HIERARCHICAL-R1: HIERARCHICAL DATA SELECTION FOR RL POST-TRAINING OF LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) post-training has emerged as an effective paradigm for improving large language models (LLMs), particularly on reasoning and planning tasks. Recent approaches leverage token entropy to identify informative tokens for RL optimization. However, existing token-selection methods typically treat high-entropy tokens uniformly, overlooking their heterogeneous effects across reasoning trajectories and potentially limiting effective exploration. Motivated by a systematic analysis of high-entropy tokens, we introduce **Hierarchical-R1**, a hierarchical data selection framework that integrates sample-, trajectory-, and token-level signals for RL post-training. Specifically, **trajectory-guided token selection** differentiates high-entropy tokens from positive and negative trajectories to better preserve policy entropy and sustain exploration, while **token-guided sample selection** prioritizes samples likely to contain informative tokens, improving training efficiency. Experiments on models with different sizes across six mathematical reasoning benchmarks show that trajectory-guided token selection consistently improves accuracy over the evaluated baselines, while incorporating token-guided sample selection further reduces training cost with comparable average accuracy.

1 INTRODUCTION

Substantial progress has been achieved in large language models (LLMs) in recent years, demonstrating remarkable capabilities in natural language understanding, reasoning, and planning. Proprietary models, including OpenAI o1 (OpenAI, 2024), Claude (Anthropic, 2025), and Gemini (Team et al., 2023), as well as open-source models such as DeepSeek R1 (Guo et al., 2025), Qwen (Yang et al., 2025), and LLaMA (Grattafiori et al., 2024), have enabled a wide range of applications, including code agents (Zhang et al., 2024) and web agents (Li et al., 2025b). Recently, reinforcement learning (RL) post-training has become a key technique for further improving LLMs on downstream tasks, such as mathematical reasoning (Ahn et al., 2024). Notably, Group Relative Policy Optimization (GRPO) (Guo et al., 2025) avoids training a separate critic model by estimating advantages from group-normalized advantages, enabling memory-efficient and scalable RL post-training.

Recent advances in GRPO have sought to enhance performance through various algorithmic refinements, including improved clipping mechanisms (Yu et al., 2025; Xi et al., 2025; Wang et al., 2026a), reward shaping (Yu et al., 2025; Wan et al., 2025; Jiang et al., 2025; Chen et al., 2026), and joint optimization of cold-start supervision with RL training (Chen et al., 2025a; Li et al., 2026; Fu et al., 2025). However, existing methods largely optimize all tokens indiscriminately, without explicitly accounting for the heterogeneity of token-level learning signals. As a result, optimization may be diluted or even dominated by less informative tokens. To resolve this issue, recent studies (Wang et al., 2025; Meng et al., 2026) investigate token-specific contributions during RL training, highlighting the importance of high-entropy tokens. Specifically, Wang et al. (2025) observes that within chain-of-thought responses of LLMs, tokens exhibit varying entropy distributions: most tokens are low-entropy, while a few are high-entropy. Notably, high-entropy tokens often correspond to logical connectors (Wang et al., 2025), such as "wait" or "but", which are critical in shaping reasoning paths and enhancing performance.

Although prior works have selected tokens by entropy and identified high-entropy tokens as important ones, several key issues remain unaddressed. One limitation is that these approaches treat all

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

high-entropy tokens equally, incorporating them uniformly into the RL training. Our experiments show that the model generates different reasoning trajectories guided by the high-entropy tokens. However, we found that distinct patterns emerge across these trajectories. For correct trajectories, the high-entropy tokens help switch or continue the reasoning path, ultimately converging to the correct solution; in contrast, many incorrect trajectories exhibit a looping pattern, repeatedly switching reasoning paths even after reaching a correct answer, until exceeding the model's maximum output length. This experimental observation raises a critical question: *Do different high-entropy tokens have varying impacts on model behavior?*

On the other hand, current token selection strategies typically first perform full rollout to generate tokens and then prune less important ones. Although effective, they are computationally suboptimal because rollout is performed uniformly across samples, regardless of whether a sample is likely to produce informative tokens. As shown in Section 4.1, different samples exhibit varying potentials for generating important tokens during rollout, leading to inefficiency where some samples consume rollout resources while contributing few retained tokens. This raises another question: *Can we identify which samples are more likely to yield important tokens prior to rollout, thereby accelerating the training process?*

We analyze these two questions in Section 4.1, and our findings reveal that: at token-level, high-entropy tokens from correct trajectories maintain models' entropy, whereas those from incorrect trajectories lead to a rapid decline in entropy; at sample-level, informative samples demonstrate temporal consistency. Building on these findings, we propose **Hierarchical-R1**, a novel hierarchical data selection framework combining *trajectory-guided token selection* (TTS) for reasoning accuracy with *token-guided sample selection* for training efficiency. Hierarchical-R1 comprises two core components: *trajectory-guided token selection* and *token-guided sample selection*. Specifically, trajectory-guided token selection partitions rollouts into two subsets according to the sign of trajectory advantage, and modulates the proportion of high-entropy tokens from each subset to steer model behavior, thereby enhancing exploration and reasoning performance. Furthermore, to enhance efficiency, token-guided sample selection exploits token-level information extracted from previous rollouts to identify samples with a higher probability of yielding informative tokens, prioritizing their inclusion in subsequent training. Hierarchical-R1 achieves average accuracy comparable to TTS while reducing training cost. Our key contributions are summarized as follows:

- We conduct a comprehensive analysis of RL post-training from both token- and sample-level perspectives. The results reveal that (i): High-entropy tokens from correct trajectories preserve models' entropy, while those from incorrect trajectories lead to entropy collapse. (ii): Samples differ in their potential to generate informative tokens and informative samples demonstrate pronounced consistency across training epochs.
- Motivated by these insights, we propose Hierarchical-R1, a hierarchical data selection framework with: (i) **trajectory-guided token selection**, which modulates the proportion of high-entropy tokens across trajectories to enhance exploration and performance; and (ii) **token-guided sample selection**, which exploits token-level signals to prioritize informative samples, improving training efficiency while maintaining comparable average accuracy.
- Experiments on models with different sizes across six mathematical reasoning benchmarks show that TTS improves accuracy over the evaluated baselines, while Hierarchical-R1 achieves comparable average accuracy to TTS at a lower training cost.

2 Related Work

LLM RL Post-training. Since the success of Reinforcement Learning with Verifiable Rewards (RLVR), RL-based post-training has become a standard approach for enhancing downstream performance. Various RL algorithms have been proposed, including GRPO (Guo et al., 2025), which replaces critic models with group-normalized advantages; DAPO (Yu et al., 2025), which incorporates dynamic sampling, token-level gradient optimization, asymmetric clipping, and KL-loss removal; and GSPO (Zheng et al., 2025a), which introduces a sequence-level objective to stabilize Mixture-of-Experts RL training. Other studies also focus on enhanced clipping mechanisms (Yu et al., 2025; Wang et al., 2026a), such as BAPO (Xi et al., 2025), which adopts adaptive clipping to stabilize off-policy RL. Despite their effectiveness, these methods treat all tokens uniformly, overlooking their heterogeneous contributions during training. Recent studies have begun to explore the varying contributions of different tokens (Vassoyan et al., 2025; Lin et al., 2024; Ma et al., 2026).

Works such as (Meng et al., 2026; Chen et al., 2025b) show that most tokens exhibit limited distributional shifts after RL training, while Wang et al. (2025) reveal that a minority of high-entropy tokens are critical for shaping the reasoning path. However, these studies treat all high-entropy tokens uniformly, ignoring the distinct impacts of specific high-entropy tokens on model behavior. To fill this gap, we conduct a detailed analysis on different high-entropy tokens.

Data Selection for LLM RL Post-training. Data quality is pivotal to effective downstream adaptation, motivating extensive research on data selection for LLM RL post-training. Existing approaches can be broadly categorized into offline methods (Wang et al., 2026b; Wen et al., 2025; Li et al., 2025a), performed before training, and online methods (Sun et al., 2025; Zheng et al., 2025b; Mao et al., 2026; Yi et al., 2026; Tang et al., 2025), executed dynamically during training. Our focus is on the latter. Most online methods aim to mitigate the zero-advantage problem by filtering zero-variance samples. For instance, Sun et al. (2025) identify moderately difficult samples via an attention-based reference framework, while GRESO (Zheng et al., 2025b) removes historically zero-variance samples using a lightweight pre-rollout filter. Similarly, (Mao et al., 2026) predicts moderately difficult samples by hidden Markov modeling, while DEPO (Tang et al., 2025) jointly considers diversity, influence, and difficulty in sample selection. Another work (Yi et al., 2026) proposes an uncertainty consistency metric to select samples. However, these methods largely neglect inefficiencies in token selection. To tackle this problem, we introduce a token-guided sample selection method.

3 PRELIMINARY

Group Relative Policy Optimization (GRPO). GRPO has been widely utilized for LLM RL post-training. Specifically, given a query q_j in a B-batch-sized batch $\mathcal{B} = \{q_j\}_{j=1}^{B}$ and its corresponding answer a_j, G trajectories are sampled from the old policy model $\pi_{\theta_{\text{old}}}$, denoted as $\{o_{j,i}\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|q_j)$. Trajectories are subsequently evaluated, yielding outcome rewards $\{r_{j,i}\}_{i=1}^{G}$ and group-wise normalized advantages as $A_{j,i}^{t} = \frac{r_{j,i} - \text{mean}(\{r_{j,i}\}_{i=1}^{G})}{\text{std}(\{r_{j,i}\}_{i=1}^{G})}$. The loss is:

$$\mathbb{E}_{\mathcal{B}\sim\mathcal{D}, o_{j,i}\sim\pi_{\theta_{\text{old}}}} \frac{1}{B}\sum_{j=1}^{B}\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_{j,i}|}\sum_{t=1}^{|o_{j,i}|}\min\left[\gamma_{j,i}^{t}(\theta)A_{j,i}^{t}, \text{clip}(\gamma_{j,i}^{t}(\theta), 1-\epsilon, 1+\epsilon)A_{j,i}^{t}\right], \quad (1)$$

where $\gamma_{j,i}^{t}(\theta) = \frac{\pi_{\theta}(o_{j,i}^{t}|q_j, o_{j,i}^{<t})}{\pi_{\theta_{\text{old}}}(o_{j,i}^{t}|q_j, o_{j,i}^{<t})}$ denotes the importance ratio. In this objective, ϵ, π_θ and $|o_{j,i}|$ denote the clipping bound, the policy model and the length of the response $|o_{j,i}|$ respectively.

