C.1– F. Appendix G documents the training and evaluation protocols, hardware and software environment, and allocation settings. Additional evaluation results, reward-proxy diagnostics, audits of the realized schedules, and computational-cost measurements are provided in Appendix B. We have also uploaded all experiment code in the supplementary materials.

REFERENCES

Shipra Agrawal and Nikhil R Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1405–1424. SIAM, 2014.

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Dual mirror descent for online allocation problems. In *International Conference on Machine Learning*, pp. 613–628. PMLR, 2020.

Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing problems. In *European Symposium on Algorithms*, pp. 689–701. Springer, 2005.

Niv Buchbinder and Joseph Naor. Improved bounds for online routing and packing via a primal-dual approach. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pp. 293–304. IEEE, 2006.

Xiao Alison Chen and Zizhuo Wang. A dynamic learning algorithm for online matching problems with concave returns. *European Journal of Operational Research*, 247(2):379–388, 2015.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*, 2025.

Nikhil R Devanur and Thomas P Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 71–78, 2009.

Nikhil R Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 137–144, 2012.

Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*, pp. 182–194. Springer, 2010.

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

Zengjie Hu, Jiantao Qiu, Tianyi Bai, Haojin Yang, Binhang Yuan, Qi Jing, Conghui He, and Wentao Zhang. Vade: Variance-aware dynamic sampling via online sample-level difficulty estimation for multimodal rl. *arXiv preprint arXiv:2511.18902*, 2025.

Stefanus Jasin and Sunil Kumar. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research*, 37(2):313–345, 2012.

Heyang Jiang, Henry Liu, and Baharan Mirzasoleiman. Learning as reasoning unfolds: Progressive rollout allocation for efficient reinforcement learning. *arXiv preprint arXiv:2607.22002*, 2026.

Jiashuo Jiang, Will Ma, and Jiawei Zhang. Degeneracy is ok: Logarithmic regret for network revenue management with indiscrete distributions. *Operations Research*, 73(6):3405–3420, 2025.

Thomas Kesselheim, Andreas Tönnis, Klaus Radke, and Berthold Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pp. 303–312, 2014.

Woojeong Kim, Ziyi Yang, Jing Nathan Yan, and Jialu Liu. Spend your rollouts where it counts: Rollout allocation for group-based rl post-training. *arXiv preprint arXiv:2605.26606*, 2026.

Xiaocheng Li and Yinyu Ye. Online linear programming: Dual convergence, new algorithms, and regret bounds. *Operations Research*, 70(5):2948–2966, 2022.

Ziniu Li, Congliang Chen, Tianyun Yang, Tian Ding, Ruoyu Sun, Ge Zhang, Wenhao Huang, and Zhi-Quan Luo. Knapsack rl: Unlocking exploration of llms via optimizing budget allocation. *arXiv preprint arXiv:2509.25849*, 2025.

Mengqi Liao, Xiangyu Xi, Chen Ruinian, Jia Leng, Yangen Hu, Ke Zeng, Shuai Liu, and Huaiyu Wan. Enhancing efficiency and exploration in reinforcement learning for LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1451–1463, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.75. URL https://aclanthology.org/2025.emnlp-main.75/.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.

Ilia Mahrooghi, Aryo Lotfi, and Emmanuel Abbe. Goldilocks rl: Tuning task difficulty to escape sparse rewards for reasoning. *arXiv preprint arXiv:2602.14868*, 2026.

Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es, 2007.

Marco Molinaro and Ramamoorthi Ravi. The geometry of online packing linear programs. *Mathematics of Operations Research*, 39(1):46–59, 2014.

Trung Hieu Nguyen, Bao Nguyen, Wenao Ma, Yuzhi Zhao, Ruifeng She, and Viet Anh Nguyen. Adaptive rollout allocation for online reinforcement learning with verifiable rewards. In *International Conference on Learning Representations*, volume 2026, pp. 136100–136132, 2026.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

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

Kalyan T Talluri and Garrett J Van Ryzin. *The theory and practice of revenue management*, volume 68. Springer Science & Business Media, 2006.

Sijun Tan, Michael Luo, Justin Wong, Colin Cai, Xiaoxiang Shi, William Yuan Tang, Manan Roongta, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, et al. Deepscaler: Effective rl scaling of reasoning models via iterative context lengthening. 2025.

Tao Wang, Shuo Li, Yan Sun, Dongsheng Ding, and Edgar Dobriban. Where to spend rollouts: Hit-utility optimal rollout allocation for group-based rlvr. *arXiv preprint arXiv:2605.07114*, 2026.

Yixuan Even Xu, Yash Savani, Fei Fang, and J Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.

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.

Zhiyuan Yao, Yi-Kai Zhang, Yuxin Chen, Yueqing Sun, Zishan Xu, Yu Yang, Tianhao Hu, Qi Gu, Hui Su, and Xunliang Cai. Coba-rl: Capability-oriented budget allocation for reinforcement learning in llms. *arXiv preprint arXiv:2602.03048*, 2026.

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

Ruiqi Zhang, Daman Arora, Song Mei, and Andrea Zanette. Speed-rl: Faster training of reasoning models via online curriculum learning. *arXiv preprint arXiv:2506.09016*, 2025a.

Yuheng Zhang, Wenlin Yao, Changlong Yu, Yao Liu, Qingyu Yin, Bing Yin, Hyokun Yun, and Lihong Li. Improving sampling efficiency in rlvr through adaptive rollout and response reuse. *arXiv preprint arXiv:2509.25808*, 2025b.

Zhi Zhang, Zhen Han, Costas Mavromatis, Qi Zhu, Yunyi Zhang, Sheng Guan, Dingmin Wang, Xiong Zhou, Shuai Wang, Soji Adeshina, et al. Train less, learn more: Adaptive efficient rollout optimization for group-based reinforcement learning. *arXiv preprint arXiv:2602.14338*, 2026.

