Train at the Moving Edge: Efficient RL for Large Reasoning Models via Rollout Selection

Jiahao Wu[1,2,*] **Ning Lu**[1,3,*] **Shengcai Liu**[1,†] **Kun Wang**[4] **Yanting Yang**[5]
Baijiong Lin[6] **Chen Jason Zhang**[2] **Qing Li**[2] **Ke Tang**[1]

[1]Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of CSE, SUSTech [2]Department of Computing, PolyU
[3]Department of CSE, HKUST [4]College of Computing and Data Science, NTU
[5]Department of CS, Rutgers [6]AI Thrust, HKUST(GZ)

Abstract

Reinforcement learning (RL) has become essential for post-training large language models (LLMs) in reasoning tasks. While scaling rollouts can stabilize training and enhance performance, it introduces substantial computational overhead. In algorithms like GRPO, multiple rollouts per prompt incur prohibitive costs, as a large portion of prompts provide negligible gradients and are thus of low utility. This raises a key question: *how to identify high-utility prompts before an expensive rollout?* Our experimental analysis reveals that sample utility is non-uniform and dynamic: the strongest learning signals concentrate at the "learning edge", the intersection of intermediate difficulty and high response entropy, which shifts throughout training. Motivated by this observation, we propose HIVE, a history-informed and online-verified prompt selection framework for data-efficient RL training. HIVE first uses historical reward statistics and response entropy as a cheap prior to filter candidate prompts, and then employs prompt entropy as a real-time proxy to prune instances with stale utility. Across multiple reasoning benchmarks and base models, HIVE maintains reasoning accuracy with substantial rollout cost reduction, achieving up to 2.3× total training speedup. Code is here[‡].

1 Introduction

Reinforcement Learning (RL) has emerged as a prevalent paradigm for fine-tuning large language models (LLMs), particularly for enhancing complex mathematical and logical reasoning capabilities [16, 53, 66]. While advanced RL algorithms such as group relative policy optimization (GRPO) [43] rely on rollout scaling to stabilize training and enhance model performance, they introduce a severe computational overhead [31, 59, 66]. In the training of those RL algorithms, a large portion of computational resources is wasted on generating rollouts for low-utility prompts that are either too trivial or too intractable for the current model. This results in vanishing gradients and wasted resources, providing no informative learning signal for model updates [39, 45, 66, 77]. Therefore, we investigate the following research question in this paper: *how to identify prompts that are likely to produce useful gradients before rollout?*

Existing methods face limitations in efficiently selecting high-utility prompts in an online manner. Static or offline filters rank prompts before training using fixed heuristics, making them inexpensive but unable to adapt to the changing model during RL training [30, 55, 65]. Rollout-based online filters, such as dynamic sampling, better capture the current utility of each prompt by checking reward variance from newly generated responses [6, 59, 64, 66]. However, they already incur

[*]Equal contribution. [†]Corresponding author. [‡]Code: JiahaoWuGit/HIVE-Efficient-Rollout-RL.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

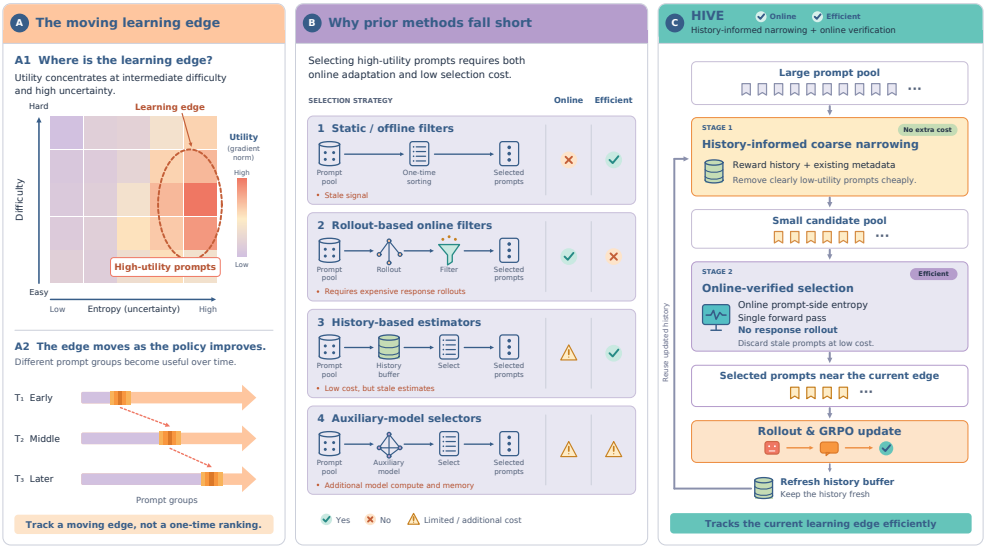


Figure 1: **RLVR prompt selection is a moving-edge tracking problem.** (A) Utility concentrates at intermediate difficulty and high uncertainty, and the learning edge is dynamic. (B) Prior methods face limitations in efficiently selecting online high-utility prompts. (C) HIVE combines history-informed coarse narrowing with low-cost online verification to track the current learning edge.

substantial rollout cost. History-based estimators such as GRESO [77] avoid additional rollout cost by estimating prompt utility from past reward statistics, but the metadata can become stale as the policy updates [31, 50, 77]. Auxiliary-model selectors, such as PCL [14], attach external or jointly trained predictors to estimate prompt difficulty or utility, requiring extra compute and memory.

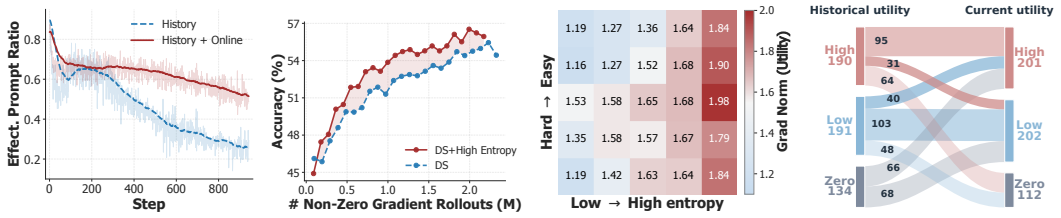
To identify what makes a prompt useful during RL training, we first empirically analyze GRPO training dynamics. Among prompts with non-zero gradients, those with high response-side entropy produce greater learning progress under the same rollout budget (Figure 2b). More broadly, utility, measured by the length-normalized gradient norm, concentrates at the intersection of intermediate difficulty and high response-side entropy (Figure 2c), which we refer to as the learning edge. However, this edge is dynamic: prompts selected using historical metadata quickly lose real-time utility, whereas prompts verified under the current policy are more likely to be effective (Figure 2a); prompts move substantially among High, Low, and Zero utility classes across training steps (Figure 2d). These observations suggest that an effective prompt selection framework for efficient LLM RL should be **online**, to track the moving learning edge, and **efficient**, to avoid excessive selection overhead.

Based on previous analysis, we propose **HIVE** (**H**istory-**I**nformed **V**erification of the learning **E**dge), a dual-stage framework for rollout-efficient RLVR (Figure 3). HIVE first performs history-informed coarse narrowing using reward trajectories and response-side entropies accumulated from previous training steps. This stage targets efficiency by retaining exploration probability for prompts that are around the edge. HIVE then utilizes prompt entropy as a high-fidelity and efficient proxy to perform online verification under the current policy, filtering stale candidates before the rollout phase.

Evaluation for HIVE is conducted across multiple reasoning benchmarks (e.g., math and tool calling) and models: Qwen3-4B-Base [61], Qwen2.5-Math-1.5B/7B [60], DeepSeek-R1-Distill-Qwen-1.5B [16], Qwen2.5-14B/32B [52], and Llama-3.2-3B-Instruct [15]. We compare the observed accuracy–cost tradeoff of HIVE with baselines like Dynamic Sampling [66], GRESO [77] and PCL [14], achieving up to **2.3× faster total training time** for models like Qwen2.5-Math-7B. Finally, we show that HIVE lowers computational overhead, achieving **up to 9.2 million rollouts reduction** with broadly comparable reported reasoning scores. Beyond mathematical reasoning, HIVE also attains a similar observed tool-calling score on our BFCL-V2 split while using **4.8× fewer rollouts** than Dynamic Sampling.

2 Empirical Observations: The Moving Learning Edge

This section dissects the training dynamics of GRPO-style RLVR to identify what makes a prompt useful for policy optimization and why effective selection in RLVR is an online tracking problem.



(a) (b) (c) (d)

Figure 2: **Empirical observations.** (a) Selection only based on historical statistics becomes increasingly stale during training, while online verification retains more effective prompts. (b) Among prompts with non-zero gradients, prioritizing high-entropy prompts improves accuracy at the same rollout budget. (c) Gradient utility concentrates at an intermediate-difficulty, high-entropy learning edge. (d) A large fraction of sampled prompts shift between High, Low, and Zero utility classes across steps 500 and 1000, indicating that prompts selected by historical metadata no longer match the current policy's utility distribution.

We first recall why low-utility prompts produce no GRPO learning signal, then show that the useful prompts concentrate at a moving edge characterized by intermediate difficulty and high uncertainty. The experimental details are reported in Appendix C.

Zero-variance prompts produce vanishing advantages. Group relative policy optimization (GRPO) [43] is widely used in RLVR for reasoning models [16, 61]. For a prompt q, GRPO samples a group of responses $\{o_i\}_{i=1}^{G}$ and optimizes

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|q)} \left\{ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left(\min\left[\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t}\mid q,o_{i,<t})}\hat{A}_{i,t}, \right.\right.\right. \\ \left.\left.\left. \mathrm{clip}\left(\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t}\mid q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta D_{\mathrm{KL}}[\pi_\theta \,\|\, \pi_{\mathrm{ref}}]\right)\right\}, \tag{1}$$

where $P(Q)$ is the prompt distribution, G is the response group size, and $o_{i,t}$ denotes the t-th token of response o_i. The KL penalty with weight β anchors π_θ to the reference policy π_{ref}. Since RLVR gives a sequence-level reward r_i, all tokens in o_i share the group-normalized advantage $\hat{A}_{i,t} = \hat{A}_i$:

$$\hat{A}_i = \begin{cases} (r_i - \bar{r})/\sigma_r, & \sigma_r > 0, \\ 0, & \sigma_r = 0, \end{cases} \qquad \bar{r} = \frac{1}{G}\sum_{j=1}^{G} r_j, \quad \sigma_r = \mathrm{std}(\{r_j\}_{j=1}^{G}). \tag{2}$$

When all sampled responses for a prompt receive the same reward, the reward variance is zero and the prompt contributes no informative policy-gradient signal. Such prompts are either already easy for the current policy or still too hard to discriminate among sampled responses. Rollout scaling therefore wastes compute unless it avoids these zero-variance regions.

Utility concentrates at an intermediate-difficulty, high-entropy edge. Zero-variance filtering is necessary but not sufficient: among prompts with non-zero gradients, some still provide much stronger learning signal than others. Following previous analyses [12, 26, 41, 47], we use length-normalized gradient norm as a proxy for per-prompt utility and study how it varies with empirical difficulty and response-side entropy.

Figure 2c shows two consistent patterns. First, within a fixed difficulty band, gradient norm increases with response-side entropy. Second, at a fixed entropy level, utility follows an inverted-U shape over difficulty: trivial prompts and intractable prompts have low utility, while intermediate-difficulty prompts have higher utility. The strongest learning signal appears at the intersection of these two conditions. We call this region the *learning edge*. This observation explains why merely removing zero-variance prompts is too coarse: *the best rollout budget should be allocated to prompts that are both learnable and uncertain under the current policy*.