Dynamic Sampling Policy Optimization (DAPO). DAPO extends GRPO through four key enhancements: dynamic sampling, asymmetric clipping, token-level policy optimization, and the elimination of KL regularization. The loss is:

$$\begin{aligned}\mathbb{E}_{\mathcal{B}\sim\mathcal{D}, o_{j,i}\sim\pi_{\theta_{\text{old}}}} &\frac{1}{B}\sum_{j=1}^{B}\frac{1}{\sum_{i=1}^{G}|o_{j,i}|}\sum_{i=1}^{G}\sum_{t=1}^{|o_{j,i}|}\min\left[\gamma_{j,i}^{t}(\theta)A_{j,i}^{t}, \text{clip}(\gamma_{j,i}^{t}(\theta), 1-\epsilon_{\text{l}}, 1+\epsilon_{\text{h}})A_{j,i}^{t}\right] \\ &\text{s.t.} \quad 0 < \left|\{o_{j,i} \mid \texttt{is_equivalent}(a_j, o_{j,i})\}\right| < G,\end{aligned} \quad (2)$$

where ϵ_{l} and ϵ_{h} are clipping lower and upper bound.

4 METHODOLOGY

In this section, we investigate RL post-training from token- and sample-level perspectives and propose a hierarchical data selection framework for RL optimization. In Section 4.1, we find that high-entropy tokens from incorrect trajectories tend to reduce model entropy rapidly, whereas those from correct trajectories exhibit the opposite effect. Furthermore, a sample-level analysis reveals samples' heterogeneity and temporal consistency. Motivated by these insights, we introduce Hierarchical-R1, a hierarchical data selection method, comprising trajectory-guided token selection and token-guided sample selection in Sections 4.2 and 4.3. The detailed algorithm is provided in Appendix A.

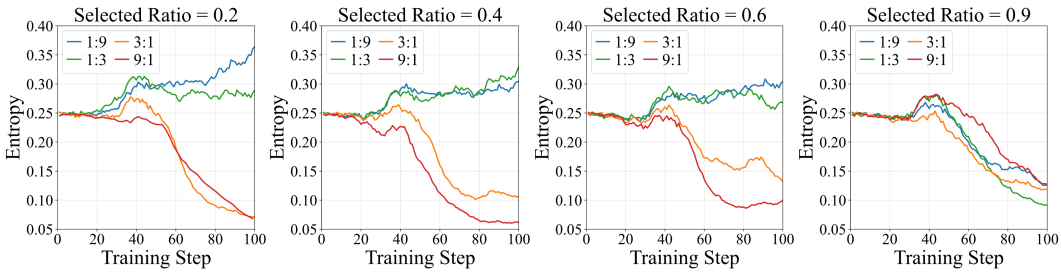


Figure 1: Entropy dynamics of Qwen3-8B during RL training at selected ratios $r \in \{0.2, 0.4, 0.6, 0.9\}$. The ratios in the legends indicate the proportions of selected tokens from incorrect versus correct trajectories.

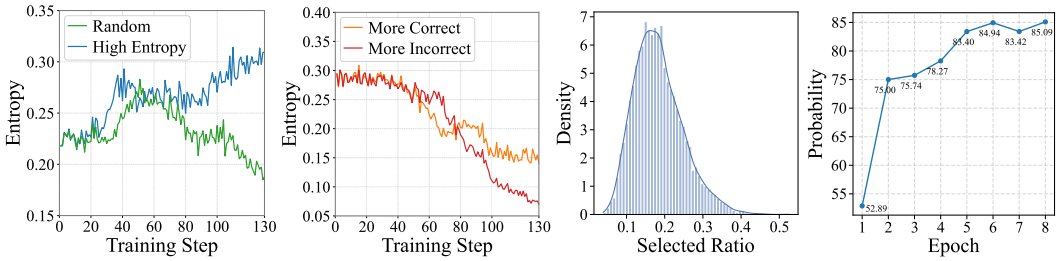


Figure 2: (a) Entropy dynamics under high-entropy and random selection with matched selected ratios and proportions of tokens from incorrect and correct trajectories. (b) Entropy dynamics with all response tokens updated, using prompts with a majority of correct or incorrect trajectories. (c) Distribution of selected ratios across samples. (d) Probability that a sample in the top 10% by selected ratio remains in this group in the next epoch.

4.1 Motivation

Analysis on the Role of High-Entropy Tokens in RL Training. High-entropy tokens have been shown to reshape reasoning paths and guide models toward diverse trajectories, improving RL training performance (Wang et al., 2025). Yet, whether all high-entropy tokens contribute equally remains unclear. We investigate this by analyzing model entropy dynamics under varying high-entropy token configurations. Specifically, we use a selected ratio $r \in \{0.2, 0.4, 0.6, 0.9\}$ to control the fraction of response tokens involved in training. We further vary the composition by sampling high-entropy tokens from incorrect and correct trajectories at ratios of 9:1, 3:1, 1:3, and 1:9. The resulting entropy dynamics are shown in Fig. 1, where only selected tokens participate in training. At low r, higher proportions of high-entropy tokens from correct trajectories better preserve model entropy, whereas those from incorrect trajectories induce rapid entropy collapse. As r increases, this difference gradually diminishes, particularly at $r = 0.9$, where entropy dynamics are nearly identical across proportions. This is attributable to the increasing dominance of low-entropy tokens, which attenuates the influence of high-entropy tokens.

Second, **to disentangle token entropy from trajectory composition**, we compare high-entropy and random token selection while matching both the selected-token ratio and the proportions of tokens from correct and incorrect trajectories. Random selection yields substantially faster entropy decay (Fig. 2 (a)), showing that trajectory composition alone cannot explain the observed behavior. Third, **to test whether trajectory correctness itself is sufficient**, we update all response tokens while separately using prompts dominated by correct or incorrect trajectories. Both settings exhibit rapid entropy decay (Fig. 2 (b)). Together, these results indicate that the observed entropy dynamics arise from the interaction between token entropy and trajectory outcome: high-entropy selection is important, and its effect further depends on the trajectory from which the token originates.

Insight (i): High-entropy tokens from different trajectories contribute distinctly to the model's entropy dynamics. Controlling their relative participation provides an effective mechanism for regulating policy entropy during RL training.

Inefficiency of Existing Token Selection. Existing token selection methods first perform rollouts and subsequently prune unimportant tokens, incurring substantial inefficiency as considerable com-

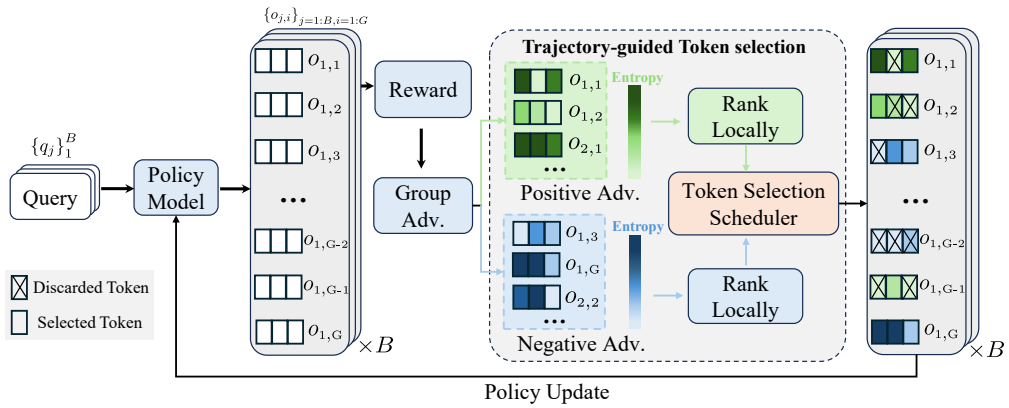


Figure 3: Pipeline of trajectory-guided token selection. In this framework, trajectories are first partitioned into positive and negative advantage subsets, within which tokens are locally ranked by entropy. High-entropy tokens are then scheduled from both subsets for training

putation is wasted on samples most of whose tokens are discarded from training. To validate this, we use entropy as a metric, identify high-entropy tokens as important tokens and prune unimportant tokens during training. The distribution of the proportion of selected tokens for each sample is demonstrated in Fig. 2 (c), revealing substantial cross-sample heterogeneity, with some contributing only marginally informative content. Furthermore, we investigate whether samples yielding a high proportion of informative tokens exhibit temporal consistency throughout training. Specifically, we compute the top 10% persistence probability, defined as the fraction of samples ranked in the top 10% by selected ratio at the previous epoch that remain in the top 10% at the subsequent epoch. As reported in Fig. 2 (d), informative samples exhibit strong temporal consistency.

Insight (ii): Given pronounced sample heterogeneity and temporal consistency, historical information can be exploited to prioritize samples likely to generate informative tokens before the subsequent epoch's rollout, thereby enriching learning signals and accelerating RL convergence.

4.2 Trajectory-guided Token Selection

Based on *Insight (i)*, we propose a trajectory-guided token selection method (TTS), illustrated in Fig. 3, which differentiates high-entropy tokens using trajectory-level signals rather than treating them equally.. This method consists of three procedures in each policy update step: trajectory partitioning, dual subset token selection scheduling and token-level selective RL training. By controlling which high-entropy tokens participate in training, the method regulates the model's learning behavior during RL optimization and thereby improves overall performance. Details of these three procedures are elaborated as follows.

Trajectory Partitioning. At each policy model update step, given a sampled batch $\mathcal{B}$, we rollout G responses for each query in the batch with the policy model. The corresponding rewards and group-relative normalized advantages are then assigned. Based on the sign of each trajectory's advantage, we partition all trajectories in the batch into two subsets, denoted as $\mathcal{T}^+ = \{o_{j,i}|A_{j,i} > 0, j = 1:B, i = 1:G\}$ and $\mathcal{T}^- = \{o_{j,i}|A_{j,i} < 0, j = 1:B, i = 1:G\}$, corresponding to positive and negative advantages, respectively. Our method is based on DAPO, where zero advantage samples are filtered out from the training batch. Therefore, advantages of trajectories in $\mathcal{T}$ are either positive or negative.