A Related Work

Difficulty-aware sampling and filtering. Recent work has explored improving RL efficiency by prioritizing training examples and rollouts according to their difficulty or informativeness. At the query level, online curriculum and sample-selection methods dynamically adjust the training distribution to favor examples with greater learning value. Goldilocks RL (Mahrooghi et al., 2026) emphasizes problems of intermediate difficulty, while SPEED-RL (Zhang et al., 2025a), SEC (Chen et al., 2025), and VADE (Hu et al., 2025) similarly use online estimates of difficulty or training utility to prioritize informative samples. Related approaches operate after rollout generation by filtering or pruning trajectories with limited contribution to policy updates. CPPO (Lin et al., 2025) removes completions with low learning signals, PODS (Xu et al., 2025) selects informative subsets of generated responses, and GFPO (Shrivastava et al., 2025) filters rollouts based on their utility relative to generation cost. These methods collectively reduce computation spent on overly easy, overly difficult, redundant, or otherwise uninformative samples.

Adaptive rollout allocation. Recent studies have increasingly treated rollout generation as a resource allocation problem, replacing the fixed group size in standard GRPO (Shao et al., 2024) with query-dependent rollout budgets. Some methods adapt allocation according to estimated difficulty, assigning more rollouts to challenging or uncertain queries and fewer to those with saturated outcomes (Liao et al., 2025; Zhang et al., 2025b). Others formulate allocation through explicit estimates of training value, uncertainty, or marginal utility, including Knapsack RL (Li et al., 2025), CoBA-RL (Yao et al., 2026) and VIP (Nguyen et al., 2026). A related direction makes allocation decisions progressively from intermediate rollout outcomes: Pilot-Commit (Kim et al., 2026), AERO (Zhang et al., 2026), and VIGOR (Jiang et al., 2026) use early samples to determine whether additional generation is worthwhile. Collectively, these methods improve rollout efficiency by directing computation toward queries where additional exploration is expected to provide the greatest learning value.

Online resource allocation. Online resource allocation is a central topic in the online optimization literature and has been applied in many operational settings, including ad assignment (Mehta et al., 2007), revenue management (Talluri & Van Ryzin, 2006; Jasin & Kumar, 2012; Jiang et al., 2025), and online packing/covering problems (Buchbinder & Naor, 2005; 2006; Feldman et al., 2010). The existing algorithmic literature can be broadly organized according to how it exploits dual information. A dominant class of methods uses dual prices as signals for making online allocation decisions. Within this class, some papers adopt a learn-once strategy: they reserve an initial set of arrivals for estimation, compute a price vector from that sample, and then use the same prices for all future decisions (Devanur & Hayes, 2009; Feldman et al., 2010; Molinaro & Ravi, 2014; Devanur & Jain, 2012). Other papers advocate adaptive updating, where the pricing rule is revised at multiple points over time through repeated re-solving, leading to stronger robustness and improved horizon dependence (Agrawal et al., 2014; Li & Ye, 2022; Chen & Wang, 2015). In parallel, other works depart from this paradigm by developing primal-oriented algorithms (Kesselheim et al., 2014) or first-order schemes that do not require repeated computation of dual prices (Agrawal & Devanur, 2014; Balseiro et al., 2020).

B Additional Results and Diagnostics

B.1 Pass@16 Evaluation

In addition to the avg@16 results reported in the main text, we evaluate pass@16 performance to measure success coverage under repeated sampling. Pass@16 is the percentage of problems with at least one correct response among these 16 samples:

$$\text{pass@16} = \frac{100}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbf{1} \left\{ \sum_{j=1}^{16} r_{x,j} > 0 \right\}.$$

For binary rewards, this is equivalent to averaging the maximum reward among the 16 responses for each problem. Table 3 shows that CERO improves the pass@16 macro-average over GRPO on all

Table 3: Evaluation performance (pass@16) comparison across different models and benchmarks.

Base Model	Method	AIME	AMC	MATH	MINERVA	Olympiad	Avg.
DSR1-Q1.5B	GRPO	**20.00**	55.42	83.40	**46.69**	40.89	49.28
	KnapsackRL	**20.00**	56.63	82.80	42.65	40.59	48.53
	CERO (ours)	**20.00**	**62.65**	**84.20**	44.85	**43.26**	**50.99**
	Random-matched CERO	13.33	51.81	81.40	45.96	39.70	46.44
Qwen3-4B	GRPO	40.00	79.52	92.00	51.10	62.67	65.06
	KnapsackRL	**46.67**	80.72	**92.20**	51.84	**64.00**	**67.09**
	CERO (ours)	40.00	**83.13**	**92.20**	**52.21**	59.70	65.45
	Random-matched CERO	43.33	73.49	92.00	51.10	61.04	64.19
Q2.5-7B-Inst	GRPO	33.33	73.49	91.40	47.79	57.78	60.76
	KnapsackRL	30.00	74.70	**92.00**	**48.53**	**60.30**	61.10
	CERO (ours)	**36.67**	**78.31**	89.40	**48.53**	58.81	**62.34**
	Random-matched CERO	30.00	71.08	88.80	46.69	56.74	58.66

three backbones, with gains of 1.71, 0.39, and 1.58 points. It achieves the highest macro-average on DeepSeek-R1-Distill-Qwen-1.5B and Qwen2.5-7B-Instruct, reaching 50.99 and 62.34, respectively. On Qwen3-4B, KnapsackRL achieves the highest score of 67.09, compared with 65.45 for CERO. Together with the avg@16 results, these findings show improvements over GRPO in both average response accuracy and repeated-sampling success coverage.

B.2 Reliability of the Reward-Variation Proxy

We examine whether CERO's pre-decision reward-variation proxy captures differences in subsequently observed group variance, both across prediction levels and over the training trajectory.

Diagnostic setup. We analyze 32,000 executed prompt–round groups from the DSR1-Q1.5B run. For each group, we pair the pre-decision prediction $\hat{q}_{i,k}$ with the unbiased sample variance $v_{i,k} = \widehat{\mathrm{Var}}_8(r_{i,k})$ of the eight binary rewards generated afterward. Figure 6 aggregates these pairs into seven nonempty prediction bins and plots the mean prediction $\bar{q}_\ell$ against the mean observed variance $\bar{v}_\ell$. Vertical error bars show nominal 95% normal-approximation intervals,

$$\bar{v}_\ell \pm 1.96 \, \frac{s_\ell}{\sqrt{n_\ell}},$$

where n_ℓ is the number of groups in bin ℓ and s_ℓ is the sample standard deviation of their observed group variances. These intervals use group-level standard errors within a single run, without adjustment for repeated-prompt or temporal dependence.