Historical metadata becomes stale as the edge moves. As the policy updates, the model's state of knowledge changes, and historical metadata such as reward trajectories and response entropies can quickly become stale [10, 31, 53]. To quantify this effect, we compare prompts selected by historical metadata from the previous training window with prompts selected by online metrics recomputed

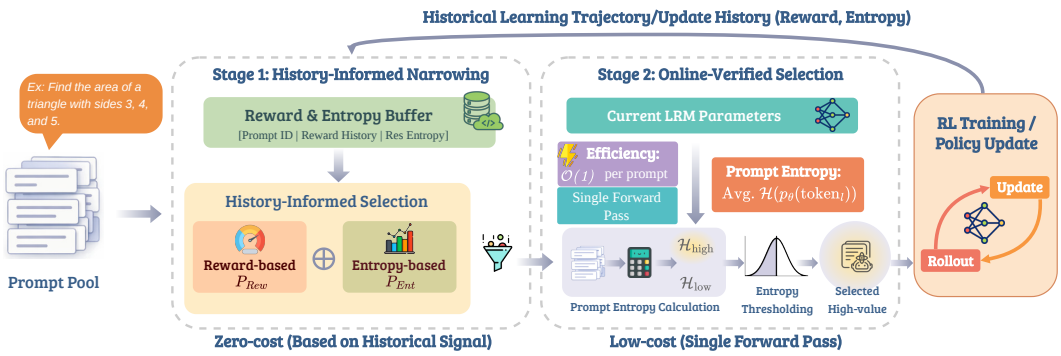


Figure 3: Overview of HIVE. **HIVE tracks the moving learning edge by decomposing prompt selection into two stages. Stage 1** uses historical reward trajectories and response-side entropy for efficient coarse narrowing. **Stage 2** computes prompt-side entropy under the current policy for low-cost online verification, rejecting stale candidates before rollout.

under the current policy. As shown in Figure 2c, historical selection yields lower current gradient norms, whereas online selection isolates prompts with high utility. Figure 2d further shows that prompts frequently move among High, Low, and Zero utility classes across training steps. Together, these results indicate that: *a selector must revalidate candidate prompts under the current policy to keep tracking the moving edge*.

3 Methodology

Section 2 shows that useful RLVR prompts concentrate near a moving learning edge and historical metadata alone is unreliable because the edge moves as the policy changes. Based on these observations, HIVE is designed as a two-stage pre-rollout selector (Figure 3 and Algorithm 1). Stage 1 uses historical reward and response-entropy traces as a cheap prior to form a candidate pool; Stage 2 revalidates those candidates under the current policy using prompt-side entropy before expensive rollouts for the standard GRPO update.

3.1 Design Objective

A practical RLVR prompt selector must be both **online**, adapting as the policy evolves and the learning frontier shifts during training, and **efficient**, avoiding full response rollouts that could erase its computational benefits. HIVE achieves this via two stages. Stage 1 targets efficiency: it uses historical reward trajectories and response-side entropies to cheaply narrow the full prompt pool to a candidate set. Stage 2 restores online precision: it verifies candidates with prompt-side entropy computed under the current model parameters. The final selected prompts are then passed to the standard rollout and GRPO update. The detailed procedure is provided in Appendix F.

3.2 Stage 1: History-Informed Coarse Narrowing

Stage 1 removes prompts that are unlikely to lie near the learning edge while keeping enough exploration to handle edge movement. It combines a reward-based score, which tracks difficulty saturation, with an entropy-based score, which favors historically uncertain prompts.

Reward-history score. For each prompt $x_i \in \mathcal{D}$, HIVE stores a trace $\mathcal{T}_i^{(t)} = \{R_{i,j}\}_{j=1}^{n_i}$ before training step t, where $R_{i,j} = \{r_{i,j}^{(k)}\}_{k=1}^{G}$ is the reward group recorded at the j-th previous visit. Let $\zeta_{i,j} = \mathbf{1}\{\mathrm{Var}(R_{i,j}) = 0\}$ indicate whether that visit produced a zero-variance reward group. HIVE summarizes recent saturation by the length of the trailing zero-variance run:

$$z_i^{(t)} = \max\left\{ k \in \{0, \dots, n_i\} : \prod_{j=n_i-k+1}^{n_i} \zeta_{i,j} = 1 \right\},$$
$$s_{\text{rew}}(x_i) = \max\left\{ \left(p_{e,\tau_i}^{(t)}\right)^{z_i^{(t)}}, \epsilon_p \right\}, \qquad \tau_i \in \{\text{easy}, \text{hard}\}. \tag{3}$$

Here τ_i distinguishes all-correct and all-incorrect zero-variance prompts, $p_{e,\tau_i}^{(t)} \in (0,1)$ is the corresponding exploration probability, and $\epsilon_p = 0.01$ is a per-prompt score floor. Repeated zero-variance outcomes decay the score exponentially down to this floor, preserving exploration for prompts that may later become learnable as the policy improves.

HIVE adapts $p_{e,\tau}$ online using the observed zero-variance ratio $\hat{\rho}_\tau^{(t)}$: the probability decreases above target α, increases below it, and remains unchanged at equality:

$$p_{e,\tau}^{(t+1)} = \left[p_{e,\tau}^{(t)} + \Delta p \cdot \operatorname{sign}\left(\alpha - \hat{\rho}_\tau^{(t)}\right) \right]_{[p_{\min}, p_{\max}]}, \qquad \tau \in \{\text{easy}, \text{hard}\}, \tag{4}$$

where $[\cdot]_{[p_{\min}, p_{\max}]}$ denotes projection to a valid probability range. This update prevents HIVE from persistently over-sampling either mastered prompts or currently intractable prompts.

Response-entropy score. Historical response-side entropy provides a low-cost prior for uncertainty. For the most recent rollout group of x_i, let

$$\begin{aligned} U_i^{(r)} &= \frac{1}{L_{i,r}} \sum_{\ell=1}^{L_{i,r}} \mathcal{H}\left(p_\theta(\cdot \mid x_i, o_{i,<\ell}^{(r)}) \right), \qquad H_i = \frac{1}{G} \sum_{r=1}^{G} U_i^{(r)}, \\ s_{\text{ent}}(x_i) &= \frac{H_i - H_{\min}}{H_{\max} - H_{\min} + \epsilon}, \qquad \mathcal{H}(p) = -\sum_{v \in \mathcal{V}} p_v \log p_v. \end{aligned} \tag{5}$$

The normalization is computed within the current metadata pool. It makes the entropy score comparable across training stages even as the policy becomes more confident.

Candidate sampling. The Stage 1 acceptance probability combines recent reward saturation and historical uncertainty:

$$P_{\text{S1}}(x_i) = \lambda s_{\text{rew}}(x_i) + (1-\lambda) s_{\text{ent}}(x_i), \qquad \lambda \in [0,1]. \tag{6}$$

HIVE samples prompts with Bernoulli probability $P_{\text{S1}}(x_i)$, yielding a candidate set $\mathcal{C}_t$ for Stage 2. Stage 1 reuses existing metadata and requires no extra response generation or model forward pass.

3.3 Stage 2: Online-Verified Selection

Stage 1 is efficient, but Section 2 shows why it cannot be the final decision rule: reward and entropy traces become stale as the current policy moves. Stage 2 therefore verifies the candidate set $\mathcal{C}_t$ under the current policy π_{θ_t} before rollouts are generated. Instead of generating G responses per candidate, HIVE uses prompt-side entropy as a lightweight proxy for current uncertainty.

Prompt-side entropy. For a prompt $x = (x_1, \ldots, x_{L_x})$, HIVE computes

$$V_t(x) = \frac{1}{L_x - 1} \sum_{\ell=2}^{L_x} \mathcal{H}(p_{\theta_t}(\cdot \mid x_{<\ell})). \tag{7}$$

This score is online because it is evaluated with the current policy parameters, and efficient because it only requires a prompt-side forward pass. In the HIVE pipeline, this verifier complements Stage 1: reward history filters repeated zero-variance regions, while $V_t(x)$ checks whether a historically promising candidate still exhibits current-policy uncertainty.

We give a conditional ordering result for prompt and response entropies. For a fixed checkpoint and prompt set, assume the relation between response entropies and prompt entropies: $U^*(x) = aV_t(x) + b + e(x)$ with $a > 0$ and $|e(x)| \le \varepsilon_{\text{cal}}$ (Appendix A.2).

Theorem 3.1 (Conditional rank consistency)**.** *Fix two prompts before drawing n independent diagnostic rollouts per prompt. Under the stated calibration assumption, for any $\delta \in (0,1)$, their empirical response-entropy ordering agrees with their prompt-entropy ordering with probability at least $1-\delta$ if*

$$a|V_t(x) - V_t(x')| > 2\varepsilon_{\text{cal}} + 2\log|\mathcal{V}| \sqrt{\frac{\log(4/\delta)}{2n}}. \tag{8}$$

The theorem provides conditional pairwise entropy ordering, explaining the efficacy that prompt-side entropy can serve as a low-cost online proxy. The proof is given in Appendix A. The binned associations (Appendix E.1) and stage ablations (Section 4.4) also motivate this proxy empirically.

Table 1: Comprehensive evaluation of HIVE on math reasoning benchmarks and efficiency analysis. We train multiple models on DAPO+MATH. (a) HIVE maintains comparable accuracy while reducing the number of rollouts. (b) HIVE lowers rollout cost, achieving up to 3.8× rollout speedup and 2.3× total training speedup over Dynamic Sampling.

(a) Performance (%) comparison.

Method	Math500	AIME24	AMC	Gaokao	Miner.	Olymp.	Avg.	#Rollout
Qwen2.5-Math-1.5B								
DS	77.3	16.7	61.7	64.2	31.8	38.7	48.4	7.6M (1.0×)
Pre-filter	76.8	20.8	50.0	62.1	33.2	38.5	46.9	3.3M (2.3×)
GRESO	77.3	15.0	59.3	66.2	32.6	38.5	48.1	3.3M (2.3×)
PCL	74.5	21.7	47.9	61.7	33.6	37.5	46.2	3.3M (2.3×)
HIVE	77.9	16.7	60.2	66.1	31.5	40.3	48.8	**3.1M (2.5×)**
DeepSeek-R1-Distill-Qwen-1.5B								
DS	87.9	36.7	71.7	78.7	35.3	54.9	60.9	2.4M (1.0×)
Pre-filter	79.7	22.5	55.1	70.5	27.2	47.5	50.4	1.6M (1.5×)
GRESO	85.5	37.5	70.7	76.2	34.3	52.1	59.4	1.6M (1.5×)
PCL	82.4	26.7	60.5	71.7	29.9	50.3	53.6	1.6M (1.5×)
HIVE	87.2	37.5	70.1	77.2	35.1	53.2	60.1	**1.5M (1.6×)**
Llama3.2-3b-Instruct								
DS	52.4	16.7	46.0	49.2	20.0	20.2	34.1	11.4M (1.0×)
Pre-filter	48.1	4.2	20.2	39.8	17.8	16.8	24.5	5.9M (1.9×)
GRESO	51.8	16.7	46.3	49.4	19.7	19.6	33.9	5.9M (1.9×)
PCL	53.1	15.8	27.1	48.2	22.5	19.3	31.0	5.9M (1.9×)
HIVE	53.8	17.5	45.5	49.1	20.5	20.0	34.4	**4.8M (2.4×)**
Qwen3-4B-Base								
DS	84.4	22.5	59.5	70.1	38.2	49.7	54.1	3.0M (1.0×)
Pre-filter	80.3	19.2	51.8	67.4	38.7	46.8	50.7	1.6M (1.8×)
GRESO	83.5	21.8	59.6	70.8	39.9	49.5	54.2	1.6M (1.8×)
PCL	80.5	18.3	52.1	67.5	39.8	46.5	50.7	1.6M (1.8×)
HIVE	84.7	22.5	59.9	71.0	40.1	50.4	54.7	**1.5M (2.0×)**
Qwen2.5-Math-7B								
DS	82.9	34.2	79.2	71.7	35.4	43.6	57.8	13.1M (1.0×)
Pre-filter	76.8	34.2	61.4	62.4	33.6	41.3	51.6	6.3M (2.1×)
GRESO	93.2	30.0	78.9	70.2	35.2	44.1	58.6	6.3M (2.1×)
PCL	76.4	32.5	62.7	65.1	34.4	40.8	52.0	6.3M (2.1×)
HIVE	93.0	36.7	79.2	70.8	34.8	43.8	59.7	**3.9M (3.4×)**

(b) Training time (hours).