Dual Subset Token Selection Scheduling. After obtaining the two trajectory subsets, we schedule token selection separately from each subset. To avoid low-entropy tokens diluting the distinct characteristics of high-entropy tokens across different trajectories, we first compute the entropy of each token within the positive and negative trajectory subsets ($\mathcal{T}^+$ and $\mathcal{T}^-$) respectively, followed by a local ranking based on token entropy. Specifically, for all response tokens in the positive trajectory subset $\mathcal{S}^+ = \{o^t_{j,i}|A_{j,i} > 0, j = 1:B, i = 1:G, t = 1:|o_{j,i}|\}$, we compute each token's entropy, denoted as $H^t_{j,i} = -\sum_{n=1}^{V} p_n \log p_n$, where V is the vocabulary size and p_n is the probability distribution of the trajectory $o_{j,i}$ at the position of t. And then the tokens are sorted in descending

order of entropy to obtain the sorted positive trajectory tokens, denoted as $\mathcal{S}^{+}_{\text{ranked}}$. The same procedure is applied to tokens in the negative trajectory subset, resulting in $\mathcal{S}^{-}_{\text{ranked}}$. To further control the influence of high-entropy tokens on model behavior during training, we introduce a scheduling parameter λ_k at each policy model update step. Let the total number of response tokens from all samples in the batch be S, and let r be a hyperparameter controlling the total proportion of tokens participating in training. At the k-th training step, we select the top $S \cdot r \cdot (1 - \lambda_k)$ high-entropy tokens from $\mathcal{S}^{+}_{\text{ranked}}$, yielding a binary mask:

$$m^t_{j,i} = \mathbb{I}\big(o^t_{j,i} \in \text{Top}_{Sr(1-\lambda_k)}(\mathcal{S}^{+}_{\text{ranked}})\big), \tag{3}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. Similarly, the top $S \cdot r \cdot \lambda_k$ tokens from $\mathcal{S}^{-}_{\text{ranked}}$ are selected, yielding corresponding binary masks as:

$$m^t_{j,i} = \mathbb{I}\big(o^t_{j,i} \in \text{Top}_{Sr\lambda_k}(\mathcal{S}^{-}_{\text{ranked}})\big). \tag{4}$$

Token-level Selective RL Training. Based on the binary mask for each token, the RL post-training objective is expressed as:

$$\mathop{\mathbb{E}}_{\substack{\mathcal{B}\sim\mathcal{D},\\ o_{j,i}\sim\pi_{\theta_{\text{old}}}}} \frac{1}{B}\sum_{j=1}^{B} \frac{1}{\sum_{i=1}^{G}\sum_{t=1}^{|o_{j,i}|} m^t_{j,i}} \sum_{i=1}^{G}\sum_{t=1}^{|o_{j,i}|} m^t_{j,i} \min\left[\gamma^t_{j,i}(\theta)A^t_{j,i}, \text{clip}(\gamma^t_{j,i}(\theta), 1-\epsilon_{\text{l}}, 1+\epsilon_{\text{h}})A^t_{j,i}\right]$$
$$\text{s.t.} \quad 0 < \Big|\{o_{j,i} \mid \texttt{is_equivalent}(a, o_{j,i})\}\Big| < G. \tag{5}$$

This formulation prevents the participation of low-entropy tokens from diluting the distinct effects of high-entropy tokens, while allowing a greater proportion of high-entropy tokens from positive trajectories to contribute to training. Consequently, it encourages the model to maintain higher entropy during training, thereby enhancing exploration and ultimately improving performance.

4.3 Token-guided Sample Selection

Motivated by *Insight (ii)*, we extend the trajectory-guided token selection framework with a token-guided sample selection mechanism, as demonstrated in Fig. 4. This leverages token-level information extracted from historical rollouts to guide sample selection, thereby increasing the training participation of informative samples.

Our approach begins with a warm-up training phase, where all samples are uniformly sampled with replacement in the first epoch. Then, we apply the dynamic sampling of DAPO to select non-zero advantage samples participating in the policy update. For these samples, rollouts are performed to generate responses, followed by trajectory-guided token selection and token-level selective RL training. Each token is assigned a binary mask $m^t_{j,i}$. Based on this, each sample's token-level information at the first epoch is then collected as:

$$h^1_j = \frac{\sum_{i=1}^{G}\sum_{t=1}^{|o_{j,i}|} m^t_{j,i}}{\sum_{i=1}^{G}|o_{j,i}|}. \tag{6}$$

Based on the token-level information obtained in the previous training epoch, we assign a weight to each sample by normalizing the historical token-level information, denoted as:

$$\tilde{w}^1_j = \frac{h^1_j}{\max\{h^1_j\}_{j=1}^{N}}, \tag{7}$$

where N and w_j denote the size of training dataset and the weight assigned to the j-th sample, respectively. Samples that were previously excluded by the DAPO-like filtering strategy are assigned a weight of zero. Directly using these weights for sampling would prevent initially difficult samples from being selected in later training stages, even as the model's capabilities improve, which can hinder RL performance. To mitigate this, we incorporate a uniform sampling component into the weight computation, leading to the final sample weight as:

$$w^1_j = (1-\alpha)\cdot\frac{h^1_j}{\max\{h^1_j\}_{j=1}^{N}} + \alpha, \tag{8}$$

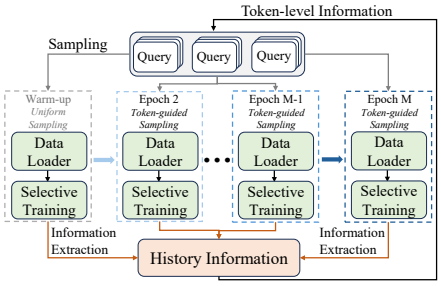


Figure 4: Pipeline overview.

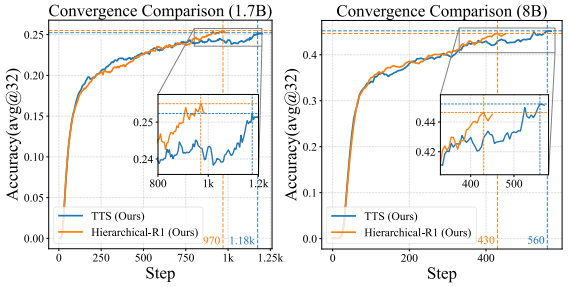


Figure 5: Training convergence comparison.

Table 1: Comparison with baselines across mathematical reasoning benchmarks on Qwen3-1.7B and Qwen3-8B. Olympiad Bench and MATH500 are abbreviated as Olym. and MATH, respectively.

Model	Qwen3-1.7B							Qwen3-8B						
Method	AIME24	AIME25	AMC	Olym.	Gaokao	MATH.	AVG	AIME24	AIME25	AMC	Olym.	Gaokao	MATH.	AVG
DAPO	23.02	20.10	52.17	44.96	71.59	77.56	48.23	42.29	31.87	68.42	55.80	79.66	84.08	60.35
FIPO	20.73	19.58	53.27	46.14	71.61	77.96	48.22	41.56	29.06	73.03	58.51	81.76	85.30	61.54
GRESO	23.75	22.18	52.63	45.12	71.70	78.46	48.97	41.66	31.25	69.65	56.83	80.45	84.58	60.73
Beyond	21.67	22.08	52.22	45.14	71.94	78.09	48.52	43.12	30.83	71.73	58.42	80.58	85.05	61.62
TTS (Ours)	24.90	23.44	**58.29**	**50.92**	**75.41**	79.60	52.09	**46.67**	37.92	**77.96**	**61.96**	**83.19**	**86.17**	**65.65**
Hierarchical-R1 (Ours)	**27.40**	**25.21**	56.43	50.51	75.23	**79.83**	**52.44**	46.56	**38.96**	77.23	60.90	82.47	86.04	65.36

where α is a parameter controlling the balance between token-guided sampling and uniform sampling. Samples are then drawn randomly according to these weights, followed by trajectory-guided token selection and token-level selective RL training. Token-level information extracted in the current round is used to update the historical record as:

$$h_j^2 = (1-\beta)\cdot h_j^1 + \beta \cdot \frac{\sum_{i=1}^{G}\sum_{t=1}^{|o_{j,i}|} m_{j,i}^t}{\sum_{i=1}^{G}|o_{j,i}|}, \tag{9}$$

where β balances the contribution of prior and newly observed token-level information. This updated record is used in subsequent training rounds.

By iterating this procedure for M epochs, historical token-level information is continuously updated, and informative samples are prioritized in each round. This approach enables samples with a higher probability of generating more important tokens to participate more in training, providing richer learning signals and thereby reducing training cost while maintaining accuracy comparable to TTS.

5 Experiment

In this section, we comprehensively evaluate Hierarchical-R1 across diverse benchmarks and model scales. We further present detailed ablation studies to evaluate the robustness and effectiveness of our method. Additional case studies are provided in Appendix D.

5.1 Experimental Setup

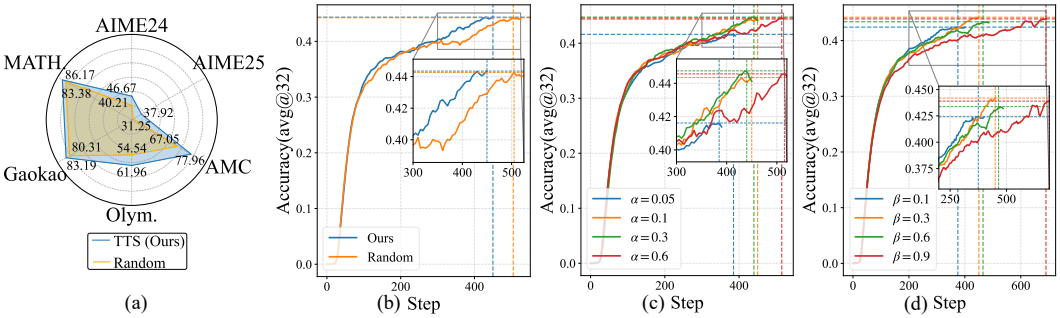
Models. We adopt Qwen3-1.7B and Qwen3-8B (Yang et al., 2025) as base models to evaluate different RL strategies, following the experimental setup of prior work (Wang et al., 2025).

Datasets. Consistent with prior work (Zheng et al., 2025b), we combine the DAPO dataset (Yu et al., 2025) with the MATH dataset (Hendrycks et al., 2021), which provides LaTeX-formatted solutions, to construct the training corpus. This design preserves formatting diversity and improves generalization. Details on dataset composition and splits are provided in Appendix B.4. For evaluation, we consider six mathematical reasoning benchmarks: AIME24 (Art of Problem Solving, 2024a), AIME25 (Art of Problem Solving, 2024a), AMC (Art of Problem Solving, 2024b), Olympiad Bench (He et al., 2024), Gaokao (Zhang et al., 2023), and MATH500 (Hendrycks et al., 2021). We report Avg@32 metric on each benchmark. All results use one seed, while Appendix C shows consistent performance across three seeds, demonstrating robustness to randomness.

Implementation Details. We implement our Hierarchical-R1 based on the DAPO algorithm. For trajectory-guided token selection, the token selected ratio r in each batch is set as 0.3, while the token scheduling coefficient λ_k is fixed at 0.4 in our main experiments. For token-guided sample selection, the hyperparameters α and β are set as 0.1/0.3 for the 8B model and 0.2/0.15 for 1.7B

Table 2: Training efficiency comparison between TTS and Hierarchical-R1.

Method	Qwen3-1.7B		Qwen3-8B	
	Convergence Step ⇓	Training Time (GPU hours) ⇓	Convergence Step ⇓	Training Time (GPU hours) ⇓
TTS (Ours)	1180	417.10	560	334.46
Hierarchical-R1 (Ours)	970	342.80 (17.8% ↓)	430	263.27 (21% ↓)



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

Figure 6: (a) Comparison of our method with the random variant. (b) Comparison between our token-guided sample selection and a random variant. (c) Ablation on α of token-guided sample selection. (d) Ablation on β of token-guided sample selection.

model, respectively. The maximum context length during both training and evaluation is limited to 4096. Additional implementation details are provided in Appendix B.

Baselines. We compare our method with several baseline methods: (1) DAPO (Yu et al., 2025), which trains models with all tokens participating. (2) FIPO (Ma et al., 2026), which introduces a discounted future-KL divergence to reweight each token. (3) GRESO (Zheng et al., 2025b), which develops a pre-rollout sample selection method to filter out zero-advantage samples. (4) Beyond the 80/20 Rule (Beyond) (Wang et al., 2025), which utilizes entropy as the metric to identify forking tokens and prevent other tokens from training.