Association across prediction levels. Figure 6 shows a strong positive association between predicted and observed bin means, with Pearson correlation $r = 0.971$ and Spearman correlation $\rho = 0.964$. Across the plotted prediction range, the observed mean group variance rises overall from approximately 0.03 to 0.20. Higher-prediction bins generally correspond to greater realized reward variation, and the ordering of the bin means is closely preserved despite a local fluctuation in the intermediate range. The proxy therefore distinguishes prediction ranges associated with substantially different levels of subsequent reward contrast. Because the predictions are recorded before the current-round responses are generated, this association reflects information available when CERO makes its allocation decisions.

Absolute calibration. The predicted values also exhibit an upward offset in numerical scale. Across the selected groups, the mean predicted variance is 0.154, compared with a mean observed variance of 0.119. Consistent with this difference, the bin means in Figure 6 lie below the diagonal marking equality between predictions and observations. The strong correlations and the displacement from the diagonal characterize two distinct properties: the proxy captures differences across prediction levels while tending to overestimate absolute reward variance. This distinction matters for CERO because $\hat{q}_{i,k}$ enters the surrogate utility and prompt-dual updates numerically, rather than serving only as a ranking statistic. The diagnostic thus identifies both the information carried by the proxy and the calibration offset present in the analyzed run.

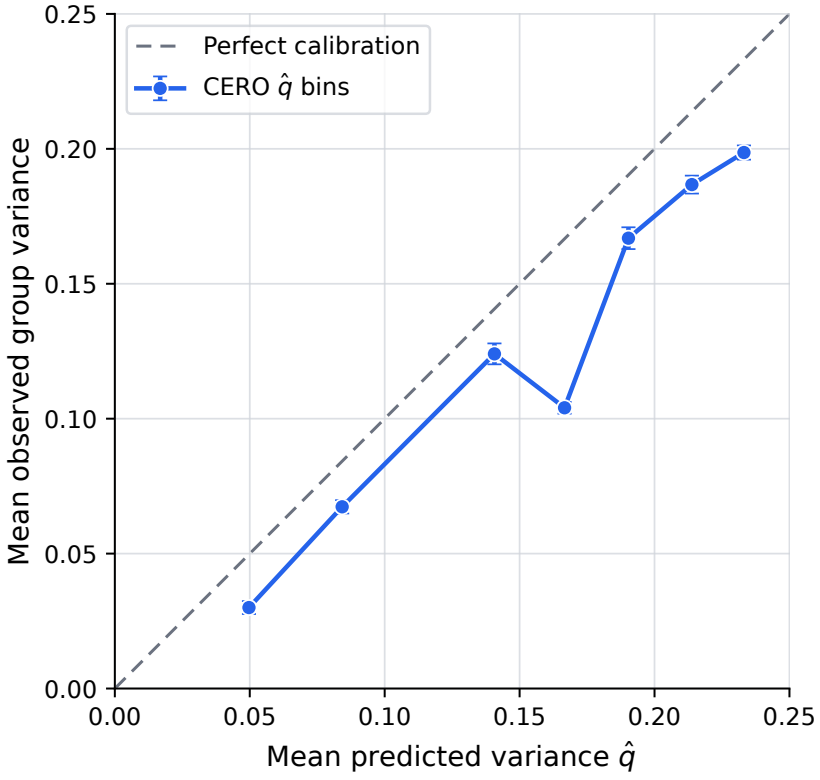

Figure 6: Calibration of CERO's reward-variation proxy on DSR1-Q1.5B. Points show predicted and observed bin means across seven prediction bins; vertical bars indicate nominal 95% within-run intervals for the observed means. The dashed diagonal marks perfect calibration.

Temporal tracking and allocation relevance. Figure 2 provides a complementary view by aggregating the same trajectory into ten consecutive equal-budget intervals of 3,200 groups each. Predicted and observed interval means co-vary over training, with Pearson correlation $r = 0.769$. Their trajectories capture common decreases and rebounds, showing that the proxy reflects changes in reward variation as training progresses, in addition to the differences across prediction levels shown in Figure 6. This behavior is relevant to group-relative supervision: mixed-outcome groups provide positive reward variance, whereas all-correct and all-incorrect groups provide no within-group reward contrast. Together, the prediction-bin and temporal analyses support $\hat{q}_{i,k}$ as an informative prospective signal for CERO's allocation mechanism. They complement the selection-lift and effective-group analyses in Section 6.2, which examine how the resulting allocation translates that signal into prompt choices and usable supervision.

B.3 Cross-Round Prompt Exposure

To examine how CERO redistributes repeated exposure over the training horizon, we compare terminal cumulative prompt exposure with GRPO. Following Section 2, we normalize each prompt's terminal exposure by the equal-share scale $R = B/M$. In Figure 7, the empirical CDF at x gives the fraction of prompts with $z_{i,K}/R \leq x$. Because both methods use fixed-size response groups and the same total rollout budget, differences in terminal exposure reflect how many times each prompt is selected over the training horizon rather than differences in group size or aggregate exposure.

Figure 7 shows that CERO induces a broader terminal-exposure distribution than GRPO. CERO spreads terminal exposure across three levels, with 57.5% of prompts at $0.84R$, 27.8% at $1.12R$, and a 14.7% high-exposure tail at $1.40R$. GRPO is more concentrated, with 41.0% and 58.1% of prompts at $0.84R$ and $1.12R$, respectively. Thus, CERO shifts exposure away from the more concentrated GRPO pattern: a larger fraction of prompts receive below-equal-share exposure, while a smaller subset is revisited sufficiently often to reach substantially above the equal-share level. Since