Method	Train	Other	Rollout	Total
Qwen2.5-Math-1.5B				
DS	10.9	6.0	77.3	94.2
Pre-filter	11.0	6.0	45.9	62.9
GRESO	11.3	6.4	46.0	63.7
PCL	11.4	7.0	46.3	64.7
HIVE	11.2	6.2	**42.1**	**59.5**
DeepSeek-R1-Distill-Qwen-1.5B				
DS	9.6	6.1	68.7	84.4
Pre-filter	10.1	6.2	45.8	62.1
GRESO	10.4	6.6	45.9	62.9
PCL	10.5	7.1	46.0	63.6
HIVE	10.1	6.7	**42.7**	**59.5**
Llama3.2-3b-Instruct				
DS	20.6	10.2	145.0	175.8
Pre-filter	21.4	10.0	57.3	88.7
GRESO	21.8	10.5	57.5	89.8
PCL	22.0	11.2	57.6	90.8
HIVE	22.2	10.6	**46.9**	**79.7**
Qwen3-4B-Base				
DS	9.7	6.2	86.0	101.9
Pre-filter	10.2	6.3	45.9	62.4
GRESO	10.5	6.7	46.1	63.3
PCL	10.6	7.3	46.1	64.0
HIVE	10.2	6.8	**42.8**	**59.8**
Qwen2.5-Math-7B				
DS	32.1	12.6	153.7	198.4
Pre-filter	32.4	12.9	66.1	111.4
GRESO	32.8	13.2	66.3	112.3
PCL	33.0	14.0	66.5	113.5
HIVE	32.7	12.9	**40.2**	**85.8**

Median verification gate. Given the Stage 1 candidate set $\mathcal{C}_t$, HIVE uses a median gate:

$$\gamma_t = \text{median}\{V_t(x) : x \in \mathcal{C}_t\}, \qquad \mathcal{B}_t = \{x \in \mathcal{C}_t : V_t(x) \geq \gamma_t\}. \tag{9}$$

The final set $\mathcal{B}_t$ is the rollout batch for the current GRPO step. The median threshold keeps throughput stable, adapts to the current candidate distribution, and rejects stale low-uncertainty prompts.

Computational cost. For $|\mathcal{C}_t|$ candidates, Stage 2 costs $\mathcal{O}(|\mathcal{C}_t|)$ times forward computation. A rollout-based verifier would require $\mathcal{O}(|\mathcal{C}_t|GL_r)$ times. Thus prompt-side verification is negligible relative to multi-rollout filtering. Section 4.3 empirically confirms this: online verification adds only 0.82 seconds per iteration, less than 0.4% of the step time.

4 Experiments

We evaluate HIVE along the edge-tracking objectives introduced in Section 3. First, does HIVE preserve reasoning accuracy while using fewer rollouts? Second, is HIVE cheap enough to be practical? Third, does HIVE keep rejecting stale or low-utility prompts as the learning edge moves?

4.1 Experimental Settings

Models and datasets. We evaluate Qwen2.5-Math-1.5B/7B [60], DeepSeek-R1-Distill-Qwen-1.5B [16], Qwen3-4B-Base [61], Qwen2.5-14B/32B [52], and Llama-3.2-3B-Instruct [15]. We use a context length of 4K for Qwen2.5-Math-1.5B/7B, 8K for DeepSeek-R1-Distill-Qwen-1.5B, Qwen3-4B-Base, and Llama-3.2-3B-Instruct, and 16K for Qwen2.5-14B/32B. We train on two math-reasoning corpora following [77]: DAPO+MATH (DM), which combines DAPO [66] and MATH [19], and a 30K subset of Open-R1 Hugging Face [23].

Training and evaluation. HIVE is implemented with verl [45] and vLLM [27]. Main experiments use 8×A100-80G GPUs and train each method for 1000 steps. We evaluate on six mathematical reasoning benchmarks: Math500 [19], AIME24 [2], AMC [3], Minerva Math [28], Gaokao [73], and

Table 2: Performance (%) and rollout comparison of Qwen2.5-Math-1.5B and Qwen2.5-Math-7B trained on Open-R1.

Method	Math500	AIME24	AMC	Gaokao	Miner.	Olymp.	Avg.	#Rollout
Qwen2.5-Math-1.5B								
DS	77.1	16.7	50.3	65.5	30.9	39.7	46.7	3.8M (1.0×)
GRESO	76.1	20.0	50.6	65.1	30.0	39.2	46.8	1.6M (2.4×)
HIVE	76.9	18.3	48.7	65.4	30.1	40.3	46.6	**1.5M (2.5×)**
Qwen2.5-Math-7B								
DS	82.8	34.2	63.5	67.3	35.7	46.3	55.0	11.4M (1.0×)
GRESO	82.3	35.0	64.4	66.8	36.5	45.7	55.1	3.4M (3.4×)
HIVE	82.6	36.7	63.9	66.7	36.2	45.9	55.3	**2.5M (4.6×)**

Olympiad Bench [18]. Following [77], we report the checkpoint with the best average score on the six benchmarks. Appendix B gives full training details.

Baselines. We compare against Dynamic Sampling (DS) [66], a rollout-based online filter; Pre-filter [14], a fixed offline prompt filter applied before RL training; GRESO [77], a history-based selector; and PCL [14], an auxiliary prompt-curriculum selector. Since the original PCL evaluation uses a fixed wall-clock budget, whereas our main protocol fixes training at 1000 steps, we evaluate Pre-filter and PCL with a conservative two-run protocol. For the rollout budget in Table 1, we run each method once using the original PCL settings and once using our common settings, and report the higher average benchmark score.

4.2 Main Results: Accuracy Under Lower Rollout Cost

HIVE preserves accuracy while reducing rollouts. We compare HIVE against Dynamic Sampling (DS) [66], a rollout-based online filter, and GRESO [77], a history-based selector. Table 1a reports results on DAPO+MATH across five model settings. HIVE consistently uses fewer rollouts than both baselines while maintaining comparable average accuracy. On Qwen2.5-Math-7B, HIVE reduces rollouts by 70% compared with DS (13.1M to 3.9M) and achieves the highest average accuracy (59.7% vs. 57.8% for DS and 58.6% for GRESO), reaching comparable accuracy with substantially fewer generated responses.

The gains transfer across datasets and model scales. Table 2 evaluates Qwen2.5-Math-1.5B/7B on Open-R1. HIVE again keeps comparable or better average accuracy while reducing rollouts, reaching 4.6× fewer rollouts than DS on Qwen2.5-Math-7B. Figure 4 extends the comparison to Qwen2.5-14B and Qwen2.5-32B. HIVE follows a similar convergence trajectory to DS and GRESO while requiring fewer rollouts; for Qwen2.5-14B over 800 steps, HIVE uses 1.56M rollouts compared with 1.74M for GRESO and 3.45M for DS.

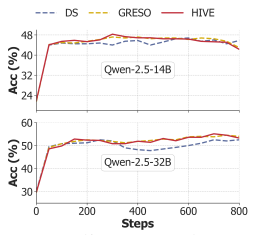


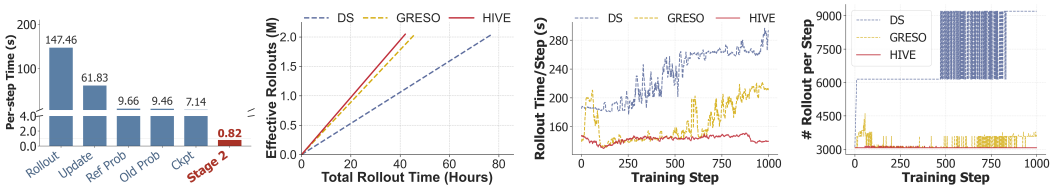
Figure 4: Scaling up to larger models, Qwen2.5-14B/32B.

Wall-clock savings come from rollout reduction, not hidden overhead. Table 1b decomposes end-to-end training time. HIVE reduces rollout time by up to 3.8× and total training time by up to 2.3× over DS. On Qwen2.5-Math-7B, total training time drops from 198.4 hours (DS) and 112.3 hours (GRESO) to 85.8 hours (HIVE), while rollout time drops from 153.7 and 66.3 to 40.2.

Performance on tool calling tasks. Besides, we also verify the effectiveness of HIVE on tool calling tasks. We train Qwen2.5-1.5B on 80% prompts samples from BFCL-V2 [25] and evaluate the performance on the remaining 20% subset [17]. As shown in Table 3, HIVE achieves the highest BFCL-V2 score while using only 1.1M rollouts instead of 5.3M. This corresponds to a 4.8× rollout reduction relative to DS. Under the same low-rollout regime, HIVE also outperforms the static Pre-filter baseline, the history-only GRESO selector, and the auxiliary PCL selector. These results suggest that moving-edge tracking is not only specific to mathematical reward signals, but also applicable on tool-calling tasks, where HIVE continues to concentrate rollout compute on informative prompts.

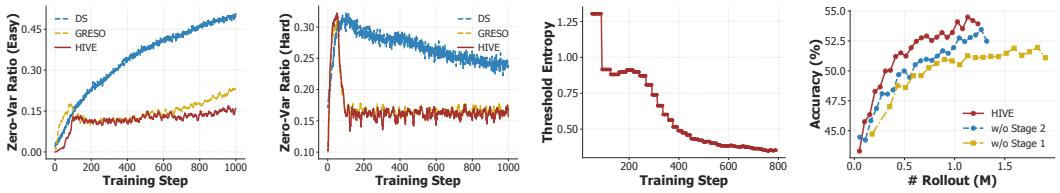
Table 3: Performance comparison (%) on tool calling tasks.

Method	BFCL-V2	#Rollout
DS	79.08	5.3M
Pre-filter	77.04	1.4M
GRESO	78.96	1.4M
PCL	76.84	1.4M
HIVE	**79.26**	**1.1M**



(a) Train-Time Breakdown (b) Efficiency Comparison (c) Generation Time (d) Rollouts per Step

Figure 5: Efficiency analysis of Qwen2.5-Math-1.5B trained on DAPO+MATH. (a) Stage 2 prompt-side verification adds little time relative to rollout generation. (b) Lines show average effective-rollout throughput using the rollout times in Table 1b. (c) HIVE maintains lower generation latency throughout training. (d) HIVE rolls out fewer prompt groups per step, concentrating compute on verified candidates.



(a) Easy zero-variance (b) Hard zero-variance (c) Entropy threshold (d) Stage ablation

Figure 6: **Zero-variance ratios and component analysis.** (a) Easy zero-variance ratio: HIVE keeps the lowest fraction of all-correct prompts throughout training. (b) Hard zero-variance ratio: HIVE rapidly drops below DS and tracks GRESO. (c) The Stage 2 median prompt-entropy threshold γ adapts as the policy becomes more confident. (d) Removing either Stage 1 or Stage 2 worsens the accuracy–rollout tradeoff.

Extension to non-binary verifiable rewards. We further train Qwen2.5-Math-1.5B with GRPO ($G = 8$) on 4×4 Sudoku, a constraint-satisfaction task with a reward adapted from Reasoning Gym [48]: an exact solution receives 1.0, and other answers receive $0.9m/16$, where m is the number of matching cells. Table 4 shows that HIVE reaches a 62.7% exact-solve rate with 1.15M rollouts, using 16.7% fewer rollouts than GRESO and 31.5% fewer than PCL, while improving the observed solve rate by 1.6 and 1.2 percentage points, respectively. These results provide additional evidence that HIVE's accuracy–cost tradeoff extends to a verifiable reward with partial credit. The comparison uses one run per method; DS is omitted due to the compute budget.

Table 4: Sudoku with non-binary rewards. Exact-solve rate (avg@4); one run per method.