5.2 Main Results

We investigate the effectiveness of trajectory-guided token selection by comparing our TTS with several baselines on mathematical reasoning benchmarks using 1.7B and 8B models. As shown in Tab. 1, our TTS yields average accuracy gains of 3.12% and 4.03% respectively. Unlike token selection strategies based solely on future KL divergence or entropy, our method leverages trajectory-level information to selectively control the participation of tokens during training, enabling more precise regulation of model behavior in RL training, leading to substantial performance gains.

We evaluate the contribution of sample selection by comparing Hierarchical-R1 with TTS (Fig. 5 and Tab. 2). Incorporating sample selection reduces training cost by 17.8% and 21% on Qwen3-1.7B and Qwen3-8B, respectively, while maintaining comparable average accuracy, with mean Avg@32 changing from 52.09 to 52.44 and from 65.65 to 65.36 (Tab. 1). These results demonstrate that sample selection primarily improves training efficiency without compromising overall performance. Appendix E further compares convergence behavior with Beyond.

5.3 Ablation Studies

In this section, we conduct detailed ablation analysis on the 8B model to validate the effectiveness and robustness of Hierarchical-R1.

Comparison with Random Variants. To validate the effectiveness of Hierarchical-R1 and isolate its contributions, we evaluate TTS against two random token selection baselines with selection ratio fixed at $r = 0.3$. The first samples tokens uniformly across all trajectories (Fig. 6 (a)). The second additionally fixes $\lambda_k = 0.4$ and performs uniform sampling within the correct and incorrect trajectory subsets, preserving the token selection quotas of TTS (Tab. 3). TTS outperforms both baselines across all benchmarks, underscoring the benefit of exploiting trajectory-level information. We further compare token-guided sample selection with a random sample selection variant, in which samples are assigned random weights for weighted sampling. As shown in Fig. 6 (b), token-guided sample selection reaches comparable accuracy in fewer training steps than random selection.

Table 3: Performance across r and λ_k. Baseline variant ("Random" in Table) samples tokens uniformly within the correct and incorrect trajectory subsets, using the same r and λ_k as TTS.

Method / Setting	AIME24	AIME25	AMC	Olym.	Gaokao	MATH.	AVG
$r = 0.1$							
Beyond	41.25	30.52	69.53	56.83	80.05	84.46	60.44
TTS (Ours)	44.27	37.08	72.74	58.62	82.39	85.73	63.47
$r = 0.2$							
Beyond	42.80	30.10	73.00	60.43	82.71	85.10	62.35
TTS (Ours)	46.25	35.40	74.80	60.73	83.05	85.70	64.32
$r = 0.3,\ \lambda_k = 0.4$							
Random	40.21	32.50	68.52	54.97	78.86	84.54	59.93
TTS (Ours)	46.67	37.92	77.96	61.96	83.19	86.17	65.65
TTS: varying λ_k with $r = 0.2$							
$\lambda_k = 0.2$	45.63	35.21	74.55	60.41	82.68	85.86	64.06
$\lambda_k = 0.4$	46.25	35.40	74.80	60.73	83.05	85.70	64.32
$\lambda_k = 0.6$	40.94	31.46	74.48	59.85	82.29	84.85	62.31
$\lambda_k = 0.9$	36.14	29.06	69.25	57.66	80.87	84.14	59.52

Analysis under Different Token Selection Settings. To evaluate the robustness of our TTS, we compare its performance with baselines under varying r. Results in Tab. 3 demonstrate that our method consistently outperforms baselines across different r, demonstrating strong robustness. We further investigate the impact of λ_k on performance by fixing $r = 0.2$ and varying $\lambda_k \in \{0.2, 0.4, 0.6, 0.9\}$. We observe that entropy begins to decay rapidly when $\lambda_k = 0.6$. Larger λ_k values introduce more high-entropy tokens from negative trajectories, accelerating entropy collapse and restricting exploration, which ultimately degrades performance. In contrast, smaller λ_k values favor high-entropy tokens from positive trajectories, better preserving model entropy during training and facilitating exploration. Notably, $\lambda_k = 0.2$ and 0.4 yield consistently strong performance, indicating low sensitivity to this hyperparameter.

Analysis under Varying Sample Selection Settings. To investigate the impact of hyperparameters in token-guided sample selection, we train models under different settings and examine their convergence behavior. Results are demonstrated in Fig. 6 (c and d). First, we vary the parameter $\alpha \in \{0.05, 0.1, 0.3, 0.6\}$. When α is too large, the contribution of token-guided sampling is diluted by the uniform sampling component, leading to slower convergence. Conversely, overly small α reduces dataset diversity. Samples that are initially difficult are filtered out by dynamic sampling strategy and assigned near-zero weights, which persist across subsequent epochs. As model capacity increases, these samples would become moderately challenging and offer informative learning signals; however, their exclusion limits the model's exposure to such signals, ultimately degrading peak performance. These observations validate that α balances uniform and token-guided sampling. Notably, $\alpha = 0.1$ and 0.3 yield comparable performance, highlighting robustness to its choice.

We then evaluate the historical update parameter $\beta \in \{0.1, 0.3, 0.6, 0.9\}$. Smaller β overemphasizes historical weights, causing information lag and preventing early low-weighted samples from participating in training, thereby reducing data diversity and peak performance. In contrast, excessively large β relies almost exclusively on the most recent token-level information, amplifying noise and slowing convergence relative to $\beta = 0.3$.

6 Conclusion

In this work, we propose Hierarchical-R1, a hierarchical data selection framework for RL post-training that systematically leverages sample-, trajectory- and token-level information. By distinguishing contributions of different high-entropy tokens, trajectory-guided token selection preserves model entropy and promotes exploration, while token-guided sample selection accelerates convergence by prioritizing informative samples. Extensive experiments show that TTS consistently improves accuracy over the evaluated baselines, and incorporating token-guided sample selection further reduces training cost while maintaining comparable average accuracy. These results demonstrate the effectiveness of structured multi-level data selection for RL post-training.

AI USE STATEMENT

In this work, we used generative AI tools to improve the clarity, readability, and language of the manuscript. We did not use generative AI tools to develop research methodology, design experiments, implement methods, analyze or interpret results, or formulate scientific claims, while the remaining required disclosure tasks are not applicable to this work. All AI-assisted revisions were carefully reviewed by the authors to ensure technical accuracy and consistency with the intended meaning. We take full responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We support reproducibility by providing detailed descriptions of our method, experimental setup, model configurations, training hyperparameters, datasets, and evaluation protocols in the main paper and Appendix. Additional implementation and experimental details are included in the supplementary materials to facilitate independent verification and reliable reproduction of our results. Upon publication, we will release source code to reproduce training and evaluation results.

REFERENCES

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 225–237, 2024.

Anthropic. Claude 3.7 sonnet. https://www.anthropic.com/claude/sonnet, 2025.

Art of Problem Solving. Aime problems and solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`, 2024a. Accessed: 2025-04-20.

Art of Problem Solving. Amc problems and solutions. `https://artofproblemsolving.com/wiki/index.php?title=AMC_Problems_and_Solutions`, 2024b. Accessed: 2025-04-20.

Guanzheng Chen, Michael Qizhe Shieh, and Lidong Bing. Longrlvr: Long-context reinforcement learning requires verifiable context rewards. *arXiv preprint arXiv:2603.02146*, 2026.

Shuang Chen, Yue Guo, Zhaochen Su, Yafu Li, Yulun Wu, Jiacheng Chen, Jiayu Chen, Weijie Wang, Xiaoye Qu, and Yu Cheng. Advancing multimodal reasoning: From optimized cold start to staged reinforcement learning. *arXiv preprint arXiv:2506.04207*, 2025a.

Xingwu Chen, Tianle Li, and Difan Zou. Reshaping reasoning in llms: A theoretical analysis of rl training dynamics through pattern selection. *arXiv preprint arXiv:2506.04695*, 2025b.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. *arXiv preprint arXiv:2506.19767*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Lingjie Jiang, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*, 2025.

Xinran Li, Guangda Huzhang, Siqi Shen, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Jun Zhang. Getting your llms ready for reinforcement learning with lightweight sft. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, et al. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. *arXiv preprint arXiv:2509.13312*, 2025b.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. Critical tokens matter: Token-level contrastive estimation enhances llm's reasoning capability. *arXiv preprint arXiv:2411.19943*, 2024.

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. Fipo: Eliciting deep reasoning with future-kl influenced policy optimization. *arXiv preprint arXiv:2603.19835*, 2026.

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.

Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Xue Wang, Guoyin Wang, Bolin Ding, and Jingren Zhou. Sparse but critical: A token-level analysis of distributional shifts in rlvr fine-tuning of llms. *arXiv preprint arXiv:2603.22446*, 2026.

OpenAI. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/, 2024.

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

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. *arXiv preprint arXiv:2506.05316*, 2025.

Xinyu Tang, Zhenduo Zhang, Yurou Liu, Wayne Xin Zhao, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Towards high data efficiency in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2509.01321*, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Jean Vassoyan, Nathanaël Beau, and Roman Plaud. Ignore the kl penalty! boosting exploration on critical tokens to enhance rl fine-tuning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6108–6118, 2025.

Zhongwei Wan, Zhihao Dou, Che Liu, Yu Zhang, Dongfei Cui, Qinjian Zhao, Hui Shen, Jing Xiong, Yi Xin, Yifan Jiang, et al. Srpo: Enhancing multimodal llm reasoning via reflection-aware reinforcement learning. *arXiv preprint arXiv:2506.01713*, 2025.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Shumin Wang, Yuexiang Xie, Wenhao Zhang, Yuchang Sun, Yanxi Chen, Yaliang Li, and Yanyong Zhang. On the entropy dynamics in reinforcement fine-tuning of large language models. *arXiv preprint arXiv:2602.03392*, 2026a.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *Advances in Neural Information Processing Systems*, 38: 122721–122764, 2026b.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Tanglifu Tanglifu, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 318–327, 2025.

Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, et al. Bapo: Stabilizing off-policy reinforcement learning for llms via balanced policy optimization with adaptive clipping. *arXiv preprint arXiv:2510.18927*, 2025.

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

Hao Yi, Yulan Hu, Xin Li, Sheng Ouyang, Lizhong Ding, and Yong Liu. Learn more with less: Uncertainty consistency guided query selection for rlvr. *arXiv preprint arXiv:2601.22595*, 2026.

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

Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13643–13658, 2024.

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.

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*, 2025a.

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. *arXiv preprint arXiv:2506.02177*, 2025b.

Appendix

A Algorithm

In this section, we provide the details of our Hierarchical-R1 (see Algorithm 1).