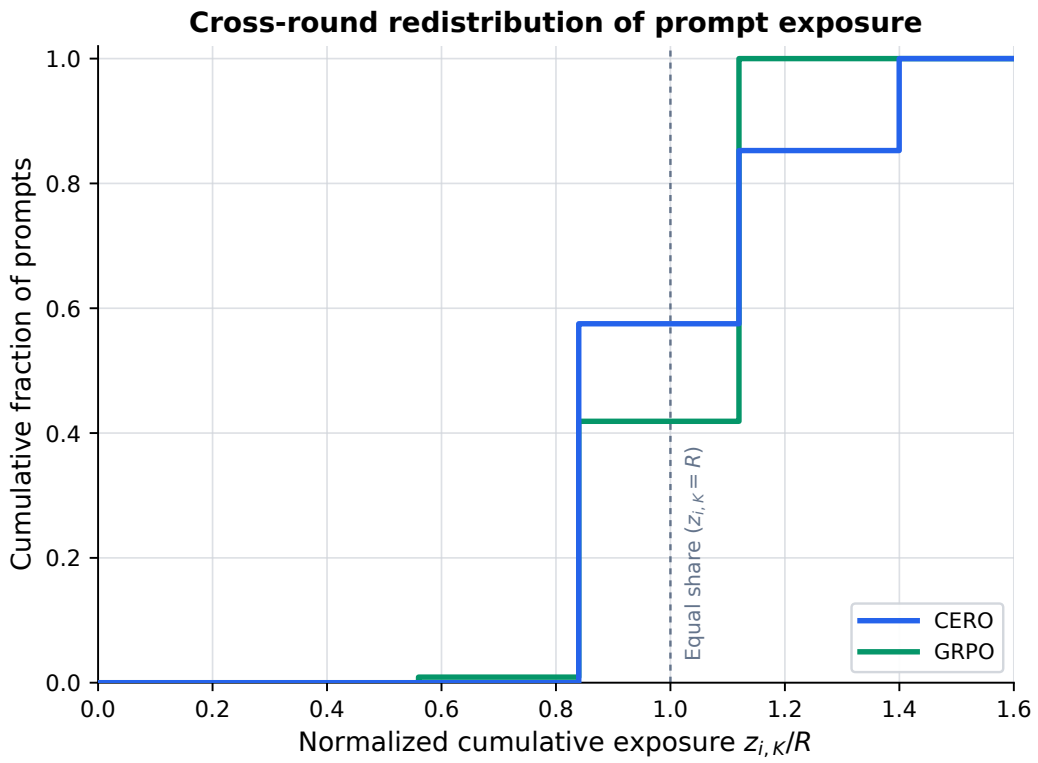

Figure 7: Empirical CDF of normalized terminal prompt exposure $z_{i,K}/R$ for CERO and GRPO on DSR1-Q1.5B. The dashed line marks the equal-share exposure $R = B/M$.

both methods use the same total rollout budget, this broader distribution reflects a redistribution of repeated selections across prompts rather than additional aggregate exposure. In particular, CERO trades fewer revisits to some prompts for more persistent revisiting of others over the training horizon.

B.4 Computational Cost

We compare the training cost of CERO, Random-matched CERO, KnapsackRL, and vanilla GRPO. For every backbone, all methods use the same number of GPUs, 500 scheduler rounds, and a matched budget of 256,000 training responses. End-to-end GPU-hours are calculated as the Slurm elapsed time multiplied by the number of allocated GPUs. This measurement includes initialization, in-training validation, and checkpointing.

Figure 8 provides a detailed comparison for the 1.5B backbone. In addition to end-to-end GPU-hours, we report core training time normalized by the number of generated response tokens. The normalized measurement excludes validation and checkpointing but includes rollout generation, reward computation, log-probability computation, policy optimization, and allocation-related operations.

On the 1.5B backbone, CERO, Random-matched CERO, KnapsackRL, and vanilla GRPO consume 16.94, 17.15, 17.53, and 17.57 GPU-hours, respectively. Their token-normalized core costs are 84.2, 84.5, 83.6, and 84.2 seconds per million generated response tokens. The normalized measurements differ by only approximately 1.1% of their mean value, with CERO being nearly identical to vanilla GRPO. CERO therefore combines adaptive allocation with token-normalized core training cost comparable to the baselines in the 1.5B experiment.

We also measure end-to-end GPU-hours for the larger backbones. On Qwen3-4B-Base, CERO uses 36.31 GPU-hours, Random-matched CERO uses 36.78 GPU-hours, KnapsackRL uses 35.85 GPU-hours, and vanilla GRPO uses 36.89 GPU-hours. These values correspond to wall-clock times of

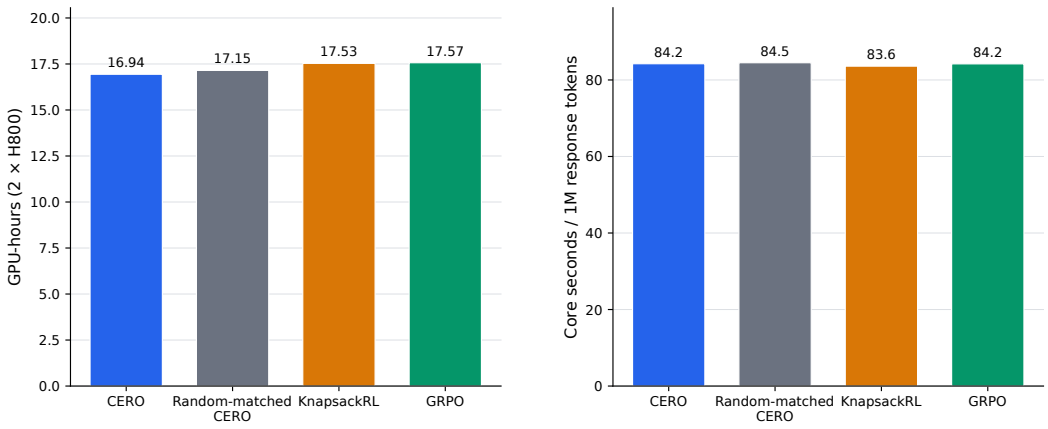

Figure 8: Training-compute comparison on DeepSeek-R1-Distill-Qwen-1.5B under a matched response budget. The left panel reports end-to-end GPU-hours, while the right panel reports core training time normalized by generated response tokens.

9.08, 9.19, 8.96, and 9.22 hours, respectively, using four H800 GPUs. The differences among the four methods are within approximately 3% of their mean compute cost.

On Qwen2.5-7B-Instruct, CERO, Random-matched CERO, KnapsackRL, and vanilla GRPO use 31.46, 32.78, 34.99, and 31.72 GPU-hours, respectively. Their corresponding wall-clock times are 7.86, 8.19, 8.75, and 7.93 hours on four H800 GPUs. CERO therefore remains comparable to the baselines in end-to-end computational cost as the backbone size increases.