Method	Exact-solve (%)	#Rollout
GRESO	61.1	1.38M
PCL	61.5	1.68M
HIVE	**62.7**	**1.15M**

4.3 Mechanism Analysis: Efficiency and Moving-Edge Adaptation

Online verification is cheap enough to run before rollouts. Figure 5a isolates the per-step overhead of Stage 2. Computing prompt-side entropy adds only 0.82 seconds, less than 0.4% of the iteration time, while the rollout phase alone takes 147.46 seconds. This supports the computational efficiency objective: HIVE gains online information without paying for response generation during selection.

HIVE improves effective rollout throughput. Figure 5b shows that HIVE accumulates effective rollouts per rollout-generation hour fastest. Figure 5c further shows that HIVE maintains low rollout-generation latency throughout training, while GRESO becomes slower later in training. This is consistent with metadata staleness: a history-only selector increasingly admits prompts that look useful in the log but no longer produce efficient current updates. Figure 5d shows that HIVE also generates fewer rollout groups per step, concentrating compute on a smaller set of verified prompts near the current edge.

HIVE lowers the observed zero-variance ratio. The zero-variance ratio measures how often selected prompts produce no informative gradient. Figures 6a and 6b show that DS wastes a large fraction of rollouts on easy zero-variance prompts as the model improves, while GRESO remains

vulnerable to stale historical records. HIVE maintains a lower zero-variance ratio for both easy and hard prompts, indicating that online verification helps track the moving edge.

4.4 Component Study: Why Both Stages Matter

The entropy gate adapts with the current policy. Figure 6c tracks the Stage 2 median threshold γ, which decreases as the model becomes more confident. This behavior is expected: as the policy improves, the prompt-side uncertainty distribution shifts, and a fixed threshold would either overprune or admit stale prompts. The median gate keeps the verification rule relative to the current candidate pool. **Both stages contribute in the reported ablation.** Figure 6d compares the full HIVE pipeline with variants that remove either stage. Without Stage 2, HIVE loses online precision and becomes more vulnerable to stale metadata. Without Stage 1, the verifier receives too many low-value candidates, reducing efficiency. The full method gives the best accuracy–rollout tradeoff, supporting the decomposition into efficient historical narrowing followed by online verification.

Ablation on Stage 2 score. We hold Stage 1 and the GRPO settings fixed and replace only the Stage 2 score with prompt length, rare-token ratio, math-symbol density, or a formatting/tokenization score. Appendix E.2 defines these controls. As shown in Table 5, the alternatives reach 43.57–45.05% average accuracy, compared with 48.80% for prompt entropy. Their rollout counts are close but not identical (2.8–3.1M), with HIVE using the most rollouts in this comparison. The observed 3.75–5.23-point gaps suggest that these surface-based rankings do not recover the entropy gate's benefit under similar rollout costs. This supports the choice of prompt entropy over the tested substitutes, without ruling out correlations between entropy and surface properties.

Table 5: Stage 2 score comparison on Qwen2.5-Math-1.5B / DAPO+MATH.

Stage 2 signal	Avg. (%)	#Rollout
Prompt length	45.05	2.9M
Rare-token ratio	44.52	3.0M
Math-symbol density	43.57	2.8M
Format/tokenization	44.66	2.9M
Prompt entropy	**48.80**	3.1M

5 Related Work

RLVR for LLM reasoning. Reinforcement learning is a standard post-training tool for aligning and improving LLMs [5, 8, 40, 42, 63]. Recent reasoning systems increasingly rely on reinforcement learning with verifiable rewards (RLVR), where correctness can be checked directly for tasks such as mathematics and code [16, 21, 24, 34, 72]. Algorithmic work improves this training loop through value-enhanced PPO variants such as VC-PPO [67] and VAPO [68], group-based methods such as RLOO [1] and GRPO [43], replay-based updates [29, 32, 71], and objective modifications such as Dr. GRPO [34] and GSPO [76]. These methods improve optimization stability or sample reuse, but they do not directly solve the pre-rollout allocation problem studied here: which prompts deserve expensive response generation at the current training step.

Prompt selection and data-efficient RL training. Data curation is essential for efficient LLM training. Offline filtering methods rank prompts before training using various heuristic metrics [11, 30, 46, 51, 55, 65, 78]. They are efficient once training begins, but they are policy-agnostic and cannot adapt as the model's competence changes. Rollout-based online methods, including Dynamic Sampling in DAPO [66] and related online down-sampling or difficulty-filtering approaches [4, 13, 33, 38, 49, 59], are more precise yet more costly, requiring more rollouts. History-based and estimator-based approaches such as GRESO [77], Bayesian task selection [7, 37, 44, 69], and curriculum estimators use past signals to reduce selection cost, but their signals become stale as training continues. Auxiliary-model methods such as DOTS [50] and PCL [14] add external or on-policy predictors for difficulty, improving adaptivity at the cost of additional compute or memory.

HIVE combines history-based filtering with online prompt verification to preserve efficiency while correcting stale selections before rollout. See Appendix D for detailed related work.

6 Conclusion

RLVR prompt selection is best understood as a moving-edge tracking problem. Useful prompts are not merely those that look difficult or uncertain in isolation; they are prompts that remain informative

for the current policy. HIVE efficiently identifies those prompts with a dual-stage mechanism: history-informed narrowing reduces the candidate pool, and prompt-side online verification filters stale candidates under the current model. Across math reasoning benchmarks and model scales, HIVE reduces rollout and training cost while preserving or even improving final performance.

7 Limitations and Broader Impacts

Limitations. Our rank-consistency analysis is intended as a conservative sufficient condition for using prompt-side entropy as an online verifier, rather than a tight characterization of every empirical prompt pair. The proof uses worst-case concentration and bounded-approximation arguments, which may yield loose margins for modern LRMs. This does not affect HIVE's algorithmic design or empirical conclusions, as the method does not rely on exact prompt-level ranking; it only requires prompt-side entropy to provide a useful low-cost signal for filtering stale candidates. This role is validated by our prompt–response entropy analysis, component ablations, and end-to-end efficiency gains. Tighter variance-dependent guarantees remain an interesting direction for future work.

Broader impacts. HIVE improves the efficiency of RL post-training by reducing unnecessary rollouts, thereby lowering training cost, energy consumption, and carbon footprint. This may make RLVR-style post-training more accessible to resource-constrained research groups. We do not foresee specific negative societal impacts beyond the general risks associated with stronger LLM capabilities.

Acknowledgments and Disclosure of Funding

This work was supported by National Key Research and Development Program of China under Grant 2022YFA1004102, and in part by the National Natural Science Foundation of China under Grants T2495254, 62502192. This work was also in part supported by grant GHX/315/22GD, the Hong Kong Research Grants Council under Theme-based Research Scheme (project no. T43-513/23-N), the NSFC/RGC Joint Research Scheme (N_PolyU5179/25), the Hong Kong Research Grants Council (PolyU25600624), the Innovation and Technology Fund (ITS/052/23MX and PRP/009/22FX), and industrial sponsors and the Research Matching Grant Scheme (P0045948, P0048183, P0048191, P0046453, and P0060272). The authors declare no competing interests.

References

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

[2] Art of Problem Solving. AIME problems and solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`, 2024.

[3] Art of Problem Solving. AMC problems and solutions. `https://artofproblemsolving.com/wiki/index.php?title=AMC_Problems_and_Solutions`, 2024.

[4] Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2026.

[5] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[6] Weizhe Chen, Sven Koenig, and Bistra Dilkina. Lspo: Length-aware dynamic sampling for policy optimization in llm reasoning. *arXiv preprint arXiv:2510.01459*, 2025.

[7] 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.

[8] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[9] Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[10] Ganqu Cui, Yuchen Zhang, Wanli Ouyang, Yu Cheng, Bowen Zhou, Ning Ding, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

[11] Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise reasoning via reinforcement learning. *arXiv preprint arXiv:2504.05185*, 2025.

[12] Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Mohammad Taher Pilehvar, Yadollah Yaghoobzadeh, and Samira Ebrahimi Kahou. Bert on a data diet: Finding important examples by gradient-based pruning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[13] Thomas Foster, Anya Sims, Johannes Forkel, Mattie Fellows, and Jakob Foerster. LILO: Learning to reason at the frontier of learnability. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[14] 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 *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL `https://openreview.net/forum?id=zqOCacBD3P`.

[15] Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[16] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 2025.

[17] Shangmin Guo, Omar Darwiche Domingues, Raphaël Avalos, Aaron Courville, and Florian Strub. Sample, predict, then proceed: Self-verification sampling for tool use of LLMs. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.

[18] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 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 (ACL)*, 2024.

[19] 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. *arXiv preprint arXiv:2103.03874*, 2021.

[20] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963.

[21] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

[22] Wenke Huang, Quan Zhang, Yiyang Fang, Jian Liang, Xuankun Rong, Huanjin Yao, Guancheng Wan, Ke Liang, Wenwen He, Mingjun Li, et al. Mapo: Mixed advantage policy optimization. *arXiv preprint arXiv:2509.18849*, 2025.

[23] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

[24] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[25] Charlie Cheng-Jie Ji, Huanzhi Mao, Fanjia Yan, Shishir G. Patil, Tianjun Zhang, Ion Stoica, and Joseph E. Gonzalez. Gorilla openfunctions v2. `https://gorilla.cs.berkeley.edu/blogs/7_open_functions_v2.html`, 2024.

[26] Hyeonji Kim, Sujeong Oh, and Sanghack Lee. Mitigating length bias in RLHF through a causal lens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

[27] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

[28] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, et al. Solving quantitative reasoning problems with language models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[29] Siheng Li, Zhanhui Zhou, Wai Lam, Chao Yang, and Chaochao Lu. Repo: Replay-enhanced policy optimization. *arXiv preprint arXiv:2506.09340*, 2025.

[30] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025.

[31] Zhicong Li, Lingjie Jiang, Yulan Hu, Xingchen Zeng, Yixia Li, Xiangwen Zhang, Guanhua Chen, Zheng Pan, Xin Li, and Yong Liu. No more stale feedback: Co-evolving critics for open-world agent learning. *arXiv preprint arXiv:2601.06794*, 2026.

[32] Jing Liang, Hongyao Tang, Yi Ma, Jinyi Liu, Yan Zheng, Shuyue Hu, Lei Bai, and Jianye Hao. Squeeze the soaked sponge: Efficient off-policy rft for large language model. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[33] Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. CPPO: Accelerating the training of group relative policy optimization-based reasoning models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[34] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Proceedings of the Second Conference on Language Modeling (COLM)*, 2025.

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the Seventh International Conference on Learning Representations (ICLR)*, 2019.

[36] Ning Lu, Shengcai Liu, Jiahao Wu, Weiyu Chen, Zhirui Zhang, Yew-Soon Ong, Qi Wang, and Ke Tang. Safe delta: Consistently preserving safety when fine-tuning LLMs on diverse datasets. In *Proceedings of Forty-second International Conference on Machine Learning (ICML)*, 2025.

[37] Yixiu Mao, Yun Qu, Qi Wang, Heming Zou, and Xiangyang Ji. Dynamics-predictive sampling for active RL finetuning of large reasoning models. *arXiv preprint arXiv:2603.10887*, 2026. URL `https://arxiv.org/abs/2603.10887`.

[38] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, et al. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

[39] Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Faster, more efficient RLHF through off-policy asynchronous learning. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[40] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[41] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[42] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[43] 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.

[44] Qianli Shen, Daoyuan Chen, Yilun Huang, Zhenqing Ling, Yaliang Li, Bolin Ding, and Jingren Zhou. BOTS: A unified framework for bayesian online task selection in llm reinforcement fine-tuning. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[45] 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 (EuroSys)*, 2025.

[46] Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*, 2025.

[47] Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in RLHF. In *Proceedings of First Conference on Language Modeling (COLM)*, 2024.

[48] Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning Gym: Reasoning environments for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2505.24760*, 2025. URL `https://arxiv.org/abs/2505.24760`.