Algorithm 1 Hierarchical-R1

1: **Input:** Training dataset $\mathcal{D}$, token scheduling parameter λ_k, Total training epoch M, The size of the training dataset N, Policy model θ
2: **Warm-up Stage:** All samples are sampled uniformly. For these samples, dynamic sampling, trajectory-guided token selection and token-level selective RL training are conducted orderly in each training batch. During this process, each token is assigned with a binary mask $m_{j,i}^t$ and the token-level information for each sample is stored: $h_j^1 = \frac{\sum_{i=1}^{G}\sum_{t=1}^{|o_{j,i}|} m_{j,i}^t}{\sum_{i=1}^{G}|o_{j,i}|}$;
3: **for** m **in** $\{1, 2, \cdots, M\}$ **do**
4: Utilize the stored historical token-level information to yield a weight for each sample: $w_j^m = (1-\alpha)\cdot\frac{h_j^m}{\max\{h_j^m\}_{j=1}^{N}} + \alpha$;
5: Conduct the token-guided sample selection based on the weights $\{w_j^m\}_{j=1}^{N}$, performing weighted sampling with replacement;
6: For these sampled data, conduct dynamic sampling, trajectory-guided token selection and token-level selective RL training orderly in each training batch, where binary masks $m_{j,i}^t$ are obtained;
7: The stored historical token-level information is updated: $h_j^{m+1} = (1-\beta)\cdot h_j^m + \beta \cdot \frac{\sum_{i=1}^{G}\sum_{t=1}^{|o_{j,i}|} m_{j,i}^t}{\sum_{i=1}^{G}|o_{j,i}|}$;
8: **end for**
9: **Output:** The trained model θ_M.

B Additional Implementation details

This section describes the training and evaluation settings, dataset composition and overlap, and computing infrastructure.

B.1 Training Details

For all experimental results in this paper, we adopt verl framework (Sheng et al., 2025) to perform RL training. Our method is implemented based on DAPO algorithm (Yu et al., 2025), removing overlong reward shaping. The training hyperparameters are reported in Tab. 4.

B.2 Evaluation Details

We evaluate our method and baseline methods on different benchmarks. The details of benchmarks we used are illustrated in Tab. 5. The Avg@32 metric is adopted as evaluation metric in our paper. Additionally, the evaluation hyperparameter settings are demonstrated in Tab. 6.

B.3 Infrastructure

In our paper, all experiments are conducted on 8 NVIDIA H100 GPUs.

B.4 Data Contamination

Following prior work (Zheng et al., 2025b), our training corpus combines the DAPO dataset (Yu et al., 2025) with 7,500 examples from the training split of MATH (Hendrycks et al., 2021).

Table 4: Training hyperparameter settings.

Hyperparameters	Value
Learning rate	1e-6
Rollout group number	16
Max context length	4096
Rollout batch size	384
Train batch size	128
Mini batch size	128
Micro batch size per GPU	16
Rollout temperature	1.0
Rollout top p	0.7
KL coefficient	0.0
Clipping upper bound	0.28
Clipping lower bound	0.2

Table 5: Evaluation benchmarks.

Benchmark	Number of prompts
AIME24	30
AIME25	30
AMC	83
Olym.	675
Gaokao	385
MATH.	500

Table 6: Evaluation hyperparameter settings.

Hyperparameters	Value
Rollout temperature	1.0
Rollout top p	0.7
Max context length	4096
Number of rollout per prompt	32

MATH500 contains 500 examples from the MATH test split and is therefore disjoint from the MATH training data used in our experiments.

We found no overlap between our training corpus and AIME24, AIME25, or Gaokao. Across AMC and OlympiadBench, we identified 12 overlapping evaluation examples, all present in the publicly released DAPO dataset. The DAPO dataset has been widely adopted in prior RL studies (Wang et al., 2025; Zheng et al., 2025b). The identified overlap accounts for only a small fraction of the evaluation examples in these two benchmarks and is expected to have a negligible impact on the overall evaluation results.

C Robustness Across Seeds

We assess robustness to randomness by comparing our method with baseline method on Qwen3-8B with three different random seeds while keeping all the other settings fixed. Tab. 7 reports the mean and standard deviation across seeds alongside the results in Tab. 1.

Both TTS and Hierarchical-R1 outperform Beyond in mean Avg@32 on all six benchmarks, with average gains of 3.898 and 3.909 percentage points, respectively. These gains are consistent with the main results and support the robustness of the improvements.

Table 7: Comparison between our method with baseline on Qwen3-8B with one and three random seeds. "Single" reproduces Tab. 1; "Three seeds" reports the mean ± standard deviation of Avg@32 across three seeds.

	Beyond		TTS (Ours)		Hierarchical-R1 (Ours)	
Benchmark	Single	Three seeds	Single	Three seeds	Single	Three seeds
AIME24	43.12	43.401 ± 0.547	46.67	46.251 ± 0.451	46.56	46.944 ± 0.275
AIME25	30.83	30.103 ± 0.695	37.92	37.397 ± 0.737	38.96	39.167 ± 0.169
AMC	71.73	71.643 ± 0.209	77.96	77.924 ± 0.275	77.23	77.451 ± 0.167
Olym.	58.42	58.397 ± 0.109	61.96	61.805 ± 0.110	60.90	60.844 ± 0.062
Gaokao	80.58	80.849 ± 0.206	83.19	83.189 ± 0.047	82.47	82.459 ± 0.030
MATH.	85.05	84.963 ± 0.062	86.17	86.184 ± 0.054	86.04	85.949 ± 0.114
AVG	61.62	61.560 ± 0.197	65.65	65.458 ± 0.248	65.36	65.469 ± 0.079

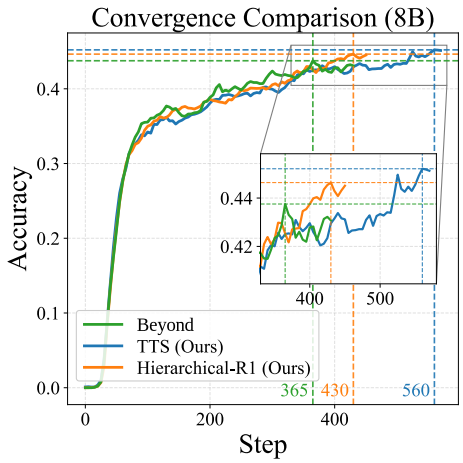


Figure 7: Convergence comparison of Beyond, TTS, and Hierarchical-R1 on Qwen3-8B. Curves are smoothed with an exponential moving average (span 6). Dashed lines mark the peak smoothed accuracy and its training step within the displayed interval.

D Case Study

In this section, representative cases are presented in Fig. 8-Fig. 16 to validate the distinct patterns between positive and negative trajectories. Specifically, rollout responses during training are illustrated, including both positive and negative trajectories generated from the same prompt. The top 10% high-entropy tokens are highlighted in red. Results reveal a strong correlation between high-entropy tokens and logical connectors (e.g., "wait", "alternatively"), in both positive and negative trajectories, suggesting their role in reasoning transitions and continuation. Nevertheless, the two trajectory types exhibit markedly different behaviors: negative trajectories tend to fall into infinite reasoning path switching loops until exhausting the context budget, whereas positive trajectories converge to correct solutions.

E Additional Convergence Analysis

Tab. 2 is designed to isolate the contribution of token-guided sample selection. TTS and Hierarchical-R1 share the same token-selection mechanism and differ only in whether token-guided sample selection is applied. This controlled comparison shows that adding sample selection reduces the reported training steps and GPU hours while maintaining comparable average accuracy (Tab. 1).

Fig. 7 further compares the training curves of Beyond, TTS, and Hierarchical-R1 on Qwen3-8B. TTS achieves a higher peak accuracy than Beyond but reaches its peak later (step 560 versus 365 in the displayed smoothed curves). This pattern may reflect more sustained exploration under the higher policy entropy maintained by TTS. Adding token-guided sample selection reduces this step count to 430, narrowing the gap relative to Beyond while maintaining peak accuracy comparable to TTS and higher than Beyond.

Question：
Compute $\dbinom{8}{4}$

Negative Trajectory:
<think>\nOkay, so I need to compute the binomial coefficient, which is written as $\\dbinom{8}{4}$. Let me recall what that means. The binomial coefficient $\\dbinom{n}{k}$ represents the number of ways to choose k elements from a set of n elements without considering the order. The formula for this is $\\dbinom{n}{k} = \\frac{n!}{k!(n - k)!}$, right?\n\nAlright, so applying that formula to $\\dbinom{8}{4}$, I should plug in n = 8 and k = 4. Let me write that out:\n\n$$\n\\dbinom{8}{4} = \\frac{8!}{4!(8 - 4)!} = \\frac{8!}{4!4!}\n$$\n\nNow, I need to compute the factorials. Let me remember that 8! means 8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4 \u00d7 3 \u00d7 2 \u00d7 1, but maybe I can simplify this before calculating everything. Since both the numerator and the denominator have 4! terms, maybe I can cancel them out or compute them step by step.\n\nFirst, let me compute 8!:\n\n8! = 8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4 \u00d7 3 \u00d7 2 \u00d7 1\n\nBut instead of calculating the entire factorial, maybe I can write 8! as 8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4! because 4! is part of the 8!.\n\nSo:\n\n8! = 8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4!\n\nTherefore, substituting back into the equation:\n\n$$\n\\dbinom{8}{4} = \\frac{8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4!}{4! \u00d7 4!}\n$$\n\nAh, here I can cancel out the 4! in the numerator and the denominator:\n\n$$\n\\dbinom{8}{4} = \\frac{8 \u00d7 7 \u00d7 6 \u00d7 5}{4!}\n$$\n\nNow, compute 4!:\n\n4! = 4 \u00d7 3 \u00d7 2 \u00d7 1 = 24\n\nSo now the equation becomes:\n\n$$\n\\dbinom{8}{4} = \\frac{8 \u00d7 7 \u00d7 6 \u00d7 5}{24}\n$$\n\nNow, let me compute the numerator step by step. Let's calculate 8 \u00d7 7 first:\n\n8 \u00d7 7 = 56\n\nThen multiply that by 6:\n\n56 \u00d7 6 = 336\n\nThen multiply by 5:\n\n336 \u00d7 5 = 1680\n\nSo the numerator is 1680.\n\nNow divide that by the denominator, which is 24:\n\n1680 \u00f7 24\n\nHmm, let me do this division. Let me see how many times 24 goes into 1680.\n\nFirst, maybe simplify the fraction. Let's see if 1680 and 24 have common factors.\n\nDivide numerator and denominator by 24:\n\nWait, 24 \u00d7 70 = 1680?\nLet me check:\n\n24 \u00d7 70 = 24 \u00d7 7 \u00d7 10 = 168 \u00d7 10 = 1680. Yes!\nSo 24 \u00d7 70 = 1680. Therefore, 1680 \u00f7 24 = 70.\n\nSo, $\\dbinom{8}{4} = 70$.\n\nWait, let me verify that again to make sure I didn't make a mistake. Let me check the calculation again step by step.\n\nStarting with $\\frac{8 \u00d7 7 \u00d7 6 \u00d7 5}{24}$:\n\nFirst, 8 \u00d7 7 = 56\n\n56 \u00d7 6 = 336\n\n336 \u00d7 5 = 1680\n\nYes, that's correct.\n\nThen 1680 divided by 24. Let me divide 1680 by 24:\n\nAlternatively, divide numerator and denominator by 8 first:\n\n1680 \u00f7 8 = 210\n\n24 \u00f7 8 = 3\n\nSo now, it's 210 \u00f7 3 = 70. Yep, same result. So that's correct.\n\nAlternatively, maybe I can compute it step by step:\n\nDivide 1680 by 24. Since 24 \u00d7 10 = 240, 24 \u00d7 70 = 24 \u00d7 7 \u00d7 10 = 168 \u00d7 10 = 1680. Correct.\n\nTherefore, the answer is 70.\n\nWait, just to make sure, I remember that $\\dbinom{8}{4}$ is a known value. Let me think, for example, $\\dbinom{8}{4}$ is the middle term in the 8th row of Pascal's triangle. Let me recall that the 8th row (if we start counting from row 0) would