The end-to-end measurements capture the practical cost of each complete training protocol under the matched response budget, including differences in response lengths and update counts. CERO and Random-matched CERO perform 488, 486, and 487 policy updates on the 1.5B, 4B, and 7B backbones, respectively, while GRPO and KnapsackRL perform 500. Zero-allocation rounds account for the lower update counts.

Each cost measurement is obtained from one completed run per method and backbone. Across the reported runs, CERO combines adaptive prompt selection and cross-epoch budget pacing with comparable end-to-end training cost. On the 1.5B backbone, its token-normalized core cost matches GRPO at 84.2 seconds per million generated response tokens.

C Utility Properties and Full-Information Oracle

C.1 Exact Conjugate, Fenchel Representation, and Stability

Fix $q \in [0, 1/4]$, set $c = 4\lambda q/R$, and write $U_c(z) = 1 - e^{-cz}$ on $[0, Z]$. Define

$$U^*_{c,Z}(\theta) := \inf_{0 \le z \le Z} \{\theta z - U_c(z)\}. \tag{13}$$

Lemma 1 (Closed-form compact conjugate)**.** *If $c = 0$, then $U^*_{c,Z}(\theta) = 0$ for every $\theta \ge 0$. If $c > 0$, then*

$$U^*_{c,Z}(\theta) = \begin{cases} \theta Z - 1 + e^{-cZ}, & 0 \le \theta \le ce^{-cZ}, \\ \dfrac{\theta}{c} \log \dfrac{c}{\theta} + \dfrac{\theta}{c} - 1, & ce^{-cZ} < \theta < c, \\ 0, & \theta \ge c. \end{cases} \tag{14}$$

The minimizer is given by equation 9.

Proof. For $c > 0$, differentiate $\theta z - 1 + e^{-cz}$. Its derivative is $\theta - ce^{-cz}$ and is strictly increasing in z. If $\theta \le ce^{-cZ}$, the objective is nonincreasing on $[0, Z]$ and the minimum is at Z. If $ce^{-cZ} < \theta < c$, the unique root is $z = c^{-1} \log(c/\theta)$. If $\theta \ge c$, the minimum is at 0. Substitution yields equation 14. The case $c = 0$ is immediate. □

We also write $U_q^*(\theta) := U_{4\lambda q/R,Z}^*(\theta)$ for the compact conjugate parameterized by q; in particular, $U_i^*(\theta) = U_{q_i}^*(\theta)$ in Section 2.3.

Lemma 2 (Exact compact biconjugate and stability)**.** *For every $q \in [0, 1/4]$ and $z \in [0, Z]$,*

$$U(q, z) = \min_{0 \le \theta \le \bar{c}} \{\theta z - U_q^*(\theta)\}. \tag{15}$$

Moreover, for $q, q' \in [0, 1/4]$,

$$\sup_{0 \le z \le Z} |U(q, z) - U(q', z)| \le \frac{4\lambda Z}{R} |q - q'|, \tag{16}$$

$$\sup_{0 \le \theta \le \bar{c}} |U_q^*(\theta) - U_{q'}^*(\theta)| \le \frac{4\lambda Z}{R} |q - q'|. \tag{17}$$

Proof. The slope of $U(q, \cdot)$ lies in $[0, \bar{c}]$. For a fixed z, choose the supporting slope $\theta_z = \partial_z U(q, z)$, with one-sided derivatives at the endpoints. Concavity gives

$$U(q, s) \le U(q, z) + \theta_z (s - z), \tag{18}$$

so $U_q^*(\theta_z) = \theta_z z - U(q, z)$. The reverse inequality follows directly from the definition of the conjugate.

For stability,

$$|\partial_q U(q, z)| = \frac{4\lambda z}{R} e^{-4\lambda q z/R} \le \frac{4\lambda Z}{R}. \tag{19}$$

The first bound follows from the mean-value theorem. Applying a common uniform perturbation before taking the infimum over $z \in [0, Z]$ gives the conjugate bound. □

C.2 Full-Information Oracle

We characterize the fixed-rate full-information benchmark when the target rates $q = (q_1, \ldots, q_M)$ are known. Since the objective depends only on terminal prompt exposures, the oracle can first determine the optimal exposure vector and then distribute these exposures across the K rounds.

Terminal-exposure formulation. Writing $z_i = \sum_{k=1}^K N_{i,k}$ and $c_i = 4\lambda q_i / R$, the fixed-rate benchmark can be expressed as

$$\mathrm{OPT}_K^{\mathrm{stat}}(q) = \max_{\substack{z \in \mathbb{Z}_{\ge 0}^M \\ z_i \le Z,\ \sum_i z_i \le B}} \sum_{i=1}^M \left(1 - e^{-c_i z_i}\right). \tag{20}$$

Every feasible rollout plan induces an exposure vector satisfying these constraints. Conversely, every such integer vector admits a feasible round-level realization, as shown below. For the continuous relaxation, the uniform assignment $N_{i,k} = z_i / K$ is feasible because $z_i \le K N_{\max}$ and $\sum_i z_i \le B \le KC$.

Continuous water-filling solution. Relaxing the integrality constraint gives a separable concave optimization problem. Let $\xi \ge 0$ be the multiplier for the total budget. For $c_i > 0$, an interior optimum satisfies $c_i e^{-c_i z_i} = \xi$. Consequently, for $\xi > 0$, the optimal exposure has the clipped water-filling form

$$z_i^{\mathrm{cont}}(\xi) = \left[\frac{1}{c_i} \log \frac{c_i}{\xi}\right]_{[0,Z]}, \qquad c_i > 0, \tag{21}$$

where $[x]_{[0,Z]} = \min\{Z, \max\{0, x\}\}$.

Let $\mathcal{I}_+ = \{i : c_i > 0\}$. When $0 < B < Z|\mathcal{I}_+|$, the budget is binding and ξ can be chosen so that $\sum_{i \in \mathcal{I}_+} z_i^{\mathrm{cont}}(\xi) = B$. The left-hand side is continuous and nonincreasing in ξ, allowing the price to be found by one-dimensional bisection. Prompts with $c_i = 0$ contribute zero utility and may be assigned zero exposure. When $B \ge Z|\mathcal{I}_+|$, assigning Z units to every positive-rate prompt is optimal; any remaining budget may be left unused or assigned to zero-rate prompts without changing the objective.