[49] Yan Sun, Jia Guo, Stanley Kok, Zihao Wang, Zujie Wen, and Zhiqiang Zhang. Efficient reinforcement learning for large language models with intrinsic exploration. In *NeurIPS 2025 ER Workshop*, 2025.

[50] Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. Improving data efficiency for llm reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[51] Xinyu Tang, Zhenduo Zhang, Yurou Liu, Wayne Xin Zhao, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. DEPO: Towards high data efficiency in reinforcement learning with verifiable reward. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[52] Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL `https://qwenlm.github.io/blog/qwen2.5/`.

[53] Akiyoshi Tomihari. Learning dynamics in rl post-training for language models. *arXiv preprint arXiv:2601.04670*, 2026.

[54] Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, et al. A comprehensive survey in llm(-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585*, 2025.

[55] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[56] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qing Li, and Ke Tang. Disentangled contrastive learning for social recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*, 2022.

[57] Jiahao Wu, Wenqi Fan, Jingfan Chen, Shengcai Liu, Qijiong Liu, Rui He, Qing Li, and Ke Tang. Condensing pre-augmented recommendation data via lightweight policy gradient estimation. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[58] Jiahao Wu, Qijiong Liu, Hengchang Hu, Wenqi Fan, Shengcai Liu, Qing Li, Xiao-Ming Wu, and Ke Tang. Leveraging chatgpt to empower training-free dataset condensation for content-based recommendation. In *Companion Proceedings of the ACM on Web Conference 2025*, 2025.

[59] Yixuan Even Xu, Yash Savani, Fei Fang, and J Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *Transactions on Machine Learning Research*, 2026. URL `https://openreview.net/forum?id=MfHOmgqVXM`.

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

[61] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[62] Yanting Yang, Minghao Chen, Qibo Qiu, Jiahao Wu, Wenxiao Wang, Binbin Lin, Ziyu Guan, and Xiaofei He. Adapt2reward: Adapting video-language models to generalizable robotic rewards via failure prompts. In *Computer Vision – ECCV 2024*, 2025.

[63] Yanting Yang, Shenyuan Gao, Qingwen Bu, Li Chen, and Dimitris N. Metaxas. Seeing farther and smarter: Value-guided multi-path reflection for VLM policy optimization. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[64] Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*, 2025.

[65] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. In *Proceedings of the Second Conference on Language Modeling (COLM)*, 2025.

[66] 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. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[67] Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025.

[68] 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.

[69] Yongcheng Zeng, Zexu Sun, Bokai Ji, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Haifeng Zhang, Xu Chen, and Jun Wang. CurES: From gradient analysis to efficient curriculum learning for reasoning llms. In *Proceedings of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

[70] Enci Zhang, Xingang Yan, Wei Lin, Tianxiang Zhang, and Lu Qianchun. Learning like humans: Advancing llm reasoning capabilities via adaptive difficulty curriculum learning and expert-guided self-reformulation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

[71] Hongzhi Zhang, Jia Fu, Jingyuan Zhang, Kai Fu, Qi Wang, Fuzheng Zhang, and Guorui Zhou. Rlep: Reinforcement learning with experience replay for llm reasoning. *arXiv preprint arXiv:2507.07451*, 2025.

[72] Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, Shimiao Jiang, Shiqing Kuang, Shouyu Yin, Chaohang Wen, Haotian Zhang, Bin Chen, and Bing Yu. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*, 2025.

[73] Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*, 2023.

[74] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

[75] Changmeng Zheng, Dayong Liang, Wengyu Zhang, Xiao-Yong Wei, Tat-Seng Chua, and Qing Li. A picture is worth a graph: A blueprint debate paradigm for multimodal reasoning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 419–428, 2024.

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

[77] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[78] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Appendix

In this appendix, we provide the rank-consistency analysis for Theorem 3.1 (Appendix A), experimental settings (Appendix B), preliminary-study details (Appendix C), detailed related work (Appendix D), additional experiments (Appendix E), the HIVE algorithm (Appendix F), and asset licenses (Appendix G).

A Conditional Rank-Consistency Analysis

This section proves the conditional statement in Theorem 3.1. The result is meant to formalize the intuition behind Stage 2 under simplified assumptions. The main empirical support for the online verification part of HIVE comes from the correlation and ablation results reported in the paper.

A.1 Observable Quantities and Sampling

Fix a policy checkpoint θ and sampling policy $q(\cdot \mid x)$. Let $\mathcal{V}$ be the finite vocabulary and use natural logarithms. For a prompt with $L_x \geq 2$ scored positions, define

$$V(x) = \frac{1}{L_x - 1} \sum_{\ell=2}^{L_x} \mathcal{H}(p_\theta(\cdot \mid x_{<\ell})) \,. \tag{10}$$

For n independent rollouts $y^{(r)} \sim q(\cdot \mid x)$ with nonempty responses of lengths L_r, define

$$\begin{aligned} U^{(r)}(x) &= \frac{1}{L_r} \sum_{\ell=1}^{L_r} \mathcal{H}\Big(p_\theta(\cdot \mid x, y^{(r)}_{<\ell})\Big) \,, \\ U^*(x) &= \mathbb{E}_{y \sim q(\cdot|x)}[U^{(r)}(x)], \qquad \hat{U}_n(x) = \frac{1}{n} \sum_{r=1}^{n} U^{(r)}(x). \end{aligned} \tag{11}$$

The scoring distribution p_θ uses a fixed temperature convention, held constant within each comparison.

A.2 Explicit Calibration Assumption

To relate prompt-side predictive entropy $V(x)$ to expected response-side predictive entropy $U^*(x)$, we impose the following bounded affine calibration assumption for a fixed policy checkpoint and a specified prompt set. This assumption is meant to capture a linear relationship between the two quantities up to a bounded error, in part verified in the binned linearity shown in Figure 7, empirically.

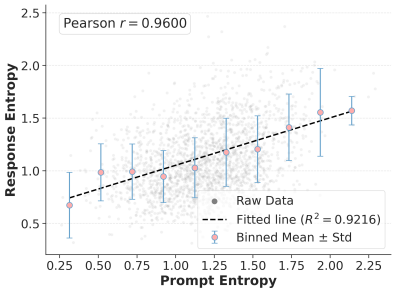


Figure 7: **Binned prompt–response entropy association.** The plot summarizes 2,048 prompts; grey points are individual observations and blue points are bin means with standard deviations. The reported $r = 0.9600$ and $R^2 = 0.9216$ describe the fitted bin means.

Assumption A.1 (Bounded affine calibration)**.** For the fixed checkpoint and a specified prompt set $\mathcal{X}$, there exist constants $a > 0$, $b \in \mathbb{R}$, and $\varepsilon_{\text{cal}} \geq 0$ such that

$$\begin{aligned} &U^*(x) = aV(x) + b + e(x), \\ &|e(x)| \leq \varepsilon_{\text{cal}} \quad \text{for every } x \in \mathcal{X}. \end{aligned} \tag{12}$$

This is an assumption about prompt and response entropies.

A.3 Finite-Rollout Error

Lemma A.2 (Concentration across independent rollouts)**.** *For a fixed prompt x and $\eta > 0$,*

$$\Pr\Big(|\hat{U}_n(x) - U^*(x)| > \eta\Big) \leq 2 \exp\left(-\frac{2n\eta^2}{(\log|\mathcal{V}|)^2}\right). \tag{13}$$

Proof. Each token-distribution entropy lies in $[0, \log|\mathcal{V}|]$, so each response average lies in the same interval. Hoeffding's inequality [20] applies to the n independent response averages. □

A.4 Pairwise Rank Consistency

Theorem A.3 (Conditional pairwise ordering)**.** *Fix two prompts $x, x' \in \mathcal{X}$ before drawing their diagnostic rollouts. Under Assumption A.1, for any $\delta \in (0,1)$, define*

$$\eta_n(\delta) = \log|\mathcal{V}| \sqrt{\frac{\log(4/\delta)}{2n}}. \tag{14}$$

With probability at least $1-\delta$,

$$\left| [\hat{U}_n(x) - \hat{U}_n(x')] - a[V(x) - V(x')] \right| \le 2\varepsilon_{\mathrm{cal}} + 2\eta_n(\delta). \tag{15}$$

Consequently, if $a|V(x) - V(x')| > 2\varepsilon_{\mathrm{cal}} + 2\eta_n(\delta)$, then on the same event

$$\mathrm{sign}(\hat{U}_n(x) - \hat{U}_n(x')) = \mathrm{sign}(V(x) - V(x')). \tag{16}$$

For the expected entropies $U^*(x), U^*(x')$, the sufficient condition is $a|V(x) - V(x')| > 2\varepsilon_{\mathrm{cal}}$.

Proof. Applying Lemma A.2 at tolerance $\eta_n(\delta)$ gives failure probability at most $\delta/2$ for each prompt. A union bound therefore yields, with probability at least $1-\delta$,

$$|[\hat{U}_n(x) - \hat{U}_n(x')] - [U^*(x) - U^*(x')]| \le 2\eta_n(\delta).$$

The calibration assumption gives

$$|[U^*(x) - U^*(x')] - a[V(x) - V(x')]| = |e(x) - e(x')| \le 2\varepsilon_{\mathrm{cal}}.$$

The triangle inequality proves Eq. (15). A margin exceeding this error preserves the sign because $a > 0$. Omitting the sampling error gives the assertion for expected entropies. □

B Detailed Experimental Setting for Main Study

This section gives the full experimental protocol for the main study. We follow the general setup of GRESO [77] and make three aspects explicit: the shared training and evaluation protocol, the budget-matched baseline comparison, and the HIVE-specific selector configuration.

B.1 Shared Training and Evaluation Protocol

Models and datasets. We evaluate Qwen2.5-Math-1.5B [60], DeepSeek-R1-Distill-Qwen-1.5B [16], Qwen3-4B-Base [61], Qwen2.5-Math-7B [60], and Llama-3.2-3B-Instruct [15]. We additionally evaluate Qwen2.5-14B/32B [52] in the scaling study. The context length is 4096 for Qwen2.5-Math-1.5B/7B, 8192 for DeepSeek-R1-Distill-Qwen-1.5B, Qwen3-4B-Base, and Llama-3.2-3B-Instruct, and 16384 for Qwen2.5-14B/32B. Training uses two math-reasoning corpora: DAPO+MATH, which combines DAPO [66] and MATH [19], and a 30,000-example Open-R1 math subset [23].

Optimization and rollout. Our implementation is based on verl [45] and vLLM [27]. We train the main models on 8×A100 GPUs, use rollout sampling temperature 1.0, and sample $G = 8$ responses per prompt. All models are trained for 1000 steps with AdamW [35], learning rate 10^{-6}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 0.01. Following Yu et al. [66], we remove the KL-divergence term. The actor is trained with Fully Sharded Data Parallel (FSDP) [74].

For Qwen2.5-Math-1.5B/7B, Qwen3-4B-Base, and Llama-3.2-3B-Instruct, the final training batch contains $B_{\mathrm{t}} = 256$ prompts, the post-gate rollout target is $B_{\mathrm{cand}} = 384$, and the mini-batch size is 512. For DeepSeek-R1-Distill-Qwen-1.5B, we use $B_{\mathrm{t}} = 128$, $B_{\mathrm{cand}} = 192$, and mini-batch size 512. The larger Qwen2.5-14B/32B scaling runs use the same optimizer, sampling temperature, and selector hyperparameters, with the longer 16384-token context length noted above.

Rewards and evaluation. For reward assignment, we give a score of 0.1 when an answer can be successfully extracted and 1.0 when the extracted answer is correct. We evaluate every 50 steps on Math500 [19], AIME24 [2], AMC [3], Minerva Math [28], Gaokao [73], and Olympiad Bench [18]. Following Zheng et al. [77], we report the checkpoint with the best average performance across the six benchmarks.

Prompt template. All math-reasoning rollouts use the following template.

Question Template

Please solve the following math problem: {{Question Description}}. The assistant first thinks about the reasoning process step by step and then provides the user with the answer. Return the final answer in \boxed{} tags, for example \boxed{1}. Let's solve this step by step.