Figure 8: Case 1.

be:\n\nRow 0: 1\n\nRow 1: 1 1\n\nRow 2: 1 2 1\n\nRow 3: 1 3 3 1\n\nRow 4: 1 4 6 4 1\n\nRow 5: 1 5 10 10 5 1\n\nRow 6: 1 6 15 20 15 6 1\n\nRow 7: 1 7 21 35 35 21 7 1\n\nRow 8: 1 8 28 56 70 56 28 8 1\n\nWait, so row 8 (which is the 9th row if counting from 1) has entries 1, 8, 28, 56, 70, 56, 28, 8, 1. Therefore, the middle term is 70. So yes, $\\dbinom{8}{4} = 70$. So that confirms my earlier calculation.\n\nAlternatively, maybe I can compute it using another method, like multiplying step by step and simplifying fractions before multiplying.\n\nLet me try that approach. Starting with $\\frac{8 \u00d7 7 \u00d7 6 \u00d7 5}{4 \u00d7 3 \u00d7 2 \u00d7 1}$.\n\nLet me write that as:\n\n(8/4) \u00d7 (7/ something) \u00d7 ... Wait, maybe simplify step by step.\n\nFirst, note that 8 divided by 4 is 2. So:\n\n(8/4) \u00d7 7 \u00d7 6 \u00d7 5 / (3 \u00d7 2 \u00d7 1)\n\nWait, let me do that:\n\nThe denominator is 4! = 24. So maybe break down the numerator and denominator into factors and cancel.\n\nNumerator factors: 8, 7, 6, 5.\n\nDenominator factors: 4, 3, 2, 1.\n\nLet me write them all out:\n\nNumerator: 8 \u00d7 7 \u00d7 6 \u00d7 5\n\nDenominator: 4 \u00d7 3 \u00d7 2 \u00d7 1\n\nLet me factor each number into primes:\n\nNumerator:\n\n8 = 2\u00b3\n\n7 = 7\n\n6 = 2 \u00d7 3\n\n5 = 5\n\nSo overall numerator factors: 2\u00b3 \u00d7 7 \u00d7 2 \u00d7 3 \u00d7 5 = 2\u2074 \u00d7 3 \u00d7 5 \u00d7 7\n\nDenominator:\n\n4 = 2\u00b2\n\n3 = 3\n\n2 = 2\n\n1 = 1\n\nSo denominator factors: 2\u00b2 \u00d7 3 \u00d7 2 \u00d7 1 = 2\u00b3 \u00d7 3\n\nTherefore, divide numerator by denominator:\n\n(2\u2074 \u00d7 3 \u00d7 5 \u00d7 7) / (2\u00b3 \u00d7 3) = (2^(4-3) \u00d7 3^(1-1) \u00d7 5 \u00d7 7) = 2\u00b9 \u00d7 1 \u00d7 5 \u00d7 7 = 2 \u00d7 5 \u00d7 7 = 10 \u00d7 7 = 70. Same result. That's a good check.\n\nAnother way to think about $\\dbinom{8}{4}$ is that it's equal to $\\dbinom{8}{4}$, and since $\\dbinom{n}{k} = \\dbinom{n}{n - k}$, so $\\dbinom{8}{4} = \\dbinom{8}{4}$. Wait, that's trivial here, but maybe if I had a different number, but in this case, it's the same. However, if I wanted to compute it as $\\dbinom{8}{4}$, maybe I can think of the multiplicative formula:\n\n$$\n\\dbinom{n}{k} = \\frac{n \u00d7 (n - 1) \u00d7 ... \u00d7 (n - k + 1)}{k!}\n$$\n\nSo for $\\dbinom{8}{4}$, that would be:\n\n(8 \u00d7 7 \u00d7 6 \u00d7 5) / (4 \u00d7 3 \u00d7 2 \u00d7 1) which is again the same as before. So same steps, which we've already calculated as 70.\n\nAlternatively, maybe use the multiplicative formula step by step:\n\nStarting with 8/4 = 2\n\nThen multiply by 7/3 (since next term is 7 in numerator and next denominator term is 3? Wait, let me think. Let me write the formula again:\n\n$$\n\\dbinom{8}{4} = \\frac{8}{4} \u00d7 \\frac{7}{3} \u00d7 \\frac{6}{2} \u00d7 \\frac{5}{1}\n$$\n\nWait, how does that work?\n\nActually, the multiplicative formula is:\n\n$$\n\\dbinom{n}{k} = \\prod_{i=1}^k \\frac{n - k + i}{i}\n$$\n\nSo for $\\dbinom{8}{4}$, that would be:\n\nFor i = 1: (8 - 4 + 1)/1 = 5/1\n\ni = 2: (8 - 4 + 2)/2 = 6/2\n\ni = 3: (8 - 4 + 3)/3 = 7/3\n\ni = 4: (8 - 4 + 4)/4 = 8/4\n\nSo multiplying these together:\n\n(5/1) \u00d7 (6/2) \u00d7 (7/3) \u00d7 (8/4)\n\nLet me compute each term:\n\n5/1 = 5\n\n6/2 = 3\n\n7/3 = 7/3\n\n8/4 = 2\n\nSo multiply them: 5 \u00d7 3 = 15, then 15 \u00d7 (7/3) = (15/3) \u00d7 7 = 5 \u00d7 7 = 35, then 35 \u00d7 2 = 70. There we go, same

Figure 9: Case 1 (Cont.).

Positive Trajectory:

<think>\nOkay, so I need to compute the binomial coefficient $\\dbinom{8}{4}$. Hmm, binomial coefficients... right, they're used in combinations, right? Like, the number of ways to choose 4 items out of 8 without considering the order. Let me recall the formula for that. \n\nI think the formula for $\\dbinom{n}{k}$ is $\\frac{n!}{k!(n - k)!}$. Let me check that. Yeah, factorial of n divided by the product of factorial k and factorial (n - k). So in this case, n is 8 and k is 4. \n\nLet me write that down: $\\dbinom{8}{4} = \\frac{8!}{4!(8 - 4)!}$. Simplifying the denominator first, 8 - 4 is 4, so it becomes $\\frac{8!}{4!4!}$. \n\nNow, I need to compute 8 factorial, which is 8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4 \u00d7 3 \u00d7 2 \u00d7 1. But wait, maybe I don't need to compute the entire factorials because there's cancellation possible. Let me think. Since the denominator is 4! \u00d7 4!, maybe I can simplify the fractions before multiplying everything out. \n\nLet me write out 8! as 8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4! because 4! is part of 8! So 8! = 8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4!. Therefore, substituting that into the equation, we have $\\frac{8 \u00d7 7 \u00d7 6 \u00d7 5 \u00d7 4!}{4! \u00d7 4!}$. \n\nOh, right! The 4! in the numerator and denominator cancels out. So that simplifies to $\\frac{8 \u00d7 7 \u00d7 6 \u00d7 5}{4!}$. Now, 4! is 4 \u00d7 3 \u00d7 2 \u00d7 1, which is 24. So now the problem becomes $\\frac{8 \u00d7 7 \u00d7 6 \u00d7 5}{24}$. \n\nLet me compute the numerator first. 8 \u00d7 7 is 56. Then 56 \u00d7 6 is 336. Then 336 \u00d7 5 is 1680. So the numerator is 1680. Then divide that by 24. Let me compute 1680 \u00f7 24. \n\nHmm, let me see. 24 \u00d7 70 is 1680, because 24 \u00d7 70 = 24 \u00d7 7 \u00d7 10 = 168 \u00d7 10 = 1680. So 1680 divided by 24 is 70. \n\nWait, so does that mean $\\dbinom{8}{4}$ is 70? Let me double-check my steps to make sure I didn't make a mistake. \n\nStarting with the formula: $\\frac{8!}{4!4!}$. Then I rewrote 8! as 8\u00d77\u00d76\u00d75\u00d74! which is correct. Then canceled the 4! in numerator and denominator, leaving 8\u00d77\u00d76\u00d75 / 4!. Then calculated numerator as 1680 and denominator as 24. 1680 / 24 equals 70. That seems right. \n\nAlternatively, I can compute the combination using another method. For example, $\\dbinom{n}{k}$ is equal to $\\dbinom{n}{n - k}$. So $\\dbinom{8}{4}$ is the same as $\\dbinom{8}{4}$, which is the same as $\\dbinom{8}{4}$. Wait, that doesn't help. Maybe another way. Let me compute step by step. \n\nAlternatively, I can compute the product $\\frac{8}{4} \\times \\frac{7}{3} \\times \\frac{6}{2} \\times \\frac{5}{1}$. Wait, how does that work? Because the formula for combinations can be written as $\\frac{n(n - 1)(n - 2)...(n - k + 1)}{k!}$. So for $\\dbinom{8}{4}$, it's $\\frac{8 \u00d7 7 \u00d7 6 \u00d7 5}{4 \u00d7 3 \u00d7 2 \u00d7 1}$. Which is the same as before. Let me compute that. \n\nSo numerator: 8 \u00d7 7 = 56, 56 \u00d7 6 = 336, 336 \u00d7 5 = 1680. Denominator: 4 \u00d7 3 = 12, 12 \u00d7 2 = 24, 24 \u00d7 1 = 24. Then 1680 / 24 = 70. Same result. \n\nAlternatively, maybe compute step by step, simplifying as I go. Let me try that. \n\nStarting with $\\frac{8 \u00d7 7 \u00d7 6 \u00d7 5}{4 \u00d7 3 \u00d7 2 \u00d7 1}$. \n\nFirst, let me simplify 8 divided by 4. That's 2. So now I have 2 \u00d7 7 \u00d7 6 \u00d7 5 divided by 3 \u00d7 2 \u00d7 1. \n\nThen, take 6 divided by 3. That's 2. Now the equation is 2 \u00d7 7 \u00d7 2 \u00d7 5 divided by 2 \u00d7 1. \n\nWait, let me

Figure 10: Case 1 (Cont.).