Exact integer allocation. For integer exposures, the marginal utility of the ℓ-th allocation unit assigned to prompt i is

$$\Delta_i(\ell) := U(q_i, \ell) - U(q_i, \ell - 1) = (1 - e^{-c_i})e^{-c_i(\ell-1)}, \qquad \ell = 1, \dots, Z. \tag{22}$$

These marginal gains are nonnegative and nonincreasing in ℓ. Starting from $z_i = 0$ for every prompt, the oracle repeatedly assigns one unit to an unsaturated prompt with the largest currently available gain $\Delta_i(z_i + 1)$. Repeating this operation for B units produces an optimal integer exposure vector, allowing zero-gain assignments when necessary.

The optimality follows from the decreasing marginal-gain sequences. A set of the B largest gains can be chosen to contain a prefix of each prompt's sequence, with ties resolved in favor of earlier units. The greedy procedure constructs such a set while respecting $z_i \le Z$. A max-heap containing the next available marginal gain for each unsaturated prompt computes the exposure vector in $O(M + B \log(M + 1))$ time and $O(M)$ auxiliary memory.

Recovery of a feasible rollout plan. To realize any feasible integer exposure vector z, let $T = \sum_i z_i$ and list its T allocation units, placing the units for each prompt in one consecutive block. Assign the r-th unit in this list to round $1 + ((r - 1) \bmod K)$. Each prompt's units are thereby spread as evenly as possible across rounds, and the same is true of the total allocation. The resulting counts satisfy

$$\sum_{k=1}^{K} N_{i,k} = z_i, \qquad N_{i,k} \le \left\lceil \frac{z_i}{K} \right\rceil \le N_{\max}, \qquad \sum_i N_{i,k} \le \left\lceil \frac{T}{K} \right\rceil \le C. \tag{23}$$

The final inequalities use $z_i \le KN_{\max}$, $T \le B \le KC$, and the integrality of $N_{\max}$ and C. Thus, the exposure-level optimum is attainable under the original global-budget and per-round capacity constraints.

Full-capacity boundary. When $B = KMN_{\max}$ and $C = MN_{\max}$, assigning $N_{i,k} = N_{\max}$ to every prompt in every round is feasible and yields $z_{i,K} = Z$. Since each utility is nondecreasing in exposure, this allocation attains

$$\mathrm{OPT}_K^{\mathrm{stat}}(q) = \sum_{i=1}^{M} U(q_i, Z).$$

The full-allocation rule therefore achieves zero fixed-rate regret without estimating the target rates.

The oracle makes the diminishing-return structure explicit: additional units are assigned according to their remaining marginal utility rather than a prompt's initial rate alone. It characterizes the fixed-rate benchmark, while CERO uses reward-derived proxy states and online dual updates to allocate rollouts as feedback becomes available.

D Proof of the Fixed-Rate Guarantee

For the analysis, if the execution budget is exhausted before round K, we continue the virtual iterates through the remaining rounds under the same update rules, with all subsequent executed allocations set to zero.

For each round k, let $\widehat{U}_{i,k+1}$ be the post-feedback proxy utility and $\widehat{U}^*_{i,k+1}$ its compact conjugate. Let $\widetilde{\boldsymbol{N}}_k$ be the virtual proposal and $\widetilde{A}_k = \sum_i \widetilde{N}_{i,k}$. Work with the unnormalized duals $\theta_i^k = \bar{c}\vartheta_i^k$ and $\mu^k = \bar{c}\nu^k$. The normalized updates in equation 10 are equivalent to projected subgradient descent with

$$\alpha_\theta = \frac{\bar{c}}{N_{\max}\sqrt{K}}, \qquad \alpha_\mu = \frac{\bar{c}}{G_B\sqrt{K}}. \tag{24}$$

Define convex losses, after the round-k proposal and proxy utilities have been realized, by

$$h_k(\boldsymbol{\theta}) := \sum_{i=1}^{M} \left[\theta_i \widetilde{N}_{i,k} - \frac{1}{K} \widehat{U}^*_{i,k+1}(\theta_i) \right], \tag{25}$$

$$m_k(\mu) := \mu(\bar{B} - \widetilde{A}_k). \tag{26}$$

Any $\widehat{z}^*_{i,k+1}(\theta^k_i)$ is a supergradient of the concave conjugate at θ^k_i, so

$$g^k_{\theta_i} = \widetilde{N}_{i,k} - \frac{1}{K}\widehat{z}^*_{i,k+1}(\theta^k_i) \tag{27}$$

is a subgradient of h_k. Also $g^k_\mu = \bar{B} - \widetilde{A}_k$ is the gradient of m_k. The gradient bounds are

$$\|\boldsymbol{g}^k_\theta\|_2 \le \sqrt{M}N_{\max}, \qquad |g^k_\mu| \le G_B. \tag{28}$$

The domain diameters are $\bar{c}\sqrt{M}$ and $\bar{c}$. Standard projected-OGD analysis therefore gives

$$\sum_{k=1}^K h_k(\boldsymbol{\theta}^k) - \min_{\boldsymbol{\theta}\in[0,\bar{c}]^M} \sum_{k=1}^K h_k(\boldsymbol{\theta}) \le \bar{c}MN_{\max}\sqrt{K}, \tag{29}$$

$$\sum_{k=1}^K m_k(\mu^k) - \min_{\mu\in[0,\bar{c}]} \sum_{k=1}^K m_k(\mu) \le \bar{c}G_B\sqrt{K}. \tag{30}$$

These inequalities are pathwise; the losses may depend on the realized feedback and current proposal.