B.2 Baselines and Budget Matching

We compare HIVE with four prompt-selection baselines. Dynamic Sampling (DS) [66] performs rollout-based online filtering and serves as the full-cost online baseline. Pre-filter is a fixed offline selector: prompts are filtered before RL training, and the selected subset remains unchanged during policy updates. GRESO [77] is a history-based selector using past reward trajectories. PCL [14] uses an auxiliary prompt-curriculum selector.

For each model family, GRESO, Pre-filter, and PCL are run under the rollout budgets shown in Table 1. Because the original PCL paper evaluates Pre-filter and PCL under a fixed wall-clock budget, while our main comparison fixes training at 1000 steps, we evaluate these two baselines with a matched-rollout two-run protocol. One run follows the original PCL-paper settings, and the other follows our common training settings; we report the higher average benchmark score across the two runs. Selector-specific runtime overhead is included in the "Other" column of Table 1b.

B.3 HIVE-Specific Hyperparameters and Initialization

Table 6 lists the selector hyperparameters used by HIVE, their notation in Section 3 and Algorithm 1, their deployed values, and their roles. Unless a row explicitly depends on B_{t}, the same value is used across the main-result models and the Qwen2.5-14B/32B scaling study.

Initialization. The data profiler storing $\mathcal{T} = \{\mathcal{T}_i\}$ is initialized as an empty dictionary, so a prompt x_i has no history until its first rollout. For an unseen prompt, $z_i^{(t)}{=}0$, hence $s_{\mathrm{rew}}(x_i){=}1$ and Stage 1 accepts it with probability 1. Thus, every prompt receives one free visit before history-based filtering applies. We initialize $p_{e,\mathrm{easy}}^{(0)}{=}p_{e,\mathrm{hard}}^{(0)}{=}0.5$; the larger hard target $\alpha_{\mathrm{hard}}{=}2\alpha/3$ then retains hard zero-variance prompts more aggressively. In our main runs, typical steady-state values are $p_{e,\mathrm{easy}}{\approx}0.05{-}0.15$ and $p_{e,\mathrm{hard}}{\approx}0.6{-}0.9$. Stage 2 is stateless and needs no initialization. The selector uses the same global random seed as the verl trainer (42 unless specified otherwise), making the Stage 1 Bernoulli draws deterministic given a checkpoint.

Exact batch-sizing protocol. A GRPO update consumes exactly $B_{\mathrm{t}} \cdot G$ rollouts. HIVE obtains this fixed-size update batch with the following three-step accumulation procedure, which instantiates Algorithm 1.

1. *Candidate accumulation.* Pull $b_{\mathrm{raw}}{=}32$ prompts from the dataloader, apply the Stage 1 Bernoulli filter with acceptance probability $P_{\mathrm{S1}}(x_i)$, then apply the Stage 2 entropy band gate: sort the survivors by $V_t(x)$ in descending order, drop the top k_{off} fraction, keep the next k_{keep} fraction, and round the kept count down to a multiple of G. Append the kept prompts to $\mathcal{C}_t$ until $|\mathcal{C}_t| \geq B_{\mathrm{cand}}^{\mathrm{adapt}}$, initialized as B_{cand}.
2. *Rollout filtering.* Sample G responses for every $x \in \mathcal{C}_t$, score them, discard groups whose reward variance is exactly zero, and append the surviving prompt-response groups to $\mathcal{R}_t$.
3. *Top-up and slicing.* Let $M_t = B_t - |\mathcal{R}_t|/G$ be the number of missing complete prompt groups. When $M_t > 0$, update the post-gate rollout target as

$$B_{\mathrm{cand}}^{\mathrm{adapt}} \leftarrow \begin{cases} \max\{b_{\min}, \min\{B_{\mathrm{cand}}, \lceil \eta M_t/(1-\hat{\rho}_{\mathrm{zv}})\rceil\}\}, & \hat{\rho}_{\mathrm{zv}} < 1, \\ B_{\mathrm{cand}}, & \hat{\rho}_{\mathrm{zv}} = 1. \end{cases} \tag{17}$$

   Here $\hat{\rho}_{\mathrm{zv}}$ is measured over the latest rollout round. Repeat candidate accumulation and generation until enough non-zero-variance groups are available. Then keep the first B_t complete groups in arrival order and pass their $B_t G$ responses to GRPO. The slice is taken

Table 6: Complete list of HIVE-specific hyperparameters used in the main experiments. Defaults that we never override are marked with †.

Component	Symbol	Value	Description
Stage 1: history-informed selection			
Reward / entropy mixing weight	λ	0.9	Weight on the reward-history score in Eq. (6); we set λ=0.9 in deployed runs and rely on Stage 2 to enforce the entropy criterion.
Initial easy exploration prob.	$p^{(0)}_{e,\text{easy}}$	0.5	Base retention probability for prompts whose recent visits were all-correct zero-variance groups.
Initial hard exploration prob.	$p^{(0)}_{e,\text{hard}}$	0.5	Base retention probability for prompts whose recent visits were all-incorrect zero-variance groups.
Probability step size	Δp	0.01	Per-step additive update applied to $p_{e,\tau}$ in Eq. (4).
Lower clip on $p_{e,\tau}$	$p_{\min}$	$0.05^{\dagger}$	Floor of the projection range $[p_{\min}, p_{\max}]$ in Eq. (4).
Upper clip on $p_{e,\tau}$	$p_{\max}$	$0.95^{\dagger}$	Ceiling of the projection range $[p_{\min}, p_{\max}]$ in Eq. (4).
Reward-history score floor	ϵ_p	0.01	Per-prompt lower bound on $s_{\text{rew}}(x_i)$ in Eq. (3), applied before mixing with the response-entropy score.
Total target zero-variance ratio	α	0.25	Per-step zero-variance budget that drives the adaptive update of $p_{e,\tau}$.
Easy target ratio	α_{easy}	$\alpha/3$=0.083	Targeted fraction of all-correct zero-variance groups.
Hard target ratio	α_{hard}	$2\alpha/3$=0.167	Targeted fraction of all-incorrect zero-variance groups; the larger hard target retains more prompts that may enter the learnable region as the policy improves.
Easy classification threshold	r_{easy}	1.0	A reward group is counted as easy zero-variance iff every one of its G rewards equals 1.0.
Hard classification threshold	r_{hard}	0.1	A reward group is counted as hard zero-variance iff every one of its G rewards equals 0.1 (extracted but wrong).
History-trace size	$\lvert\mathcal{T}_i\rvert$	unbounded	Each prompt keeps the full sequence of past visits; only the trailing zero-variance run length $z_i^{(t)}$ enters Eq. (3).
Stage 2: online entropy verification			
Entropy keep ratio	k_{keep}	0.5	Fraction of candidates retained by the entropy gate in Eq. (9).
Entropy upper-trim ratio	k_{off}	0	Fraction of the highest-entropy candidates.
Batch sizing			
Final training batch (prompts)	B_{t}	256 / 128	256 for Qwen2.5-Math-1.5B/7B, Qwen3-4B-Base, and Llama-3.2-3B-Instruct; 128 for DeepSeek-R1-Distill-Qwen-1.5B.
Post-gate rollout target (prompts)	B_{cand}	384 / 192	Equal to 1.5 B_{t} across these configurations.
Group size (responses/prompt)	G	8	Number of responses sampled per prompt at rollout.
Dataloader micro-batch (prompts)	b_{raw}	32	Prompts pulled per inner iteration of the selector loop.
Adaptive-resample lower bound	$b_{\min}$	$64^{\dagger}$	Floor on the adaptive resample size used by the top-up loop.
Adaptive-resample safety factor	η	$1.25^{\dagger}$	Safety multiplier on the predicted resample size in the top-up loop.

in arrival order, which is uniform over surviving rollouts because dataloader fetches are independent and Stage 1/2 acceptance is conditionally independent of buffer position.

If accumulation terminates, this protocol supplies a B_t-prompt update. The number of raw prompts inspected depends on Stage 1 acceptance, the entropy gate and its rounding, and repeated generation caused by zero-variance filtering.

C Experimental Settings for Preliminary Analysis

To derive the empirical observations regarding prompt utility and metadata staleness presented in Section 2, we conducted a focused study using the following setup.

Model and Dataset. We used Qwen2.5-Math-1.5B [60] as the base policy model. The model was trained on the DAPO+MATH dataset [66].

Training Configuration. The model was trained using GRPO [43] with a group size of $G = 8$. To capture the diversity required for entropy analysis, we set the sampling temperature to $T = 1.0$. Training was conducted on 8×NVIDIA A100 (80GB) GPUs.

Metric Definitions.

- **Effective prompt ratio.** A prompt is effective at a training step if its rollout group has non-zero reward variance and therefore yields a non-vanishing GRPO advantage. The effective prompt ratio is the fraction of selected prompts that are effective under the current policy.
- **Gradient norm (utility).** We quantify the learning signal of a prompt by the L_2 norm of its policy gradient with respect to the last layer, normalized by response length. This serves as a proxy for the update magnitude contributed by the prompt.
- **Empirical difficulty.** We estimate prompt difficulty from rollout accuracy. For a prompt x evaluated over a window $\mathcal{W}$, we compute

$$A_{\mathcal{W}}(x) = \frac{1}{G|\mathcal{W}|} \sum_{t \in \mathcal{W}} \sum_{i=1}^{G} \mathbb{I}(r_{i,t} = 1). \tag{18}$$

  Low accuracy corresponds to hard prompts, while high accuracy corresponds to easy prompts.
- **Response-side entropy.** For each prompt, we compute the average token-level entropy of generated responses and then average over the G rollouts.

Figure 2a: effective-prompt retention. We compare a history-only selector with a history-plus-online selector. The history-only selector ranks candidates using cached reward and response-entropy metadata from previous training steps. The history-plus-online selector uses the same historical candidate source but revalidates candidates under the current policy before rollout. We report the effective prompt ratio throughout training.

Figure 2b: high-entropy non-zero-gradient prompts. To isolate utility differences among prompts that already pass zero-variance filtering, we compare DS with a diagnostic DS+High-Entropy variant. DS+High-Entropy prioritizes prompts with higher response-side entropy among prompts with non-zero gradients. We plot evaluation accuracy against the cumulative number of non-zero-gradient rollouts, so the comparison is made under matched informative-rollout budgets.

Figure 2c: learning edge. We bin prompts by empirical difficulty and response-side entropy, using five bins for each axis. For every bin, we report the mean length-normalized gradient norm. The heatmap shows that utility is largest when prompts are neither trivial nor intractable and still induce high response-side uncertainty.

Figure 2d: moving of the utility. We sample 515 non-zero reward variance prompts observed at steps 500 or 1000. Prompts with zero reward variance are assigned to the Zero class. The remaining non-zero-gradient prompts are split into High and Low utility classes by their length-normalized gradient norm at each step. The Sankey diagram reports transitions among these classes, quantifying how historical utility labels become stale as the policy evolves.

D Detailed Related Work

RL for LLM reasoning. Reinforcement learning has become a key technique for post-training LLMs [5, 8]. Early RLHF work [40, 42] used human preference signals to improve instruction following and alignment. More recent RLVR systems replace learned reward models with verifiable task rewards, producing large gains in mathematics and programming [16, 21, 24, 34, 72]. Building on RLVR, algorithmic work has improved PPO-style and GRPO-style training. VC-PPO [67] and VAPO [68] strengthen value-function learning, while RLOO [1] and GRPO [43] avoid explicit value learning with multi-sample baselines. Other studies introduce replay mechanisms [29, 32, 36, 71] or adjust optimization objectives to improve stability and reduce bias [9, 22, 34, 58, 62, 76]. These

works are complementary to HIVE: they improve how selected rollouts update the policy, whereas HIVE improves which prompts receive rollouts before the update.

Data-efficient LLM training. Prompt and trajectory quality strongly affect LLM training efficiency. Offline curation follows the "less is more" principle from supervised fine-tuning [54, 57, 65, 75, 78] and reinforced fine-tuning [11, 30, 46, 51, 55, 56]. These approaches can reduce dataset size before training, but they cannot respond to the model's changing competence during RLVR.