redo that step. After simplifying 8 / 4 = 2, so numerator becomes 2 \u00d7 7 \u00d7 6 \u00d7 5. Denominator is 4 \u00d7 3 \u00d7 2 \u00d7 1. But since I already simplified 8 / 4 = 2, then denominator becomes 3 \u00d7 2 \u00d7 1. Wait, no, maybe not. Let me think again. \n\nActually, when simplifying fractions, we can cancel factors in numerator and denominator. So 8/4 is 2, and then the denominator's 4 is gone. But the denominator still has 3 \u00d7 2 \u00d7 1. So perhaps another approach. Let me write the entire fraction:\n\n(8 \u00d7 7 \u00d7 6 \u00d7 5) / (4 \u00d7 3 \u00d7 2 \u00d7 1)\n\nWe can cancel the 8 and 4: 8/4 = 2. So now it's (2 \u00d7 7 \u00d7 6 \u00d7 5) / (3 \u00d7 2 \u00d7 1). \n\nThen, notice that there's a 6 in the numerator and a 3 in the denominator. 6 / 3 = 2. So now it's (2 \u00d7 7 \u00d7 2 \u00d7 5) / (2 \u00d7 1). \n\nThen, there's a 2 in the numerator and a 2 in the denominator. 2 / 2 = 1. So now it's (1 \u00d7 7 \u00d7 2 \u00d7 5) / 1. Which is 7 \u00d7 2 \u00d7 5 = 70. \n\nThat's another way to get 70. So that seems consistent. \n\nAlternatively, let me use Pascal's triangle. The 8th row (if we start counting from row 0) would have the numbers 1, 8, 28, 56, 70, 56, 28, 8, 1. So the 4th entry (since it's 0-indexed) would be 70. That also matches. \n\nSo all methods are pointing to 70. \n\nWait, but just to make sure, maybe I can compute it using the multiplicative formula for combinations. \n\nThe formula is $\\dbinom{n}{k} = \\frac{n}{k} \\times \\frac{n - 1}{k - 1} \\times \\dots \\times \\frac{n - k + 1}{1}$. \n\nSo for $\\dbinom{8}{4}$, that would be (8/4) \u00d7 (7/3) \u00d7 (6/2) \u00d7 (5/1). Let me compute each term:\n\nFirst term: 8/4 = 2\n\nSecond term: 7/3 \u2248 2.333...\n\nThird term: 6/2 = 3\n\nFourth term: 5/1 = 5\n\nMultiplying them all together: 2 \u00d7 (7/3) \u00d7 3 \u00d7 5. \n\nNotice that 2 \u00d7 3 = 6, then 6 \u00d7 3 = 18? Wait, let me compute step by step:\n\nFirst, 2 \u00d7 (7/3) = 14/3\n\nThen, 14/3 \u00d7 3 = 14\n\nThen, 14 \u00d7 5 = 70. \n\nYes, that's 70. So that works. \n\nSo all methods lead to the same answer. Therefore, I'm confident that $\\dbinom{8}{4}$ is 70.\n\n**Final Answer**\nAnswer: 70\n</think>\n\nTo compute the binomial coefficient $\\dbinom{8}{4}$, we use the standard formula for combinations:\n\n$$\n\\dbinom{n}{k} = \\frac{n!}{k!(n - k)!}\n$$\n\n---\n\n## Step 1: Apply the Formula\n\nGiven $ n = 8 $ and $ k = 4 $, we have:\n\n$$\n\\dbinom{8}{4} = \\frac{8!}{4!(8 - 4)!} = \\frac{8!}{4!4!}\n$$\n\n---\n\n## Step 2: Simplify Factorials\n\nRecall that:\n- $ 8! = 8 \\times 7 \\times 6 \\times 5 \\times 4! $\n- So we can write:\n$$\n\\dbinom{8}{4} = \\frac{8 \\times 7 \\times 6 \\times 5 \\times 4!}{4! \\times 4!}\n$$\n\nNow cancel out the $ 4! $ in the numerator and denominator:\n\n$$\n\\dbinom{8}{4} = \\frac{8 \\times 7 \\times 6 \\times 5}{4!}\n$$\n\nWe compute $ 4! = 4 \\times 3 \\times 2 \\times 1 = 24 $\n\n---\n\n## Step 3: Compute the Numerator\n\n$$\n8 \\times 7 = 56 \\\\\n56 \\times 6 = 336 \\\\\n336 \\times 5 = 1680\n$$\n\nSo the numerator is 1680.\n\n---\n\n## Step 4: Divide by Denominator\n\n$$\n\\frac{1680}{24} = 70\n$$\n\n---\n\n## Conclusion\n\nThus, the value of the binomial coefficient $\\dbinom{8}{4}$ is:\n\n**Answer**: 70

Figure 11: Case 1 (Cont.).

Question：
The truncated right circular cone has a large base radius 8 cm and a small base radius of 4 cm. The height of the truncated cone is 6 cm. How many $\\text{cm}^3$ are in the volume of this solid? [asy]\n\nimport olympiad; size(150); defaultpen(linewidth(0.8)); dotfactor=4;\n\ndraw(ellipse((0,0),4,1)); draw(ellipse((0,3),2,1/2));\n\ndraw((-3.97,.1)--(-1.97,3.1)^^(3.97,.1)--(1.97,3.1));\n\n[/asy]

Negative Trajectory:
<think>\nOkay, so I need to find the volume of this truncated right circular cone, which is also called a frustum. Let me remember how to calculate that. Hmm... I think the formula for the volume of a frustum is similar to the volume of a cone, but adjusted because it's cut off. Let me recall... I think it's something like the average of the two bases times the height? Wait, no, that's for a prism or something else. For a cone, the volume is (1/3)\u03c0r\u00b2h. So maybe for the frustum, it's the difference between two cones? \n\nLet me think. If I imagine the original cone before it was truncated, and then the part that was cut off is another smaller cone. So, the volume of the frustum would be the volume of the original cone minus the volume of the smaller cone that was removed. That makes sense. But how do I find the dimensions of the original cone?\n\nAlright, let me denote the large base radius as R, which is 8 cm, and the small base radius as r, which is 4 cm. The height of the frustum is given as h = 6 cm. So, if I can find the height of the original cone and the height of the smaller cone that was removed, then I can compute the volume.\n\nBut how do I relate these heights? I remember that similar cones have their dimensions proportional. Since the frustum is created by cutting the original cone with a plane parallel to the base, the smaller cone that's removed is similar to the original cone. Therefore, the ratio of their radii should be equal to the ratio of their heights. \n\nLet me denote H as the height of the original cone, and h' as the height of the smaller cone. Then, since the radii are in the ratio R/r = 8/4 = 2. So, the ratio of similarity is 2. Therefore, the heights should also be in the same ratio. That is, H / h' = 2 / 1. \n\nBut also, the height of the frustum is the difference between the heights of the original cone and the smaller cone. So, H - h' = 6 cm. \n\nSo, if H = 2h', then substituting into the second equation: 2h' - h' = h' = 6 cm. Therefore, h' = 6 cm, and H = 12 cm. Wait, let me check that again. \n\nIf the ratio of the radii is R/r = 2, then the ratio of the heights is also 2. So, the original cone's height is twice the height of the smaller cone. Let me call the height of the smaller cone h'. Then, the original height is 2h'. Then, the height of the frustum is 2h' - h' = h' = 6 cm. Therefore, h' = 6 cm, so the original cone's height H is 12 cm. \n\nOkay, that seems correct. So, then the volume of the frustum would be the volume of the original cone minus the volume of the smaller cone. \n\nLet me compute that. The volume of the original cone is (1/3)\u03c0R\u00b2H = (1/3)\u03c0*(8)^2*12. Let me calculate that. 8 squared is 64, times 12 is 768. Then divided by 3 is 256. So, 256\u03c0 cm\u00b3. \n\nThen the volume of the smaller cone is (1/3)\u03c0r\u00b2h' = (1/3)\u03c0*(4)^2*6. 4 squared is 16, times 6 is 96, divided

Figure 12: Case 2.

by 3 is 32. So, 32\u03c0 cm\u00b3. \n\nTherefore, the volume of the frustum is 256\u03c0 - 32\u03c0 = 224\u03c0 cm\u00b3. \n\nBut wait, let me check if there's a direct formula for the volume of a frustum. Maybe I can cross-verify. \n\nI think the formula for the volume of a frustum is (1/3)\u03c0h(R\u00b2 + Rr + r\u00b2). Let me recall. Yes, that sounds right. Let me use this formula to check my answer. \n\nGiven R = 8, r = 4, h = 6. So, plugging into the formula: (1/3)\u03c0*6*(8\u00b2 + 8*4 + 4\u00b2). Let me compute that. \n\nFirst compute 8\u00b2 = 64, 8*4 = 32, 4\u00b2 = 16. Adding them together: 64 + 32 + 16 = 112. Then, multiply by 6: 112*6 = 672. Then divide by 3: 672 / 3 = 224. Therefore, 224\u03c0 cm\u00b3. \n\nOkay, same result. So my answer is 224\u03c0. Wait, but the question says \"how many cm\u00b3 are in the volume of this solid?\" and they don't specify whether to leave it in terms of \u03c0 or to use a numerical approximation. But since the problem mentions \"Answer: $Answer\", and in math problems like this, especially with \u03c0, it's usually acceptable to leave it as \u03c0 multiplied by the coefficient. Let me check the problem again.\n\nWait, the Asymptote code is there, but maybe it's just for illustration. The problem says \"how many cm\u00b3 are in the volume of this solid,\" and in the instructions, they mention to put the answer in the form Answer: $Answer. But since they didn't specify, but given that in most math competitions and problems, unless specified, if it's a multiple of \u03c0, you leave it as is. However, sometimes they want a numerical value. Wait, let me check if I did the problem correctly.\n\nBut let me think again. Let me check the formula for the frustum again. The formula is indeed (1/3)\u03c0h(R\u00b2 + Rr + r\u00b2). So, substituting the numbers:\n\n(1/3)*\u03c0*6*(64 + 32 + 16) = 2\u03c0*(112) = 224\u03c0. So that's 224\u03c0 cm\u00b3. So if the answer requires a numerical value, they would have said \"in terms of \u03c0\" or \"numerically approximate,\" but since it's not specified, and given that the problem is likely expecting an exact value, 224\u03c0 is the exact volume. However, sometimes problems expect a numerical value, but given that the Asymptote code is given, and the problem is from a math competition style, it's more likely that 224\u03c0 is the answer. Wait, but let me check again if my calculation is correct.\n\nAlternatively, maybe I made a mistake in calculating the original cone's dimensions? Let me check that again. The height of the frustum is 6 cm. The ratio of the radii is 2:1, so the height ratio is also 2:1. So if the original cone's height is H, then the smaller cone's height is H - 6. Wait, but earlier I said that H = 12, and H - h' = 6. But if h' is the height of the smaller cone, then H - h' = 6. \n\nBut if the ratio of the heights is H / h' = 2, then H = 2h'. Therefore substituting into H - h' = 6, 2h' - h' = h' = 6. Therefore h' = 6, so H = 12. Therefore, original cone's height is 12, smaller cone's height is 6. \n\nThen, the volume of original cone is (1/3)\u03c0*(8)^2*12 = (1/3)*\u03c0*64*12. 64*12 = 768. Divided by 3 is 256\u03c0. Then the smaller cone is (1/3)*\u03c0*(4)^2*6 = (1/3)*\u03c0*16*6 = 32\u03c0. Then 256\u03c0 - 32\u03c0 = 224\u03c0. So that's correct. \n\nAlternatively, using the formula (1/3)\u03c0h(R\u00b2 + Rr + r\u00b2). Wait, h in this formula is the height of the frustum, which is 6. So yes, that's 6. So that's correct. \n\nTherefore, 224\u03c0 cm\u00b3. So unless there's a miscalculation here, that should be the answer. But let me think again about the formula. \n\nAnother way to think about the formula