Define

$$\widetilde{z}_i := \sum_{k=1}^K \widetilde{N}_{i,k}, \qquad \widetilde{T} := \sum_i \widetilde{z}_i, \qquad \widetilde{V} := [\widetilde{T} - B]_+. \tag{31}$$

Let

$$\overline{U}^*_i(\theta) := \frac{1}{K}\sum_{k=1}^K \widehat{U}^*_{i,k+1}(\theta), \tag{32}$$

$$\overline{U}_i(z) := \min_{0\le\theta\le\bar{c}}\{\theta z - \overline{U}^*_i(\theta)\}. \tag{33}$$

Separability gives

$$\min_{\boldsymbol{\theta}} \sum_{k=1}^K h_k(\boldsymbol{\theta}) = \sum_{i=1}^M \overline{U}_i(\widetilde{z}_i), \tag{34}$$

$$\min_{0\le\mu\le\bar{c}} \sum_{k=1}^K m_k(\mu) = -\bar{c}\widetilde{V}. \tag{35}$$

Hence

$$\sum_{k=1}^K [h_k(\boldsymbol{\theta}^k) + m_k(\mu^k)] \le \sum_i \overline{U}_i(\widetilde{z}_i) + \bar{c}(MN_{\max} + G_B)\sqrt{K} - \bar{c}\widetilde{V}. \tag{36}$$

Let $\mathbf{N}^*$ attain equation 4, define $z^*_i = \sum_k N^*_{i,k}$, and set $n^*_i = z^*_i/K$. Then $0 \le n^*_i \le N_{\max}$ and

$$\sum_i n^*_i = \frac{1}{K}\sum_i z^*_i \le \frac{B}{K} = \bar{B} \le C. \tag{37}$$

Thus $\boldsymbol{n}^* \in \mathcal{P}_C$. Since $\mathcal{P}_C$ is integral, the integer proposal maximizes the same linear function over the continuous polytope. Therefore

$$\sum_i (\theta^k_i - \mu^k)\widetilde{N}_{i,k} \ge \sum_i (\theta^k_i - \mu^k)n^*_i. \tag{38}$$

Using $\mu^k \ge 0$, $\sum_i n^*_i \le \bar{B}$, and the proxy Fenchel representation yields

$$h_k(\boldsymbol{\theta}^k) + m_k(\mu^k) \ge \frac{1}{K}\sum_i \left[\theta^k_i z^*_i - \widehat{U}^*_{i,k+1}(\theta^k_i)\right] \tag{39}$$

$$\ge \frac{1}{K}\sum_i \widehat{U}_{i,k+1}(z^*_i). \tag{40}$$

Summing and combining with equation 36 gives

$$\sum_i \frac{1}{K} \sum_k \widehat{U}_{i,k+1}(z_i^*) \le \sum_i \overline{U}_i(\widetilde{z}_i) + \bar{c}(MN_{\max} + G_B)\sqrt{K} - \bar{c}\widetilde{V}. \tag{41}$$

Let $e_{i,k} = |\widehat{q}_{i,k+1} - q_i|$ and

$$\delta_{i,k} := \frac{4\lambda Z}{R} e_{i,k}, \qquad \Delta := \frac{1}{K} \sum_{i,k} \delta_{i,k}. \tag{42}$$

By Lemma 2, both the proxy utility and its conjugate differ from their target versions by at most $\delta_{i,k}$. Consequently,

$$\mathsf{OPT}_K^{\mathrm{stat}}(\boldsymbol{q}) - \sum_i U(q_i, \widetilde{z}_i) + \bar{c}\widetilde{V} \le \bar{c}(MN_{\max} + G_B)\sqrt{K} + 2\Delta. \tag{43}$$

Since $Z = KN_{\max}$,

$$\Delta = \frac{4\lambda N_{\max}}{R} \sum_{i,k} e_{i,k} = \frac{4\lambda M N_{\max}}{\bar{N}} \bar{e}_K^{\mathrm{stat}}. \tag{44}$$

It remains to pass from virtual proposals to executed allocations. The prefix clipping rule implies

$$\sum_i (\widetilde{z}_i - z_{i,K}^{CERO}) = \widetilde{V}. \tag{45}$$

Every target utility has derivative at most $\bar{c}$, so

$$0 \le \sum_i [U(q_i, \widetilde{z}_i) - U(q_i, z_{i,K}^{CERO})] \le \bar{c}\widetilde{V}. \tag{46}$$

This clipping loss is exactly absorbed by the violation term on the left of equation 43. Therefore

$$\mathrm{Reg}_K^{\mathrm{stat}} \le \bar{c}(MN_{\max} + G_B)\sqrt{K} + 2\Delta. \tag{47}$$

Substitute $\bar{c} = \lambda/R$, $R = K\bar{N}$, use equation 44, and divide by M. Since every prompt utility lies in $[0, 1]$, the average regret is also at most one. Combining the two bounds gives equation 11.

E Proof of the Same-Path Guarantee

We first record a stability inequality for one incremental utility.

Lemma 3 (Increment stability). *For any $q, q' \in [0, 1/4]$ and $z, n \ge 0$ satisfying $z + n \le Z$,*

$$|[U(q, z+n) - U(q, z)] - [U(q', z+n) - U(q', z)]| \le \frac{4\lambda}{R} n|q - q'|. \tag{48}$$

Proof. Let $\kappa = 4\lambda/R$ and

$$G(q; z, n) = \int_z^{z+n} \kappa q e^{-\kappa q s}\, ds. \tag{49}$$

Then

$$\partial_q G(q; z, n) = \int_z^{z+n} \kappa(1 - \kappa q s) e^{-\kappa q s}\, ds. \tag{50}$$

Because $|(1-x)e^{-x}| \le 1$ for $x \ge 0$, the derivative magnitude is at most κn. Apply the mean-value theorem. □

For each prompt choose the midpoint of its realized rate range,

$$q_i^\circ := \frac{1}{2}\left(\min_k q_{i,k} + \max_k q_{i,k}\right). \tag{51}$$

Then $|q_{i,k} - q_i^\circ| \le V_K/2$ for every i, k. For any feasible allocation $\mathbf{N}$, let

$$J^{\mathrm{dyn}}(\mathbf{N}) := \sum_{i,k} \Delta U_{i,k}(\mathbf{N}; q_{i,k}), \tag{52}$$

$$J^{\circ}(\mathbf{N}) := \sum_{i} U(q_i^\circ, z_{i,K}). \tag{53}$$

Using equation 48 and total budget feasibility,

$$|J^{\mathrm{dyn}}(\mathbf{N}) - J^{\circ}(\mathbf{N})| \le \frac{4\lambda}{R} \sum_{i,k} N_{i,k} |q_{i,k} - q_i^\circ| \tag{54}$$

$$\le \frac{2\lambda V_K}{R} B = 2\lambda M V_K. \tag{55}$$

Applying this inequality to the dynamic optimum and to the executed CERO allocation gives

$$\mathrm{Reg}_K^{\mathrm{dyn}} \le \mathsf{OPT}_K^{\mathrm{stat}}(\boldsymbol{q}^\circ) - \sum_i U(q_i^\circ, z_{i,K}^{CERO}) + 4\lambda M V_K. \tag{56}$$