Online selection addresses this limitation by conditioning selection on the current policy. Rollout-based methods discard uninformative prompts via real-time sampling [13, 33, 38, 49, 59, 66] or prioritize medium-difficulty prompts [4]. They are online and precise, but they require repeated response generation before filtering. Estimator-based methods instead maintain historical logs [77] or Bayesian utility estimates [7, 44, 69]. These methods are efficient, but stale metadata can misrepresent the current policy's learning edge. Auxiliary-model approaches use external scorers or jointly trained predictors to estimate difficulty, as in DOTS [50] and PCL [14]; these improve adaptivity but add extra computation or memory. Stage-wise curriculum refreshes [70, 72] periodically re-estimate difficulty, but they still do not provide cheap per-step online verification.

HIVE is motivated by the observation that none of these categories simultaneously satisfy the online, and efficient requirements. Its history-informed stage inherits the low overhead of historical filtering, while its prompt-side entropy verifier corrects stale metadata without generating responses.

E Additional Experiments

E.1 Correlation Between Prompt Entropy and Response Entropy

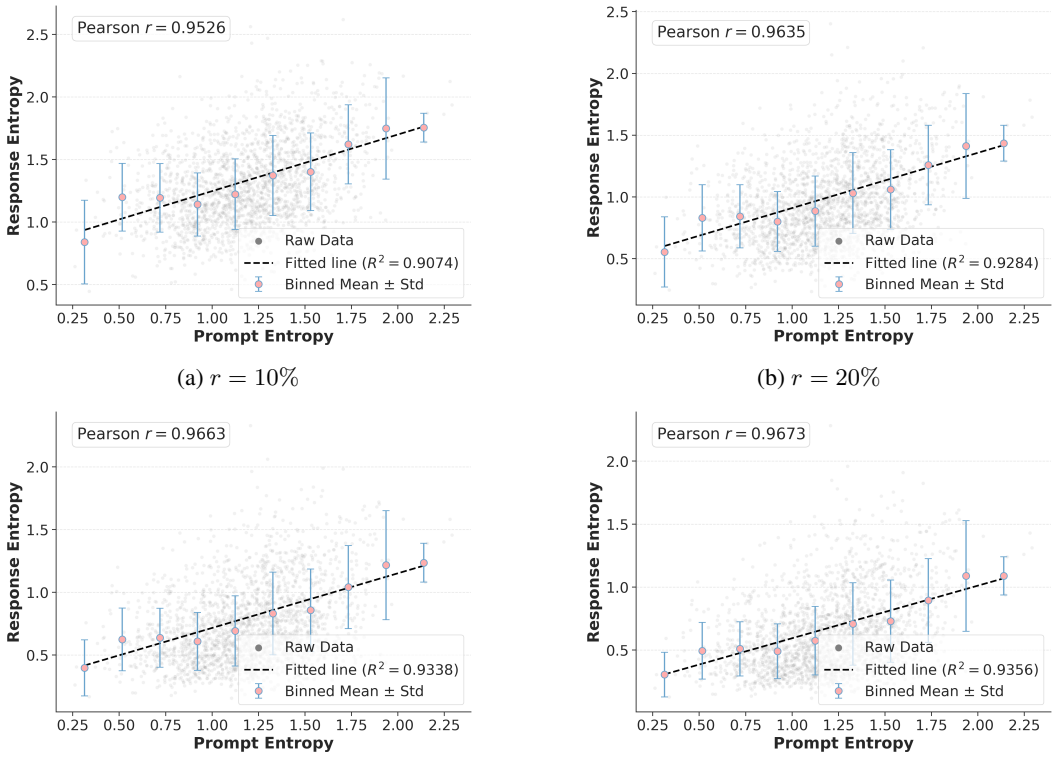
Building on the empirical validation of assumption in Section A, we conduct an additional robustness study by redefining the response entropy $S_{\text{resp}}(x)$ as the entropy computed over only the top-$r\%$ tokens of the response distribution, sweeping $r \in \{10, 20, 30, 40\}$. To suppress intrinsic decoding stochasticity, we follow the same binned-mean protocol on 2,048 prompts, binning by $S_{\text{prompt}}(x)$ (proxied by $V(x)$) and averaging $S^{(r)}_{\text{resp}}(x)$ within each bin. As shown in Figure 8, the prompt–response relationship remains linear across all ratios. This indicates that the prompt–response entropy relationship is robust to this response-entropy definition, further supporting the bounded-noise linear approximation used in Section A.

E.2 Surface-Feature Controls for Stage 2

Table 5 compares alternative Stage 2 scores on Qwen2.5-Math-1.5B trained on DAPO+MATH. Each control keeps Stage 1 and the GRPO configuration fixed and replaces only the score used by the entropy gate. We report one run per control. The four scores are defined as follows:

- **Prompt length:** the attention-masked token count actually seen by the model after applying the chat template and left truncation to a maximum of 1,536 tokens.
- **Rare-token ratio:** the fraction of content tokens whose token types belong to the rarest types covering the bottom 10% of training-corpus token mass. The realized cutoff is a corpus count of at most 637.
- **Math-symbol density:** the number of characters in a predefined set of mathematical, Greek, and set-notation symbols, together with characters in LaTeX commands matching `\\[A-Za-z]+`, divided by the decoded character count.
- **Formatting/tokenization:** the equal-weight mean of six features after batch-wise z-score normalization: tokens-per-word fragmentation and the densities of punctuation, whitespace, newlines, non-ASCII characters, and code fences.

These controls use surface statistics without a current-policy forward pass. Input perplexity is a distinct online alternative that requires policy scoring; it is not evaluated in this comparison.



(a) $r = 10\%$ (b) $r = 20\%$

(c) $r = 30\%$ (d) $r = 40\%$

Figure 8: Relationship between prompt entropy and response entropy computed over the top-$r\%$ tokens in the response distribution, sweeping $r \in \{10, 20, 30, 40\}$. Each panel reports the binned mean trend under the corresponding ratio setting.

E.3 Case Study

To gain deeper insight into the behavior of our selective filtering algorithm, we analyze a case study based on prompts from the MATH dataset. We divide these prompts into four categories: Frequently Skipped Prompts (Easy), Frequently Skipped Prompts (Hard), Frequently Selected Prompts, and Prompts Frequently Selected by Stage 1 but Skipped by Stage 2. We observe that frequently skipped easy prompts often involve simple computations or routine use of formulas, leading to high solution rates across sampled responses. In contrast, hard prompts that are often skipped tend to be too difficult for the model, leading to low or zero success across rollouts, limiting their value for training. Frequently selected prompts tend to be moderately difficult, making consistent contributions to model learning. Furthermore, we analyze the prompts selected in stage 1 but filtered after stage 2. Most prompts are relatively easy, showing that the historical information is not reliable for filtering.

Frequently Skipped Prompts (Easy)

1. **Question:** What is the value of $(2x + 5)^2$ when $x = 3$? **Solution:** 121.
2. **Question:** Solve for x: $5^{x+4} = 125^x$? **Solution:** 2.
3. **Question:** A squirrel travels at a constant 4 miles per hour. How long does it take for this squirrel to travel 1 mile? Express your answer in minutes. **Solution:** 15.
4. **Question:** Evaluate $\log_5 625$. **Solution:** 4.
5. **Question:** What value of x will give the minimum value for $x^2 - 10x + 24$? **Solution:** 5.

Frequently Skipped Prompts (Hard)

1. **Question:** Let $\triangle PQR$ be a right triangle with $PQ = 90$, $PR = 120$, and $QR = 150$. Let C_1 be the inscribed circle. Construct $\overline{ST}$ with S on $\overline{PR}$ and T on $\overline{QR}$, such that $\overline{ST}$ is perpendicular to $\overline{PR}$ and tangent to C_1. Construct $\overline{UV}$ with U on $\overline{PQ}$ and V on $\overline{QR}$ such that $\overline{UV}$ is perpendicular to $\overline{PQ}$ and tangent to C_1. Let C_2 be the inscribed circle of $\triangle RST$ and C_3 the inscribed circle of $\triangle QUV$. The distance between the centers of C_2 and C_3 can be written as $\sqrt{10n}$. What is n? **Solution:** 725.

2. **Question:** Consider the region A in the complex plane that consists of all points z such that both $\frac{z}{40}$ and $\frac{40}{\bar{z}}$ have real and imaginary parts between 0 and 1, inclusive. Find the area of A. **Solution:** $1200 - 200\pi$.

3. **Question:** Ephram is growing 3 different variants of radishes in a row of 13 radishes total, but he forgot where he planted each radish variant and he can't tell what variant a radish is before he picks it. Ephram knows that he planted at least one of each radish variant, and all radishes of one variant will form a consecutive string, with all such possibilities having an equal chance of occurring. He wants to pick three radishes to bring to the farmers market, and wants them to all be of different variants. Given that he uses optimal strategy, the probability that he achieves this can be expressed as $\frac{m}{n}$, where m and n are relatively prime positive integers. Find $m + n$. **Solution:** 17.

4. **Question:** There are 1000 rooms in a row along a long corridor. Initially, the first room contains 1000 people, and the remaining rooms are empty. Each minute, the following happens: for each room containing more than one person, someone in that room decides it is too crowded and moves to the next room. All these movements are simultaneous (so nobody moves more than once within a minute). After one hour, how many different rooms will have people in them? **Solution:** 31.

5. **Question:** A regular hexagon has side length 6. Congruent arcs with radius 3 are drawn with the center at each of the vertices, creating circular sectors as shown. The region inside the hexagon but outside the sectors is shaded as shown. The original answer is in the form $a\sqrt{b} - c\pi$. Please find the value of a + b + c. **Solution:** 75.

Frequently Selected Prompts After Stage 2

1. **Question:** Let z be a complex number such that

$$|z - 12| + |z - 5i| = 13.$$

Find the smallest possible value of $|z|$. **Solution:** $\frac{60}{13}$.

2. **Question:** Find all real solutions to $x^4 + (2 - x)^4 = 34$. Enter all the solutions, separated by commas. **Solution:** $1 + \sqrt{2}, 1 - \sqrt{2}$.

3. **Question:** Let b be a real number randomly selected from the interval $[-17, 17]$. Then, m and n are two relatively prime positive integers such that mn is the probability that the equation $x^4 + 25b^2 = (4b^2 - 10b)x^2$ has *at least* two distinct real solutions. Find the value of $m + n$. **Solution:** 63.

4. **Question:** Let a, b, c, d be real numbers such that

$$\frac{(a - b)(c - d)}{(b - c)(d - a)} = \frac{2}{5}.$$

Find the sum of all possible values of

$$\frac{(a - c)(b - d)}{(a - b)(c - d)}.$$

Solution: $-\frac{3}{2}$.

5. **Question:** The function f, defined on the set of ordered pairs of positive integers, satisfies the following properties:

$$\begin{aligned} f(x,x) &= x, \\ f(x,y) &= f(y,x), \quad \text{and} \\ (x+y)f(x,y) &= yf(x,x+y). \end{aligned}$$

Calculate $f(14,52)$. **Solution:** 364.

Prompts Frequently Selected by Stage 1 but Skipped by Stage 2

1. **Question:** A polynomial with integer coefficients is of the form

$$9x^4 + a_3x^3 + a_2x^2 + a_1x + 15 = 0.$$

Find the number of different possible rational roots of this polynomial. **Solution:** 16.

2. **Question:** When an integer is divided by 15, the remainder is 7. Find the sum of the remainders when the same integer is divided by 3 and by 5. **Solution:** 3.

3. **Question:** Find the number of positive divisors of 2002. **Solution:** 16.

4. **Question:** What is the base five product of the numbers 121_5 and 11_5? **Solution:** 1331.

5. **Question:** If $\sin x = 3\cos x$, then what is $\sin x \cos x$ **Solution:** $\frac{3}{10}$.