Figure 13: Case 2 (Cont.).

for the frustum is that the volume is the average of the areas of the two bases multiplied by the height. Wait, no, that would be the case for a prism or a cylinder. But for a cone, the volume is (1/3)\u03c0r\u00b2h, so for the frustum, the formula isn't just the average of the areas times height. However, the formula that we used earlier is correct. Let me verify with another approach.\n\nAlternatively, think of integrating the area of circular cross-sections. The radius of the frustum varies linearly from R to r over a height h. Let me model this. \n\nLet me set up a coordinate system where the origin is at the base of the frustum. So, the radius at height 0 is R = 8 cm, and at height h = 6 cm, the radius is r = 4 cm. The radius as a function of height y would be a linear function. Let me find the equation of the radius as a function of y. \n\nThe radius decreases from 8 to 4 over a height of 6 cm. The slope of the radius with respect to height is (4 - 8)/6 = (-4)/6 = -2/3. Therefore, the equation for the radius r(y) is R(y) = 8 - (2/3)y. \n\nThen, the volume can be calculated by integrating the area of each cross-section from y = 0 to y = 6. \n\nSo, Volume = \u222b[0 to 6] \u03c0[R(y)]\u00b2 dy = \u03c0 \u222b[0 to 6] [8 - (2/3)y]\u00b2 dy.\n\nLet me compute that integral. \n\nFirst expand the square:\n\n[8 - (2/3)y]^2 = 64 - 2*8*(2/3)y + (2/3 y)^2 = 64 - (32/3)y + (4/9)y\u00b2.\n\nTherefore, the integral becomes \u03c0 \u222b0^6 [64 - (32/3)y + (4/9)y\u00b2] dy. \n\nCompute term by term:\n\nIntegral of 64 dy from 0 to 6 is 64y evaluated from 0 to 6: 64*6 = 384.\n\nIntegral of -(

Positive Trajectory:

<think>\nOkay, so I need to find the volume of this truncated right circular cone, which is also called a frustum. Let me recall the formula for the volume of a frustum. I think it's something like the average of the two bases times the height? Wait, but I need to be precise here. Let me think.\n\nI remember that the volume of a frustum can be calculated using the formula: \n\nV = (1/3) * \u03c0 * h * (R\u00b2 + Rr + r\u00b2)\n\nWhere R is the radius of the larger base, r is the radius of the smaller base, and h is the height of the frustum. Let me verify this formula. \n\nAlternatively, another way to think about it is that a frustum is basically a large cone minus a smaller cone that's been cut off. So maybe I can compute the volume of the original large cone and then subtract the volume of the smaller cone that was removed. That might be a good approach too. Let me try both methods to check if they give the same answer.\n\nFirst, let's try the direct formula. The problem gives me the large base radius R = 8 cm, small base radius r = 4 cm, and height h = 6 cm. Plugging these into the formula:\n\nV = (1/3) * \u03c0 * 6 * (8\u00b2 + 8*4 + 4\u00b2)\n\nLet me compute each part step by step.\n\nFirst, calculate the terms inside the parentheses:\n\n8\u00b2 = 64\n\n8*4 = 32\n\n4\u00b2 = 16\n\nAdding them up: 64 + 32 + 16 = 112\n\nThen multiply by h, which is 6:\n\n112 * 6 = 672\n\nThen multiply by 1/3 and \u03c0:\n\nV = (1/3) * \u03c0 * 672 = 224 * \u03c0 cm\u00b3\n\nWait, so that would be 224\u03c0 cm\u00b3. Is this correct? Let me check with the other method to be sure.\n\nSo, the other approach is to think of the frustum as the difference between two cones. Let me denote the original large cone with base radius R = 8 cm and height H, and then the smaller

Figure 14: Case 2 (Cont.).

cone that was cut off has base radius r = 4 cm and height H - h, where h is the height of the frustum, which is 6 cm. Wait, but I need to find H first?\n\nBut how do I find H? Because the height of the frustum is 6 cm, which is the difference between the height of the original cone and the smaller cone.\n\nBut since the cones are similar (since the frustum is a truncated cone), the ratio of their radii is equal to the ratio of their heights. So, the ratio of the radii R/r = 8/4 = 2. Therefore, the heights should be in the same ratio. Let me denote the height of the smaller cone as h', then the height of the original cone is H = h' + 6 cm. Also, since the ratio of radii is 2, we have R/r = H / h' => 8/4 = (h' + 6)/h'\n\nSo, 2 = (h' + 6)/h'\n\nSolving for h':\n\n2h' = h' + 6 => 2h' - h' = 6 => h' = 6 cm\n\nTherefore, the original height H = h' + 6 = 6 + 6 = 12 cm.\n\nTherefore, the volume of the original large cone is (1/3)\u03c0R\u00b2H = (1/3)\u03c0*(8)\u00b2*12\n\nLet me compute that:\n\n(1/3) * \u03c0 * 64 * 12\n\nFirst compute 64 * 12 = 768\n\nThen divide by 3: 768 / 3 = 256\n\nSo the volume of the original cone is 256\u03c0 cm\u00b3.\n\nNow, the volume of the smaller cone that was cut off is (1/3)\u03c0r\u00b2h', where h' is 6 cm. So:\n\n(1/3)\u03c0*(4)\u00b2*6\n\nCalculate 4\u00b2 = 16\n\nMultiply by 6: 16 * 6 = 96\n\nDivide by 3: 96 / 3 = 32\n\nSo the volume of the smaller cone is 32\u03c0 cm\u00b3.\n\nTherefore, the volume of the frustum is the difference between the two:\n\n256\u03c0 - 32\u03c0 = 224\u03c0 cm\u00b3\n\nWhich matches the result from the first method. So that gives me confidence that the answer is 224\u03c0 cm\u00b3. Let me check once again with the formula to make sure I didn't make any calculation errors.\n\nOriginal formula: (1/3)\u03c0h(R\u00b2 + Rr + r\u00b2). Plugging in:\n\n(1/3)\u03c0*6*(64 + 32 + 16) = 2\u03c0*(112) = 224\u03c0. Yes, that's correct. \n\nAlternatively, if I didn't remember the formula, but I could derive it by using the difference between the two cones. Since that method also gave the same result, I think the answer is correct.\n\nSo, the volume is 224\u03c0 cubic centimeters. But let me check if the problem expects a numerical value or if it's okay with \u03c0. The problem says \"how many cm\u00b3 are in the volume\", and since it's a math competition problem, they might expect the answer in terms of \u03c0. However, looking at the Asymptote code and the problem statement, they might want a numerical multiple of \u03c0. \n\nWait, in the problem statement, they don't specify whether to leave it in terms of \u03c0 or to use a numerical approximation. But since all the given values are integers and it's a standard frustum volume problem, the answer is likely expected to be in terms of \u03c0. The Asymptote code is just for drawing, but the problem says \"how many cm\u00b3\", so maybe they want an exact value, so 224\u03c0. \n\nAlternatively, sometimes problems might expect a number, but since \u03c0 is involved, unless told otherwise, I think 224\u03c0 is acceptable. Let me check the problem again.\n\nThe problem says: \"How many cm\u00b3 are in the volume of this solid?\" and the answer needs to be in the form Answer: $Answer. Since the answer is in terms of \u03c0, and they didn't specify to use a decimal approximation, I think 224\u03c0 is correct. \n\nBut maybe the problem expects a numerical value? Wait, maybe I made a mistake in the formula? Let me check again.\n\nWait, the formula for the volume of a frustum is indeed (1/3)\u03c0h(R\u00b2 + Rr + r\u00b2). Alternatively, some might remember it as

Figure 15: Case 2 (Cont.).

(1/3)\u03c0h*(R\u00b2 + Rr + r\u00b2). So that is correct. Let me check with an example. If the frustum is a cylinder, then R = r, and then the formula becomes (1/3)\u03c0h*(R\u00b2 + R\u00b2 + R\u00b2) = (1/3)\u03c0h*3R\u00b2 = \u03c0R\u00b2h, which is correct for a cylinder. So that makes sense. \n\nAlternatively, if the smaller radius is zero, then it's a cone, so the formula becomes (1/3)\u03c0h(R\u00b2 + 0 + 0) = (1/3)\u03c0hR\u00b2, which is correct. Therefore, the formula is correct. \n\nSo, given that, then 224\u03c0 is correct. However, maybe the problem expects an exact value, but in some cases, perhaps they want it multiplied out? Wait, but 224 is already simplified. \n\nAlternatively, let me check if the height is correct. Wait, the height of the frustum is given as 6 cm. In my second method, I considered the height of the original cone as 12 cm, and then subtracted the smaller cone with height 6 cm. So the difference is 12 - 6 = 6 cm, which matches the given height of the frustum. So that's correct.\n\nSo, all steps check out. Therefore, the answer is 224\u03c0 cm\u00b3. Therefore, Answer: 224\u03c0\n\nWait, but let me check again if the problem says \"truncated right circular cone\" which is a frustum. Yes, and the formula is correct. Therefore, I think that's the answer.\n\n**Final Answer**\nAnswer: 224\\pi\n</think>\n\nTo find the volume of a **truncated right circular cone**\u2014also known as a **frustum**\u2014we use the standard formula for the volume of a frustum of a cone:\n\n$$\nV = \\frac{1}{3} \\pi h (R^2 + Rr + r^2)\n$$\n\nWhere:\n- $ R $ is the radius of the larger base,\n- $ r $ is the radius of the smaller base,\n- $ h $ is the vertical height of the frustum.\n\n---\n\n\#\#\# Step 1: Identify Given Values\n\nFrom the problem:\n- Large base radius $ R = 8 $ cm\n- Small base radius $ r = 4 $ cm\n- Height of the frustum $ h = 6 $ cm\n\n---\n\n\#\#\# Step 2: Plug Into the Formula\n\n$$\nV = \\frac{1}{3} \\pi (6) (8^2 + 8 \\cdot 4 + 4^2)\n$$\n\nCompute each part:\n$$\n8^2 = 64, \\quad 8 \\cdot 4 = 32, \\quad 4^2 = 16\n$$\n$$\n64 + 32 + 16 = 112\n$$\n$$\nV = \\frac{1}{3} \\pi \\cdot 6 \\cdot 112 = 2\\pi \\cdot 112 = 224\\pi\n$$\n\n---\n\n\#\#\# Step 3: Final Answer\n\nThe volume of the solid is:\n\n$$\n\\boxed{224\\pi}\n$$\n\nAnswer: 224\\pi

Figure 16: Case 2 (Cont.).