The average proxy error relative to $\boldsymbol{q}^\circ$ satisfies

$$\frac{1}{MK} \sum_{i,k} |\widehat{q}_{i,k+1} - q_i^\circ| \le \bar{e}_K^{\mathrm{dyn}} + \frac{1}{MK} \sum_{i,k} |q_{i,k} - q_i^\circ| \tag{57}$$

$$\le \bar{e}_K^{\mathrm{dyn}} + \frac{V_K}{2}. \tag{58}$$

Apply Theorem 1 with reference vector $\boldsymbol{q}^\circ$ and substitute the last inequality into equation 56. In addition, equation 3 and total budget feasibility imply that every same-path objective is at most λM, so the normalized regret is at most λ. Dividing by M and combining the two bounds yields equation 12.

F Proofs of the Beta-state Propositions

Proof of Proposition 1. Conditional on $P_{i,k}$, each pair of future Bernoulli rewards disagrees with probability $2P_{i,k}(1 - P_{i,k})$. The unbiased sample variance for binary data satisfies

$$2\widehat{\mathrm{Var}}_G(r) = \binom{G}{2}^{-1} \sum_{1 \le j < \ell \le G} \mathbf{1}\{r^{(j)} \ne r^{(\ell)}\}.$$

Taking conditional expectations first given $P_{i,k}$ and then given $\mathcal{H}_i^k$ proves the claim. □

Proof of Proposition 2. Let $t = n + s_0$ and $\widehat{m} = a/t$. Then

$$\widehat{q}_n = \frac{t}{t+1} \widehat{m}(1 - \widehat{m}). \tag{59}$$

The function $x(1 - x)$ is one-Lipschitz on $[0, 1]$, so

$$|\widehat{q}_n - p(1-p)| \le |\widehat{m} - p| + \frac{1}{4(t+1)}. \tag{60}$$

Moreover,

$$|\mathbb{E}\widehat{m} - p| = \frac{|a_0 - p s_0|}{t} \le \frac{s_0}{t}, \tag{61}$$

and

$$\mathbb{E}|\widehat{m} - \mathbb{E}\widehat{m}| \le \sqrt{\mathrm{Var}(\widehat{m})} = \frac{\sqrt{np(1-p)}}{t} \le \frac{\sqrt{n}}{2t}. \tag{62}$$

Combining the bounds proves equation 7. □

G Implementation Details

Training. We use two locally modified veRL-based (Sheng et al., 2025) stacks with matched learner and decoding hyperparameters, vLLM 0.8.3 for generation, and FSDP for actor training. Bfloat16 is used for generation and actor computation. Experiments use two NVIDIA H800 GPUs for the 1.5B backbone and four for the 4B and 7B backbones. All methods use GRPO with within-group standard-deviation normalization, advantage clipping to $[-5, 5]$, and loss averaged over valid response tokens. AdamW uses a constant learning rate of 10^{-6}, $(\beta_1, \beta_2) = (0.9, 0.999)$, zero weight decay, and no warmup. We use one PPO epoch per collected batch, no reshuffling, and a global PPO mini-batch size of 32. The lower/upper policy clipping parameters are $0.20/0.28$, the dual-clip constant is 10, the importance-ratio cap is 2, and the gradient-norm threshold is 1. KL and entropy regularization are disabled. Training prompts are left-truncated to 2,048 tokens and responses are capped at 2,048 tokens; sampling uses temperature 1.0, top-p 1.0, and top-k -1. Training rewards are computed using the rule-based mathematical verifier from DAPO (Yu et al., 2025), which treats $+1$ as success and 0 as failure. The CERO run in Table 1 is included in the multi-seed results in Table 2.

A scheduler round is one allocation opportunity, whereas a training step is one policy-update call on a collected rollout batch. Zero-allocation rounds advance the scheduler without updating the policy. A data epoch is one traversal of the training data loader, distinct from a PPO epoch over a collected rollout batch.

Allocation settings. We use Algorithm 1 with $g = 4$ responses per allocation unit and $N_{\max} = 2$, $C = 132$, and $B = 64{,}000$ in allocation units. CERO uses a $\mathrm{Beta}(1, 1)$ prior, $\lambda = \rho = 1$, $\nu^1 = 0.5$, and $\gamma_\vartheta = \gamma_\nu = 1/\sqrt{500}$. Ties use a fixed seed permutation. Reserve-aware catch-up was enabled but never activated in the reported CERO runs. KnapsackRL uses positive and negative base allocations of two, a reward-history buffer of length 16, a minimum initialized history length of four, and robust epoch estimation.

CERO-greedy uses the same reward-proxy estimator, utility function, and fixed 512-response quota as CERO-fixed, but replaces prompt-dual scores with the marginal utility of one additional complete group. In our experiments, each group contains eight responses, corresponding to $N_{\max} = 2$ allocation units. At the beginning of round k, the selection score for prompt i is

$$\Delta_{i,k}^{\mathrm{grp}} := U(\hat{q}_{i,k}, z_{i,k-1} + N_{\max}) - U(\hat{q}_{i,k}, z_{i,k-1}),$$

where $\hat{q}_{i,k}$ is the pre-decision proxy and $z_{i,k-1}$ is the cumulative executed exposure in the control run. We use CERO's exponential utility with the same saturation parameter and exposure normalization. Each round scores the full training prompt pool and selects the 64 prompts with the largest scores, generating one eight-response group per prompt. This control preserves the reward-based utility and its diminishing-return structure while replacing dual-based selection with direct marginal-utility ranking.

Evaluation. We follow Tan et al. (2025) and evaluate the checkpoint obtained after 500 scheduler rounds using 16 responses per problem, temperature 0.6, top-p 1.0, and top-k -1. All methods use the same rule-based mathematical equivalence checker without an LLM judge. Metric definitions are given in Section 6 and Appendix B.1; computational costs and schedule audits are reported in Appendix B.4.