F Algorithm

Algorithm 1 expands the fixed-size training-batch procedure in Appendix B.3. The target $B_{\text{cand}} = 1.5B_t$ counts prompts *after* both selection stages. Within each raw micro-batch, Stage 1 performs Bernoulli selection using the historical acceptance probability P_{S1} in Eq. (6). Stage 2 ranks the survivors by current prompt entropy, applies the keep rule and rounding specified in Appendix B.3, and appends the retained prompts to $\mathcal{B}_t$. Candidate accumulation continues until the current post-gate target is reached.

Zero-variance filtering and top-up. Each generation round produces G responses per selected prompt. A group is admitted to the response buffer $\mathcal{R}_t$ only when its reward variance is nonzero; groups remain intact and retain their arrival order. If fewer than B_t complete groups have accumulated, HIVE estimates the zero-variance ratio from the latest generation round, updates the next post-gate target using Eq. (17), and repeats selection and generation with the same policy. When the latest round is entirely zero-variance, the next target is reset to B_{cand}, avoiding division by zero.

Slicing and policy update. Once enough groups are available, HIVE takes the first B_t complete groups in arrival order and performs one GRPO update on their B_tG responses. Excess groups are not used in this update. After the policy update, the history and exploration statistics are refreshed using $\mathcal{A}_t$, which records every generated group across all rounds, including zero-variance and excess groups. All generated responses count toward rollout cost, regardless of whether they enter $\mathcal{R}_{\text{train}}$.

A complete list of every selector hyperparameter (including λ, α, Δp, ϵ_p, $p_{\min}/p_{\max}$, k_{keep}, k_{off}, the trace and unseen-prompt initialization, and the exact batch-sizing protocol that produces a B_{t}-prompt GRPO update) is provided in Appendix B.3.

Algorithm 1 Training Iteration in HIVE

Require: Dataset $\mathcal{D}$; policy π_θ; history $\mathcal{T}$; training batch size B_t (prompts); group size G; post-gate rollout target $B_{\text{cand}} = 1.5B_t$; raw micro-batch size b_{raw}; resampling parameters $b_{\min}, \eta$; selector settings in Appendix B.3.

$\mathcal{R}_t \leftarrow [\,]$ (response buffer); $\mathcal{A}_t \leftarrow [\,]$ (all generated groups)
$B_{\text{cand}}^{\text{adapt}} \leftarrow B_{\text{cand}}$
repeat
  $\mathcal{B}_t \leftarrow [\,]$ *# Fresh post-gate prompt buffer*
  repeat
    Draw a raw micro-batch $\mathcal{X}$ of b_{raw} prompts from $\mathcal{D}$
    $\mathcal{S} \leftarrow \{x_i \in \mathcal{X} : \mathcal{T}_i^{(t)} = \varnothing\}$ {Unseen: bypass Stage 1}
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x_i \in \mathcal{X} : \mathcal{T}_i^{(t)} \neq \varnothing,\ \text{Bernoulli}(P_{\text{S1}}(x_i)) = 1\}$
    Compute $V_t(x)$ for $x \in \mathcal{S}$ and sort by decreasing entropy
    Retain the top $G\lfloor k_{\text{keep}}|\mathcal{S}|/G \rfloor$ prompts ($k_{\text{off}} = 0$)
    Append the retained prompts to $\mathcal{B}_t$
  until $|\mathcal{B}_t| \geq B_{\text{cand}}^{\text{adapt}}$
  # Generate and filter complete prompt-response groups
  $\mathcal{W} \leftarrow$ generate G responses per prompt in $\mathcal{B}_t$; score rewards and entropies
  Append all groups in $\mathcal{W}$ to $\mathcal{A}_t$
  $\mathcal{Z} \leftarrow \{w \in \mathcal{W} : \text{Var}(\text{rewards}(w)) = 0\}$
  Append responses from $\mathcal{W} \setminus \mathcal{Z}$ to $\mathcal{R}_t$ in arrival order, keeping groups intact
  $\hat{\rho}_{\text{zv}} \leftarrow |\mathcal{Z}|/|\mathcal{W}|$; $M_t \leftarrow B_t - |\mathcal{R}_t|/G$
  if $M_t > 0$ **then**
    if $\hat{\rho}_{\text{zv}} < 1$ **then**
      $B_{\text{cand}}^{\text{adapt}} \leftarrow \max\{b_{\min}, \min\{B_{\text{cand}}, \lceil \eta M_t/(1-\hat{\rho}_{\text{zv}}) \rceil\}\}$
    else
      $B_{\text{cand}}^{\text{adapt}} \leftarrow B_{\text{cand}}$
    end if
  end if
until $|\mathcal{R}_t| \geq B_t G$
$\mathcal{R}_{\text{train}} \leftarrow$ first B_t complete groups in $\mathcal{R}_t$; discard excess from training
Update π_θ with GRPO on the $B_t G$ responses in $\mathcal{R}_{\text{train}}$
Update $\mathcal{T}$ with rewards and response entropies from all groups in $\mathcal{A}_t$
$n_{\text{total}} \leftarrow |\mathcal{A}_t|$; count $n_{\text{easy}}, n_{\text{hard}}$ over all groups in $\mathcal{A}_t$
for $\tau \in \{\text{easy}, \text{hard}\}$ **do**
  if $n_\tau/n_{\text{total}} > \alpha$ **then**
    $p_{e,\tau} \leftarrow p_{e,\tau} - \Delta p$
  else if $n_\tau/n_{\text{total}} < \alpha$ **then**
    $p_{e,\tau} \leftarrow p_{e,\tau} + \Delta p$
  else
    Keep $p_{e,\tau}$ unchanged $\{n_\tau/n_{\text{total}} = \alpha\}$
  end if
  $p_{e,\tau} \leftarrow \min\{p_{\max}, \max\{p_{\min}, p_{e,\tau}\}\}$
end for

G Asset Licenses

Table 7 identifies the model checkpoints, data distributions, and principal software assets used in this study, distinguishing declared licenses from unresolved documentation.

Open-R1 subset and version. Our 30K subset is drawn from `open-r1/OpenR1-Math-220k`, configuration `default`, split `train`: the preprocessing script selects rows 0–29,999 in source order, without shuffling, and uses the `problem` and `answer` fields. The original Hugging Face revision was not recorded; a fresh download therefore does not establish an identical ordered subset.

Table 7: Asset sources and declared licenses. HF denotes Hugging Face and GH denotes GitHub; source names and version identifiers link to the corresponding assets.

Asset / version	**Distribution or source**	**License**
Models		
Qwen2.5-Math-1.5B/7B	HF: Qwen (1.5B, 7B)	Apache-2.0
Qwen2.5-1.5B/14B/32B	HF: Qwen (1.5B, 14B, 32B)	Apache-2.0
Qwen3-4B-Base	HF: Qwen/Qwen3-4B-Base	Apache-2.0
Llama-3.2-3B-Instruct	HF: meta-llama/Llama-3.2-3B-Instruct	Llama 3.2 Community; AUP
DeepSeek-R1-Distill-Qwen-1.5B	HF: deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B	MIT; base model Apache-2.0
Training and evaluation data		
DAPO-Math-17K-cleaned	HF: haizhongzheng/DAPO-Math-17K-cleaned; train	Apache-2.0[†]
MATH (7,500 training rows)	HF: xDAN2099/lighteval-MATH; train	MIT[†]
Open-R1 (30K; see above)	HF: open-r1/OpenR1-Math-220k; default/train	Apache-2.0
MATH-500	HF: HuggingFaceH4/MATH-500; test	MIT[†]
AIME 2024	HF: HuggingFaceH4/aime_2024; train	Apache-2.0[†]
AMC (AIMO validation)	HF: AI-MO/aimo-validation-amc; train	Apache-2.0
Gaokao 2023 (English)	GRESO raw-data/gaokao-2023en-test.jsonl, `b277843`	Not stated
Minerva Math	HF: math-ai/minervamath; test	Not stated
Olympiad Bench	HF: math-ai/olympiadbench; test	Apache-2.0[†]
BFCL-V2 (80/20 split)	HF: gorilla-llm/BFCL, `1726e899`; GH: Gorilla evaluator, `30124c4`	Apache-2.0 (data and evaluator)
4×4 Sudoku	HF: zzy1123/sudoku_dedup; train/test	Not stated
Software and reward design		
Reasoning Gym (reward adaptation)	GH: open-thought/reasoning-gym; revision not recorded	Apache-2.0
GRESO (training code)	GH: Infini-AI-Lab/GRESO	See note below
verl	GH: volcengine/verl; local fork	Apache-2.0
vLLM	GH: vllm-project/vllm	Apache-2.0

[†]License declared by the linked upstream release or repository; the exact HF mirror does not separately state a license. "Not stated" denotes missing distribution-level license documentation, not unrestricted reuse. GRESO retains Apache-2.0 notices in inherited verl files, but its package metadata points to a repository-wide license file that is absent from the inspected distribution. These gaps preclude a blanket claim that licenses and terms are documented for all assets. Sudoku data and the Reasoning Gym reward adaptation are separate assets.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and Section 1 state the moving-edge tracking claim and the edge-aware, online, and efficient selection objectives. These claims are supported by the methodology in Section 3, experiments in Section 4, and the limitations discussion.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Section 7 discusses the conservative nature of the conditional rank-consistency analysis, its potentially loose worst-case bounds, and the need for tighter variance-dependent guarantees.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: Section 3.3 states a conditional pairwise ordering result. Appendix A gives the explicit calibration assumption, finite-rollout concentration bound, and proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: Section 3 and Algorithm 1 specify HIVE. Section 4 and Appendix B disclose the main training protocol, selector hyperparameters, initialization and random seed, budget-matched baseline comparisons, and checkpoint-selection rule.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The public HIVE repository provides installation instructions, data-preparation scripts, and training/evaluation commands for the main math study. The release focuses on HIVE; baseline, 14B/32B scaling, tool-calling, Sudoku, and ablation pipelines are not bundled. The data guide notes that historical dataset revisions were not archived, and the installation guide explains the separate runtime requirements for Qwen3.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 4 and Appendix B specify models, datasets, context lengths, batch sizes, optimizer, selector settings, evaluation cadence, and checkpoint selection. Section 4.2 describes the tool-calling split and Sudoku reward; Appendix E.2 defines the Stage 2 score controls.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

   Justification: Due to training cost, the main RL results lack repeated-seed error bars or significance tests. Figure 7 reports within-bin standard deviations across prompts, not training-run uncertainty. Sudoku and the Stage 2 score controls each use one run per method; avg@4 averages sampled responses, not independent training runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Section 4 specifies 8×A100-80GB GPUs for the main study, and Table 1b gives its training-time breakdowns.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work studies RL post-training efficiency using public math-reasoning and tool-calling resources and a Sudoku task. It does not involve human-subject experiments, private data collection, or deployment decisions.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Section 7 discusses potential benefits for training cost, energy use, and research accessibility, and acknowledges general risks associated with stronger LLM capabilities. It does not identify application-specific harms beyond those general risks.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The release contains training code and data-preparation utilities, but no new model checkpoints or high-risk datasets requiring controlled release. Experiments use existing public resources and a Sudoku task.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No].

    Justification: Table 7 records model, data, and software sources, including BFCL-V2, Sudoku, Reasoning Gym, and all six math evaluation sets. Declared upstream licenses are distinguished from missing distribution-level documentation; unresolved data and GRESO licensing details prevent an affirmative answer for all assets. Appendix G specifies the Open-R1 30K selection rule and discloses its unrecorded historical revision.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The released HIVE code includes a README, installation and data guides, model configurations, runnable commands, attribution, and an Apache-2.0 license. Documentation covers runtime compatibility and dataset-snapshot limitations. The release provides the main math-training pipeline; it does not introduce a new dataset, benchmark, or model checkpoint.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: The paper does not involve human subjects, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes].

    Justification: The paper's core method is explicitly designed for RL post-training of LLMs/LRMs, and Sections 3 and 4.2 describe how the models are used and evaluated.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.